A
Dissertation
On

"A Novel And Improved Approach For Enhancing The Security In Cloud Computing"

Submitted in Partial fulfillment of the requirement
For the award of Degree of

MASTER OF TECHNOLOGY
Computer Science and Engg.
Delhi Technological University, Delhi

SUBMITTED BY

SHAHID KHAN
University Roll No: 2K11/CSE/13

Under the Guidance of:

Ms. RICHA MISHRA

Assistant Professor
Delhi Technological University

DEPARTMENT OF COMPUTER ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
BAWANA ROAD, DELHI-110042
2011-2013

CERTIFICATE

This is to certify that the work contained in this dissertation entitled "**A Novel And Improved Approach For Enhancing The Security In Cloud Computing**" submitted in the partial fulfillment, for the award for the degree of M. Tech. in Computer Science and Engg. at **DELHI TECHNOLOGICAL UNIVERSITY** by **SHAHID KHAN, Roll No. 2K11/CSE/13** is carried out by him under my supervision. This matter embodied in this project work has not been submitted earlier for the award of any degree or diploma in any university/institution to the best of our knowledge and belief.

(Ms. RICHA MISHRA)
Project Guide
Assistant Professor
Department of Computer Engineering
Delhi Technological University

AKNOWLEDGEMENT

First of all, let me thank the almighty god, my parents and my dear friends who are the most graceful and merciful for their blessing that contributed to the successful completion of this project.

I feel privileged to offer sincere thanks and deep sense of gratitude to Ms. RICHA MISHRA, project guide for expressing his confidence in me by letting me work on a project of this magnitude and using the latest technologies and providing their support, help & encouragement in implementing this project.

I would like to take this opportunity to express the profound sense of gratitude and respect to all those who helped us throughout the duration of this project. DELHI TECHNOLOGICAL UNIVERSITY, in particular has been the source of inspiration, I acknowledge the effort of those who have contributed significantly to this project.

Shahid Khan

(Roll No.: 2K11/CSE/13)

TABLE OF CONTENTS

List Of Figures

List Of Tables

Abstract

Cloud computing provides the facility to access shared resources and common infrastructure, offering services on demand over the network to perform operations that meet changing business needs. The location of physical resources and devices being accessed are typically not known to the end user. Data , which is, stored by the user in the cloud has now been placed under the supervision of cloud provider. Users have lost control over their data.

The aim of the thesis is to find how Predicate Based Encryption (PBE) could be used within the Cloud to protect data. Predicate Based Encryption (PBE) is a novel family

of public key encryption schemes that allows for expressive, and fine-grained, access control to be integrated within the cryptographic process. Providing an efficient means to realize distributed encrypted access control.

Predicate encryption (PE) provides both the access control of cipher texts and the privacy of cipher texts is a new paradigm of public-key encryption. An important application of predicate encryption is a searchable encryption system in a cloud storage, where it enables a client to securely outsource its data to an untrusted cloud server and to search over it even without revealing a key-word itself.

Chapter: 1 Introduction

1.1 CLOUD COMPUTING

Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications.

The following definition of cloud computing has been developed by the U.S. National Institute of Standards and Technology (NIST):

> ***Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.***

1.2 PROBLEM STATEMENT

The use of encryption schemes is often described through an analogy depicting the transmission on a plain-text message M from one entity, Alice to another entity, Bob. Here Alice wishes to ensure that only Bob will be able to read M . This analogy has persisted due to its ability to describe a prevalent communication style, that of unicast communication. However, this simple analogy does not necessarily represent the entirety of communication styles that are actively used, it does not take into account multicast communication: What if Alice's wish were to send her message not to Bob but to Bobs plural?

Traditional symmetric and asymmetric encryption schemes can be leveraged to provide Alice with a secure means through which she can send her message. However, with symmetric schemes each recipient will be in a position to decrypt all cipher-texts that have been encrypted with the same key. With asymmetric schemes the encrypting entity needs to explicitly state for whom decryption is permissible .To reference the different styles of communication, symmetric schemes represent broadcast communication and asymmetric schemes unicast communication. A multicast encryption scheme is required that allows for a more expressive fine-grained means through which Alice can specify access over her data.

1.3 SCOPE OF WORK

The investigation was divided broadly into three stages.

Data Security and the Cloud :: The initial stage sought to provide a clear definition for Cloud Computing and the security issues therein, looking to identify precisely where and when threats can occur to data and how these threats ought to be mitigated.

Predicate Based Encryption :: The next stage focused solely upon PBE schemes discussing how they work and what they allow for. This provided a foundation upon which their deployment as part of a crypto-system could be explored and to define the types of problem that PBE schemes can be used to solve.

Leveraging PBE :: The final stage of the investigation built upon, and combined the results, of the previous stages. Here the investigation looked to determine the problems that PBE schemes can be used to solve within the Cloud, and the quality of solution provided

1.4 ORGANISATION OF THESIS

Chapter 2 deals with literature review of the topic.

Chapter 3 gives research background and data security requirements.

Chapter 4 introduces our proposed approach.

Chapter 5 deals with experimental results of the system.

Chapter 6 finally concludes the work.

Chapter: 2

Literature Review

2.1 Cloud Computing

Cloud computing [INT+01] is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications.

The following definition of cloud computing has been developed by the U.S. National Institute of Standards and Technology (NIST):

> ***Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.***

Cloud computing is based on five attributes [CSP+02]: multitenancy (shared resources),massive scalability, elasticity, pay as you go, and self-provisioning of resources .

Multitenancy (shared resources)

Unlike previous computing models, which assumed dedicated resources (i.e., computing facilities dedicated to a single user or owner), cloud computing is based on a business model in which resources are shared (i.e., multiple users use the same resource) at the network level, host level, and application level.

Massive scalability

Although organizations might have hundreds or thousands of systems, cloud computing provides the ability to scale to tens of thousands of systems, as well as the ability to massively scale bandwidth and storage space.

Elasticity

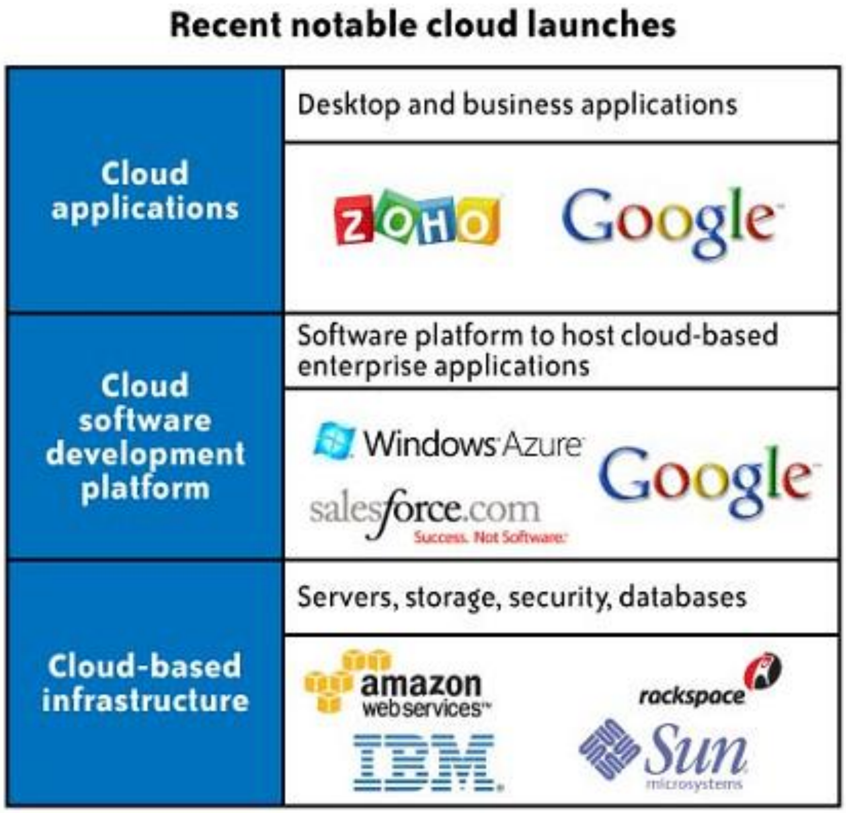
Users can rapidly increase and decrease their computing resources as needed, as well as release resources for other uses when they are no longer required.

Pay as you go

Users pay for only the resources they actually use and for only the time they require them.

Self-provisioning of resources

Users self-provision resources, such as additional systems (processing capability, software,storage) and network resources



Recent notable cloud launches

Cloud applications	Desktop and business applications
	ZOHO Google
Cloud software development platform	Software platform to host cloud-based enterprise applications
	Windows Azure salesforce.com Success. Not Software. Google
Cloud-based infrastructure	Servers, storage, security, databases
	amazon webservices rackspace IBM Sun microsystems

FIGURE - 1

2.1.1 The SPI Framework for Cloud Computing

A commonly agreed upon framework for describing cloud computing services goes by the acronym “SPI” [CSP+02]. This acronym stands for the three major services provided through the cloud:software-as-a-service (SaaS), platform-as-a-service (PaaS), and infrastructure-as-a-service(IaaS). Figure illustrates the relationship between services, uses, and types of clouds.

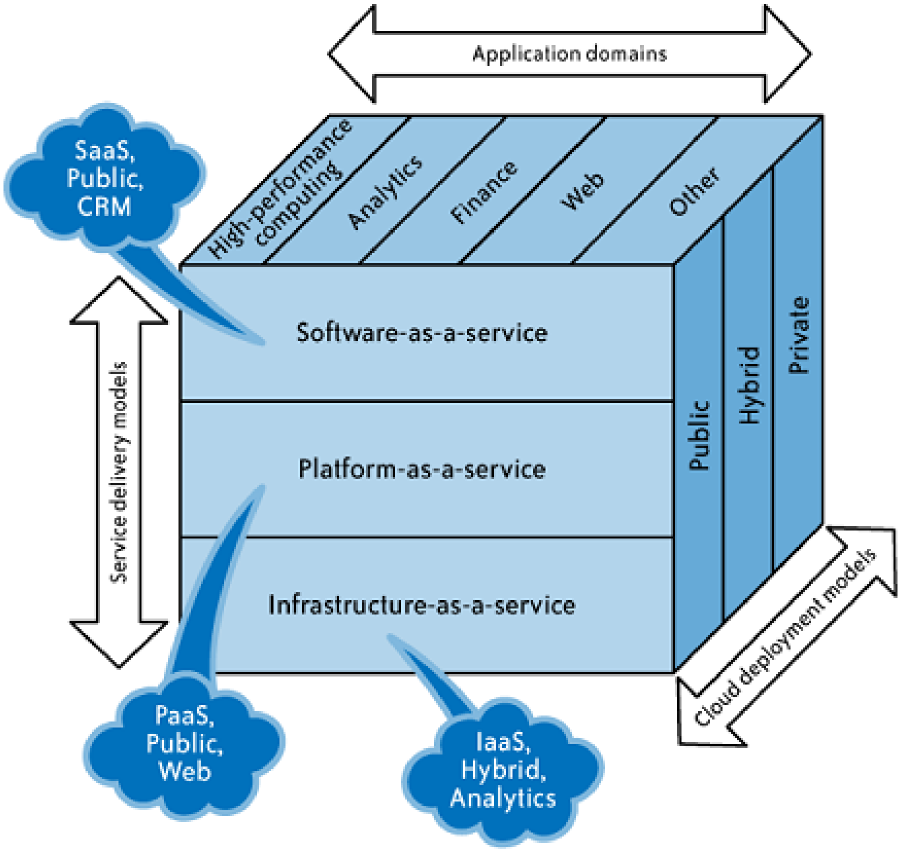
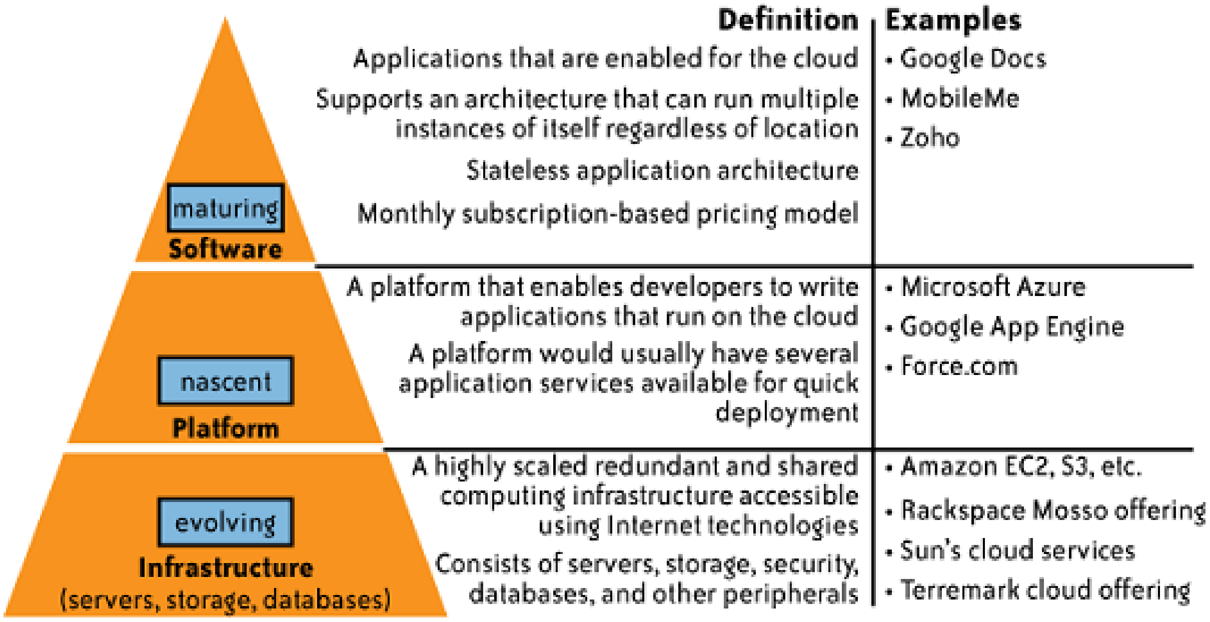


FIGURE - 2

The Cloud Services Delivery Model

As we noted earlier, a cloud services delivery model is commonly referred to as an SPI and falls into three generally accepted services



	Definition	Examples
Software	Applications that are enabled for the cloud Supports an architecture that can run multiple instances of itself regardless of location Stateless application architecture Monthly subscription-based pricing model	• Google Docs • MobileMe • Zoho
Platform	A platform that enables developers to write applications that run on the cloud A platform would usually have several application services available for quick deployment	• Microsoft Azure • Google App Engine • Force.com
Infrastructure	A highly scaled redundant and shared computing infrastructure accessible using Internet technologies Consists of servers, storage, security, databases, and other peripherals	• Amazon EC2, S3, etc. • Rackspace Mosso offering • Sun's cloud services • Terremark cloud offering

While cloud-based software services are maturing, cloud platform and infrastructure offerings are still in their early stages

FIGURE - 3

2.1.2 Cloud Deployment Models

The term cloud is a metaphor for the Internet and is a simplified representation of the complex internetworked devices and connections that form the Internet. Private and public clouds are subsets of the Internet and are defined based on their relationship to the enterprise. Private and public clouds may also be referred to as internal or external clouds ; the differentiation is based on the relationship of the cloud to the enterprise[CSP+02].

The public and private cloud concepts are important because they support cloud computing which enables the provisioning of dynamic, scalable, virtualized resources over Internet connections by a vendor or an enterprise IT organization to customers for a fee. The end users who use the services offered via cloud computing may not have knowledge of, expertise in, or control over the technology infrastructure that supports them.

The majority of cloud computing infrastructure consists of reliable services delivered through data centers and built on servers with different levels of virtualization technologies. The services are accessible anywhere that access to networking infrastructure is available. The cloud appears as a single point of access for all consumer computing needs. Commercial offerings should meet the quality of service requirements of customers and typically offer service-level agreements (SLAs). Open standards are critical to the growth of cloud computing, and open source software has provided the foundation for many cloud computing implementations (e.g., the use of Xen in AWS).

Public Clouds

Public clouds (or external clouds) describe cloud computing in the traditional mainstream sense, whereby resources are dynamically provisioned on a fine-grained, self-service basis over the Internet, via web applications or web services, from an off-site, third-party provider who shares resources and bills on a fine-grained, utility-computing basis.

A public cloud is hosted, operated, and managed by a third-party vendor from one or more data centers. The service is offered to multiple customers (the cloud is offered to multiple tenants) over a common infrastructure; In a public cloud, security management and day-to-day operations are relegated to the third-party vendor, who is responsible for the public cloud service offering. Hence, the customer of the public cloud service offering has a low degree of control and oversight of the physical and logical security aspects of a private cloud.

Private Clouds

Private cloud or internal clouds are terms used to describe offerings that emulate cloud computing on private networks. These (typically virtualization automation) products claim to deliver some benefits of cloud computing without the pitfalls, capitalizing on data security , corporate governance, and reliability concerns.

Organizations must buy, build, and manage them and, as such, do not benefit from lower upfront capital costs and less hands-on management. The organizational customer for a private cloud is responsible for the operation of his private cloud. Private clouds differ from public clouds in that the network, computing, and storage infrastructure associated with private clouds is dedicated to a single organization and is not shared with any other organizations (i.e., the cloud is dedicated to a single organizational tenant)

Hybrid Clouds

A hybrid cloud environment consisting of multiple internal and/or external providers is a possible deployment for organizations. With a hybrid cloud, organizations might run non-core applications in a public cloud, while maintaining core applications and sensitive data in-house in a private cloud.

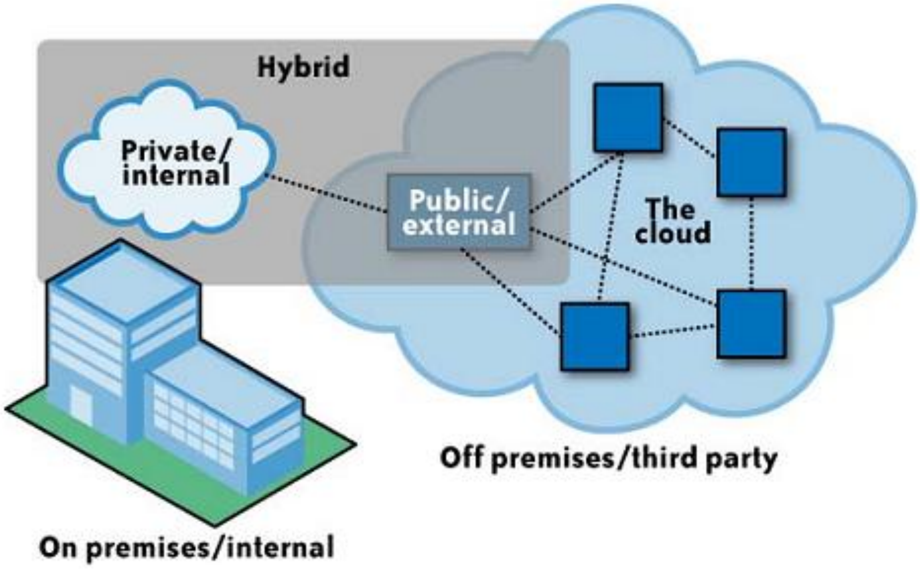


FIGURE - 4

2.1.3 Barriers to Cloud Computing Adoption in the Enterprise[CSP+02]

Security

Because cloud computing represents a new computing model, there is a great deal of uncertainty about how security at all levels (e.g., network, host, application, and data levels) can be achieved. That uncertainty has consistently led information executives to state that security is their number one concern with cloud computing. The subsequent chapters present a detailed examination of those concerns to determine whether they are grounded.

Privacy

The ability of cloud computing to adequately address privacy regulations has been called into question. Organizations today face numerous different requirements attempting to protect the privacy of individuals' information, and it is not clear (i.e., not yet established) whether the cloud computing model provides adequate protection of such information, or whether organizations will be found in violation of regulations because of this new model.

Connectivity and Open Access

The full potential of cloud computing depends on the availability of high-speed access to all. Such connectivity, rather like electricity availability, globally opens the possibility for industry and a new range of consumer products. Connectivity and open access to computing power and information availability through the cloud promotes another era of industrialization and the need for more sophisticated consumer products.

Reliability

Enterprise applications are now so critical that they must be reliable and available to support 24/7 operations. In the event of failure or outages, contingency plans must take effect smoothly, and for disastrous or catastrophic failure, recovery plans must begin with minimum disruption. (See the Cloud Computing Incidents Database at http://wiki.cloudcommunity.org/wiki/CloudComputing:Incidents_Database.) Each aspect of reliability should be carefully considered when engaging with a CSP, negotiated as part of the SLA, and tested in failover drills. Additional costs may be associated with the required levels of reliability; however, the business can do only so much to mitigate risks and the cost of a failure. Establishing a track record of reliability will be a prerequisite for widespread adoption.

Interoperability

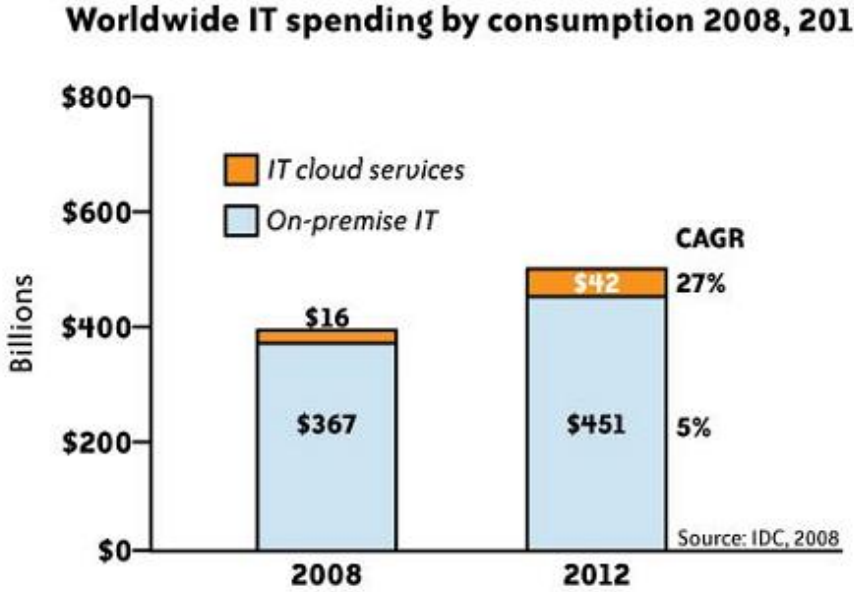
The interoperability and portability of information between private clouds and public clouds are critical enablers for broad adoption of cloud computing by the enterprise. Many companies have made considerable progress toward standardizing their processes, data, and systems through implementation of ERPs. This process has been enabled by scalable infrastructures to create single instances, or highly integrated connections between instances, to manage the consistency of master and transaction data and produce reliable consolidated information.

Even with these improved platforms, the speed at which businesses change may still outpace the ability of IT organizations to respond to these changes. SaaS applications delivered through the cloud provide a low-capital, fast-deployment option. Depending on the application, it is critical to integrate with traditional applications that may be resident in a separate cloud or on traditional technology. The standard for interoperability is either an enabler or a barrier to interoperability, and permits maintenance of the integrity and consistency of a company’s information and processes

Economic Value

The growth of cloud computing is predicated on the return on investment that accrues. It seems intuitive that by sharing resources to smooth out peaks, paying only for what is used, and cutting upfront capital investment in deploying IT solutions, the economic value will be there.There will be a need to carefully balance all costs and benefits associated with cloud computing—in both the short and long terms. Hidden costs could include support, disaster recovery, application modification, and data loss insurance. There will be threshold values whereby consolidating investments or combining cloud services makes sense; for example, it might not be efficient or cost-effective to utilize multiple autonomous SaaS applications.

Each may contract for disaster recovery program services. There is a point where economies of scale mean these functions should be combined in a similar service. Application usage may begin with a low volume of transactions that can be supported with semi-automated master data management. As usage expands and interoperability requirements for the business process become more onerous, a new approach is needed. This evolution may be the most cost-effective approach; however, there is a risk that the business transition costs from one solution to another may change the cost and benefit equation, and hence the solution that should be employed.



Source: IDC, 2008

FIGURE - 5

IT Governance

Economic value is an aspect of IT governance. Effective governance processes that align IT and the business are critical to set the appropriate context for making investment decisions and to balance short-term and long-term needs.

2.2 CLOUD SECURITIES ISSUES

Security issues come under many guises both technical and socio-technical in origin. To cover all the security issues possible within the cloud, and in-depth, would be herculean a task. Existing efforts look to provide a taxonomy over the issues seen. The Cloud Security Alliance is a non-profit organisation that seeks to promote the best practises for providing security assurance within the cloud computing landscape.

Security issues found within the cloud are:

1. *Abuse and Nefarious Use of Cloud Computing*
2. *Insecure Application Programming Interfaces*
3. *Malicious Insiders*
4. *Shared Technology Vulnerabilities*
5. *Data Loss/Leakage*
6. *Account, Service and Traffic Hijacking*
7. *Unknown Risk Profile*

2.3 CLOUD THREAT MODELS

2.3.1 Threat Overview

Before the data life cycle threat models are introduced, some time will be spent detailing the origin and nature of the threats.

2.3.1.1 Origin

The origin of threats to data can be divided into the following categories:

Outsiders :: Outsiders are entities that exist outside of the system and attempt to subvert/circumvent the security infrastructure of the service or masquerade as a legitimate service to ensnare users. Their motivation will stem from simple curiosity or from malice.

Insiders :: More serious threats originate from current or past employees of the CSP. Employees will have intricate knowledge of the actual infrastructure including that of security and as part of their remit may have had direct access to the data itself or through other means.Similar to insiders their motive may be out of curiosity or of malice.

Natural :: Although both insiders and outsiders can induce `errors' within the infrastructure, other errors can occur naturally from the software itself or from hardware failure. For example, when Google pushed a software update to Google Docs [Vas09]. The software malfunction changed the sharing settings of several users documents to include those with whom the affected users had share documents with before.

2.3.2 Goals

Attackers will also have a goal that drives them. This normally implies targeting a particular asset. These are resources that are either tangible or abstract respectively data and data consistency. Hasan, Myagmar et al. [HM+05] provides an incomplete list of what these assets maybe.One can add to this list more cloud specific assets. An incomplete list of possibly targeted assets includes:

- Communication channel
- Storage media
- Data management software
- Data availability
- Data secrecy
- Data integrity
- Data consistency
- Virtual image availability
- Virtual image secrecy
- Virtual image integrity
- Virtual Image consistency
- Service availability

2.3.3 Means

Attackers will usually accomplish their goals by exploiting some technical exploit to gain access to the assets. This can include service hijacking, service impersonation or through poorly secured APIs. If the attacker is a malicious insider they may not need to exploit technical insecurities and will have direct access to the data or will gain access through privilege escalation.

2.3.4 The Lifecycle of Data

The typical lifecycle of data can be describe as the following stages:

Stage 1: *Data Creation/Transmission* -- this is the initial stage, data is created by the user and then pushed to the Cloud for consumption.

Stage 2*: Data Reception* -- data is received in the Cloud before being written to storage and logs taken of activity.

Stage 3: *Output Preparation* -- data is prepared to be returned to the consumer, this involves any transformations that needs to be performed on the data prior to its return i.e.

serialisation.

Stage 4: *Data Retrieval* -- data is received by the consumer from the cloud and has now within the domain of the user.

Stage 5: *Data Backup* -- the CSP will replicate data for archival purposes. This may involve the transferral of a copy of the data to an external store.

Stage 6: *Data Deletion* -- data is permanently deleted from the cloud.

2.3.5 Data Lifecycle Threat Models

While the CIAA Model can be used to model threats to data, it lacks context. The Data Lifecycle Threat Model seeks to classify the threats, first by their placement within the data lifecycle and then using the CIAA model. It is through this classification that one is able to provide some context to the threats and able to produce a more fine grained classification. Hasan, Myagmar et al. [HM+05] introduced a data lifecycle model to describe the threats associated with remote storage. This threat model can be used as a basis upon which a model for cloud resident data can be introduced.

2.3.5.1 Overall Model

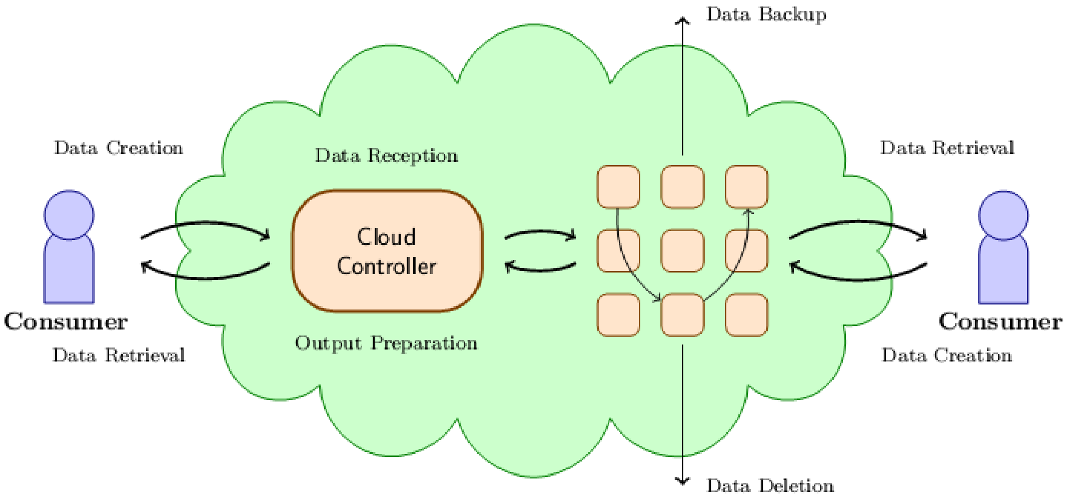


FIGURE - 6

Recall from earlier that in the remote storage context one can assume that the data will be centrally stored within a single store. Within the cloud this cannot be guaranteed. Consumers interacting with both the Cloud Controller and also with the virtualised resources will also be presented with the appearance of a single entity. The data (including the virtual machine) may also be replicated on various virtualised resources and also among different physical machines. By combining this data center model with that of the data lifecycle model for remote storage systems a threat model for the cloud can be produced. Figure summarises this new model.

The difference between this model and the existing one is that the single data store has been replace by virtual resources. This increases the number of places within the model where these stages can exist. The data creation and retrieval will always occur with the consumer. However, the data reception and output preparation stages can now occur either in the Cloud Controller, if interacting with the controller, or with one of the many virtual resources that will exist.Moreover, as a result one must add an additional stage, representing the internal migration of data within the cloud, to the data lifecycle:

Stage 7: *Data Migration* -- data is migrated from a resource inside the cloud to another for availability or scalability purposes.

The threats occurring at this new stage can be viewed as an amalgamation of the first and fourth stages of the data lifecycle. That is the sending agent will transmit data and the receiving agent will receive the data. Furthermore, data migration can also be seen as a type of data backup and the threats to those stages can be included as well.

2.3.5.2 User Centric Model

The overall threat model represents an omniscient narrator's view of the cloud; they see and are privy to all aspects. However, one must also consider the user's point of view. The cloud represents an unknown risk profile to a user. They are not aware of all aspects of the internal workings of the cloud. Thus, one can also present a user centric

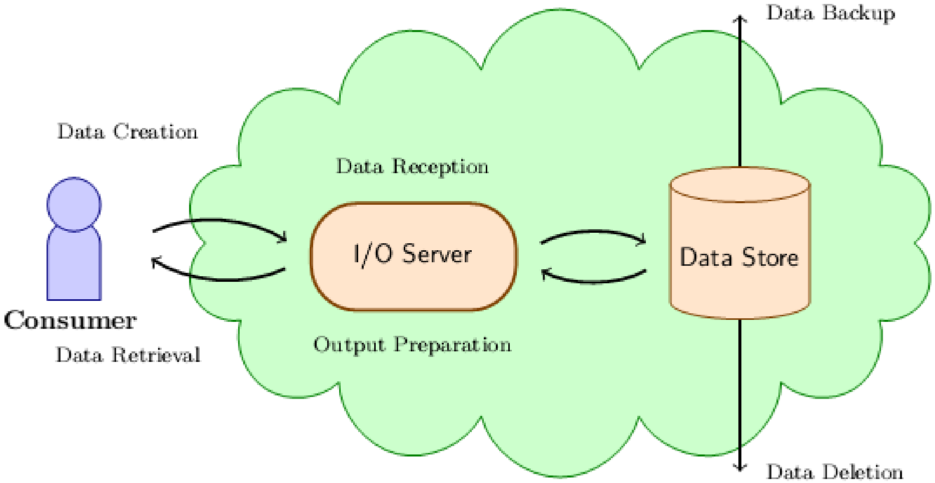


FIGURE - 7

threat model that uses as a basis the clients viewpoint. In this model the client views the cloud as a single entity. The more astute reader will notice that this is reminiscent of the original data lifecycle model. Data goes in the cloud, data comes out the cloud, data gets archived, data is deleted. The user is not aware that the data could migrate between nodes within the cloud. Figure illustrates this user centric model

Chapter: 3 Research Background

3.1 *Data Security Requirements*

Some of the security requirements which will aide when developing or analysing any solution and also when describing the security offered are :-

- Confidentiality
- Remote Access
- Non-Repudiation
- Integrity and Consistency
- Availability and Fault Tolerance
- Major concerns of solution
- Trusts
- Complete and Sound Solution

3.1.1 Confidentiality

The data that is to be entrusted to the cloud may be of a sensitive nature and will thus be subject to several confidentiality measures. Although confidentiality of data is primarily seen as being solved via encryption, as discussed in Broadfoot and Martin [BM03], other aspects need to be considered.

User and Resource Privacy :: Within the Cloud, confidentiality of data also extends to how the data is being processed/used and also the users actions. The means by which CSP can store or process the data is bound by law and these laws must be adhered to. Such data includes and is not limited to: auditing records indicating access attempts and changes (and their results) to the data; properties of the data including size, access policies and origin; and even the existence of the data itself.

Deducible Data :: the hidden information pertaining to an individual can be deduced from existing information i.e. Bob's sexuality. An attacker should not be able to use existing information or information relating to the confidential data i.e. meta-data to deduce any other information. Such attacks should be made as difficult as possible. For those who have multiple personae on the web relating to the different facets of their private life. The ability of an attacker to link the two should be hard.

The requirement of confidentiality is an aspect that both the user and CSP need to be made aware of. Specically, the confidentiality of the plain-text data itself should be the responsibility of the user before it goes into the cloud. Guarantees towards user and resource privacy, and deducible data is best made by the CSP. They are in a better position to provide such guarantees.

3.1.2 Remote Access

The Cloud, by nature, is inherently a `public place'. Services are exposed over HTTP, a public medium. Access to these services need to be controlled and access kept to authorised personnel. Moreover as the data is held remotely, trust needs to be established with the service and with the security provided by the service over the data itself. Access to the data needs to be regulated. CSPs must ensure that entities trying to access the data are not only who they say they are (authentication) but also that they have the right to do so (authorisation) This is made more difficult as CSP will be interacting with multiple users from multiple companies (domains) each of whom will require different management and access policies; and all done remotely.

Authentication :: CSPs must ensure that those trying to access the service are who they say they are. Unauthenticated users and impostors should not be able to access the data. The identity of the entities must be assured. This will imply some form of identity management.

Authorisation :: Once the identity of an entity has been established access to the data held by the CSP needs to be regulated and controlled. Authenticated entities should not be able to access data that they are not authorised to access. For example, two users from different companies should not be able to access each others remote data held by the CSP unless the access has been explicitly allowed.

Location :: Users may be accessing the service/resource from different locations. Authentication of the user should always be performed and should not be linked to the device from which the entity accesses the service.

Revocation :: An important requirement is that of revocation. The revocation of access to individual data and to the service itself must be permissible.

With remote access, guarantees towards location privacy, authentication of identity and authorisation to the data that is resident within the cloud should be made by the CSP. Revocation, and thus assignation, of access to the data should be made by the user themselves.

3.1.3 Non-Repudiation

Both the CSP and user should not be able to deny the origin or refute the integrity of data. Moreover, a verifiable record of the data's lifecycle should exist. The lifecycle of the data and the operations performed on the data should be attestable if a CSP attempts to defraud a user and vice versa. This is especially important if flexible payment models are used. A dishonest CSP could claim that more resources were used by the user and thus be in a position to bill the consumer more than what was actually the case. This implies that records need to be kept concerning usage that can be verified by both users and consumers. Guarantees towards non-repudiation should be made by both the user and the CSP.

3.1.4 Integrity and Consistency

The mobility of data within the cloud only increases the threats that can affect the integrity of data. Data is being transported to-and-from the user and service providers, and also internally within the cloud. The integrity of the data must be guaranteed when it has been placed within the Cloud. Consistency problems can arise from omission and commission failures. Omission failures occur when an entity fails to act upon input.

Commission failures are those that occur when an entity though responds to input the output is not what was expected. It is possible for hardware to fail or connections to be lost at which time the data may be in an inconsistent state or unrecoverable. If data is replicated for some reason e.g. to combat availability, scalability or archival purposes, the consistency of the replicated data must be ensured. The consistency of replicated data can be affected by omission and commission failures. Also, consistency problems can arise during multi-author collaboration when access to the data is performed concurrently.

With regards to integrity and consistency this is a joint responsibility, though it can be divided cleanly between the two entities. The user can make guarantees towards the integrity of the data before it goes into the cloud i.e. pre-cloud insertion integrity, and the CSP needs to ensure the integrity and consistency of the data once it is in the cloud i.e. post-insertion.

3.1.5 Availability and Fault Tolerance

Another problem with entrusting data to a service provider is ensuring the availability of the data once it has been placed within the cloud i.e. resource availability. This is essentially a guarantee that can only be made by the CSP themselves. Users only have the assurances made by their service provider that data will be made available. If the data were to be made unavailable for some reason, users will not be able to access their data and become inconvenienced. The inconvenience caused could also lead to profit loss for both the user and the CSP. Moreover, internally the nodes within the Cloud must also be resilient to node failure and the data held on the nodes must still be available. Internally the Cloud must be fault tolerant.

3.1.6 Major Concerns Of The Solution

3.1.6.1 Empowerment of a user

Within computer security the protection of one's data can be seen as being synonymous with the protection of one's privacy. The `right to privacy' is a fundamental right and is enshrined in many a countries constitution. A user empowered over the privacy of their data can be seen as being empowered over the protection of their data. A thorough grasp of this notion of data privacy is fundamental when building a solution to empower users. As such a better solution can thus be designed and what it means for a user to become empowered can be determined.

However one of the major problems with this interpretation of data privacy, is its definition. Data Privacy is a rather vague and often misunderstood term. For example, take three existing solutions: None of Your Business (NOYB) [GTF08]; Privacy Manager [MP09]; and Content Cloaking (CoClo) [DVZ10]. Each of these three solutions each have a different take on how to protect the privacy of data.

Encryption :: The CoClo solution dictated the encryption of data prior to its insertion into the cloud. Data was hidden completely from unauthorised users and the CSP.

Obfuscation :: With Privacy Manager data was obfuscated. While `obfuscation' does not necessarily imply the encryption of data, obfuscated data can still nonetheless be operated upon by a CSP with the CSP not learning anything about the underlying data. Examples of obfuscation techniques can be found in Kantarcioglu [Kan08].

Contextual Integrity :: The NOYB solution sought to destroy the link between the data and its creator, as well as hide the data itself.

3.1.6.1.2 Contextual Privacy

The NOYB solution uses a privacy model based upon the contextual integrity of data: Contextual Privacy. Originating from Nissenbaum [Nis04], the privacy of data is based upon its context.

Data Privacy holds if the data cannot be linked back to the originator of the data: the data will be viewed out of context. `Big Brother' will be able to collect data and use the data but will not be able to link the data back to the data originator. Ostensibly, this appears to be madness, one's data will still be public.

However, the success of the contextual privacy is dependent on what the `context' of the data is and also what the data itself is. With NOYB the original context of the users data is their Facebook profile [GTF08]. This information is then broken down into atoms representing data items that are common to all i.e. christian name, age and gender, before being randomly distributed among other profiles. While, the person's information will still reside on Facebook it will be viewed out of context of the persons own profile. Facebook, will thus be able to use the information but will no longer be able to link the data back to the original users.

Furthermore, the non-common data items such as email addresses, telephone numbers and addresses are first broken down into atoms representing individual characters. While this is an interesting privacy model, users' data is nonetheless publicly available.

3.6.1.3 A Kafkaesque Approach

Solove [Sol07] argues that a better understanding of privacy can come from not seeing privacy as a single concept but rather as a pluralistic one. The privacy of data will take on various forms depending upon the context under which the data is being examined. This resembles the views expressed in the previous section in which the privacy of data holds if the data stays `out-of-context' i.e. cannot be linked back to the data originator. The underlying metaphor used by Solove [Sol07] stems from Franz Kafka's The Trial ; the approach is Kafkaesque.

The Trial depicts a bureaucracy that uses people's information to make decisions about them yet prevents the peoples ability to participate in how their information is used. Much like the problems associated with users placing their data within the cloud. Here the collection of data is not the main issue, rather it is the unauthorised and unchecked processing i.e. storage, use and analysis, of the collected data.

Taxonomy of Privacy

By conceptualising privacy as what happens to data once it has been collected Solove [Sol07] demonstrates that a taxonomy of privacy violations can be constructed. The taxonomy presented by Solove presents four general categories of privacy problems.

The four general categories are:

a***) Information Collection*** representing the collection of information itself;

b) ***Information Processing*** representing the use of information by the data collector;

c) ***Information Dissemination*** representing the unauthorised release of information;

d) ***Invasion*** representing the use of information to intrude upon an individuals life.

3.1.7 Types of Trust in CSP

Trust is an important notion. To trust an entity implies that one has confidence or faith in that entity. In the previous section the trust imparted to the data collector/CSP by the user varied and was dependent upon the view the user had of the CSP. For instance, the Orwellian approach stipulated no trust in the data collector, while the Kafkaesque approach stipulated that one could trust the CSP somewhat. From these differences a `taxonomy of trust' can be constructed that describes how a user views and comprehends the actions of the CSP. They are:

No Trust :: the Orwellian approach, one does not trust the CSP at all. Used by the CoClo solution [DVZ10].

Some Trust :: the users trust the CSP to some degree. Used by the NOYB and Privacy Manager solutions [GTF08; MP09].

Complete Trust :: the user completely trusts the CSP.

(i) No Trust

With the Orwellian approach, one thought of the CSP as being inherently malicious: In CSP they did not trust. The typical response was to interact with the CSP using obfuscated data. As was discussed previously, this approach has one major drawback. When one starts to not trust the CSP with data and thus send obfuscated data, some functionality of the service will be lost if the

CSP is unprepared, or such measures cannot fit into operation of the service [MP09]. This is especially pertinent if one sends encrypted data. Having no trust in the CSP appears to lead to more problems than it does solutions. Moreover, from a legal point of view the transmission of obfuscated data may result in a breach of any service level agreement that the data creator has signed, or agreed upon, with their CSP.

(ii) Complete Trust

At the other end of the spectrum from having no trust in the CSP is to have complete trust. While this was not specified by any of the solutions mentioned thus far, its use has been seen. Grendel is a complete service-side solution touted by Hedlund [Hed10]. Grendel stores users data in an encrypted format and only decrypts the data when it is being used by an authorized entity. This approach implies that the user has complete faith in the CSP to provide security over their data. The user has effectively acquiesced to the CSP providing all guarantees over the protection of their data. This acquiescence also extends to how the data is used. Users are left with no real empowerment over the control over the protection of their data. Therein lies a problem, all the security is now service-side. Any consumer must have complete trust in the CSP to keep their data secure. Though the cryptographic operations maybe secure the CSP still is responsible entirely for the security of the data. Moreover, key management and storage is also completely service-side, the client must have faith in the CSP concerning key management and storage. This leads one to acknowledge that there is a trade-off between trusting, and thus letting, the CSP protect and use data, and its security.

(iii) Some Trust

The third and final mode of trust implies that the user has some trust in the operation of the CSP but is allowed to have reservations about said operation. Such trust can originate from the service level agreements provided by the CSP in which a certain standard of protection has been promised. The reservations, will originate from the unknown risk profile of the service. With this mode of trust, one needs techniques that allow for the user to trust the CSP to some degree over the protection of their data and remain empowered. Two such examples that have already been mentioned in this chapter were NOYB and Privacy Manager. They allowed the user to use the service and more importantly not lose functionality provided by said service. However, with these examples, particularly that of Privacy Manager, it was noted that for the solution to work the CSP needs to have sufficient knowledge of the techniques used by the user to protect their data so that the CSP can work with the data and respect the users choice.

3.1.8 Complete and Sound Solution

When designing a solution that allows those that place their data in the cloud, one needs to take into account the following salient points.

User empowerment does not alone equal Data Obfuscation.

When ensuring the confidentiality over one's data the aim should be for the data creator to gain control over how the data is being used rather than solely preventing its initial collection. This will include the data creator being in a position to dictate to whom access to their data should be granted.

The user need not be responsible for ensuring all security guarantees.

By design the service user has entrusted their data to some service. It is obvious that some form of co-operation must occur between the service user and the CSP over the protection of this data. A service user cannot be responsible for all aspects governing the confidentiality of their data. A good solution must recognise this and allow the service user to have trust in the CSP's ability to offer data protection.

Some CSPs may be evil but some are more evil than others.

The maliciousness of a CSP will differ from CSP to CSP. The degree of trust afforded to each individual CSP will as a result also differ. Service users who can recognise this difference should have the option of allowing trusted CSPs access to their data, and also be able to revoke such access.

Chapter: 4

Proposed Approach

4.1 Predicate Based Encryption

Predicate Based Encryption (PBE), represents a family of asymmetric encryption schemes that allows for selective fine-grained access control as part of the underlying cryptographic operation [KSW08]. The origins of PBE are in Identity Based Encryption (IBE) [Sha85]. In IBE schemes an entity's encryption key is derived from a simple string that represents the entity's own public identity e.g. an email address. For example, given an entity Albert his corresponding encryption key will be Enc(Albert) == albert@foobar.com. During encryption, the resulting cipher-text will effectively be labelled with the string representing the encryption key, the entity's public identity. An entity's decryption key will be derived from the same string used for the encryption key e.g. Albert's decryption key will be derived from his e-mail address.

On receipt of a cipher-text message the recipient will be able to decrypt the cipher-text if and only if the two identities, contained within the decryption key and cipher-text, are `equal'. PBE schemes offer a richer scheme in which an entity's `identity' can be constructed from a set of attributes and decryption is associated with access policies that offers a more expressive means with which to describe the relation between the attributes.

Generally speaking, within PBE schemes entities and cipher-texts are each associated with a set of attributes. These attributes are used to describe some aspect of the entity, the data that is being encrypted, and the environment. An entity will be able to decrypt a cipher-text only if there is a match between the attributes of the cipher-text and the decrypting entity. Matching is achieved through predicates (access policies) that denote: a) the set(s) of authorised attributes that an entity must possess in order to decrypt and access the plain-text; and b) the relationship between the attributes.

4.1.1 General Predicate Based Encryption

Common to all PBE schemes are four operations allowing for encryption, decryption and key generation. The precise value for encryption and decryption keys is dependent upon both the construction of the scheme and placement of predicates.
A general PBE scheme is defined as follows:

Definition 1 (General Predicate Based Encryption (PBE) scheme). A general PBE scheme consists of the four operations:

Setup :: initialises the crypto-scheme and generates a master secret key MSK, used to generate decryption keys, and a set of public parameters MPK.

(MSK; MPK) := Setup()

KeyGen :: generates a decryption key Dec(entity) based upon the master secret key and some entity supplied input.

Dec(entity) := KeyGen(MSK; input)

Encrypt :: encrypts a plain-text message M using the public parameters and supplied encryption key for an entity.

CT := Encrypt(M; MPK; Enc(entity))

Decrypt :: decrypts a cipher-text if and only if the attributes held by the entity can satisfy the access policy.

M := Decrypt(CT; MPK; Dec(entity))

4.1.2 Types of Predicates

When looking at the different PBE schemes, three groupings of schemes emerge based upon the predicates being used, the scheme's aim and the schemes construction.

Identity Based :: Identity Based Encryption (IBE) schemes are used to encrypt messages using a single attribute that refers to the users identity i.e. email address or passport number Shamir [Sha85]; Boneh and Franklin [BF01]. In these schemes the access policy is also comprised of a single attribute. To add extensibility to IBE schemes these `single' attributes were extended using string concatenation to encode more information.

Attribute Based :: Attribute Based Encryption (ABE) schemes [GS+06] further generalizes upon IBE schemes and uses general predicates styled as boolean formula covering operations such as conjunction, disjunction and thresholds. The attributes themselves need not necessarily refer to an entity's identity, or even to an entity, and can refer to non-identity related aspects such as TCP/IP port numbers and addresses.

Specific Predicate Based :: The final family of schemes are those that use specific predicates during their construction. Although these schemes can be referred to by the predicate being used, the general term Predicate Based Encryption can also be used. Specific predicates that have been used include that of inner product [KSW08] and hidden vector [BW07] predicates.

4.1.3 Access Control

Through the use of attributes and predicates, a more richer form of access control has been built into the cryptography when compared to more traditional asymmetric encryption schemes such as RSA and El Gamal [RSA78; ElG85]. In traditional asymmetric schemes cipher-texts can only be decrypted using a single key that is paired with a single encryption key.

One can summarise the relationship between decryption keys and cipher-texts as being one-to-one: one cipher-text can be decrypted only by one decryption key. However, with PBE schemes this relationship is more flexible. Decryption in a PBE scheme will occur if the predicate can be satisfied by a given set of attributes. This relationship can be described as being one-to-many: one cipher-text can be decrypted by many decryption keys.

By specifying access control in terms of attributes and predicates, the access control offered by PBE is analogous to Attribute Based Access Control (ABAC) and involves the assignment of descriptive attributes to entities, resources and the environment [YT05]. Access to a resource is decided upon by access policies that described the sets of attributes required, often as a boolean formula, for access to occur. ABAC is considered to be more flexible and scalable when compared to other existing access control techniques such as Role Based Access Control (RBAC) [SC+96] and Lattice Based Access Control (LBAC) [SS94].

Moreover, ABAC is able to combine the functionality of both RBAC and LBAC into a single access control model. The authorisation architecture for ABAC consists of the following four actors:

An Attribute Authority (AA) is responsible for the creation and management surrounding attributes used to describe the resources, entities and the environment respectively.

The Policy Authority (PA) is responsible for the creation and management of the access policies that govern access to resources.

The Policy Decision Point (PDP), is where the access policies governing access to a resource are evaluated against the attributes attributed to the requesting entity, resource and the environment.

A Policy Enforcement Point (PEP) is a fixed point where the right of the entity requesting access to a resource is challenged. This differs from the PDP in that here the challenge is issued, and at the PDP the challenge is performed.

PBE does not replicate the functionality seen in ABAC completely. Unlike ABAC the environment, entities and resources are not assigned to attributes directly, they are attached to crypto-graphic keys. The use of which, will have an affect upon the access control. Furthermore, the distribution of these four actors to entities will differ based upon the placement of access policies i.e. the type of PBE scheme. Moreover, there is no central point at which policy enforcement and decision will occur.

4.2 Working Of Predicate Based Encryption Scheme

PBE uses WATERS scheme for its working. The Waters scheme consists of four fundamental algorithms:

Setup : takes as input a security parameter and outputs a master public key MPK (public parameters) and a master secret key MSK :

(MSK , MPK) := Setup (1 ^ n)

Encrypt : takes as input a master public key, an access structure A and a message M in some associated message space. It returns a cipher-text CT :

CT := Encrypt (MPK , A , M)

KeyGen : takes as input the master secret key and a set of attributes S that describe the key. It returns the decryption key Dec(S) :

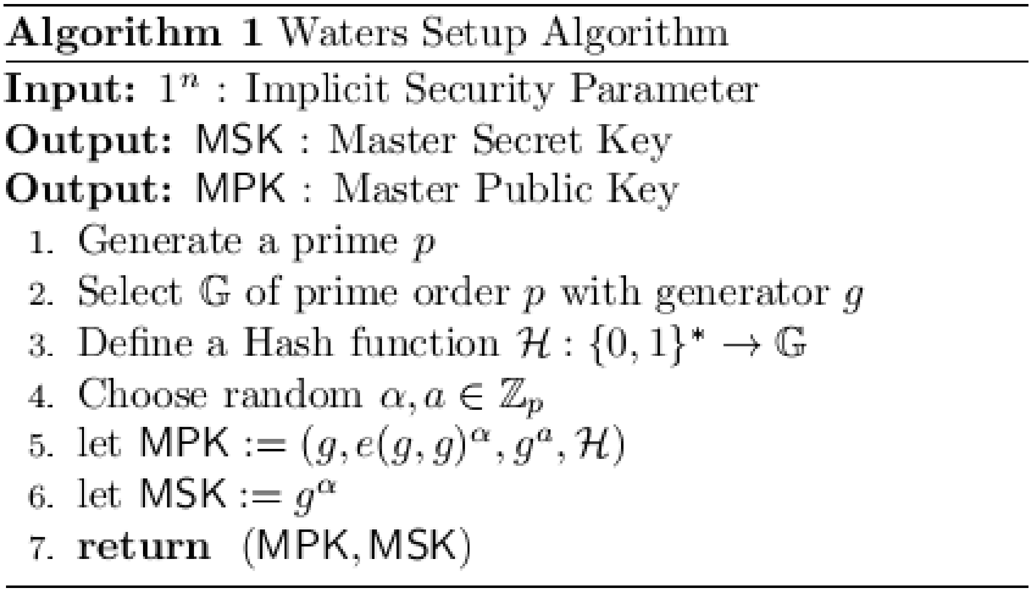
Dec(S) := KeyGen (MSK , S)

Decrypt : takes as input the public parameters MPK, a decryption key Dec(S) and a cipher-text C. It outputs either a message M or the distinguished symbol $ if the set of attributes S do not satisfy the access structure A.

Decrypt (MPK , Dec (S) , C) = (M If correct private key OR $ If incorrect private key)

4.2.1 Setup

The Setup algorithm takes as input a security parameter and outputs a master public key MPK (public parameters) and a master secret key MSK.

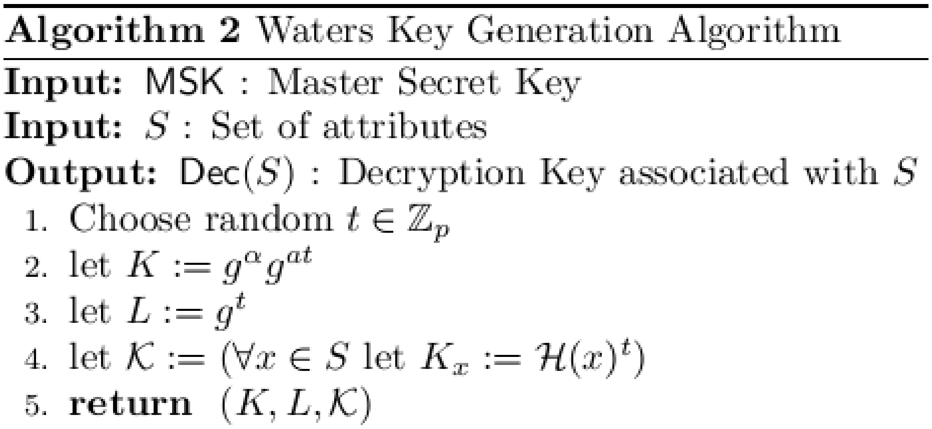


Algorithm 1 Waters Setup Algorithm

```
Input: $1^n$ : Implicit Security Parameter
Output: MSK : Master Secret Key
Output: MPK : Master Public Key
  Generate a prime $p$
  Select $\mathbb{G}$ of prime order $p$ with generator $g$
  Define a Hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{G}$
  Choose random $\alpha, a \in \mathbb{Z}_p$
  let $\mathsf{MPK} := (g, e(g,g)^\alpha, g^a, \mathcal{H})$
  let $\mathsf{MSK} := g^\alpha$
  return $(\mathsf{MPK}, \mathsf{MSK})$
```

4.2.2 KeyGen

The KeyGen takes as input the master secret key and a set of attributes S that describe the key. It returns the decryption key Dec(S).

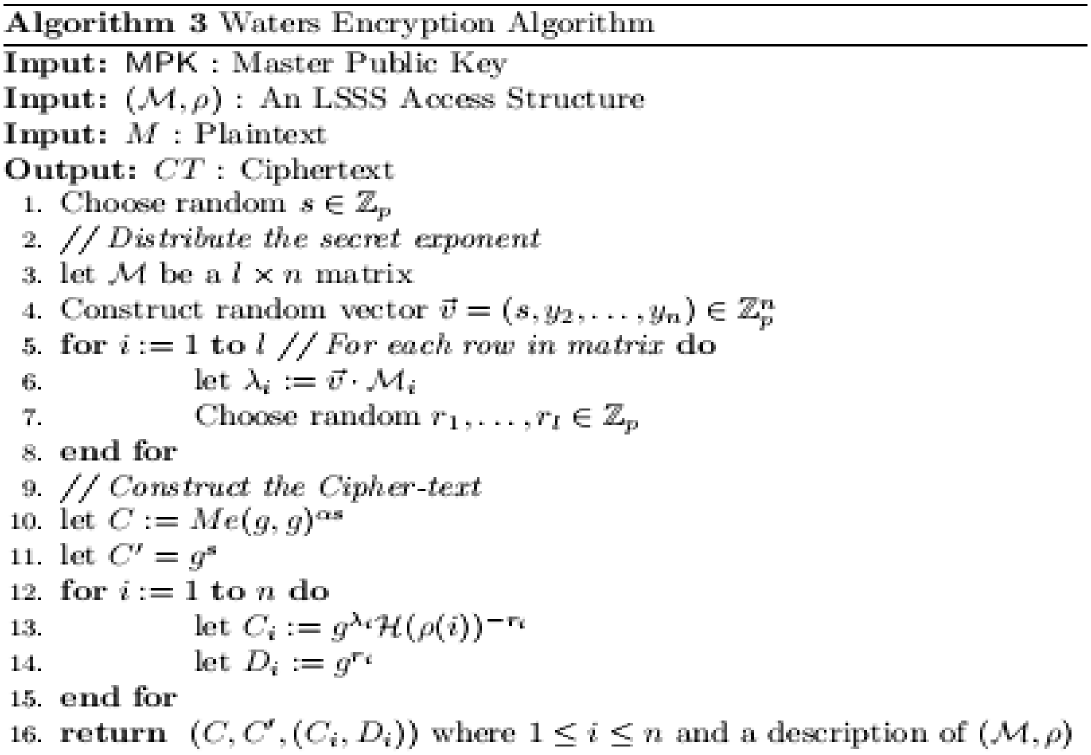


Algorithm 2 Waters Key Generation Algorithm

```
Input: MSK : Master Secret Key
Input: $S$ : Set of attributes
Output: $\mathsf{Dec}(S)$ : Decryption Key associated with $S$
  Choose random $t \in \mathbb{Z}_p$
  let $K := g^\alpha g^{at}$
  let $L := g^t$
  let $\mathcal{K} := (\forall x \in S \text{ let } K_x := \mathcal{H}(x)^t)$
  return $(K, L, \mathcal{K})$
```

4.2.3 Encrypt

The encryption algorithm works by first using the access structure to distributed the secret s among the attributes and constructs a series of randomised variables to hold the result of this distribution. It then constructs the cipher-text and publishes that along with a description of the access structure.

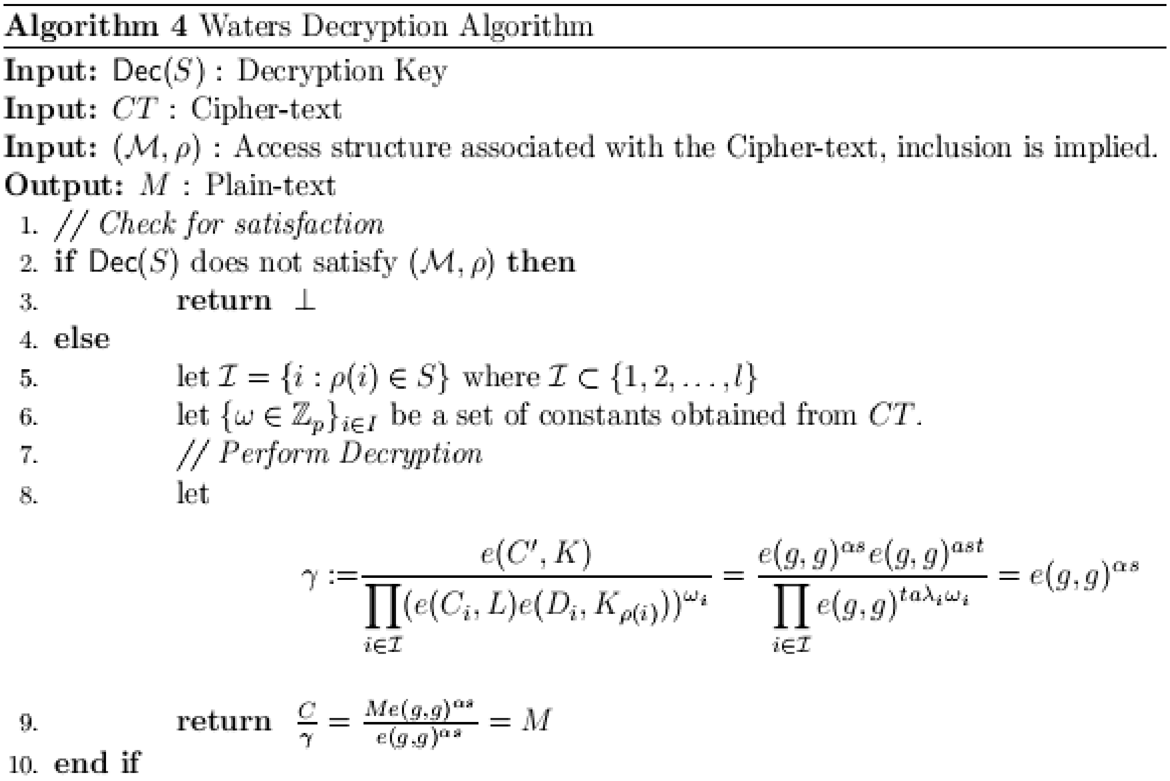


Algorithm 3 Waters Encryption Algorithm

```
Input: MPK : Master Public Key
Input: $(\mathcal{M}, \rho)$ : An LSSS Access Structure
Input: $M$ : Plaintext
Output: $CT$ : Ciphertext
Choose random $s \in \mathbb{Z}_p$
// Distribute the secret exponent
let $\mathcal{M}$ be a $l \times n$ matrix
Construct random vector $\vec{v} = (s, y_2, \ldots, y_n) \in \mathbb{Z}_p^n$
for $i := 1$ to $l$ // For each row in matrix do
        let $\lambda_i := \vec{v} \cdot \mathcal{M}_i$
        Choose random $r_1, \ldots, r_l \in \mathbb{Z}_p$
end for
// Construct the Cipher-text
let $C := Me(g, g)^{\alpha s}$
let $C' = g^s$
for $i := 1$ to $n$ do
        let $C_i := g^{\lambda_i} \mathcal{H}(\rho(i))^{-r_i}$
        let $D_i := g^{r_i}$
end for
return $(C, C', (C_i, D_i))$ where $1 \leq i \leq n$ and a description of $(\mathcal{M}, \rho)$
```

4.2.4 Decrypt

Decryption of a cipher-text is relatively straight forward. First the algorithm checks to ascertain if the private key satisfies the access structure. If satisfaction does not occur then the algorithm returns the distinguished symbol $. Otherwise, the algorithm will identify the attributes that satisfied the access structure.



```
Algorithm 4 Waters Decryption Algorithm
Input: $\mathsf{Dec}(S)$ : Decryption Key
Input: $CT$ : Cipher-text
Input: $(\mathcal{M}, \rho)$ : Access structure associated with the Cipher-text, inclusion is implied.
Output: $M$ : Plain-text
// Check for satisfaction
if $\mathsf{Dec}(S)$ does not satisfy $(\mathcal{M}, \rho)$ then
        return $\perp$
else
        let $\mathcal{I} = \{i : \rho(i) \in S\}$ where $\mathcal{I} \subset \{1, 2, \ldots, l\}$
        let $\{\omega \in \mathbb{Z}_p\}_{i \in I}$ be a set of constants obtained from $CT$.
        // Perform Decryption
        let
            $$\gamma := \frac{e(C', K)}{\prod_{i \in \mathcal{I}} (e(C_i, L) e(D_i, K_{\rho(i)}))^{\omega_i}} = \frac{e(g,g)^{\alpha s} e(g,g)^{ast}}{\prod_{i \in \mathcal{I}} e(g,g)^{ta\lambda_i\omega_i}} = e(g,g)^{\alpha s}$$
        return $\frac{C}{\gamma} = \frac{Me(g,g)^{\alpha s}}{e(g,g)^{\alpha s}} = M$
end if
```

4.3 Modes of Operation of PBE used in cloud

Several interesting modes of operation/use can be constructed for PBE crypto-systems. These modes are:

Public Key Infrastructure (PKI) :: where a third-party controls over the Key Authority;

Multicast Content Provision (MCP) :: where an encrypting entity controls the Key Au-thority;

Duty Delegation Infrastructure (DDI) :: where a decrypting entity control the Key Authority.

4.3.1 Third-Party Owner: Public Key Infrastructure Mode

Throughout this chapter the KA has been presumed to be an independent third-party, that is the interest of the KA is solely that of key provision and management. This setup rep-resents the `standard use case' for PBE. The KA facilitates the cryptography as used by the encrypting and decrypting entities: the cryptography is oered as a service. In essence, the third-party owner is providing a PKI in the traditional sense.

Other entities use the crypto-system to encrypt data under the proviso that entitlement to decryption keys has been validated by the third-party. This `generic' mode of operation can thus be referred to, simply, as the Public Key Infrastructure mode of operation. Taking into account the number of encrypting and decrypting entities it is clear that combination of many decrypting and many encrypting entities provides the only mode of operation.

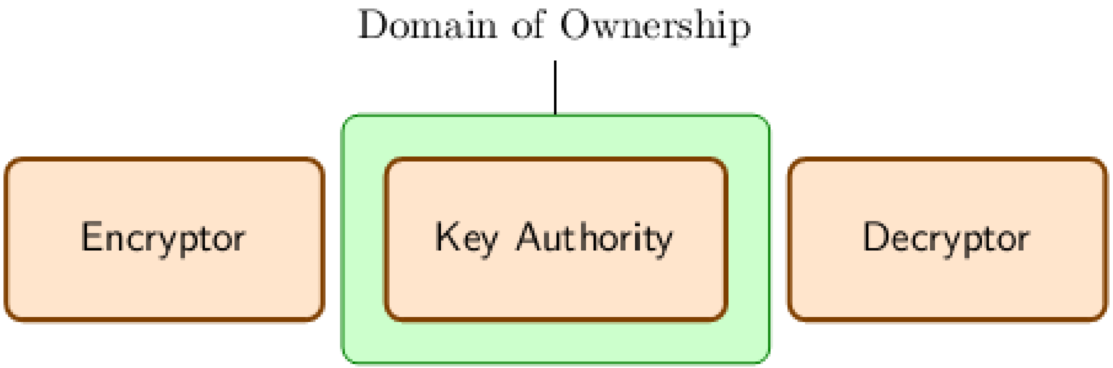


FIGURE - 8

The other combinations are just specific instances that can easily be represented within the context of many encrypting and many decrypting entities.With a third-party being the KA it implies that the third-party will not have a vested interest in the data that is to be encrypted. Rather, the `vested interest' is in the entity's themselves. Given Ciphertext-Policy scheme's disposition towards the provision of user-centric access control, Ciphertext-Policy schemes are more preferable to use as the underlying PBE scheme. Moreover, if a Key-Policy scheme were to be used then this would imply that not only would the third-party owner have some say in access policy creation but also that access control is defined per decrypting entity

4.3.2 Encrypting Entity as Owner: Multicast Content Provision Mode

With an encrypting entity in control over the KA, the owner is encrypting their own data and is assigning the ability to decrypt the data to other entities. Here, the encrypting entity should be viewed as the distributor of their own content. PBE is used to mitigate access to the content to select entities i.e. multicast access, through the assignment of decryption keys----cf.Targeted Broadcast Goyal, Saha. [GS+06].This mode of operation can be referred to as: MCP.

This mode of operation encompasses several different ways in which ones content can be provided: the content stream and the content pool. The stream prescribes the active distribution of content. The pool is when the content has been collected together i.e. a database, and stored together. With these viewpoints, and mode of operation, to only have a single decrypting entity leaves no obvious benefits or usefulness.

This leaves only situations in which the number of encrypting and decrypting entities involved is: one, many, and many, many, respectively. The only difference being is that with the latter, the owner has given the ability to encrypt, and thus add content for distribution, to other entities as well as itself. With the provision of a stream/pool of content this mode of operation is inherently data-oriented. Ciphertext-Policy schemes are suited in this situation if the access afforded to a decrypting entity is to be permissive, or based upon who the entity is.

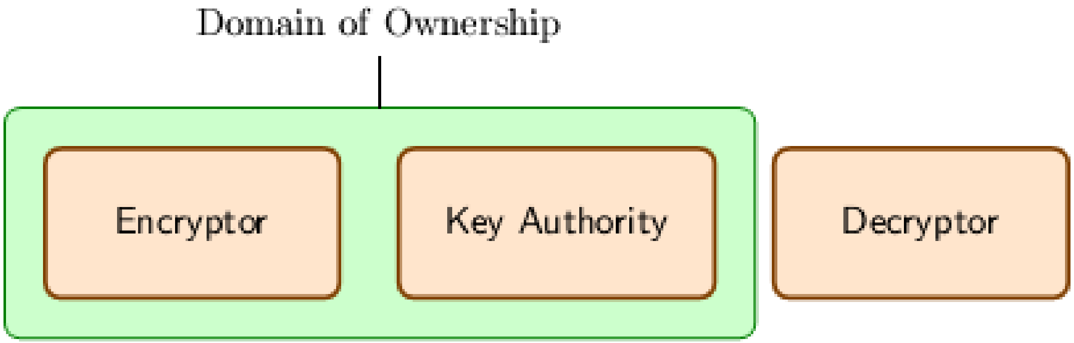


FIGURE - 9

However, the use of Key-Policy schemes is more suitable, not only do they provided proscriptive access control but that the decryption key assigned to a decrypting entity imitates, albeit crudely, a search query. Here the use of KP-PBE is analogous to key word based encrypted search [BDC+04]. Regardless of the underlying scheme this mode of operation affords the owner complete control over key management and to whom decryption of their content is permissible.

4.3.3 Decrypting Entity as Owner: Duty Delegation Infrastructure Mode

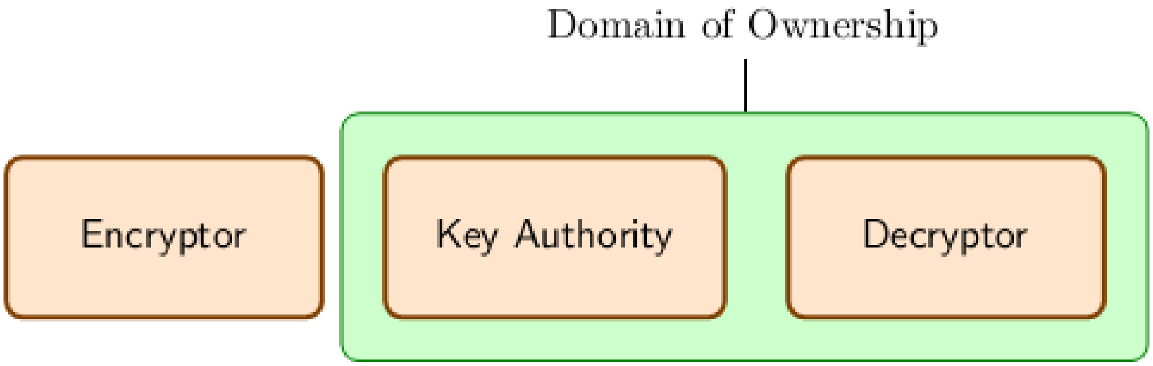


FIGURE - 10

A decrypting entity in control over the KA implies that PBE will be used to mitigate access to data that has been encrypted for the owner by other entities. PBE allows for the owner to delegate access the cipher-text to other, select, entities. The reason for this mode of operation is to facilitate the delegation of decryption duties to other entities. As such this mode of operation can be referred to as a DDI.

This mode of operation is similar to the MCP mode in that the aim is to restrict access to content to other entities, however, the flow of information is inward rather than outward. Within this setting a single encrypting entity or a single decrypting entity provides no obvious benefits, only many encrypting and many decrypting entities . As with the MCP mode of operation, a more proscriptive form of access control is prefer-able i.e. Key-Policy schemes.

The emphasis is on restricting explicitly the cipher-texts that a decrypting entity (aside from the owner) can decrypt. The owner already allows the decrypting entity the ability to read the owners' messages but the trick is to limit which messages the entity can read. A permissive form of access control i.e. Ciphertext-Policy, does not allow for this. Moreover, with Ciphertext-Policy schemes the access policy is decided, ultimately, by the encrypting entity rather than the Key Authority.

Chapter: 5
Experimental Results

5.1 Comparison with Existing Encryption Systems

The use of PBE schemes comes selective fine-grained access control over access to encrypted data. If this functionality was to be compared to existing asymmetric and symmetric crypto-systems in terms of communication styles then :

5.1.1 Communication Styles

Type of Communication Styles	Access Control	Target Access
Symmetric Systems	Broadcast	Everyone
Asymmetric systems	Unicast	Particular entity
PBE systems	Multicast	Particular set of entities

TABLE - 1

5.1.2 Comparing Number of Cryptographic Operations and Cost

Type of Communication Styles	Number of operations	Cost Involved
Symmetric Systems	Twice (No of Recipients)	Everyone
Asymmetric systems	No of Recipients	Particular entity
PBE systems	No. of entities	Particular set of entities

TABLE - 2

5.1.3 Key Management

Type of Communication Styles	Number of keys Involved
Symmetric Systems	Large
Asymmetric systems	Medium
PBE systems	Little

TABLE - 3

5.1.4 Cipher-text Management Differences

With the asymmetric schemes multiple copies of the plain-text, encrypted with different encryption keys, exist: one copy per receiving entity. This increases the number of cipher-texts that contain the same message that can then be used in known-plaintext attacks by a recipient against other recipients, or in a cipher-text-only attack by others. Similarly with the combined approach although there was a single copy of the encrypted data, multiple copies of the group key did exist. Thus, allowing attackers the ability to conduct ciphertext-only attacks on the group key. With PBE crypto-systems only one cipher-text is created and reduces the risk of such attacks.

5.2 Security Issues

This section presents several security issues that arise from the use of PBE schemes as part of a crypto-system.

5.2.1 Attribute Revocation

An interesting dilemma with PBE schemes is the issue of key revocation. With these schemes the instant revocation of cryptographic keys is difficult. As keys will be issued using attributes that have an expiration date, there is a window between when the KA decides to revoke an decryption key (or encryption key) and when the attributes naturally expires. As a result, newly unauthorised entities can still decrypt messages. There is a trade-off between how long the KA wishes to allow an attribute to be active i.e. its validity period, and the risk involved of use by an unauthorised entity.

5.2.2 System Integrity

The use of PBE within a crypto-system relies upon several components to work together. Compromising any one of this components can affect the operational integrity of the PBE scheme.Moreover, multiple decrypting entities may collude in an attempt to decrypt messages that own their cannot do so.

5.2.3 A Reputable Key Authority

So far the maliciousness of the KA has been assumed to be false, the KA partakes not in any fraudulent nor malicious activity. With the different modes of operation the problem of a malicious KA is only prevalent when a third-party is owner: the PKI mode of operation. The remaining two modes alleviate the problem of a malicious KA to a fair extent.The owner/operator of the KA has s vested interest in its correct operation. For example with the MCP mode of operation the aim is for the crypto-system owner to restrict access to the content that it encrypts itself. To be fraudulent with respect to this access seems to be an unfounded position for the owner to take.With a third-party owner, entities are reliant on this owner to perform their duties dutifully and more importantly correctly.

Suppose that the owner is malicious. The owner will be able to purposefully assign fraudulent decryption keys to other entities or themselves. When using non-attribute hiding schemes ,a malicious KA owner can assign themselves a decryption key that will be able to decrypt a cipher-text upon inspection of the cipher-text and its meta-data. With non-attribute hiding schemes extra guarantees/measures are needed from the crypto-system owner to not commit fraud. These guarantees/measures may be technical or legal based and are outwith the scope of this discussion.

5.3 Various Scenarios of using PBE in Cloud

Scenario I : ***Embedded within a Service***.

A CSP incorporates a CP-PBE scheme in PKI mode within an existing service. Providing service users the means with which they can share data amongst each other within the domain of a particular service.

Scenario II: ***PBE-as-a-Service.***

Similar to Scenario I with the exception that PBE is used out with the confines of a particular service domain.

Scenario III: ***`Database' Submission*.**

A CSP offers a service in which service users can submit data to a database. A KP-PBE scheme in DDI mode is used to restrict access over the submitted content to authorised employees of the CSP and other affiliated parties.

Scenario IV: ***`Database' Access*.**

The antithesis of Scenario III. A CSP offers content that can only be accessed by subscribed users. A KP-PBE scheme is used in MCP mode to restrict subscribed users' access to the CSP's content.

Scenario V: ***`Distributed Security'*.**

The service user deploys their own CP-PBE scheme in MCP mode. PBE is used to mitigate access over data the user pushes to the Cloud.

5.3.1 Scenario I: Embedded within a Service

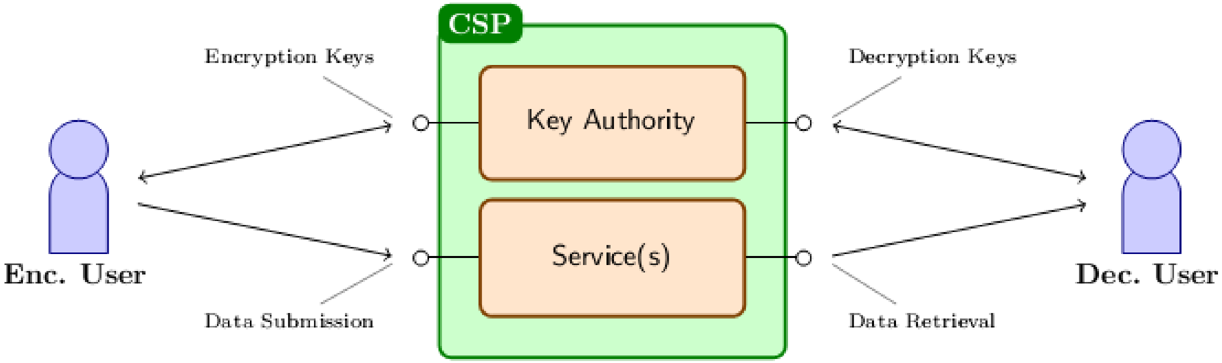


FIGURE - 11

Embedding a CP-PBE scheme within an existing service allows for the complete integration of PBE within the service itself. Figure outlines the core interactions within this scenario. Upon registering with the service, new users are provided with a decryption key derived from a set of attributes that describe the user. When pushing data to the service, service users encrypt their data under access policies of their choosing.

Moreover, as the KA is embedded within the service, the CSP can also aid in policy administration. In other words, the CSP can offer service users a means through which to specify policy rules and thus construct access policies.Due to embedding PBE within the service itself, the attribute universe can be extended to refer to service specific functionality such as the CSP's ability to read the users data for targeted ads.

For example, take an Online Social Network (OSN). The attribute universe can identify service users through attributes such as: identification number, gender, school networks, interests, location, D.O.B. et cetera. Functionality such as, the visibility of users' messages can also be included within the same attribute universe. Following on from indicating the visibility of a user's message is the notion that the CSP can also be referenced within the policy rules and treated as a user in their own right. Though this does not adhere strictly to the PKI's mode it nonetheless allows the encrypting user to explicitly opt-in the CSP when the user constructs policy rules.

Moreover, such an attribute can be combined with a date attribute to provide the CSP with a limited window of opportunity to access encrypted data. This is a powerful construct, the user has explicitly stated within an access policy that the CSP can access the encrypted data, and more importantly for how long.A problem for this scenario is that service users need to trust the CSP. As the CSP is the KA, the CSP could easily construct their own arbitrary decryption keys, and use these keys to decrypt service users' messages. Service users still need to trust the CSP to not be malicious in this respect. This trust can be enforced through legal means i.e. privacy policies and service level agreements. Furthermore, using PBE as described requires that each CSP provide their own scheme. With each service that the user interacts with a different PBE scheme needs to be taken into account. That is, separate key management facilities and algorithms need to be managed by the user; one per service.

5.3.2 Scenario II: PBE-as-a-Service

Providing a CP-PBE scheme as a service in itself allows service users to interact with multiple services, offered by different CSPs, and use a common solution. This reduces the overhead concerning key management, and also the different encryption and decryption algorithms that a service user must be aware of. Providing `PBE-as-a-Service' makes this scenario an example of Security-as-a-Service more specifically it can be tied directly into Identity-as-a-Service services.This implies that not only can this scenario be used to protect SaaS level services but also at the Platform as a Service (PaaS) level as a service. Figure outlines the core interactions within this scenario.

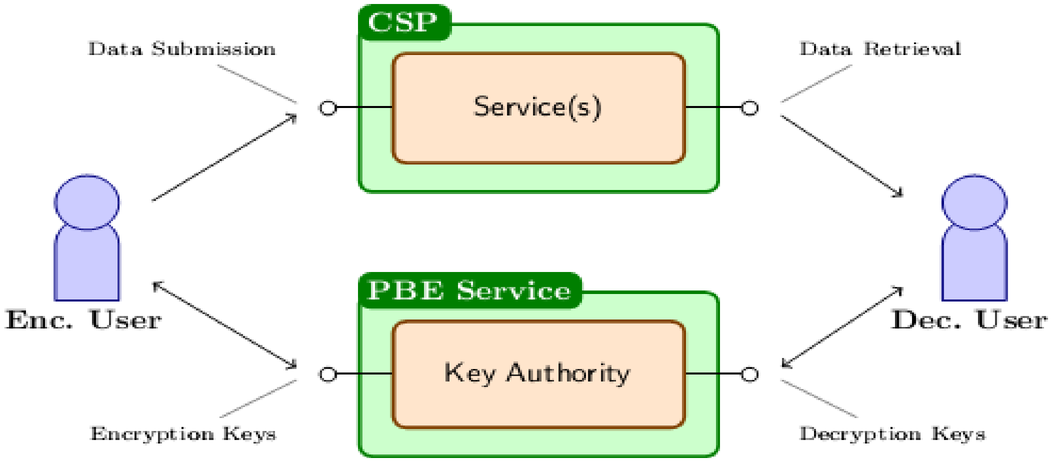


FIGURE – 12

Like Scenario I, users possess decryption keys comprised of a set of attributes used to describe the user themselves. When interacting with different services the user can push encrypted data knowing that only others who can satisfy the access policy can access the underlying plain-text data. Policy administration service can also be offerd by the PBE service. As with Scenario I CSPs should be treated as users in their own right. Hence, when constructing policy rules the encrypting user can explicitly opt-in the CSPs as someone with whom they wish to share their data with.

Moreover, CSPs `service users' can themselves integrate this pre-existing PBE service into their own service offerings. Unlike Scenario I, however, service users can treat the CSP like any other user and not have to worry about the CSP's maliciousness. A CSP in this setting is unable to construct arbitrary decryption keys. Users can use a service even though they may have no trust in the CSP offering that service. When the user does have trust in the CSP to access the user's data, then the CSP can be opted in.

However, the service agnosticism does present several interesting issues. For one, the attribute universe is defined by a third-party who may not possess knowledge of service specific information, offering users attributes that describe information outwith the context of a service. This may affect the expressiveness of policy rules as defined by encrypting users.

Offering a PBE service does reduce the overhead in terms of key management and knowledge of algorithms, how-ever, this reduction in overhead comes at some cost. Scenario I is highly compartmentalised,CSPs are only able to access the data that has been pushed to their service. With the solution described in this section the PBE service provider, if malicious, has the ability to construct decryption keys to access data that a user has pushed to multiple services.

5.3.3 Scenario III: `Database Submission'

The rationale behind the DDI mode is controlling access over submitted data. When a service user uses this mode of operation it implies that the service user themselves will be controlling access to data that has been submitted to them: This has no obvious benefits or uses. It is better for the user to utilise the MCP mode of operation. On the other hand with a CSP as the KA, an interesting use case does emerge.

With DDI mode the CSP is using PBE to facilitate selective access over data, submitted expressly for use by the CSP, under the provison that said data will only be accessible by the CSP's own employees and affiliates. That is data is submitted by service users and the CSP allows authorised entities access to select subsets of all data that has been submitted. This essentially describes the operation of a database. Databases can be thought of in terms of : a) who is submitting the data; b) who is accessing the data; and c) the data itself. With DDI mode, the database is stored with the CSP. Data is submitted for the CSP by service users, and is being accessed by the CSP's employees and other affiliates. Figure outlines the core interactions for this scenario.

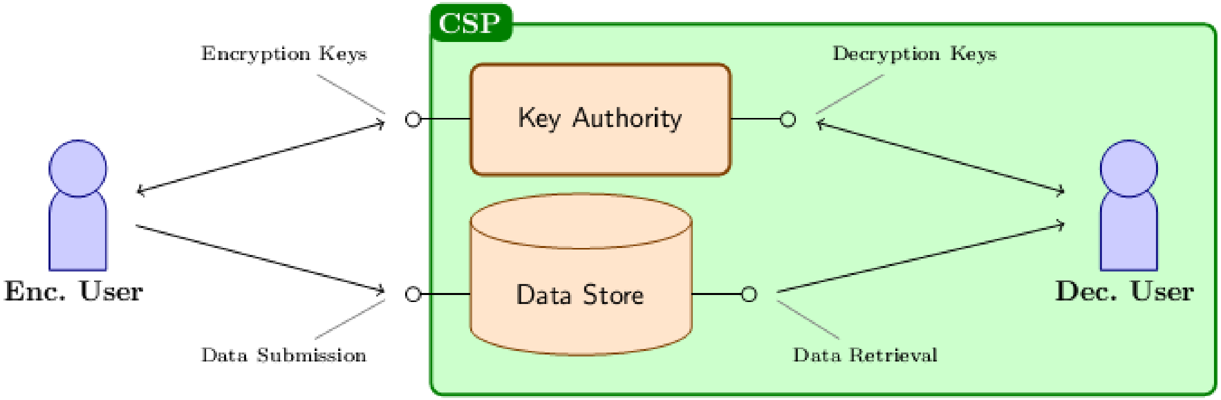


FIGURE - 13

Within databases one needs to control access not only to the database itself but also to the data stored within. Obviously, users' access over the data within the database needs to be proscriptive and users ability to access individual records needs to be curtailed. Key-Policy schemes provide such access control. As such data is encrypted under a set of attributes, and decryption keys from a predicate. Another and more intuitive reason for using Key-Policy schemes stems from the decryption keys. Decryption keys in this setting represent simple search queries. Use of a KP scheme is analogous to keyword based search albeit with a more expressive query mechanism. When submitting data, service users encrypt their data under a set of attributes. The CSP can then internally determine on a per employee basis what the employee can access from this database.

Moreover, with numerical attributes it is feasible to limit the period during which the employee can use their query. The analogy of a `database' implies the deployment of PBE as part of either a SaaS, or PaaS service. For example, a course submission service could use DDI mode to control submission of students submission. During submission, students could encrypt their course under a set of attributes that describe : a) the student's matriculation number; b) the course code; c) the assignment number; and d) submission time. Lecturers and teaching assistants can then be assigned decryption keys pertaining to the course(s) they are associated with.

5.3.4 Scenario IV: `Database Access'

CSPs can utilise MCP mode to restrict access to the content that they produce in much the same manner as was seen with Scenario III. The difference lies in the involved actors and origin of the data. Figure outlines the actors involved and core interactions between them.

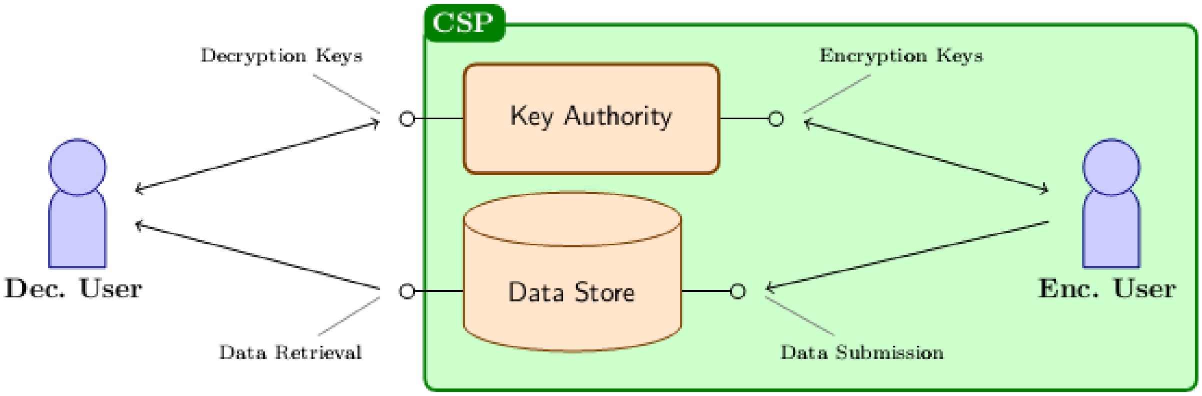


FIGURE - 14

Employees of the CSP together with authorised affiliates are those permitted to encrypt data. Decryption of data is performed by service users. When signing up to a service, users are assigned a decryption key derived from some policy rule that describes the data they are allowed to access. Such policy rules can be used to indicate service related information such as subscription level and periods of time. This provides the CSP with a means to enforce different subscription models i.e. freemium. With MCP mode the act of restricting access can also extend to any content streams that the CSP produces and broadcasts. This implies that this scenario can be used at both the PaaS and SaaS service levels.

5.3.5 Scenario V: `Distributed Security'

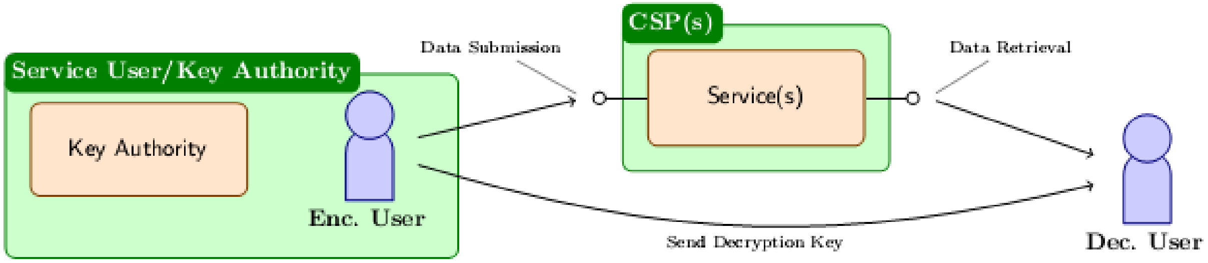


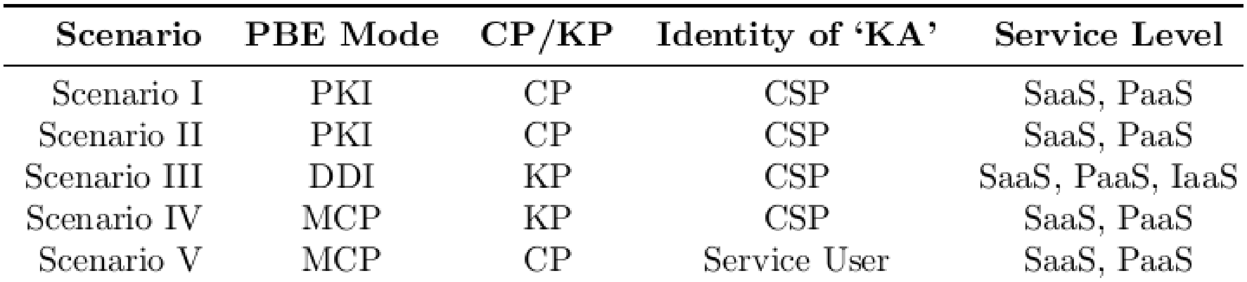
FIGURE - 15

Scenarios I and II present solutions to protecting data they wish to share via the cloud. The underlying problem with these solutions is the reliance upon a third-party to act as the KA. When deployed by a service user the MCP mode presents an improvement with respect to trusting a third-party. With the MCP mode there is no need for a trusted third-party over key management, users provide their own PBE scheme. When signing up to a service or when starting to interact with a new `friend', the encrypting user i.e. the Key Authority, will create and assign to the entity a decryption key. This decryption key will be derived from a set of attributes that describes how the decrypting user relates to the encrypting user.

Through use of a Ciphertext-Policy scheme, the encrypting user can encrypt their data locally under a policy rule, and then push the cipher-text into the cloud. Attributes can be used to indicate, for example: a) type of relationship with another entity i.e. an associate, a friend, a close friend, a really close friend, family, or CSP; b) visibility of message i.e. is the message to be private or public; and c) an identifier for each domain that the user has interacted with i.e. on a OSN.

This use of the MCP mode of operation presents a decentralised solution to the problems associated with Scenarios I and II. Implying this scenarios use at the PaaS and SaaS service levels, with each `user' providing their own encryption scheme. From this two interesting observations can be made that can affect the management and deployment of this mode.

First,the encrypting user is responsible for the remit of the Key Authority. The user is responsible for policy rule administration, responsible for the the correct assignment of decryption keys, responsible for key management et cetera. Secondly, the security guarantees provided cover unidirectional communication. Each user utilises PBE to protect their own data only. For n users communicating it requires each user to take into account the decryption keys, and associated decryption algorithms for n - 1 other PBE schemes.



Scenario	PBE Mode	CP/KP	Identity of 'KA'	Service Level
Scenario I	PKI	CP	CSP	SaaS, PaaS
Scenario II	PKI	CP	CSP	SaaS, PaaS
Scenario III	DDI	KP	CSP	SaaS, PaaS, IaaS
Scenario IV	MCP	KP	CSP	SaaS, PaaS
Scenario V	MCP	CP	Service User	SaaS, PaaS

Chapter: 6 Conclusion and Future Work

6.1 CONCLUSION

Predicate Based Encryption (PBE) presents an interesting and also novel family of asymmetric encryption schemes. PBE combines Attribute Based Access Control (ABAC) with asymmetric encryption, allowing for a single-encryptor/multi-decryptor environment to be realised using a single scheme. Replicating such functionality using more traditional techniques requires a more complex approach. Even so, PBE is inherently more flexible, data can be encrypted for a decrypting entity prior to the creation of the decrypting entity's decryption key: The precise list of decrypting entities need not be known a priori. Such novelty lies with the cryptographic keys. Cryptographic keys are not just numbers, there is no one-to-one pairing of encryption and decryption keys, and encryption and decryption keys are created separately from each other.

Though understanding the operation and potential use of PBE schemes was not inherently diffcult it was not trivial either. Understanding the means through which PBE schemes can be constructed also proved to be a somewhat tedious affair. Unfortunately with PBE, a disproportionate amount of time was spent searching for, and collating, information from different sources; different papers begat different schemes which in turn begat different terminology. How-ever, these efforts were not without dividends, by learning more about PBE scheme construction,various issues surrounding cryptographic keys were identified and addressed. For instance, blinding values prevent decrypting entities from combining their decryption keys.The inclusion of numerical attributes also presents an underlying hidden cost ,which together with key revocation techniques will increase the space needed to store decryption keys. Furthermore, by looking at different schemes a means to categorise the different PBE schemes emerged from such schemes' emergent properties.

PBE schemes can be used to protect service user's data in three different scenarios: Scenario I saw the inclusion of PBE within a service; Scenario II saw the provision of PBE as-a-Service ; and Scenario V saw PBE being deployed by the user themselves. In each of these three scenarios PBE can be used by service users to specify precisely with whom they wish to share their data, for what purpose, and for how long. Although Scenario V may be a privacy zealot's ideal choice-- they are in full control--its practical feasibility has yet to be determined; the ability for service users' to act competently as a Key Authority is still unclear. The remaining two scenarios, on the other hand, do appear to be more promising. However, these scenarios in themselves do present a dilemma between usability and the guarantees made over end-to-end security.

When looking to protect CSP's data, PBE can facilitate keyword search with complex queries over encrypted data: Scenario III by the CSP; and in Scenario IV by a service user. This use of PBE is rather interesting in that the focus of these scenarios is on the CSP and not service user, and is most certainly worthy of further investigation.

The use of PBE within the cloud appears to be concentrated at both the PaaS and SaaS service layers. Though some may be surprised at PBE's lack of use at the IaaS layer, this was not totally unexpected. The primary interaction between a service user and CSP at this level is over managing virtual machines: Not much else happens.

6.2 FUTURE WORK

Areas of future work that should be addressed include:

- ✓ The effect that access policy composition, especially concerning numerical comparisons, has upon the performance of the encryption, decryption and key generation functions.
- ✓ The QoS measurements and guarantees that can be made, aside from function effciency,over PBE.

- ✓ The effect that access policy composition has upon, if any, the size of cipher-text produced.
- ✓ The representation (encoding) of access policies/rules and list of attributes.
- ✓ Existing standards and technologies that should be leveraged or adhered to.
- ✓ The construction of a means through which policy rule administration, for both users and CSPs, can be achieved.

References

1. [INT+01] Introduction to Cloud Computing, http://www.priv.gc.ca/resource/fs-fi/02_05_d_51_cc_e.pdf
2. [CSP+02] “Cloud Security and Privacy” by Tim Mather, Subra Kumaraswamy, and Shahed Latif.
3. [Vas09] Jessica E. Vascellaro. Google Discloses Privacy Glitch. English. Wall Street Journal. Mar. 2009. url: http://blogs.wsj.com/digits/2009/03/08/1214/.
4. [HM+05] Ragib Hasan, Suvda Myagmar et al. `Toward a threat model for storage systems'. In: StorageSS '05: Proceedings of the 2005 ACM workshop on Storage security and survivability. Fairfax, VA, USA: ACM, 2005, pp. 94{ 102. isbn: 1-59593-233-X. doi: http://doi.acm.org/10.1145/1103780. 1103795.
5. [BM03] Philippa J. Broadfoot and Andrew P. Martin. A Critical Survey of Grid Security Requirements and Technologies. Tech. rep. PRG-RR-03-15. Wolf-son Building Oarks Road Oxford OX1 3QD: Oxford University Computing Laboratory, 2003. url: http://www.comlab.ox.ac.uk/files/930/RR-03-15.ps.gz.
6. [GTF08] Saikat Guha, Kevin Tang and Paul Francis. `NOYB: privacy in online social networks'. In: WOSP '08: Proceedings of the first workshop on Online social networks. Seattle, WA, USA: ACM, 2008, pp. 49{54. isbn: 978-1-60558-182-8. doi: http://doi.acm.org/10.1145/1397735.1397747.
7. [MP09] Miranda Mowbray and Siani Pearson. `A client-based privacy manager for cloud computing'. In: COMSWARE '09: Proceedings of the Fourth Inter-national ICST Conference on COMmunication System softWAre and mid-dlewaRE. Dublin, Ireland: ACM, 2009, pp. 1{8. isbn: 978-1-60558-353-2. doi: http://doi.acm.org/10.1145/1621890.1621897.
8. [DVZ10] Gabriele D'Angelo, Fabio Vitali and Stefano Zacchirolo. `Content Cloacking: Preserving Privacy with Google Docs and other Web Applications'. To appear in the 25th Symposium On Applied Computing (SAC'10), March 22-26, 2010, Sierre Switzerland. Mar. 2010.
9. [Kan08] Murat Kantarcioglu. `A Survey of Privacy-Preserving Methods Across Ho-rizontally Partitioned Data'. In: Privacy-Preserving Data Mining. Ed. By Ahmed K. Elmagarmid, Amit P. Sheth et al. Vol. 34. Advances in Data-base Systems. Springer US, 2008, pp. 313{335. isbn: 978-0-387-70992-5. url: http://dx.doi.org/10.1007/978-0-387-70992-5_13.

10. [Nis04] Helen Nissenbaum. `Privacy as Contextual Integrity'. In: Washington Law Review 79.1 (2004). url: http://ssrn.com/abstract=534622.
11. [Sol07] Daniel J. Solove. `'I've got nothing to hide' and Other Misunderstandings of Privacy'. In: San Diego Law Review 44 (2007). GWU Law School Pub-lic Law Research Paper No. 289, pp. 745{772. url: http://ssrn.com/abstract=998565
12. [Hed10] Marc Hedlund. Protecting \Cloud" Secrets with Grendel. English. Wesabe, Inc. Jan. 2010. url: http://blog.wesabe.com/2010/01/04/protecting-cloud-secrets-with-grendel/
13. [KSW08] Jonathan Katz, Amit Sahai and Brent Waters. `Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products'. In: Ad-vances in Cryptology { EUROCRYPT 2008 (2008), pp. 146{162. url: http://dx.doi.org/10.1007/978-3-540-78967-3_9.
14. [Sha85] Adi Shamir. `Identity-Based Cryptosystems and Signature Schemes'. In: Advances in Cryptology (1985), pp. 47-53. doi: 10.1007/3-540-39568-7_5
15. [BF01] Dan Boneh and Matt Franklin. `Identity-Based Encryption from the Weil Pairing'. In: Advances in Cryptology | CRYPTO 2001 2139/2001 (2001), pp. 213-229. doi: 10.1007/3-540-44647-8_13.
16. [GS+06] Vipul Goyal, Amit Sahai et al. `Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data'. In: Conference on Computer and Communications Security: Proceedings of the 13th ACM conference on Computer and communications security. Oct. 2006.
17. [BW07] Dan Boneh and Brent Waters. `Conjunctive, Subset, and Range Queries on Encrypted Data'. In: Theory of Cryptography. Ed. by Salil Vadhan. Vol. 4392. Lecture Notes in Computer Science. Springer Berlin / Heidel-berg, 2007, pp. 535-554. doi: 10.1007/978-3-540-70936-7_29. url: http://dx.doi.org/10.1007/978-3-540-70936-7_29.
18. [RSA78] R. L. Rivest, A. Shamir and L. Adleman. `A method for obtaining Digital signatures and public-key cryptosystems'. In: Commun. ACM 21.2 (1978), pp. 120{126. issn: 0001-0782. doi: http://doi.acm.org/10.1145/35934 0.359342.
19. [ElG85] T. ElGamal. `A public key cryptosystem and a signature scheme based on discrete logarithms'. In: IEEE Transactions on Information Theory. 31 (1985). Could not find PDF, pp. 469-472
20. [YT05] Eric Yuan and Jin Tong. `Attributed Based Access Control (ABAC) for Web Services'. In: Proceedings of the IEEE International Conference on Web Services. ICWS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 561{569. isbn: 0-7695-2409-5. doi: http://dx.doi.org/10.1109/ICWS.2005.25. url: http://dx.doi.org/10.1109/ICWS.2005.25.
21. [SC+96] Ravi S. Sandhu, Edward J. Coyne et al. `Role-Based Access Control Models'. In: Computer 29.2 (1996), pp. 38{47. issn: 0018-9162. doi: http://doi.ieeecomputersociety.org/10.1109/2.485845.

22. [SS94] R S Sandhu and P Samarati. `Access Control: Principle and Practice'. In:IEEE COmmunications Magazine 32.9 (Sept. 1994), pp. 40-48
23. [BDC+04] Dan Boneh, Giovanni Di Crescenzo et al. `Public Key Encryption with Keyword Search'. In: Advances in Cryptology - EUROCRYPT 2004 3027/2004 (2004), pp. 506{522. url: http://www.springerlink.com/content/0hafhrbbvt2l7vn3