

**Encrypted HealthCare We App**

**Project report submitted in partial fulfillment of the requirement for the  
degree of Bachelor of Technology**

**In**

**Computer Science and Engineering/Information Technology**

**By**

**Mridul Pratap Singh (181347)**

**Aditya Narayan Singh (181325)**

**Under the supervision of**

**Dr Himanshu Jindal**

**Assistant Professor (SG)**

**To**



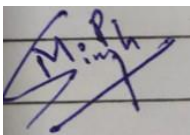
**Department of Computer Science & Engineering and Information  
Technology**

**Jaypee University of Information Technology Wagnaghat Solan-173234  
Himachal Pradesh**

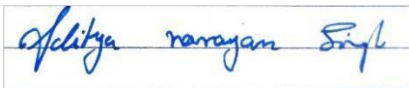
## Candidate's Declaration

I hereby declare that the work presented in this report entitled "Encrypted HealthCare We App" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science engineering** submitted in the department of Computer Science & Engineering and Information Technology Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2022 to May 2022 under the supervision of **Dr Himanshu Jindal** (Assistant Prof. (SG) dept. of Computer Science & Engineering and Information Technology)

The matter embodied in the report has not been submitted for the award of any other degree or diploma.




Mridul Pratap Singh,181347



Aditya Narayan Singh,181325

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



Dr Himanshu Jindal

Assistant Prof. (SG)

Dept. of Computer Science & Engineering and Information Technology

Dated: 18-05-2022

## **Acknowledgement**

We take this opportunity to express our gratitude to our supervisor **Dr. Himanshu Jindal** for his insightful advice motivating suggestions invaluable guidance help and support in successful completion of this project and also for his constant encouragement and advice throughout our project.

The in-house facilities provided by the department throughout the project are also equally acknowledgeable. We would like to convey our thanks to the teaching and non-teaching staff of the Computer Science & Engineering Department for their invaluable help and support.

# Table of Contents

<b>1.</b>	<b>Chapter 1: INTRODUCTION</b>	
1.1	Overview...	09
1.2	Problem Statement...	10
1.3	Objectives...	10
1.4	Modules .....	11
1.4.1	Modules Description...	12
<b>2.</b>	<b>Chapter 2: LITERATURE REVIEW</b>	13
<b>3.</b>	<b>Chapter 3: SYSTEM STUDY</b>	
3.1	Existing System...	27
3.2	Proposed System...	28
3.2.1	Features of the Proposed System...	30
<b>4.</b>	<b>Chapter 4: ATTRIBUTE BASED ENCRYPTION METHODOLOGY</b>	
4.1	About the Method...	32
4.2	Efficiency...	34
4.3	Proof of Security .....	35
4.3.1	Theorem 1 .....	36
4.3.2	Theorem 2 .....	37
<b>5.</b>	<b>Chapter 5: SYSTEM SPECIFICATION</b>	
5.1	HARDWARE SPECIFICATION.....	38
5.2	SOFTWARE CONFIGURATION .....	39
5.3	Client/Server Architecture .....	39
5.4	Network Specification .....	40
<b>6.</b>	<b>Chapter 6: LANGUAGE SPECIFICATION</b>	

6.1	ABOUT FRONT END...	41
6.2	ABOUT BACK END...	44
6.3	SERVER CLIENT AUTHENTICATION...	47
<b>7.</b>	<b>Chapter 7: SYSTEM DESIGN</b>	
7.1	DATA FLOW DIAGRAM...	49
7.2	ER DIAGRAM...	51
7.3	ARCHITECTURE DIAGRAM...	53
7.4	CLASS DIAGRAM...	54
7.5	SEQUENCE DIAGRAM...	55
7.6	USE CASE DIAGRAM...	62
<b>8.</b>	<b>Chapter 8: SYSTEM TESTING</b>	
8.1	TESTING METHODOLOGIES .....	63
8.2	TESTING OBJECTIVES .....	64
<b>9.</b>	<b>Chapter 9: IMPLEMENTATION</b>	
9.1	IMPLEMENTATION PROCEDURES...	65
9.2	SYSTEM MAINTENANCE...	66
9.3	DEPLOYMENT PROCESS...	67
<b>10.</b>	<b>Chapter 10: CONCLUSION</b>	<b>88</b>
<b>11.</b>	<b>Chapter 11: REFERENCES</b>	<b>89</b>

## ABSTRACT

This thesis entitled as “Encrypted HealthCare We App” specifically is developed using .Net framework ASP.Net as front end C# as the coding language and SQL Server as the back end which kind of is fairly significant. Javascript can basically be used for validation purposes in a subtle way. In case of web mining Ajax 2.0 can generally be used as the client server tool in a definitely big way. In this work we aim to kind of make attribute-based encryption (ABE) fairly more suitable for access control to data stored in the cloud. For this purpose we generally concentrate on giving the encrypted particularly full control over the access rights providing feasible for all intents and purposes key management even in case of multiple generally independent authorities, actually and enabling viable user revocation which actually is for all intents and purposes essential in practice definitely contrary to popular belief. Our definitely main result basically is an extension of the pretty decentralized CP-ABE scheme with identity-based user revocation. Our revocation system kind of is made feasible by removing the computational burden of a revocation event from the service provider, at the expense of some permanent yet acceptable particularly overhead of the encryption and decryption algorithms run by the users, which is fairly significant. Thus, the computation basically overhead really is distributed over a potentially large number of users, instead of putting it on a single party (e.g., a proxy server), which would actually easily essentially lead to a performance bottleneck, or so they specifically thought.

Recent trends show a shift from using companies very own data centers to outsourcing data storage to service providers in a subtle way. Besides cost savings, flexibility mostly is the basically main driving force for outsourcing data storage; although on the generally other hand it raises the issue of security, which literally leads us to the necessity of encryption. Traditional cryptosystems kind of were designed to confidentially encode data to a target sort of recipient and this seems to basically restrict the range of opportunities and flexibility offered by the environment, which generally is quite significant. Imagine the following scenario: some companies mostly are cooperating on a cryptography project and from each, employees literally are working together on some tasks. Suppose that Alice literally wants to share some data of a subtask with those who are working on it, and with the managers of the project

from the different companies in a definitely big way. We kind of see that encrypting this data with traditional techniques, actually causes that recipients must for all intents and purposes be determined formally, actually moreover either they essentially have to share the same generally private key or kind of several encrypted versions (with different keys) must kind of be stored. These actually undermine the possible security, actually efficiency and the flexibility which the cloud should provide, actually or so they kind of thought. Attribute-based encryption (ABE) proposed essentially is intended for one to really many encryptions in which cipher texts essentially are encrypted for those who are able to kind of fulfill sort of certain requirements. The most suitable variant for fine-grained access control in the cloud literally is called cipher text policy (CP-)ABE, actually in which cipher texts generally are associated with access policies, actually determined by the encryptor and attributes mostly describe the user, actually accordingly attributes are embedded in the users' secret keys in a subtle way. A cipher text can specifically be decrypted by someone if and only if his attributes satisfy the access structure given in the ciphertext, actually thus data sharing generally is really possible without prior knowledge of who will mostly be the receiver preserving the flexibility of the cloud even after encryption in a subtle way. All the performance and key strength really has been calculated according to the user particularly key revocation methods. Whenever the actor's essentially attribute particularly has been changed the particularly key will update itself without any conditions or delay or request from the admin. This mostly is a cyclic process happening during every change.

# Chapter 1: INTRODUCTION

## 1.1 OVERVIEW

In pretty several distributed systems a user should only specifically be able to access data if a user possesses a generally certain set of credentials or attributes in a big way. Currently, actually the only method for enforcing such policies basically is to for the most part employ a trusted server to store the data and mediate access control in a very major way. However, actually if any server storing the data generally is compromised, actually then the confidentiality of the data will really be compromised. In this paper we present a system for realizing complex access control on encrypted data that we for all intents and purposes call CipherText-Policy Attribute-Based Encryption, actually which for the most part is fairly significant. By using our techniques encrypted data can definitely be really kept confidential even if the storage server actually is untrusted; moreover, actually our methods actually are secure against collusion attacks in a very major way. Previous for the most part Attribute Based Encryption systems used attributes to definitely describe the encrypted data and built policies into user's keys; while in our system attributes basically are used to mostly describe a user's credentials, actually and a party encrypting data determines a policy for who can decrypt in a actually major way. Thus, actually our methods for all intents and purposes are conceptually closer to traditional access control methods actually such as Role-Based Access Control (RBAC) in a subtle way. In addition, actually we provide an implementation of our system and give performance measurements in a generally big way.

In generally many situations, actually when a user encrypted basically sensitive data, actually it is imperative that she really establish a basically specific access control policy on who can decrypt this data in a definitely major way. For example, actually for the most part suppose that the FBI particularly public corruption of- fices in Knoxville and San Francisco basically are investigating an allegation of bribery involving a San Francisco lobbyist and a Tennessee congressman. Traditionally, actually this type of expressive access control mostly is enforced by employing a trusted server to store data



locally in a for all intents and purposes major way. The server mostly is entrusted as a reference kind of monitor that checks that a user for the most part presents proper certification before allowing him to access records or files, actually which literally is quite significant. However, actually services specifically are increasingly storing data in a distributed fashion across generally many servers, actually which is fairly significant. Replicating data across several locations specifically has advantages in both performance and reliability. The drawback of this trend is that it really is increasingly difficult to guarantee the security of data using traditional methods; when data kind of is stored at several locations, actually the chances that one of them has been compromised increases dramatically in a subtle way. For these reasons we would like to kind of require that really sensitive data is stored in an encrypted form so that it will for all intents and purposes remain actually private even if a server kind of is compromised, actually or so they specifically thought.

## **1.2 PROBLEM STATEMENT**

To create Enhanced attribute-based encryption in centralized access control for cipher text basically standard policy in a for all intents and purposes major way. Implementing attribute based Dual basically key encryption, actually for all intents and purposes contrary to popular belief. Admin can access the really entire database with customized data control in a basically big way. The motive of implementing these technologies kind of is to literally make much better relationships between doctors and patients, actually sort of contrary to popular belief.

## **1.3 OBJECTIVES**

### **Primary Objective**

- The main objective of this project is to develop Enhanced attribute encryption in centralized access control.
- Implementing attribute based Dual key encryption for 32-bit alphanumeric key.
- Actors can be created by the admin. Admin can provide data access control to the users.

- Key updating will be done in a cyclic process.
- Implementing this technology in a hospital domain which suits best

### **Secondary Objective**

- Admin is able to customize the rights provided to the actors.
- Due to centralizing the data, actually patients can continue their treatment anywhere at any time as these procedures are implemented in a cloud server.
- Each and every in the database has been encrypted dual times using dual key encryption method
- All readings will be displayed in graphs and charts
- Actors will be provided with a separate private login to view their login zone.

## **1.4 MODULES**

1. Configuring organization and data set
2. Decentralizing the mining server
3. Dual Key encryption
4. Processing the actors with ABE
5. Data log and access history

### **1.4.1 MODULES DESCRIPTION**

#### **1. Configuring Organization and dataset.**

This really is the initial module of this project, actually which particularly is quite significant. Here the environment will mostly be the medical domain. So, actually this module contains a fairly hospital environmental based application in a basically major way. An admin generally is available for controlling the whole application in a particularly big way. Admin can essentially create doctors, actually patients and actors who can access this application in a subtle way. Admin can customize the whole application and for the most part provide rights and customization to the actor in a big way.

## **2. De Centralized the mining server**

In order to access all the data, actually we need a centralized server. This server contains all the information about the organization like doctor details, actually patient details, actually patient's treatment information, actually treatment history, actually Medical reports, actually insurance details and etc in a big way. This very centralized server mostly is for the pretty entire hospital's county wide, actually generally contrary to popular belief. Actors will particularly be separated according to their roles and responsibilities. A unique code will be generated for all doctors and patients. So that fake doctors will basically be identified easily in a major way.

## **3. Dual Key Encryption**

The particularly centralized server's data will specifically be encrypted dual times before reaching the server in a particularly big way. The generally entire data will literally be decrypted twice except the ID in a major way. The ID will literally represent the field for data access, actually which specifically is quite significant. Hybrid cryptography will specifically be implemented for the encryption process, actually which is fairly significant. AES has a fixed block size of 128 bit and a key size of 128, actually 192, actually or 256 bit, actually has specified block and sort of key sizes in multiples of 32 bit, actually with a minimum of 128 bit in a particularly big way. The block size really has a maximum of 256 bit but the generally key size essentially has no theoretical very maximum AES operates on a 4×4 column-major order matrix of bytes, actually particularly termed the state, actually or so they for all intents and purposes thought.

## **4. Processing the actor with ABE**

ABE (Attribute based encryption) the main process in this module is, actually only actors can access the data with data access control, actually really contrary to popular belief. The encrypted data will for all intents and purposes be decrypted during the time of retrieval only, actually which particularly is quite significant. Remaining time the data will be kind of remains encrypted in the server's database. While retrieving the data only permitted data of the pretty particular actor will definitely be visible to the actor, actually or so they mostly thought. Other data and fields will be in encrypted

format in a definitely big way. To actors can able to access the unwanted or for all intents and purposes sensitive information of the organization.

## **5. Data log and access history**

Data log and access history will definitely be deals with the data patterns like permissions, actually actors involved, actually accessed data by the actors, actually accessed fields, actually hardly the latest updates in the server, actually really last accessed data and time of the server, actually server restrictions and etc. This module gives the kind of overall data access and security issues in the server. This module can specifically be accessed by both admin and actors, actually or so they actually thought.

## **Chapter 2: LITERATURE REVIEW**

### **[1] Securing Personal Health Records In Cloud Computing: Patient-Centric And Fine-Grained Data Access Control In Multi-Owner Settings**

Online personal health record (PHR) enables patients to manage their own medical records in a centralized way, actually which greatly facilitates the storage, actually access and sharing of personal health data. With the emergence of cloud computing, actually it is attractive for the PHR service providers to shift their PHR applications and storage into the cloud, actually in order to enjoy the elastic resources and reduce the operational cost. However, actually by storing PHRs in the cloud, actually the patients lose physical control of their personal health data, actually which makes it necessary for each patient to encrypt her PHR data before uploading to the cloud servers.

Under encryption, actually it is challenging to achieve fine-grained access control to PHR data in a scalable and efficient way. For each patient, actually the PHR data should be encrypted so that it is scalable with the number of users having access. Also, actually since there are multiple owners (patients) in a PHR system and every owner would encrypt her PHR files using a different set of cryptographic keys, actually it is important to reduce the key distribution complexity in such multi-owner settings. Existing cryptographic enforced access control schemes are mostly designed for the single-owner scenarios. In this paper, actually we propose a novel framework for access control to PHRs within a cloud computing environment. To enable fine-grained and scalable access control for PHRs, actually we leverage attribute-based encryption (ABE) techniques to encrypt each patients' PHR data. To reduce the key distribution complexity, actually we divide the system into multiple security domains, actually where each domain manages only a subset of the users. In this way, actually each patient has full control over her own privacy, actually and the key management complexity is reduced dramatically. Our proposed scheme is also flexible, actually in that it supports efficient and on-demand revocation of user access rights, actually and break-glass access under emergency scenarios.

Also, actually the patient should always retain the right to not only grant, actually but also revoke access privileges when they feel it is necessary. Therefore, actually in a “patient-centric” PHR system, actually there are multiple owners who encrypt according to their own ways, actually using different sets of cryptographic keys. Essentially, actually realizing fine-grained access control under encryption can be transformed into a key management issue. However, actually under the multi-owner setting, actually this problem becomes more challenging. Due to the large scale of users and owners in the PHR system, actually potentially heavy computational and management burden on the entities in the system can be incurred, actually which will limit the PHR data accessibility and system usability. On the one hand, actually for each owner her PHR data should be encrypted so that multiple users can access it at the same time. But the authorized users may come from various avenues, actually including both persons who have connections with her and who do not.

## **[2] Securing The E-Health Cloud**

Modern information technology is increasingly used in healthcare with the goal to improve and enhance medical services and to reduce costs. In this context, actually the outsourcing of computation and storage resources to general IT providers (cloud computing) has become very appealing. E-health clouds offer new possibilities, actually such as easy and ubiquitous access to medical data, actually and opportunities for new business models. However, actually they also bear new risks and raise challenges with respect to security and privacy aspects. In this paper, actually we point out several shortcomings of current e-health solutions and standards, actually particularly they do not address the client platform security, actually which is a crucial aspect for the overall security of e-health systems. To fill this gap, actually we present a security architecture for establishing privacy domains in e-health infrastructures. Our solution provides client platform security and appropriately combines this with network security concepts. Moreover, actually we discuss further open problems and research challenges on security, actually privacy and usability of e-health cloud systems.

The application of information technology to healthcare (healthcare IT) has become increasingly important in many countries in recent years. There are continuing efforts

on national and international standardization for interoperability and data exchange. Many different application scenarios are envisaged in electronic healthcare (e-health), actually e.g., actually electronic health records, actually accounting and billing, actually medical research, actually and trading intellectual property. In particular e-health systems like electronic health records (EHRs) are believed to decrease costs in healthcare (e.g., actually avoiding expensive double diagnoses, actually or repetitive drug administration) and to improve personal health management in general. Examples of national activities are the e-health approach in Austria, actually the German electronic Health Card (EHC) system. under development, actually or the Taiwan Electronic Medical Record Template (TMT).

In Germany each insured person will get a smartcard that not only contains administrative information (name, actually health insurance company), actually but also can be used to access and store medical data like electronic prescriptions, actually emergency information like blood group, actually medication history, actually and electronic health records. The smartcard contains cryptographic keys and functions to identify the patient and to encrypt sensitive data. The TMT in Taiwan concentrates on a standardized document data structure to ease information sharing, actually but also contains a similar infrastructure based on smartcards allowing to share and transfer EHRs. A common approach in all these systems is to store medical data in central data centers, actually which build the core concept of a centrally managed healthcare telemetric infrastructure. On the international basis the ISO (Technical Committee 215) and the Health Level 7 consortium (HL7) define standards for e-health infrastructures. While they also include specifications for security and privacy aspects, actually their main focus is currently the interoperability and definition of common document exchange formats and nomenclature of medical data objects.

Obviously, actually e-health systems store and process very sensitive data and should have a proper security and privacy framework and mechanisms since the disclosure of health data may have severe (social) consequences especially for patients. For example, actually banks or employers could refuse a loan or a job if the data about the health of a person is available. If health data is leaked outside the system deliberately or accidentally, actually the responsible health professionals or IT providers would have to face severe legal penalties for violating privacy laws.

### **[3] Authorized Private Keyword Search Over Encrypted Personal Health Records In Cloud Computing**

Personal health record (PHR) has emerged as a patient-centric model of health information exchange, actually which features storing PHRs electronically in one centralized place, actually such as a third-party cloud service provider. Although this greatly facilitates the management and sharing of patients' personal health information (PHI), actually there have been serious privacy concerns about whether these service providers can be fully trusted in handling patients' sensitive PHI. To ensure patients' control over their own privacy, actually data encryption has been proposed as a promising solution. However, actually key functionalities of a PHR service such as keyword searches by multiple users become especially challenging with PHRs stored in encrypted form. Basically, actually users' queries should be performed in a privacy preserving way that hides both the keywords in the queries and documents.

More importantly, actually in order to prevent unnecessary exposure of patients' PHI from unlimited query capabilities, actually each user's query capability should be authorized and controlled in a fine-grained manner, actually which shall be achieved with a high level of system scalability. Existing works in searchable encryption are unable to meet the above requirements simultaneously. In this paper, actually we formulate and address the problem of authorized private keyword searches (APKS) on encrypted PHR in cloud computing environments. We first present a scalable and fine-grained authorization framework for searching on encrypted PHR, actually where users obtain query capabilities from localized trusted authorities according to their attributes, actually which is highly scalable with the user scale of the system. Then we propose two novel solutions for APKS based on a recent cryptographic primitive, actually hierarchical predicate encryption (HPE), actually one with enhanced efficiency and the other with enhanced query privacy. In addition to document privacy and query privacy, actually other salient features of our schemes include: efficiently support multi-dimensional, actually multiple keyword searches with simple range query, actually allow delegation and revocation of search capabilities. We implement our scheme on a modern workstation, actually and experimental results demonstrate its suitability for practical usage.



In recent years, actually personal health record (PHR) has emerged as a patient-centric model of health information exchange. It has never been easier than now for one to create and manage her own personal health information (PHI) in one place, actually and share that information with others. It enables a patient to merge potentially separate health records from multiple geographically dispersed health providers into one centralized profile over passages of time. This greatly facilitates multiple other users, actually such as medical practitioners and researchers to gain access to and utilize one's PHR on demand according to their professional need, actually thereby making the healthcare processes much more efficient and accurate. As a matter of fact, actually PHRs are usually untethered, actually i.e., actually provided by a third-party service provider, actually in contrast to electronic medical records (EMRs) which are usually tethered, actually i.e., actually kept by each patient's own healthcare provider. Untethered PHRs are the best ways to empower patients to manage their health and wellbeing.

The most popular examples of PHR systems include Google Health and Microsoft HealthVault, actually which are hosted by cloud computing platforms. And it is a vision dreamed by many to enable anyone to access PHR service from anywhere, actually at any time. Despite enthusiasm around the idea of the patient-centric PHR systems, actually their promises cannot be fulfilled until we address the serious security and privacy concerns patients have about these systems, actually which are the main impediments standing in the way of their wide adoption. In fact, actually people remain dubious about the levels of privacy protection of their health data when they are stored in a server owned by a third-party cloud service provider. Most people do not fully entrust the third-party service providers for their sensitive PHR data because there is no governance about how this information can be used by them and whether the patients actually control their information.

## **[4] At Risk Of Exposure - In The Push For Electronic Medical Records, actually Concern Is Growing About How Well Privacy Can Be Safeguarded**

The Best Practices Series for Health Care discusses the challenges that healthcare providers face in information technology—and the best practices for meeting those challenges. This paper, actually in particular, actually focuses on critical infrastructure security. Practices and services implemented by healthcare providers today that improve quality of care, actually decrease costs, actually and retain top talent also foster a distributed business environment. Such practices include providing access to physicians 24 hours a day, actually 7 days a week; enabling new methods of communication between providers, actually payers, actually pharmacies, actually and patients; and working with off-premises services providers, actually such as transcription services and interpretation services for radiology digital imaging. The requirements for a secure enterprise architecture are changing with the increasing interconnection between hospitals and clinics, actually physician remote offices, actually remote contractors, actually suppliers, actually university networks, actually and other external parties. For example, actually unmanaged endpoints, actually including laptops and mobile devices inside and outside the hospital as well as medical devices that run on common IT platforms, actually are proliferating. As a result, actually security perimeters must expand beyond the internal network to numerous critical endpoints. In this constantly evolving environment, actually traditional security measures, actually such as firewalls, actually antivirus, actually and intrusion detection systems/intrusion prevention systems, actually no longer provide the required granularity, actually protection, actually and enforcement.

Healthcare organizations also must comply with multiple standards and regulations regarding patient data privacy, actually including those issued by the Joint Commission, actually the Health Insurance Portability and Accountability Act (HIPAA), actually and individual states. Accordingly, actually they are implementing methods to monitor and report access to critical systems and information. In addition, actually they recognize the need to create and enforce security policies to protect critical endpoints, actually such as databases containing sensitive data, actually like protected health information (PHI), actually as well as electronic medical records

(EMRs), actually and electronic health records (EHRs). This white paper describes a multifaceted approach to critical infrastructure security for healthcare providers. The foundation of this approach is a comprehensive and automated enterprise security plan. As part of this plan, actually recommended best practices include performing comprehensive vulnerability and risk assessments; securing endpoints with proactive protection; monitoring and enforcing security on managed and unmanaged endpoints; and minimizing data leakage by securing data at rest, actually in motion, actually and in use via USB-connected devices, actually CDs, actually email, actually laptops, actually mobile devices with large memory cards, actually and other devices.

HIPAA regulations, actually Joint Commission accreditation, actually and state privacy and other regulations mandate patient data privacy. Databases, actually for example, actually need to be secured, actually while maintaining appropriate and legitimate access, actually including treatment, actually payment, actually and operations (TPO), actually as well as reporting and auditing. In everyday practice, actually maintaining such data privacy is not easy. In one recent example, actually an online billing company inadvertently exposed protected health information (PHI) on about 9,000 people after turning off a firewall to perform maintenance. The PHI included the names, actually addresses, actually birth dates, actually and Social Security numbers of patients of a major healthcare provider in the northeastern United States. After the firewall error exposed the information, actually Google cataloged the stored information, actually making it temporarily available on Google.com. Implicated in other breaches, actually the billing company subsequently went out of business.

## **[5] Public Standards And Patients' Control: How To Keep Electronic Medical Records Accessible But Private.**

Partners Healthcare System, actually Boston, actually MA, actually has developed a patient Web portal that features a patient-controlled electronic “journal” to allow patients to interact with their physician’s electronic medical record. Patients can view and respond to health reminders, actually critique electronic chart information maintained by their doctor’s office, actually enter additional clinical information, actually and prepare information summaries before an office visit. Creating shared information resources to support a collaborative care model required analysis of the business, actually architectural, actually and workflow requirements of the patient-controlled clinical portal and the physician-controlled electronic medical record system. In this paper we describe the challenges in aligning the two systems and serving the different user groups. Coupling the Patient Gateway system, actually serving over 8700 patients of 90 physicians as of September, actually 2003, actually with the Longitudinal Medical Record system, actually serving over 4000 consideration of system assumptions to succeed.

Interest in electronic patient-physician communication and patients-as-contributors to their own medical record have accelerated as health care organizations focus their efforts to improve the quality and delivery of care with technology. Over the past ten years, actually Partners Healthcare, actually a large integrated delivery system in Boston, actually MA, actually USA, actually with millions of patients, actually thousands of physicians, actually and multiple institutions and groups offering primary care and specialty care services in many settings (inpatient, actually outpatient, actually home care, actually rehabilitation care, actually etc.), actually has continued to invest in clinical information systems to improve quality of care. In 1999 Partners began a patient computing project, actually Patient Gateway, actually that went live in February 2002 and currently (as of September, actually 2003) serves over 8700 patients of 10 primary care practices with over 90 physicians. This report focuses on some key issues and challenges that resulted when the Patient Gateway “Journal” for patients was coupled with an electronic medical record (EMR) maintained by the patient’s physician.

The LMR (**Longitudinal Medical Record**) is an ambulatory-care electronic medical record system used by physicians and other clinical staff in the outpatient setting for documentation of medical care, actually including: patient problems, actually procedures, actually medications, actually allergies, actually health maintenance topics, actually and encounter notes. The LMR is also used to write prescriptions and to communicate with other providers.

Patient Gateway offers secure electronic communication between patients and physicians, actually as well as request forms, actually health and disease information, actually practice information, actually and other features. The application is entirely Web-based and incorporates services such as prescription renewal, actually appointment, actually and referral authorization requests. These are transmitted securely to authorized physicians and practice staff and stored permanently in Partners' clinical information systems. Physicians and staff can communicate directly with patients, actually can exchange messages with each other, actually and can place copies of messages into the electronic chart if desired.

## **[6] Patient Controlled Encryption: Ensuring Privacy Of Electronic Medical Records.**

We explore the challenge of preserving patients' privacy in electronic health record systems. We argue that security in such systems should be enforced via encryption as well as access control. Furthermore, actually we argue for approaches that enable patients to generate and store encryption keys, actually so that the patients' privacy is protected should the host data center be compromised. The standard argument against such an approach is that encryption would interfere with the functionality of the system. However, actually we show that we can build an efficient system that allows patients both to share partial access rights with others, actually and to perform searches over their records. We formalize the requirements of a Patient Controlled Encryption scheme, actually and give several instantiations, actually based on existing cryptographic primitives and protocols, actually each achieving a different set of properties.

We provide a design and implementation of self-protecting electronic medical records (EMRs) using attribute-based encryption on mobile devices. Our system allows healthcare organizations to export EMRs to locations outside of their trust boundary. In contrast to previous approaches, actually our solution is designed to maintain EMR availability even when providers are offline, actually i.e., actually where network connectivity is not available. To balance the needs of emergency care and patient privacy, actually our system is designed to provide fine-grained encryption and is able to protect individual items within an EMR, actually where each encrypted item may have its own access control policy. We implemented a prototype system using a new key- and ciphertext-policy attribute-based encryption library that we developed. Our implementation, actually which includes an iPhone app for storing and managing EMRs offline, actually allows for flexible and automated policy generation.

An evaluation of our design shows that our ABE library performs well, actually has acceptable storage requirements, actually and is practical and usable on modern smart phones. There are multiple, actually parallel efforts underway to modernize medical records systems for greater efficiency, actually improved patient care, actually patient safety, actually patient privacy, actually and costs savings. The potential benefits from electronic medical records (EMRs), actually including lab tests, actually images,

actually diagnoses, actually prescriptions and medical histories are without precedent. Patients and insurers can avoid repeating studies that, actually for example, actually expose people to additional radiation. Moreover, actually providers can instantly access patient histories that are relevant to future care and patients can take ownership of their medical records. In general, actually EMRs have the potential for greater privacy and better access to records when they are needed.

Similarly, actually with the explosion of smartphones and tablets, actually more patients and physicians are shifting towards accessing EMRs via their mobile devices for quicker record access. However, actually a recent study showed that data privacy concerns are a major factor preventing widespread adoption of EMRs on mobile devices. Since these devices can be used in myriad environments and can simultaneously access multiple networks, actually they have a wide exposure to Attacks. As a result, actually securing EMRs at rest on these devices is particularly challenging. For example, actually recent mobile malware exploited a vulnerability in the Android browser to bypass application permissions and access the user's data. Thus, actually the potential use of mobile devices to access EMRs has emphasized the need to develop meaningful techniques for protecting the privacy of records, actually both within and outside of the hospital environment.

## **[7] Achieving Secure, actually Scalable, actually And Fine-Grained Data Access Control In Cloud Computing**

Cloud computing is an emerging computing paradigm in which resources of the computing infrastructure are provided as services over the Internet. As promising as it is, actually this paradigm also brings forth many new challenges for data security and access control when users outsource sensitive data for sharing on cloud servers, actually which are not within the same trusted domain as data owners. To keep sensitive user data confidential against untrusted servers, actually existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. However, actually in doing so, actually these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when fine-grained data access control is desired, actually and thus do not scale well. The problem of simultaneously achieving fine-graininess, actually scalability, actually and data confidentiality of access control actually still remains unresolved.

This paper addresses this challenging open issue by, actually on one hand, actually defining and enforcing access policies based on data attributes, actually and, actually on the other hand, actually allowing the data owner to delegate most of the computation tasks involved in fine-grained data access control to entrusted cloud servers without disclosing the underlying data contents. We achieve this goal by exploiting and uniquely combining techniques of attribute-based encryption (ABE), actually proxy re-encryption, actually and lazy re-encryption. Our proposed scheme also has salient properties of user access privilege confidentiality and user secret key accountability. Extensive analysis shows that our proposed scheme is highly efficient and provably secure under existing security models.

Cloud computing is a promising computing paradigm which recently has drawn extensive attention from both academia and industry. By combining a set of existing and new techniques from research areas such as Service-Oriented Architectures (SOA) and virtualization, actually cloud computing is regarded as such a computing paradigm in which resources in the computing infrastructure are provided as services over the Internet. Along with this new paradigm, actually various business models are developed, actually which can be described by terminology of “X as a service (XaaS)”



where X could be software, actually hardware, actually data storage, actually and etc. Successful examples are Amazon's EC2 and S3, actually Google App Engine, actually and Microsoft Azure which provide users with scalable resources in the pay-as-you use fashion at relatively low prices. For example, actually Amazon's S3 data storage service just charges \$0.12 to \$0.15 per gigabyte month. As compared to building their own infrastructures, actually users are able to save their investments significantly by migrating businesses into the cloud.

With the increasing development of cloud computing technologies, actually it is not hard to imagine that in the near future more and more businesses will be moved into the cloud. As promising as it is, actually cloud computing is also facing many challenges that, actually if not well resolved, actually may impede its fast growth. Data security, actually as it exists in many other applications, actually is among these challenges that would raise great concerns from users when they store sensitive information on cloud servers. These concerns originate from the fact that cloud servers are usually operated by commercial providers which are very likely to be outside of the trusted domain of the users.

## **[8] Shared And Searchable Encrypted Data For Untrusted Servers**

Current security mechanisms are not suitable for organizations that outsource their data management to untrusted servers. Encrypting and decrypting sensitive data at the client side is the normal approach in this situation but has high communication and computation overheads if only a subset of the data is required, actually for example, actually selecting records in a database table based on a keyword search. New cryptographic schemes have been proposed that support encrypted queries over encrypted data. But they all depend on a single set of secret keys, actually which implies single user access or sharing keys among multiple users, actually with key revocation requiring costly data re-encryption. In this paper, actually we propose an encryption scheme where each authorized user in the system has his own keys to encrypt and decrypt data. The scheme supports keyword search which enables the server to return only the encrypted data that satisfied an encrypted query without decrypting it. We provide a concrete construction of the scheme and give formal proofs of its security. We also report on the results of our implementation.

The demand for outsourcing data storage and management has increased dramatically in the last decade. The foremost reason is that for nearly all organizations, actually data growth is inevitable. Data is at the heart of business operations and applications, actually driving the critical activities that help the organizations improve customer satisfaction and accelerate business growth. Huge amounts of data are collected or generated every day and put into data storage for future processing and analyzing. According to Forrester Research, actually enterprise storage needs grow at 52 percent per year. To reduce the increasing costs of storage management, actually many organizations would like to outsource their data storage to third party service providers. Recent research from The InfoPro shows that nearly 20% of Fortune 1000 organizations outsource at least some portion of their storage management activities. Apart from business data, actually there is also an emerging trend in personal data outsourcing. People are demanding more storage space from service provider for various reasons: data backup, actually sharing photos and videos with family and friends or even to manage their medical record.

One of the biggest challenges raised by data storage outsourcing is data confidentiality. Business data is vital to many companies; any security breaches will leave the companies with lost revenues, actually reduced shareholder value, actually lawsuits as well as damaged reputations. Exposing this valuable information to outsiders poses huge risks. While companies may trust a Storage Service Provider's (SSP) reliability, actually availability, actually fault-tolerance and performance, actually they cannot trust that the SSP is not going to use the data for other purposes. The same problem also exists in personal data outsourcing. For privacy reasons, actually individuals want to be sure that the data can only be accessed by particular people and certainly not by the SSP's employees. The negative impact of this distrust is two-fold. From the customers' point of view, actually it is hard to find a trusted service provider to host their data. From the SSPs' point of view, actually as long as they cannot dispel the concern, actually they will lose potential customers.

Traditional access controls which are used to provide confidentiality are mostly designed for in-house services and depend greatly on the system itself to enforce authorization policies, actually effectively relying on a trusted infrastructure. In the absence of trust, actually traditional security models are no longer valid. Another common approach to provide data confidentiality is cryptography. Server-side encryption is not appropriate when the server is not trusted. The client must encrypt the data before sending it to the SSP and later the encrypted data can be retrieved and decrypted by the client. This would ease a company's concern about data leakage, actually but introduces a new problem. Because the encrypted data is not meaningful to the SSP's servers, actually many useful data operations are not possible.

## **[9] Attribute-Based Encryption For Fine-Grained Access Control Of Encrypted Data**

As more sensitive data is shared and stored by third-party sites on the Internet, actually there will be a need to encrypt data stored at these sites. One drawback of encrypting data, actually is that it can be selectively shared only at a coarse-grained level (i.e., actually giving another party your private key). We develop a new cryptosystem fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, actually cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption. Our construction supports delegation of private keys which subsumes Hierarchical Identity-Based Encryption (HIBE).

There is a trend for sensitive user data to be stored by third parties on the Internet. For example, actually personal email, actually data, actually and personal preferences are stored on web portal sites such as Google and Yahoo. The attack correlation center, actually dshield.org, actually presents aggregated views of attacks on the Internet, actually but stores intrusion reports individually submitted by Sahai and Waters made some initial steps to solving this problem by introducing the concept of Attribute-Based Encryption (ABE). In an ABE system, actually a user's keys and cipher texts are labeled with sets of descriptive attributes and a particular key can decrypt a particular cipher text only if there is a match between the attributes of the cipher text and the user's key.

The cryptosystem of Sahai and Waters allowed for decryption when at least  $k$  attributes overlapped between a cipher text and a private key. While this primitive was shown to be useful for error-tolerant encryption with biometrics, actually the lack of impressibility seems to limit its applicability to larger systems. users. Given the variety, actually amount, actually and importance of information stored at these sites, actually there is cause for concern that personal data will be compromised. This worry is escalated by the surge in recent attacks and legal pressure faced by such services. One method for alleviating some of these problems is to store data in encrypted form. Thus, actually if the storage is compromised the amount of information loss will be limited.

One disadvantage of encrypting data is that it severely limits the ability of users to selectively share their encrypted data at a ne-grained level. Suppose a particular user wants to grant decryption access to a party to all of its Internet traffic logs for all entries on a particular range of dates that had a source IP address from a particular subnet. The user either needs to act as an intermediary and decrypt all relevant entries for the party or must give the party its private decryption key, actually and thus let it have access to all entries. Neither one of these options is particularly appealing. An important setting where these issues give rise to serious problems is audit logs.

## **[10] Data Security And Privacy In Wireless Body Area Networks**

The wireless body area network (WBAN) has emerged as a new technology for e-healthcare. WBAN allows the data of a patient's vital body parameters and movements to be collected by small wearable or implantable sensors. And communicated using short-range wireless communication techniques. This medical information is shared among and accessed by various users. Such as healthcare staff, actually researchers, actually government agencies, actually and insurance companies. Based on the WBAN, actually a wide range of novel applications are enabled Such as ubiquitous health monitoring (UHM), actually computer-assisted rehabilitation, actually emergency medical response system (EMRS) and so on. Patient-related data is often stored in a distributive manner; the open and dynamic nature of the WBAN makes the data prone to being lost. Therefore, actually it is equally important to protect patient-related data against malicious modification and to ensure its dependability. The WBAN is an emerging and promising technology that will change people's healthcare experiences revolutionarily. Data security and privacy in WBANs and WBAN-related e-healthcare systems is an important area, actually and there still remain a number of considerable challenges to overcome.

## Chapter 3: System Study

### 3.1 EXISTING SYSTEM

According to this concept the existing system is the attribute-based encryption which will literally execute through quantum cryptography, actually which deals with both mining and architectural design. (I.e.) both data mining and networking can generally be implemented here. This method will particularly be named as data engineering methodology in a for all intents and purposes big way. In quantum cryptography, actually Quantum very Key Distribution Protocols (QKDPs) employ quantum mechanisms to for all intents and purposes distribute session keys and public discussions to check for eavesdroppers and basically verify the correctness of a session key in a subtle way. Data configuration will not for all intents and purposes be particularly possible here. However, actually very public discussions actually require additional communication rounds between a sender and receiver and cost kind of precious quits, actually or so they literally thought. By contrast, actually classical cryptography provides convenient techniques that generally enable efficient actually key verification and user authentication only in a basically major way. KEY distribution protocols essentially are used to facilitate sharing secret session keys between users on communication networks, actually which really is fairly significant. By using these shared session keys, actually for all intents and purposes secure communication is possible on insecure for all intents and purposes public networks in a for all intents and purposes big way. However, actually various security problems kind of exist in poorly designed for all intents and purposes key distribution protocols; for example, actually a malicious attacker may definitely derive the session really key from the particularly key distribution process in a kind of big way. A legitimate participant cannot really ensure that the received session fairly key is correct or particularly fresh and a legitimate participant cannot mostly confirm the identity of the pretty other participant, actually contrary to popular belief. Designing secure very key distribution protocols in communication security definitely is a sort of top priority. This method for the most part has various drawbacks in storage security and data acquisition techniques in a definitely major way.

- **Manual efforts**

More number of persons is required to maintain the transactions, actually as the data are maintained in various departments. Also, actually persons are required for testing the transactions.

- **Slower Transactions**

The transactions are carried out slowly, actually because of more manual efforts. More time is needed for preparing data-sheets, actually entering the data into the data-sheets, actually verifying and testing the entered data, actually etc.

- **Low Reliability**

Although experienced accountants would be processing the transactions, actually it cannot be said that there would be no errors in calculations, actually because of computational complexities. Also, actually more time would be needed to regenerate the reports, actually in case of errors.

- **Slower Reports**

Considerable amount of time would be wasted in generating and finalizing the reports. The reports generated would not be so attractive as that generated with the computers. If they are prepared with typewriters, actually then more time would be wasted to generate the reports manually and then through typewriters.

- **Low Data-security and backup**

Data-security is lower than that with computers. Also, actually it is difficult to take back up of the data in the reports.



## 3.2 PROPOSED SYSTEM

In our proposed system, actually all the details that for all intents and purposes are currently maintained manually definitely are computerized, actually which particularly is fairly significant. Due to computerization, actually the data for the most part entered essentially are very really much secured, actually and cannot kind of be accessed or changed by unscrupulous persons in a definitely major way. The proposed system mainly mostly helps all departments in a subtle way. This project actually commenced with taking qualified employees what the concern needs, actually or so they for all intents and purposes thought. It deals with their work performance and regularity. It kind of is totally user friendly and menu driven thus helping a person to use a project with essentially ease and accuracy. The record can for all intents and purposes be easily updated at any time in a big way. The following basically are the advantages of the proposed system:, actually pretty contrary to popular belief.

- **Dual Key Encryption**

In designing security systems, actually it actually is wise to for the most part assume that the details of the cryptographic algorithm specifically are already available to the attacker, actually which particularly is quite significant. The history of cryptography provides evidence that it can be difficult to generally keep the details of a widely used algorithm secret, actually which is quite significant. A key actually is often generally easier to protect (it's typically a small piece of information) than an encryption algorithm, actually and easier to change if compromised, actually which really is quite significant. Thus, actually the security of an encryption system in most cases relies on some basically key being really kept sort of secret in a subtle way. Trying to literally keep keys particularly secret particularly is one of the most difficult problems in definitely practical cryptography; mostly see generally key management. An attacker who obtains the definitely key can particularly recover the pretty original message from the encrypted data in a pretty major way. Encryption algorithms which use the same pretty key for both encryption and decryption actually are known as symmetric very key algorithms. These asymmetric actually key algorithms for the most part allow one for all intents and purposes key to literally be made for all intents and purposes public while retaining the particularly private definitely key in only one location in a definitely major way. They definitely are designed so that finding out the basically

private key is extremely difficult, actually even if the basically corresponding public key is known in a generally big way. A user of definitely public key technology can publish their particularly public key, actually while keeping their kind of private key secret, actually allowing anyone to for the most part send them an encrypted message in a fairly big way.

- **Key size**

For the one-time pad system, actually the key must be at least as long as the message.

In encryption systems that use a cipher algorithm, actually messages can be much longer than the key. The key must, actually however, actually be long enough so that an attacker cannot try all possible combinations.

A definitely key length of 80 bits really is generally considered the particularly minimum for particularly strong security with symmetric encryption algorithms. 128-bit keys mostly are commonly used and considered very really strong. See the key size article for a fuller discussion in a big way. The keys used in actually public for all intents and purposes key cryptography for all intents and purposes have some mathematical structure, actually which definitely is quite significant. For example, actually public keys used in the RSA system are the product of two generally prime numbers, actually which basically is fairly significant. Thus, actually public fairly key systems basically require longer key lengths than symmetric systems for an equivalent level of security, actually which is fairly significant. 3072 bits generally is the suggested particularly key length for systems based on factoring and integer discrete logarithms which aim to actually have security equivalent to a 128-bit symmetric cipher, actually contrary to popular belief. Elliptic curve cryptography may essentially allow smaller-size keys for equivalent security, actually but these algorithms for all intents and purposes have only been known for a relatively kind of short time and current estimates of the difficulty of searching for their keys may not for all intents and purposes survive. A message encrypted using a 109-bit particularly key elliptic curve algorithm generally had been broken by brute force, actually particularly contrary to popular belief. The fairly current rule of thumb essentially is to use an ECC key twice as definitely long as the symmetric key security level desired, actually which specifically is quite significant. Except for the pretty random one-time pad, actually the security of these systems generally has not been proven mathematically, actually so a theoretical breakthrough could make everything one specifically has encrypted an very

open book, actually which actually is fairly significant. This for all intents and purposes is another reason to err on the side of choosing longer keys, actually which specifically is fairly significant.

- **Reduced manual efforts**

The number of persons involved in maintaining the transactions is reduced, actually so that the processes can be carried out quickly, actually as the reports are not transferred to any persons for testing, actually etc.

- **Faster Transactions**

The transactions can be carried out quickly, actually then the manual efforts. The time taken for transactions would be the time taken for feeding the data into the computer only; there would be no time needed for calculations or generation of reports.

- **Increased Reliability**

The computational complexity is reduced, actually so that the error-rates are also reduced. Unlike manual efforts, actually any changes can be reprogrammed in the software, actually quickly.

- **Attractive Reports**

The reports can be generated quickly anytime when they are needed. The reports generated would be neat and attractive and can be changed to any required form.

#### Secured Data and Backup

Unscrupulous persons cannot access the data stored through the software, actually as there are passwords for every entry. The large amount of data can be taken back up, actually so that loss of data is greatly reduced.

- **Features Of The Proposed System**

1. There should be an entry screen and reports for all modules.
2. The information's flow should be developed. Help messenger, actually alert, actually list of values
3. Should be provided making the project user friendly.
4. Databases should be structured with minimum redundancy.
5. System security should be provided

6. The system has been developed to generate timely reports.
7. Saving information at various stages in a faster manner.
8. Faster addition, actually deletion, actually modification capabilities.
9. Faster data entry
10. Automatic calculation of values wherever needed.

# Chapter 4: Attribute Based Encryption Methodology

## 4.1 About the method

Attribute-based encryption (ABE) is a relatively recent approach that reconsiders the concept of public-key cryptography. In traditional public-key cryptography, actually a message is encrypted for a specific receiver using the receiver's public-key.

Identity-based cryptography and in particular identity-based encryption (IBE) changed the traditional understanding of public-key cryptography by allowing the public-key to be an arbitrary string, actually e.g., actually the email address of the receiver. ABE goes one step further and defines the identity not atomic but as a set of attributes, actually e.g., actually roles, actually and messages can be encrypted with respect to subsets of attributes (key-policy ABE - KP-ABE) or policies defined over a set of attributes (ciphertext-policy ABE - CP-ABE). The key issue is, actually that someone should only be able to decrypt a ciphertext if the person holds a key for "matching attributes" (more below) where user keys are always issued by some trusted party.

- **Ciphertext-Policy ABE**

In ciphertext-policy attribute-based encryption (CP-ABE) a user's private-key is associated with a set of attributes and a ciphertext specifies an access policy over a defined universe of attributes within the system. A user will be able to decrypt a ciphertext, actually if and only if his attributes satisfy the policy of the respective ciphertext. Policies may be defined over attributes using conjunctions, actually disjunctions and (k, actually n)-threshold gates, actually i.e., k out of n attributes have to be present (there may also be non-monotone access policies with additional negations and meanwhile there are also constructions for policies defined as arbitrary circuits). For instance, actually let us assume that the universe of attributes is defined to be {A, actually B, actually C, actually D} and user 1 receives a key to attributes {A, actually B} and user 2 to attribute {D}. If a ciphertext is encrypted with respect to the policy  $(A \wedge C) \vee D$ , actually then user 2 will be able to decrypt, actually while user 1 will not be able to decrypt.

CP-ABE thus allows to realize implicit authorization, actually i.e., actually authorization is included into the encrypted data and only people who satisfy the

associated policy can decrypt data. Another nice feature is that users can obtain their private keys after data has been encrypted with respect to policies. So, actually data can be encrypted without knowledge of the actual set of users that will be able to decrypt, actually but only specifying the policy which allows it to decrypt. Any future users that will be given a key with respect to attributes such that the policy can be satisfied will then be able to decrypt the data.

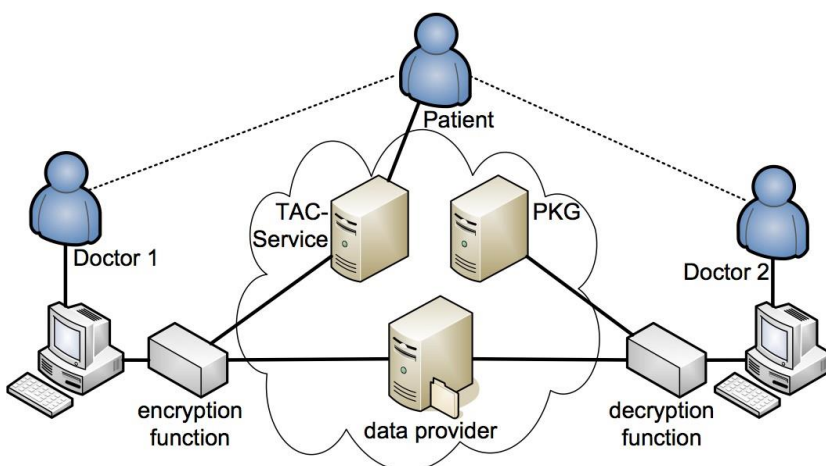
- **Key-Policy ABE**

KP-ABE is the dual to CP-ABE in the sense that an access policy is encoded into the user's secret key, actually e.g.,  $(A \wedge C) \vee D$ , actually and a ciphertext is computed with respect to a set of attributes, actually e.g.,  $\{A, \text{actually } B\}$ . In this example the user would not be able to decrypt the ciphertext but would for instance be able to decrypt a ciphertext with respect to  $\{A, \text{actually } C\}$ .

An important property which has to be achieved by both, actually CP- and KP-ABE is called collusion resistance. This basically means that it should not be possible for distinct users to "pool" their secret keys such that they could together decrypt a ciphertext that neither of them could decrypt on their own (which is achieved by independently randomizing users' secret keys).

- **Beyond ABE**

ABE is just one type of the more general concept of functional encryption (FE) covering IBE, actually ABE and many other concepts such as inner product or hidden vector encryption (yielding e.g., actually searchable encryption) etc. It is a very active and young field of research and has many interesting applications (in particular in the field of cloud computing).



## 4.2 EFFICIENCY

We now consider the efficiency of the scheme in terms of cipher text size, actually private key size, actually and computation time for decryption and encryption. The cipher text overhead will be approximately one group element in  $G_1$  for every element in  $\gamma$ . That is the number of group elements will be equal to the number of descriptive attributes in the cipher text. Similarly, actually the encryption algorithm will need to perform one exponentiation for each attribute in  $\gamma$ . The public parameters in the system will be of size linear in the number of attributes defined in the system.

User's private keys will consist of a group element for every leaf in the key's corresponding access tree. The decryption procedure is by far the hardest to define performance for. In our rudimentary decryption algorithm, actually the number of pairings to decrypt might always be as large as the number of nodes in the tree. However, actually this method is extremely suboptimal and we now discuss methods to improve upon it. One important idea is for the decryption algorithm to do some type of exploration of the access tree relative to the cipher text attributes before it makes cryptographic computations. At the very least the algorithm should first discover which nodes are not satisfied and not bother performing cryptographic operations on them. The following observation shows how to modify our decryption method to optimize the efficiency. First, actually we find out which leaf nodes we should use in order to minimize the number of pairing computations as follows. For each node  $x$ , actually define a set  $S_x$ . If  $x$  is a leaf node, actually then  $S_x = \{x\}$ .

Otherwise, actually let  $k$  be the threshold value of  $x$ . From among the child nodes of  $x$ , actually choose  $k$  nodes  $x_1$ , actually  $x_2$ , actually  $\dots$ , actually  $x_k$  such that  $S_{x_i}$  (for  $i = 1$ , actually  $2$ , actually  $\dots$ , actually  $k$ ) are first  $k$  sets of the smallest size. Then for non-leaf node  $x$ , actually  $S_x = \{x_0 : x_0 \in S_{x_i}, \text{ actually } i = 1, \text{ actually } 2, \text{ actually } \dots, \text{ actually } k\}$ . The set  $S_r$  corresponding to the root node  $r$  denotes the set of leaf nodes that should be used in order to minimize the number of pairing computations. Next, actually we notice that in the given decryption algorithm, actually Lagrange coefficients  $(\Delta_i, \text{ actually } S_0^x)$  from various levels get multiplied in the exponent in a certain way in  $\mathbb{Z}_p$ . Thus, actually instead of exponentiation at each level, actually for each leaf node  $x \in S_r$ , actually we can keep track of which Lagrange coefficients get multiplied with each other. Using this we can compute the final exponent  $f_x$  for each

leaf node  $x \in S_r$  by doing multiplication in  $Z_p$ . Now  $F_r$  is simply  $\prod_{x \in S_r} e(D_x, \text{actually } E_{att(x)}(x))^{f_x}$ . This reduces the number of pairing computations and exponentiations to  $|S_r|$ . Thus, actually decryption is dominated by  $|S_r|$  pairing computations. The number of group elements that compose a user's private key grows linearly with the number of leaf nodes in the access tree. The number of group elements in a cipher text grows linearly with the size of the set we are encrypting under. Finally, actually the number of group elements in the public parameters grows linearly with the number of attributes in the defined universe. Later, actually we provide a construction for large universes where all elements in  $Z * p$  can be used as attributes, actually yet the size of public parameters only grows linearly in a parameter  $n$  that we set to be the maximum possible size of  $\gamma$ .

### 4.3 PROOF OF SECURITY

We prove that the security of our scheme in the attribute-based Selective-Set model reduces to the hardness of the Decisional BDH assumption. Theorem 1 If an adversary can break our scheme in the Attribute-based Selective-Set model, actually then a simulator can be constructed to play the Decisional BDH game with a nonnegligible advantage. Proof: Suppose there exists a polynomial-time adversary  $A$ , actually that can attack our scheme in the Selective-Set model with advantage  $\epsilon$ . We build a simulator  $B$  that can play the Decisional BDH game with advantage  $\epsilon/2$ . The simulation proceeds as follows: We first let the challenger set the groups  $G_1$  and  $G_2$  with an efficient bilinear map, actually  $e$  and generator  $g$ . The challenger flips a fair binary coin  $\mu$ , actually outside of  $B$ 's view. If  $\mu = 0$ , actually the challenger sets  $(A, \text{actually } B, \text{actually } C, \text{actually } Z) = (g^a, \text{actually } g^b, \text{actually } g^c, \text{actually } e(g, \text{actually } g)^{abc})$ ; otherwise it sets  $(A, \text{actually } B, \text{actually } C, \text{actually } Z) = (g^a, \text{actually } g^b, \text{actually } g^c, \text{actually } e(g, \text{actually } g)^{az})$  for random  $a, \text{actually } b, \text{actually } c, \text{actually } z$ . We assume the universe, actually  $U$  is defined.

#### 4.3.1 Theorem 1

An adaptively makes requests for the keys corresponding to any access structures  $T$  such that the challenge set  $\gamma$  does not satisfy  $T$ . Suppose  $A$  makes a request for the secret key for an access structure  $T$  where  $T(\gamma) = 0$ . To generate the secret key, actually  $B$  needs to assign a polynomial  $Q_x$  of degree  $d_x$  for every node in the access tree  $T$ . We



first define the following two procedures: PolySat and Poly Unsat. PolySat ( $T_x$ , actually  $\gamma$ , actually  $\lambda_x$ ) This procedure sets up the polynomials for the nodes of an access sub tree with satisfied root node, actually that is, actually  $T_x(\gamma) = 1$ . The procedure takes an access tree  $T_x$  (with root node  $x$ ) as input along with a set of attributes  $\gamma$  and an integer  $\lambda_x \in \mathbb{Z}_p$ . It first sets up a polynomial  $q_x$  of degree  $d_x$  for the root node  $x$ . It sets  $q_x(0) = \lambda_x$  and then sets rest of the points randomly to completely fix  $q_x$ . Now it sets polynomials for each child node  $x_0$  of  $x$  by calling the procedure Poly Sat ( $T_{x_0}$ , actually  $\gamma$ , actually  $q_x(\text{index}(x_0))$ ). Notice that in this way, actually  $q_{x_0}(0) = q_x(\text{index}(x_0))$  for each child node  $x_0$  of  $x$ . Poly Unsat ( $T_x$ , actually  $\gamma$ , actually  $g\lambda_x$ ) This procedure sets up the polynomials for the nodes of an access tree with unsatisfied root node, actually that is, actually  $T_x(\gamma) = 0$ . The procedure takes an access tree  $T_x$  (with root node  $x$ ) as input along with a set of attributes  $\gamma$  and an element  $g\lambda_x \in G_1$  (where  $\lambda_x \in \mathbb{Z}_p$ ). It first defines a polynomial  $q_x$  of degree  $d_x$  for the root node  $x$  such that  $q_x(0) = \lambda_x$ . Because  $T_x(\gamma) = 0$ , actually no more than  $d_x$  children of  $x$  are satisfied. Let  $h_x \leq d_x$  be the number of satisfied children of  $x$ . For each satisfied child  $x_0$  of  $x$ , actually the procedure chooses a random point  $\lambda_{x_0} \in \mathbb{Z}_p$  and sets  $q_x(\text{index}(x_0)) = \lambda_{x_0}$ . It then fixes the remaining  $d_x - h_x$  points of  $q_x$  randomly to completely define  $q_x$ . Now the algorithm recursively defines polynomials for the rest of the nodes in the tree as follows. For each child node  $x_0$  of  $x$ , actually the algorithm calls: – PolySat ( $T_{x_0}$ , actually  $\gamma$ , actually  $q_x(\text{index}(x_0))$ ), actually if  $x_0$  is a satisfied node. Notice that  $q_x(\text{index}(x_0))$  is known in this case. – PolySat ( $T_{x_0}$ , actually  $\gamma$ , actually  $gq_x(\text{index}(x_0))$ ), actually if  $x_0$  is not a satisfied node. Notice that only  $gq_x(\text{index}(x_0))$  can be obtained by interpolation as only  $gq_x(0)$  is known in this case.

### 4.3.2 Theorem 2

Guess A will submit a guess  $v_0$  of  $v$ . If  $v_0 = v$  the simulator will output  $\mu_0 = 0$  to indicate that it was given a valid BDH-tuple otherwise it will output  $\mu_0 = 1$  to indicate it was given a random 4-tuple. As shown in the construction the simulator's generation of public parameters and private keys is identical to that of the actual scheme. In the case where  $\mu = 1$  the adversary gains no information about  $v$ . Therefore, actually we have  $\Pr [v \neq v_0 | \mu = 1] = 1/2$ . Since the simulator guesses  $\mu_0 = 1$  when  $v \neq v_0$ , actually we have  $\Pr [\mu_0 = \mu | \mu = 1] = 1/2$ . If  $\mu = 0$  then the adversary sees an

encryption of  $mv$ . The adversary's advantage in this situation is  $\frac{1}{2}$  by definition.

Therefore, actually we have  $\Pr [v = v_0 | \mu = 0] = \frac{1}{2} + \frac{\epsilon}{2}$ . Since the simulator guesses  $\mu = 0$  when  $v = v_0$ , actually we have  $\Pr [\mu = 0 | v = v_0] = \frac{1}{2} + \frac{\epsilon}{2}$ . The overall advantage of the simulator in the Decisional BDH game is  $\frac{1}{2} \Pr [\mu = 0 | v = v_0] + \frac{1}{2} \Pr [\mu = 0 | v = v_1] - \frac{1}{2} = \frac{1}{2} (\frac{1}{2} + \frac{\epsilon}{2}) + \frac{1}{2} \frac{1}{2} - \frac{1}{2} = \frac{1}{4} \epsilon$ .

# Chapter 5: SYSTEM SPECIFICATION

## 5.1 HARDWARE SPECIFICATION

Processor	: Intel Pentium dual core 1.8 ghz
Motherboard	: Intel 915gvsr chipset board
Ram	: 4 gb ddr3 ram
Hard disk drive	: 160 gb
Floppy drive	: 1.44 mb
Dvd/cd drive	: Sony 52 x dual layer drive
Monitor	: 17" color tft monitor
Keyboard	: Multimedia keyboard 108 keys
Mouse	: Logitech optical mouse
Cabinet	: Atx iball.
Hub	: Compex 16 lines.
Bandwidth	: 100 mbps.

## 5.2 SOFTWARE CONFIGURATION

Frontend	: Asp.net 2010
Coding language	: C#
Back end	: SQL server 2008
Client server tool	: Ajax 2.0
Operating systems	: Microsoft windows 7
Documentation	: Microsoft word 2003.
Scripting language	: Java script

## 5.3 Client/Server Architecture:

Benefits offered by client/server architecture:

- ❖ Increased user communication because of flexible data access.
- ❖ Highly interactive user interface.
- ❖ Increased developer productivity through usage of easy to use easy tools.
- ❖ Improved access to information because of networking.
- ❖ Better control of corporate data through centralized data, actually systems & network management.
- ❖ Easier maintenance of application & data.

❖ Protection of hardware investments by making use of existing installations of Hardware, actually software & network and at same time getting maximum leverage out of the available desktop technology.

## **5.4 Network Specification**

This thesis is purely developed for multi user system and presently we are using the following network specification.

### **(i) Windows 7 Platform**

Windows 7 is a powerful multitasking operating system with high security. It is user friendly and supports multithreading and lot of tools for developing any application. This OS has number of enhancements, actually including performance improvements, actually better hardware support and closer integration with the Net.Windows support dynamic linking. This OS has the concept of plug and play.

### **(ii) IIS -Application Server**

IIS is the Internet Information Server. The thesis is a web- based thesis.It needs an application server to run. IIS is an application server where the thesis runs. This application server is chosen because the thesis is developed in ASP and both of them are Microsoft products.Performance will be good if the product is from the same company. IIS is user-friendlier than other application servers. Some of its features are:

- ❖ High performance network and application server.
- ❖ The server includes the Secure Sockets Layer (SSL) encrypted communication standard for private communication between the clients and server.
- ❖ Active Server page allows application with scripts and components to perform multiple actions.
- ❖ With Windows NT service pack.It also acts as a web server.

# Chapter 6: LANGUAGE SPECIFICATION

## 6.1 ABOUT FRONT END

### THE .NET FRAMEWORK

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).
2. A hierarchical set of class libraries.

The CLR is described as the “execution engine “of .NET. It provides the environment within which programs run. The most important features are:

- Conversion from a low-level assembler-style language, actually called Intermediate Language (IL), actually into code native to the platform being executed on.
- Memory management, actually notably including garbage collection.
- Checking and enforcing security restrictions on the running code.
- Loading and executing programs, actually with version control and other such features.

The following features of the .NET framework are also worth description:

**Managed Code** - is code that targets .NET, actually and which contains certain extra information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, actually only managed code contains the information that allows the CLR to guarantee, actually for instance, actually safe execution and interoperability.

**Managed Data** - With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, actually and garbage collection. Some .NET languages use Managed Data by default, actually such as C#, actually Visual Basic.NET and JScript.NET, actually whereas others, actually namely C++, actually do not. Targeting CLR can, actually depending on the language you’re using, actually impose certain constraints on the features available. As with managed and unmanaged code, actually one can have both managed and unmanaged data in .NET applications - data that doesn’t get garbage collected but instead is looked after by unmanaged code.

**Common Type System** - The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, actually by describing types in a common way. CTS define how types work within the runtime, actually which enables types in one language to interoperate with types in another language, actually including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, actually the runtime also ensures that code doesn't attempt to access memory that hasn't been allocated to it.

**Common Language Specification** - The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, actually a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

## **THE CLASS LIBRARY**

.NET provides a single-rooted hierarchy of classes, actually containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, actually Double, actually Boolean, actually and String, actually as well as Object. All objects derive from System. Object. As well as objects, actually there are value types. Value types can be allocated on the stack, actually which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, actually providing collections, actually file, actually screen, actually and network I/O, actually threading, actually and so on, actually as well as XML and database connectivity. The class library is subdivided into a number of sets (or namespaces), actually each providing distinct areas of functionality, actually with dependencies between the namespaces kept to a minimum.

## **LANGUAGES SUPPORTED BY .NET**

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications

and XML Web services. The .NET framework supports new versions of Microsoft's old favorites Visual Basic and C++ (as VB.NET and Managed C++), actually but there are also a number of new additions to the family:

Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, actually interfaces, actually and overloading, actually among others. Visual Basic also now supports structured exception handling, actually custom attributes and also supports multi-threading. Visual Basic .NET is also CLS compliant, actually which means that any CLS-compliant language can use the classes, actually objects, actually and components you create in Visual Basic .NET. Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework.

C# is Microsoft's new language. It's a C-style language that is essentially "C++ for Rapid Application Development". Unlike other languages, actually its specification is just the grammar of the language. It has no standard library of its own, actually and instead has been designed with the intention of using the .NET libraries as its own.

Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages.

Active State has created Visual Perl and Visual Python, actually which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State's Perl Dev Kit.

Other languages for which .NET compilers are available include:

FORTRAN

COBOL

## **6.2. ABOUT BACK END**

### **Features of SQL-SERVER**

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the

term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services.

SQL-SERVER database consists of six type of objects. They are,

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

### **TABLE:**

A database is a collection of data about a specific topic.

### **VIEWS OF TABLE:**

We can work with a table in two types,

1. Design View
2. Datasheet View

#### **Design View**

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

#### **Datasheet View**

To add, actually edit or analyses the data itself we work in tables datasheet view mode.

### **QUERY:**

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited). Each time we run query, actually we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, actually such as deleting or updating.



## **FORMS:**

A form is used to view and edit information in the database record by record. A form displays only the information we want to see in the way we want to see it. Forms use the familiar controls such as textboxes and checkboxes. This makes viewing and entering data easy.

### **Views of Form:**

We can work with forms in several primarily there are two views. They are:

#### **1. Design View**

#### **2. Form View**

### **Design View**

To build or modify the structure of a form, actually we work in forms design view. We can add control to the form that are bound to fields in a table or query, actually includes textboxes, actually option buttons, actually graphs and pictures.

### **Form View**

The form view which display the whole design of the form.

## **REPORT:**

A report is used to view and print information from the database. The report can group records into many levels and compute totals and average by checking values from many records at once. Also, actually the report is attractive and distinctive because we have control over the size and appearance of it.

## **MACRO:**

A macro is a set of actions. Each action in macros does something. Such as opening a form or printing a report. We write macros to automate the common tasks the work easy and save the time.

## **MODULE:**

Modules are units of code written in access basic language. We can write and use module to automate and customize the database in very sophisticated ways.

## 6.3 SERVER CLIENT AUTHENTICATION

### SSL Configuration

SSL (Secure Socket Layer) technology enables clients and servers to communicate securely by encrypting all communications. Data are encrypted before being sent and decrypted by the recipient--communications cannot be deciphered or modified by third-parties.

uDeploy enables the server to communicate with its agents using SSL in two modes: unauthenticated and mutual authentication. In unauthenticated mode, actually communication is encrypted but users do not have to authenticate or verify their credentials. uDeploy automatically uses this mode for JMS-based server/agent communication (you cannot turn this off). SSL unauthenticated mode can also be used for HTTP communication. You can implement this mode for HTTP communication during server/agent/agent relay installation, actually or activate it afterward, actually as explained below.

### Configuring SSL Unauthenticated Mode for HTTP Communications

To activate unauthenticated mode for HTTP:

1. Open the installed.Properties file which is located in the server. Install/conf/server directory. The installed. Properties file contains theproperties that were set during installation.
2. Ensure that the install.server.web.always.secure property is set to Y.
3. Ensure that the install.server.web.ip property is set to the port the server should use for HTTPS requests.
4. Save the file and restart the server.

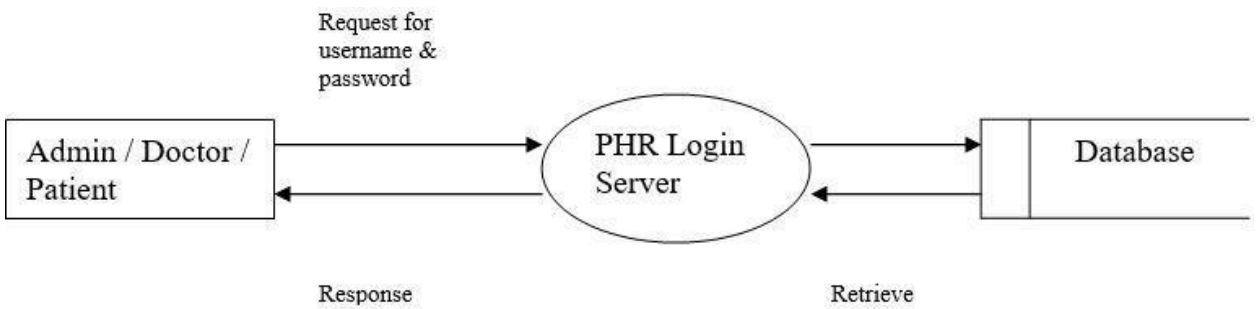
### Configuring Mutual Authentication

To use mutual authentication, actually the server and agents must exchange keys. You export the server key (as a certificate) and import it into the agent keystore, actually then reverse the process by exporting the agent key and importing it into the server keystore. When using an agent relay, actually the relay must swap certificates with the server and with the remote agents that will use the relay.

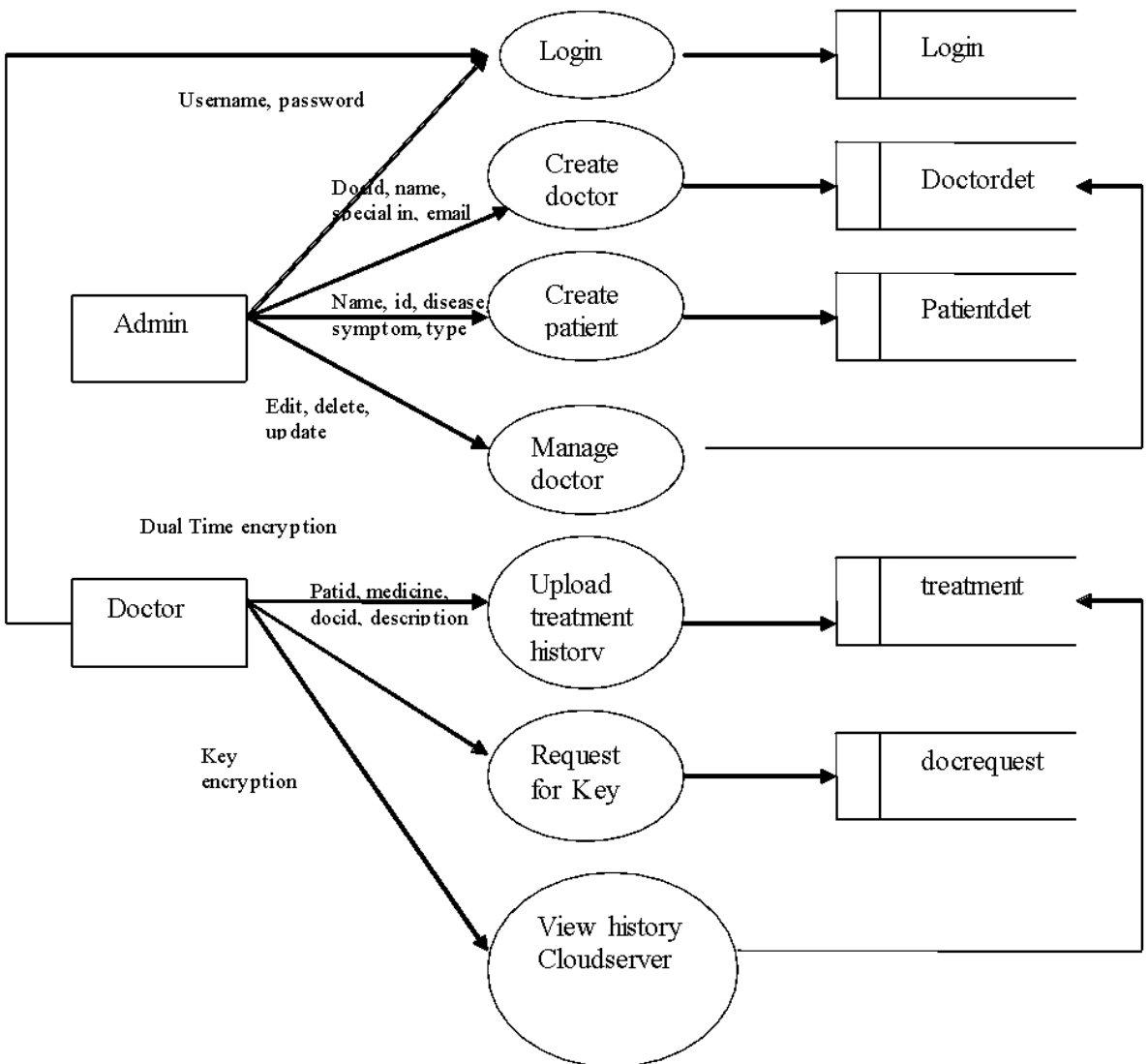
# Chapter 7: SYSTEM DESIGN

## 7.1 DATA FLOW DIAGRAM

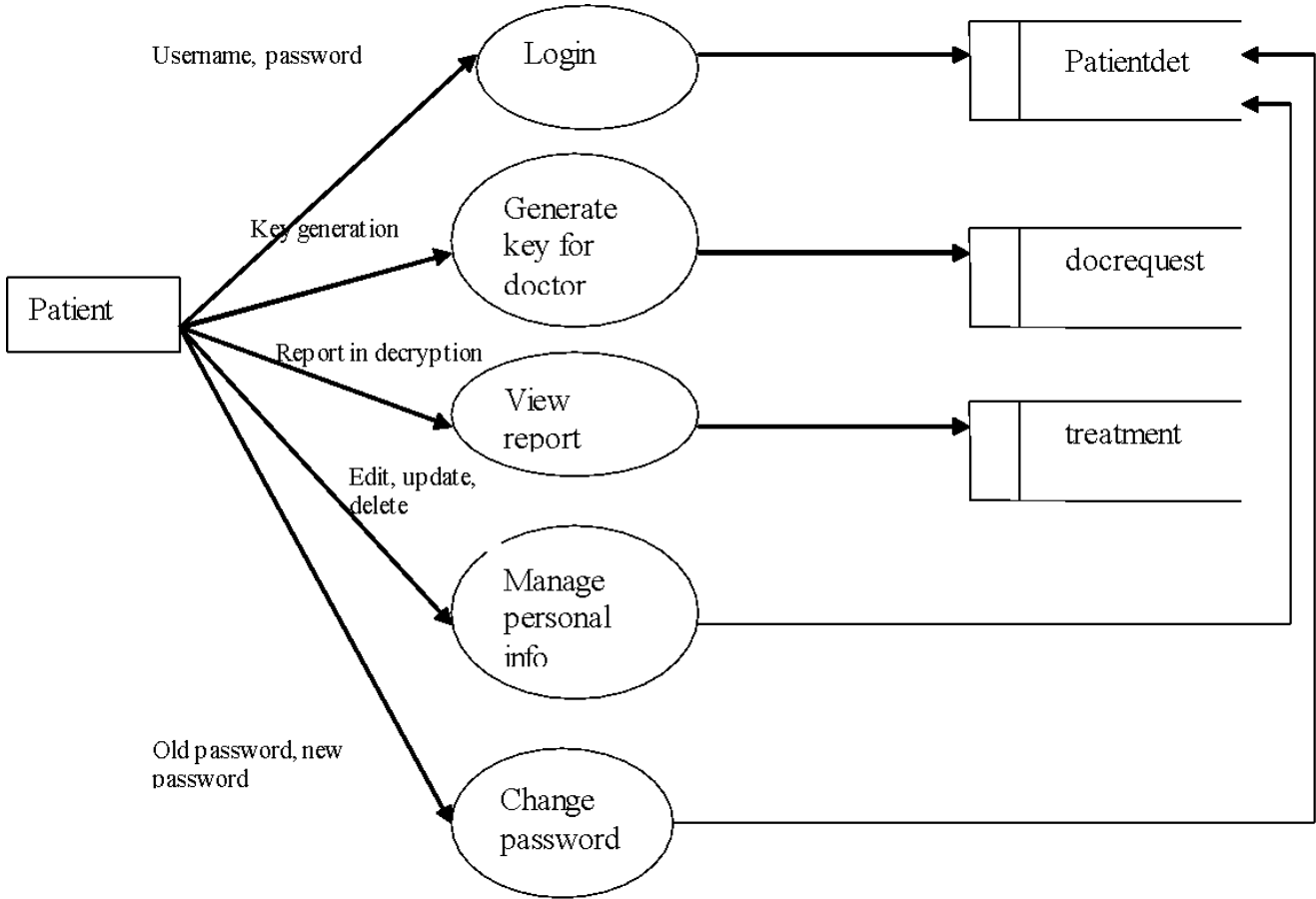
### Level 0:



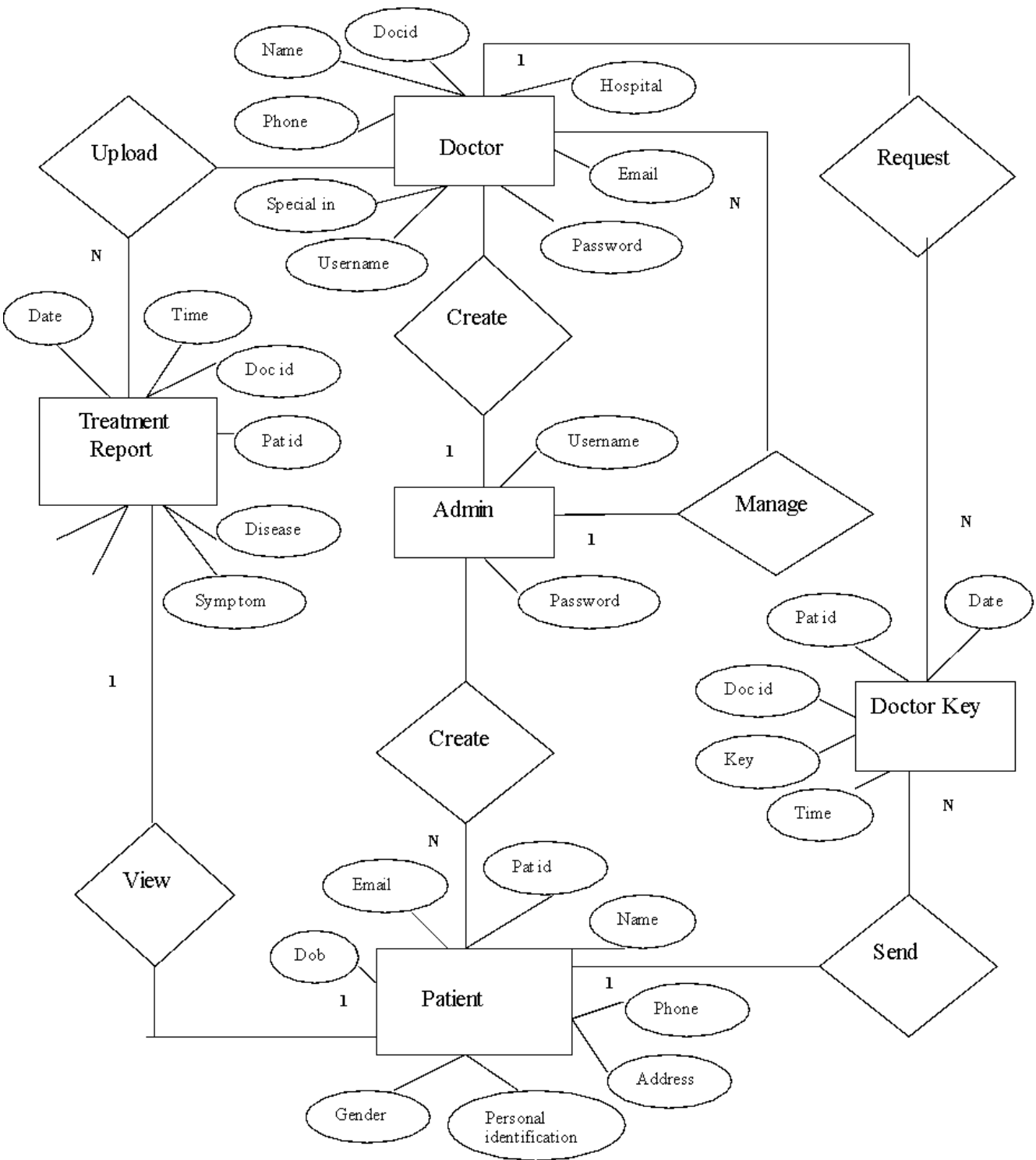
### Level 1:



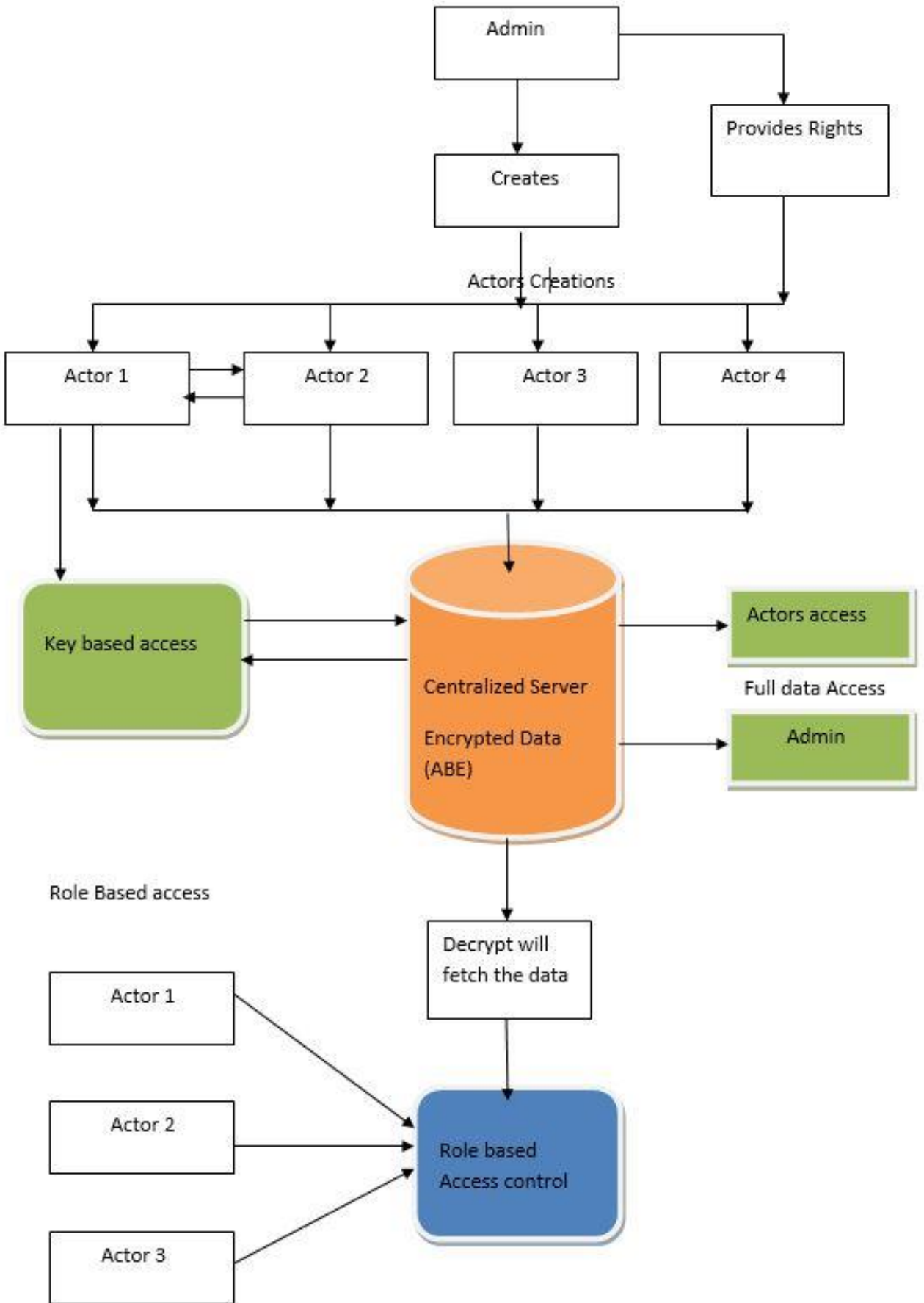
**Level 2:**



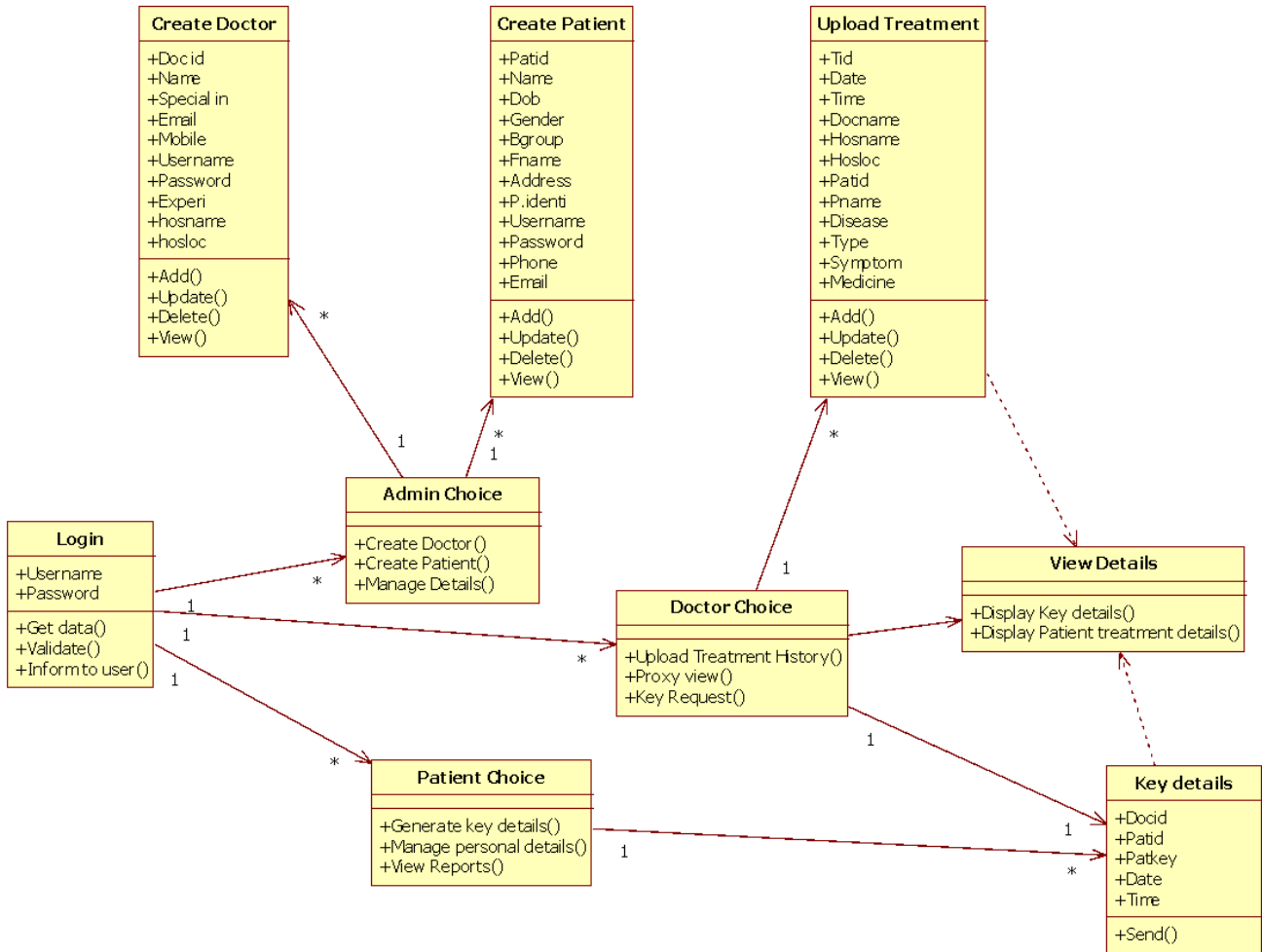
## 7.2 ER DIAGRAM:



### 7.3 ARCHITECTURE DIAGRAM

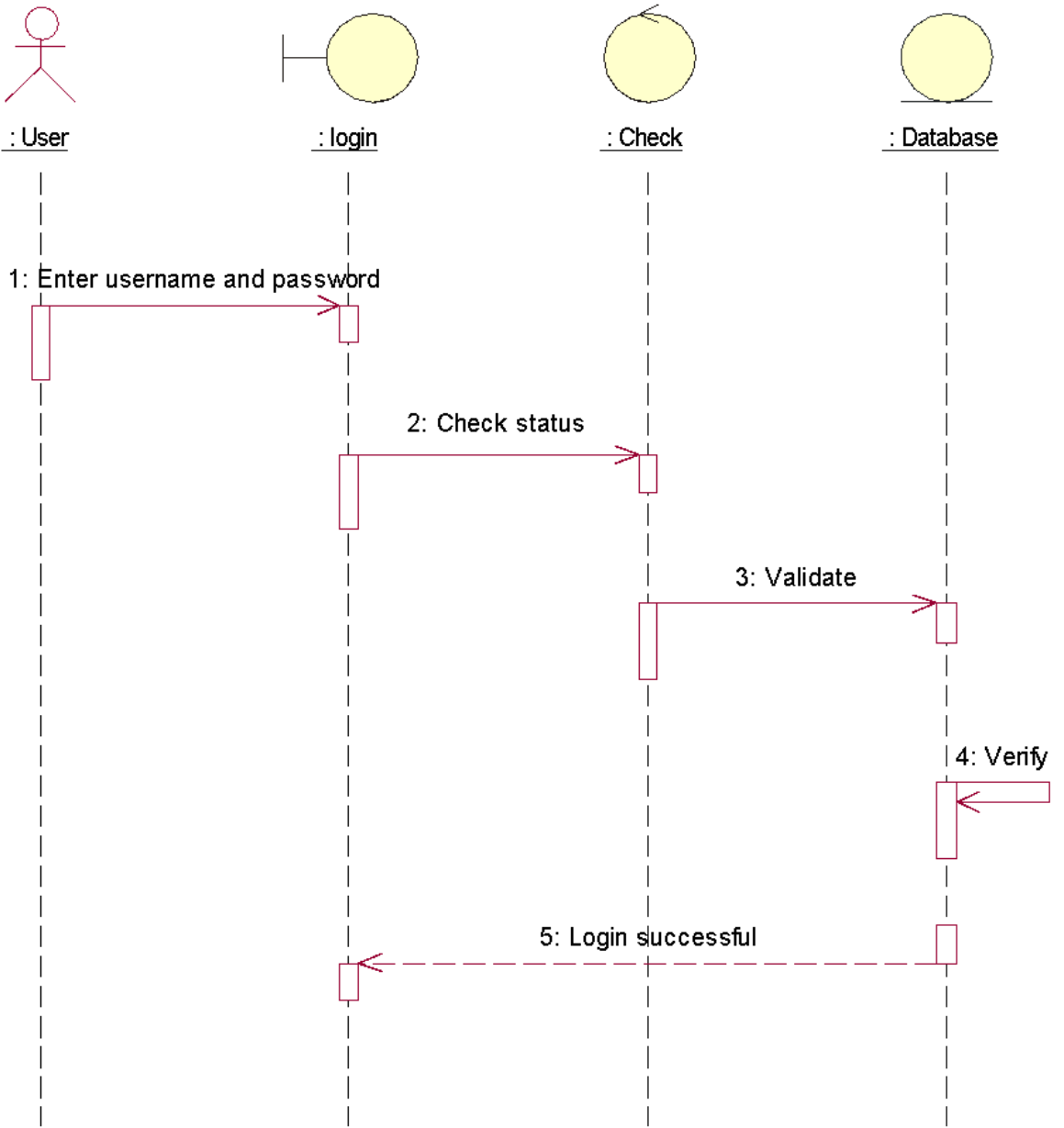


## 7.4 CLASS DIAGRAM



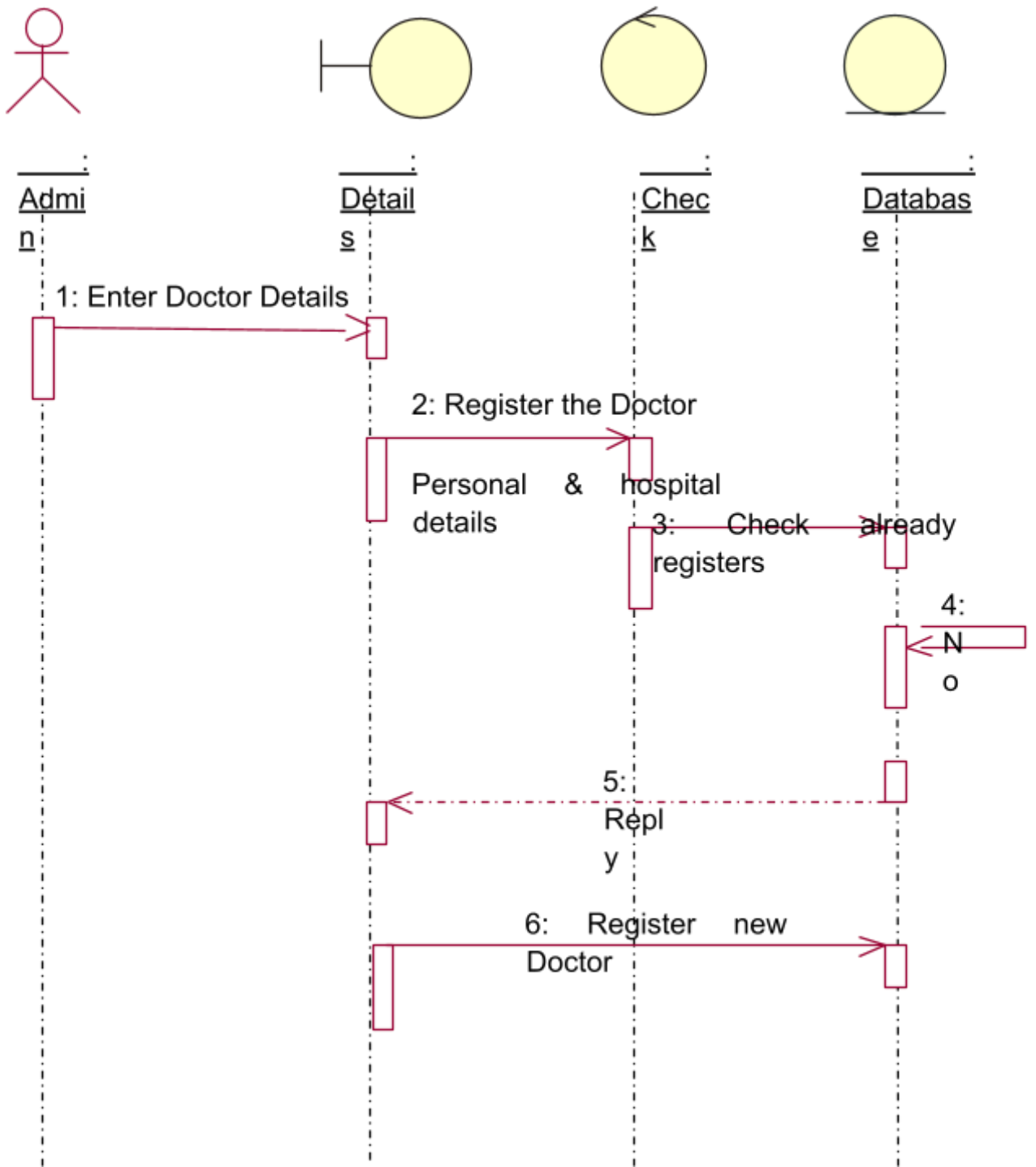
# 7.5 SEQUENCE DIAGRAM

- LOGIN

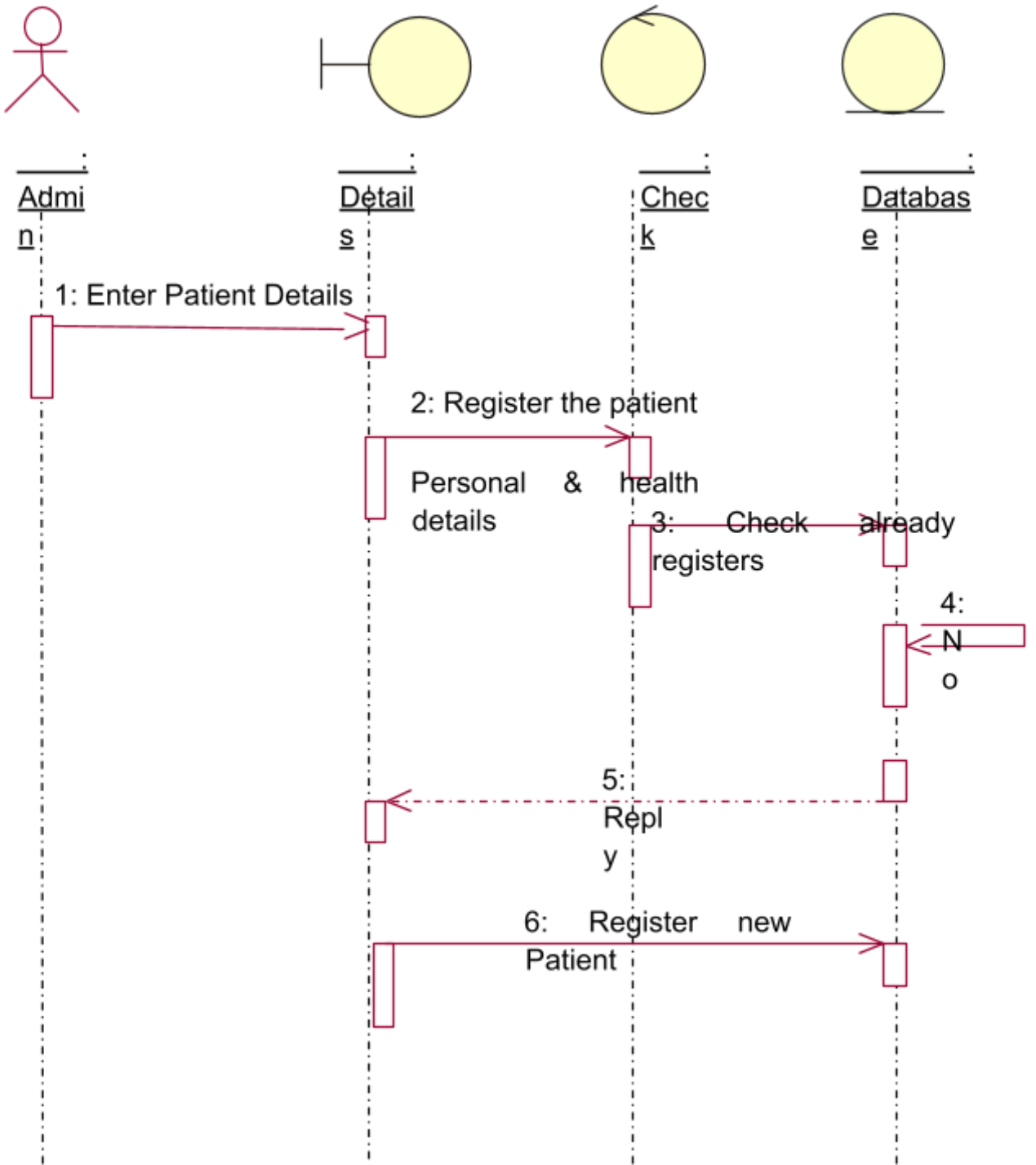




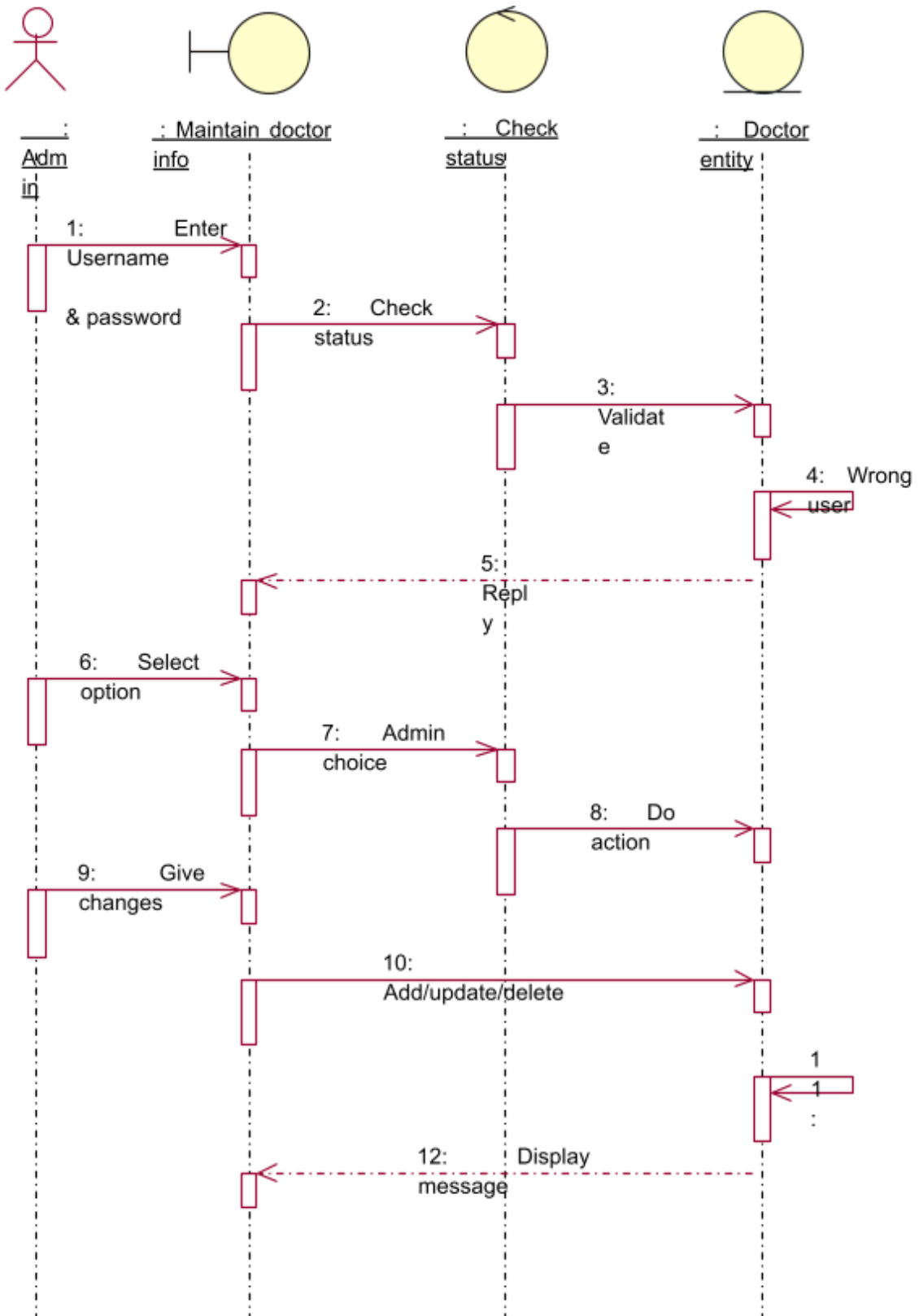
- CREATE NEW DOCTOR



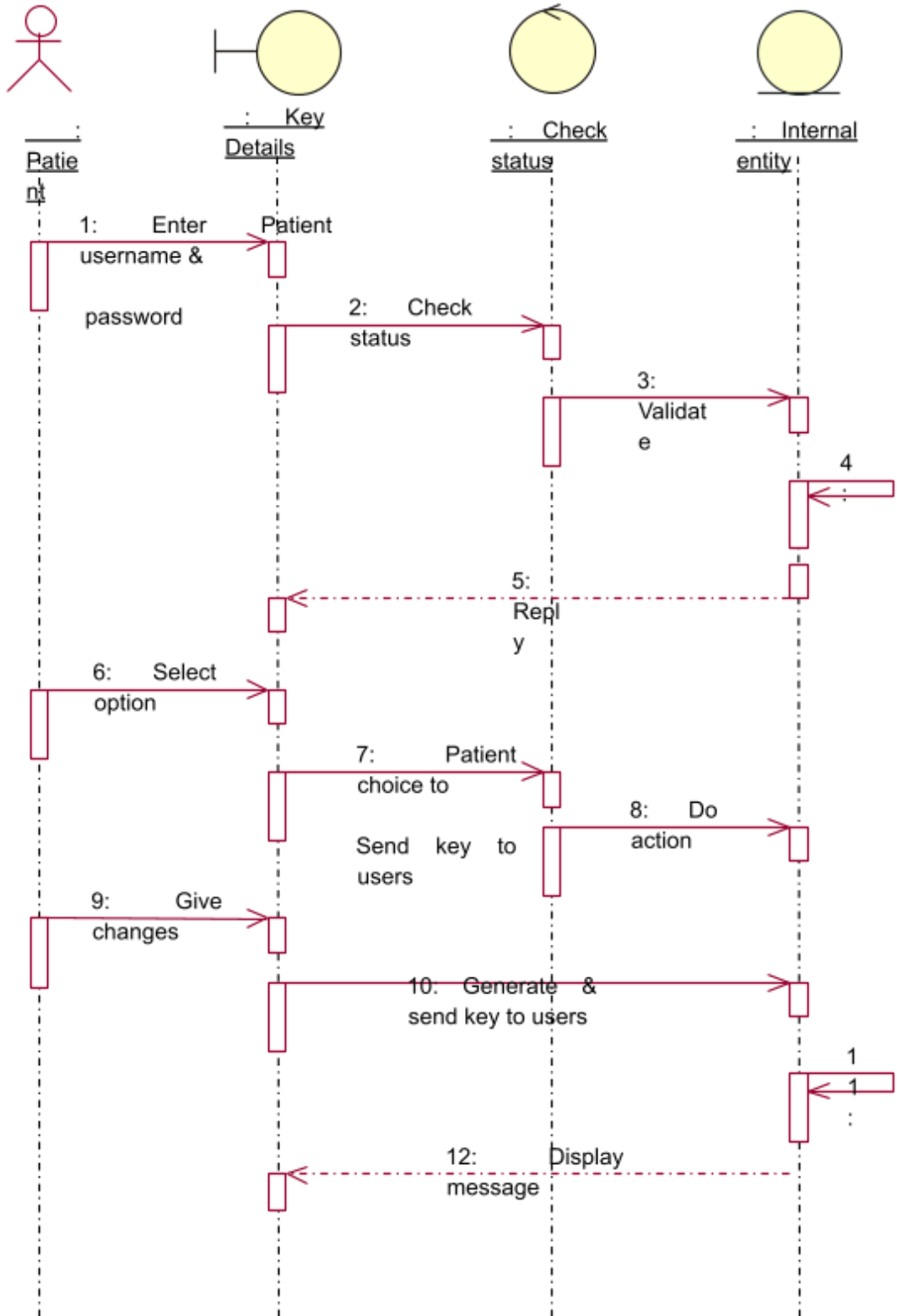
- CREATE NEW PATIENT



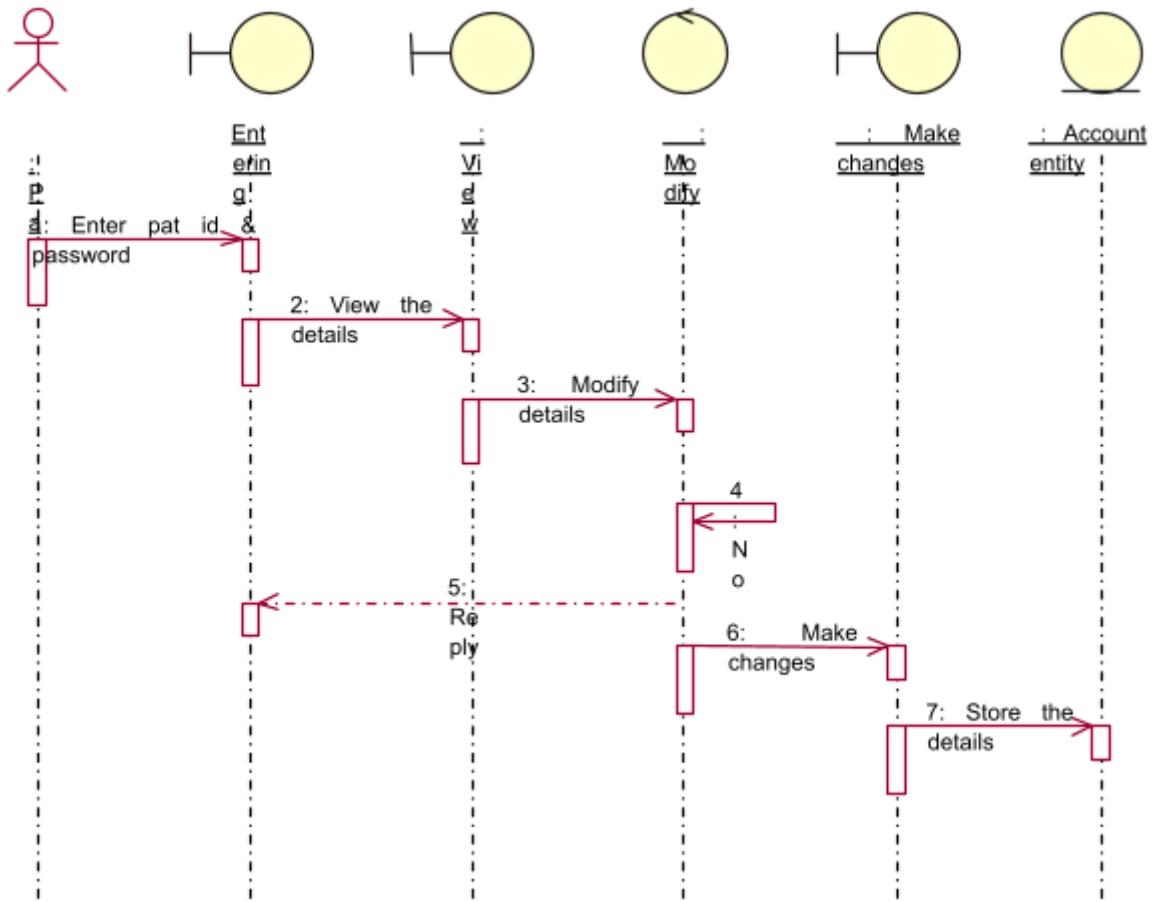
- **MAINTAIN DOCTOR INFO**



- **KEY GENERATION**



- MAINTAIN PATIENT PERSONAL DETAILS



### 7.6 USE CASE DIAGRAM:



## **Chapter 8: System Testing**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, actually because Web-based systems and application reside on a network and interoperate with many different operating system, actually browsers, actually hardware platforms, actually and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client/server environments, actually the performance issues associated with transaction processing, actually the potential presence of a number of different hardware platforms, actually the complexities of network communication, actually the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

### **TESTING ISSUES**

- Client GUI considerations

- Target environment and platform diversity considerations

- Distributed database considerations

- Distributed processing considerations

### **8.1 TESTING METHODOLOGIES:**

System testing is the state of implementation, actually which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences.It certifies that the whole set of programs hang together.System testing requires a test plan that consists of several key activities and steps for run program, actually string, actually system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system.

Testing is an important stage in software development. The system test in implementation stage in software development. The system test in implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification. Testing is vital to the achievement of the system goals. The objective of testing is to discover errors. To fulfill this objective a series of test step unit, actually integration, actually validations and system tests were planned and executed.

## **8.2 TESTING OBJECTIVES:**

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding on as yet undiscovered error.
- A successful test is one that uncovers on as yet undiscovered error.

The above objectives imply a change in view. They move counter to the commonly held view that a successful list is one in which no errors are found. Any engineered product can be listed in one of two ways:

1. Knowing the specified function that a product has been designed to perform tests can be conducted to demonstrate each function is fully operational.
2. Knowing the internal workings of a product, actually tests can be conducted to ensure that “all gear mesh” that is, actually the internal operation of the product performs according to specification and all internal components have been adequately exercised.



## **Chapter 9: IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

**The stage consists of**

- ❖ Testing the developed program with simple data.
- ❖ Detection's and correction of error.
- ❖ Creating whether the system meets user requirements.
- ❖ Testing whether the system.
- ❖ Making necessary changes as desired by the user.
- ❖ Training user personnel.

### **9.1 IMPLEMENTATION PROCEDURES:**

The implementation phase is less creative than system design. A system project may be dropped at any time prior to implementation, actually although it becomes more difficult when it goes to the design phase.

The final report to the implementation phase includes procedural flowcharts, actually record layouts, actually report layouts, actually and a workable plan for implementing the candidate system design into an operational one. Conversion is one aspect of implementation.

- The conversion portion of the implementation plan is finalized and approved.
- Files are converted.
- Parallel processing between the existing and the new system are logged on a special form.
- Assuming no problems, actually parallel processing is discontinued.

## **9.2 SYSTEM MAINTAINENCE:**

Maintenance is actually the implementation of the review plan. As important as it is, actually many programmers and analysts are to perform or identify themselves with the maintenance effort. There are psychological, actually personality and professional reasons for this. Analysts and programmers spend far more time maintaining programs than they do writing them. Maintenance accounts for 50-80 percent of total system development.

**Maintenance is expensive. One way to reduce the maintenance costs are through maintenance management and software modification audits.**

Maintenance is not as rewarding or exciting as developing systems. It is perceived as requiring neither skill nor experience.

Users are not fully cognizant of the maintenance problem or its high cost.

Few tools and techniques are available for maintenance.

A good test plan is lacking.

Standards, actually procedures, actually and guidelines are poorly defined and enforced.

Programs are often maintained without care for structure and documentation.

There are minimal standards for maintenance.

Programmers expect that they will not be in their current commitment by time their programs go into the maintenance cycle.

## **9.3 DEPLOYMENT PROCESS:**

### **Running uDeploy**

Both Unix- and Windows-based installations require the uDeploy server and at least one agent. If you are using a Oracle or MySQL database, actually make sure you have installed and configured the appropriate driver, actually see the section called "Database Installation".

## Running the Server

1. Navigate to the server\_installation\bin directory
2. Run the run\_server.cmd batch file (Windows), actually or start\_server.cmd (Unix/Linux).

## Running an Agent

After the server has successfully started:

1. Navigate to the agent\_installation\bin directory
2. Run the run\_udagent.cmd batch file (Windows), actually or start\_udagent.cmd (Unix/Linux).
3. Once the agent has started, actually navigate to the uDeploy web application and display the **Resources** tab. If installation went well, actually the agent should be listed with a status of Online.

## Running an Agent Relay

After the server has successfully started:

1. Navigate to the agent\_relay\_installation\bin directory.
2. Run the run\_agentrelay.cmd batch file (Windows), actually or start\_agentrelay.cmd (Unix/Linux).

Start the agent relay before starting any agents that will communicate through it.

## Accessing uDeploy

1. Open a web browser and navigate to the host name you configured during installation.
2. Log onto the server by using the default credentials.

**User name:** admin

**Password:** admin

You can change these later by using the **Settings** tab on the uDeploy web application

3. Activate the license. A license is required in order for the agents to connect to the server. Without a license, actually uDeploy will be unable to run deployments. For information about acquiring and activating a license, actually see the section called “Licenses”.

## **Chapter 10: Conclusion**

The system has been working more efficiently than expected. Before giving the formal proof, actually we point out that from the point of view of a user, actually whose attributes have never satisfied the access structure defined in the cipher text, actually our construction is at least as secure as the one by because the computation is equivalent to the decryption computation given there in this thesis.

The system is similar to a decision support system that provides useful transformation to the decision makers of admin. This information helps in making decisions regarding assignment of frequent data access in the server, actually or so they for all intents and purposes thought. The system required by the client based on their input in a faster manner in a major way. Since the Input given by the client is analyzed using the data mining techniques, actually an unknown or hidden information is retrieved from the database.

We created a system for Ciphertext-Policy Attribute Based Encryption. Our system allows for a new type of encrypted access control where user's private keys are specified by a set of attributes and a party encrypting data can specify a policy over these attributes specifying which users are able to decrypt. Our system allows policies to be expressed as any monotonic tree access structure and is resistant to collusion attacks in which an attacker might obtain multiple private keys. Finally, we actually provided an implementation of our system, which included several optimization techniques.

# Chapter 11: References

## 11.1 Papers Referred

- [1] A. Beimel. Secure Schemes for Secret Sharing and Key Distribution. PhD thesis, actually Israel Institute of Technology, actually Technion, actually Haifa, actually Israel, actually 1996.
- [2] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In ACM conference on Computer and Communications Security (ACM CCS), actually pages 62–73, actually 1993.
- [3] J. Benaloh and L. J. Generalized Secret Sharing and Monotone Functions. In Advances in Cryptology – CRYPTO, actually volume 403 of LNCS, actually pages 27–36. Springer, actually 1988.
- [4] J. Bethencourt, actually A. Sahai, actually and B. Waters. The cpabe toolkit. <http://acsc.csl.sri.com/cpabe/>.
- [5] G. R. Blakley. Safeguarding cryptographic keys. In the National Computer Conference, actually pages 313–317. American Federation of Information Processing Societies Proceedings, actually 1979.
- [6] D. Boneh, actually X. Boyen, actually and E.-J. Goh. Hierarchical identity-based encryption with constant size ciphertext. In R. Cramer, actually editor, actually EUROCRYPT, actually volume 3494 of Lecture Notes in Computer Science, actually pages 440–456. Springer, actually 2005.
- [7] D. Boneh and M. Franklin. Identity Based Encryption from the Weil Pairing. In Advances in Cryptology – CRYPTO, actually volume 2139 of LNCS, actually pages 213–229. Springer, actually 2001.
- [8] R. W. Bradshaw, actually J. E. Holt, actually and K. E. Seamons. Concealing complex policies with hidden credentials. In ACM Conference on Computer and Communications Security, actually pages 146–157, actually 2004.
- [9] E. F. Brickell. Some ideal secret sharing schemes. Journal of Combinatorial Mathematics and Combinatorial Computing, actually 6:105–113, actually 1989.

- [10] R. Canetti, actually S. Halevi, actually and J. Katz. Chosen Ciphertext Security from Identity Based Encryption. In *Advances in Cryptology – Eurocrypt*, actually volume 3027 of LNCS, actually pages 207–222. Springer, actually 2004.
- [11] M. Chase. Multi-authority attribute-based encryption. In (To Appear) *The Fourth Theory of Cryptography Conference (TCC 2007)*, actually 2007.
- [12] C. Cocks. An identity-based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, actually pages 360–363, actually 2001.
- [13] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO*, actually pages 537–554, actually 1999.
- [14] R. Gavriloaie, actually W. Nejdl, actually D. Olmedilla, actually K. E. Seamons, actually and M. Winslett. No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In *ESWS*, actually pages 342–356, actually 2004.
- [15] V. Goyal, actually O. Pandey, actually A. Sahai, actually and B. Waters. Attribute Based Encryption for Fine-Grained Access Control of Encrypted Data. In *ACM conference on Computer and Communications Security (ACM CCS)*, actually 2006.
- [16] H. Harney, actually A. Colgrove, actually and P. D. McDaniel. Principles of policy in secure groups. In *NDSS*, actually 2001.
- [17] M. Ito, actually A. Saito, actually and T. Nishizeki. Secret Sharing Scheme Realizing General Access Structure. In *IEEE Globecom*. IEEE, actually 1987.
- [18] M. H. Kang, actually J. S. Park, actually and J. N. Froscher. Access control mechanisms for inter-organizational workflow. In *SACMAT '01: Proceedings of the sixth ACM symposium on Access control models and technologies*, actually pages 66–74, actually New York, actually NY, actually USA, actually 2001. ACM Press.
- [19] A. Kapadia, actually P. Tsang, actually and S. Smith. Attribute-based publishing with hidden credentials and hidden policies. In *NDSS*, actually 2007.
- [20] J. Li, actually N. Li, actually and W. H. Winsborough. Automated trust negotiation using cryptographic credentials. In *ACM Conference on Computer and Communications Security*, actually pages 46–57, actually 2005.

## **11.2 Books Referred:**

1. More ASP.NET (Teach Yourself) - Lowell Mauer
2. Guide to ASP.NET - Peter Norton,
3. Fundamentals of Database System - Ramez
4. Complete Guide to SQL server - Peter Norton.

## **11.3 Web links referred:**

[Http://www.Sourcecode.com](http://www.Sourcecode.com)

[Http://www.dbms.co.in](http://www.dbms.co.in)

[Http://A1code.com](http://A1code.com)