

**CLOUD COMPUTING ACCESS AUTHENTICATION
THROUGH MOBILE DEVICE
BASED ON FACE RECOGNITION**

تحقق الوصول إلى الحوسبة السحابية من خلال جهاز
الموبايل اعتماداً على تمييز الوجه

**By
Ali Mohammed Saab
Supervisor**

Prof. Alaa Hussein Al-Hamami

**This Thesis is submitted as the Partial Fulfillment of the
Requirements for the Master Degree in Computer Science**

College of Computer Sciences and Informatics

Amman Arab University

2015



Form (9)

College of Scientific Research and Graduate Studies

Authorization

We, the undersigned, pledge to grant Amman Arab University for discretion in the publication of the academic content of the thesis, so that the intellectual property rights of a Master thesis be back to the university in accordance with the laws, regulations and instructions relating to intellectual property and patent.




Advisor Name	Co-advisor Name	Student Name
Prof. Alaa Hussein Al Hamami		Ali Mohammed Saab
Signature: <i>Alahamami</i> Date: <i>9-11-2015</i>	Signature: Date:	Signature: <i>Ali S. Mohammed</i> Date: <i>10/11/2015</i>

شارع الأردن - موبس - عمان 11953 - الأردن
Jordan Street - Mubis - Telephone +962 7 8064 0040 - P.O.Box 2234 Amman 11953 - Jordan
Email: sauga@aau.edu.jo / Web: www.aau.edu.jo

Committee Members' Decision

The thesis entitled: "Cloud Computing Access Authentication through Mobile Device Based on Face Recognition" was submitted by the student, **Ali Mohammed Saab** was examined and approved on 16/6/2015.

Committee Members

Name		Signature
Prof. Alaa Al-Hamami	Chair/Advisor	
Dr. Shihadeh Alqrainy	Member	
Dr. Mohammad Otair	Member	

Acknowledgment

Praise is to Allah for the blessings, I look back with appreciation to my brilliant professors and my family. I look with gratitude to those who have touched my life with knowledge and encouragement.

I want to express my thanks to my supervisor Professor Alaa Al-Hamami for his guidance and support. He was always available when needed his assistance, therefore, I would like to convey my sincere gratitude to him.

Also, I would like to especially thank my parents for everything they have done for me and for helping me to get to where I am today.

As I express my gratitude, I must never forget that the highest appreciation is not to utter words, but to live by them.

The Researcher

Dedication

I dedicate this thesis to my wonderful family and to my supervisor Professor Alaa Al-Hamami.

I would like to thank my parents, my wife and my friends for all they have done for me and for helping me get to where I am today. They have been very supportive in my decision to receive a higher education. So, I dedicate my success to each one of them.

Ali Mohammed Saab

List of Contents

AUTHORIZATION	I
DECISION OF THE EXAMINATION COMMITTEE	II
ACKNOWLEDGMENT	III
DEDICATION	IV
LIST OF CONTENTS	V
APPENDIX	VII
LIST OF FIGURES	VIII
LIST OF TABLE	IX
LIST OF ABBREVIATIONS	X
ABSTRACT	XII
ABSTRACT IN ARABIC	XV
CHAPTER ONE	1
INTRODUCTION & BACKGROUND	1
1.1 INTRODUCTION.....	1
1.2 CLOUD SERVICES	4
1.3 DEPLOYMENT OF CLOUD MODELS	6
1.4 MOBILE DEVICE	8
1.5 MOBILE INTERNET PROTOCOL IPV6	9
1.6 MOBILE CLOUD COMPUTING.....	11
1.7 MOBILE SECURITY OF CLOUD COMPUTING	13
1.8 USER AUTHENTICATION OF CLOUD COMPUTING.....	16
1.9 BIOMETRICS AUTHENTICATION TECHNIQUES.....	20
1.10 STATEMENT OF PROBLEM.....	23
1.11 CONTRIBUTION.....	25
1.12 THESIS ORGANIZATION.....	26
CHAPTER TWO	27
RELATED WORK	27
2.1 INTRODUCTION.....	27
2.2 RELATED WORK.....	30
2.3 SUMMARY	51

CHAPTER THREE	53
THEORETICAL DESIGN	53
3.1 INTRODUCTION.....	53
3.2 FACE DETECTION PROCEDURE	58
3.3 FACE RECOGNITION PROCEDURE	61
3.4 THESE EIGEN APPEARANCES CAN NOW BE UTILIZED TO SPEAK TO BOTH EXISTING AND NEW CONFRONTS	63
3.5 EYE DETECTION PROCEDURE	64
3.6 FACE TILTING MEASUREMENT PROCEDURE.....	64
3.7 USER AUTHENTICATION FLOWCHART:.....	65
3.8 PREPARE IMAGE:	69
3.9 EYES DETECTION:	70
3.10 FACE DETECTION:	71
3.11 EYES LINE SLOPE (FACE ROTATION):.....	72
3.12 SUMMARY:	73
CHAPTER FOUR.....	74
THE EXPERIMENTAL WORKS.....	74
4.1 INTRODUCTION	74
4.2 CLIENT (ANDROID APP)	77
4.2.1 REGISTRATION	78
4.2.2 LOGIN.....	80
4.2.3 LOGIN TRIALS	81
4.2.3.1 AUTHORIZED LOGIN	81
4.2.3.2 UNAUTHORIZED LOGIN	82
4.2.4 ACCESSING CLOUD SERVER FILES.....	84
4.3SUMMARY	86
CHAPTER FIVE	90
CONCLUSIONS AND FUTURE WORK.....	90
5.1 INTRODUCTION.....	90
5.2 CONCLUSIONS	91
5.3 FUTURE WORK.....	93
REFERENCES.....	95

APPENDIX

Description	Page
Appendix	A1
Face Lock Application Public class Main Activity	A1
Get the size of the Image View	A3
Decode the JPEG file into a Bitmap	A4
Create an image file name	A6
Action take photo	A7
Face Detector	A32
Eigen face recognizer	A34

List of Figures

No.	Description	Page
Figure (1)	Deployment of cloud models	6
Figure (2)	Authentication Layered Model	15
Figure (3)	Proposed model	53
Figure (4)	The classifier extraction features	56
Figure (5)	Calculated by subtracting pixel	57
Figure (6)	Face degree	61
Figure (7)	User authentication flowchart	62
Figure (8)	Prepare Image flowchart	65
Figure (9)	Eyes detection flowchart	66
Figure (10)	Face detection flowchart	67
Figure (11)	Eyes Line Slope (Rotation) flowchart	68
Figure (12)	Model process	72
Figure (13)	GUI for the application	73
Figure (14)	Registration process	74
Figure (15)	Login screen	76
Figure (16)	Authorized login	77
Figure (17)	Authorized access	78

Figure (18)	Unauthorized login	79
Figure (19)	Invalid user	79
Figure (20)	Invalid angle	80
Figure (21)	Information stored at the server	81

List of Table

No.	Description	Page
1	Table of the result	81

List of Abbreviations

Abb.	Meaning
3D	Three Dimension
ABAC	Attribute-Based Access Control
ABI	Allied Business Intelligence
API	Application Program Interface
ATM	Automatic Teller Machines
EOTP	Encrypted One Time Password
FRS	Face Recognition System
GSM	Global System for Mobile
IaaS	Infrastructure as a Service
IDC	International Data Corporation
IP	Internet Protocol
IT	Information Technology
MAC	Media Access Control
MCC	Mobile Cloud Computing
NIST	National Institute of Standards and Technology
OTP	One Time Password
PaaS	Platform as a Service
PDAs	Personal Data Assistant
PIN	Personal Index Number

QoE	Quality of Experience
QoS	Quality of Service
RBAC	Role-Based Access Control
SaaS	Software as a Service
SMDP	Semi-Markov Decision Process
UBAC	User-Based Access Control

CLOUD COMPUTING ACCESS AUTHENTICATION THROUGH MOBILE DEVICE BASED ON FACE RECOGNITION

Prepared By

Ali Mohammed Saab

Supervisor

Prof. Alaa Hussein Al-Hamami

ABSTRACT

Cloud computing is a recent term that explains methods and techniques in the development of computer science, which considered quantum leap to the next generation by providing everything the user needs through the cloud.

In addition, the main purpose of Cloud computing with virtual methodology is to store data and personal files for users which are displayed through a variety of interfaces and software setups using the World Wide Web. This methodology offers the opportunity for users by providing them with sources when submitting the application which is considered only subject to regulation by several programming points.

Cloud computing is not just restricted to pc platforms, but also expanded to include mobiles in order to provide users with the combination of both

mobile and cloud computing services to deliver rich computational resource to mobile users.

Meanwhile, in order to provide users with the best services, mobile cloud computing was able to tackle cloud security through application security methodology that relies on biometrics for humans that is owned by the user of the unique human organs that proves the user's identity and that includes analysis of all of the qualities and characteristics of the user, specifically the facial area, This method which considered the most prominent of analysis methodologies that relies on the physical characteristics of the human body.

This thesis will display the way to prove the user's identity by analyzing the face of the user in the current study by capturing instant image to the user through the mobile front camera and then the user face will be analyzed so it can be recognized by the system. Once you confirm the validity of the user's identity, you can access to personal information and enjoy the best sources provided by using computerized mobile phone computing.

The main purpose in this thesis is not only to display what cloud computing has in services for its users, but to explain what cloud computing can make for users in terms of maintaining their personal data and information by

integration of confidential data with advanced technology to identify the user to and determine their identity by mobile phone.

This compendium, which outlines developments in the world of technology and its course increase, It is possible that these developments will lead to follow the modern methodologies of credibility, which is working to improve protection through the use of the front camera that resides in cell phones, that takes high-resolution image of the user in order to prove his identity and to verify the integrity of his information.

تحقق الوصول إلى الحوسبة السحابية من خلال جهاز الموبايل اعتمادا على تمييز الوجه

إعداد

علي محمد صعب

إشراف

الأستاذ الدكتور علاء حسين الحمامي

المخلص

الحوسبة السحابية هي مصطلح حديث، يشرح الأساليب والتقنيات في تطوير علوم الحاسوب، والتي تعتبر نقلة نوعية للجيل القادم من خلال توفير كل ما يحتاجه المستخدم من خلال السحابة.

بالإضافة إلى ذلك، فإن الغرض الرئيسي من الحوسبة السحابية هي لتخزين البيانات والملفات الشخصية للمستخدمين والتي يتم عرضها من خلال مجموعة متنوعة من الواجهات والبرامج باستخدام الشبكة العنكبوتية العالمية. تقدم هذه المنهجية الفرصة للمستخدمين من خلال تزويدهم بالمصادر عند تقديم الطلب والتي تعتبر خاضعة للتنظيم من قبل عدة نقاط في البرمجة.

الحوسبة السحابية لا تقتصر فقط على أجهزة الحواسيب الشخصية، ولكن أيضا تشمل الهواتف النقالة من أجل تزويد المستخدمين الجمع بين كل من خدمات الهاتف النقال والحوسبة السحابية لتقديم الموارد الحاسوبية الغنية لمستخدمي الهاتف الجوال.

وفي الوقت نفسه، من أجل تزويد المستخدمين بأفضل خدمات، الحوسبة السحابية النقالة التي تكون قادرة على معالجة الأمنية السحابية من خلال منهجية تطبيق الأمن التي تعتمد على القياسات الحيوية للإنسان التي تثبت هوية المستخدم والتي تضم تحليل كل من الصفات

والخصائص للمستخدم، وتحديدًا في منطقة الوجه. وهذه الطريقة تعتبر من أبرز منهجيات التحليل التي تعتمد على الخصائص الفيزيائية للجسم البشري.

في هذه الرسالة سيتم عرض وسيلة لإثبات هوية المستخدم عن طريق تحليل وجه المستخدم في الدراسة الحالية من خلال النقاط صورة فورية للمستخدم عن طريق الكاميرا الأمامية للهاتف المحمول ثم تحليل وجه المستخدم ليتم التعرف عليها من قبل النظام، وعند التأكد من صحة هوية المستخدم، يمكن الوصول إلى المعلومات الشخصية والتمتع بأفضل المصادر المقدمة باستخدام حوسبة الهاتف المحمول المحوسبة.

الهدف الرئيسي في هذه الأطروحة لا يعرض فقط ما لدى الحوسبة السحابية من الخدمات لمستخدميها، ولكن لشرح ما يمكن للحوسبة السحابية أن تقدم للمستخدمين من حيث الحفاظ على البيانات الشخصية والمعلومات من خلال البيانات السرية المتكاملة مع التكنولوجيا المتقدمة لتعريف المستخدم وتحديد هويته عن طريق حوسبة الهاتف المحمول المحوسبة.

Chapter one

Introduction & Background

1.1 Introduction

Cloud computing is a modern methodology in the real life. The cloud computing permits users to have the accessibility to the stored files or data from anywhere through using Internet. Cloud computing can result in several benefits such as enhancing throughput and accessibility, decreasing costs, and needs less training but at the same time it has several security issues (Pawle & Pawar, 2013).

The classical definition of cloud computing in general is a set of computers or servers that are linked together to create a system. It is the new form of application in the field of the Internet and it has become the common topic of researchers in industrial and scientific communities. It is considered to be a group of PCs and servers that are globally accessed through the Internet. Also it provides consumers with the resources and computing infrastructure as part of their requirements, hence consumers can use the services and applications that are available on the cloud through their Internet connection (Asrani, 2013).

The main definition of cloud computing in specific as the National Institute of Standards and Technology (NIST) is describes is a computing ideal that provides network access to resource pool, and this access be transparent, convenient and on-demand. Cloud technologies also offer elasticity and flexibility of cloud shared resources. These resource pools are conducted through several servers, networks, storage, applications and services, which can be utilized by the main user with a less organizing effort and communication with the cloud provider. Cloud computing is a modern technology, which provides beneficiary opportunities in many sectors (Pawle & Pawar, 2013).

The cloud computing shifts both of the application software and databases to the huge data centers, thus it has now become the upcoming future generation architecture of any Information Technology (IT) organization. Cloud computing is not just restricted to personal computers, but also it is developing to move forward, with several web-based and mobile applications that are developed with cloud technologies. It is important to note that the terms mobility and ubiquity are the main characteristics of the next generation network (Tammaana & et al, 2013; Fernando & et al., 2013).

IT organizations have recently considered cloud computing as the future generation architecture. Comparing to traditional methodologies, Cloud computing determined to transform the application software and the

databases in to large data centers, as the database and services management are not trusted. In cloud computing, both software and data are not completely implicated on the user's PC (Guha & Shrivastava, 2013).

Cloud computing gives users and businesses to utilize applications without the need to download them and to access their data files using any device through Internet access. Cloud computing offers are different types of on-demand services by using the internet such as software, hardware, server, infrastructure and database. The basic idea of Mobile Cloud Computing (MCC) tends to seize the advantages of Cloud Computing that are available for mobile users while at the same time offering additional functionality to the cloud as well. MCC will assist in reducing the disadvantages of mobile devices specifically the processing power and data storage. Meanwhile, through moving the execution of commutation application to the cloud, it might also assist in enhancing the mobile battery life (Krishnamoorthy, 2013)

Moreover, Cloud computing offers a variety of issues to a variety of people. The core specifications most interpretations have in common are on demand secure access to metered services from almost anywhere, scalability, reliable resources of cloud computing, and displacement of data and services from internal to external organization (Jansen & Grance, 2011).

1.2 Cloud Services

Software as a Service (SaaS): An application is presented as a facility or a service to users that can be accessed through the Internet. The capability that is offered to the user is to utilize the provider's requests, which runs on a cloud infrastructure. These applications can be accessed from different types of devices through an Internet web browser, or an application. The user does not deal with or manage the cloud infrastructure, which includes operating system, network servers, data storage, or sometimes application ability, with the exclusion of restricted user specific document support configuration settings (Khan & Ahirwar, 2011).

For instance, Google Doc can assist users in while not requiring download of any supportive application for that. Other suppliers such as Amazon. com provide cloud services and users are required to pay just for the services that they need (Khan, & Ahirwar, 2011).

Platform as a Service (PaaS): In this model PaaS services support application design, development test, hosting and deployment. The capability that is offered to the user is to utilize the cloud infrastructure that is built with the help of programming languages, services, software libraries and tools that are provided by the provider. The user does not deal with or organize the cloud infrastructure such as network, servers, operating systems, or storage, however the user has to manage the application

deployment and possibly settings configuration for the application-hosting environment (Mell & Grance, 2011).

Infrastructure as a Service (IaaS): The benefit that is offered to the user here are providing data storage, computer networks, and other basic resources whereas the user is allowed to utilize and process quantitative software, in which operating systems and software applications are included. The user here also does not manage or organize the cloud infrastructure, but can manage data storage, operating systems, and apply applications; yet it has restricted the management of network parts (Mell & Grance, 2011).

1.3 Deployment of Cloud Models

The cloud resource can be utilized in three main techniques, that are based upon the organizational structure and the district, called Private, Public and hybrid cloud service usage, as shown in Figure (1):-

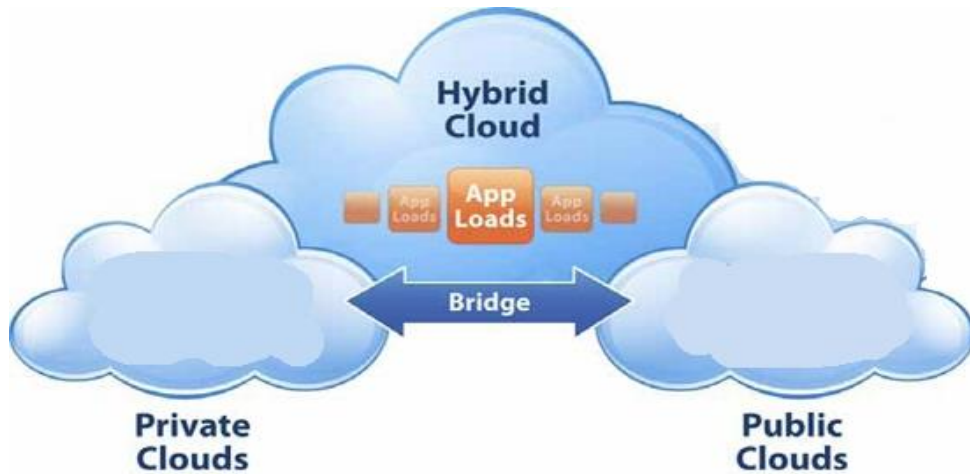


Figure (1): Deployment of cloud models

1.3.1 Public Clouds

The cloud substructure is created to be obtainable by the broad or public use or by a great industry group. Hence, public cloud is described the general use to customer, which means opened to several users over a shared infrastructure. It is utilized, operated, and organized by a third-party seller from one or more data centers; therefore the customer who uses the shared cloud services providing has a less privileges of control and oversight of the physical and logical security issues than of a private cloud.

The resources in the Public cloud are animatedly accessed by the Internet, specifically through web applications or web services, from a third-party provider who dividends resources and bills on a fine-grained, utility-computing foundation (Ravindranath & Dr. Raja, 2013).

1.3.2 Private Clouds

The cloud substructure is opened exclusively for an organization. It may be achieved and controlled by that organization. Hence, private clouds is describe the exclusive use to the customer, so it differs from public clouds in that system, computing, and storage substructure related to private clouds is devoted to only one organization and is not common with other organizations.

The organizations will buy and build the private cloud for the customer, then this cloud is managed by a vendor and owned by a customer, so it has high grade of management and clarity, it is easier for a user to fulfill with proven organization security principles, policies, and regulatory obedience (Ravindranath & Raja, 2013).

1.3 Hybrid Clouds

A Hybrid cloud is described in the general and exclusive use to the customer based on the cloud computing type, which means it includes multiple private and/or public clouds. The organizations in this model might run non-core applications in a public cloud, while preserving core applications and complex data in-house in a private cloud (Lizhao & et al., 2013).

1.4 Mobile Device

Mobile has turned to be a very common device in the domain of computing. There has also been an increase in development and sales of mobile devices such as smartphones, tablets etc. assisting different varieties of mobile computing and networking tools. People select these devices as their first priority for work and entertainment activities.

The dependency of the mobile phone has increased incredibly in recent years, as people highly depend on mobile phones and use them as mini-computers that most of the time travel with them as to keep them linked 24 hours a day, the dependency will definitely continue growing. Mobile phones nowadays are considered to be an essential fragment of the business world and the basic significance of mobile database is predictable (Khan & Ahirwar, 2011).

With the incredible improvements in technology, people are expecting more and they require to be provided with services anytime anywhere. A modern thesis from Allied Business Intelligence (ABI) Research has declared that 'cloud computing' will absolutely change the next generation of mobile applications development, and their utilization. Cloud computing will efficiently decrease the need of progressive handsets in order to execute mobile applications (Khan & Ahirwar, 2011).

1.5 Mobile Internet Protocol IPV6

The following model Mobile Internet Protocol version 6 (IPv6) permits an IPv6 node to act like a mobile and to make a qualitative modification in its position on an IPv6 network and while preserving availability. Mobile nodes Linking conservation is not produced by adjusting the transport layer protocols, but by controlling the modification of reports at the Internet layer through using Mobile IPv6 messages, selections, and progressions that guarantee the right supply of data neglecting the mobile node's position (Davies, 2012).

The major advantage of Mobile IPv6 is that even though when the mobile node changes its place and source, still the connections in which the mobile node is communicating are maintained. In order to achieve this, connections to mobile nodes are done with a particular address that is continuously appointed to the mobile node, the mobile node is constantly ready. Mobile

IPv6 provides Transport layer linking when a node tends to change one connection to another through making address conservation for mobile nodes at the Internet layer (Davies, 2012).

Classical IPv6 networks have a shortage in the native support for mobile nodes. Mobile nodes might modify their point of link to the Internet but still are handy by a static IP address. With mobility support, communication is applicable while the mobile node triggers around (Liske, 2005).

1.6 Mobile Cloud Computing

Cloud computing is not just restricted to PCs (personal computers); it has a major impact even on mobile technology. Mobility is the key feature of the next generation network.

The term Mobile cloud computing (MCC) is considered to be the integration of both mobile networks and cloud computing in order to carry out the advantages for net operators, mobile users, and cloud providers. cloud computing dwells as users keep their data and tasks on the Internet rather instead of being kept on separate mobile devices or customer system. This is called on-demand services (Guha & Shrivastava, 2013).

Mobile cloud computing is a modern model that utilizes the idea of clouds in moving the data storage and process from mobile devices to more powered and centralized computing platforms that are located in clouds. Users can access these platforms over wireless connections via web browsers on their mobile devices. This is typical to cloud computing, but the user sideways has been modified to achieve flexibility and makes it applicable for mobile devices, but the core idea after it is constant which is cloud computing.

The Mobile cloud computing is becoming a great part of the development of IT technology also in industry fields, Cloud computing is considered to be an evolving facility model that enables mobile devices to decrease the energy feastings and to run different facilities on the clouds distantly. Thus, a combination of electronic devices like smart phones, Personal Data Assistant (PDAs), tablets, cloud computing and global mobile network, resources are meeting with each other to arise as a fresh bitch of MCC (Khan & et al, 2013).

The main idea of MCC intends to seize the benefits of Cloud Computing that are available for mobile users while at the same time providing additional functionality to the cloud. MCC will help in solving limitations of mobile devices specifically in the processing power and data storage. Meanwhile, by moving the execution of communication application to the cloud, this will enhance the mobile battery life (Geetha, 2013).

1.7 Mobile Security of Cloud Computing

When cloud computing users save their personal data or information to different services across the Internet, illegal people could access it sometimes. Hence, safety and security is the core drawback in cloud computing. In order to offer security, a better authentication methodologies in cloud computing is required.

The main concepts of cloud computing is the best contract about the way to achieved safety and security in different planes. The achieved security lets information managers to see that security is first and is only the concern with cloud computing (Ravindranath & Dr. Raja, 2013).

The migration to the cloud system offers many benefits, and the main benefit of cloud computing system is that the stores applications and data in the database centers which can be reached from anywhere and at any time, so this movement to the application software, data and services is not fully trustworthy and this system is not without security risks (Butoi & et al., 2013).

For full understanding of reducing security risks, the focus should be on security issues that could occur in the cloud such as privacy, several security rules, active of the services provided, trust between the objects, and vigorously creating trust domains (Waghmare & Prof. Chavan, 2013).

In the area of MCC, the risk is comprising data duplication, constancy, controlled scalability, unreliability, undependable availability of cloud resources, mobility, trust, security, and privacy. These risks have become an obstacle in the fast development of MCC's subscriber.

To attract potential consumers, the cloud service provider has to target all the security issues to provide a completely secure environment in MCC. This includes risk of user's data being stored on cloud servers, the security warnings caused by several simulated technologies, and imposition through different attacks. As MCC depends on cloud computing, all the security problems are genetic by MCC with the extra drawbacks of resource restriction mobile devices. Due to resource restriction, the security systems presented for the cloud-computing platform cannot be directly executed on a mobile device. A Lightweight secure framework is required to offer security with less interaction and processing overhead on mobile devices (Waghmare & Prof. Chavan, 2013).

Insufficient authentication could result in data leakage and security attacks. Meanwhile, In this thesis, the security problems of cloud computing is highlighted, specifically on authentication. In order to overcome the authentication issue in cloud computing, there are several outdated methodologies, as well as biometric methodologies as explained in the following section with some of its disadvantages (Waghmare & Prof. Chavan, 2013).

Cloud computing has to tackle security and privacy issues before large organizations start using it. Moreover, the organizations can reduce their expenditures on the resources but still they have to spend more money on the bandwidth. Sufficient bandwidth is needed to provide exhaustive and complex data on the network. Due to this, several organizations are waiting for a reeducation in the cost of bandwidth before switching to cloud computing systems (Arsani, 2013).

As MCC is a cloud computing based platform hence all the security problems in cloud computing are genetic in MCC but with a few limitations of resource limitation mobile devices. Due to this resource restriction, the security algorithm that is designed for the cloud environment cannot be executed on mobile devices. A lightweight secure framework is required that will provide security with less communication and execution load on mobile devices (Waghmare & Prof. Chavan, 2013).

1.8 User Authentication of Cloud Computing

Authentication indicates the trust mechanism between two entities, or verification parties. These parties must contain both an ID and a key. Verification is founded through executing a cryptographic process on both objects identities and keys. The cryptographic procedure (verification algorithm), then starts to build the basis of the confidence between these objects. A network transportation or verification flow is required for providing the link between these parties in order to perform the authentication algorithm.

Regarding the International Data Corporation (IDC), there are many problems or disputes that face cloud computing such as safety and security, availability, performance, absence of interoperability standards, causing back internal integrating with IT, and lack of capability to modify. According to the IDC's review on the cloud services, security is considered to be the major concern issue that faces cloud computing. The model can be explained graphically as in Figure (2).

Key, Identity	Key, Identity	Key, Identity	Key, Identity
Authentication Algorithm	Authentication Algorithm	Authentication Algorithm	
Authentication Channel [optional]			
Authentication Flow			

Figure (2): Authentication Layered Model

The layers are tightly coupled and a channel is not required as in most authentication systems. In fact, the benefit of a verification channel is letting it to be applicable to lightly link the layers and provide a large range of verification ideas. In the layered model, each group is not applicable. The layered model will supply the framework with a procedure work to organize different groups of channels, flows, algorithms and entities as well as the reliance type they deliver.

Authentication is concerned with identifying the ID of more than one party in a discussion or period. Such a methodology is acknowledged as inappropriate in approximately all data communication periods. This is because of the absence of information authentication like face recognition or situational information. Moreover, in data communications the synonymous of facial recognition would be a Media Access Control (MAC) or Internet

Protocol (IP) address. For the time being, most of the data items are commonly not confident any longer as distinguishing authentication information. Hence, in data communications sessions, identification identities are held by engaging them to an authentication process.

In order to provide security in cloud computing, a proper authentication technique is required. Typically, when authentication is established, it depends upon the information about one or more of the following (Pawle & Pawar, 2013):

- I. Subject knowledge, such as keyword or confidential information.
- II. Ownership of the user, such as smart card, identification, passports.
- III. Biometrics of the user, such as fingerprint, voice, face and etc.

1.8.1 Traditional Authentication Techniques

In order to authenticate an authorized user in cloud computing some existing authentication schemes have been explained. Firstly, ordinary name and password is expended in cloud computing. however these methods are easily hacked. Then, graphical three Dimension (3D) password was introduced by several systems but it involves further area and time consuming process.

There are two types of traditional authentication techniques that were firstly used in the past such as (Pawle & Pawar, 2013):

- I. Password is the most commonly used way of verification that requires a login and password combination but it is not secure.
- II. One Time Password (OTP), where the password is postulated only once and upon demand, and the password is effective for a restricted amount of time. This technique can prevent a password from being hacked which makes it secured. The only drawback of these systems is that they are expensive.

In this thesis, a new biometric authentication system is proposed in order to overcome the security concern, and it is Face Recognition System (FRS), which is used to identify authorized users.

1.9 Biometrics Authentication Techniques

Nowadays, biometrics is considered to be the safest and securest system that is being deployed. It helps to reduce a lot of drawbacks that are presented by the above-stated methods of user verification. Biometrics can be understood as a powered technique to exclusively differentiate users based upon their behavioral or physiological features.

Since biometrics is used as an authentication technique, then the password is human organs or physiological characteristics. Several biometrics techniques are explained below (Majge & Kulkarni, 2011):

- Voice Recognition – As explained by the name, voice recognition technique requires verification with spoken data. Voice recognition is used to validate user's character depending upon the voice tone and speech style. On the other hand, this technique is no longer considered to be the safest, due to the easy way to record a user's voice and hence be used by an illegal user. Also speech of a user may differ due to illness, so identifying a user by the voice is difficult.
- Signature Recognition – Signature recognition is expected to distinguish a user's identity depending on the style of their rare signature. People cannot sign in a consistent manner always, so confirming an authorized user is difficult.

- **Retinal Recognition** – Retinal recognition is expended to distinguish people through the shape of their blood vessels on the retina. This method is considered to be very indiscreet and expensive.
- **Iris Recognition** – Iris recognition is a similar method to the retinal recognition, which distinguishes users depending on the unique patterns within the circular region adjacent the pupil of the eye. This technique is also invasive and expensive.
- **Fingerprint Recognition** – Fingerprint recognition is very common nowadays; it denotes to the programmed method of authenticating user through matching two human fingerprints. Several factors can affect the recognition procedure such as the dryness of fingers, and soiled fingers in which will make the system show an error.
- **Hands Geometry Recognition** – Hand Geometry method depends on the geometric shape of the hand. This method takes into consideration several human characteristics such as; the scope of the palm, finger length, width and so forth. But this technique has a few disadvantages such as it is suitable only for adults because children's hands size will be changed over the time. Moreover, constant use of jewelries will conclude in modifying the hands geometry, and it is not suitable for users suffering from arthritis, as they will not be able to put the hand on the scanner in an appropriate manner.

- Palm recognition – Palm recognition depends on edges, principal lines and rumples on the top of the palm. The cost of this technique tends to be high and is not suitable for youngsters as their palm lines modify as soon as they are fully grown up.

All of the above-mentioned methods tend not to be applicable and are not very much useful due to their various disadvantages. In this thesis in order to solve drawbacks of all these security techniques and to provide proper security for user authentication in cloud computing, a new method is suggested which is to use a biometric technique termed “FACE RECOGNITION”. In addition, face recognition is flexible in authenticating users (Majge & Kulkarni, 2011).

The persons face shows a core part in social interaction. Facial identification is one of the most preferable techniques of biometrics because it is an impartial, non-intrusive, easy-to-use, method which requires little physical contact in comparison with other biometrics systems (Majge & Kulkarni, 2011).

Face recognition depends on several factors in a human face such as; the figure and position of the eyes, lips, nose, eyebrows, and chin or on the whole analysis of the face image. The system will identify a user through a captured face image that will be taken remotely without having to touch the person being identified to be touched, hence the verification process does not need communication with the person (Majge & Kulkarni, 2011).

In this thesis, face recognition system is highlighted due to its security which makes it a superior decision to be used rather than other outdated or other biometric validation methods. Using face recognition system further perfects the security level of cloud supplier in terms of authentication security. In addition, Due to the flexibility the high technology devices provide these days, which contributes in providing biometrical authentication such as front camera, photo snapshot, and fingerprints.

1.10 Statement of Problem

Security has been a serious problem facing cloud computing for the last decades; therefore the mobile cloud requires a secure communication between cloud and the user. The security services in any communication include access control, authentication, non-repudiation, authorization service to mobile user and so on.

The basic challenge currently and domain problem includes protecting and securing cloud computing data and the access approach. Since cloud security is an evolving sub- domain in computers security and network security, several methods have been proposed in order to combat this. Perhaps the most popular of these is the simplest and most effective; which is biometric face recognition.

While technology keeps changing its environment day-to-day, users requests also modifies. Users demand service quality, high level of data security, and authentication flexibility. Meanwhile, Cloud computing transports both the software application and databases to the outsized data cores, in which the controlling of the data and services may not be secured and trusted.

Cloud computing requires looking for a new way to limit the access to any confidential and secure information. Explaining how this is done, in this thesis SaaS and PaaS layers consist a few less levels of safety tools that assist securing data and the access to universal platform.

In this thesis, an optimal mechanism is presented to reduce any risk that may occur in communication through the transferred data or information between user and cloud. By using Biometric recognition (Face recognition) the Mobile user can be authenticated so it is possible to utilize the information available for this user.

1.11 Contribution

Cloud computing requires a huge security system that protects its own user's data and personal services. The problem is addressed to increase the security and make a fast recovery of client's data while authenticating their identity.

The main contribution of this thesis focuses on the security of the system and enhancing the level of data safety and privacy through introducing a new authentication techniques which is the biometric recognition (Face recognition) through the mobile phone to produce a resolution for the problem of cloud security, and to guarantee the secure access to restrict data/services in the cloud using a mobile phone also that will facilitate the work of the people who use the mobile. These contributions are summarized as the follows:

- The security authentication in this thesis depends specifically on the Android application that is being used by a mobile phone.
- Access methodology includes a username and password that are provided by a user then the system will have a photo capture for the current user in order to be compared with the user photo that is previously stored in the database to ensure the identification of the client.
- A user will be authenticated to access the data that is stored in the cloud system, after approving the users identity.

1.12 Thesis organization

This thesis includes four chapters in addition to chapter one. The following is a summary for the chapters:

- Chapter two: presents the main problem that is discussed in this research and a summary of the most important related works.
- Chapter three: introduces a description of a proposed solution of the problem, and is explained by flowcharts and algorithms.
- Chapter four: discusses the experimental works and results.
- Chapter five: introduces conclusions and future works for this research.

CHAPTER TWO

Related work

2.1 Introduction

One of the dangerous issues facing cloud computing these days is security, since cloud users have their full information and data across the Internet, this would lead to unauthorized access by unauthorized people.

Through the explosion of mobile applications and computing model, cloud computing is becoming unsecured because Mobile Cloud Computing (MCC) essentially required outsource computations and storage tasks to cloud servers due to its memory deficiency. Sometimes cloud servers that are being used are not trusted and hence it would causes an unsecured migration or resourcing to the mobile, which will leads to the critical problem of security.

Each system faces issues related to securing its information system and preventing the illegal usage of the data, and with the increased usage of the Internet, changing demands, and new trends of using cloud for storing data and information. Cloud security is now very common for research and development, Cloud security indicates all the broad set of policies, procedures, techniques and methodologies that are efficiently applied to secure applications, data, and infrastructure of cloud computing.

Cloud security algorithms on the mobile cloud can be implemented but the main issue is that computation power of security algorithms that is used is high and mobile devices tend to have little computation power and battery life.

The confidentiality and integrity of accessing resources or data in cloud must be protected. It is considered that cloud computing is not trustworthy system, and mobile devices are semi trusted.

Biometric authentication is the most common methodology analysis and measurement of human features that depends on the physical characteristics of any human body. As long as these features have been taken and stored, they can be used authorization a user. Moreover, the idea is a combination of machinery and physical features of any user that group together to create an authentication system. Authorizing a user indicates whether you are whom you claim to say, hence either to let users access the system or being denied. Biometrics is explained as an emerging technology. But the major problem still present in this methodology which is the security of data in which security is cloud computing major issue as well. Mobile security risks tend to be more than in the traditional cloud.

The benefits of a biometric system are obvious and clear. As long as they are an intrinsic part of the user, there is no need for the user to recall a password, keyword or Personal Index Number (PIN).

In order to maintain a user secure access to the system, mobile devices are linked to the mobile network through base stations such as satellite, access points, and so forth. that create and take the lead of the link between the networks and mobile devices. Mobile user's requests then information such as ID and location are sent to the central processors which are linked to servers. Then mobile network operators are able to provide services to users such as authentication and authorization.

Mobile cloud computing indicates that both data processing and storage are held outside the mobile device. The security of data and storage in both cloud and mobile computing can assist to keep an eye on intruders and to embrace checks on unauthorized access to data and network.

In this thesis, the basic issue and problem involves security and protecting cloud computing data and to enhance its access methodology. The main idea is to eliminate any illegal intruder that is trying to access the confidential information. In order to guarantee the unauthorized usage of any resource is secured, there has has always been several attempts in encrypting the lowest layer data of any architecture.

The goal of this thesis is to contrive a natural function that uses the Application Program Interface (API) delivered by numerous cloud services. The mobile application directed the Android platform since its vital coverage. APIs were expended from the next cloud services: Dropbox, Google Drive and Box.

2.2 Related Work

(Jensen & et al., 2009): The main concept of cloud computing is offering ascendable capitals as a service through the Internet. The core function for the cloud is economic benefits, as long as it provides the limitation of expenditures. In order to achieve this, moreover, some challenges still exist to be resolved such as safety and security trust problems, as long as the user's personal data has to be published to the cloud and hence makes the defense weak of the data holder. The author focuses on mechanical security problems that arise from the practice of cloud service.

The authors presented several problems of cloud computing security, examined issues related to application of XML signature and the web services security frameworks (which attack the cloud computing system). Also they presented the cloud computing browsing context (Saas) and (PaaS), and described the danger of overflowing spasms on cloud systems (IaaS).

In general, a perfect initial opinion for enhancing cloud computing security issues that involves strengthening the security capabilities of web browsers and web services.

(Zhang & et al., 2009): Cloud computing offers an elastic computing design and several resources that allow resource on demand and immediate pay as you go functions models. The author believes that modern applications can motivate these models to reach new features that the legacy applications do not contain. The author maintains to create elastic applications with augmented resource constrained platform to support security, such as mobile devices, and elastic computing resources provided by the cloud.

The elastic application contains more than one web lets, each of them can be run on a device such as mobile cloud system or cloud system itself, and it can be transported between them depending on the dynamic modifications of the computing environment or user requirements on the device.

The author explains the basic idea of this new application model, identifies its unique security specifications, and overviews the design constraints to create elastic applications. As a first step the author proposed a solution for security problems and authentication between web lets running on a mobile

and those on the cloud. Then he presented a secure methodology for how to authorize the user and access the data via external web services.

The author analyzes the major security vulnerabilities to elastic application and reveals security objectives that should be contained by the infrastructure. Then presented an authentication technique and secure communication methodology for elastic applications.

(Chow & et al., 2010): Cloud computing is an important aggregation requirement for mobile security. Ordinary techniques have caused several limitations and computational restrictions that must be taken into consideration by mobile security technologies to be effective. The author has presented how cloud computing have all of these issues by using a fundamental approach on a flexible device that provides authentication.

The author has explained how cloud computing can contain all of these problems through using a basic approach on a flexible device that supports authentication decisions that is called Trust Cube that takes into consideration the authentication infrastructure, as well as a behavioral authentication approach that is called implicit authentication that translates users behavior into authentication scores.

New authentication challenges are presented by cloud computing such as growing requests for practical authentication to use services and data for

companies and users. Meanwhile, cloud computing offers several skills such as central analysis and control. There is another important trend to be realized which is the shift in devices from ordinary regular PCs towards mobile devices and mobile platforms. Generally, mobile platforms usages differ from PCs, as well as support behavioral data, such as locations and call logs.

The author explains in particular implied verification, in which a client uses previous social data to verify, and that is well adjusted for mobile devices. Based on these ideas, the author has built cloud authentication system.

The system is capable of permitting different verification techniques in a policy-driven manner, from TCG-style device integrity measurements to passwords. The system is compatible to support modern, cloud-oriented authentication methodologies. Specifically, the author combined the system with verification, and explained simple end-to-end use cases along with the authentication framework.

(Roberts II & Al-Hamdani, 2011): In this paper, the authors discussed the basic security problems with cloud based computing and cloud operating systems. Recently, cloud computing has witnessed a huge increase in popularity in which major companies began to announce cloud based goods, marketing the use of the cloud and releasing an Open Source cloud OS such

as Google and Microsoft. Meanwhile, increasing cloud popularity will lead to increasing the aware of cloud and the demand for security. The author discussed sole safety awareness for cloud computing and common security matters as well. Meanwhile, the author also discussed the solution to these issues and evaluated them. In the proposed system, the author gave customers the ability to choose exact security points for items in order to produce a security matters that all users must be alert of before starting to utilize cloud based services. The author presented security levels for data in the cloud that all users must take into consideration.

(Das & Debbarma, 2011): Biometrics authentication is a beneficiary method to be exchanged with password authentication. For instance, fingerprint authentication is one of the best techniques.

Regarding the transaction, finger print is needed at the Automatic Teller Machines (ATM) with the utilization of high resolution fingerprint scanners. As banks security techniques plays a critical role in limiting attacks on customers. These techniques are fundamental when taking into account vulnerabilities in civil litigation. Meanwhile, banks must fulfill several standards as to guarantee an integrated, secured banking environment for their customers.

ATMs have turned to be a fundamental technology in which it produces financial services to an incredible number of the population in several countries. Biometrics and fingerprint authentication, carry on to have an approval to be considered as a reliable method which provides a secure access through verification and validation processes.

The author tends to focus basically on fingerprint scanning and identifies a high-level for the modification of identity of existing ATM systems through using security protocol such as PIN and biometric fingerprint technique. Meanwhile, the author was able to develop a fingerprint technique as a biometric mechanism in order to enhance the security characteristics of the ATM for a banking transaction banking system.

(Chang & et al., 2012): In this paper the author has employed an open source Ubuntu Enterprise Cloud to establish a Ubuntu Cloud Computing, where a cloud controller (CLC) can be attached to a number of cluster controllers (CC), upon which they can initiate couple of cloud services, for example, SaaS, PaaS, and/or IaaS. A cloud controller (CLC) will setup connection to mobile devices or thin clients via wired Ethernet or wireless Wi-Fi or 3G Network. Mobile device or thin clients designated to below capacity embedded platform with Linux system in which Jam VM virtual machine is used to develop the J2M Environment and GNU Class path acts as sort of a Java Class Libraries. Finally, the rapid facial recognition and

fingerprint identification accomplishes fast access control in Ubuntu Cloud Computing for preventing illegal incursions outside the cloud computing system. It takes below 2.2 seconds to finish the authentication and therefore our proposed approach outperforms two alternatives benchmarks. Furthermore, the low capacity embedded platforms with Linux system are connected to Private Small-Cloud Computing via wired Ethernet or wireless Wi-Fi or 3G network. Finally, the rapid facial recognition and fingerprint identification, which out performs two alternatives benchmarks, accomplishes fast access control in Ubuntu Cloud Computing for preventing illegal in cursions outside the cloud computing system.

(Soyata & et al., 2012) Face recognition applications for airport security and surveillance can benefit from the collaborative coupling of mobile and cloud computing as they become widely available today. This paper discusses work with the design and implementation of face recognition applications using they mobile-cloudlet-cloud architecture named MOCHA and its initial performance results. The challenge lies with how to perform task partitioning from mobile devices to cloud and distribute compute load among cloud servers (cloudlet) to minimize their spouse time given diverse communication latencies and server computation powers. They preliminary simulation results show that optimal task partitioning algorithms significantly affect response time with heterogeneous latencies and

computation powers. Motivated by these results, they design, implement, and validate the basic functionalities of MOCHA as a proof-of-concept, and develop algorithms that minimize the overall response time for face recognition. The authors' experimental results demonstrate that high-powered cloud lets are technically easy and indeed help reduce overall processing time when face recognition applications run on mobile devices using the cloud as the back-end servers.

These architectures are designed to minimize the overall response time of the face detection and face recognition algorithms given heterogeneous communication latencies and computation powers of cloud servers at diverse geographical placements. They have designed MOCHA to integrate mobile devices (e.g., smart phones), the cloud let, and multiple cloud servers and demonstrated that cloud lets are technically feasible and beneficial at minimal additional costs. They have used intuitive barrier-based synchronization for utilizing multiple cloud servers for parallelism. To the best of their knowledge, this is the first work to show such an architecture with the three components working together with specific algorithms, applications, and initial results. The authors' simulation results show that, 1) more intelligent task partitioning algorithms employed by the cloud let permits response-time improvement by offloading work from the mobile device, 2) they sometimes decrease as the number of cloud servers increase and this

improvement is more emphasized when cloud lets are in place, and 3) communication latencies affect the response time considerably, which can be partially coalesced when cloud lets are used as buffers. The experimental results validate the simulation results and show that MOCHA indeed reduces the overall response time for face recognition. They plan to extend the experiments using real cloud services (e.g., AWS) and mobile devices (e.g. Android phones) with more heterogeneous latencies and computation powers in large scale. The authors' future work also includes more sophisticated synchronization algorithms permitting cloud-to-cloud communications, rather than multiple cloudlet-cloud communications links.

(Ajayan, 2013): In increased handling of mobile computing, peeling its crowded services is difficult because of its general complications. The main mobile computing issues are resource scarcity, frequent disconnections, battery life and mobility. The main aim behind mobile cloud computing is to strengthen the mobile user by offering a high functionality. The idea of divesting data and cloud computing computations is extended to identify the inherited issues in mobile computing through using the same resource providers more than the mobile device itself. The term "Mobile cloud in Disaster relief" refers to when massive earthquake or floods or cyclones result in human damage and many much property devastation. Disaster help

usually suffers issues due to the manpower limitations, lack of transportation, battery life and weak communication.

The author used photographs that were taken from several mobile devices of a disaster struck area and then transmitted it to cloud servers. Then the server groups put these images all together to produce a big panoramic image, and this image explains the current status of the topography of the area after the disaster.

This paper declared this one of the services that are provided by cloud computing and produced to customers to make a beneficiary use of it. The topographic panoramic image production will play a vital role in disaster management and recovery processes. Utilizing the offloading technique can enhance the energy and the battery life.

The authors focuses on analyzing cloud computing services produced to users and the possibilities to upload a content using major free cloud storage providers that will enhance for future usages with real-time video captured by users on the spot of an event.

(Pocatilu & et al., 2013): This paper declared almost all cloud services provides free Application Program Interfaces (APIs) for developers. As it can be seen from the examples, the code required to use the APIs is intuitive, easy to use and it generally follows the same pattern.

All platforms include an authentication and authorization phase that uses a Web based access or a dedicated Android activity. The user is required to authorize the application. The authorization tokens can be stored so that further use of the application does not require user interaction at this level.

The authors focus to analyze mobile application developers' possibilities for syncing content using major free cloud storage providers. It describes cloud computing in mobile context and highlights cloud providers API's.

(Liang & et al., 2013): Mobile Cloud Computing (MCC) gives the ability to mobile devices to share their computing data and files storage and other functions into the cloud in order to have more capacity and functioning. One of the basic and core issues that the author focuses on is the way cloud can effectively take control of the large requirements from mobile users when cloud resource is restricted. The author put the light on a different MCC adaptive resource allocation model that is explained in order to attain the best resource allocation through considering both mobile devices and cloud. In order to accomplish this objective, the adaptive resource allocation was modeled as a Semi-Markov Decision Process (SMDP) in order to snapshot the dynamic received and sent of resource demands. More imitations are adapted to highlight the proposed idea that could reach higher system prize and less service obstructiveness in comparison to outdated techniques.

The author proposed an SMDP based model that is used to assign cloud resources in standings of VMs that depend on desires that are taken from mobile users through taking into account the advantages and expenditures of together cloud and mobile devices in order to get the maximum system benefits and to achieve different Quality of Service (QoS) levels for mobile users.

(Ghadirli & Rastgarpour, 2013): These days, the quick growth of cloud computing helped many industries to move their computing activities to clouds. Authors of virtual learning are searching for the methodology to use cloud through mobile devices. The author presents a model to involve the compensations of both mobile sharp learning technology and cloud computing. The design of proposed system depends on multi-layer design of mobile cloud computing. Even with the challenges that are facing the system, it has increased the life of mobile device battery. It will decrease the storage capacity and processing capacity. The proposed system gives users the ability to enjoy through the intelligent learning anywhere anytime, reduces costs of training and hardware dependency, and increases consistency, data reliability and efficiency.

Intelligent learning programs and data will be integrated in the “data center” layer in the cloud. In which it depends on multi-layer architecture of mobile cloud computing. In the proposed model, the relationship between QoS and

Quality of Experience (QoE) is for deciding the performance of cloud based systems. It has several valuable compensations such as it increases the mobile battery life and space of storage and processing capacity, also reduces learning costs and hardware dependency.

(AL-Khashab, 2013) This thesis has declared that the concept of cloud computing is still unclear to many. Therefore, this thesis tried to clear the basic concepts of cloud computing such as general meaning of cloud, security related issue, characteristic, deployment and service model. Then, it focuses on the authentication concept to communicate through cloud, to make cloud users feeling that their data will be secure and available to them.

In this study, the researcher has designed a successful implementation of cloud authentication; when the users used the internet and before any communication goes across a network they need to be authenticated with the cloud. A third party is responsible to check if the user is authorized or not, and after that, it gives identification to the user for safety access to the cloud.

The proposed model focused on two points, the first point is to prove authentication through image as determined by the user. this image is used to prove if the cloud user is authorized or not.

The second point is to generate authentication through using multiple password technique in the cloud; it is a new research field which is gaining

interest from cloud users because the probability of brute force attack for breaking the password can be reduced when there is an increase in generated multiple passwords from single password.

(Zhang & et al., 2013): A mobile cloud computing system is conducted of different services and means as to be assigned by the cloud service provider to mobile cloud users. Meanwhile, cloud resources a mobile cloud computing system is conducted of mixed resources and services to be assigned by the cloud service provider to mobile cloud users. On the other hand, few of these resources are changeable (users can use cloud storage from different and several locations) in which they offer typical tasks to the clients. furthermore, some of the cloud resources are integral that the user will require them as a packet (e.g., users need both wireless connection and storage to post a photo online) The author models the resource allocation process of a mobile cloud computing system as a public sale with best and concession options.

The auction mechanism is given to the service provider. The authors indicated some numerical results and have reached to the changing in the service provider's revenue, user's utilities to user's types, the maximal resource allocation schemes for the service provider, and the effect of user's premium and discount options to the users' utilities and the service provider's revenue.

(Choksi, 2014): The evolution of cloud computing, it has changed the corporate and educational industry. Cloud computing is effectively convenient, cost effective, and on demand service that is proposed to users. From this point of view, there are several security concerns when utilizing cloud services. Security is very important in cloud computing ever since individuals and companies have had their private information stored in cloud. Many approaches for authentication in cloud services have been presented, which are classified to insecure, intricate, or highly expensive. The author presented a study to compare different authentication schemes in cloud and explained the original different evaluation criteria.

Hence, the author has proposed various authentication techniques for cloud computing. Authenticating and identifying cloud users are getting more focus. The author has concluded that the proposed cloud schemes lack resistance to attacks and unauthorized accesses. Meanwhile, none of the techniques fully satisfies the criteria of the evaluation.

(Vijayalakshmi & Arunapriya, 2014): In this paper, the authors focused on the authentication of data storage using decentralized access controller in clouds, in which individual authorized users can access stored information. The authors aim is to identify if a user is secured from the cloud during verification, the cloud architecture is devolved, the access control data and verification are collision impervious, preventing revoked users from

accessing after being revoked, and they system is capable to replay to attacks. The scheme supports creating, changing, and understanding the saved data in the cloud and offers decentralized verification and robust as well.

Access control is very important when unauthorized users try to access the data from the storage, it is also important to identify the source of information. The solution to overcome this problem is access control, verification and confidentiality defense through deploying suitable encryption methodologies such as User-Based Access Control (UBAC), Role-Based Access Control (RBAC), and Attribute-Based Access Control (ABAC).

The system offers user cancellation and eliminates the repeated attacks. The cloud system cannot classify the identity of the user who store the information, but once and only confirms the user's identifications.

(Choudhury & Abudin, 2014): The main concept of cloud computing is having several computers interconnected over a real time network like the internet. Cloud computing is a group of IT services that are presented to users over a network with the ability to have their important requirements. Cloud computing offers the facility to share different properties and services that are related to different organizations or sites. The author proposed a security

technique that depends on Encrypted One Time Password (EOTP). One time password is encrypted by a public key of user in order to get EOTP.

This paper explained cloud-computing security as becoming essential requirements nowadays. The author used public key cryptography (RAS algorithm) to encrypt and decrypt one time password. In the proposed system when a one-time encrypted password is encrypted it is then sent directly to the user through the network. Third party such as Global System for Mobile (GSM) mobile number or email is not needed. The authors' aim is to enhance the system security, efficiency and to eliminate dependency on third party. The system is highly protected and dependable.

(Alkury, 2014) This thesis was interesting about the cloud computing subject, which presented different services to the number of users. With cloud computing offering a good storage that when dealing with it from different times and locations, because the information that stored on it does not need any space and do not need transferring from one place to another.

The storage service of information is presented in three forms: The first form is related to the user and can be accessed only by the same user. The second form is mutual between the individuals and the organizations that cannot access to this information except by the members of the organization. The third form is related only to the organizations and all the members can access it.

although there are great benefits of cloud computing, but it is exposed to huge risks and the main important risk is the intruder, because of this it should provide the security to all types of storage and decrease the risks to encourage individuals to use this technology.

There are many factors of the security, from the perspective author in this thesis; there are two important points to achieve the security: the first point is avoiding the denial of service by monitoring the system and the second point is determining which users can access to the cloud and select the user privilege from the admin. These two points are the main factors to provide two directions: the first direction focused on security and privacy and the second direction focused on availability and performance.

(Al-Hamami & AL-Juneidi, 2015) This thesis explains the Mobile Cloud Computing (MCC) which refers to the availability of Cloud Computing (CC) services in a mobile environment and it is the combination of the heterogeneous fields like mobile phone device, cloud computing and wireless networks.

Nowadays the term MCC has become the buzzword and a major discussion thread in the IT world, Because new technology brings new threats, the security issue is the most important problem that the cloud computing technology has brought, especially the issue of authentication or

identification, and how to provide safe technological environment for both companies and individuals in order to securely use MCC and to create a kind of trust between providers and users of this technology.

this thesis has designed a new efficient model for mobile cloud computing based on fingerprint, the implemented model works on storage all the user's fingerprints with their password on cloud server, so when they want to access the cloud computing through mobile phone device they must scan any one of their fingerprints and its password. After the experiment, the results in this thesis is summarized in: the Applicable security is excellent because of using many passwords, the flexibility in the use of any fingerprint to prove personal authentication. finally, the intruders will not be able to take advantage of mobile cloud computing service because of the use of several layers of security.

(Shetty & et al, 2015): The authors presented their attempts to produce a secure access to the cloud services, while highlighting authentication, confidentiality and data integrity.

Cloud computing is an aggregation of technologies such as network, data storage, operating systems and virtualization with inherent security problems like data theft and infrastructure misuse. Due to these security problems, sectors are interested to enhance their security system such as banking

defense, healthcare and finance and because of their security concerns; they are hesitant to utilize cloud services and hence are eliminated of its benefits.

In this paper, the authors provided extensive prototype software that can provide a secure access to cloud services. Even though the file size in cloud computing increases by 30.6122%, at the end a processing time of 0.4392s/MB is needed for the encryption, hence, increasing the transmission time.

(Ye & et al., 2015) This paper proposes a mobile face identify authentication system. In this system, Android application has to capture the face, and verify the face by Web services. It is introduced how to implement an Android Client of this system in details. Using MB-LBP features, Ada Boost and Cam Shift algorithm, get face images by camera on mobile device. Then, rotate, the crop the face images and convert them to grayscale in order to reduce the amount of face data. At last, post data and get validation results using sub-thread to realize real-time face verification. It is introduced the implementation of mobile face identity authentication system on android platforms. On Android, they complete three main functions including real-time face capture, interaction with server and show obtained result on UI. In the implementing of face capture, they improved the speed and correct rate of face detection using Adaboost learning algorithm, MB-LBP operator and Cam Shift tracking algorithm. Then normalize the facial image by eye

locations to reduce the size of communication data. At last, using multithread, make the Android application capture new facial image and request face verification service at the same time. Thus, the face verification is close to real-time in a good network condition. In practice, the solution is implementation running on Note3 can reach 30fps of face detection, and its correct rate is above 90%. In a good network condition, the cost time of verifying a person is about 2 seconds on average.

2.3 Summary

Cloud computing is transporting the data and services to large data centers, in which the control of both data and database is a major issue and not trustworthy. On the other hand, cloud computing is becoming enormously attractive, and even with the professionals different overviews and thoughts of the reasons behind the success of cloud computing, it is clear that the system is emerging as a significant driver in the IT market place.

In this thesis, depending on the major problem behind security vulnerabilities in cloud computing system, an authentication technique has been developed to reduce the unauthorized access to user's data. Basically, the thesis has put the light on the security of data and database of cloud computing through the authentication technique used which is the "biometric face recognition technique".

Meanwhile, other thesis have discussed how to increase the number of users that use cloud computing, and enhancing the QoS while other researchers focused on the security in which how to retrieve users stored data in the system.

In this thesis, Cloud Computing data security is considered a zone that is rich with challenges and of privilege importance, therefore, several authors problems are yet to be proposed. The thesis enlightens several potential measurements for future research on this area.

The security issue is explained in a respective manner in which identifying a users ID (authentication) through mobile devices to the system, and to enhance the dependability of both mobile and cloud security system. The authentication technique that is being used will be presented in details in the following chapter.

CHAPTER THREE

Theoretical Design

3.1 Introduction

Mobile computing has become a powerful trend in the field of IT technology. Cloud computing is a developing service model allowing mobile devices to decrease the energy feastings and to execute different services on the clouds distantly. Thus, a mixture of electronic devices like smart phones, PDA, tablets, pervasive mobile network and cloud computing, resources are joining together to appear as a new field of MCC.

cloud computing is one of the recent technologies in the field of development information technology, as known any new technology includes many benefits, at the same time, it is accompanied with many risk.

The mobile device is consider one of the methods that used for access to the cloud computing. However, the mobile cloud needs to provide a secure communication between cloud and user. The security services in any communication include access control, authentication, non-repudiation, authorization service to mobile user and so on.

This thesis, is an attempt to provide mechanism to reduce any risk that may occur in communication through the transfer data or information between

user and cloud. The proposed mechanism takes advantage of multi authentication factors in which it merges inherence factor (something user is or does e.g., fingerprint) with knowledge factor (something user knows, e.g., username and password), so this mechanism will increase the overall security level of the system in order to provide users with maximum level of security for their data on the cloud.

The applied mechanism includes the following stages, as shown in Figure (3):

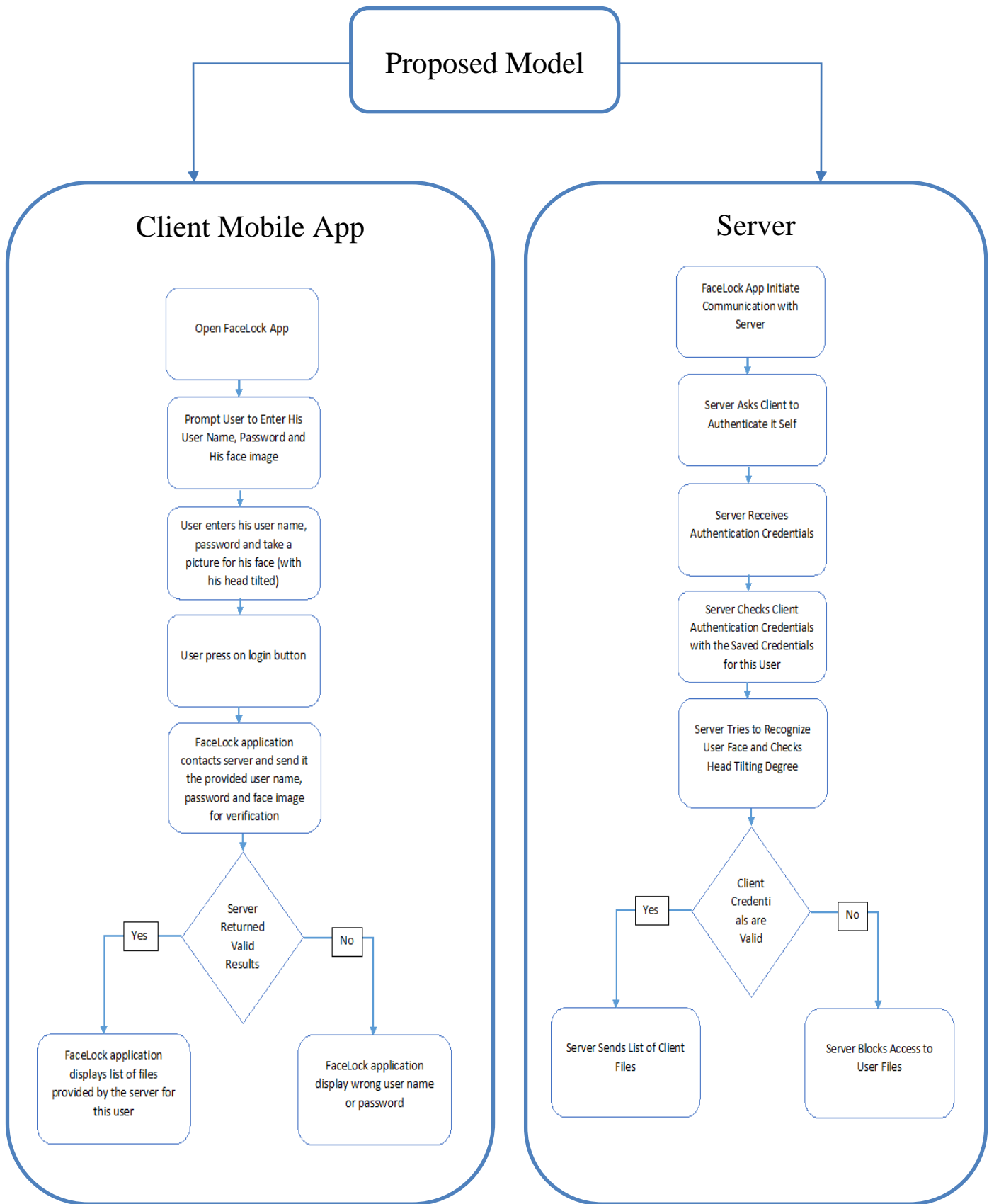


Figure (3): Proposed model

Stage 1: Collecting user information (Client):

The first stage takes place at the users side; the user could be at anywhere while connecting to the cloud as long as he/she has Internet connection. At this stage there are two authentication factors for which user information has collected, the first factor is inherence factor that authenticates user through his/ her biometrics properties, and in this model we will use face recognition in which the user takes a picture for his/her face, while this picture will be used to validate user identity.

The second authentication factor is the knowledge factor, for this factor the method uses User Name, Password and the degree of tilting user face in the picture, User name and password are entered by the user, the tilting degree is calculated from the picture that the user has taken. The collecting stage includes three part of information:

1. User Name
2. Password
3. Picture of the user face.

Stage2: Validating user information:

The second stage takes place at server side (Cloud Server), after collecting user information (User name, password and user face picture), user application connects to the cloud server securely (through HTTPS connection) and sends the authentication information, and the cloud server checks this information as follows:

- User name checking: the server checks if the sent user name is valid by comparing it with the registered users in its local DB.
- Password checking: the server checks if the password belongs to the registered user from the previous step.
- Face recognition: the server performs face recognition for the received picture and tries to identify the person in it, and if the person identified as the same person with the user name and password sent before then this validation step is successful, the user can continue the process otherwise the validation step will fail.
- Face tilting checking: After validating the users identity, the face image is processed by the server to calculate the tilting degree of the head.
- The server validation stage main check is the merged knowledge-inherence authentication factor; this authentication factor is composed of the following stages:
 - ✓ Face detection: in this stage the server detects if the sent picture has any faces in it and identifies their regions.
 - ✓ Face recognition: in this stage the server tries to recognize the face that was detected in the previous stage.
 - ✓ Eye detection: in this stage the server detects the eyes in the face as a preparation step for the next stage.

- ✓ Face tilting measure: the server calculates the tilting degree of the eyes line in order to determine the face tilt degree.

3.2 Face detection procedure

Face detection is performed using Haar-like feature cascade classifier, and in order for this classifier to detect faces objects, this classifier is first trained with a few hundred sample face images which are called positive samples, and some negative examples of other objects. all these images are scaled to a particular size for example 20x20.

The classifier extracts features from the image using Haar-features as shown in Figure (4).

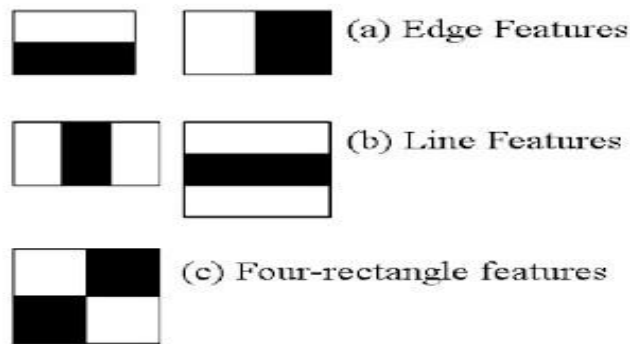


Figure (4): The classifier extraction features (Viola & Jones, 2004)

Each feature has one value; this value is calculated by subtracting pixels values under the white rectangle from pixels values under the black rectangle, as shown in Figure (5).



Figure (5): calculated by subtracting pixel (Viola & Jones, 2004)

The course training procedure includes two sorts of exchange offs. As a rule classifiers with more highlights will accomplish higher identification rates and lower false positive rates. In the meantime classifiers with more highlights oblige of an opportunity time to register. On a fundamental level one could characterize an improvement system in which: i) the quantity of classifier stages, ii) the quantity of highlights in every stage, and iii) the limit of every stage, are exchanged off while keeping in mind the end goal to minimize the normal number of assessed highlights. Shockingly discovering this ideal is a massively troublesome issue (Viola & Jones, 2004).

Practically speaking an exceptionally basic structure is utilized to deliver a compelling classifier that is very effective. Every stage in the course lessens the false positive rate and declines the location rate. A target is chosen for the base diminishment in false positives and the greatest lessening in location. Every stage is prepared by including highlights until the target location and false positive rates are met (these rates are controlled by testing

the indicator on an acceptance set). Stages are included until the general focus for false positive and identification rate is met (Viola & Jones, 2004). After training the classifier can be used to determine if an input image is showing a face or not, the classifier traverse a window of the size 20x20 and test if the region inside this window is showing a face or not. In this model used Open CV is used as an implementation of the Haar-feature classifier (Viola & Jones, 2004).

3.3 Face recognition procedure

Sirovich and Kirby developed Eigen Principal Component Analysis (PCA) in 1987 that is used for face recognition and reused by Matthew Turk and Alex Pentland in face classification. The method is explained as the following (Wilson & Fernandez, 2006):

1. Set up a preparation fixed number of face pictures. The pictures establishing the preparation set ought to have been captured under the similar illumination circumstances, and must be standardized to have the eyes and mouths adjusted over all pictures. They should likewise be all re-sampled to a typical pixel determination (row * column). Every picture is dealt with as one vector, just by linking the columns of pixels in the first picture, bringing about a solitary column with (row * column) components. Due to this execution, it is expected that all pictures of the preparation set are put away in a solitary matrix (T), where every segment of the framework is a picture Deduct the mean. The average image has to be calculated and then subtracted from each original image in T.
2. Compute the eigenvectors and Eigen values of the covariance framework. Every eigenvector has the same dimensionality number of segments (S) as the first pictures, and in this manner can itself be seen as a picture. The eigenvectors of this covariance framework are thusly

called Eigen faces. They are the headings in which the pictures contrast from the mean picture. Typically this will be a computationally lavish step (if at all conceivable), yet the reasonable appropriateness of Eigen confronts comes from the likelihood to register the eigenvectors of S proficiently, while never figuring S expressly, as point-by-point underneath.

3. Pick the essential parts. Sort the Eigen values in diving arrange and orchestrate eigenvectors as needs be. The quantity of rule parts k is resolved self-assertively by setting an edge ϵ on the total variance (v). Complete difference $v = n(\lambda_1 + \lambda_2 + \dots + \lambda_n)$, n= number of information pi.
4. Number of principle (k) is the smallest number fulfills:

$$\frac{n(\lambda_1 + \lambda_2 + \dots + \lambda_k)}{v} > \epsilon$$

These Eigen appearances can now be utilized to speak to both existing and new confronts

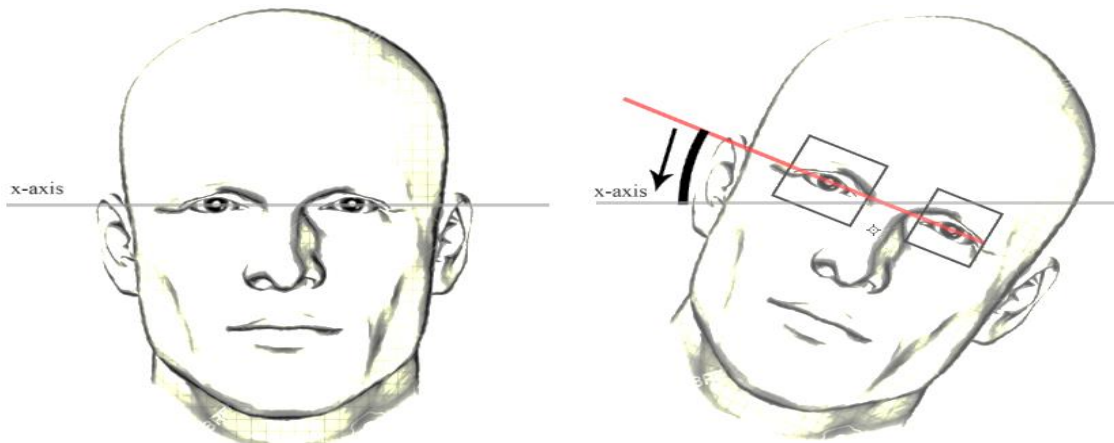
The extended is another (mean-subtracted) picture on the Eigen countenances and consequently record how that new face varies from the mean face. The Eigen values connected with every Eigen face show how many pictures in the preparation set shift from the mean picture are in that course. They lose data by anticipating the picture on a subset of the eigenvectors, yet minimize this misfortune by keeping those Eigen faces with the biggest Eigen values. For example, on the off chance that we are working with a 100 x 100 picture, then we will acquire 10,000 eigenvectors. In functional applications, most faces can regularly be recognized utilizing a projection on somewhere around 100 and 150 Eigen confronts, so that the greater part of the 10,000 eigenvectors can be disposed of. Open CV has an execution of the Eigen PCA, which was utilized as a part of this model to perceive faces (Saha & Bhattacharjee, 2013).

3.4 Eye detection procedure

The same Haar-like highlight course classifier was utilized to distinguish eye pictures however with an alternate preparing set. In this model we utilized Open CV as an execution of the Haar-highlight classifier (Open CV, 2015).

3.5 Face tilting measurement procedure

So as to discover the tilting degree the server distinguishes the head locale and afterward recognizes eyes district, then the tilting degree is the degree



between the x-axis and the eyes line, tilting client face degree figuring is portrayed in the figures beneath, as shown in Figure (6):

Figure (6): face degree (Open CV, 2015)

3.6 User authentication flowchart:

this part presents the authenticate process, as shown in Figure (7)

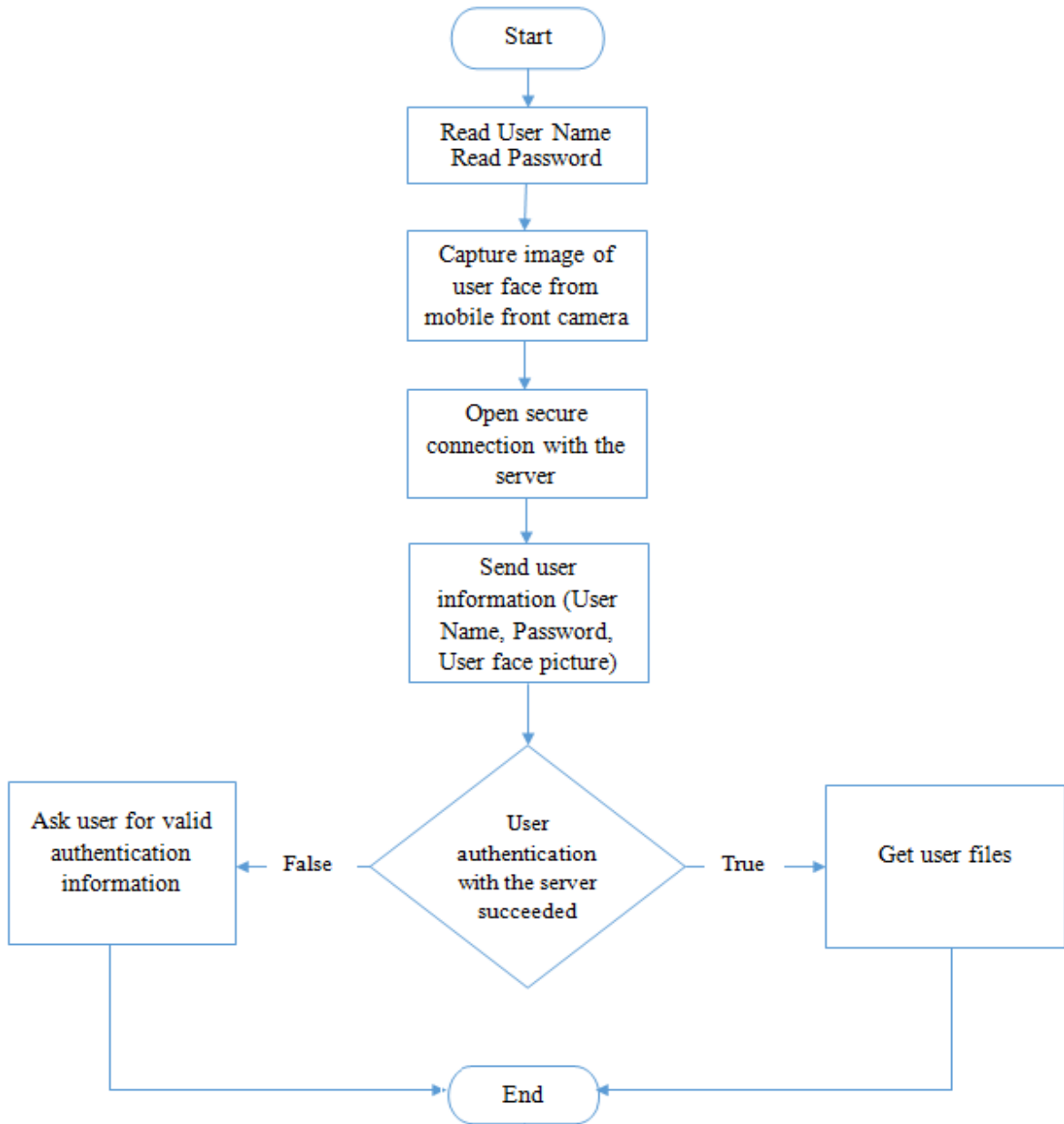


Figure (7): User authentication flowchart

- **Algorithm**

// Input: User Name, Password and User Face Picture

Output: Authentication result and if authenticated -> User Profile and User files//

- Step1: IF (User Name != "" And Password != "") Then
- Step2 : For I:0:RegisteredUser.Length-1
- Step3: IF (Registered User[I].User Name == User Name) Then
- Step4 : IF (Registered User[I]. Password == Password) Then
- Step5 : Img ← User Face Picture
- Step6 : Img640 ← ResizeImageto640x480(Img)
- Step7 : Img Gray ← Convert To Gray (Img640)
- //Initialize faces& eye detectors
- Step8 : Face Haar Cascade ← new Face Haar Cascade
("FacesHaar.xml")
- Step9 : Eye Haar Cascade ← new Eye Haar Cascade
("EyesHaar.xml")
- //Detect Eyes
- Step10: Eyes Detected Rects ← Eye Haar Cascade (ImgGray)
- Step11: IF (Eyes Detected Rects != null) Then
- Step12: Rotation Angle ← Eye Line Rotation Over
XAxis(EyesDetectedRects)

- Step13: IF(Rotation Angle == 30) Then
- Step14: Rotated Face ← Rotate(Img Gray, -1 * Rotation Angle)
- Step15: Detected Face ← Detect Face (Rotated Face)
- Step16: Recognized User ← Recognize User(Detected Face)
- Step17: IF (Recognized User != null)
- Step18: IF(Recognized User .Name == User Name)
- Step19 : Send Prepare User Files(Recognized User. Name)
- Step20: return True
- Step21: ELSE // IF(Recognized User. Name == User Name)
- Step22 : return false
- Step23: END IF // IF(Recognized User. Name == User Name)
- Step24: ELSE //IF (Recognized User != null)
- Step25: return false
- Step26: END IF //IF (Recognized User != null)
- Step27: ELSE // IF(Rotation Angle == 30) Then
- Step28: return False
- Step29: END IF // IF(Rotation Angle == 30) Then
- Step30: ELSE // IF (Eyes Detected Rects != null) Then
- Step31: return False
- Step32: END IF// IF (Eyes Detected Rects != null) Then
- Step33: ELSE //IF (Registered User[I].Password == Password) Then

- Step34: return False
- Step35: END IF// IF (Registered User[I].Password == Password)
Then
- Step36: ELSE // IF (Registered User[I].User Name == User Name)
Then
- Step37: return False
- Step38: END IF// IF (Registered User[I].User Name == User Name)
Then
- Step39: Next
- Step40: ELSE // IF (User Name != "" And Password != "") Then
- Step41: return False
- Step42: END IF// IF (User Name != "" And Password != "") Then

3.7 Prepare Image:

In order for the face and eye detection algorithms to work the image should be resized to 640x480 and converted to gray scale this is mandatory for faster image processing, as shown in Figure(8).

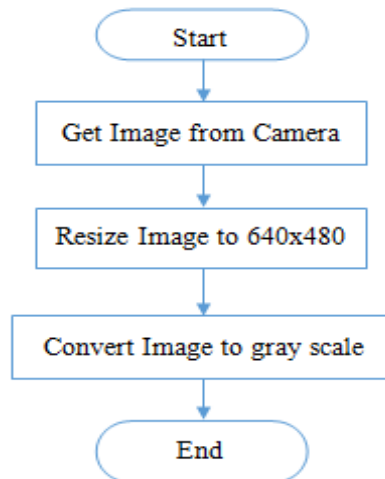


Figure (8): Prepare Image flowchart

- **Algorithm**

- Step1 : $Img \leftarrow UserFacePicture$
- Step2 : $Img640 \leftarrow ResizeImageTo640x480(Img)$
- Step3 : $ImgGray \leftarrow ConvertToGray(Img640)$

3.8 Eyes Detection:

The prepared image for the user will be processed with Haar Cascade to detect eyes location in order to extract head tilting degree, prior to processing stage the Haar Cascade will be initialized with pre-trained set, as shown in Figure (9).

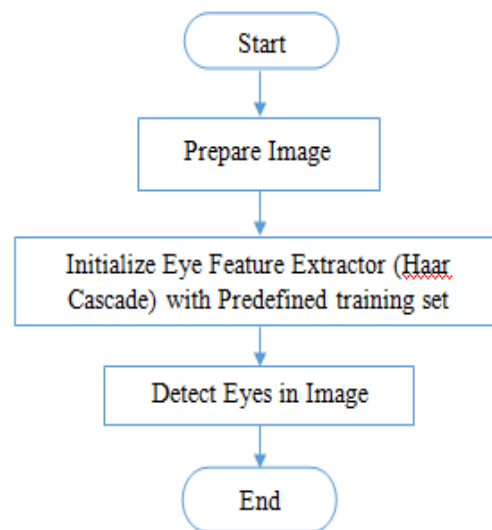


Figure (9): Eyes detection flowchart

- **Algorithm**
- Step1 : EyeHaarCascade \leftarrow new EyeHaarCascade("EyesHaar.xml")
- Step2: EyesDetectedRects \leftarrow EyeHaarCascade(ImgGray)

3.9 Face Detection:

After detecting eyes location the user face will be processed to recognize user, the first stage in face recognition is face detection which is performed with Haar Cascade that was initialized with pre-trained face detection set, after detecting face region the face region image will be sent to face recognizer that will recognize user face, as shown in Figure (10).

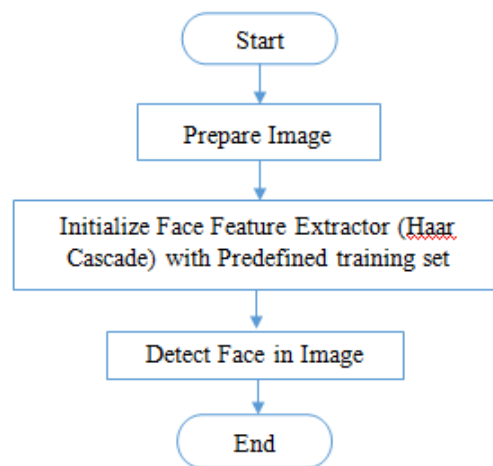


Figure (10): Face detection flowchart

- **Algorithm**
- Step1 : FaceHaarCascade \leftarrow new
FaceHaarCascade("FacesHaar.xml")
- Step2: DetectedFace \leftarrow DetectFace(Image)

3.10 Eyes Line Slope (face Rotation):

After detecting eye and prior to face recognition the users face will be rotated to ensure best results, the face tilting degree is calculated from the detected eyes rectangles, as shown in Figure (11)

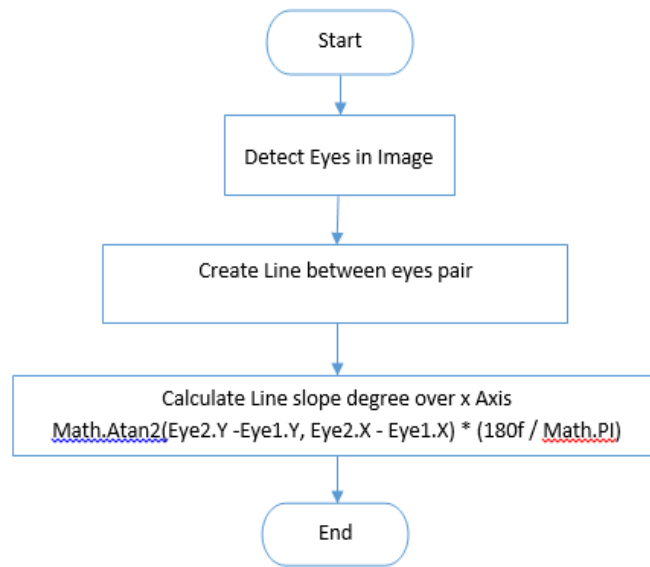


Figure (11): Eyes Line Slope (Rotation) flowchart

- Algorithm

- Step1 : $\text{eyeRects} \leftarrow \text{EyeDetection}(\text{Image})$
- Step2 : $\text{Rectangle R1} \leftarrow \text{eyeRects}[0].\text{rect}$
- Step3 : $\text{Rectangle R2} \leftarrow \text{eyeRects}[1].\text{rect}$
- Step4 : $\text{PointF P1} \leftarrow \text{new PointF}(\text{R1.X} + \text{R1.Width} / 2f, \text{R1.Y} + \text{R1.Height} / 2f)$
- Step5 : $\text{PointF P2} \leftarrow \text{new PointF}(\text{R2.X} + \text{R2.Width} / 2f, \text{R2.Y} + \text{R2.Height} / 2f)$
- Step6 : $\text{LineSegment2DF line} \leftarrow \text{new LineSegment2DF}(\text{P1}, \text{P2})$

- Step7 :double deltaY ← line.P2.Y - line.P1.Y
- Step8 :double deltaX ← line.P2.X - line.P1.X
- Step9 :IF (deltaX != 0)
 - //Atan2: the angle whose tangent is the quotient of two specified numbers
- Step10 :angle ← Math.Atan2(deltaY, deltaX) * (180f / Math.PI)
- Step11 :ELSE //IF (deltaX != 0)
- Step12 :angle ← 90
- Step13 : END IF //IF (deltaX != 0)

3.11 Summary:

Part three examined the suitable system for proposed issue, and announced the proposed model through algorithms and flowcharts. Each of them clarifies one of the parts in the model. Additionally this section concentrated on the model goes as it for tolerating accomplished results.

CHAPTER FOUR

The Experimental Works

4.1 Introduction

Storing data in the cloud has many benefits one of them is the ability for users to access their private information securely from any device connected to the Internet. The proposed security model insures secure login by users to access private information stored on the cloud from user mobile device, this secure logging mechanism relies on user biometric (face image) along with traditional user name and password, these factors are sent to the cloud server for validation.

The user biometric factor is validated in two aspects; the first aspect is face recognition in which the user identity is validated. The second aspect is the head tilting degree, the user tilts his/her head to the left by 30 degrees before sending his/her face picture to the server, the server checks picture and recognizes the user face and then it checks if the tilting degree is 30 degrees to grant user access permission.

In this model the user will use his mobile to access his private information on the cloud server, the model consists of user mobile program and a cloud server, which are both, connected to the Internet.

In this model Android devices were targeted for building test application that will be used to authenticate users through biometric and informative measures. Android devices were targeted because of its wide and vast spread.

The model was implemented on an Android device (for the user part) and a cloud server that is hosted on Windows machine (for the cloud server part). The user starts the program on his/her Android device to access the information stored at the cloud server. After starting the program the user will be asked to enter username, password and takes a picture for his/her face. This information will be sent to the server for validation to determine if the user is allowed to access the information stored on the server. The server will check if the user is a registered user and then checks user's password.

The last step will be on the user face picture; the server will detect the face in the picture and will recognize it and checks if the face in the picture belongs to the user and whether the user face is tilted by 30 degrees.

Based on the above model there are two main parts the client (Android App) and the server (Cloud Server).

- The client sends authentication data consisting of username, password and face picture to the server and after successfully authenticating his/her identity by the server the program permits user to download his/her private files from the cloud.
- The server receives the authentication data and validates it with the saved client information stored in its local DB and checks the face picture to check client identity and then checks the head-tilting angle.

The whole process is demonstrated in the Figure (12).

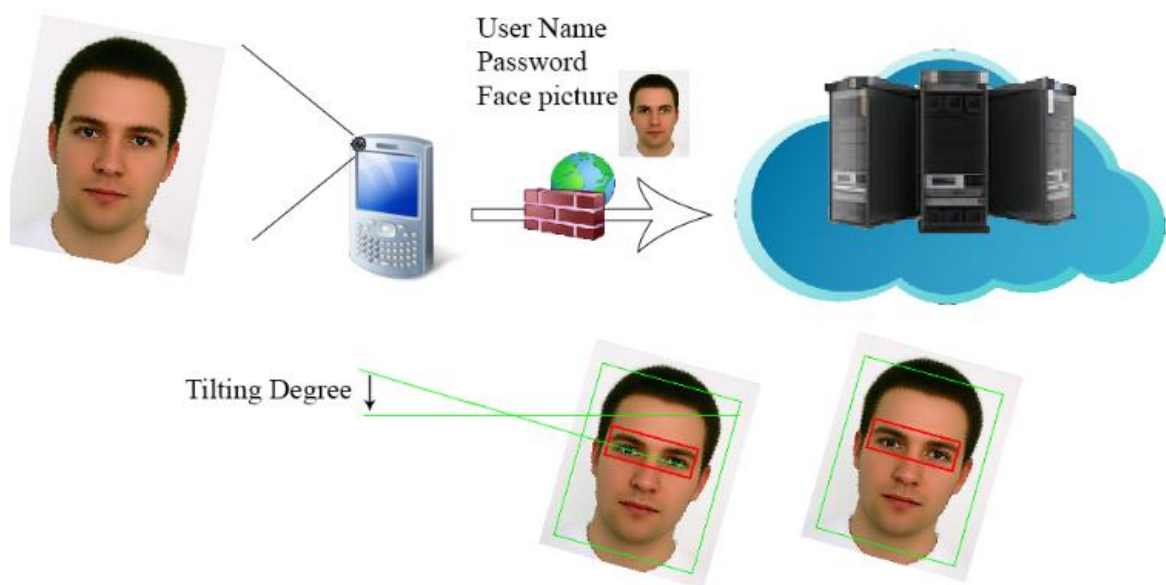


Figure (12): Model process

4.2 Client (Android App)

The client can use any Android device to connect to the cloud and accesses the stored information, the client runs the Android mobile application that is called “Face Lock”, and the GUI for the application is shown in Figure (13).

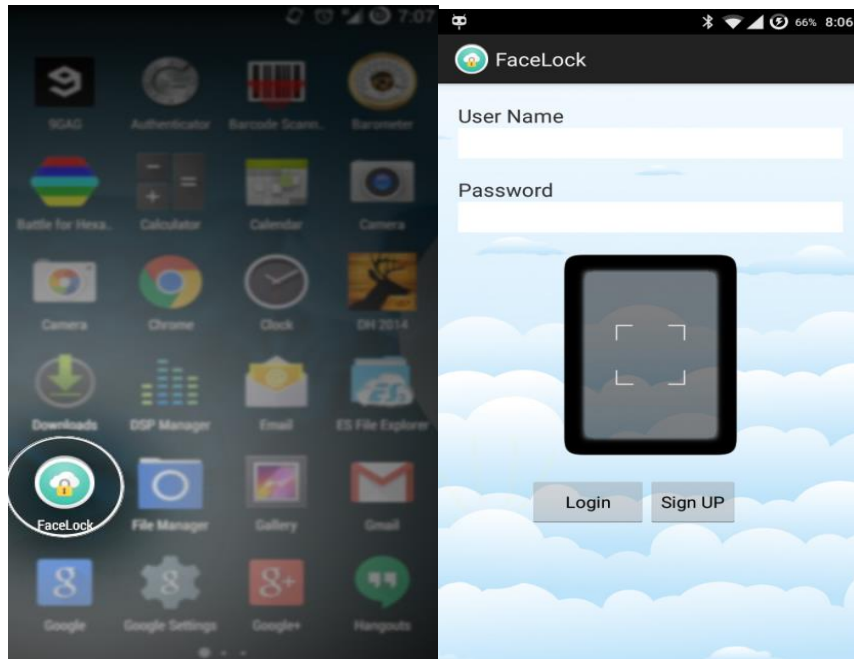


Figure (13): GUI for the application

The application has two main functionalities. for user authentication, User Registration that is done through the Sign UP process and User Authentication, which is done through the Login process. New users must register themselves before they are able to access the cloud server. After registration, users can access the cloud server by proving their login credentials including user name, password and face picture.

4.2.1 Registration

In order to allow new users to use the system, a registration module was added to collect user information and login credentials. these credentials also known as knowledge factors and consists of two parts:

1. Something user knows which is the user name, password and face tilting degree.
2. Something user is which is a picture of his/her face.

New users open the application “Face Lock” and then press on “Sign Up” button, the registration process screen appears as shown in Figure (14).

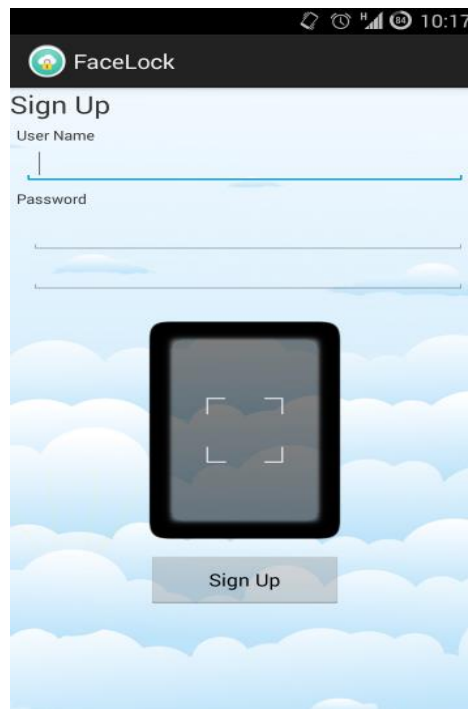


Figure (14): Registration process

The user should enter a valid unique user name that consists of minimum 6 letters with no spaces and a valid password that consists of minimum 6 letters with numbers, symbols and letters in it.

After entering user credentials the user should provide a picture for his/her face taken by the front camera of client device, the user should tilt his/her face for increased security, so that the server will check the tilting angle of the face and prevent access if the tilting angle is different (in case of unauthorized access be user picture).

This screen collects user information and sends it to the cloud server. if the user name is available (not taken by any other user) the cloud server processes the user face picture and detects the tilting angle of the face and stores this information in the server DB for future use in user credentials validation.

4.2.2 Login

After registration users can use “Face Lock” app to access their private files at the cloud server using their credentials what they know and what they are.

The credentials are entered through the Login screen as shown in Figure (15).

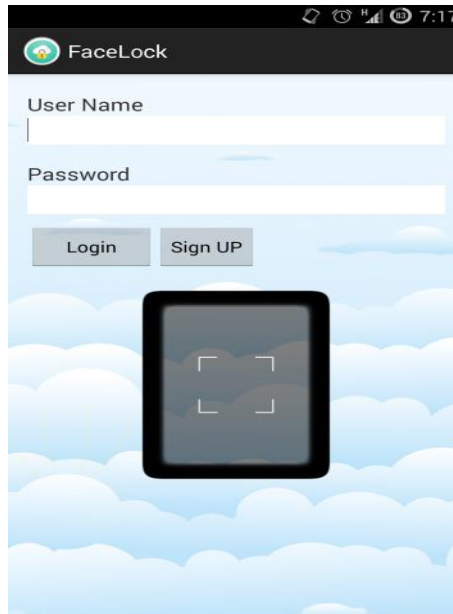


Figure (15): Login screen

The user should enter his/her user name and password and then click on the black rectangle to take a picture with the smart phone front camera, users should keep in mind to tilt their head by the same degree they tilted it at the registration process.

After taking the picture the user should press on the login button, and the application will communicate with the cloud server and send to it the user credentials along with the user face picture.

If the credentials provided by the user are correct and if the face picture is a picture of his/her face and the head tilting of the face in the picture is correct then the login process completes successfully and the user is granted login permission by the server that will last until the user closes the application.

And if the credentials are wrong then the user will not be granted login permission and the cloud server files will not be visible to her/ him.

4.2.3 Login Trials

4.2.3.1 Authorized Login

In this scenario we will try to log into the system with a valid user name and password and with a valid image with correct tilting angle as shown in Figure (16).

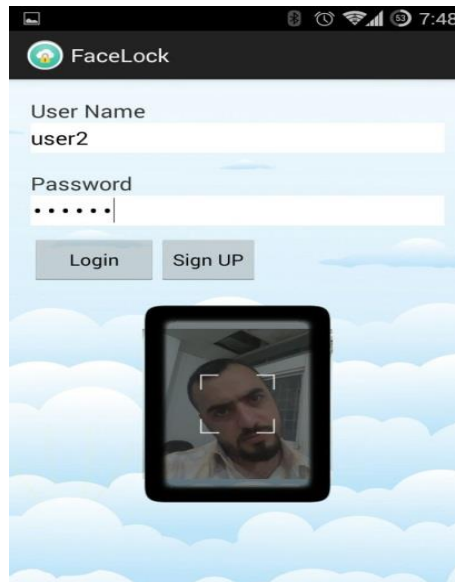


Figure (16): Authorized login

After correct authorizing access by server the user can access secret files as shown in Figure (17).

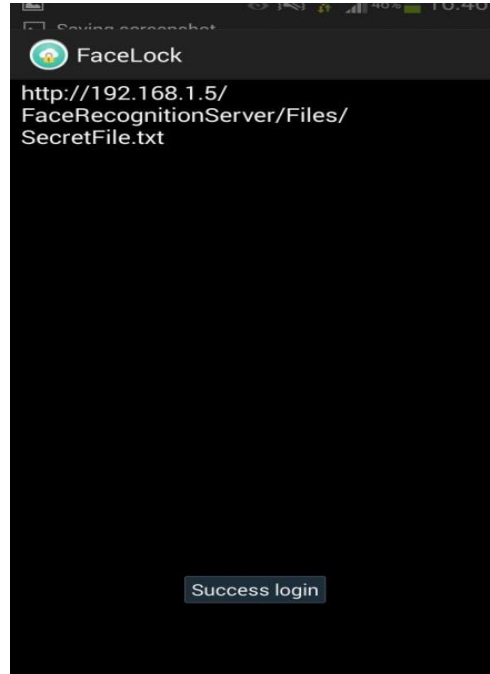


Figure (17): Authorized access

4.2.3.2 Unauthorized Login

In this scenario we will try to log into the system as described below:

1. **Invalid credentials:** in this trial we will use user as an invalid user name with the password 123456 as an invalid password along with an invalid user image, in this situation the system will send the user name and password to the server for validation, then the server will decline the login and prevent accessing secret files.

The mobile application will then display a failure login message as shown in Figure (18).



Figure (18): Unauthorized login

2. Valid credentials (stolen user name and password) with invalid user image as shown in Figure (19)

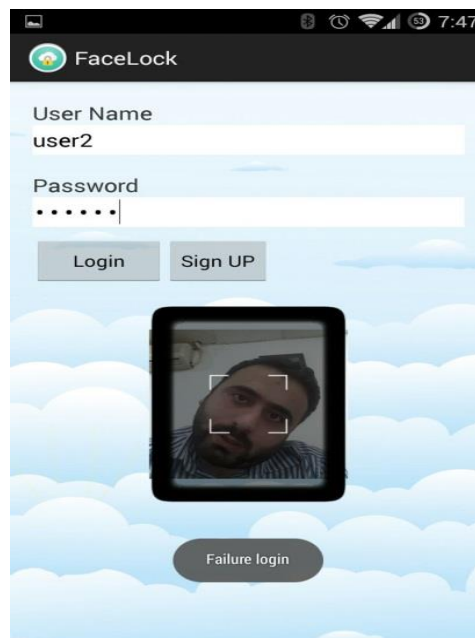


Figure (19): Invalid user

3. Valid credentials with valid image but with invalid head tilting angle as shown in Figure (20)

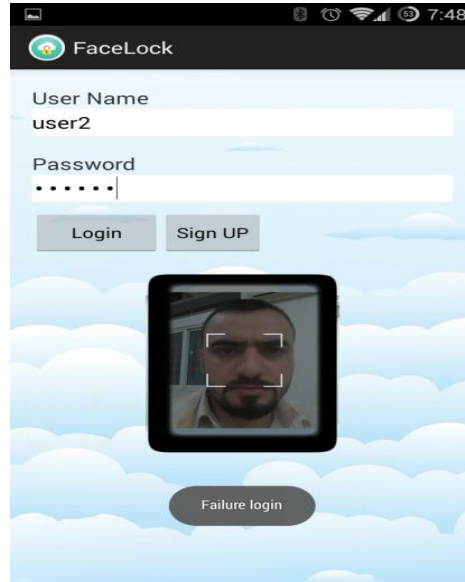


Figure (20): Invalid angle

4.2.4 Accessing Cloud Server Files

After user authenticates himself/herself to the server, user can access the private information stored at the server. The server has a separate storing location for each user, these locations are separate and no user can access information of another user, and it is only for the authorized users.

Following is how user can access the secure information stored at the server after authentication, as shown in Figure (21).

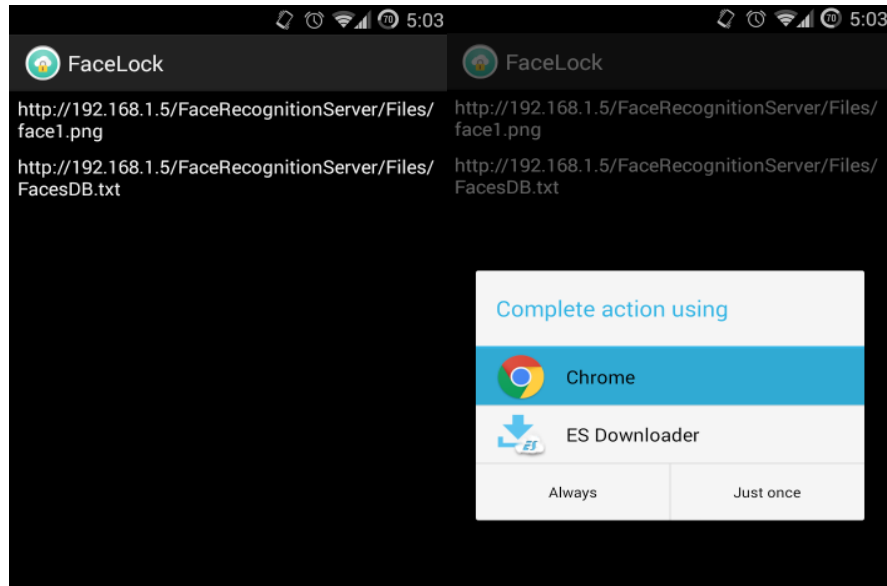

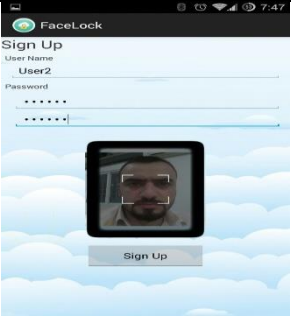
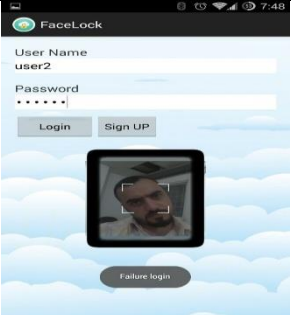


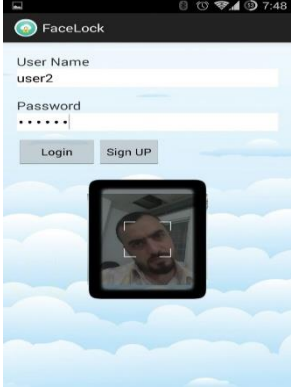



Figure (21): Information stored at the server




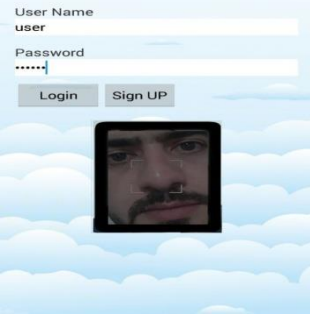
4.3 Summary

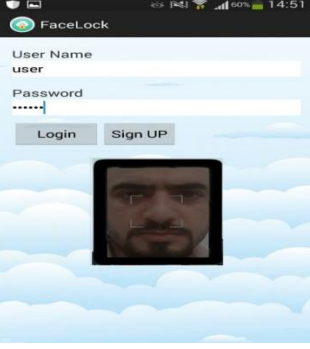


After making the experiment on random number of people the following results were taken to clarify the efficiency of the program and what has been reached in accordance with the following table (1):

Table (1): Table of the result

Image	Recognition Time	Face Detected	Recognized	Tilting Degree	Result
	243 ms	Yes	Yes	12	Pass
	252 ms	Yes	Yes	1	Failed
	110 ms	No	No	Nan	Failed

	238 ms	Yes	Yes	16	Pass
	268 ms	Yes	Yes	14	Failed
	102 ms	No	No	Nan	Failed
	207 ms	Yes	Yes	1	Failed

	113 ms	No	No	Nan	Failed
	50 ms	No	No	Nan	Failed
	160 ms	No	No	Nan	Failed
	164 ms	No	No	Nan	Failed

	200 ms	Yes	No	Nan	Failed
	208 ms	Yes	Yes	11	Pass
	240 ms	Yes	Yes	2	Failed

CHAPTER FIVE

Conclusions and future work

5.1 Introduction

This thesis has summarized the important points, and has suggested some ideas for future works.

Cloud computing offers important goals with the shift of deploying a new platform from regular PC towards smart phones which offers a richer behavior of data and a variety of options to its users such as the on demand delivery of IT resources and many applications on the Internet that are available to Cloud users. As cloud users try to access the cloud, their data will start to be used for other applications such as advertising, and utilizing the cloud's data aggregation facility. Through this concept, cloud computing has brought the challenges and opportunities for authentications as it has become a perfect fit for all kinds of mobile security while traditional computations has limitations. The aim is to build a system that is capable to look at face geometry and recognize the authenticated user rather than a user telling the system its identity.

5.2 Conclusions

The term cloud computing refers to the consumption of computer resources which includes deploying several remote servers and networks that permit centralized data storage and access to the services that are provided by computer. Originally, Cloud computing describes a large groups of objects that are visible from a distance while refereeing the on-demand delivery of IT resources and several applications on the internet in which the delivery of hosted services are over the internet. Cloud computing depends on the idea of sharing resources instead of having limited servers or personal devices to process applications.

Cloud computing goal is to give the opportunity to businesses to expend computer possessions as a utility instead of having to construct and conserve computing infrastructure in house, and to apply traditional supercomputing or high performance computing power.

Despite the increase of worrying about security in cloud computing, cloud security concentrates on data privacy and safety that most organizations mean to overcome.

Meanwhile, cloud computing as any system in the world requires a defense system to prevent or limit the unauthorized access to the data and resources,

in which having a highly secured system that protects the core data of the cloud.

Verification is all about creating the character of one or two parties in a dialogue or session as data in cloud computing is considered to be an area that is full of challenges. A high technique is used to authenticate a users identity which is the biometric face recognition, and this mentioned authentication mechanism builds a trust between the user and the cloud in which it will increase the safety, confidence, accessibility and performance of the cloud to produce a resolution for the problem of cloud security, to as will as guarantee the secure access to restricted data/services in the cloud using a mobile phone. Also, it will facilitate the work of people who use the mobile.

To sum up with, the authentication system was developed through the Mobile Android system that served us with solutions and ideas to overcome the access issue and limit the secured access to data, storage and computer resources.

As new technology is evolving, this could lead to deploying new authentication method through getting the advantage of newly developed features in smart phones for instance using the front camera to identify users by capturing a high resolution picture that authenticates the identity.

5.3 Future Work

In this thesis, a collection of security issues was presented and the main solutions to overcome them were investigated such as biometric authentication. As can be derived from the research, an excellent beginning idea for enhancing cloud security includes empowering the security abilities of both web browsers and web service frameworks. Hence, as part of the thesis work, development will take place in carrying on hardening the foundation of cloud computing security, which indicates the following:

1. Deploying the face recognition technique on several operating systems not only the android, such as Windows and IOS.
2. Furthermore, along with the face recognition technique, the thesis future aim is to expand the idea of face recognition to include scanning the whole face features and not only 30 degree of the face which could be more secured and safer.
3. Meanwhile, conducting another biometric technique, such as the fingerprint and voice recognition in which smart phones nowadays support these features.
4. Developing a biometric device is an important step to be considered in the future work, as its main function is to work on the facts of some human characteristics, for example voice recognition, fingerprint, , face recognition, and eye print or the pattern in the retina and so on. this

could serve the security services with finding wanted people and identifying the identity of intruders in different kind of services in which could eliminate the crime world and makes people feel safe with the development of new technologies while building highly secured systems that cannot be hacked and what is more important is to let users feel safe with using their data freely and securely by trusting the new devices and their methods.

References

Ajayan, A. (2013). **Topographical Panoramic Image production using Mobile Cloud** *International Journal of Engineering Trends and Technology (IJETT)*, Volume4, Issue5.

Al-Hamami A., H., and AL-Juneidi, J. (2015). **Secure Mobile Cloud Computing Based-On Fingerprint**, *World of Computer Science and Information Technology Journal (WCSIT)*, ISSN: 2221-0741, Vol. 5, No. 2, 2015.

AL-Khashab, R. (2014). **An Authentication Model for Cloud Computing Application**, Master`s thesis, Amman Arab University.

AL-Kuary, N. (2014). **A proposed Model for Risks Management measurement in Cloud Computing Environment (Software as a Service)**, Master`s thesis, Amman Arab University.

Asrani, P. (2013). **Mobile Cloud Computing. International Journal of Engineering and Advanced Technology (IJEAT)**, 2(4), 606-609.

Butoi, A., Tomai, N., and Mocean, L. (2013). **Cloud-Based Mobile Learning**, 17(2), 27-40.

Choksi, S. (2014). **Comparative Study on Authentication Schemes for Cloud Computing**. International Journal of Engineering Development and Research, Volume 2, Issue 2.

Choudhury, G., and Abudin, J. (2014). **Modified Secure Two Way Authentication System in Cloud Computing Using Encrypted One Time Password**. International Journal of Computer Science & Information Technologies, 5(3).

Chow, R., Jakobsson, M., Masuoka, R., Molina, J., Niu, Y., Shi, E., and Song, Z. (2010). **Authentication in the clouds: a framework and its application to mobile users**. In Proceedings of the 2010 ACM workshop on Cloud computing security workshop (pp. 1-6).ACM

Das, S., and Debbarma, J. (2011). **Designing a Biometric Strategy (Fingerprint) Measure for Enhancing ATM Security in Indian e-banking System** International Journal of Information and Communication.

Davies, J. (2012). **Understanding IPv6: Your Essential Guide to IPv6 on Windows Networks**. "O" Reilly Media, Inc."

Fernando, N., Loke, W., and Rahayu, W. (2013). **Mobile cloud computing: A survey**. Future Generation Computer Systems, 29(1), 84-106.

Ghadirli, M., and Rastgarpour, M. (2013). **A Paradigm for the Application of Cloud Computing in Mobile Intelligent Tutoring Systems.** arXiv preprint arXiv: 1304.4047.

Guha, V., & Shrivastava, M. (2013). **Review of Information Authentication in Mobile Cloud over SaaS & PaaS Layers.** International Journal of Advanced Computer Research (IJACR), 3(1), 9

Jansen, W., and Grance, T. (2011). **Guidelines on security and privacy in public cloud computing.** NIST special publication, 800, 144

Jensen, M., Schwenk, J., Gruschka, N., and Iacono, L. (2009). **On technical security issues in cloud computing.** In **Cloud Computing, 2009.CLOUD'09.** IEEE International Conference on (pp. 109-116). IEEE

Khan, N., Kiah, M., Khan, U., and Madani, S. (2013). **Towards secure mobile cloud computing: A survey.** Future Generation Computer Systems, 29(5), 1278-1299.

Khan, A., and Ahirwar, K. (2011). **Mobile cloud computing as a future of mobile multimedia database.** International Journal of Computer Science and Communication, 2(1), 219-221.

Krishnamoorthy, G. (2013). **Analyzing the Critical Issues of Mobile Users in Cloud Computing. International Journal on Computer Science & Engineering**, 5(4).

Liang, H., Xing, T., Cai, X., Huang, D., Peng, D., and Liu, Y. (2013). **Adaptive computing resource allocation for mobile cloud computing. International Journal of Distributed Sensor Networks**, 2013.

Liske, T. (2005). **Mobility in IPv6**. Technische Universität Dresden.

Lizhao, L., Shunzhi, Z., Zhonghai, S., and Qi, L. (2013). **Mobile Computing Clouds Interactive Model and Algorithm Based On Multi-core Grids. TELKOMNIKA Indonesian Journal of Electrical Engineering**, 11(9), 5267-5276.

Majje S., and Kulkarni W., (2011). **BIOMETRICS AUTHENTICATION TECHNIQUES IN ATM**, BIOINFO Security Informatics, vol. 1, no. 1, pp. 6-10.

Mell, P., and Grance, T. (2011). **The NIST definition of cloud computing**. National Institute of Standards and Technology.

http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html,

Retrieved 1/2/2015

Pawle, A., and Pawar, V. (2013). **Face Recognition System (FRS) on Cloud Computing for User Authentication**. International Journal of Soft Computing and Engineering (IJSCE), 3.

Pocatilu, P., Boja, C., and Ciurea, C. (2013). **Syncing Mobile Applications with Cloud Storage Services**. Informatica Economică, 17(2), 96-108.

Ravindranath, K., and Raja, S. (2013). **A Survey on Energy aware offloading Techniques for Mobile Cloud Computing**, International Journal of Computer Trends and Technology (IJCTT), volume 4, Issue 7.

Roberts II, C., and Al-Hamdani, W. (2011). **Who can you trust in the cloud?: A review of security issues within cloud computing**. In Proceedings of the 2011 Information Security Curriculum Development Conference (pp. 15-19), ACM.

Saha, R., and Bhattacharjee, D. (2013). **Face Recognition Using Eigen faces**. International Journal of Emerging Technology and Advanced Engineering (IJETAEE), Volume 3, Issue 5.

Shetty, J., Anala, R., & Shobha, G. (2015). **an Approach to Secure Access to Cloud Storage Service**. International Journal of Research, 2(1), 364-368.

Tamma, S., Matta, R., and Satyanarayana, S. (2013). **Providing Cloud Services over Mobile Cloud Data**. International Journal of Computer Trends and Technology (IJCTT), volume4, Issue4.

Vijayalakshmi, A., and Arunapriya, R. (2014). **Authentication of data storage using decentralized access control in clouds**. Journal of Global Research in Computer Science, 5(9), 1-4.

Viola, P., & Jones, M. (2004). **Rapid object detection using a boosted cascade of simple features**. In **Computer Vision and Pattern Recognition**, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-511) IEEE

Waghmare, M., and Chavan, T. (2013). **Outsourcing with secure accessibility in mobile cloud computing**. International Journal of Computer Trends and Technology (IJCTT), volume4, Issue4.

Wilson, I., and Fernandez, J. (2006). **Facial feature detection using Haar classifiers**. ١٣٣-١٢٧، (٤)٢١ Sciences in Colleges, Journal of Computing

Zhang, X., Schiffman, J., Gibbs, S., Kunjithapatham, A., and Jeong, S. (2009). **Securing elastic applications on mobile devices for cloud computing**. In Proceedings of the 2009 ACM workshop on Cloud computing security (pp. 127-134).ACM.

Zhang, Y., Niyato, D., and Wang, P. (2013). **An auction mechanism for resource allocation in mobile cloud computing systems**. In *Wireless Algorithms, Systems, and Applications* (pp. 76-87). Springer Berlin Heidelberg.

Appendix

Face Lock Application

```
public class MainActivity extends ActionBarActivity {

    private static final int ACTION_TAKE_PHOTO_B = 1;

    private String mCurrentPhotoPath;

    private static final String JPEG_FILE_PREFIX = "IMG_";

    private static final String JPEG_FILE_SUFFIX = ".jpg";

    private Bitmap mImageBitmap;

    private void dispatchTakePictureIntent() {

        Intent takePictureIntent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);

        File f = null;

        try {

            f = setUpPhotoFile();

            mCurrentPhotoPath = f.getAbsolutePath();

            takePictureIntent
```

```
.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(f));

    catch (IOException e) {

        e.printStackTrace();

        f = null;

        mCurrentPhotoPath = null;

        startActivityForResult(takePictureIntent,
ACTION_TAKE_PHOTO_B);

    private File setUpPhotoFile() throws IOException {

        File f = createImageFile();

        mCurrentPhotoPath = f.getAbsolutePath();

        return f;

    private void handleBigCameraPhoto() {

        if (mCurrentPhotoPath != null) {

            setPic();

            mCurrentPhotoPath = null;

        private void setPic()
```



```
/* There isn't enough memory to open up more than a couple  
camera photos */
```

```
/* So pre-scale the target bitmap into which the file is  
decoded */
```

```
/* Get the size of the ImageView */
```

```
inttargetW = imageView1.getWidth();
```

```
inttargetH = imageView1.getHeight();
```

```
/* Get the size of the image */
```

```
BitmapFactory.OptionsbmOptions = new  
BitmapFactory.Options();
```

```
bmOptions.inJustDecodeBounds = true;
```

```
BitmapFactory.decodeFile(mCurrentPhotoPath, bmOptions);
```

```
intphotoW = bmOptions.outWidth;
```

```
intphotoH = bmOptions.outHeight;
```

```
/* Figure out which way needs to be reduced less */
```

```
intscaleFactor = 1;
```

```
if ((targetW > 0) || (targetH > 0)) {  
  
    scaleFactor = Math.min(photoW / targetW, photoH /  
targetH);  
  
    /* Set bitmap options to scale the image decode target */  
  
    bmOptions.inJustDecodeBounds = false;  
  
    bmOptions.inSampleSize = scaleFactor;  
  
    bmOptions.inPurgeable = true;  
  
  
    /* Decode the JPEG file into a Bitmap */  
  
    Bitmap bitmap =  
BitmapFactory.decodeFile(mCurrentPhotoPath, bmOptions);  
  
    ExifInterface exif = null;  
  
    try {  
  
        exif = new ExifInterface(mCurrentPhotoPath);  
  
    } catch (IOException e) {  
  
        // TODO Auto-generated catch block  
  
        e.printStackTrace();  
    }  
}
```

```

        int rotation =
exif.getAttributeInt(ExifInterface.TAG_ORIENTATION,

        ExifInterface.ORIENTATION_NORMAL);

        int rotationInDegrees = exifToDegrees(rotation);

        Matrix matrix = new Matrix();

        if (rotation != 0f) {

            matrix.preRotate(rotationInDegrees);

            bitmap = Bitmap.createBitmap(bitmap, 0, 0,
bitmap.getWidth(),

            bitmap.getHeight(), matrix, true);

            mImageBitmap =
Bitmap.createScaledBitmap(bitmap, 640, 480, false);

            /* Associate the Bitmap to the ImageView */

            imageView1.setImageBitmap(bitmap);

            imageView1.setVisibility(View.VISIBLE);

            private static int exifToDegrees(int exifOrientation) {

                if (exifOrientation ==

```

```
ExifInterface.ORIENTATION_ROTATE_90) {  
  
    return 90;  
  
    } else if (exifOrientation ==  
ExifInterface.ORIENTATION_ROTATE_180) {  
  
    return 180;  
  
    } else if (exifOrientation ==  
ExifInterface.ORIENTATION_ROTATE_270) {  
  
    return 270;  
  
    return 0;  
  
private File createImageFile() throws IOException {  
  
    // Create an image file name  
  
    String timeStamp = new  
SimpleDateFormat("yyyyMMdd_HH:mm:ss")  
  
        .format(new Date());  
  
    String imageFileName = JPEG_FILE_PREFIX + timeStamp  
+ "_";
```

```

        File storageDir =
Environment.getExternalStoragePublicDirectory(Environment.DIRECT
ORY_PICTURES);

        File imageF = File.createTempFile(imageFileName,
JPEG_FILE_SUFFIX,

                storageDir);

        return imageF;

        @Override

        protected void onActivityResult(int requestCode, int resultCode,
Intent data) {

                switch (requestCode) {

                case ACTION_TAKE_PHOTO_B: {

                        if (resultCode == RESULT_OK) {

                                handleBigCameraPhoto();

                                        break;

                                } // ACTION_TAKE_PHOTO_B

                } // switch

                ImageView imageView1;

        @Override

```

```

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    TextViewtxtView = (TextView)
findViewById(R.id.textView4);

    txtView.setOnClickListener(new OnClickListener() {

        @Override

public void onClick(View v) {

            MainActivity.this.startActivity(new
Intent(MainActivity.this
SignUpActivity.class));

            Button btnLogin = (Button) findViewById(R.id.btnLogin);

            btnLogin.setOnClickListener(new OnClickListener() {

                @Override

                public void onClick(View v) {

                    TextView UN = (TextView)

```

```

findViewById(R.id.pass1);

        UserName = UN.getText().toString();

        TextView P = (TextView)
findViewById(R.id.pass2);

        Password = P.getText().toString();

        Thread thread = new Thread(new Runnable() {

                @Override

                public void run() {

                        Login l = new Login();

                        try {

if (l.DoLogin(UserName, Password, mImageBitmap)) {

runOnUiThread(new Runnable() {

                @Override

                public void run() {

Intent myIntent = new Intent(

```

```
MainActivity.this,
        CloudFiles.class);

//myIntent.putExtra("key", value);

// //Optional parameters

MainActivity.this

        .startActivity(myIntent);

        Toast.makeText(getApplicationContext(),
                "Success login",
                Toast.LENGTH_LONG).show();

    } else {

runOnUiThread(new Runnable() {

@Override

public void run() {

        Toast.makeText(getApplicationContext(),
                "Failure login",
                Toast.LENGTH_LONG).show();
```



```
} catch (IOException | XmlPullParserException e) {  
  
// TODO Auto-generated catch block  
  
e.printStackTrace();  
  
thread.start();  
  
        imageView1 = (ImageView)  
findViewById(R.id.imageView1);  
  
        imageView1.setOnClickListener(new OnClickListener() {  
  
            @Override  
  
            public void onClick(View v) {  
  
                // TODO Auto-generated method stub  
  
                dispatchTakePictureIntent();  
  
  
  
                static String UserName = "";  
  
                static String Password = "";
```

```
public class Login {  
  
    private final String SERVER_IP = "192.168.1.5";  
  
    private final String NAMESPACE =  
"http://FaceRecognitionServer.org/";  
  
    private final String METHOD_NAME = "Login";  
  
    private final String METHOD_NAME_SIGNUP = "Register";  
  
    private final String SIGNUP_SOAP_ACTION = NAMESPACE +  
METHOD_NAME_SIGNUP;  
  
    private final String GET_FILES_METHOD_NAME = "GetFiles";  
  
    private final String ServerURL = "/FaceRecognitionServer/";  
  
    private final String URL = ServerURL +  
"FaceRecognitionServer.asmx";  
  
    private final String SOAP_ACTION = NAMESPACE +  
METHOD_NAME;  
  
    private final String GET_FILES_SOAP_ACTION =  
NAMESPACE + GET_FILES_METHOD_NAME;  
  
    public String getLocalIpAddress(Context context) {
```

```

return getWifiInetAddress(context).getHostAddress();

private
Enumeration<InetAddress> getWifiInetAddresses(final Context context)

    final WifiManager wifiManager = (WifiManager)
context.getSystemService(Context.WIFI_SERVICE);

    final WifiInfo wifiInfo = wifiManager.getConnectionInfo();

    final String macAddress = wifiInfo.getMacAddress();

    final String[] macParts = macAddress.split(":");

    final byte[] macBytes = new byte[macParts.length];

    for (int i = 0; i < macParts.length; i++) {

        macBytes[i] = (byte) Integer.parseInt(macParts[i],
16);

        try {

            final Enumeration<NetworkInterface> e =
NetworkInterface.getNetworkInterfaces();

            while (e.hasMoreElements()) {

                final NetworkInterface networkInterface =
e.nextElement();

```

```

        if
        (Arrays.equals(networkInterface.getHardwareAddress(), macBytes)) {

            return networkInterface.getInetAddresses();

        } catch (SocketException e) {

            Log.wtf("WIFIIP", "Unable to
            NetworkInterface.getNetworkInterfaces()");

            return null;

            public InetAddress getWifiInetAddress(final Context context)

            final Enumeration<InetAddress> e =
            getWifiInetAddresses(context);

            while (e.hasMoreElements()) {

                final InetAddress inetAddress = e.nextElement();

                if (inetAddress.getClass() == InetAddress.class) {

                    return inetAddress;

                }

            }

            return null;

```

```
public boolean DoLogin(String UserName, String Password, Bitmap
Face) throws IOException, XmlPullParserException {
```

```
    SoapObject request = new SoapObject(NAMESPACE,
METHOD_NAME);

    request.addProperty("UserName", UserName);

    request.addProperty("Password", Password);

    request.addProperty("ImageB64", getEncodeData(Face));

    SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);

    envelope.dotNet = true;

    envelope.setOutputSoapObject(request);

    // String LocalIP = getLocalIpAddress(ctx).replace("\r", "");

    String remoteIP = SERVER_IP;

    HttpTransportSE androidHttpTransport = new
HttpTransportSE("http://" + remoteIP + URL, 100000);

    androidHttpTransport.call(SOAP_ACTION, envelope);

    SoapPrimitive result = (SoapPrimitive)
envelope.getResponse();
```

```

        return (result.toString().equals("true"));

    public boolean SignUp(String UserName, String Password, Bitmap
    Face) throws IOException, XmlPullParserException {

        SoapObject request = new SoapObject(NAMESPACE,
    METHOD_NAME_SIGNUP);

        request.addProperty("UserName", UserName);

        request.addProperty("Password", Password);

        request.addProperty("ImageB64", getEncodeData(Face));

        List<String> logs = new ArrayList<String>();

        logs.add(getEncodeData(Face));

        SoapSerializationEnvelope envelope = new
    SoapSerializationEnvelope(SoapEnvelope.VER11);

        envelope.dotNet = true;

        envelope.setOutputSoapObject(request);

        String remoteIP = SERVER_IP;

        HttpTransportSE androidHttpTransport = new
    HttpTransportSE("http://" + remoteIP + URL, 100000);

```

```

        androidHttpTransport.call(SIGNUP_SOAP_ACTION,
envelope);

        SoapPrimitive result = (SoapPrimitive)
envelope.getResponse();

        return (result.toString().equals("true"));

    public String[] GetFiles() throws IOException,
XmlPullParserException {

        SoapObject request = new SoapObject(NAMESPACE,
GET_FILES_METHOD_NAME);

        SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);

        envelope.dotNet = true;

        envelope.setOutputSoapObject(request);

        String remoteIP = SERVER_IP;

        HttpTransportSE androidHttpTransport = new
HttpTransportSE("http://" + remoteIP + URL, 100000);

        androidHttpTransport.call(GET_FILES_SOAP_ACTION,
envelope);

```

```

SoapObject result = (SoapObject) envelope.getResponse();

ArrayList<String> AL = new ArrayList<String>();

for (int i = 0; i <result.getPropertyCount(); i++) {

    String responseChild =

result.getProperty(i).toString();

        AL.add("http://" + remoteIP + ServerURL + "Files/" +

responseChild);

String[] mStringArray = new String[AL.size()];

mStringArray = AL.toArray(mStringArray);

returnmStringArray;

private String getEncodeData(Bitmap bm) {

String encodedimage1 = null;

try {

        ByteArrayOutputStreambaos = new

ByteArrayOutputStream();

        bm.compress(Bitmap.CompressFormat.PNG, 50,

baos);

```



```

        byte[] b = baos.toByteArray();

        encodedimage1 = Base64.encodeToString(b,
Base64.DEFAULT);

    } catch (Exception e) {

        System.out.println("Exception: In getEncodeData" +
e.toString());

        return encodedimage1;

    private Bitmap decodeFile(Resources resources, int Id) {

        Bitmap b = null;

        final int IMAGE_MAX_SIZE = 400;

        try {

            BitmapFactory.Options o = new
BitmapFactory.Options();

            o.inJustDecodeBounds = true;

            BitmapFactory.decodeResource(resources, Id, o);

            int scale = 1;

            if (o.outHeight > IMAGE_MAX_SIZE || o.outWidth >
IMAGE_MAX_SIZE) {

```

```
        scale = (int) Math.pow(2.0, (int)
Math.round(Math.log(IMAGE_MAX_SIZE / (double)
Math.max(o.outHeight, o.outWidth)) / Math.log(0.5)));

        BitmapFactory.Options o2 = new
BitmapFactory.Options();

        o2.inSampleSize = scale;

        b = BitmapFactory.decodeResource(resources, Id, o2);

    } catch (Exception e) {

        Log.v("Exception in decodeFile() ", e.toString() + "");

        return b;

using Emgu.CV;

using Emgu.CV.CvEnum;

using Emgu.CV.Structure;

using System;

using System.Collections.Generic;

using System.Drawing;

using System.IO;
```

```
using System.Linq;

using System.Web;

namespace CloudServerFaceRecognition
{

public class FaceRecognitionManager
{

const int ACCEPTED_DEGREE = 45;

string Path = "";

List<Image<Gray, byte>> trainingImages = new List<Image<Gray,
byte>>();

List<string> labels = new List<string>();

HaarCascade face = null;

HaarCascade eye = null;

public FaceRecognitionManager(string path)
{

Path = path;
```

```

eye = new HaarCascade(System.IO.Path.Combine(Path ,
"XML\\haarcascade_eye.xml"));

face = new HaarCascade(System.IO.Path.Combine(Path ,
"XML\\haarcascade_frontalface_default.xml"));

try

    //Load of previustrained faces and labels for each image

if (!System.IO.Directory.Exists(System.IO.Path.Combine(Path,
"UsersStore\\Faces")))

System.IO.Directory.CreateDirectory(System.IO.Path.Combine(Path,
"UsersStore\\Faces"));

stringLabelsinfo = File.ReadAllText(System.IO.Path.Combine(Path,
"UsersStore\\Faces\\FacesDB.txt"));

string[] Labels = Labelsinfo.Split('% ');

intNumLabels = Convert.ToInt16(Labels[0]);

stringLoadFaces;

for (inttf = 1; tf<NumLabels + 1; tf++)

LoadFaces = "face" + tf + ".png";

```

```

trainingImages.Add(new Image<Gray,
byte>(System.IO.Path.Combine(Path, "UsersStore\\Faces\\" +
LoadFaces)));

labels.Add(Labels[tf]);

catch (Exception e)

        //System.Windows.Forms.MessageBox.Show("Nothing in
binary database, please add at least a face(Simply train the prototype with
the Add Face Button).", "Trained faces load",
System.Windows.Forms.MessageBoxButtons.OK,
System.Windows.Forms.MessageBoxIcon.Exclamation);

        Bitmap B64ToImage(string B64S)

byte[] bytes = Convert.FromBase64String(B64S);

        Bitmap image;

using (MemoryStream ms = new MemoryStream(bytes))

image = (Bitmap)Image.FromStream(ms);

return image;

string ImageToB64(Bitmap img)

string base64String = "";

```

```

using (MemoryStream ms = new MemoryStream())

    // Convert Image to byte[]

img.Save(ms, System.Drawing.Imaging.ImageFormat.Png);

    // Convert byte[] to Base64 String

base64String = Convert.ToBase64String(ms.ToArray());

return base64String;

public void AddFace(String FaceB64, string Name)

    Bitmap Face = B64ToImage(FaceB64);

    Image<Bgr, Byte>currentFrame = new Image<Bgr, byte>(Face);

int y = (int)((float)currentFrame.Height / (currentFrame.Width /
(float)RequiredHeight));

    Image<Gray, byte> gray = currentFrame.Convert<Gray,
byte>().Resize(RequiredHeight, y,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);

    //Try to detect eyes

```

```

MCvAvgComp[][] eyesDetected = gray.DetectHaarCascade(
    eye,
        1.1,
        10,
        Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DEFAULT,
        new Size(20, 20));

MCvAvgComp[] eyeRects = eyesDetected[0].OrderByDescending(a
=>a.rect.Width * a.rect.Height).Take(2).ToArray();

if (eyeRects != null &&eyeRects.Length == 2)
    {
        Rectangle R1 = eyeRects[0].rect;

        Rectangle R2 = eyeRects[1].rect;

        PointF P1 = new PointF(R1.X + R1.Width / 2f, R1.Y + R1.Height / 2f);

        PointF P2 = new PointF(R2.X + R2.Width / 2f, R2.Y + R2.Height / 2f);

        LineSegment2DF line = new LineSegment2DF(P1, P2);

        doubledeltaY = line.P2.Y - line.P1.Y;

        doubledeltaX = line.P2.X - line.P1.X;
    }

```

```
double angle;

if (deltaX != 0)

angle = Math.Atan2(deltaY, deltaX) * (180f / Math.PI);

else

angle = 90;

vartmpGray = gray.Clone();

foreach (MCvAvgCompey in eyeRects)

    {

        Rectangle eyeRect = ey.rect;

tmpGray.Draw(eyeRect, new Gray(1), 2);

    }

if (Math.Abs(angle) >= 90)

    {

if (angle > 0)

angle -= 180;

else

angle += 180;
```



```

gray = gray.Rotate(-angle, new Gray(255));

    }

else

    {

vartmpGray = gray.Clone();

tmpGray.Draw("No eye detected!", ref font, new Point(100, 400), new
Gray(0));

    }

//Face Detector 1.2,10

MCvAvgComp[][] facesDetected = gray.DetectHaarCascade(

face,

    1.1,

    5,

Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DEFAULT,

new Size(24, 24));

    Image<Gray, byte> result, TrainedFace = null;

//Action for each element detected

```

```

if (facesDetected[0].Length == 0)
    {

    vartmpGray = gray.Clone();

    tmpGray.Draw("No face detected!", ref font, new Point(100, 200), new
    Gray(0));

    }

foreach (MCvAvgComp f in facesDetected[0])
    {

        y = (int)((float)f.rect.Height / (f.rect.Width /
(float)RequiredHeight));

        TrainedFace = gray.Copy(f.rect).Resize(RequiredHeight, y,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);

        //resize face detected image for force to compare the same size
with the

        //test image with cubic interpolation type method

        trainingImages.Add(TrainedFace);

        labels.Add(Name);

    }

```

```

//Write the number of triained faces in a file text for further load

File.WriteAllText(System.IO.Path.Combine(Path,
"UsersStore\\Faces\\FacesDB.txt"),
trainingImages.ToArray().Length.ToString() + "%");

//Write the labels of triained faces in a file text for further load

for (int i = 1; i <trainingImages.ToArray().Length + 1; i++)
{

trainingImages.ToArray()[i - 1].Save(System.IO.Path.Combine(Path,
"UsersStore\\Faces\\face" + i + ".png"));

File.AppendAllText(System.IO.Path.Combine(Path,
"UsersStore\\Faces\\FacesDB.txt"), labels.ToArray()[i - 1] + "%");

public delegate void Step(Bitmap Img);

intRequiredHeight = 640;

MCvFont font = new
MCvFont(FONT.CV_FONT_HERSHEY_TRIPLEX, 1.5d, 1.5d);

public string RecognizeFace(String FaceB64)

Bitmap Face = B64ToImage(FaceB64);

Image<Bgr, Byte>currentFrame = new Image<Bgr, byte>(Face);

```

```

//Get the current frame form capture device

int y = (int)((float)currentFrame.Height / (currentFrame.Width /
(float)RequiredHeight));

    Image<Gray, byte> gray = currentFrame.Convert<Gray,
byte>().Resize(RequiredHeight, y,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);

//Try to detect eyes

MCvAvgComp[][] eyesDetected = gray.DetectHaarCascade(
eye, 1.1,10,
Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNI
NG,
new Size(20, 20));

MCvAvgComp[] eyeRects = eyesDetected[0].OrderByDescending(a
=>a.rect.Width * a.rect.Height).Take(2).ToArray();

if (eyeRects != null &&eyeRects.Length == 2)

    Rectangle R1 = eyeRects[0].rect;

    Rectangle R2 = eyeRects[1].rect;

PointF P1 = new PointF(R1.X + R1.Width / 2f, R1.Y + R1.Height / 2f);

```

```
PointF P2 = new PointF(R2.X + R2.Width / 2f, R2.Y + R2.Height / 2f);
```

```
LineSegment2DF line = new LineSegment2DF(P1, P2);
```

```
doubledeltaY = line.P2.Y - line.P1.Y;
```

```
doubledeltaX = line.P2.X - line.P1.X;
```

```
double angle;
```

```
if (deltaX != 0)
```

```
angle = Math.Atan2(deltaY, deltaX) * (180f / Math.PI);
```

```
else
```

```
angle = 90;
```

```
vartmpGray = gray.Clone();
```

```
foreach (MCvAvgCompey in eyeRects)
```

```
Rectangle eyeRect = ey.rect;
```

```
tmpGray.Draw(eyeRect, new Gray(1), 2);
```

```
if (Math.Abs(angle) >= 90)
```

```
if (angle > 0)
```

```
angle -= 180;
```

```
else
```

```

angle += 180;

if (Math.Abs(Math.Abs(angle) - cACCEPTED_DEGREE) > 5)

return "Invalid user";

gray = gray.Rotate(-angle, new Gray(255));

else

vartmpGray = gray.Clone();

tmpGray.Draw("No eye detected!", ref font, new Point(100, 400), new
Gray(0));

    }

    //Face Detector

    MCvAvgComp[][] facesDetected = gray.DetectHaarCascade(

face,

    1.1,

    5,

    Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DEFAULT,

new Size(24, 24));

    Image<Gray, byte> result = null;

```

```

intContTrain = trainingImages.Count;

int t = 0;

    //Action for each element detected

foreach (MCvAvgComp f in facesDetected[0])

    y = (int)((float)f.rect.Height / (f.rect.Width /
(float)RequiredHeight));

result = gray.Copy(f.rect).Resize(RequiredHeight, y,
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);

    //draw the face detected in the 0th (gray) channel with blue
color

    // currentFrame.Draw(f.rect, new Bgr(Color.Red), 2);

string name = "";

if (trainingImages.ToArray().Length != 0)

    {

        //TermCriteria for face recognition with numbers of trained
images like maxIteration

        MCvTermCriteria termCrit = new MCvTermCriteria(ContTrain * 3,
0.001);

```

```
//Eigen face recognizer

EigenObjectRecognizer recognizer = new EigenObjectRecognizer(

trainingImages.ToArray(),

labels.ToArray(),

                    5000,

reftermCrit);

var res = recognizer.Recognize(result);

if (res != null)

name = res.Label;

return name

{

{

return

{

{

{
```