Cloud Computing Access Authentication through Mobile Device Based on Face Recognition

Alaa Hussein Al-Hamami, Ali Mohammad Saab
Amman Arab University, Amman 11953, Jordan
alaa_hamami@yahoo.com

Abstract--- Cloud computing is not just restricted to pc platforms, but mobiles as well as 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 adapting the methodology of biometric authentication technique. This include analyzing the user's biometric features specifically the face in order to identify their identity. Biometric authentication is the most popular analysis technique that measures human features in which relies on the physical features of any human body. To sum up with, as the technology world is evolving this could lead to adapting new authentication method that enhances the security by using the front camera to identify users through a high resolution picture.
Keywords: Cloud Network; Authentication; Biometric; Face Recognition; Mobile device.

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 [1]. Cloud computing offers 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. At the same time offers 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 [2].

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. The main concepts of cloud computing is the best contract about the way to achieve safety security in different planes. The achieved security let information managers to formal that security is first and only one worry with cloud computing [3]. To attract potential consumers, the cloud service provider has to target all the security issues to provide a completely secure environment in MCC. The risk of user's data is stored on cloud servers, the security warnings caused by several simulated technologies, and imposition through different attacks. Due to resource restriction, the security systems presented for the cloud-computing platform cannot be directly executed on a mobile device. Lightweight secure framework is required to offer security with less interaction and processing overhead on mobile devices [4].

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.

II. 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. In this research, 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. The main contribution of this research focuses on the biometric recognition (Face recognition) to produce a resolution for the problem of cloud security. Also, to guarantee the secure access to restrict data/services in the cloud using a mobile phone also will facilitate the work of the people who use the mobile. These contributions are summarized as the following:

III. THE PROPOSED MODEL

The applied mechanism includes the following stages as shown in Fig 1.

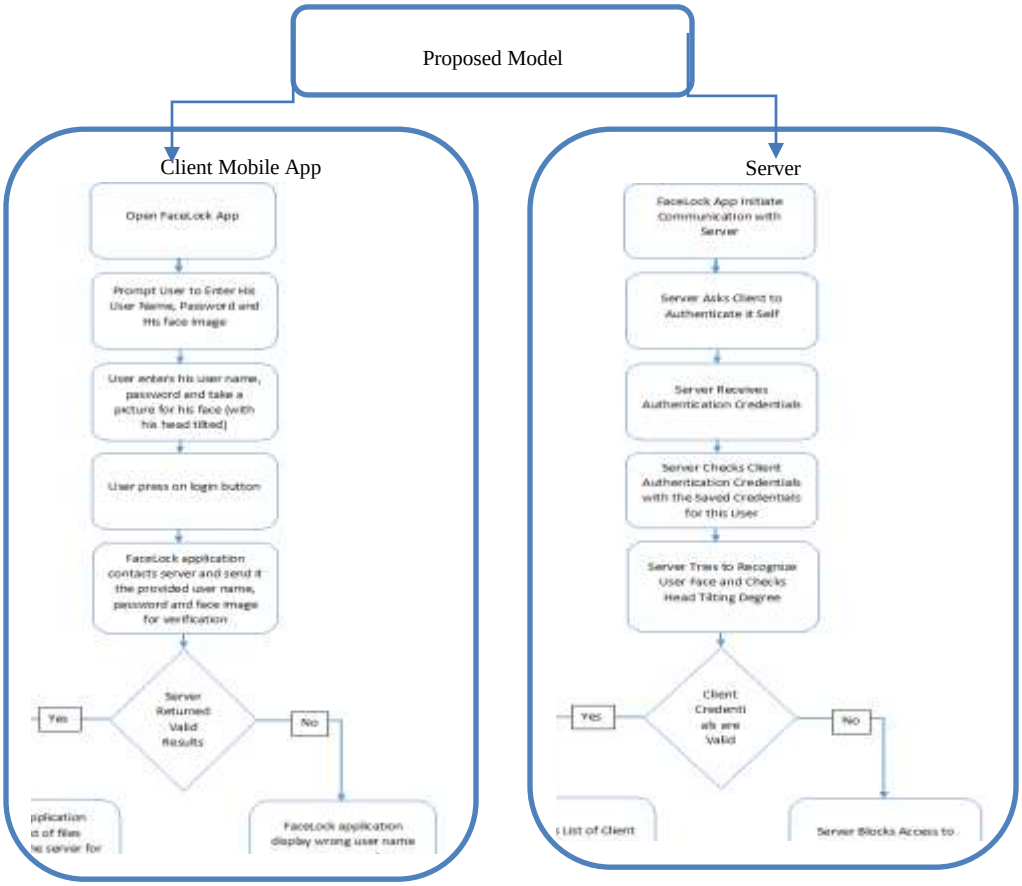


Fig. 1 Proposed Model

Stage 1: Collecting user information (Client):
The first stage takes place at the user 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, in this model we will use face recognition in which the

user takes a picture for his/her face 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 take.

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.
Face recognition: the server performs face recognition for the received picture and tries to identify the person in it, if the person identified as the same person with the user name and password that sent before, then the validation step is successful, otherwise it fails.

Face tilting checking: After validating user 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 steps:

Face detection procedure
Face detection is performed using Haar-like feature cascade classifier, 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. 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 Fig. 2.

Fig. 2 calculated by subtracting pixel [5]

After update 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 as an implementation

Face recognition procedure
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.
Compute the eigenvectors and Eigen values of the covariance framework. Every eigenvector

have 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.

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*(λ1+ λ2+… + λn), n= number of information pi. Number of principle (k) is the littlest number fulfills:

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

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 [6].

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 OpenCV as an execution of the Haar-highlight classifier [7]. Face detection is performed using Haar-like feature cascade classifier, 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. Each feature has one value; this value is calculated by subtracting pixels' values under the white rectangle from pixels' values under the black rectangle. Practically speaking an exceptionally basic structure is utilized to deliver a compelling classifier which 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. 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 as an implementation of the Haar-feature classifier.

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-hub and the eyes line, tilting client face degree figuring is portrayed in the figures beneath.

Prepare Image

In order for the face and eye detection algorithms to work the image should be resized to 640x480 and converted to grayscale this is mandatory for faster image processing.

Algorithm

Step1 : Img←UserFacePicture
Step2 : Img640 ←ResizeImageto640x480(Img)
Step3 : ImgGray←ConvertToGray(Img640)

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.

Algorithm

Step1 : EyeHaarCascade← new EyeHaarCascade(“EyesHaar.xml”)
Step2: EyesDetectedRects←EyeHaarCascade(ImgGray)

Algorithm

Step1 : FaceHaarCascade← new FaceHaarCascade("FacesHaar.xml")
Step2: DetectedFace←DetectFace(Image)
Eyes Line Slope (face Rotation)
After detecting eye and prior to face recognition the user face will be rotated to insure best results, the face tilting degree is calculated from the detected eyes rectangles.
Algorithm
Step1 : eyeRects←EyeDetection(Image)
Step2 : Rectangle R1 ←eyeRects[0].rect
Step3 : Rectangle R2 ←eyeRects[1].rect
Step4 : PointF P1 ← new PointF(R1.X + R1.Width / 2f, R1.Y + R1.Height / 2f)
Step5 :PointF P2 ← new PointF(R2.X + R2.Width / 2f, R2.Y + R2.Height / 2f)
Step6 :LineSegment2DF line ←new LineSegment2DF(P1, 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)
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 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. Fig. 3 describes the concept of the suggested design.

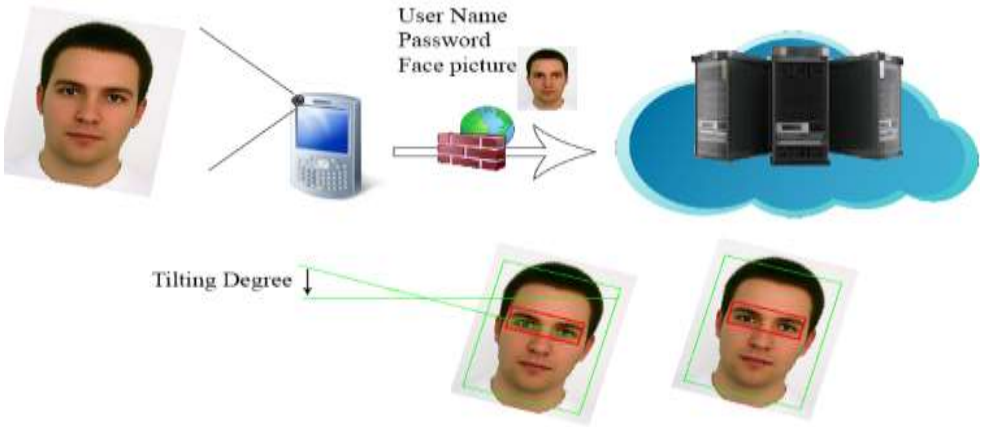


Fig. 3 the Concepts of the Suggested Design

III. 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. 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. 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 consists of two parts:

Something user knows which is the user name, password and face tilting degree.

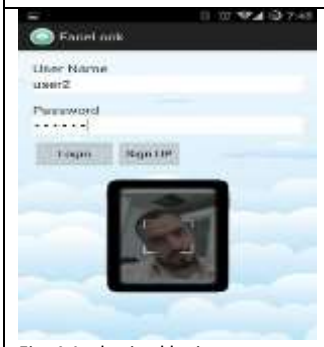
Something user is which is a picture of his/her face.

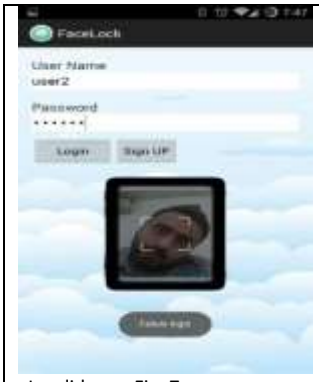
New users open the application "Face Lock" and then press on "Sign Up" button. The user should enter a valid unique user name that consists of minimum 6 letters with no spaces and a valid password that consists of minimum 6 letters with numbers, symbols and letters in it. After entering user credentials, 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 was different (in case of unauthorized access be user picture).

This screen collects user information and sends them 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 his face and store this information in the server DB for future use in user credentials validation. 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.

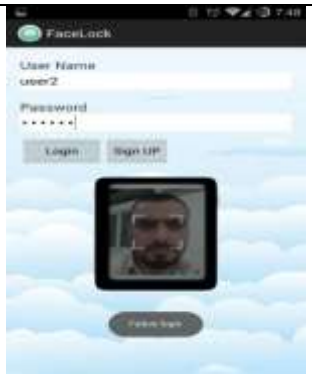
The user should enter his/her user name and password and then click on the black rectangle to take a picture with the smart phone front camera, users should keep in mind to tilt their head by the same degree they tilted it at the registration process. After taking the picture the user should press on the login button, the application will communicate with the cloud server and sends 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 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. The following figures are describing some of the result we got from the experiments:

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 Fig. 4.	After that is correctly authorizing access by server the user can be access secret files as shown in Fig. 5.	Now we will try to log in with invalid user name, password and with incorrect tilting angle as shown in Fig. 6.
 Fig. 4 Authorized login	 Fig. 5 Authorized login	 Fig. 6 Unauthorized login
1. Valid credentials (stolen user name & password) with invalid user image as shown in Fig. 7.	2. Valid credentials with valid image but with invalid head tilting angle as shown in Fig. 8.	Following is how user can access the secure information stored at the server after authentication, as shown in Fig. 9.



Invalid user Fig. 7



Invalid angle Fig. 8

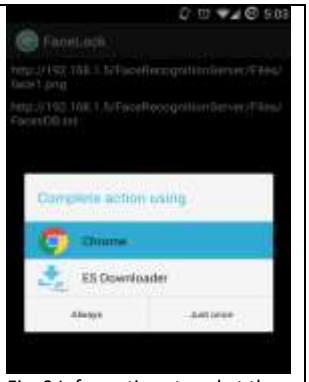


Fig. 9 Information stored at the server

Conclusions

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 high 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 user identity which is the biometric face recognition, the 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, and to guarantee the secure access to restrict data/services in the cloud using a mobile phone also will facilitate the work of people who use the mobile.

REFERENCES

1. P. Asrani. Mobile Cloud Computing. International Journal of Engineering and Advanced Technology. IJEAT, Vol. 2, No. 4, 2013, P. 606-609.
2. V. Gutha and M. Shrivastava. Review of Information Authentication in Mobile Cloud over SaaS & PaaS Layers. International Journal of Advanced Computer Research (IJACR), Vol. 3, No. 1, 2013, P. 9.
3. J. C. Roberts II and W. Al-Hamdani. 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, 2011, P. 15-19.
4. S. S. Majge & H. W. Kulkarni. Biometrics Authentication Techniques In ATM. BIOINFO Security Informatics, vol. 1, no. 1, 2011, P. 6-10.
5. A. A. Pawle & V. P. Pawar. Face Recognition System (FRS) on Cloud Computing for User Authentication. International Journal of Soft Computing and Engineering (IJSCE), Vol. 3, 2013.
6. S. Choksi. Comparative Study on Authentication Schemes for Cloud Computing. International Journal of Engineering Development and Research. Vol. 2, Issue 2, 2014.
7. J. Shetty, M. R. Anala & G. Shobha. An Approach to Secure Access to Cloud Storage Service. International Journal of Research, Vol. 2, No. 1, 2015, P. 364-368.
8. A. H. Al-Hamami & R. A. Al-Khashab. Cloud Authentication Method Based on Multiple Passwords Technique., Journal of Advanced Computer Science and Technology Research, Vol.4, No.2, 2014, P. 33-39.