"Secure Data Processing Framework for Mobile CloudComputing"

MISS. PRACHI DHOBLE[1], MR. SHUBHAM KAMBLI[2], MISS. SNEHAL DONGRE[3], MR. AMOL ZANZAD[4], MISS. PRATIKSHA ADLE[5]

***[6]Guided By-* Prof. MANJUSHA TALMALE**

[1-6]DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, GURU NANAK INSTITUTE OF TECHNOLOGY

---

1. INTRODUCTION

Mobicloud framework is one in which user can stored unlimited data in cloud and can access those data in mobile. The use of mobile devices to establish ad-hoc communication systems is a viable solution that provides global connectivity to support a broad range of applications.

With the help of wireless technology such 3G, 4G,WiMaxetc mobile device can access of network over higher bandwidth and longer distance. In general, mobile users can greatly benefited from cloud services such as data mining, searching information, storing data, retrieving data, securing data on cloud etc.

Here, we present a secure mobile cloud computing framework called mobicloud, in which mobile device can treated as client that is input and cloud is treated as server that is output. The main goal of our project is if any file or application is not open in android mobile because of its fewer configurations then it go to the cloud and start processing and it will open in our mobile. It is done by using ESSI technique i.e. extended semi-shadow image.

Cloud computing, or in simpler shorthand just "The Cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. For example, a cloud computer facility that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server). This approach should maximize the use of computing power thus reducing environmental damage as well since less power, air conditioning, rackspace, etc. are required for a variety of functions. With cloud computing, multiple users can access a single server to retrieve and update their data without purchasing them. The cloud aims to cut costs, and help the users focus on their core business instead of being impeded by IT obstacles.

The main enabling technology for cloud computing is virtualization. Virtualization software allows a physical computing device to be electronically separated into one or more "virtual" devices, each of which can be easily used and managed to perform computing tasks. With operating system–level virtualization essentially creating a scalable system of multiple independent computing devices, idle computing resources can be allocated and used more efficiently. Virtualization provides the agility required to speed up IT operations, and reduces cost by increasing infrastructure utilization. Autonomic computing automates the process through which the user can provision resources on-demand. By minimizing user involvement, automation speeds up the process, reduces labor costs and reduces the possibility of human errors.

In spite of the hype achieved by mobile cloud computing, the growth of the mobile cloud computing subscribers is still below expectations due to the risks associated with the security and privacy. This study is based on existing literature, highlights the current state of the work proposed to secure mobile cloud computing infrastructure. Itani et al. proposed an energy efficient integrity verification scheme for mobile clients to verify the integrity of the files stored on a cloud server using an incremental message authentication code. The proposed scheme offloads most of the integrity verification jobs on a cloud service provider and trusted third party to minimize the processing overhead on the mobile client. The cloud service provider redirects the stored files towards the coprocessor when instructed by a mobile client. The coprocessor computes incremental MAC on received files for integrity verification. Jia et al., proposed a secure data service that outsources data and security management on cloud without disclosing any user information with the help of proxy re- encryption and identity based encryption schemes. Although the proposed secure data service has removed security management overhead from mobile users, still mobile users have to perform cryptographic operations before uploading a file on cloud.

The cryptographic operations involve massive pairing evaluations and exponential calculations. The cryptographic operations consume a considerable amount of energy that needs to be considered while designing a secure framework for mobile cloud computing. Secondly, the cloud is responsible for performing the security management and re- encryption on behalf of the mobile user. Hsueh et al. proposed a scheme for smart phones that ensures the security, integrity, and authentication of mobile user data. The mobile user encrypts the files using

traditional asymmetric encryption techniques. The encrypted files are stored on cloud servers along with mobile user name, signature, and password. The encrypted files along with user credentials may be stored on a cloud server hosted by an adversary. The adversary can utilize credentials to impersonate the user later on. Secondly, the proposed scheme ignored the processing and storage limitations of the device.

The encryption and decryption and even hash function applied on an entire file are performed on the mobile device. Yang et al. proposed a public provable data possession scheme for a resource constrained mobile device that ensures privacy and confidentiality. Trusted third party is responsible for handling encoding/decoding, encryption/decryption, signature generation, and verification on behalf of the mobile user. Although the offloading of mobile user's jobs on trusted third party saves energy, an increase in the number of mobile users results in performance degradation. Huan et al. proposed a new mobile cloud computing framework that not only provides conventional computation services but also improves the functionality of MANET in terms of risk management, trust management, and secure routing. In spite of the advantages provided by the MobiCloud to MANET, the MobiCloud framework did not consider the trust worthiness of the cloud node. There should be a mechanism to store mobile user information on cloud servers in a secure manner.

2. OVERVIEW OF PROPOSED METHODOLOGY

Proposed methodology allows a smart phone user to capture virtually any type of document whether an expense receipt, an insurance claim , driving license etc, and deliver it to needed destination via cloud. It is operating as a kind of mobile fax .As we are using mobile as well as cloud for this purpose we can call this system as faxing using mobicloud. Steps involved in Proposed System:-

1. Sender will capture an image/ document using Camera.
2. Encrypt the document.
3. Upload/Push encrypted data on to cloud.
4. Receiver will connect to cloud
5. Retrieve the document/data.
6. Decrypt the document.
7. Get original.

3. DATA FLOW DIAGRAM

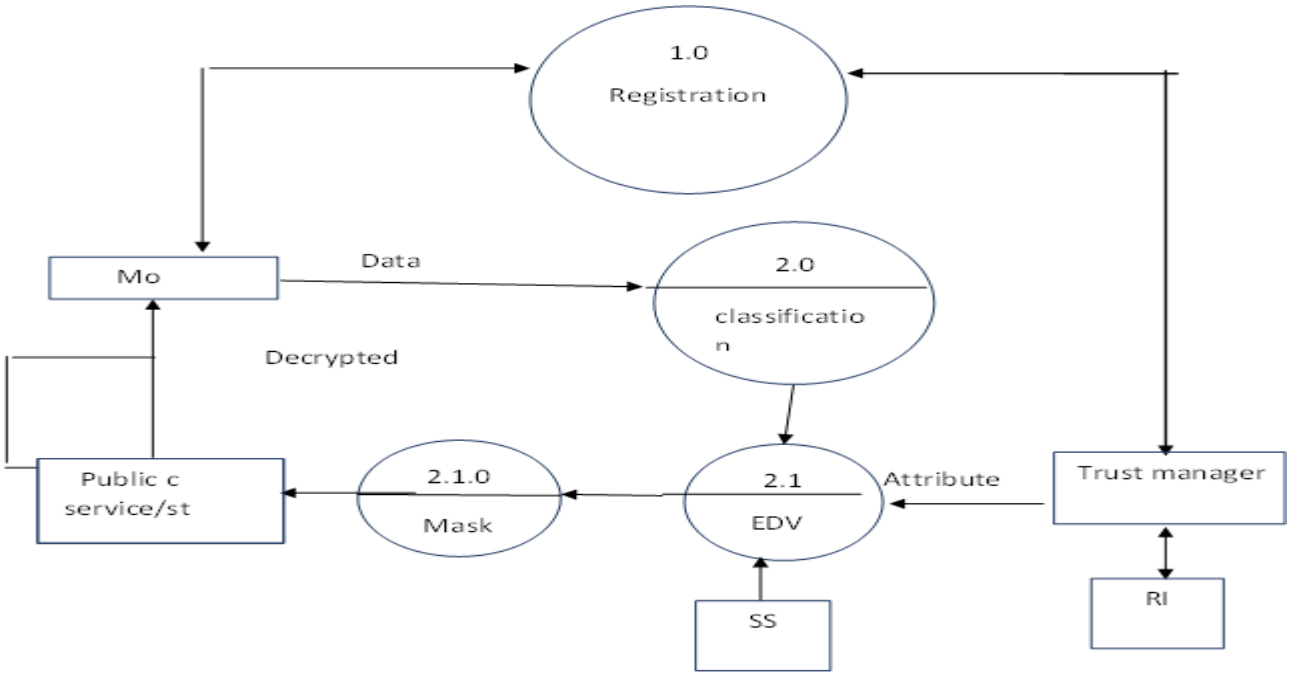

Fig 3.1: Data flow diagram

The Technical basis for the software can be described according to two parameters.

- Software Details
- Hardware Detail

4. SOFTWARE DETAILS:

- **Android SDK:**

  Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

- **SQLite**:

  SQLite is an in process library that implements a self-content, zero configuration,severless,transactional SQL database engine, The source code SQLite exists in the public domain and it free for both private and commercial purposes.

- **NetBeans:**

  The NetBeans is a tool which provide reliable and flexible application architecture. Your application does not have to look anything like an IDE. It can save you years of development time. The NetBeans platform is a generic framework for swing application.

- **Eclipse:**

  In computer programming Eclipse is an integrated development environment(IDE).It content workspace and an extensible plug-in system for customizing the environment. Written mostly in java, eclipse can be use develop application.

a. HARDWARE DETAILS:

The hardware requirements for the project are:

i. Android mobile

ii. RAM

iii. Hard Disc

b. ADVANTAGES:

5 It is flexible.

6 As this software is flexible modification can be done easily.

7 It is user friendly.

Our project is divided into five modules which are as given below:

5. USER REGISTRATION:

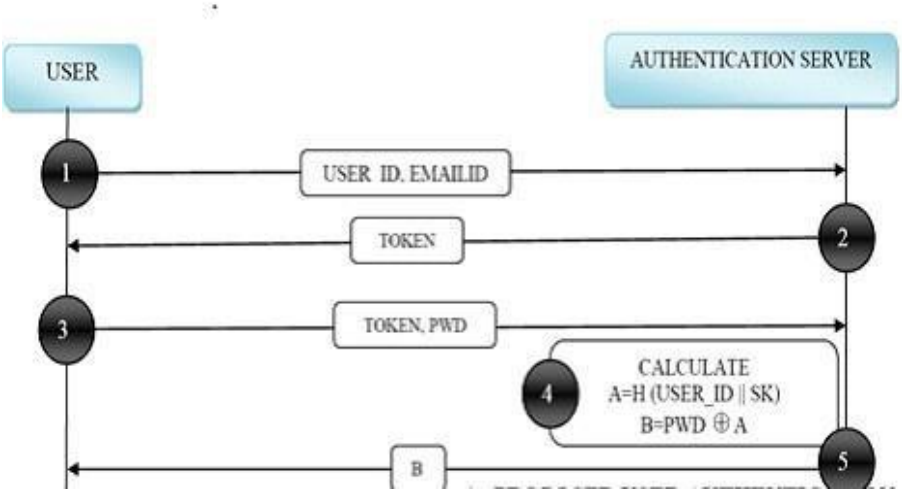

Fig 5.1 :- Registration Phase

Whenever user wants to access cloud resources, user has to register first on to the cloud.Following are the steps to register on the cloud.

1) User has to provide valid Email-Id and password to the authentication server.

2) Authentication server checks the Email-Id against the availability of that Email-Id. Email-Id should not repeat or match with existing users Email-Id.

3) After checking the availability of Email-Id, The authentication server sends a token to the users Email-Id.

4) User checks the Email and token send by the authentication server.

5) User enters the token value for further authentication and confirm for the registration.

6) After getting the valid token value, Authentication server send message of successful registration to the user.

a. ONE TIME KEY GENERATION:

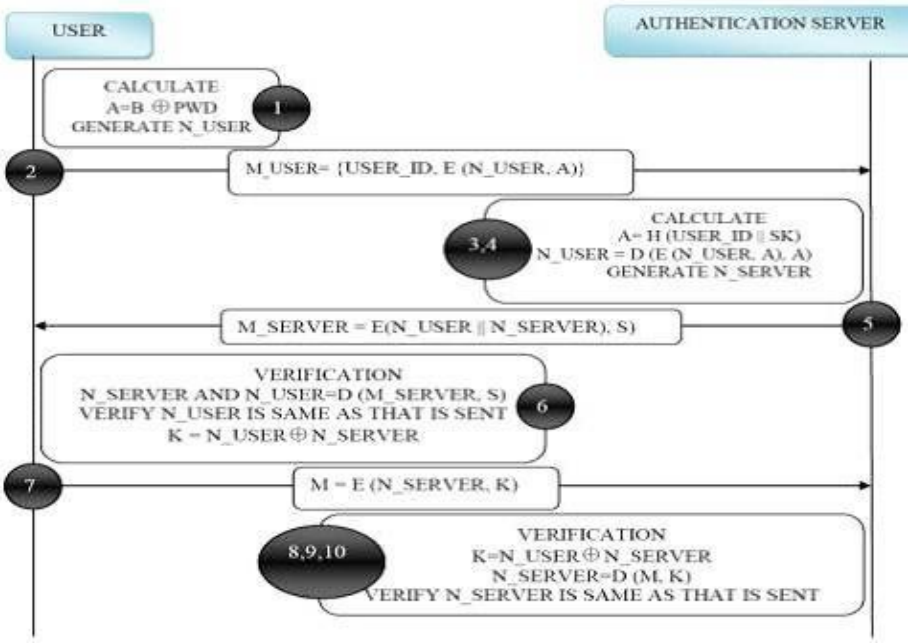

Fig 5.2: One time key generation.

When user wants to access resources on the cloud, then user should login on to the cloud. Following are the steps to login on to the cloud.

1) User should enter Valid Email-Id and password in his login interface. Users system computes the secrete key using stored values, which was already provided by the user at the time of registration.

2) The authentication server checks the user-id and password provided by the user with the user-id and password which was provided by the user at the time of registration.

3) After matching the user-Id and password authentication server generates the dynamic token from hash table and send it to the users Email-Id for STEP-2 authentication.

4) User checks his Email for getting the dynamic token for further authentication.

5) User has to enter the token value for STEP-2 authentication

6) Authentication server matches the token with the dynamic token which was send by itself.

b. CLOUD ADMIN:

If any user enter wrong user name and password then it is not accessible it is done by cloud admin. It checks the enter user name and password is register on cloud or not if register then it is accepted otherwise rejected.

Here when user will upload any data ex: images, pdf file…. to the cloud then that data will get encrypted and save in the cloud, that whenever user wants any data then he can easily access from his mobile.

c. ENCRYPTION DECRYPTION ALGORITHM:

In this module we are encrypting the Data or file which you want to send before uploading on cloud. We are performing encryption algorithm .This will be carried out at sender side only. After encryption encrypted data will be uploaded on cloud. After transmitting key sender will send one private key to receiver for performing decryption at receiver side via SMS. After getting notification receiver will retrieve the encrypted data from cloud using private key receiver will decrypt the data and get original data.

The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the ciphertext. The number of cycles of repetition are as follows:

- 10 cycles of repetition for 128-bit keys.
- 12 cycles of repetition for 192-bit keys.
- 14 cycles of repetition for 256-bit keys.

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

The AES algorithm consists of following phases:

1. **Key Expansion.**Round keys are derived from the cipher key using the Rijndael's
2. **Initial Round.AddRoundKey—**each byte of the state is combined with the round key using a bit-wise operation.
3. **Middle Rounds.**Nr = 1 till Nr-1 Repeatedly perform the following transformations:

- **SubBytes—**a non-linear substitution step where each byte is replaced with another according to a lookup table.
- **ShiftRows:** A transposition step where each row of the state is shifted cyclically a certain number of steps.
- **MixColumns: A** mixing operation which operates on the columns of the state, combining the four bytes in each column.
- **AddRoundKey:** Same as described above.

4 Final Round (no MixColumns)

- **SubBytes:**Same as described above.
- **ShiftRows: S**ame as described above.
- **AddRoundKey:** Same as described above.

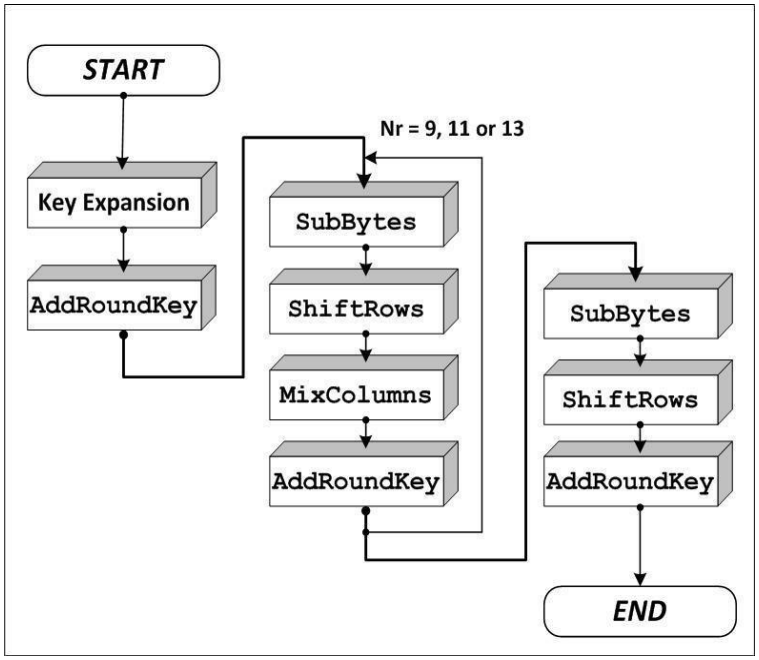

Fig5.4 : AES Flow Chart

6. Coding

```
package com.mobicloud; importjava.math.BigInteger;
import java.security.SecureRandom; import android.app.Activity;
import android.content.Intent; import android.os.Bundle;
import android.telephony.SmsManager; import android.view.View;
import android.view.View.OnClickListener; import android.widget.Button;
import android.widget.EditText; import android.widget.Toast;
import com.mobicloud.database.MyDBHandler;
public class GenerateRandomPassword extends Activity
{
        Button btn_randomPassword, btn_Login; EditTextet_userName, et_pass;
        String random, ran_dom; String result;
        // MyDBHandlermyDBHandler;
        @Override
        protected void onCreate(Bundle savedInstanceState)
        {
                // TODO Auto-generated method stub super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_generate_random_password);
                // ref
                et_userName= (EditText) findViewById(R.id.etUserNameCloudPage); et_pass = (EditText)
                findViewById(R.id.etPassCloudPage);
                        btn_randomPassword=(Button) findViewById(R.id.btn_GeneratePassCloudPage);
                btn_Login = (Button) findViewById(R.id.btn_LoginCloudPage);
btn_randomPassword.setOnClickListener(new OnClickListener()
                {
                        @Override
                        public void onClick(View arg0)
                        {
Generated
// getting mobile from username inorder to send  random
                // password
                        myDBHandler = new MyDBHandler(GenerateRandomPassword.this,
                        null, null, 1);
        result =
myDBHandler.retieveMobileNo(et_userName.getText().toString());
```

```
                    // call to generateRandromPassword() method ran_dom = generateRandromPassword();
                            }
                    });
        // call to sendSMS() method sendSMS(result, ran_dom);
                    btn_Login.setOnClickListener(new OnClickListener()
                    {
                            @Override
                            public void onClick(View v)
                            {
                                    if (et_pass.getText().toString().equals(ran_dom))
                                    {
                                            Toast.makeText(getApplicationContext(),
                                                    "Cloud Login Successful..",
                            Toast.LENGTH_SHORT)
                            .show();
            Intentintent=newIntent(GenerateRandomPassword.this,Cloud.class);
            intent.putExtra("KEY_NAME",et_userName.getText().toString());
          startActivity(intent);
        wrong password..",
}
                    });
}
        }
        else
        {
        }
    Toast.makeText(
                    getApplicationContext(), "Cloud Login Failed.."
                            + "\n You entered Toast.LENGTH_SHORT).show();
            public String generateRandromPassword()
            {
                    random = new BigInteger(32, new SecureRandom()).toString(32); return random;
            }
            private void sendSMS(String mobileNo, String message)
```

```
        {
//                  Intent smsIntent = new Intent(Intent.ACTION_VIEW);
//                  //smsIntent.setData(Uri.parse("smsto:"));
//                  //smsIntent.setType("vnd.android-dir/mms-sms");
//                  smsIntent.putExtra("address", mobileNo);
//                  smsIntent.putExtra("sms_body", message);
//                  try
//                  {
//                          startActivity(smsIntent);
//                          finish();
//                          Log.i("Finished sending SMS...", "");
//                  }
//                  catch (android.content.ActivityNotFoundException ex)
//                  {
//                          Toast.makeText(GenerateRandomPassword.this,"SMSfaild, please try again later.",
Toast.LENGTH_SHORT).show();
//                  }
                    try
                    {
                        SmsManagersmsManager=SmsManager.getDefault();
                        smsManager.sendTextMessage(mobileNo, null, message, null,
                        Toast.makeText(getApplicationContext(),"SMS sent..", Toast.LENGTH_SHORT).show();
                    }
null);
                  catch(Exception e)
                  {
                          Toast.makeText(getApplicationContext(),"SMS faild due to :"+"\n
1. No network."+"\n 2. Not enough balance.",Toast.LENGTH_LONG).show(); e.printStackTrace();
                  }
        }
}
```

7.1 HOME PAGE

Fig 7.1: Home page

The above snapshot shows the homepage of the application's registration page where the user can newly login into the account or can sign in into an already registered account.

7.2 USER REGISTRATION

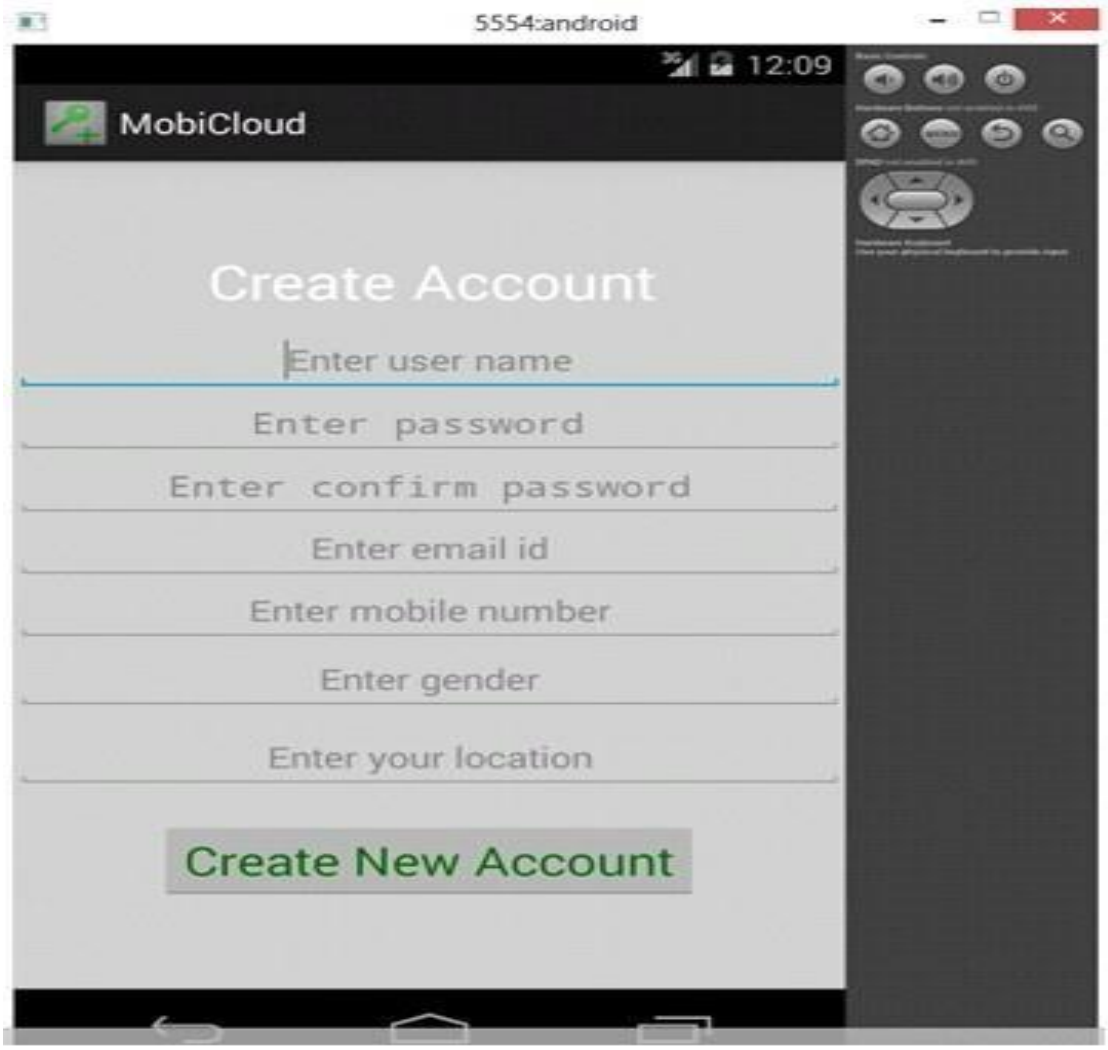

Fig 7.2: User registration

The above snapshot shows the user registration page of the application where the new user registers for the new account.

7.3 USER LOGIN

Fig 7.3: User login

The above snapshot shows the user login page of the application where the user of the application can now enter id and password for login.

7.4 ONE TIME KEY GENERATION

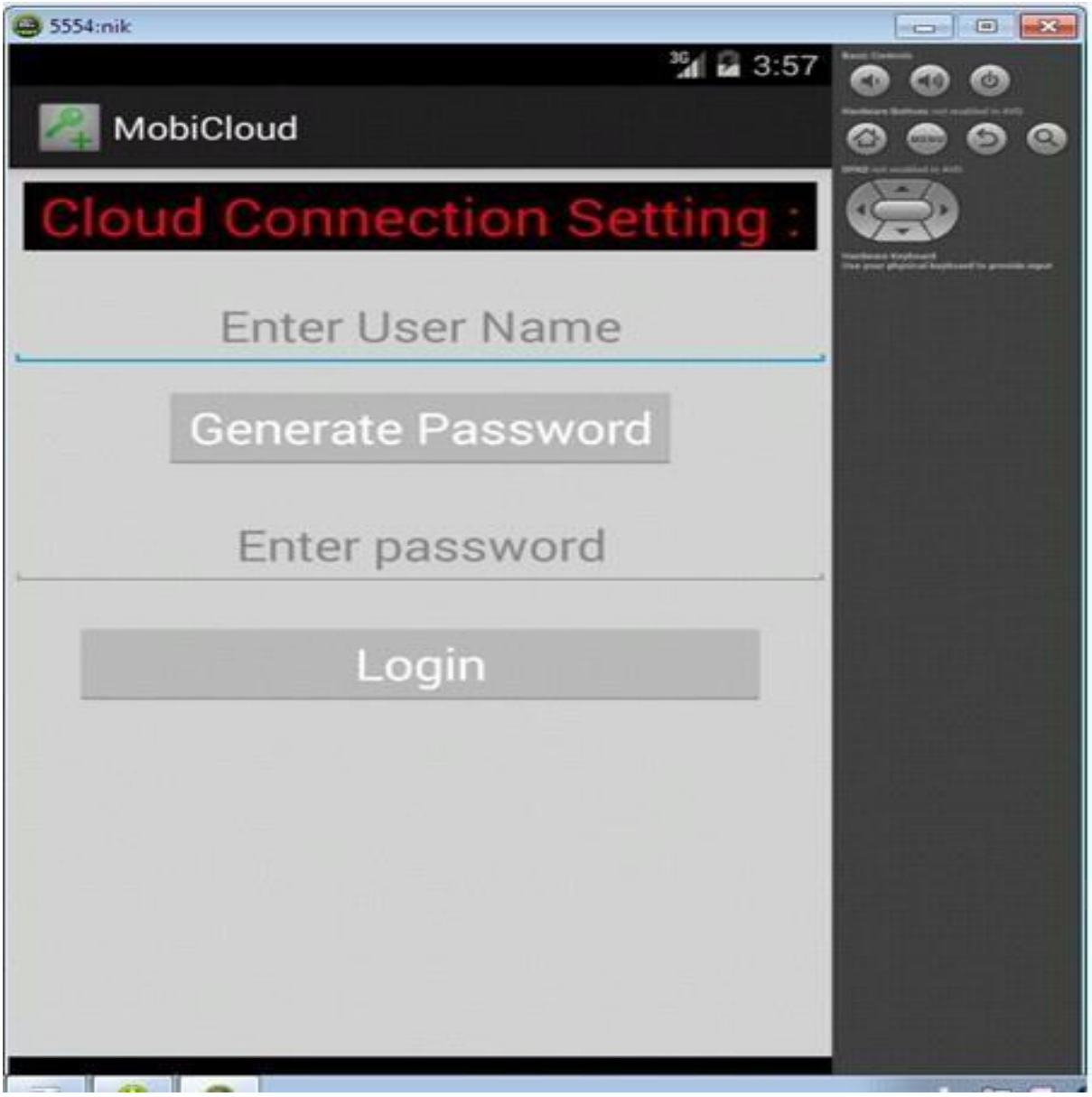

Fig 7.4: One time key generation

The above snapshot shows the one time key generation pattern in which a user has to enter a user id to receive a alphanumeric password in his mobile, this varies each time when the user login into account.

7.5 CLOUD CONNECTION

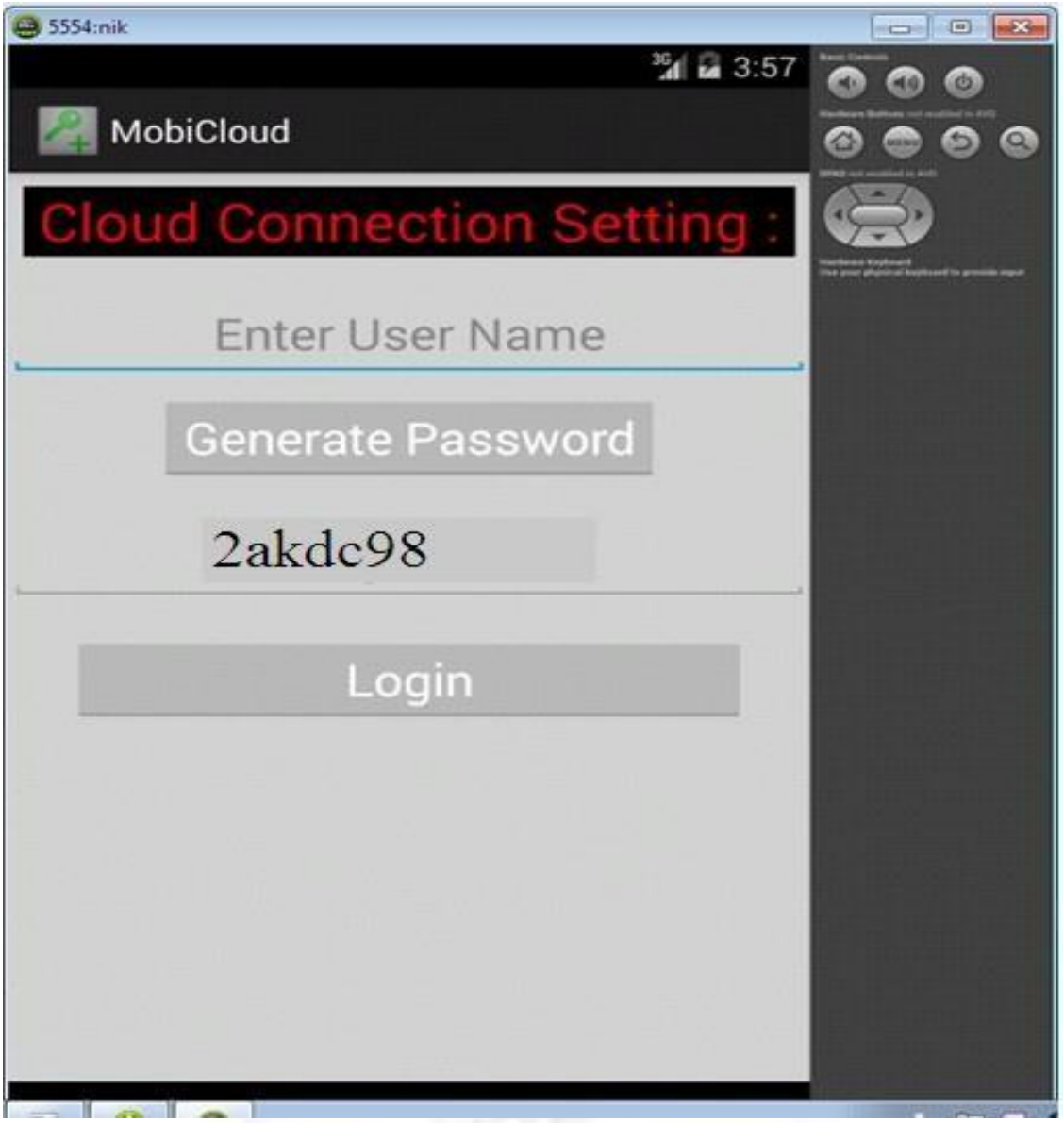

Fig 7.5: Cloud connection

The above snapshot shows the connection with the cloud in which a user receives alphanumeric values after clicking the generate password button which used for security purpose during the exchange of information.

7.6 LOGIN SUCCESSFUL

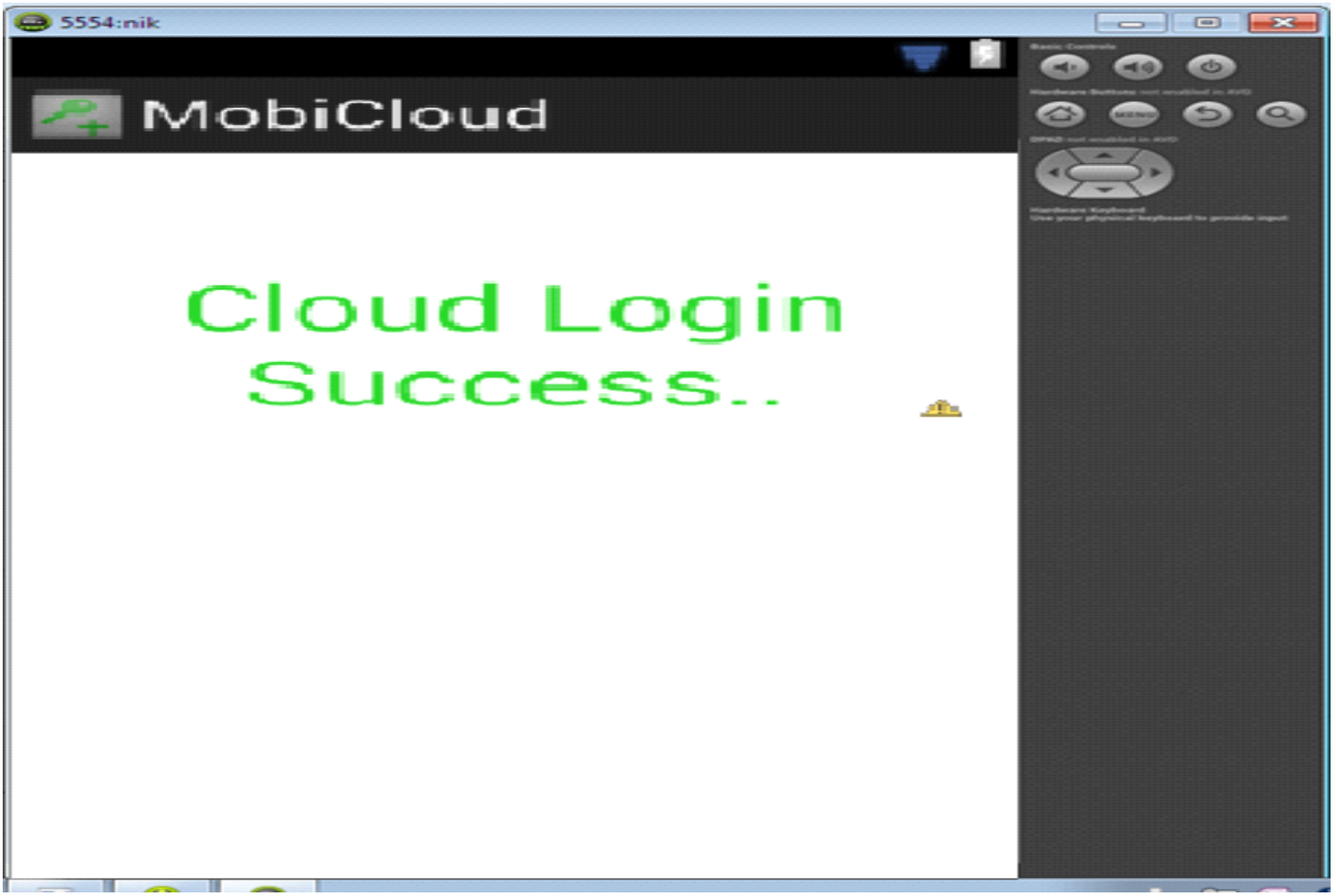

Fig 7.6: Login successful

The above snapshot shows the successful login page whenafter the user has login the application the cloud login is success.

7.7 IMAGE BROWSING

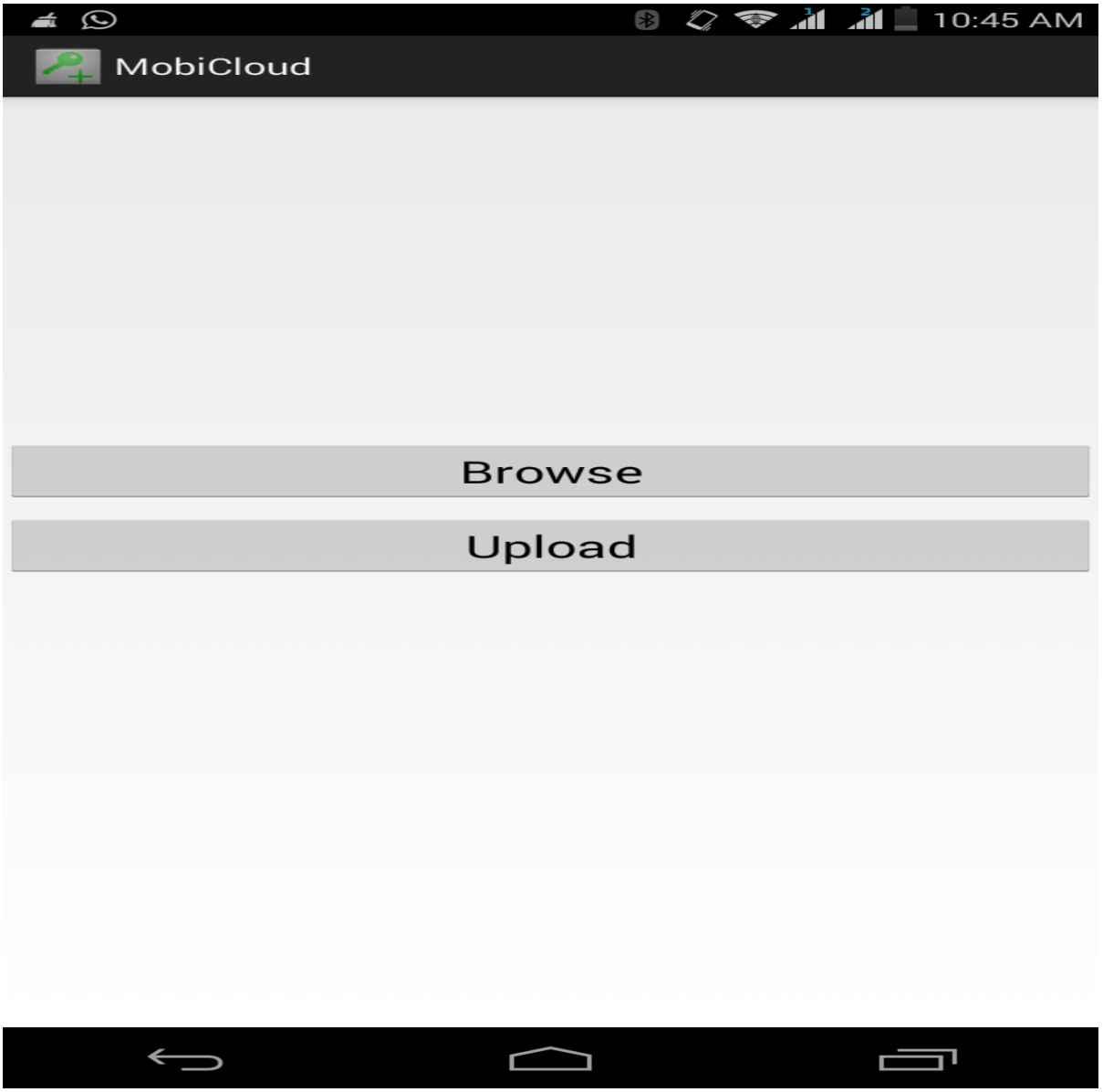

Fig7.7 : Image Browse

The above snapshot shows that if user wants to upload any images to cloud then user will select the imges from mobile.

7.8 SELECTED IMAGE

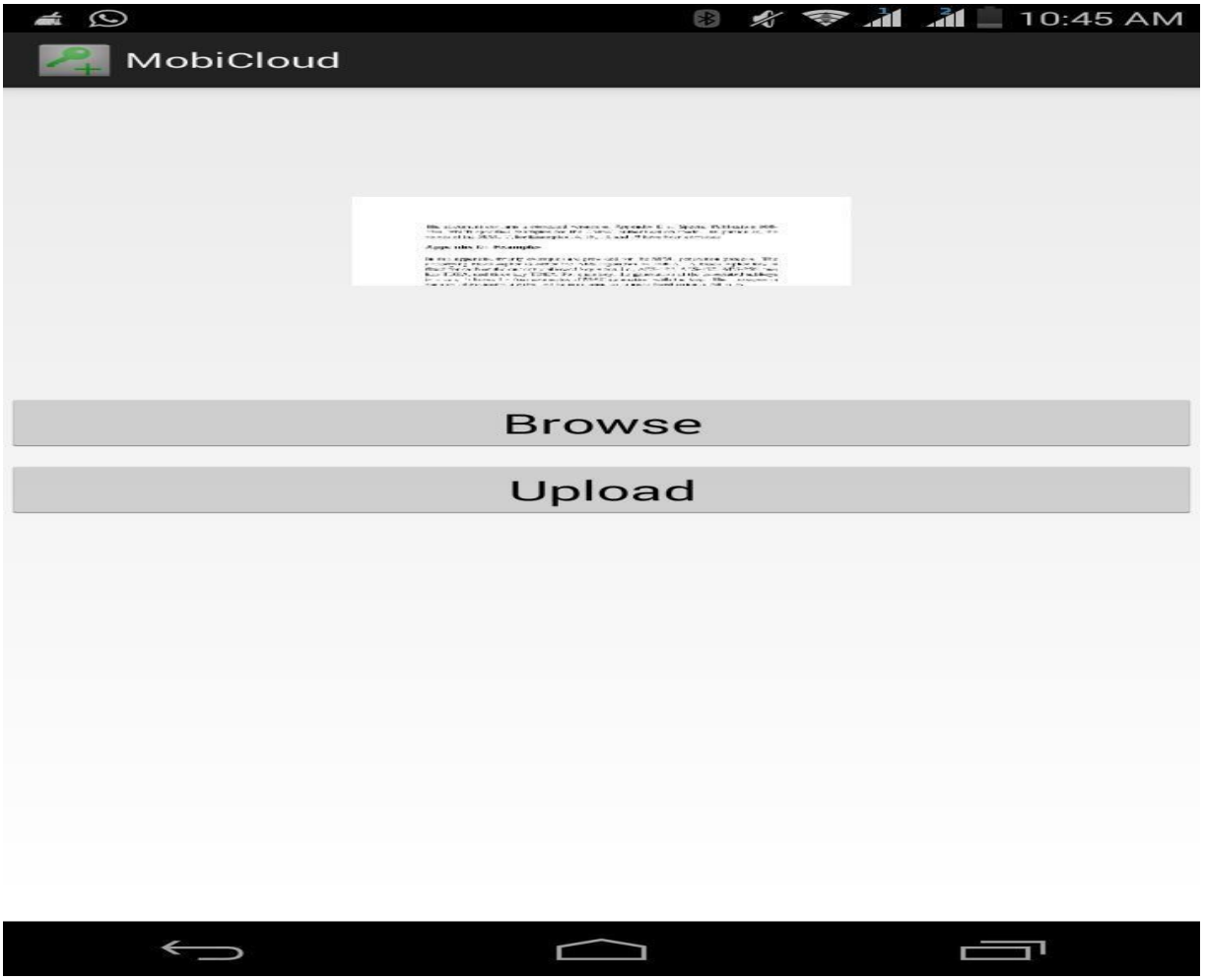

Fig 7.8: Selected Image

The above snapshot shows the selected image and now user will upload that image to cloud.

7.9 TEXT OPERATION

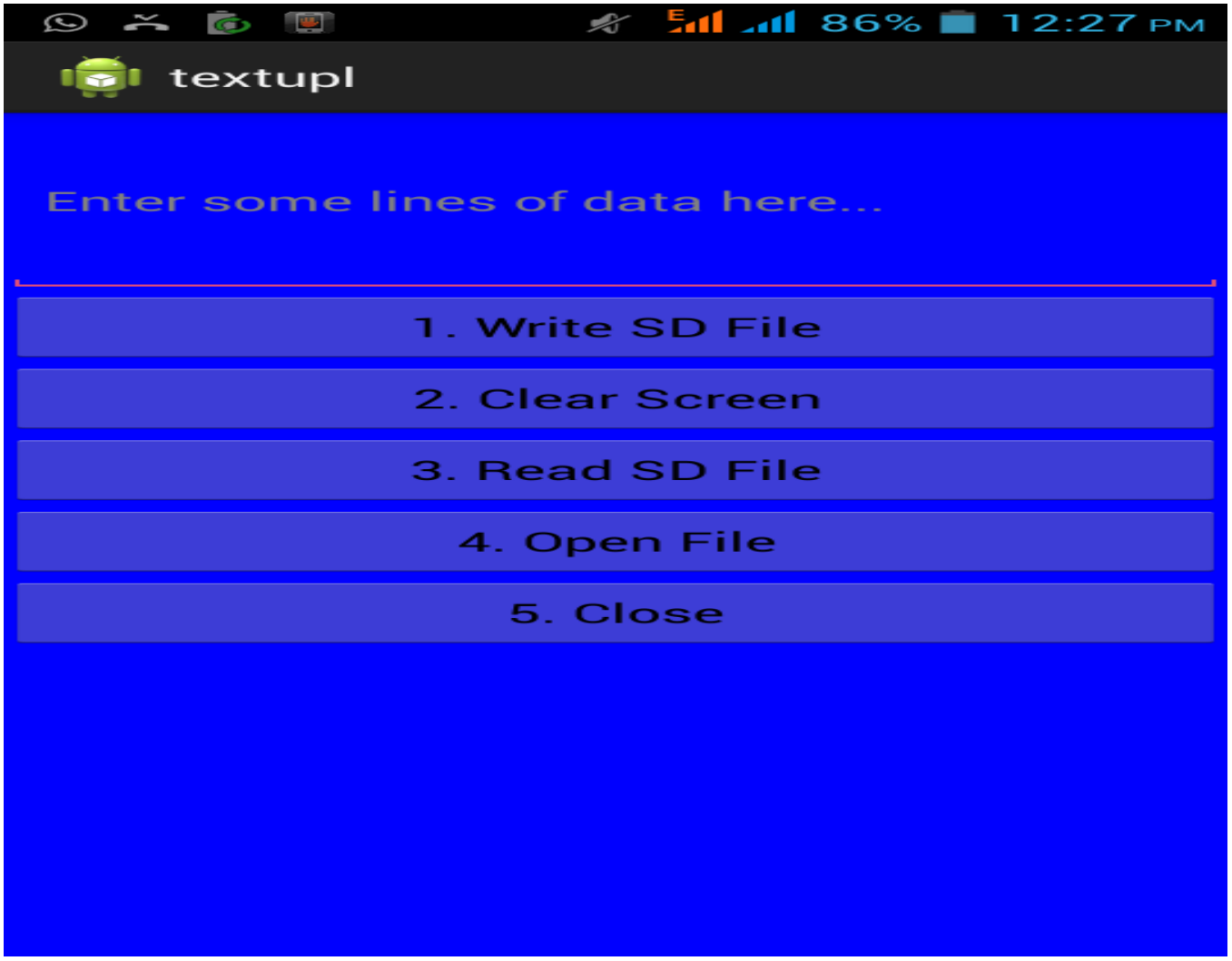

Fig 7.9 Text Operation

The above screenshot shows the options available in the text related access and retrival of data from and to the cloud.

7.10 INSERTION OF TEXT

Fig 7.10: Insertion of Text

In the above snapshot the user will enter some text in the given field and this text then will be saved to the cloud.

7.11 RETRIVAL OF DATA

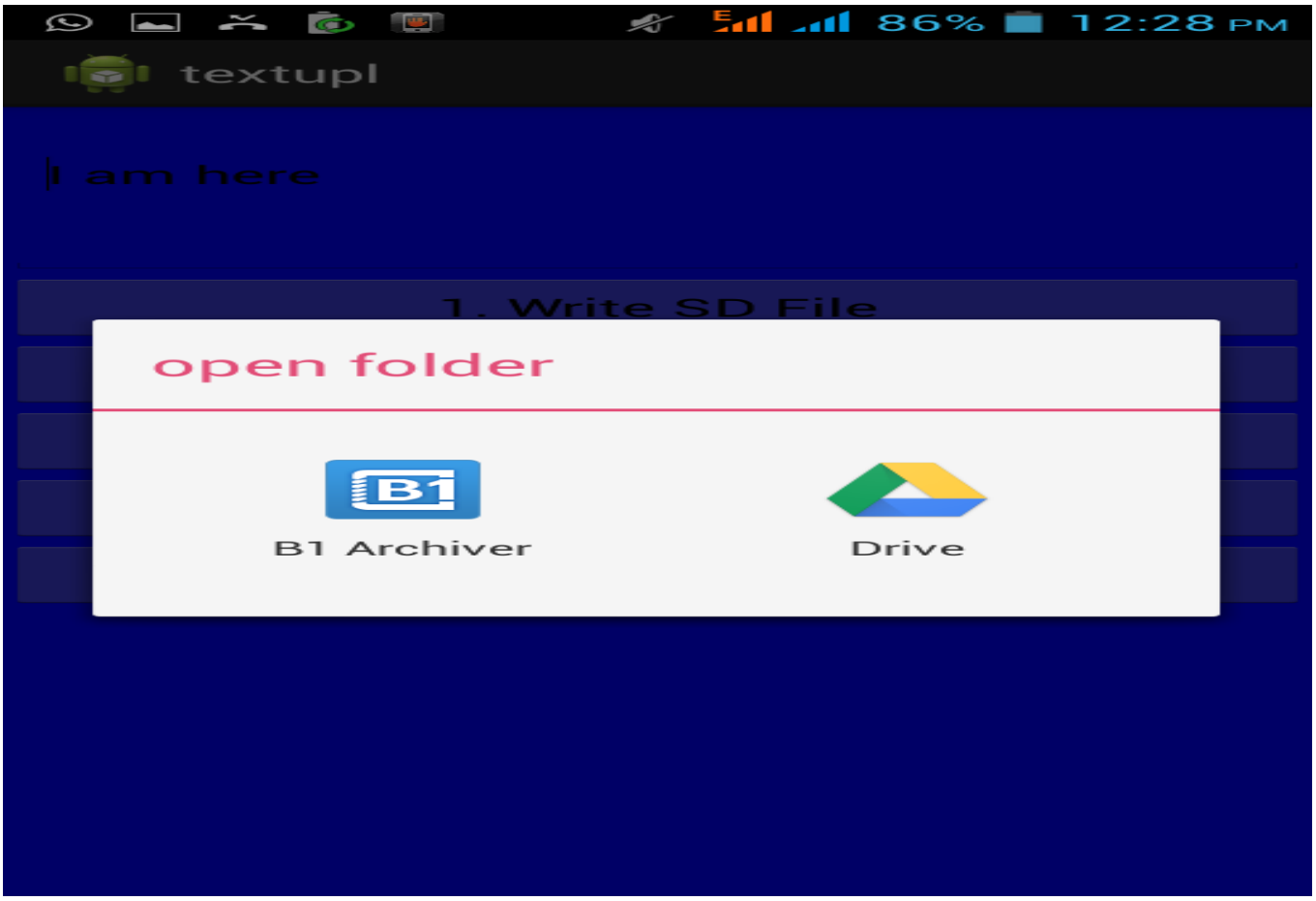

Fig 7.11: Retrival of data

From the above snapshot of the application the user can retrieve the data from the cloud.

7.12 SELECTION OF FILE

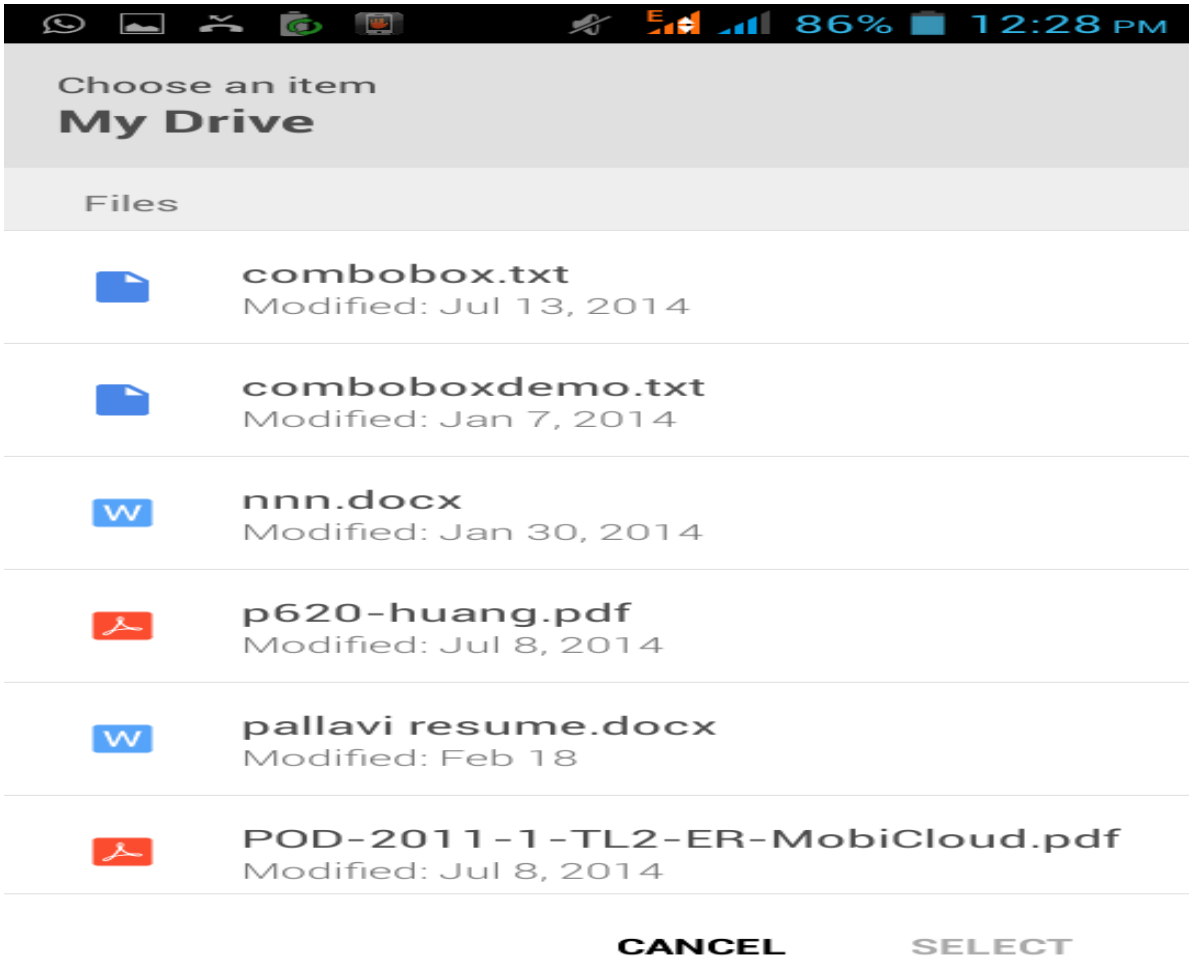

fig 7.12:Selection of file

In the snapshot above the user will select the particular file.

7.13 DOWNLODING THE FILE

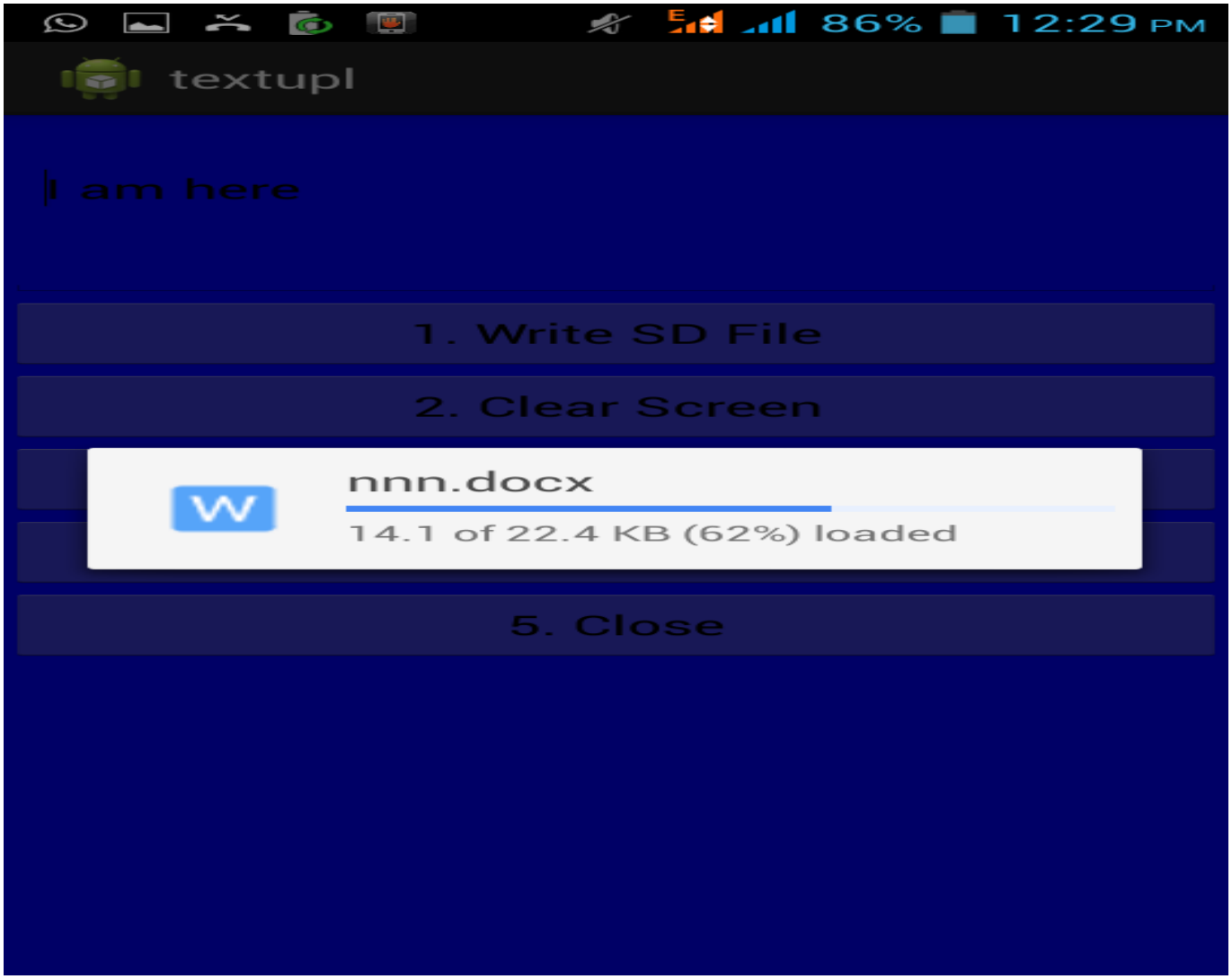

Fig 7.13: Downloading the file

The snapshot shows the downloading of particular selected file.

8. APPLICATIONS

So far, there are a large number of mobile applications which have taken the advantage of Mobile Cloud Computing. These applications have made a huge impact on the market and their value has increased a lot. Here are few of the applications supported by Mobile Cloud Computing.

- **Mobile Commerce:**

  Mobile Cloud Computing made life easy for commerce by providing mobile commerce (m-commerce) using mobile hand held devices. The applications like finance, shopping, ticketing etc are facing some serious challenges because of low bandwidth, complex mobile architecture and serious security risks. However, the Mobile Cloud Computing provides the solution for these challenges by integrating m-commerce applications into the cloud environment.

- **Mobile Learning :**

  The hybrid of electronic learning and mobility gave birth to mobile learning (m-learning). However, issues like high price of mobile devices and bandwidth, low network transmission rate and lack of electronic educational resources are proving to be main obstacle in the way of m-learning. But the cloud provides large storage and high processing capabilities, which introduce the idea of cloud based m- learning and eliminate the barriers of m-learning.

- **Mobile Healthcare:**

  The mobile medical applications have so many limitations like, small storage capacity, privacy and security of data etc. However, Mobile Cloud Computing eliminates the issues of traditional medical applications used for medical treatment. Therefore, the m-healthcare helps the mobile users to access medical resources in efficient way because of the availability of on-demand services on the cloud.

- **Mobile Gaming:**

  Mobile gaming (m-gaming) is one the most popular service for the cloud service providers in terms of revenue generation. Usually, all the mobile games require high computing resources like, graphic rendering. However, in the cloud the m-game can off-load game engine which requires graphic rendering to the cloud server. This way, mobile users can only interact with the screen displays on their devices while all other computation is being performed at the cloud servers.

- **Security:**

The data is more secure as we used AES algorithm and DES means Advanced Encryption Standard and data decryption standard. (AES) -128 bits is used for encrypting thev mobile user's data in the cloud. AES is a symmetric block cipher that is intended to replace Data Encryption Standard (DES) for a wide range of applications.

- **Remotely accessible:**

The data is access from anywhere as cloud is present centrally and we can access data from office, home, on the road, client side etc. we cannot need to carry laptop when we go for conference and for any presentation because all data which we want are present on cloud and we can access those data from cloud .

- **Data sharing:**

The data is easily share from mobile to cloud and cloud to mobile is because we create a virtual space on cloud by registering mobile user and on those cloud we store large amount of data. As we know there are two types of cloud private cloud and public cloud the data sharing between private and public cloud is possible.

- **Enhanced processing power and data storage space:**

As we know we cannot store large amount of data into mobile that's why we use cloud for storing large amount of data. Cloud computing can be defined as the trend in which resources are provided to a local client on demand basis, usually by means of the internet connection .Cloud computing can be termed as model of information processing, storage. The Mobile Cloud Computing provides mobile users a platform to store large amount of data on the cloud. The storage space is always a bigger concern for the mobile users which Mobile Cloud Computing eliminates. The mobile users get storage facility by connecting with cloud through wireless network.

- **Long Battery Output Time**

  Battery output lifetime has always been a problem in advance mobile device like smart phones, Tablet pc's especially when they execute heavy applications. MCC facilitates the user by executing heavy and time taking applications in the cloud using cloud resources.

  The execution of applications at cloud end significantly saves the mobile devices battery power.

- **More Data and Application Reliability :**

  Using MCC, the data reliability is increased to a great extent because data is stored and backed up on different servers in the cloud. This advantage of MCC gets rid of the chances of losing data and application on the user's mobile device.

- **Scalability :**

  The cloud service providers can expand their cloud services with less effort and modification to infrastructure. They can easily add applications and services without any concern about resource usage.

  The aim of the framework is to design and implement a secure data processing framework for mobile cloud applications. The proposed framework utilizes decentralized approach and the proposed multi-tenant data management divides the data in to two security levels critical data and non-critical or normal data. In this framework, attribute based encryption scheme with the Advanced Encryption Standard (AES) -128 bits is used for encrypting the mobile user's data in the cloud.

  Advanced Encryption Standard not only assures security but also improves the performance in a variety of settings such as smartcards, hardware and software implementations.

  Cloud computing has reached a maturity that leads it into a productive phase. This means that most of the main issues with cloud computing have been addressed to a degree that clouds have become interesting for full commercial exploitation. This however does not mean that all the problems listed above have actually been solved, only that the according risks can be tolerated to a certain degree. Cloud computing is therefore still as much a research topic, as it is a market offering.

Papers:

[1] Dijiang Huang, Zhibin Zhou, Le Xu, Tianyi Xing, Yunji Zhong, "Secure Data Processing Framework for MobileCloudComputing"2010.vol no-41,pp 391-40.

[2] Shaikh,F.B;haider "security threats in cloud computing,Internationalconference"2011.

[3] K.Kumar,Y.H.LO,"Cloud computing for mobile users",IEEE journal on cloud computing,2012 , vol no-22,no.11,pp.612-613.

[4] H.Cancepa, D.Lee,"A virtual Cloud computing provider for mobile devices",June 2010.

[5] W.Itanai.A.kayssi,A.Chehab,"Energy-Efficient Incermental Integrity for security storage in mobile cloud computing',Dec2011, vol no-7,no-4,pp523-552.

[6] Fei Li,Rahulamathavan,YRajarajan,M law "Attribute based encryption scheme for mobile cloud computing",2013.

Website:

[1] http://Andriodhive fetch data.com

[2] http://andriodeexample.com

[3] http://www.javatpoint.com