DATA OBSCURITY IN DEVICES DURING SECURED WIRELESS COMMUNICATION

Mr. Rishabh Mishra[1] , Dr. Rakesh Sharma[2]
[1] M.tech student (Computer Science)
[2] Associate Professor, Department of Computer Science,
Noida International University, Greater Noida , Uttar Pradesh, India

***Abstract*— This paper discusses about the software which is developed for the Android mobile devices widely used in day to day life activities. In this era of rapid online communication it is desirable to exchange classified multimedia content, securely across any geographical coordinate. This could be essential in exchange of classified communications like private images, criminal investigations, business communications , and the likes. Although there are some similar technologies developed for this purpose, there is main issue of security. By using this software we will send the image through parse cloud service using Android. The communication is not just encoded over the transmission channels and cloud but the exclusive feature of Data Obscurity in device ensures the content security at the user's device / handset level also, thus security can achieved at all levels. For providing security at the transmission channels and cloud, we are using AES algorithm for encryption and decryption. This paper discusses the proposed system, overview of the design, the various modules of the system and its implementation.**

***Index Terms*— Data Obscurity, Parse Cloud Service, Security, Encryption, Decryption.**

I. INTRODUCTION

At present smart phone and social media boom has rocked the world. Both has enable a user to stay connect with their friends at any time, from anywhere. As the uses of such technologies are increasing, security of the communicated data has become the main concern. We often get the news about email hacking, identity theft, spoofing etc which has created a fear among users regarding their data security. Proposed application is using the cloud services and encryption at cloud level to counter the emerging security threats. Simultaneously, when mobile becoming so popular, cloud computing is also gaining popularity. Development of cloud computing offers a advantage that, service providers no longer need to worry about resource management. Resources are managed by cloud providers, and service providers can use resources depending on their demand.

As "cloud" is a collection of tangible super computers spread across the globe, authentication and authorization for data access is gaining more importance.

The proposed methodology suggests the encryption of the image to be uploaded on the cloud. The integrity and confidentiality of the data uploaded by the user is ensured doubly by encrypting it and also providing access to the data only on successful authentication. In addition, users can access data and services anytime and anywhere. This lets users share data more easily than before. Users can access the same data in the same way from any device. In our application we are using hybrid application of Mobile and cloud computing to ensure data obscurity.

As an additional security feature, while sending, the captured image would be stored in the external memory at the sender's device but it would not appear in the image gallery and at the receivers end, the image will not be saved anywhere in recipient's device by the Android application. Thereby security is achieved at all levels.

Literature Survey:

Recent Market shows that multimedia communication like image and video transfer is very important in our day to day life. There are many ways through which we can transfer them from source to destination. As these all have some advantages also has certain disadvantages related to them. Current security scenario has created a doubt in user mind regarding online exchange. After Gmail hacking scandal it has emerged in a survey conducted in U.S.A by Pew Research Center that the public feels most secure using landline phones, least secure on social media as shown in the Fig 1.

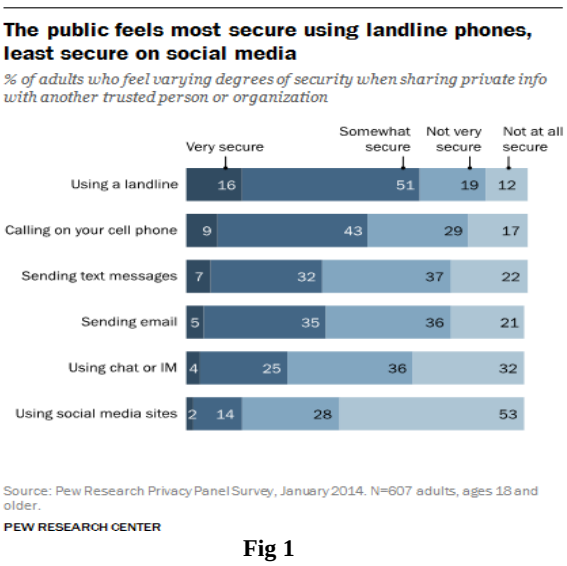

Fig 1

A. Using social networking sites likes Whatsapp:

F-Secure analyzed that WhatsApp, a popular messaging service was targeted by a particularly notorious Android application which targeted its users. WhatsApp is a messaging service that enables you to chat and share media files with your friends for free. It's growing popularity is attributed to free messaging service as people don't have to pay for text messages. Once this nasty application is installed, according to F-Secure, it uploads the user's WhatsApp conversations to another website from where any user having your phone number can buy its copies.

BalloonPop2

BalloonPop2 is the app that raises the concern, F-Secure reported that BalloonPop2 was available in Google Play for sometime, but later it was removed and currently it is available on the developer's website. Once you install the game, it works—though it is a dull, objectionable affair. According to F-Secure this app scans the details of your WhatsApp account behind the scenes and also checks the serial number of your SIM card, probably to map your WhatsApp account to a phone number. The app then copies the contents of the two directories of WhatsApp:

1) The contents of Profile Pictures folder.
2) The files ending with ".db.crypt" stored in the WhatsApp/Databases/.

BalloonPop2 then uploads these files to the WhatsAppCopy website and from here anyone who knows your phone number can search for these files. If they want to get a copy of your conversations, all they have to do is to buy it from WhatsAppCopy. Whether these files are readable or not ,is not confirmed. Security Watch is investigating whether the files uploaded by BalloonPop2 swipes are encrypted or not.

WhatsAppCopy operation seems to be ethically wrong and illegal, but as per the WhatsAppCopy website this whole operation is engineered as a "backup" service. The concept behind its operation is that you would install the game on your device and purchase your own records. This is a flimsy excuse, considering that this app copies your data so, it should be sold as a backup app, which is not the case, and also it's name creates confusion with other Android games having similar names. It is clearly meant to mislead. At best, BalloonPop2 and WhatsAppCopy, both fall into the grey-area of surveillance apps. These apps captures the text conversations and calls, and meant for the people interested in spying on others. At worst, this whole operation is nothing but a glaring attempt to steal your data.

WhatsApp itself recommends: "If you do not want an image or information about yourself to be accessible to other people, DO NOT put it in your WhatsApp profile".

B. Using Email:

Is Gmail, an email service provided by Google safe for work? For most of us, the answer is yes, except under circumstances where Gmail may not be an appropriate option. The default settings of Gmail provide fairly reasonable security. The data that users can view in Gmail are encrypted as per the industry-standard 128 bit encryption. The Gmail data is send by the Google to its users through transport layer security 1.1. At the user's end, authentication of encrypted data is done using the SHA1 cryptographic hash function and then decoded by the ECDHE_RSA key exchange mechanism. Two-factor authentication feature of Google and strong passwords on secure machines provides perfect safety while working on Gmail. Technically the Gmail data in transit can be intercepted via infected machines and spoofed digital certificates and evident from the Google's transparency report, that Google complies with prosecutorial and other information requests cited by government.

You can decide yourself if using Gmail at work is a better option or not, by assessing the nature of communication you do on it. If you work as an anti-regime activist or involved with the cause that run against the government's interests in a country which actively surveil its own citizens, then it's better to avoid Gmail as a work option. Governments may summon Google for Gmail account information and in certain cases Google has to comply. Governments have enough resources and money to crack Gmail's encryption or they can spoof certificates, which enables them to impersonate Google and execute the man-in-the-middle attacks.

A number of cyber experts speculated that state-sponsored hackers of Iran compromised Diginotar, the Dutch certificate authority last year for spying on their own citizens. It is not clear yet this was the case, but compromises of Diginotar and another security provider, Comodo in the past revealed that such a threat exists. Even if states were not responsible for such attacks, someone was, and the compromise of a certificate authority clearly demonstrate that someone is impersonating someone or something else, which also implies that some user are unaware that they are sending data to or through a source that is not what it claims to be. More generally, if you frequently deal in very sensitive, confidential information of any kind, then you should avoid using Gmail, because such information are aggressively sought after by criminals, hobbyists, and the state-sponsored-hackers.

Of course, nobody wants their Gmail account get compromised or being surveil – regardless of what their work necessitate. There are some suggestions for those, who still want to continue with Gmail regardless of the nature of their trade:

1) Access and use the Gmail account from a secured and well-protected system equipped with a total security solution.

2) Users should avail the benefit of two-factor authentication feature of Google, which will help to protect against account hijacks.

3) Remember to logout whenever you are leaving your computer even for a short span of time, because all the security available in the cyber world would not protects against a malicious forwarding rule.

4) As always, keep the browser and operating system up-to-date and refrain from using insecure networks, like unencrypted public Wi-Fi.

II. PROPOSED SYSTEM

Our system allows a smart phone user to capture image, and deliver it to needed destination via cloud.

Steps involved in Proposed System:-

1. Users (Sender & Receiver) register themselves on http://www.parse.com cloud service for obtaining user id.
2. Sender logs in the Android application with user id and password of the account created on cloud.
3. After login, sender captures the image through device camera using Image Capture feature of the Android application.
4. The captured image gets encrypted by the Android application using AES algorithm.
5. The encrypted image gets further secured by an encrypted password key, set by the sender through the Android application.
6. The doubly-secured, encrypted image is saved by the Android application in only the external memory (SD card) of the device and not in the internal memory. The image is not listed in the Picture Gallery of device and it also cannot be viewed from the File Manager option of the handset.
7. Sender transmits the image to the recipient using Send Image feature of the Android application. In order to deliver the image the recipient's user id is required.
8. When the sender has completed the transmission, the image gets automatically omitted from the Android application as a security feature.
9. When the sender has sent the image, it gets saved on the cloud database in an encoded form which disallows it to be viewed even on the cloud.
10. The recipient shall be able to view the images using Received Images feature of the Android application using which all the images sent by different senders will be listed from cloud database.
11. In order to view the image, the recipient taps on the image icon of the received images and inputs the respective password in order to decode the image.
12. The decoded image is not saved anywhere in the recipient's device by the Android application.

III. MODULAR DESIGN

Our proposed system is divided into four distinct modules described as follows:

1. Login:

This module will enable the users to log in the Android application with their respective user id and password of the account created on cloud. We are obtaining and integrating free cloud service from a provider called http://www.parse.com. The exchanged application data like encrypted images and other credentials of the users shall be stored in the database tables located on cloud.

The users therefore shall have to register themselves on parse.com prior to using the application. The inputted user id and password in this module will be authenticated from the cloud data using 3G/GPRS connection only after which the user shall be permitted to login from the Android device.

2. Image Capture:-

This module will enable the user to capture the image after login, using device camera. The captured image shall get encrypted by the Android application using AES (Advanced Encryption Standard) algorithm for which we use classes from the javax.crypto package. The encrypted image will get further secured by an encrypted password key, set by the sender through the Android application.

The doubly-secured, encrypted image is saved by the application in only the external memory (SD card) of the device and not in the internal, physical memory. The image is not also listed in the Picture Gallery of device and it also cannot be viewed from the File Manager option of the handset.
The encoded image is displayed in the Android application in a viewable form till the time it is not transmitted to the sender.

3. Send Image:-

This module will enable the sender to transmit the encrypted image to the recipient using recipient's cloud user id. The image will be contained in a Java object and sent to the output stream. It will make use of the internet i.e. device's 3G/GPRS connection to reach the recipient. The delay in transmission depends upon the size of image along with network strength and bandwidth.

When the sender has completed the transmission, the image gets automatically omitted from the sender's Android device / application as a security feature. And it will be saved on cloud database in an encoded form which disallows it to be viewed even on the cloud because it can be decoded through the application only.

4. Received Image:-

This module shall enable the recipient to view the images sent by different senders. The images will be listed in a non-readable format from cloud database and in order to view them, the recipient taps on the image icon of the received images and inputs the respective password, set by the sender during encryption, in order to decode it.

Upon input of the correct password, which is authenticated from the cloud database, the image will be transferred and decoded on the recipient's device after which it becomes viewable. As a security feature, the decoded image will not be saved anywhere in the recipient's device by the Android application.

IV. TECHNICAL REQUIREMENTS

The Technical requirements for this application like hardware specification , software specification for mobile and laptop are listed in Table1.

HARDWARE SPECIFICATIONS (LAPTOP)	
CPU:	Intel Pentium i3
RAM:	4 GB
Hard Disk:	250 GB
Others:	USB Internet Data Card (preferably 3G)
SOFTWARE SPECIFICATIONS (LAPTOP)	
Operating System:	Windows 7 (64 – bit)
Technology (Front-end)**:**	JDK 7
Database (Back-end)**:**	Cloud supported at parse.com
Web Server:	--
Tools/IDE:	Eclipse Juno (64 – bit)
HARDWARE SPECIFICATIONS (MOBILE DEVICE)	
Handset Model:	Any
CPU:	Quad-core, ARM Cortex or Qualcomm Snapdragon
RAM:	2 GB
Physical (Built-in) Memory:	16/32 GB
Extendable Memory:	SD card, 16/32 GB
Camera:	8/16 Megapixel, LED Flash
Data Connection (3G/GPRS):	Preferably 3G
Others:	USB data cable
SOFTWARE SPECIFICATIONS (MOBILE DEVICE)	
Operating System:	Android 4.2 or higher

Table.1

IV. IMPLEMENTATION

For implementing this application we are using software like eclipse, jdk and java. We have done all GUI Using Netbeans and Eclipse and all programming has been done in Java. Our proposed system is divided into four distinct modules described as follows:

A. User Authentication:

The user authentication module on the client side involved the development of a login screen in the application. For this purpose, standard Graphical User Interface (GUI) that consists of buttons and textboxes were developed. The button is also associated with an action that sends the input parameters in the textboxes to the remote database via a web service.

At first user has to register on Parse.com with a userId and Password, these credential will be used for accessing our android application.

1.Sign In:

Sign In is an activity in which user authentication carries out by a system. When application starts after welcome screen you will see the screen as in Fig.2.

This screen asks you about your username and passwords. If you are registered user then you can provide your username and password but if you are not registered then you can register yourself using sign up.

The inputted user id and password in this module will be authenticated from the cloud data using 3G/GPRS connection only after which the user shall be permitted to login from the Android device

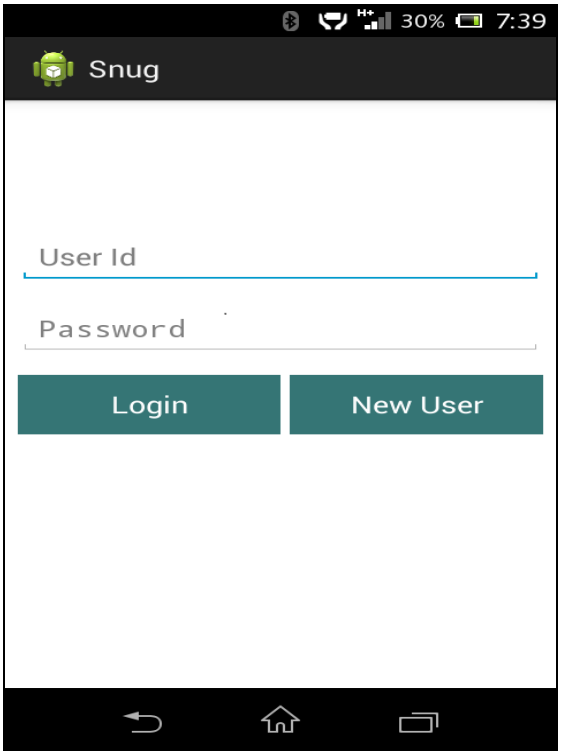

Fig 2.Log In Screen

2.Sign Up:

If user is not registered he can register using sign up. Sign up is best option for new users to get registered. You have to fill your all information as shown in fig 3,Then application will store his/her new username and passwords in the database and he /she becomes registered user.

After sign up whenever user want to login the application he can login with this username and password which are provided during sign up.

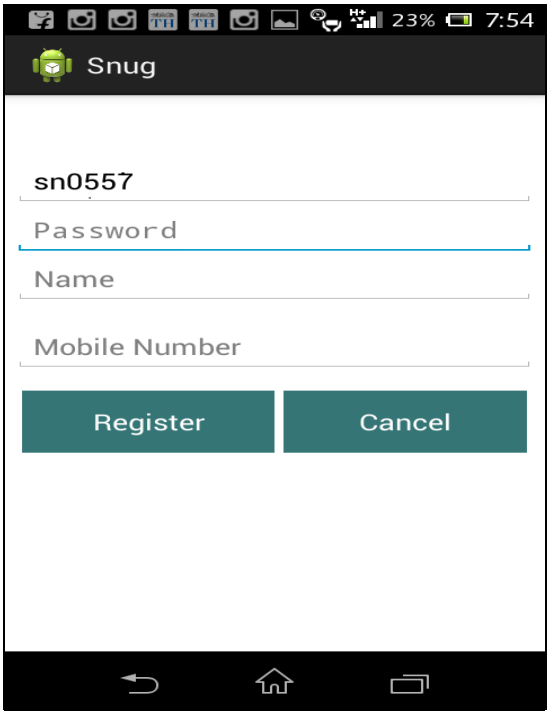

Fig 3.Sign Up Screen

B. Main Menu:

After authentication User will provided with main menu screen. As you can see in Fig 4.

The Main Menu involved the development of a screen in the application. For this purpose, standard Graphical User Interface (GUI) that consists of three buttons (viz Capture Pic, Receive image and Send) and a image icon were developed. These buttons are also associated with an action that sends perform the following function:

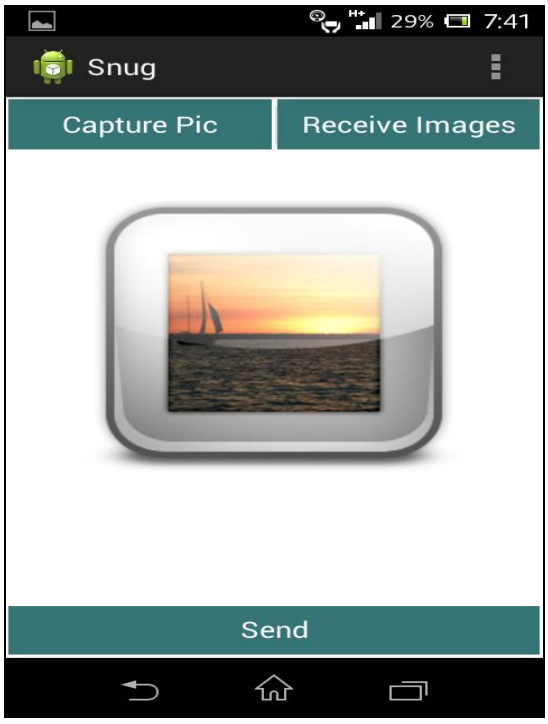

Fig 4.Main Menu Screen

1) ***Capture Pic*** button will start the camera and allows the user to take the picture, which he want to send.
2) ***Send*** button will prompt the user to enter the userid of the recipient to send the image.

3) ***Receive image*** button allows the receiver to view the list of images received. After which user can view the respective image by clicking on the image and entering a key.

C. Send Image:

On click of Send button in the main menu , the user will able to Send the captured image ,After entering the userid of the recipient as shown in Fig 5.The image will be contained in a Java object and sent to the output stream. It will make use of the internet. This image will be stored on parse cloud.

For Storing this image on the parse cloud we use classes from the com.parse package .We create a ParseObject and set its attributes like receiverId, senderId, imageName,etc using java and save it using the SaveInBankground method. Each ParseObject contains key-value pairs of JSON-compatible data. This data is schemaless, so there is no need to specify ahead of time what keys exist on each ParseObject.On successful save, each ParseObject is assigned a ObjectId which can be useful for retrieving the particular object.

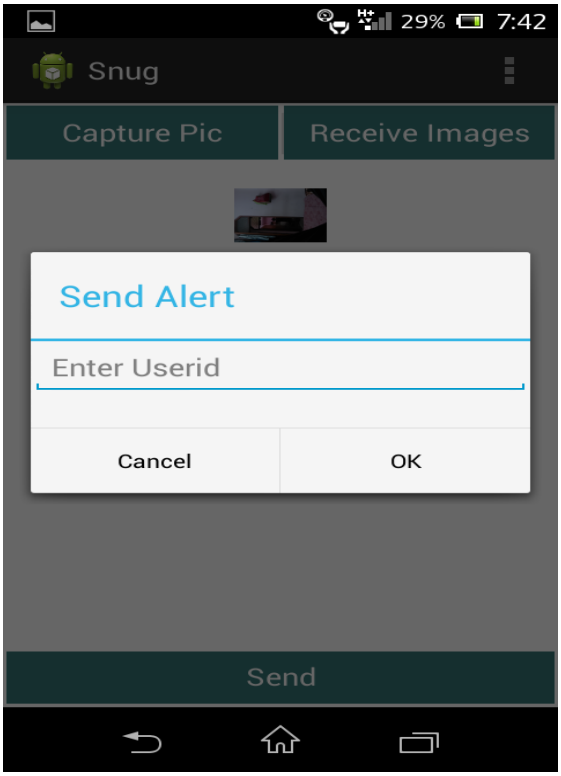

Fig 5.Send Button Screen

D. Receive Image:

On click of receive button the receiver will get the list of images sent to his id. As the image is stored on the parse in the form of ParseObject, so the whole ParseObject can be retrieved using ParseQuery. The ParseQuery offers different ways to retrieve a list of objects rather than just a single object. This is done by creating a ParseQuery, putting conditions on it (here, ObjectId), and then retrieving a List of matching ParseObject using the findInBackground method with a FindCallback. The findInBackground method assures that the network request is done on a background thread, and runs its callback in the main thread.

Once retrieved, images is listed in a non-readable format from parse cloud database and the recipient can view them by tapping on the image icon of the received images and providing the respective password, set by the sender during encryption, in order to decode it.This password, which is authenticated from the cloud database, the image will be transferred and decoded on the recipient's device after which it becomes viewable

This key will be communicated to him by the sender for security reasons. The receiver can not download the images but can only view it. Whenever the receiver wants to view the image he has to enter the valid key and if in between the sender changes the key, the receiver will need this latest key to view the image.

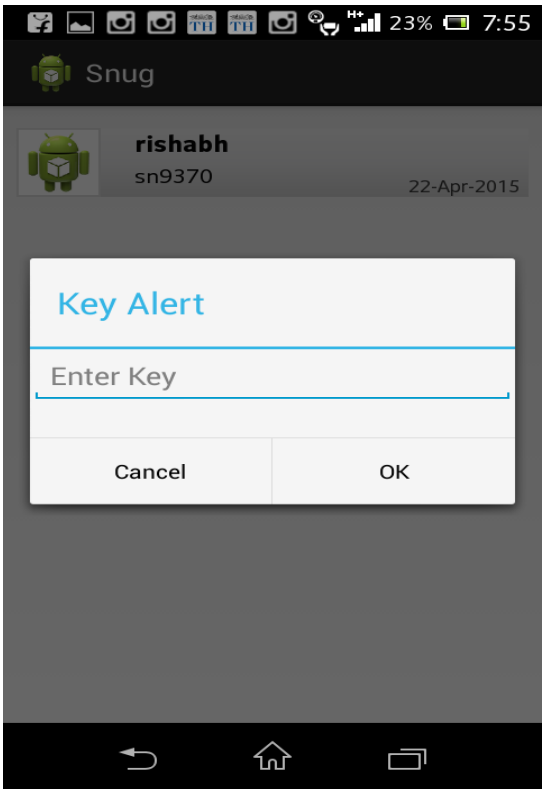

Fig 6.Receive Image Screen

V. FUTURE SCOPE

This idea is extendable towards delivering a wide range of encrypted content like audio, video and text (apart from images) with handset/device obscurity. The ease-of-use for this application could be further improved through inclusion of speech synthesis while selecting the recipients to whom the content is to be sent.

The security features of the application could be further fortified through implementation of additional, multi-layer encryption algorithms along with more device and user level authentications like OTP, IRIS Recognition, etc.
The addition of Data Mining and Artificial Intelligence concepts is another futuristic scope within this application for the improvement of its user friendliness and intelligent analysis of data being exchanged through it

VI. CONCLUSION

In this paper, an Android based mobile application for "*DATA OBSCURITY IN DEVICES DURING SECURED WIRELESS COMMUNICATION* "is presented. This app will help the users to exchange multimedia images effectively, in a secured manner, ensuring the confidentiality of communication. It may be utilized effectively in classified communications like criminal investigations, business communications, and the like. The reliable and fail-safe cloud service shall not only guarantee the integrity of the stored information but also its security because the multimedia content shall be stored in an encoded manner plus every user's data is enveloped in his/her own individual user account on the cloud.

REFERENCES

[1] Rich Maggiani, "Cloud Computing Is Changing How We Communicate", In *IEEE 978-1-4244-4358-1/09, 2009.*

[2] No author stated, The Notorious Nine, Cloud Security Alliance, February 2013
http://www.cloudsecurityalliance.org/topthreats

[3] Ted Samson, Nine Top Threats to Cloud Computing Security, Info World February 25, 2013 [Online] Available:
http://www.infoworld.com

[4] Jianfeng Yang and Zhibin Chen,"Cloud Computing Research and Security Issues", In *IEEE 978-1-4244-5392-4/10, 2010*

[5] Minqi Zhou, Rong Zhang, Wei Xie, Weining Qian and Aoying Zhou "Security and Privacy in Cloud Computing: A Survey", In *Sixth International Conference on semantics, Knowledge and Grids, 2010*.

[6] Krešimir Popović and Željko Hocenski, "Cloud computing security issues and challenges", *InMIPRO, 2010.*

[7] Farhan Bashir Shaikh and Sajjad Haider, "Security Threats in Cloud Computing", In *6th International Conference on Internet Technology and Secured Transactions, 2011*

.

[8] Balachandra Reddy Kandukuri, Ramakrishna Paturi V and Dr. Atanu Rakshit, "Cloud Security Issues", In
IEEE International Conference on Services Computing, 2009.
http://www.ijarcsse.com/docs/papers/Volume_3/11November2013/V3I11-0110.pdf

[9] Debajyoti Mukhopadhyay, Gitesh Sonawane, Parth Sarthi Gupta, Sagar Bhavsar, Vibha Mittal in
"Enhanced Security for Cloud Storage using File Encryption"
http://arxiv.org/ftp/arxiv/papers/1303/1303.7075.pdf

[10] Mobile Threat Monday: Android App Sells Your WhatsApp Conversations article on http://securitywatch.pcmag.com

Rishabh Mishra received his B.Tech. degree in Computer Science from V.I.E.T Ghaziabad, India, in 2011 and he is currently pursuing M.Tech. in Computer Science from Noida International University , U.P ,India.

Dr. Rakesh Sharma is Associate Professor in Department of Computer Science at Noida International University , U.P.

