ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

A Result Analysis on Secure Communication for Mobile Application Using Cloud Hosted Key

S.W.Thakare, Prof.N.R.Chopde

2nd Year M.E Student (CSE),GHRCEM, SGBAU Amravati University, India

H.O.D, Dept of CSE, GHRCEM, SGBAU Amravati University, India

ABSTRACT: User's data may be stored in a cloud to take advantage of its scalability, accessibility, and economics. & it must be protected from untrusted cloud provider. Basically develop 'Group Chat+' application for android mobile users. It is work on android, smartphone, PDA mobile users & also on laptop, desktop users virtually. It implements the idea of Android Cloud to Device Messaging securely. This application helps to sends data from servers to their applications on Android devices & Client to clients or /group of clients communication securely & vice-versa. The application provides a simple, lightweight mechanism on both server & clients. For security of users database AES Algorithms is implemented for encrypting as well as decrypting message through mobile application in cloud network. Symmetric Cryptographic Key is applied on sensitive data to be stored at cloud. For that, a key management scheme is proposed where encrypted key shares are stored in the cloud and automatically deleted based on passage of time or user activity. Also, Subscription to user data is maintained through regular re-generation of shares. Our application is able to resist the attacks from various fields over confidentiality, integrity, availability, and privacy. So that, SHA1 algorithm is implemented. It one way to secure hash value that implement in protection of all users password security throughout the network of mobile cloud.

KEYWORDS: Group Chat + Application, Android Mobiles, Mobile Cloud Computing, Security, Trust, Confidentiality.

I. INTRODUCTION

With the rising use of internet mobile applications has been increasing effectively. Different techniques and methods have been developed and used to protect the database. Cryptography is an absolutely necessary field that ensures the security of database. By applying encryption method, database attacks can be prevented. Encryption of data helps to change the data into a format that is not readable [6]. Without the proper key, this format can't be deciphered even if attacker hacks the information. Application of encryption in login phase makes it difficult for unauthorized users to access the database. Sharing message, group message resource among cloud users is to provide security a major problem, so cloud computing provides an economical and efficient solution. Due to frequent change in sharing data in a multi-owner manner to an untrusted cloud is still a challenging issue. In this proposal a secure data sharing scheme, for dynamic group in the cloud. By providing AES encryption while attached the data securely & it will be get decrypted using AES decryption any cloud user can securely share data with others [14]. Meanwhile, the storage overhead and encryption computation cost of the scheme are none with the number of revoked users. In addition, we analyze the security of this scheme with rigorous proofs. After sign up & login of users to our created Group Chat+ application. it generate IMEI NO of own mobile device of users login is one of the easiest and most popular forms of authentication that can be used for securing access to accounts. IMEI NO is often referred to as secure and stronger forms of authentication, and allowing them to install across multiple users mobile devices. It provides a multiple levels of security to share data among multi-owner manner using AES Algorithm. & also for users password SHA1 algorithm is applied [12].

Piotr K. Tysowski, M. Anwarul Hasan et al. (2013) proposed a query encryption method using hybrid encryption in which two layer of encryption was applied i.e. AES and Shamir Secret Algo [2]. The method of encryption which combines a symmetric and an asymmetric encryption method to take the advantages of each type of method are called hybrid encryption method. In symmetric key encryption, one common key is used by sender and receiver where as in asymmetric key encryption, two keys are used (a public key and a private key) [17]. This paper proposes a technique

DOI: 10.15680/ijircce.2015.0305090

which is similar to previous paper but here SHA1 Algorithms is applied instead of Shamir Secret Algo is said to be a good replacement of cryptosystem. Between these two, AES is a symmetric key cryptography where SHA1 is a public key cryptography. & it converts the key in encrypted format. The other part of the paper includes related work about avoid attack, work done, implementation, discussion, result analysis and conclusion respectively [19].

In these approaches, data owners store the encrypted data files in trusted storage and distribute the corresponding decryption keys only to authorized users [21]. To solve the challenges presented above, we propose a secure multi-owner data sharing scheme for dynamic group in the cloud. The main contributions of this paper include two levels of security is a unique and study of implementation of an extremely secured system, employing 2 levels of security [23].
Level 1: Level 1 security provides a simple text based Password.

That converts user's communication between plaintext to cipher text using AES Encryption & AES Decryption.
Level 2: After the successful entry of the above level, the Level 2 security system will then generate a unique IMEI No. of own user login mobile device on his device. The authentic user will be informed of this IMEI NO after login session. And for users password security SHA1 algorithms is implemented successfully. The other part of the project includes related work, methodology, implementation, result, discussion and conclusion respectively [19].

II. RELATED WORKS

This chapter describes the literature survey of all the references that are considered for the overall preparation of the system. It is a discussion of the literature in a given area of the study. It is concise overview of what has been studied, argued and established about a topic, and it is usually organized chronologically. Following is the listing of the things that were required to be studied for our project. Various access control techniques have been proposed for encrypted file storage in the cloud [2].

Piotr K. Tysowski, M. Anwarul Hasan et al. 2013 "Cloud-hosted key sharing towards secure and scalable mobile applications in clouds" [2]. In this system users can audit the cloud storage for very high communication & cost. i.e. users storage data is typically billed at a small fraction of a dollar per GB of data per month. Expensive key re-generation and re-distribution is occurring every time a single user. It used as AES & Shamir Secret Sharing Algorithm on CSP. But, we know that users data distributed via open network in cloud. So, key generate for users only secure communication not storage data. Wenjun Luo et al.(2014) "Efficient Sharing of Secure Cloud Storage Services" Suppose Bob, the boss in Company A, pays a secure cloud storage service and authorizes all the employees in that company to share such a service [04]. There exists a user hierarchy: Bob is the user at the upper level and all the employees in the company are the users at the lower level. In this paper, they design and construct a scheme, which enables the user at the upper level to efficiently share the secure cloud storage services with all the users at the lower level. W. E. Burr, D. F. Dodsonet.al. (2012) in their thesis report examinationed varied approaches supported NIST (National Institute of Standards and Technology) [16] recommends secret sharing as a technique to be used to protect long-term credentials in its level 3 security definition for a CSP (Cloud Service Provider). Secret key sharing allows a secret such as key information to be divided into multiple shares [13]. These shares may be distributed among key generators using the concept of threshold decryption [14], or portions of a private key are distributed among users. The challenge is that the client must assemble a key from multiple sources. & key shares being distributed on demand by some authority with distributed across a network for some time. P. Zimmermann et.al. (2011) during this work they projected to use the DEPSKY [16] storage system, shares are necessarily distributed across multiple clouds to form distributed trust and to restrict access [17]. Each cloud provider has access to a single share and thus cannot decode the stored data; this requires support for a cloud-of-clouds. Also, because the data shares are unencrypted, each cloud must be independent and collusion assumed to be impossible. R. Geambasu et.al. (2010) during this work they projected to use the Vanish system [19] distributes shares onto a DHT (Distributed Hash Table) that underlies a peer-to-peer file sharing network. It suggests the concept of "self-destructing data," where copies of data become unreadable over time due to the effect of user churn on the index. It requires that each user obtain key shares from multiple other nodes that form the index, if the user is operating a mobile device. J. Baek and Y. Zheng et.al. (2009) during this work they projected to use a straightforward approach employing PGP encryption [20] would encounter challenges with scalability for instance. The symmetric key used for encryption of user data may need to be encoded with the public key of each recipient. & it is preferable for a one-time encryption. If the same private key is shared by all users, then revocation would require some form of authentication to trust provider.

III. PROBLEM STATEMENT

This chapter describes the conclusion obtained from the literature survey. From the conclusion so obtained the statement of the problem is defined. There have been some researches to speed up the development of mobile application but having less security. So that, providing data confidentiality, in multi-tenant environments, becomes more challenging and conflicting. This is largely due to the fact that users outsource their data on cloud servers, which are controlled and managed by possible untrusted Cloud Service Provider (CSP). That is why, it is compulsory to provide secrecy by encrypting data before their storage in cloud servers while keeping the decryption keys out of reach of CSP and any malicious users. Nonetheless, the confidentiality preservation becomes more complex with resilient data sharing among dynamic groups [2]. The project mainly deals with security and data storage issues in cloud computing. Some researchers put privacy under security as they are closely related to each other. Our approach is to use android Group Chat+ mobile application for mobile users get communication become secure & data storage in WAMP as a cloud server provider. In cloud computing, security is a prime concern. If any cloud doesn't provide security then that application does not have any meaning. There have been continually challenged with how to provide secure services to support end-user applications. For this we will design a cloud system for storing data but which is fully secured and feasible for the user. We will design our scheme so that it will fulfill all the above requirements [13]. The chances of security loss for that we had restricted unauthorized users/attacker. While these are all important points to note, it's equally relevant to the discussion to consider that this new model brings not only a wealth of benefit, but also introduces new set of technical challenges for providing security to the cloud.

IV. EXISTING SYSTEM

In this chapter description about the existing system, that have already solve some nearby problem or which is similar in some kind to the proposed system . Also the existing proposed system is described the working of the system different phases. In this system users audit the cloud storage for very high communication & cost. I.e. user's storage data is typically billed at a small fraction of a dollar per GB of data per month. Expensive key re-generation and re-distribution does occur every time a single user's access is revoked. Subscription to user's data is maintained through regular re-generation of shares. Minimal communication required with the cloud provider for mobile users [2]. Then AES & Shamir Secret Sharing Algorithm is used for data security. On the server end, an F1-class front-end instance was run as a Java servlet on the Google App Engine (GAE) cloud, configured at 600 MHz processing and 128 MB of RAM. A connection was established between the desktop or mobile Android client and an instance running on the GAE cloud via HTTP requests. Using JSON for data interchange and the Google Json library for marshaling between Java objects used by the Java client and server implementations. Only a (J2SE) SDK 1.6 classes is whitelisted on Android and GAE cloud are supported. Only data sharing security is available but for users password there will be no security assign.

V. PROPOSED MODEL

The main aim of our project is to using an Android application that uses secure communication of mobile application using cloud hosted key. When mobile device users install an application on his device then it includes Group Chat+ application. & it must approve the use of this feature of application via server [4]. Uninstalling the application also has the effect of unregistering. This application provides a solution to send as IMEI no as notification to the users on his handset. When it receives a notification from the server it makes a special non- modal transient sign up message box on phone [5]-[8]. Figure 1 Shows Flow Diagram of System Model.

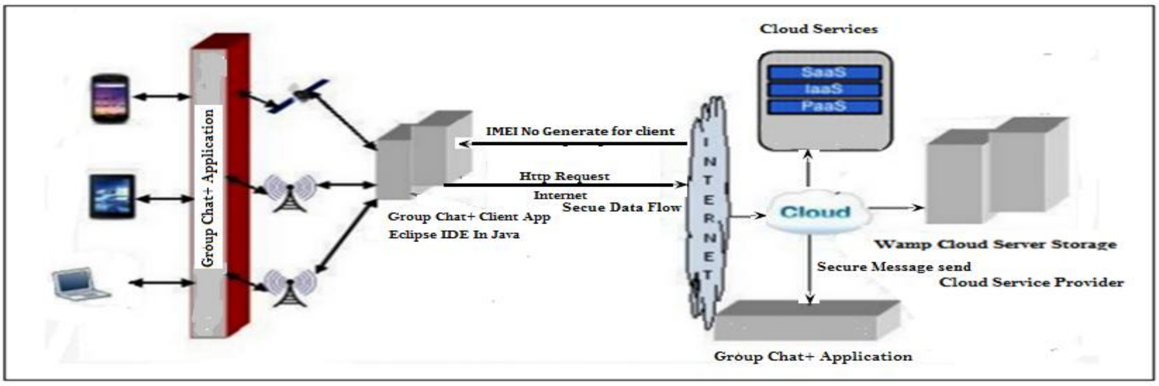


Figure 1 Flow Diagram of System Model

Most of changes overcome in this project and get implemented are as follows:-

- Platform: - Wamp Lite as Cloud Server Provider.
- Create Mobile Application: Group Chat+ using Eclipse as Java IDE tool.
- Algorithms: - SHA1 & AES Algorithms used.
- SDK Version: - Android 2.2 version used.
- Front End: PHP
- Jdk 7.0 version for coding, Command Prompt
- Back End: MySQL.

VI. SYSTEN IMPLEMENTATION

After the requirement specification was made, we went ahead with implementing the system design. In this phase, we used the requirement specification as the input to produce the desired system. Among the various software designs, we have used object oriented approach for designing our application**.** The whole implementation is done in windows operating system and the code is developed by using HTML, PHP and MYSQL, JAVA. Eclipse IDE as a Java Developer and Wamp Server are used for the implementation. In second part, the system model is an adaption proposed by Piotr K. Tysowski, M. Anwarul Hasan (2013) [2]. They have implemented the model using core java but here PHP, Eclipse IDE as a Java Developer and MYSQL are used. In database, separate tables are there for AES encryption & AES decryption. SHA1 Algorithm provides protection of users password that one way to secure hash value with all tables have a common column i.e. User id, from id, password, and message in encrypted format, & chat key. Figure 2 Shows Proposed System

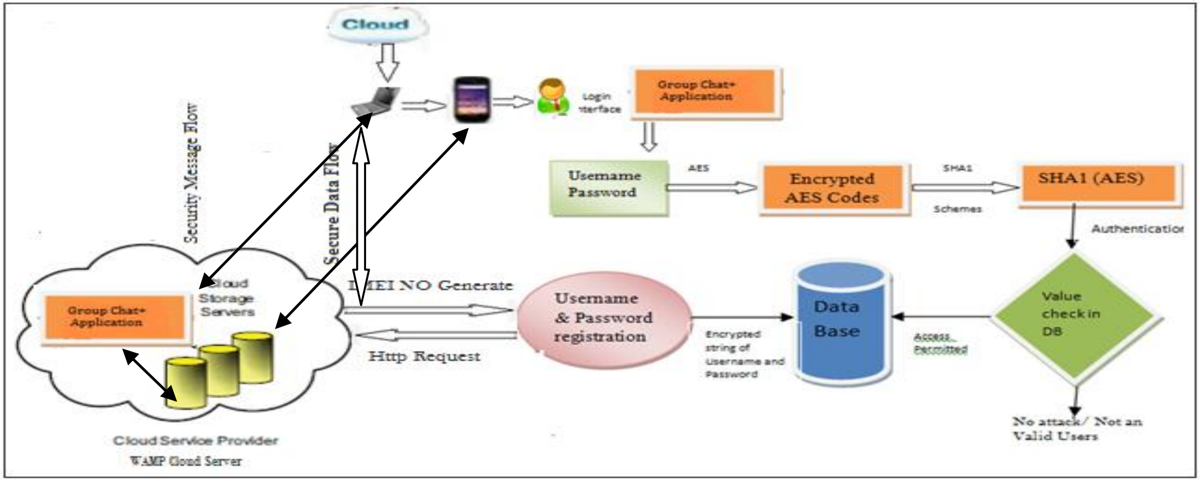


Figure 2 Proposed System

Since, we are working with the mobile application. I.e. the database need not be developed externally and imported to the phone instead we use the Android in-built notification feature for storing the recent notification from the application server. For developing the application server we use MySQL as the back end for storing the user id, password, registration id and authentication id with his chat key. We make the front end with the help of PHP language; create mobile application as Group Chat+ with the help of Eclipse IDE as a Java Developer. Also to create AVD (Android Virtual Device) using SDK 2.4.1 for mobile devices this is used to send the messages to other mobile devices securely.
The model proposed by Piotr K. Tysowski, M. Anwarul Hasan (2013) is used in this paper [2]. The only difference in paper is that here, SHA1 Algorithm is used instead of Shamir Secret Sharing Algorithm as it is a good replacement of cryptosystem. So, basically the system model is an adaption of previously proposed model. The Proposed Model includes three phases i.e. registration phase, login phase and verification phase [6].

1] In registration phase: A new user registers his/her name by selecting unique username and password and the data get sent to server and when server receives the username and password, it saves these information in the database in a table and along these data; server keeps a unique key for each username generated by the server itself. Then server sends back a confirmation request to the client to his own mobile device IMEI NO Figure 3 Shows Model of registration phase
2] In login phase: when a registered user tries to login, the username and password get encrypted by applying AES encryption algorithm which uses user's secret key. After that, a query gets generated automatically using the encrypted username and password [11]. Then SHA1 Algorithm is used secure hash value where server's public key is used to encrypt the query and once it gets encrypted it is sent to the server. Figure 4 shows model of login phase.
3] **In Verification Phase**: When server receives the query, it uses AES decryption method to decrypt the query where the server private keys are used. Server then checks the username and password and again the username and password gets decrypted using AES decryption algorithm [13]. After getting decrypted, if the username and password match to the database table, the login request gets accepted otherwise invalid users found. Figure 5 shows the Model of verification phase.

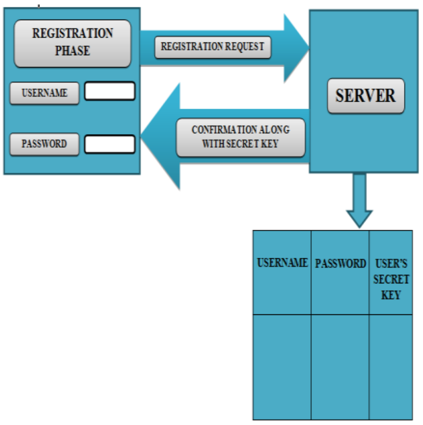


Figure 3 Model of Registration Phase

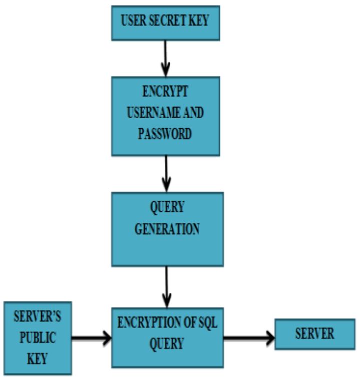


Figure 4 Model of Login Phase

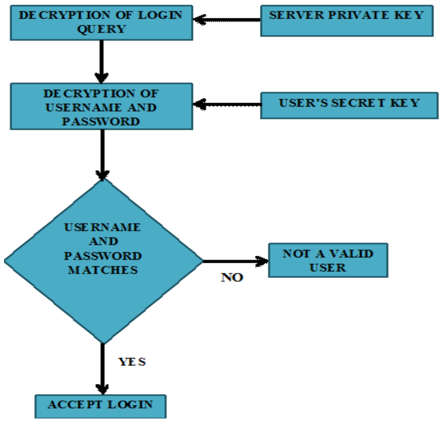
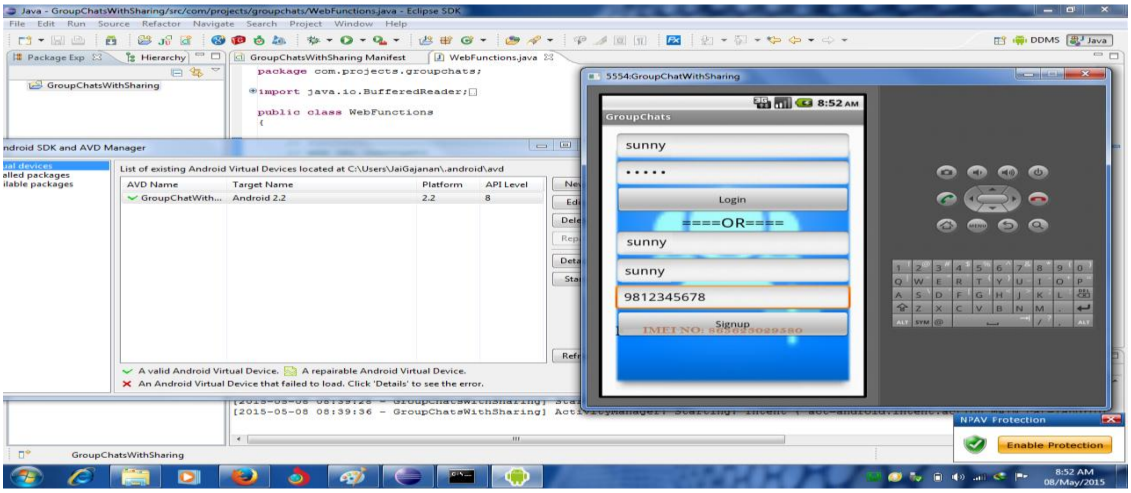

DECRYPTION OF LOGIN QUERY
SERVER PRIVATE KEY
DECRYPTION OF USERNAME AND PASSWORD
USER'S SECRET KEY
USERNAME AND PASSWORD MATCHES
NO
NOT A VALID USER
YES
ACCEPT LOGIN

Figure 5 Model of Verification Phase

Also after login by user he can check his current location, update time, manage account and hide location etc.

Server System & Mobile Application Working
The system execution details are as follows-
The online Wamp as a cloud server project provides dbfgroupchatsharing databse, performance schema, MySQL, security & testing phase. The Wamp server provided application as dbfgroupchatsharing databse as tblusers, tblchats, tblgroups, tblgroupchats. Then Check the static IP of server provided by command prompt for configure both Wamp server & Eclipse IDE for Mobile application. After finding the server static IP it is insert in created Mobile application of java code for configure to both Wamp server & Eclipse IDE for Java Developer. For that, we test whether authorized clients are connected to Wamp server via static IP through Group Chat+ application for preserving security. The Group Chat+ mobile application apk file is export on desktop location for use purpose with particular server static IP that preserve security. & is ha having 16 year certificate validity. Using Eclipse IDE as java developer launching android

version 2.2 mobile device virtually for server & also for client that uses on laptop/desktop. the launch android mobile device virtually for running & debugging via online as android emulator the created Group Chat+ Application use for android devices and also for laptop, desktop users virtually. Figure 6 shows user registration via virtual created mobile application. It shows list of entries for creating user account of member. After successful login User IMEI No get generated on his device as a security. Using Group Chat+ apk users chat between one and more users/group of users secularly. Because of configure static IP address of server with having dual algorithms.



```
package com.projects.groupchats;
import java.io.BufferedReader;
public class WebFunctions
{
```

List of existing Android Virtual Devices located at C:\Users\JaiGajanan\.android\avd

AVD Name	Target Name	Platform	API Level
GroupChatWith...	Android 2.2	2.2	8

✓ A valid Android Virtual Device. A repairable Android Virtual Device.
✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.

```
[2015-05-08 08:39:36 - GroupChatsWithSharing] ActivityManager: Starting: Intent { act=android.intent.ac
```

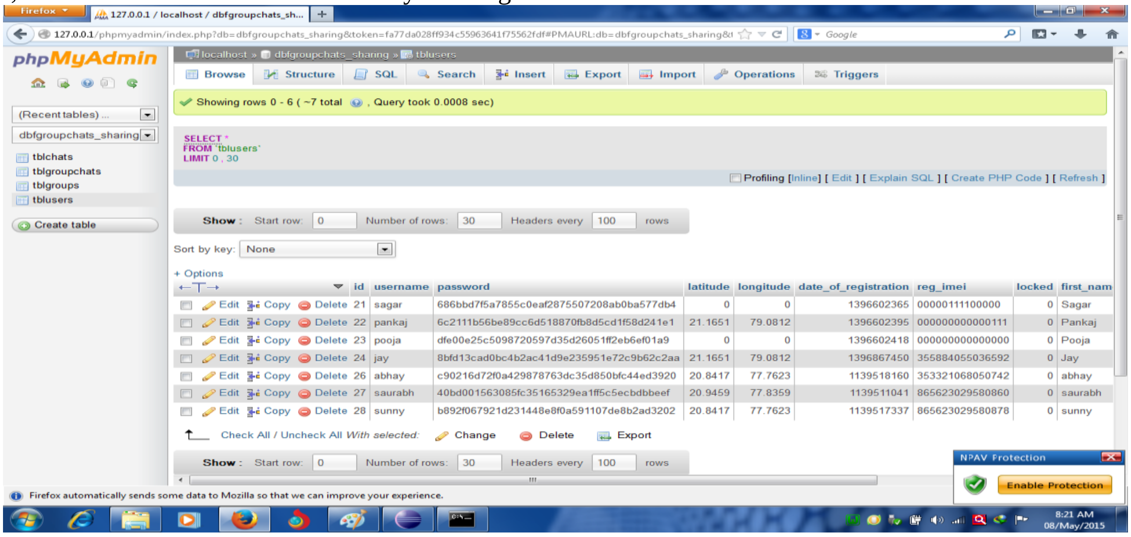
GroupChats

sunny
•••••
Login
====OR====
sunny
sunny
9812345678
Signup
IMEI NO: 865623029580

Java - GroupChatsWithSharing/src/com/projects/groupchats/WebFunctions.java - Eclipse SDK
File Edit Run Source Refactor Navigate Search Project Window Help
Package Exp
Hierarchy
GroupChatsWithSharing Manifest
WebFunctions.java
GroupChatsWithSharing
Android SDK and AVD Manager
5554:GroupChatWithSharing
8:52 AM
NPAV Protection
Enable Protection
8:52 AM 08/May/2015

Figure 6 User Registration on Group Chat+ Mobile Application

It perform no. of actions, as load contacts, add group, delete groups, send sms to groups, send sms to contacts, set location, delete location, hide location etc. created users list on Wamp as a cloud server having users password are encrypted because of security purpose. tblusers perform no. of actions as UID, Uname, UPW, latitude, longitude, date of registration, register IMEI NO, locked, first name etc. user information in which user can see the date, our current location, hide location on his own device only. See Figure7 List of tblusers.



✓ Showing rows 0 - 6 (~7 total , Query took 0.0008 sec)

```
SELECT *
FROM `tblusers`
LIMIT 0 , 30
```

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

	id	username	password	latitude	longitude	date_of_registration	reg_imei	locked	first_nam
□ Edit Copy Delete	21	sagar	686bbd7f5a7855c0eaf2875507208ab0ba577db4	0	0	1396602365	00000111100000	0	Sagar
□ Edit Copy Delete	22	pankaj	6c2111b56be89cc6d518870fb8d5cd1f58d241e1	21.1651	79.0812	1396602395	000000000000111	0	Pankaj
□ Edit Copy Delete	23	pooja	dfe00e25c5098720597d35d26051ff2eb6ef01a9	0	0	1396602418	000000000000000	0	Pooja
□ Edit Copy Delete	24	jay	8bfd13cad0bc4b2ac41d9e235951e72c9b62c2aa	21.1651	79.0812	1396867450	355884055036592	0	Jay
□ Edit Copy Delete	26	abhay	c90216d72f0a429878763dc35d850bfc44ed3920	20.8417	77.7623	1139518160	353321068050742	0	abhay
□ Edit Copy Delete	27	saurabh	40bd001563085fc35165329ea1ff5c5ecbdbbeef	20.9459	77.8359	1139511041	865623029580860	0	saurabh
□ Edit Copy Delete	28	sunny	b892f067921d231448e8f0a591107de8b2ad3202	20.8417	77.7623	1139517337	865623029580878	0	sunny

Check All / Uncheck All *With selected:* Change Delete Export

127.0.0.1 / localhost / dbfgroupchats_sh...
127.0.0.1/phpmyadmin/index.php?db=dbfgroupchats_sharing&token=fa77da028ff934c55963641f75562fdf#PMAURL:db=dbfgroupchats_sharing&t
phpMyAdmin
(Recent tables) ...
dbfgroupchats_sharing
tblchats
tblgroupchats
tblgroups
tblusers
Create table
localhost » dbfgroupchats_sharing » tblusers
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Triggers
Profiling [Inline] [Edit] [Explain SQL] [Create PHP Code] [Refresh]
+ Options
NPAV Protection
Enable Protection
Firefox automatically sends some data to Mozilla so that we can improve your experience.
8:21 AM 08/May/2015

Figure 7 List of tblusers

Also user can perform other actions such as create one or more groups, delete groups in his account etc. shows process of sending messages to one or more or groups of users with any attachment securely. List of users chats having secure message send between one or more users with encrypted form having generated chat key for particular users through Wamp as a cloud server. Also, on cloud server see result of users from id, to id, date of message send, attachment path,

& chat key etc. List of group chat users shows group chats between one or more group users securely having group id, from id, date of message, path & group key etc. In Manage group chat+ application. Here users are created/deleted groups in his account. Also, tblgroups perform no. of actions as Group ID, Group name, Admin ID, Group Members, Created On date etc. user logout screen. After secure communication of one or more group of users using mobile application with chat key on his mobile device. Hence, finally users can successfully logout from Group chat+ application.

VII. **METHODOLOGY**

Proposed Algorithm

The following algorithm is based on the principle of limiting access to encrypted data in the cloud through the process of storing and removing encrypted key shares in the cloud.

A] Main technique SHA1 Algorithm

SHA-1 is cryptographic function that is designed by National Security Agency. The full form of SHA is secure hash function. It produces a 160 bit message digest. It produce the hash value with wide variety of applications including TLS, SSL and SSH [11]. It has a 512 bit block size and has 80 numbers of rounds. It is used for computing a compressed representation of a message. If we give an input message of arbitrary length < 264 bits i.e. 2 billion GB of data, it produces a 160-bit output called the message digest. It is claimed to be secure because it is practically infeasible to compute the message corresponding to a given message digest. Also it is extremely improbable to detect two messages hashing to the same value.

Secure Hashing Algorithm

Step 1:-Padding. The Figure 8 shows SHA1 iteration.
Add Padding to the end of the genuine message length is 64 bits and multiple of 512.
Step2:- Appending length.
In this step the excluding length is calculated.
Step3:- Divide the Input into 512-bit blocks.
In this step we divide the input in the 512 bit blocks.
Step4:-Initialize chaining variables. In this step we initializing chaining variables here we initialize 5 chaining variables of 32 bit each=160 bit of total.
Step5:-Process Blocks. Table 1 Shows Example of SHA-1 algorithm [14].
1) Copy the chaining variables 2) Divide the 512 into 16 sub blocks 3) Process 4 rounds of 20 steps each **[2].**

B] AES Algorithm

AES is a block cipher with a block length of 128 bits. It allows for three different key lengths: 128, 192, or 256 bits. Most of our discussion will assume that the key length is 128bits. It also has the notion of a word. A word consists of four bytes that is 32 bits [8]. Therefore, each column of the state array is a word, as is each row. Each round of processing works on the input state array and produces an output state array. AES, notified by NIST as a standard in 2001, is a slight variation of the Rijndael cipher invented by two Belgian cryptographers Joan Daemen and Vincent Rijmen [17]. The AES algorithm consists of ten rounds of encryption, seen in Figure 14. Each round includes a transformation using the corresponding cipher key to ensure the security of the encryption. Each round consists of the following operations: a) Substitute bytes b) Shift rows c) Mix columns d) Add round key [12].

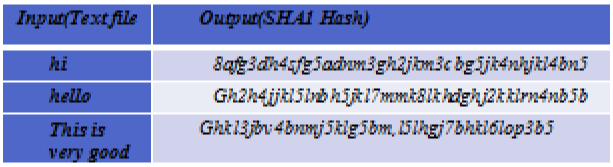
For decryption, each round consists of the following four steps: 1) Inverse shift rows, 2) Inverse substitute bytes, 3) Add round key, and 4) Inverse mix columns. The third step consists of XOR the output of the previous two steps with four words from the key schedule. The tenth round is similar to rounds one to nine [10].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015



Input(Text file	*Output(SHA1 Hash)*
hi	*8afg3dh4sfg5adnm3gh2jkm3c bg5jk4nhjkl4bn5*
hello	*Gh2h4jjkl5lnb h5jkl7mmk8lk hdghj2k klrn4nb5b*
This is very good	*Ghkl3jbv 4bnmj5klg5bm, l5lhgj7bhkl6lop3b5*

Table 1 SHA1 Example

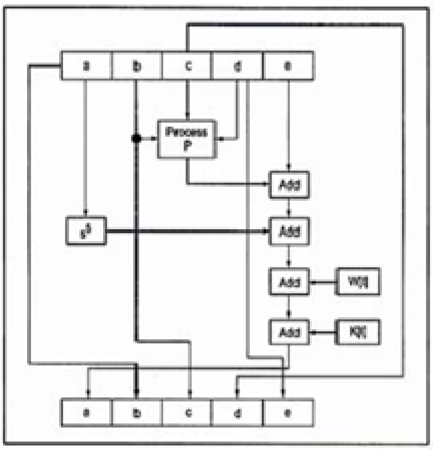


Figure 8 SHA1Iteration

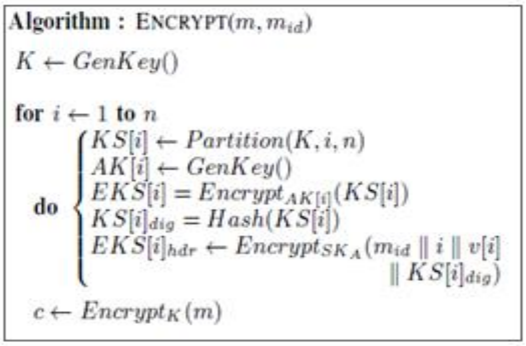
The insufficient key shares remain encrypted data stored in the cloud is no longer useful to the authorized user set. The data owner will not re-generate a new data key, and the cipher text may be discarded at the end, as it cannot be decrypted with the remaining shares. Another possibility is to simply reveal the data key to the cloud provider and allow global unrestricted access to the user data, as once enough time has passed. A transaction is useful application.

The main advantages of the technique are summarized:
1] Key shares are stored securely in the cloud. Even if every key share is protected with a unique access key, and the access keys are never shared with the cloud provider. 2]If a user shares an access key with the cloud provider in an unauthorized manner, then only a limited number of key shares will be temporarily accessible to the provider until they are deleted; even if the secret data key *K* is decoded by the provider, it will eventually be rotated. 3] The same key material stolen from a compromised user. Because key shares are encrypted, then even if the provider is malicious the cloud is uneconomical to the provider until the next key update. From basic probability theory, the chance that any user will have access to a sufficient number of shares is:

$$\left(\prod_{0\leq i\leq d}\left(\frac{n-i-w}{n-i}\right)\middle|0\leq d\leq(n-t)\right)$$

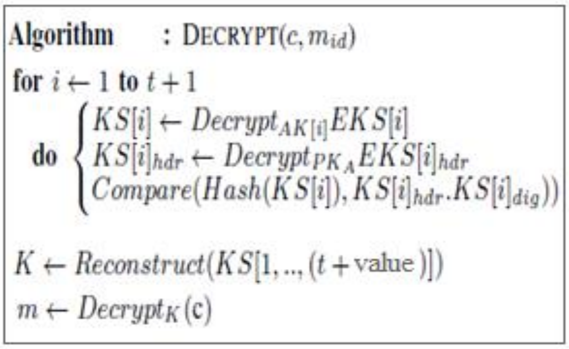
Where *d* is the total number of deletions, *w* is the amount of shares generated for the user to the required value time *t*+. The proposed approach is the storage space on the user's device, not only of key shares, but also of their corresponding access keys. Each type is assigned as:-

1) Key generation and encryption: Consider a technique based SHA 1 Algorithm. U is the set of users accessing the cloud, and an access structure ΓU is a list of subsets of U such that each subset is trusted. Any trusted subset Utr of parties, where Utr ∈ ΓU, can recover the secret from the set KS of shares stored in the cloud. Any untrusted subset, however, cannot obtain information about the secret. The access control structure can be defined such that any (t+ value add). See Figure 9 in the ENCRYPT operation, & Figure 13 shows key generation, which user *A*, and proceeds to generate key shares and encrypt a message *m* to be stored in the cloud and identified with a unique identifier *mid*. It generates a symmetric key *K* and divides it into multiple shares *KS*[1] to *KS*[*n*], where *n* is the current total number of shares, and a minimum of *t*+ value shares are required for decryption, where $t+1\leq n$. Parameter *t* may be decreased or increased in value for a corresponding adjustment in the level of security, while parameter *n* determines the number of users supported and the storage requirements for the shares. Each share *KS*[*i*] is encrypted as *EKS*[*i*], using a symmetric encryption key *AK*[*i*] belonging to user *A*, known as an access key; it is also possible for the same access key *AK*[*i*] to protect multiple shares, instead, to conserve associated storage and communication. The encrypted shares are stored in a key database in the cloud and cannot be read in plaintext form by the provider, although they remain accessible for download by users. The message flow is shown in Figure 11. The plaintext user data *m* requiring protection is assigned a unique record identifier of *mid* and encrypted by *A* as cipher text *c* using *K*, is uploaded to the provider, and is stored in the cloud. Since the cloud provider cannot unlock any share stored in the key database, it is unable to decode *c*. To the cipher text of the user data is appended a description key *L* identifying the set of key shares eligible to decrypt the data, of which only the share time value is required by any user.



```
Algorithm : ENCRYPT($m, m_{id}$)
$K \leftarrow GenKey()$
for $i \leftarrow 1$ to $n$
do
    $KS[i] \leftarrow Partition(K, i, n)$
    $AK[i] \leftarrow GenKey()$
    $EKS[i] = Encrypt_{AK[i]}(KS[i])$
    $KS[i]_{dig} = Hash(KS[i])$
    $EKS[i]_{hdr} \leftarrow Encrypt_{SK_A}(m_{id} \| i \| v[i] \| KS[i]_{dig})$
$c \leftarrow Encrypt_K(m)$
```

Figure 9 AES Encryption Algorithm



```
Algorithm : DECRYPT($c, m_{id}$)
for $i \leftarrow 1$ to $t+1$
do
    $KS[i] \leftarrow Decrypt_{AK[i]} EKS[i]$
    $KS[i]_{hdr} \leftarrow Decrypt_{PK_A} EKS[i]_{hdr}$
    $Compare(Hash(KS[i]), KS[i]_{hdr}.KS[i]_{dig}))$
$K \leftarrow Reconstruct(KS[1,..,(t + value)])$
$m \leftarrow Decrypt_K(c)$
```

Figure 10 AES Decryption Algorithm

2) Metadata: For malicious modification of a share by the cloud provider, A will also create a signed metadata header K for each share and upload it with the encrypted share. The metadata will consist of the following fields: the record identifier mid, the key share identifier i ∈ {0..n}, the key share version v, and a digest K[i]dig of the key share content, such as a cryptographic hash of it.

3) Decryption: As per Algorithm Figure 10. symmetric access keys K[x] to K[y] from A, where the range of keys is of at least size t+ values, the required time value; this assumes that each key K[i] permits decryption of the key share KS[i] stored in the cloud, where i is in the range 1 to n, such that all shares are in L. Again, it is also possible to have one access key unlock multiple shares, instead. To ensure that a share downloaded from the cloud is the correct one and it has not been modified by cloud provider, B may inspect the metadata associated with the cipher text, and decode it using A's known public key PKA.

For one sample configuration, suppose that the total number of shares *n* for a particular data record is 110, and that the number of shares required for decryption, *t*+. Shown Figure 12 Data owner *A* will provide access key *AK*[1] to *B*, which provides access to 5 shares stored in the cloud; only 3 are required. *B* may download the cipher text, as well as all three encrypted key shares, directly from the cloud. *B* will then be able to decrypt the required user data. The entire users may be greater in number than the total number of shares *n*; thus, the same key shares may be randomly assigned to multiple users.

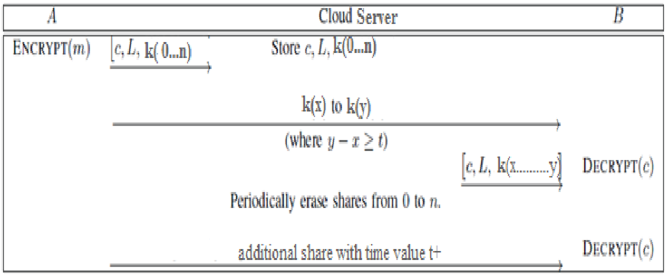


Figure 11 Message Flow

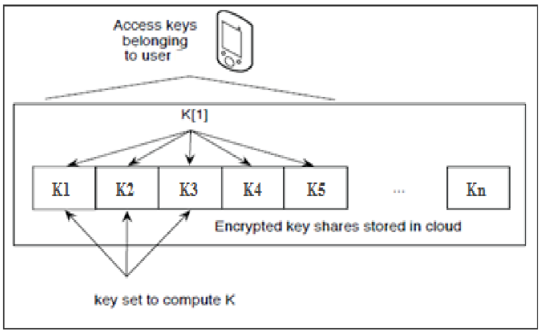

Access keys belonging to user
K[1]
K1
K2
K3
K4
K5
...
Kn
Encrypted key shares stored in cloud
key set to compute K

Figure 12 Key shares

4) Key share deletion: The individual key shares in the cloud are independently deleted by the cloud provider. This process can occur at regularly scheduled time intervals, such that a random share is deleted every day, for instance. If user B had locally cached all decrypted key shares, then B will continue to be able to decrypt data from the cloud until his cache needs to be refreshed, or the entire key store expires. Suppose that key share KS[1] is erased and that B needs to re-fetch key shares from the cloud. B will find that EKS[1] (the encrypted version of KS[1]) is no longer available, and so another key share must be randomly chosen from the available set. B will be required to use an appropriate access key in the set AK[x] to AK[y] to access another available key share outside of the initial three, such as KS[4]. The size of the valid remaining key store in the cloud will decrease from the initial maximum until the store is re-generated. Access keys must only be re-generated for shares that belonged to users whose access rights were revoked from the last time that shares were generated.

5) Key share replacement: Once the number of outstanding valid shares to random replacement decreases to t, it becomes impossible for users to download sufficient valid shares to replace those that are deleted, even if additional access keys are obtained from the data owner or other users. The key store in the cloud then expires; the content owner A can then proceed to replace the deleted shares in the cloud with newly-generated valid shares of a new version of the symmetric key K. A new access key will also be generated for each share, or set of shares, to protect them. Thus, for instance, key A[1] will protect the new key shares KS[1] to KS[5], from 1 to 5, and so on; a total of n key shares are again stored in the cloud with headers reflecting the new version. In this case, the user data that is stored in the cloud and encrypted with the older key K must be replaced with a version that is encrypted with the new key K; this may be done by user A.

I] Encoder Process

The Encryption and decryption process consists of a number of different transformations applied consecutively over the data block bits, in a fixed number of iterations, called rounds. The encryption process has different stages, like as sub byte, row shift, mix column, and key generation. The encoder requires 10 rounds to complete the process. These inputs are sequentially applied in different rounds. The number of rounds depends on the length of the key used for the encryption process. For key length of 128 bits, the number of rounds required are10. (Nr = 10). The four different transformations are described in detail below [11].

1] Sub Bytes Transformation: It is a non-linear substitution of bytes that operates independently on each byte of the State using a substitution table (S box). In this sub bytes step the data in the plain text is substituted by some pre-defined values from a substitution box. The substitution box which is used commonly is Rinjdael substitution box. The substitution box is invertible.

2] Shift Rows Transformation: Cyclically shifts the rows of the State over different offsets. The operation is almost the same in the decryption process except for the fact that the shifting offsets have different values. In shift rows operation the rows in the 4×4 matrix is shifted to left r bits and r varies with the rows of the matrix(r=0 for row1, r=1 for row2, r=2 for row3, r=3 for row 4). This process is illustrated in Figure 15. This has the effect of moving positions of lower positions in the row, while the lowest bytes wrap around to the top of the row. Figure 8 shows row shift operations are as follows:-

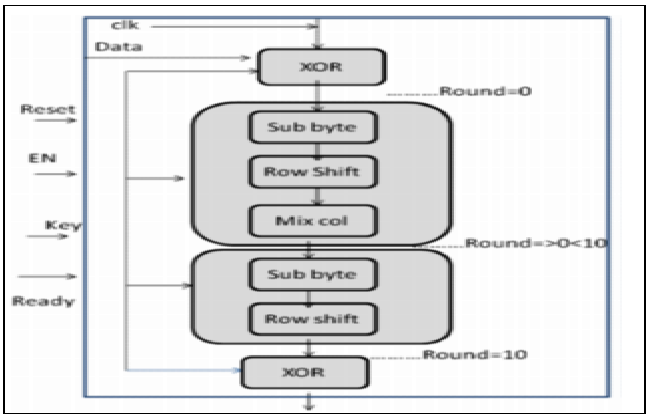


Figure 13 Block Diagram of Key generation

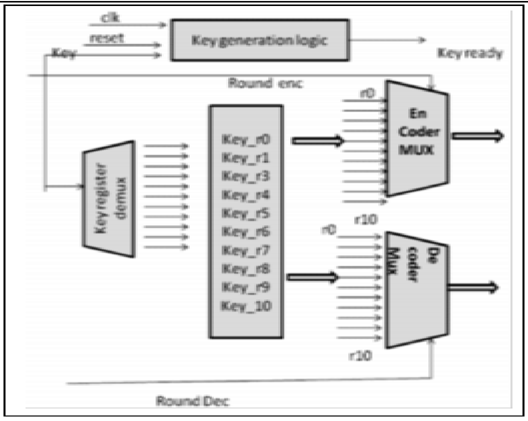


Figure 14 Block Diagram of Encrypt process

3] Mix Columns Transformation: This transformation operates on the State column-by column, each column as a four-term polynomial. The columns are considered as polynomials over Galois Field (28) and multiplied by modulo x4 + 1 with a fixed polynomial a(x) = {03} x3+ {01} x2+ {02} x.

4] Add Round Key Transformation: In this transformation, a Round Key is added to the State by a simple Bitwise XOR operation. Each Round Key consists of Nb words from the key expansion. Those Nb words are each added into the columns of the State. Key Addition is the same for the decryption process. In the add round key step the 128 bit data is stored with the sub key of the current round using the key expansion operation. To add round key is used in two different places one during the start that is when round r=0 and then during the other rounds that is when 1 ≤ round≤ Nr, where Nr is the maximum number of rounds. The formula to perform the add round key is S'(x) = S(x) R(x) where S'(x) – state after adding round key ,S(x) – state before adding round key and R(x) – round key.

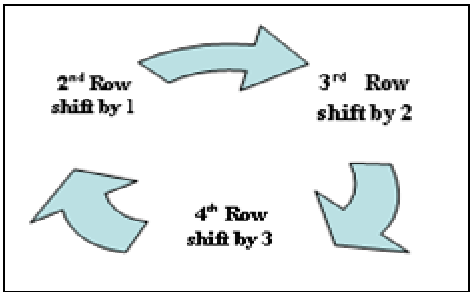


Figure 15 Row shift operation

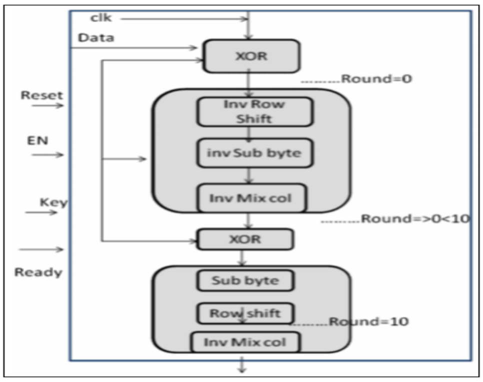


Figure 16 Block diagram of Decryption Process

5] Key generation: Each round key is a 4-word (128-bit) register generated as a product of the previous round key, a constant that changes each round, and a series of S-Box lookups for each 32-bit word of the key. The Key schedule Expansion generates a total of Nb (Nr + 1) words. We have to use the different blocks designing of AES .it consist of encoder block and decoder block, that can be able to generate the10 different key and its save in different resistors .then we used these key are required in different stages. when input data are applied the key generation block and also clock pulse and output of initial key are used to different operation, these key are required the input of the encoder ,but output of encoder is required to them as a input of decoder also used I the enable input. We have got the result and the analysis of the output wave form. Figure 13 shows block diagram of key generation & Figure 14 shows block diagram of encryption process.

II] Decoder Process

The decryption process is direct inverse of the encryption process. All the transformations applied in encryption process are inversely applied to this process. Hence the last round values of both the data and key are first round inputs for the decryption process and follows in decreasing order [12]. Figure 16 shows Block Diagram of Decryption Process.

Generation of IMEI NO:

The algorithm can be described in 3 steps:
Step 1: Generate the HMAC-SHA value Let HMK = HMAC-SHA (Key, T) // HMK is a 20-byte string.
Step 2: Generate a hex code of the HMK. Hex HMK=To Hex (HMK).
Step 3: Extract the 15-digit IMEI NO from the string IMEI = Truncate (Hex HMK) the Truncate function in Step 3 does the dynamic truncation and reduces the IMEI NO to 15-digit [9].

VIII. EXPERIMENTAL RESULT & ANALYSIS

Performance Evaluation

There are two types of performance evaluation first is to determine the performance using server system IP configure on mobile client devices and second is to after successful user login via mobile application, server generate mobile device IMEI NO send to client device are as follows.

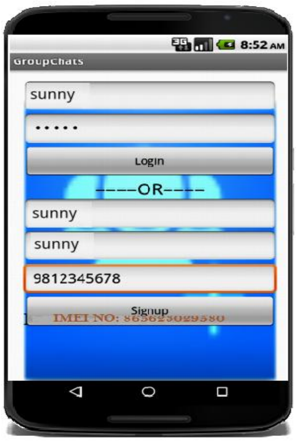
Result Phase

In this section, we first describe How the System model & Group Chat+ application get work in our project. Then, we represent the security goals provide by Group Chat+ application that secure communication between one or more users/group of users [15].

1] Whether our mobile applications are connect to server or not? Answer is YES.

For that open any browser in our mobile and type the Wamp server System static IP address in URL of browser. If Wamp server main page is open then confirm that, Mobile users are connected to server with security. See Figure 17 testing phase on mobile application as below. This is because the server static IP address is configured to both Wamp Server and clients of mobile users or Eclipse for AVD clients. This means that Wamp server manages all records of IMEI NO of connected users & provides security service or to conduct a transaction will not be able to abuse it, since it will be longer valid. User own mobile device IMEI NO can be used to authenticate a system via an authentication server. Also, if some more steps are carried out the server calculates subsequent IMEI NO and sends/displays it to the user who can login via Group Chat+ application. The user can also authenticate the validation server. See Figure 18 Device receives notification.

Eclipser IDE as a Java Developer using SDK concept has been strictly followed for carrying out the application development. It is create virtually on server as well as client's machine. We have tried to make our source code re-usable so that it can be used again to add new functionalities with slight or no modification. Reusable modules and classes reduce implementation time, increase the likelihood that prior testing and use has eliminated bugs and localizes code modifications when a change in implementation is required.



sunny
•••••
sunny
sunny
9812345678
IMEI NO: 865625029580

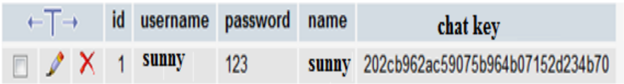
Figure 17 Testing Phase on Mobile Application

Figure 18 Device Receives Notification

Interface optimization has been used to enable significant opportunities to optimize the implementation of interfaces on a set of implemented objects. The advantages of interface optimization is that when code is compiled, because the compiler knows the full list of interfaces and the objects which implement the interfaces, it can improve execution and working set (i.e., recently referenced pages in a program's virtual address space) when implementing the interfaces on objects.

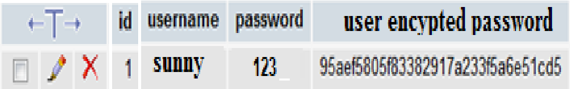
Role of WAMP server

Able to communicate with your client and fire off HTTP requests to the Cloud Server. b) Able to handle requests and queue data as needed. For example, it should be able to perform exponential back off [6]. c) Able to store the Client Login as Authorization token and client registration IDs, Password & its chat key. The Client Login Authorization token is included in the header of POST requests that send messages. Figure 19 Shows Symmetric Secret Key (Chat key) Generation after Registration & Provide to the login user.



←T→	id	username	password	name	chat key
□ ✎ ✗	1	sunny	123	sunny	202cb962ac59075b964b07152d234b70

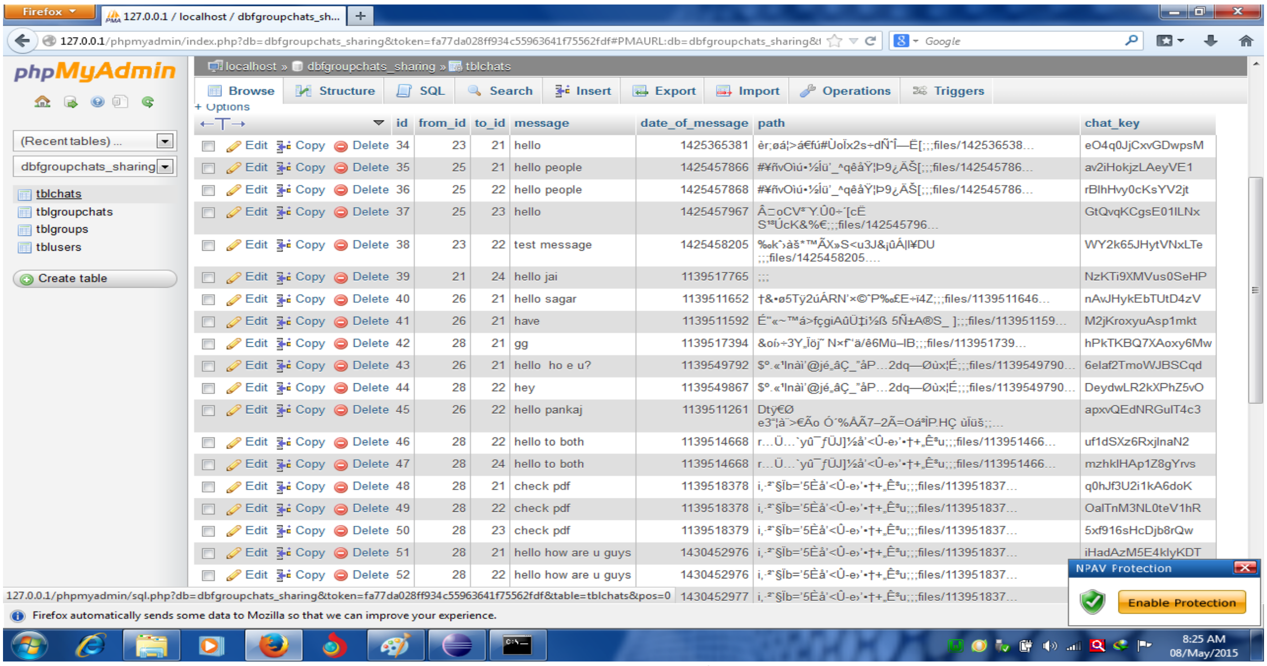
Figure 19 Secret Key Generation after Registration



←T→	id	username	password	user encypted password
□ ✎ ✗	1	sunny	123	95aef5805f83382917a233f5a6e51cd5

Figure 20 Sample Example of AES Encryption in Database

Then user's communications on network then data get encrypted using AES Encryption Algorithm. Figure 20 Shows AES Encrypted format of storage Database [9]. Chat One user to another users/groups of users having data transfer security using AES Algorithm (Create Chat key for each users) having upload files, pictures as an attachment while transfer data from one user to another in encrypted format. See Figure 21 having secured communication between one or more users/group of users with chat key. On Wamp server, result see that, The Android device receives the message sent by server and displays the message on the notification bar and also makes a toast on the screen.



127.0.0.1/phpmyadmin/index.php?db=dbfgroupchats_sharing&token=fa77da028ff934c55963641f75562fdf#PMAURL:db=dbfgroupchats_sharing&t

←T→	id	from_id	to_id	message	date_of_message	path	chat_key
☐ Edit Copy Delete	34	23	21	hello	1425365381	èr;øá¦>á€fú#ÙoÏx2s÷dÑˆÎ—Ë[;;;files/142536538...	eO4q0JjCxvGDwpsM
☐ Edit Copy Delete	35	25	21	hello people	1425457866	#¥ñvOìú•½Íü'_^qêåŸ¦Þ9¿ÄŠ[;;;files/142545786...	av2iHokjzLAeyVE1
☐ Edit Copy Delete	36	25	22	hello people	1425457868	#¥ñvOìú•½Íü'_^qêåŸ¦Þ9¿ÄŠ[;;;files/142545786...	rBlhHvy0cKsYV2jt
☐ Edit Copy Delete	37	25	23	hello	1425457967	Â□oCVª¯Y.Û0÷´[cË S¹³ÚcK&%€;;;files/142545796...	GtQvqKCgsE01ILNx
☐ Edit Copy Delete	38	23	22	test message	1425458205	‰kˆ›àš*™ÃX»S<u3J&¡ûÁ|l¥DU ;;;files/1425458205....	WY2k65JHytVNxLTe
☐ Edit Copy Delete	39	21	24	hello jai	1139517765	;;;	NzKTi9XMVus0SeHP
☐ Edit Copy Delete	40	26	21	hello sagar	1139511652	†&•ø5Tÿ2úÁRN'×©ˆP‰£E÷ï4Z;;;files/1139511646...	nAvJHykEbTUtD4zV
☐ Edit Copy Delete	41	26	21	have	1139511592	É"«~™á>fçgiAûÜ‡i½ß 5Ñ±A®S_];;;files/113951159...	M2jKroxyuAsp1mkt
☐ Edit Copy Delete	42	28	21	gg	1139517394	&oí›÷3Y„Ïöj˜ N×f"ä/ê6Mü–IB;;;files/113951739...	hPkTKBQ7XAoxy6Mw
☐ Edit Copy Delete	43	26	21	hello ho e u?	1139549792	$º.«¹lnàì'@jé„âÇ_"åP…2dq—Øùx¦É;;;files/1139549790...	6elaf2TmoWJBSCqd
☐ Edit Copy Delete	44	28	22	hey	1139549867	$º.«¹lnàì'@jé„âÇ_"åP…2dq—Øùx¦É;;;files/1139549790...	DeydwLR2kXPhZ5vO
☐ Edit Copy Delete	45	26	22	hello pankaj	1139511261	Dtÿ€Ø e3"¦à¯>€Ão Ó´%ÅÃ7–2Ã=Oáª̀IP.HÇ ùÏüš;;...	apxvQEdNRGulT4c3
☐ Edit Copy Delete	46	28	22	hello to both	1139514668	r…Ü…`yû¯ƒÜJ]½å'<Û-e›'•†+„Êªu;;;files/113951466...	uf1dSXz6RxjlnaN2
☐ Edit Copy Delete	47	28	24	hello to both	1139514668	r…Ü…`yû¯ƒÜJ]½å'<Û-e›'•†+„Êªu;;;files/113951466...	mzhklHAp1Z8gYrvs
☐ Edit Copy Delete	48	28	21	check pdf	1139518378	i,-²ˆ§Ïb='5Èå'<Û-e›'•†+„Êªu;;;files/113951837...	q0hJf3U2i1kA6doK
☐ Edit Copy Delete	49	28	22	check pdf	1139518378	i,-²ˆ§Ïb='5Èå'<Û-e›'•†+„Êªu;;;files/113951837...	OalTnM3NL0teV1hR
☐ Edit Copy Delete	50	28	23	check pdf	1139518379	i,-²ˆ§Ïb='5Èå'<Û-e›'•†+„Êªu;;;files/113951837...	5xf916sHcDjb8rQw
☐ Edit Copy Delete	51	28	21	hello how are u guys	1430452976	i,-²ˆ§Ïb='5Èå'<Û-e›'•†+„Êªu;;;files/113951837...	iHadAzM5E4klyKDT
☐ Edit Copy Delete	52	28	22	hello how are u guys	1430452976	i,-²ˆ§Ïb='5Èå'<Û-e›'•†+„Êªu;;;files/113951837...	
					1430452977	i,-²ˆ§Ïb='5Èå'<Û-e›'•†+„Êªu;;;files/113951837...	

127.0.0.1/phpmyadmin/sql.php?db=dbfgroupchats_sharing&token=fa77da028ff934c55963641f75562fdf&table=tblchats&pos=0

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Figure 21 Secure Communication between One/more Group of users using mobile application with chat key

Comparative Analysis with Performance

Comparative analysis of existing system versus proposed system is described as follows.

Existing System Analysis:

According to existing system, the performance benchmark results are shown in Table 2, for 100 key shares generated with a threshold number of 5. The results are found in Figure 22 Shows Rate of Users Deauthorization Based on Share Allocation. The performance data from the GAE server logs are shown in Table 3. The benchmark results show that the processing demands on the mobile device and cloud server are not onerous at all. It describe upload & download encrypted key share with response time, CPU time & response size. The results are shown in Figure 23 Shows total share download based on initial allocation.

Cryptographic operation	Desktop	Notebook	Mobile
Generation of encrypted key shares.	42 ms	56 ms	617 ms
Decryption of encrypted key shares.	0 ms	0 ms	22 ms
Decryption of encrypted user message.	8 ms	46 ms	163 ms

Table 2 Existing System Performance Result

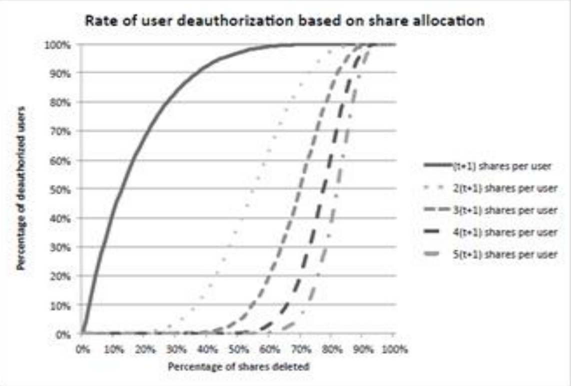
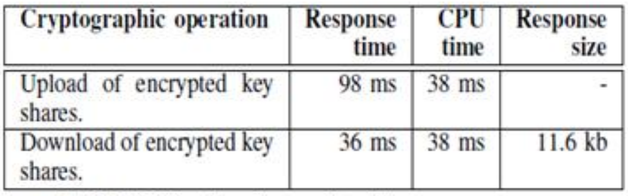


Figure7.12 Rate of Users Deauthorization Based on Share Allocation

DOI: 10.15680/ijircce.2015.0305090



Cryptographic operation	Response time	CPU time	Response size
Upload of encrypted key shares.	98 ms	38 ms	-
Download of encrypted key shares.	36 ms	38 ms	11.6 kb

Table 3 Existing Operation on Cloud Server

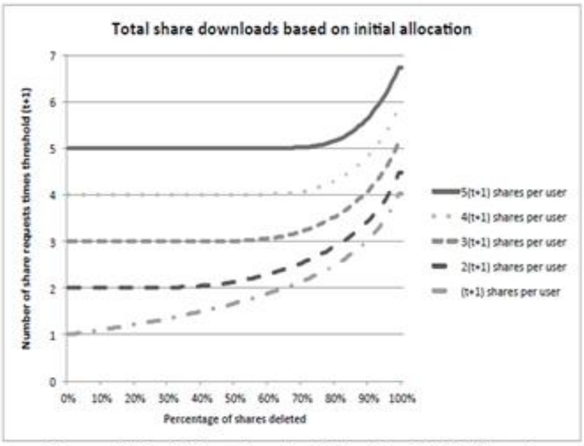


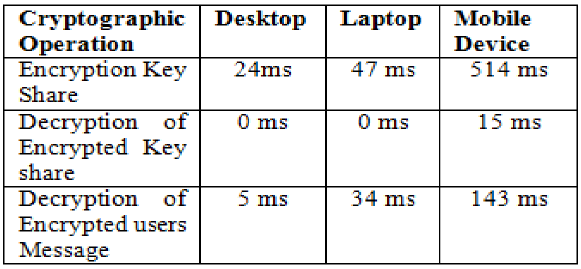
Figure 23 Total Share Download Based on Initial Allocation

Proposed System Analysis:

The implementation was run on different clients to assess their relative performance: a Samsung mobile, Intel Core i7 desktop computer, Laptop/Notebook, Intel Core 2 Duo Desktop computer, and xiomi mobile phone. The Advanced Encryption Standard (AES) cryptographic algorithm was provided by the Java Cryptography Extension (JCE) library. 128-bit AES keys were used as the access keys used to encrypt and decrypt the shares of a 112-bit data key. The Performance results are shown in Table 4, for 110 key shares generated with an increase number up to 5.

On the server end, eclipse IDE as java developer was run as a Java code with server static IP on the Wamp as a cloud server. A connection was established between the desktop or mobile Android client and an instance running on the Wamp cloud via HTTP requests, using Group Chat + mobile application for data interchange and the Jdk 7.0 library for support between Java objects used by the Java client and server. The performance data from the Wamp server logs are shown in Table 5. The results show that the processing demands on the mobile device and cloud server are not combining at all.

Through various parameters were modified to understand their effect on performance. The share deletion on revocation starting with an initial population of 10,000 authorized users, 110 total shares, and a minimum increase time value by (*t*)+ (by SHA1 algorithm) that add values up to 5 shares were randomly allocated to each user. The initial number of shares allocated was increased by a factor of five. If share deletion occurred per round; users were not allowed to request additional shares and a user became unauthorized once his remaining shares fell below the value. The results are found in Figure 24 Shows Graph indicate that de-authorized based on share allocation.



Cryptographic Operation	Desktop	Laptop	Mobile Device
Encryption Key Share	24ms	47 ms	514 ms
Decryption of Encrypted Key share	0 ms	0 ms	15 ms
Decryption of Encrypted users Message	5 ms	34 ms	143 ms

Table 4 Proposed System performance result

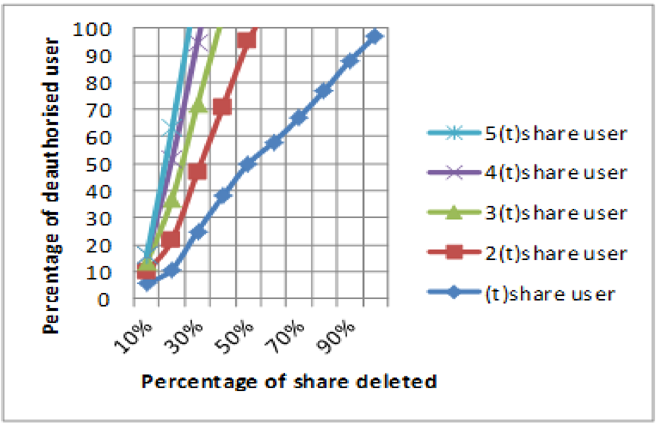


Figure 24 Shows Graph indicate that de-authorized based on share allocation

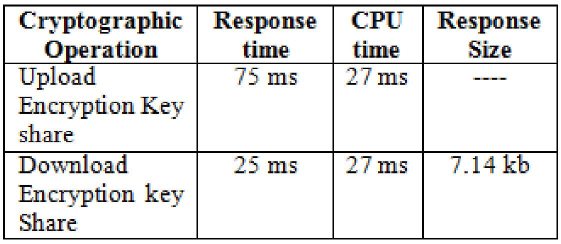
DOI: 10.15680/ijircce.2015.0305090

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015



Cryptographic Operation	Response time	CPU time	Response Size
Upload Encryption Key share	75 ms	27 ms	----
Download Encryption key Share	25 ms	27 ms	7.14 kb

Table 5 Operation of Wamp cloud server

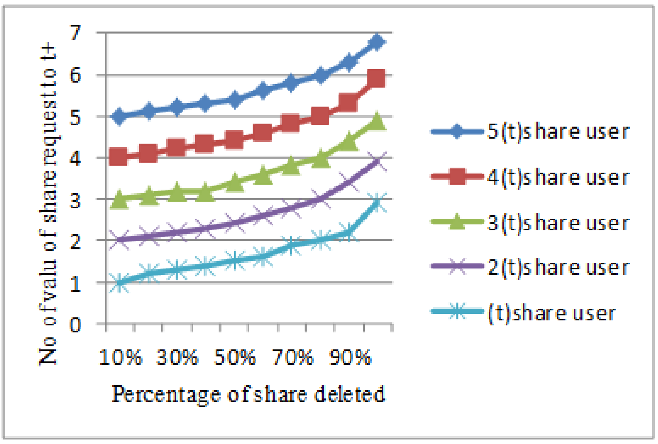


Figure 25 Graph indicate all share download according to primary allocation

Increasing the initial allocation of shares results in a delayed need for additional shares from users becoming unauthorized. So, more storage and communication is possible. It is also allowing users to request shares so that they remain authorized at all times. Keeping the same starting parameters, the initial allocation of shares was varied across a multiple of five to a multiple of one; an unlimited number of random shares were allowed to be requested by each user. The results are shown in Figure 25 Shows Graph indicate all share download according to primary allocation.

And finally after comparative analysis of existing system, the result proves that proposed system work according to the performance table & graphs indicate the values are got in less time. And using Group Chat+ mobile application proves the secure communication between one/more group of uses using cloud hosted key successfully. The main objective of 2 Level Security algorithms is a unique implementation of an extremely secured system on cloud based server.

IX. APPLICATIONS

The various applications provided by created mobile application using cloud server are as follows:
1] Identify and authenticate users before granting access.
2] Secure sensitive and confidential information.
3] Secure vendor relationship before sharing information asset.
4] Prevent external attacks.
5] Convenient access to data.
6] On demand self-service.

X. CONCLUSION

The proposed system create mobile application for mobile devices such as android, smartphones, tablets, & for pcs, desktop, laptops users virtually as their compute and storage nodes for distributed & processing of enhanced data stored on Wamp server with giving security. The application runs on Android 2.2 or higher version of mobile devices. Then, our aim is to provide secrecy to the data as well as keys that are stored in cloud systems. The proposed algorithms provide better data security and key management in cloud systems as well as against attacker. For that, identifying the different functional goals provide security of users data underlying communications. The key gives the mobile platforms to data storage, processing capability & as well as the novel trust model. The reason behind the use of two layer of encryption is that it will be more secured. SQL query is generated and encrypted by AES because even if hackers hack the information and decode the AES encryption part, it will still be more difficult for them to know about the encrypted query. Also, a SHA1 algorithm is provide users password security. This proposed method is an attempt to add more security to databases to avoid attack. Our experimental results show the proposed algorithm can work efficiently on Group Chat+ mobile application for secure data storage on cloud using network at both clients & server side.

XI. FUTURE SCOPE

Throughout this Project, we have systematically studied the secure communication of mobile application using cloud hosted key and privacy issues in mobile cloud computing based on a hybrid methodology. The current mobile application can be further modified and certain objectives can also be fulfilled for these system. Thinking of the app features parameters in a broad manner. Summarizing the presented solutions, promise and give a good base for further research in the area of mobile cloud computing. In this approach each and every process could be done smoothly hence our ultimate aim can be achieved. Further research should try to identify the different method that, how to provide group chat+ application suitable and friendly interactive services for other mobile devices users via play store. We can implement this model in various clouds computing platform to get the more efficient way of cloud computing such as SaaaS, EaaS etc.

REFERENCES

1. Shobha D. Patil1, S. B. Sonkamble., "Survey Paper on Modoc: Multi Owner Data Sharing Over Cloud," (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (1), 6-9, 2015.
2. Piotr K. Tysowski, M. Anwarul Hasan., "Cloud-Hosted Key Sharing Towards Secure and Scalable Mobile Applications in Clouds," International Conference on Computing, Networking and Communications, Vol 1, Sep-2013.
3. Sonam Agrawal, Rajan Dev Gupta., "Development and Comparison of Open Source Based Web Gis Frameworks on Wamp and Apache Tomcat Web Servers," The International Archives of the Photogrammetric, Remote Sensing and Spatial Information Sciences, Volume XL-4, 14 – 16 May 2014.
4. Kanya Devi J, Kanimozhi S, "Efficient User Revocation for Dynamic Groups in the Cloud," International Journal of Engineering and Computer Science, Volume 3, Issue 2, Page No. 3938-3942, February, 2014.
5. Deepa Noorandevarmath, Ramesh kumar H.K, C M Parameshwarappa., "Sharing Of Multi Owner Data in Dynamic Groups Securely In Cloud Environment," International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 6, June 2014.
6. V.Sathana, J.Shanthini., "Enhanced Security System for Dynamic Group in Cloud,"International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 3, March 2014, ISSN: 2277 128X.
7. Jaldi Rakesh, Janapati Venkata Krishna., "A Novel Approach for Secure Data Sharing in Multi-Owner groups in Cloud," IJCTT ,volume 16Oct 2014.
8. T.Vijayalakshmi, Balika J Chelliah, S.Alagumani and Dr.J.Jagadeesan., "An Efficient Security Based Multi Owner Data Sharing for Un-Trusted Groups Using Broadcast Encryption Techniques in Cloud," IJAIEM, Volume 3, Issue 3, March 2014.
9Raaed K. Ibrahim, Ali SH. Hussain, Roula A. Kadhim., "Implementation of Secure Hash Algorithm Sha-1 by Lab view,", IJCSMC, Vol. 4, Issue. 3, pg.61–67, March 2015.
10. Sangeeta Raheja, Shradha Verma Nisha Raheja., "Review and Analysis Of Hashing Techniques,", Volume 4, Issue 5, May 2014.
11. Chaitya B. Shah, Drashti R. Panchal, "Secured Hash Algorithm-1: Review Paper," Volume 2, Issue X, Oct 2014.
12. R.Mekala, S.A.Jiji Jasmine, Vinisha.D, "Accessing Distributed System Using Hashed Fingerprint Recognition," Vol.2, Special Issue 1, March 2014.
13. V.Sathana, J.Shanthini, "Three Level Security System for Dynamic Group in Cloud," International Journal of Computer Science Trends and Technology (IJCST), Volume1 Issue2, Nov-Dec 2013.
14. Sherman, Chow Wang et al., "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Transactions On Cloud Computing Year 2013.
15. Pankaj Verma, J.S Bhatia., "Design and Development of Gps-Gsm based Tracking System with Google map Based Monitoring," International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.3, No.3, June 2013.
16. Thulasimani Lakshmanan and Madheswaran Muthusamy, "A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes," the International Arab Journal of Information Technology, Vol. 9, No. 3, May 2012.
17. P. Tysowski and M. A. Hasan, "Towards Secure Communication for Highly Scalable Mobile Applications in Cloud Computing Systems," Centre for Applied Cryptographic Research,CACR 2011-33, 2011.
18. Kamal Dahbur, Bassil Mohammad, Ahmad Bisher Tarakji, "A Survey of Risks, Threats And Vulnerabilities in Cloud Computing," International Journal of Emerging trends in Engineering and Development, Issue1, Vol. 3,Aug-2011.
19. Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang, "A Survey of Mobile Cloud Computing: Architecture, Applications and Approaches," IEEE,Dec-2010.
20. Itani W, Kayssi, and Cheha, "Privacy as a service: Privacy- aware data storage and Processing in cloud computing architectures," In: Eighth IEEE International Conference on Dependable Autonomic and Secure Computing Chengdu, China, 711-716, 2009.
21. Alex C. Snoeren, "Adaptive Inverse Multiplexing For Wide-Area Wireless Networks," In Proceedings of IEEE Globe.Com, December 1999.

Web Resources/White Papers

22. Doncho Minkov, "Android SDK 2014," http://www.minkov.it, www.academy.telerik.com
23. CSA, "Security Guidance for Critical Areas of Focus in Cloud Computing," V2.1, 2010, http://www.cloudsecurityalliance.org/guidance/csaguide.pdf.

BIOGRAPHY

Mr.S.W.Thakre student of 2nd Year (CSE Dept.) G.H.R.C. O.E &M Amravati. & **Prof.N.R.Chopde** is a H.O.D in the C.S.E Department, G. H. Raisoni C.O.E & M, Sant Gadge Baba Amravati University, and Amravati. He received Master of Engineering (ME) degree in 2011 from SGBAU, Amravati, MS, India. Her research interests are Computer Networks (wireless Networks), HCI, Algorithms, web 2.0 etc.