User Authentication in Cloud Computing-Using Seed Chain Based One Time Password (OTP)

Debayan Bhattacharya
Department of Computer Science (Hons.)
Vidyasagar College, Kolkata, West Bengal, India

Abstract- Cloud computing has emerged as a popular model in computing world to support processing large volumetric data using clusters of commodity computers. It is the latest effort in delivering computing resources as a service. It is used to describe both a platform and a type of application, therefore removing the need of providing these services themselves. This can for example lead to cost savings, better resource utilization and removing the need of technical expertise for the customers. However, cloud services also present a couple of issues. Since the resources are put under another provider, the customer will have no control over the situation. Since the control of services and data needed for the everyday-run of a corporation is being handled by another company, further issues needs to be concerned. The consumer needs to trust the provider, and know that they handle their data in a correct manner, and that resources can be accessed when needed. This thesis focuses on *authentication* in cloud services. The current solutions used today to login to cloud services have been investigated and concluded that they don't satisfy the needs for cloud services. They are either insecure or complex. This thesis have resulted in an authentication and registration method that is both secure and easy to use, therefore fulfilling the needs of cloud service authentication. The conclusions that can be drawn is that the proposed security solution in this thesis work functions very well, and provide good security together with an ease of use for the clients who don't have so much technical knowledge.

Keywords – MD5, One Time Password, Secure Socket Layer, Steganography.

I. INTRODUCTION

Cloud computing is a universal word for anything that involves distributing hosted services over the web or Internet. It can be an internet-based computing infrastructure that allows users to access different level of IT resources remotely through internet based client-side software as if it were installed locally in users own computer. Where the IT resources include server, storage, service, application, network and so on. These resources are associated in a large computer network which is owned by a company (Both privately and publicly). Cloud computing also provides services to others devices (such as smart-phones) on demand over the Internet [13] [11]. Companies, business organizations, academic or commercial researchers and any individual can be user of cloud computing.

However, cloud services also present a couple of issues. Since the resources are put under another provider, the customer will have no control over the situation. You don't know how your data is treated in the cloud, how sensitive data is encrypted, how the provider's handle redundancy and backup of your data, can the resources always be accessed etc. The main cloud service providers are Amazon, Salesforce and Google. Examples of large and well reputed IT firms that are dynamically involved in cloud computing are Microsoft, Fujitsu, Hewlett Packard (HP), IBM, Dell and VMware [13]

One of the bigger issues is the security part and one of the most important parts for a company that is thinking of moving services to the cloud. They need to know that their data is safe, both at the provider's site and during transmissions between the host and server. Furthermore, the authentication procedure must be very secure; the best encryption algorithms in the world will not protect the data if someone has figured out your password.

Since cloud computing is a quite new subject, most of the cloud providers have not yet tighten up their security and still use insecure or complicated login methods. The authentication part of cloud computing must be easy and flexible for the millions of user that it has, but at the same time be very secure to protect the data that it stored in the cloud. At the same time the transmissions (user request & response to and from server) must also be done through very secure channel.

A. Problem

The most common login form used today, not only for cloud services, is to use static passwords. Many can agree that static password have a lot of security problems. Static passwords are often very easy to crack, since users prefer non-complex passwords. The users also rarely change their passwords or use the same password to access multiple services. Therefore, different cloud providers have lately started *two factor authentications* with *one time password*. At the same time, the security must be easy for the customers to understand and appeal to all kinds of people with different technical knowledge. And lastly, the security solutions should be very cheap or free of charge to implement, both for providers and customers, to attract more people to the cloud. So, in conclusion, for cloud services to grow even more, it needs a simple and cheap security solution.

B. *Approach chosen to solve the problem*

This paper proposes that the one-time password which is created in the server side and is provided to the user via user's email id as an email, assuming that most of the people have their own email id, the problem with a separate authentication device for two-factor authentication is solved. So in this way we can remove the use of any additional gadget (it may be OTP generator device or by using smart phone to generate OTP) to generate OTP.

Furthermore, by using open source code at the cloud authentication server, the security solution is absolutely free of charge. That solves the problem of providing a free of charge security solution.

C. Goals

The goal with this paper is to implement a working authentication solution, which can be used in cloud services. The authentication method will be provide good authentication service with an *one time password* which is generate in the cloud server and provided to the user during login time, this password is only valid one time for a certain amount of time. The password will only be given to the user after successfully given all the relevant information correctly at the login page.

The rest of the paper is organized as follows. Backgrounds of my topics are explained in section II. Proposed Security Solutions are presented in section III. Experiment and the Results are shown in section IV. Conclusion is drawn in section V.

II. BACKGROND

2.1 *Existing problems in Cloud Computing*

Cloud computing has turned into a standard information technology operation for many small or large businesses. It offers many considerable advantages, including probable expenditure savings. There are, however, major risk and disadvantages related with cloud computing.

Its dislocated nature is a benefit in many cases however can also be disadvantageous because the user has no supreme control over the software applications including secret data. Client has to depend on the provider to update, upgrade maintain and administer it. The user does not have direct access to the software to fix the problems while something goes wrong in any application and must rely on the service provider. The user can experience significant problems when the cloud provider is uncaring or incapable to fix the problem quickly.

In the same way, if a company becomes reliant on cloud-based services and the provider is unable to continue with their services, you will rapidly run into trouble. This trouble would quickly turn into much worse if the provider was not sincere to give any prior notice in time to allow your business to take an alternative cloud service.

Cloud computing can also mean big risks in the integrity, privacy areas and also greatly in users *authentication*. Using a cloud system, company's susceptible data and information will be stored on third-party servers, and user will possibly have very inadequate understanding or control regarding this information. If the provider has insufficient security, or a violation of encryption systems or procedures are performed for different reasons, thus compromised company's private and confidential data. This could have devastating consequences, and could cause lawful problems for company if third party private information (for example, customer information) is negotiated.

There are several problems in cloud computing and this thesis work is mainly focused on authentication based security issues in cloud computing and how it can be mitigated, the remaining part of the paper describes about this.

2.2 Objectives

The main objectives of my work are to securely register the user over the internet and to securely login into cloud interface by using *cloud server generated one-time password without using any OTP generating device or a smart phone as an authentication device* to access their respective services. For this purpose we need to focused in the following points-

2.2 .1 Authentication

In general authentication is the act of creating or validating something (or someone) as authentic and claims made about the topic are true. This might engage proving the identity of a person or assuring that a computer program is a trusted one. In computer networks and Internet or any web based services authentication is usually done using the login password. Knowledge of the password is adopted to ensure that the user is authentic. Each user registers first or get registered by someone else and using an assigned or self-stated password. On each subsequent use, the user must know and use the previously declared password. *The weakness of this system is that passwords can often be stolen, unintentionally revealed or forgotten.* There is a couple of possible authentication security attacks- *a) Eavesdropper attacks b) Keylogger attacks c) Man-in-the-middle attacks d) Password discovery attacks e) Replay attacks f) Social engineering attacks etc*. As cloud computing is a web based application so it might get these aforesaid attacks. In order to protect the cloud computing services from authentication attacks, it must need a very secure and strong authentication system. Therefore secure authentication in cloud computing is significantly important.

2.2.2 Static passwords

A static password which is usually a secret word or phrase picked by the user and used together with the user's username to authenticate a user when login to a specific service. Even though static passwords is used almost everywhere like an e-mail system, an online community etc., from a security point of view it has a lot of problems. The main weakness with static passwords is the human interaction when choosing the secret pass phrase. If the password is too simple, it will be exposed to different kinds of threats where an attacker will try to crack it, such as social engineering, Trojan attacks, Password attacks, key loggers or by just trying to guess the password. On the other hand, if the user picks a very hard pass phrase it will be very hard to remember, leading to writing the password down on a piece of paper and store it under the keyboard, which is a big security risk. It can also lead to more work for the IT administrators when users forget their passwords, forcing the administrator to take valued time to reset passwords. One other major risk with static passwords is at the same time user use the same password for many different sites and services, without changing it.

2.2.2 One Time Password

Since the problems with static passwords, many have now started to use one time passwords as the login procedure for different services. One time passwords: - A *one-time password* (OTP) is just what the names implies, a password that is only valid for one login. The benefit of OTPs is that it offers much higher security than static passwords, in expense of user friendliness and configuration issues. OTPs is immune against password sniffing attacks, if an attacker use software to collect your data traffic, video records you when you type on your keyboard, or use social engineering, it doesn't matter since the password that the attacker gets hold on will not be valid to use. An OTP can be generated using different methods- *a) Time-based OTPs b) Counter-synchronized OTPs c) Seed-chain OTPs d) Challenge-based OTPs*. [12] OTP is much safer than static passwords, when looked at from an access attack perspective, such as sniffing, password cracking and social engineering. However, it cannot protect against two common attacks: - *a) Man-in-the-middle attack. b) Trojan attack.* These two attacks are best solved by educate users in how to spot web pages with false certificates and how to protect your computer and keep anti-virus software up to date. That is out of the scope of this thesis.

2.2.3 MD5

MD5 [5] is an algorithm which is used to verify data integrity through the creation of a 128-bit message digesting from data input (which may be a message of any length) i.e. claimed as unique to that specific data likewise fingerprint is to the specific individual. MD5 which was developed by Professor Ronald L. Rivest of MIT is intended for use with digital signature applications. Its requirement is that large files must be compressed by a secure method before being encrypted with a secret key, under a public key cryptosystem. It is "computationally infeasible" i.e. any two messages that have been input to the MD5 [5] algorithm could produce the output, as the same message digest.

2.2.4 SSL

SSL [8] (pronounced as separate letters) is short term for Secure Sockets Layer, a protocol developed by Netscape for transmitting private documents via the Internet. It provides Transport Layer Security. It establishes an encrypted link between a server and a client— typically a web server (website) and a browser; or a mail server and a mail client (e.g., Outlook).More specifically, SSL [8] is a security protocol. Protocols describe how algorithms should be used; in this case, the SSL [8] protocol determines variables of the encryption for both the link and the data are

being transmitted. Both Netscape Navigator and Internet Explorer support SSL [8] and many Web sites use the protocol to obtain confidential user information, such as credit card numbers, social security numbers, and login credentials etc. By convention, URLs that require an SSL [8] connection start with —*"https:"* instead of —*"http:"*.

III. Proposed Security Solution

The previous chapter mentioned the problems with static passwords and also other problems associated with different cloud provider's security solutions, and how it can't be used satisfactory in a cloud environment.
There are a few ways to have a secure and easy-to-use cloud service that can satisfy these criteria:

1. By providing better password solution for login procedures than the insecure method of static passwords.
2. By providing better OTP authentication solution than those discussed in the previous chapter.
3. By having an easy-to-understand registration system that at the same time doesn't compromise the security.
4. By using a secured communication channel for every network communication between server and the user machine.
5. By offering a solution that is free of charge in order to attract more customers to the cloud services.
6. In overall, the security solution for cloud services must be easy to use, but also be very secure in order to protect the customer's data and gain the trust of the customers.

The solution presented here will be free of charge for the users, it is easy to use and at the same time it is very secure to protect the customer's authenticity and to gain trust of the customers.

3.1 Solutions

3.1.1 Proposal – 1 Authentication with OTP

3.1.1.1 Login Process

The criterion for the proposed security solution for cloud services is that it needs to be secured but at the same time it should be easy to use. The fact that cloud services have a growing market with millions of user makes it important for every user to understand and know the login process to the service.
The authentication method is used in this paper based on algorithm used by Johnsson et. al. [12] but with drastic modifications. As I think there is a problem to use the current time as a parameter for generating OTP as if there is any communication problem occurred between the client device and the cloud server side then the current time value may vary and as a result there may be a mismatch problem of the hash value which is generated separately in the client side and the cloud server side in that case the password (hash value) cannot be verified at the cloud server side and the user face unnecessary problem to get the access of his / her service. Beside that there is another major problem in this OTP generating process –the user have to depend on an extra gadget like smart phone or OTP generating device to generate OTP. And it is a very big drawback because if this device is getting stolen or the device is broken then user cannot be able get OTP.
So I suggest the following steps:-
A client who wishes to login to his / her personal account, surfs the login page through a web browser.
There will be three phase for login process-

1. At first step there will be three field in the login page- i) User Name, ii) Captcha [4][10] iii) Human verification.
2. In the second step the cloud server will send a new OTP which is created by MD5 [5] hash algorithm to the user's valid email id.
3. Now at the third step the user will use this newly created OTP along with his / her User Name for login.

- *Every communication will be done by SSL [8].*

At first the user requests to the cloud server for the login page. Then the cloud server gives a response to the user which includes three fields – 1. The user name or user-id, 2. Captcha [4][10] and 3. Human verification code, the purpose of these three fields are- The input value in the human verification field is as same as the captcha [4] [10]value, captcha [4] [10] is an auto generated random pictorial number which ensures that the end user who sends the request to the cloud server is a human being not a robot or not any auto generated system. In this way we can avoid *the flooding attack* to the server.

From the user name or user-id we can track the previously generated OTP from the database and pass these three values – user name or user-id, human verification code and previously generated OTP as a parameter to the MD5 [5] hash algorithm.

Now the MD5 [5] algorithm will generate the hash value, from this hash value we use the first five digit as an OTP which will be send to the user's valid email id via a secured (SSL[8]) connection.

Here we are using the old OTP as seed value to create a new OTP. It means- After registration when user login into his / her account for 1st time the OTP is created by using user name or user-id, human verification code and the OTP which is stored in the database generated during registration process. When 2nd time the same user wants to login his/her account then we use this previously used OTP as a parameter, which means:- 2nd time when the user inputs his / her user name or user-id and the auto generated challenge or human verification code ,then in the cloud server side it takes the following things as a parameter to the hash algorithm – 1) user name or user –id , 2) the auto generated challenge and 3) fetch the previously used OTP value from the database and after that the cloud server runs the hash algorithm and sends the new OTP to the user via an e-mail to the user's valid email id and also save this new OTP in the database for future use, irrespective of that whether he/ she cannot successfully complete his login process due to his own fault of operation (i.e. suppose he puts the wrong OTP or he waits more than specific times after getting the OTP from the server) . To achieve this we must design the database very carefully, especially during selecting the primary key. (According to this point, if I have chosen the user id as primary key then it helps to fetch the information i.e. the user's email-id , the saved OTP (previously used) etc. from the database quite efficiently).

In this method if the attacker even trace the previously used OTP during transmission it can't do anything as the OTP is valid for only one time use, after the user use it for login purpose it has no special significance.

But here we must provide some time constraints in the following way that the newly created OTP has some time specific lifetime (for an example OTP is validated for 3 minutes) and user must use this OTP during this time period otherwise the validity of this OTP will be gone and the user can't use this OTP. By this way we can protect the user in those cases where the respective user is very careless.

So in this way we can generate OTP in a very efficient way with-out use of any additional gadget (OTP generator or smart phone as an OTP generator).

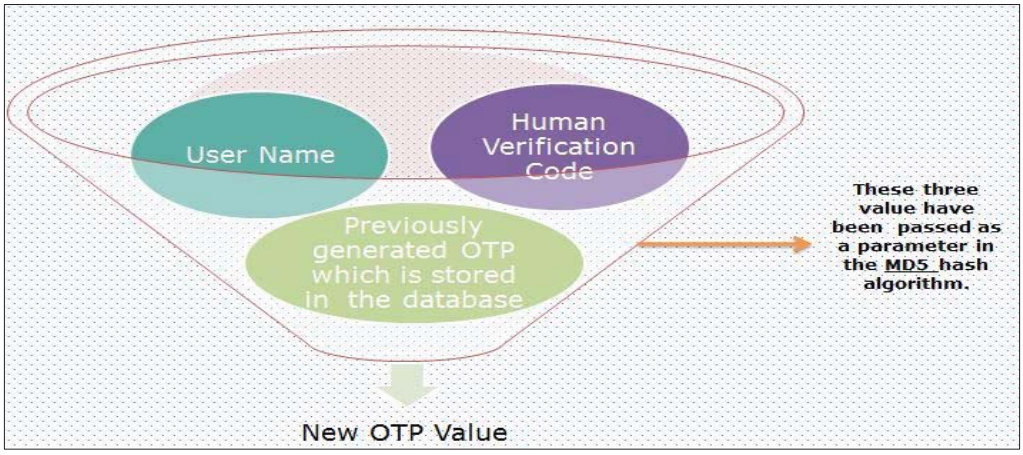


Fig: 1 Pictorial Representation of Login Process

3.1.1.2 Registration Process

How would cloud providers register clients to their authentication databases in a flexible and cheap way? If I follow the following steps, it can give desired result-

- User browse and request for specific service from the cloud by enters URL of application.
- The cloud system response and sends registration form to the user (if he / she are a new user).
- Receiving upon the registration form, the user fills all the fields and submits it.
- The server checks and processes the registration from.

The server checks the following things- the user name or user-id provide by the user should be unique (as I previously said that user name or user-id is the primary key of the relevant database), email id should be in proper format, the human verification code should be same as the captcha [4] [10] etc.. If the details is being entered by the user are correctly then in the server side the MD5 [5] hash algorithm (passing the values of i) user name or user-id, ii) date of birth, iii) human verification code as a parameter) will be executed and an OTP (5 digit) will be sent to the user valid email id message via secured network connection (by using SSL [8] connection).

- User will enter the OTP value within the specific time (3min) and press the submit button.
- After that Authentication server will send message of successful registration to the user.
- Every communication is done by SSL [8].

It will provide a safe method to registrar to a service and also in this design user does not need any static password at any stage. So, the user does not need to bother to remember any password, its only require to remember the user name, one time password will be generated and managed automatically at the server side. *But it cannot protect from the Man-In-Middle attack, people need more studies on how to spot a fake web page. It is out of the scope of this paper.*

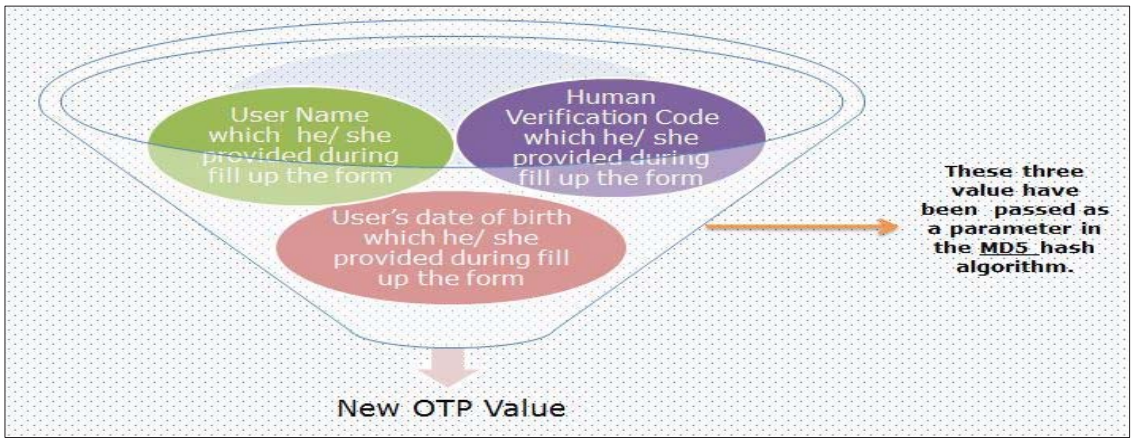


Fig: 2 Pictorial Representation of Registration Process

3.1.2 Proposal – 2 Authentication with picture embedded OTP using Steganography

3.1.2.1 Login & Registration Process:-

This proposal is similar to the first proposal, (OTP generating algorithm will be same as the Proposal-1) except that one additional security parameter is added to the system- that is data hiding into the digital picture or *Steganography* [7].

This technique which is very common in use for transmission of messages in several hidden forms, have been in practice all over the world for long time. Apart from the difficult methods of data encryption and mathematical algorithms, there is a simple method known as digital steganography [7], in which messages are embedded into innocent looking images.

In this technique when server sends the OTP (for both registration and login time) to the user then it hides the OTP into a digital image and sends this digital image to the user's mail id. After getting the mail user downloads this picture to his / her own machine and after that decodes that image using the service provider's provided software and getting his / her OTP. In this way we can provide an extra security if the attacker can be able to hack the user's mail id (we assume the worst case though all the connection is done through SSL [8]) though the attacker cannot be able to take any fraud attempt, as first of all by seeing the picture in naked eye it can't be possible to recognize that the picture content any hidden data, secondly if he / she (attacker) try to decode the image then he / she can't be able to getting success as he /she don't know which type of algorithm is used for embedding purpose.

3.1.2.2 Steganography Algorithm- LSB Substitution

3.1.2.2.1 Introduction

Hideous transmission of message is always a need for humanity. The objective of such hideous message transmission is either to make it so hard for anyone other than the intended recipient to crack the message or to make the carrier of the message so simple that no one might suspect the existence of a message at all.

The first technique for hideous transmission of message is called digital encryption, with symmetric and asymmetric tools and techniques available. It is widely used for banking and defense applications where the message needs to be so strongly encrypted that it is virtually impossible to figure out the message even if one knows about its existence.

The second technique is like the code word system used by children while playing. They might use a word which does not mean the commonly understood meaning. Digital Steganography [7] is one such innocent looking tool, where messages are carried inside digital images and the same cannot be decoded unless the recipient knows about the existence of such a message inside the image.

For unsuspecting persons, the steganographed image containing a digital message does not show any difference from its original to indicate the existence of a message at all. The image can be viewed using any of the common tools. But, editing the image might cause loss of the message.

3.1.2.2.2 Format of Image Data

Images are constructed using tiny dots named pixels. Each pixel has got its own attributes for displaying colour and transparency. There are several systems available for representing colour in image pixel. The most common system for representing colour is the RGBA [6] system – which stores pixel data in the form of red, green, blue and alpha (transparency). In this paper I uses RGBA [6] system for storing and manipulating pixel data.

Under RGBA [6] system, first 8 bits (0 to 7) of the pixel belong to Alpha value or the transparency value. The second 8 bits (8 to 15) represent blue colour, third 8 bits (16 to 23) represent green colour and the last 8 bits (24 to 31) represent red colour.

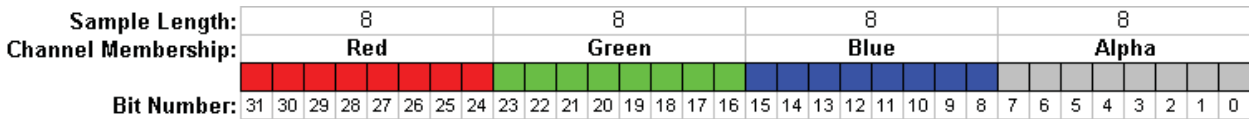


Fig: 3 Organization of Pixel under RGBA [6] System

Now that the pixel level organization of RGBA [6] system is clear, we should understand that that the maximum value for each parameter of RGBA [6] system is 28, i.e., 256. Maximum value is shown when the pixel stores one in all bits. If a change is made to the value at the least significant bit, i.e. the bit location 0 for alpha value, 8 for red value, 16 for green value and 24 for blue value, the impact is likely to be 0.39% (1/256*100). Since the change in original value is very low, we might use the least significant bit of any or all the four RGBA [6] bytes for storing the information we wish to transmit incognito.

3.1.2.2.3 Strategy for Storing Message in an Image

The present technique uses only the least significant bit of the alpha part of a pixel. This technique does not modify any colour value. Before embedding the message, the length of the message should be written into the image. This will exclude the appearance of junk values in the decoded message.

After extracting bit number 0 from the first 32 pixels, the bits should be neatly arranged inside an integer variable to know the length of message embedded into the image. Pixels following the 32nd pixel store the bits needed for reconstructing the byte value needed to create the original string.

Hence, an image with 1 million pixels (or 1 Mega Pixel) might be able to store a message containing a maximum of 1, 24,996 characters. ((1,000,000-32)/8 = 1, 24,996). Although 1 Mega Pixel image is considered a low resolution image, it could store a lot of characters in the form of a text message. Maximum size of message that could be embedded in an image at the rate of 1 bit from each pixel can be calculated using the relation n= (P-32)/8. If we increase the storage locations for message to the least significant bit of all the four components of RGBA [6] system (pixel numbers 0, 8, 16 and 24), the storage capacity increases to n = (4P-8)/8. Here, n is the maximum length of message and P is the number of pixels.

3.1.2.2.4 Image Format Suitable for Embedding Messages

Most of the image formats store image data in some form of compression. The compression algorithms used for image data can be divided into two broad categories: i) *lossy compression algorithms* (JPG, GIFF etc.) and ii) *loss-less compression algorithms* (PNG, BMP, DEB, etc.).

Lossy compression algorithms result in significantly small file size. But, the actual value of each pixel of the original image is not preserved. i.e., the algorithm achieves very high compression level and very small file size by sacrificing the exact value of each pixel and subjecting pixel values to some form of grouping. *Lossy compression* is not suitable for steganographic transmission of messages, since the pixel values may be modified by the algorithm after we embed the message.

In *loss-less compression*, the algorithm compresses the image, but does not make any changes to the value of each pixel of the original image. The *loss-less compression* algorithm is suitable for storing steganographic messages. The pixel values of the new image are the exact replica of the original image except for the bits we modified for embedding our message.

In the present case, the image containing required message should be saved in loss-less compression formats like PNG, BMP, DEB etc.

IV. EXPERIMENT & RESULT

The purpose of the experiments is to build up the security solution that has been discussed in previous chapters. The scenario is this experiment simulates a major cloud provider that wants to offer a secure, fast and easy way for customers to login and registers for their services, and all should be done through a web browser. And not only security when connecting to a cloud, it can be applied to any kind of server-client operation.

In order to the system work, I have used PHP as a scripting language (which is run on Apache server) for register a new user, login of an existing user and for generating OTP in the cloud server side. I have used MySql database for storing the user details information and OTP. And for Steganography [7] purpose I have used Core Java language.

Though I have said that cloud server will use user's e-mail id to send OTP in my proposed architecture of algorithm, I have not implemented this method in this study. I have used pop-up method in which cloud server will send OTP through a pop-up window. Not only that, but also I have not implemented this method using SSL [8] connection. Actually I only want to present my logic which can create OTP successfully. E-mail method and SSL [8] connection method is easily available in market. That is why I have used my login in a unique way. Here I fully implemented the Proposal 1 and from Proposal 2 I only showed the Steganography [7] algorithm.
So, the output of my project is being presented below-

4.1 Registration Process:-

Suppose we are intended to add a new user named Bob Biswas, so here are a few steps required for registration purpose-

Step 1:- *At first, fill all the fields shown in the registration page shown in figure 4.*

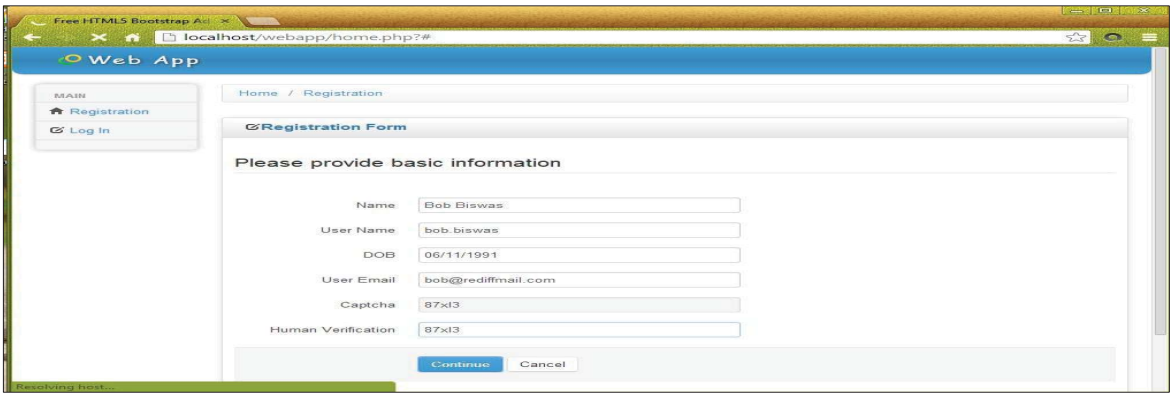


Fig: 4 Registration process step-1

Step 2:- *After that the OTP generates at the server side and sends it to the user, shown in figure 5.*

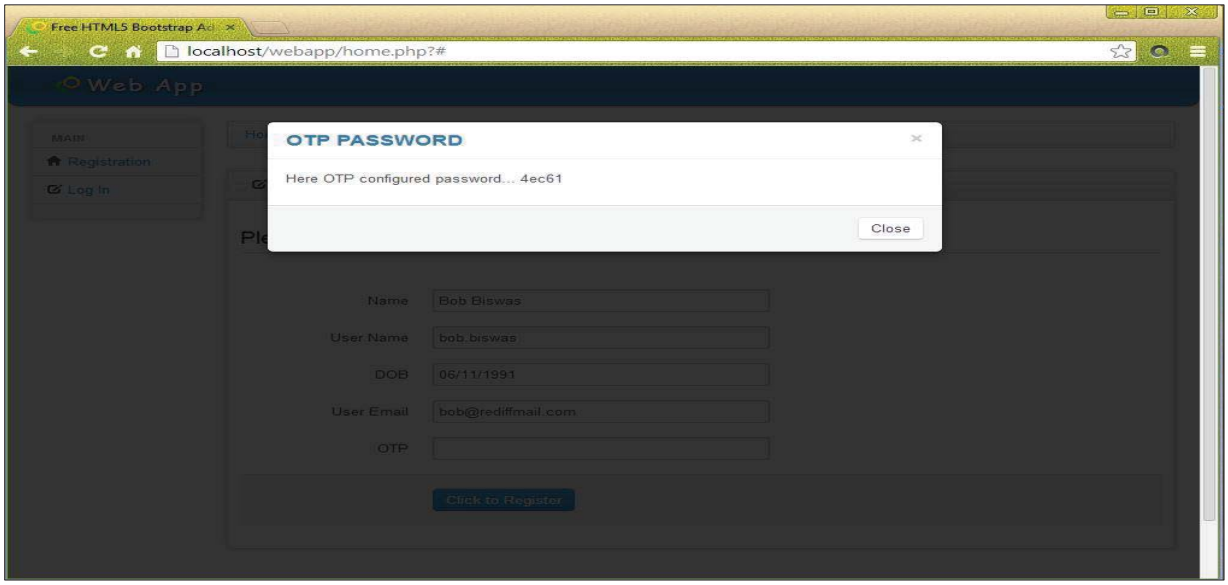
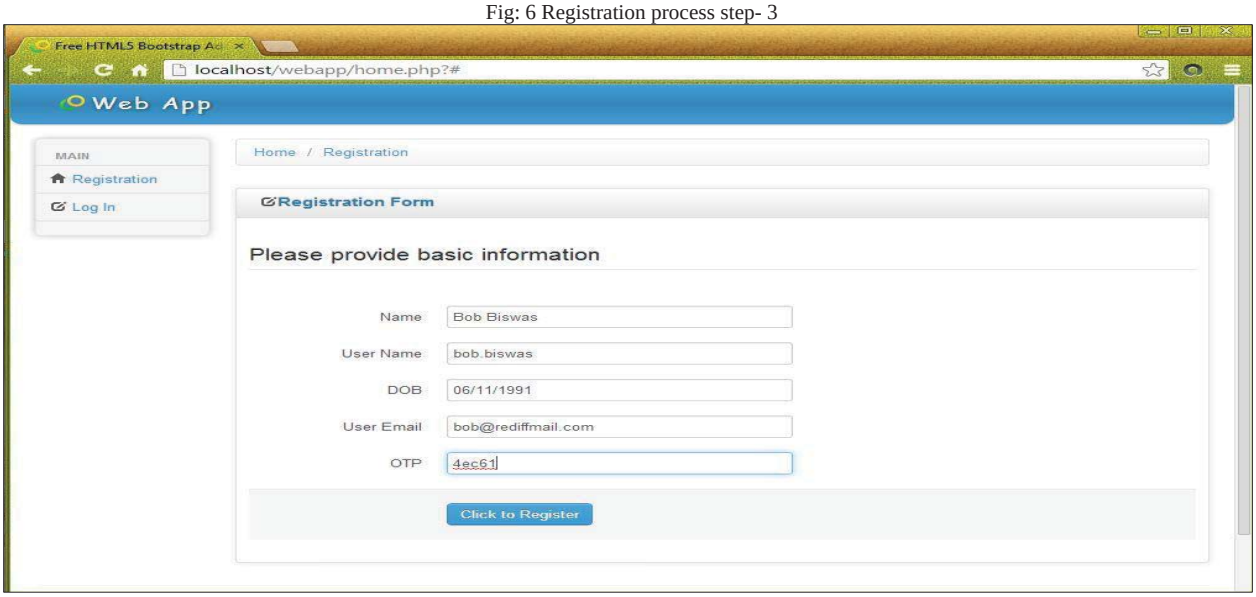


Fig: 5 Registration process step-2

Step 3:- *We will put the OTP value in the OTP field and click the* ***Click to Register*** *button, shown in figure 6.*

Fig: 6 Registration process step- 3



Step 4:- *This is the final stage where the user successfully complete his / her registration process, shown in figure 7.*

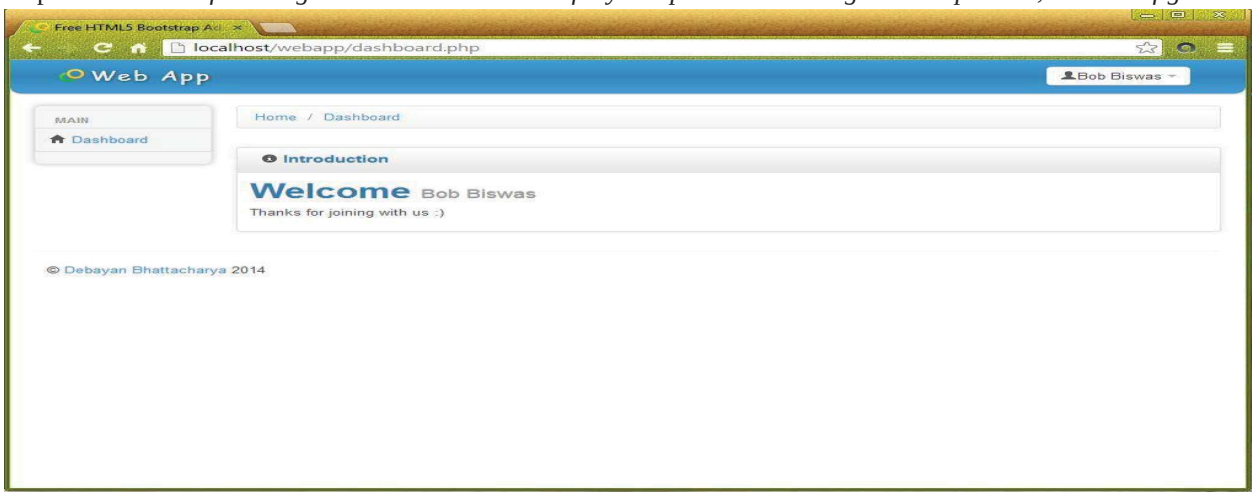


Fig: 7 Registration process step- 4

Here is the database snap shot, we can see that the new user named Bob Biswas is successfully added to the database, shown in figure 8.

id	name	uname	otp	email	dob
1	soumyajit1	soumya1	be667	soumyajit.cn@gmail.com	12/03/2013
2	soumyajit	soumya	a6223	soumyajit.cn@gmail.com	12/01/2013
4	debayan	debayan	77471	deb@gmail.com	04/08/2013
9	nil	nil	104c2	nil@gmail.com	01/16/2014
10	debbayan	bdebayan	22f22	bdebayan@gmail.com	01/16/2014
11	Debayan Bhattacharya	debayan.b	01d53	deb.bhattacharya@outlook.com	03/11/1989
12	Bob Biswas	bob.biswas	4ec61	bob@rediffmail.com	06/11/1991

Fig: 8 Database snap shot

4.2 Login process:-

To show the steps of the login process I am taking the user Bob Biswas as an example. Before login if I look at the database snap shot, here we can see that the OTP value regarding to this user is 4ec61 which is created and stored during registration time, shown in figure 9.

12	Bob Biswas	bob.biswas	4ec61	bob@rediffmail.com	06/11/1991

Fig: 9 Database Snap shot of Bob Biswas before login

Now we show the login process step by step:-

Step 1:- *At first, fills all the fields shown in the login page shown in figure 10.*

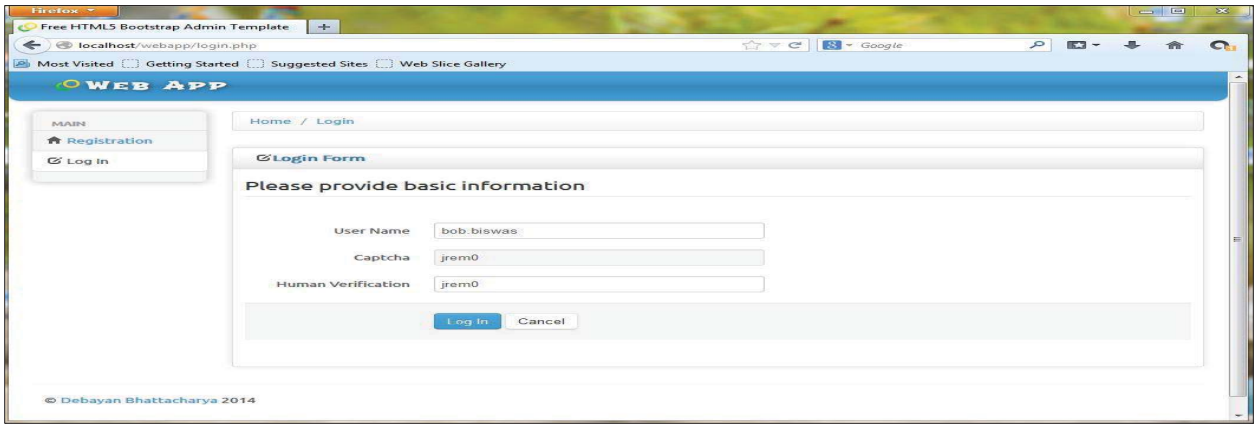


Fig 10. Login process step 1 (first time)

Step 2:- *After that the OTP generates at the cloud server side and sends it to the user, shown in figure 11.*

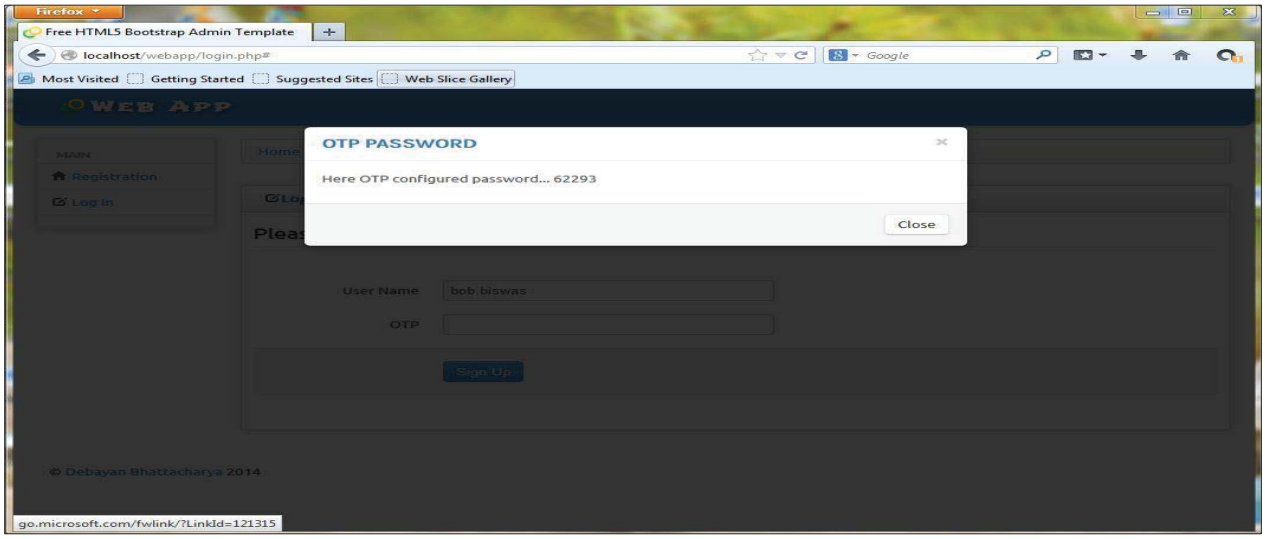


Fig: 11 Login process step 2 (first time)

Step 3:- *In this step we put the OTP value in the OTP field and click the Sing Up button, shown in figure 12.*

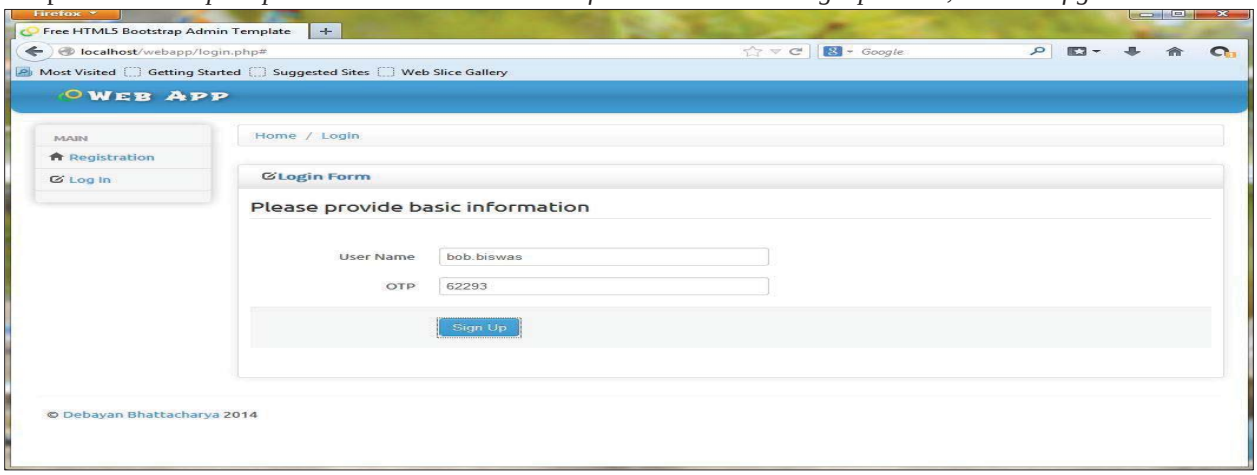


Fig: 12 Login process step 3 (first time)

Here we can see that the value of OTP is *62293*.

Step 4:- *Then the user can successfully complete his / her login.*

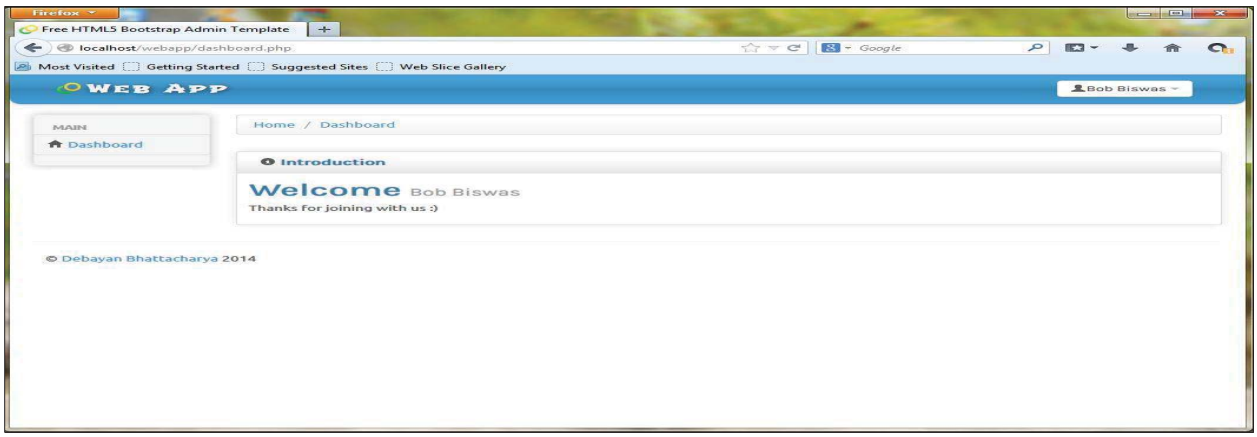


Fig: 13 Login process step 4 (first time)

So now at this stage if we look at the database (shown in figure 14) we can see that this new OTP value- *62293* is stored in the database by replacing the previous OTP value- *4ec61*, and this new OTP value is used next time to create new OTP.

id	name	uname	otp	email	dob
12	Bob Biswas	bob.biswas	62293	bob@rediffmail.com	06/11/1991

Fig: 14 Snap shot database after successful login in the database

Now suppose at second time the Bob Biswas tries to login to access his service but unfortunately he cannot successfully complete his login process due to his own fault of operation (i.e. suppose he puts the wrong OTP or he waits more than specific times after getting the OTP from the server) then also the newly created OTP is stored in the database by replacing the old one, the result is shown in the figure 15a to 15e.

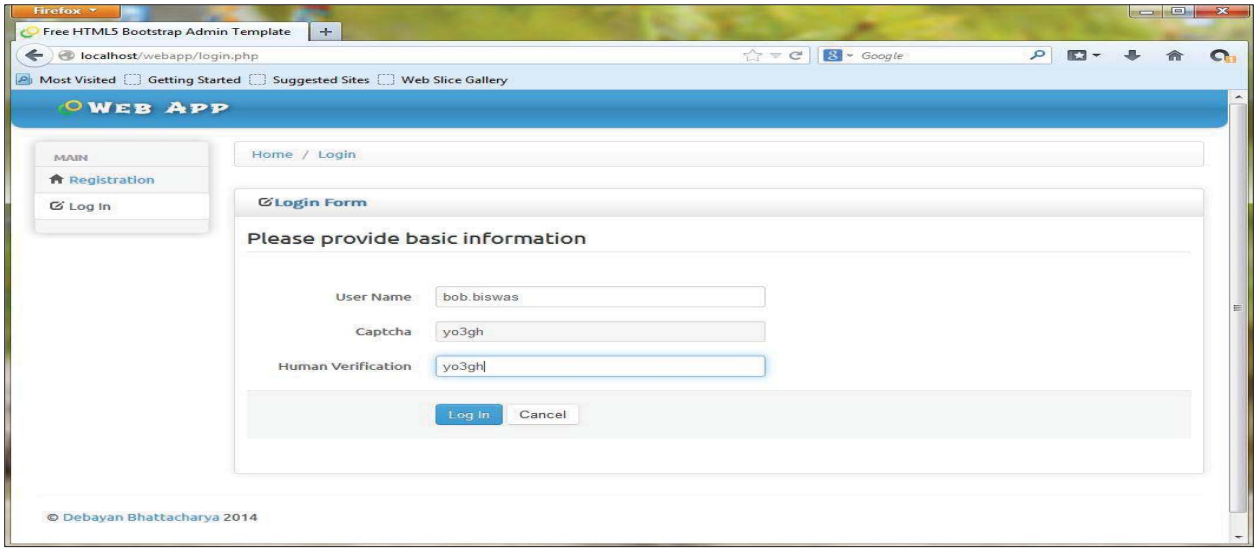


Fig: 15a Login process step 1 (second time)

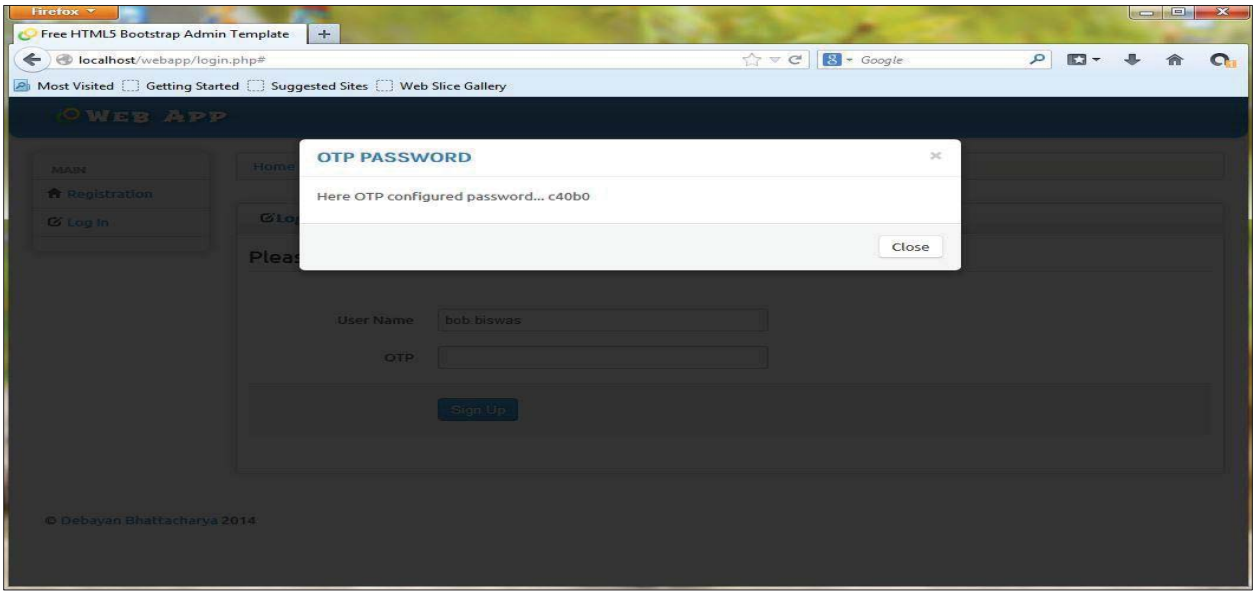


Fig: 15b Login process step 2 (second time)

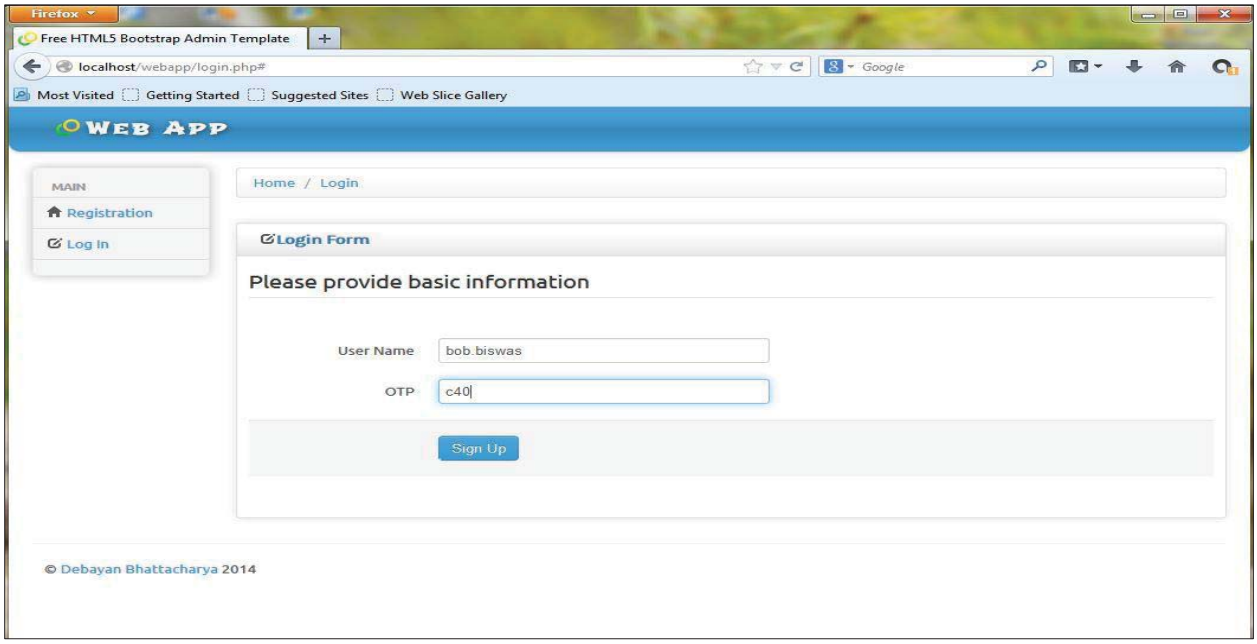


Fig: 15c Login process step 3 (second time)

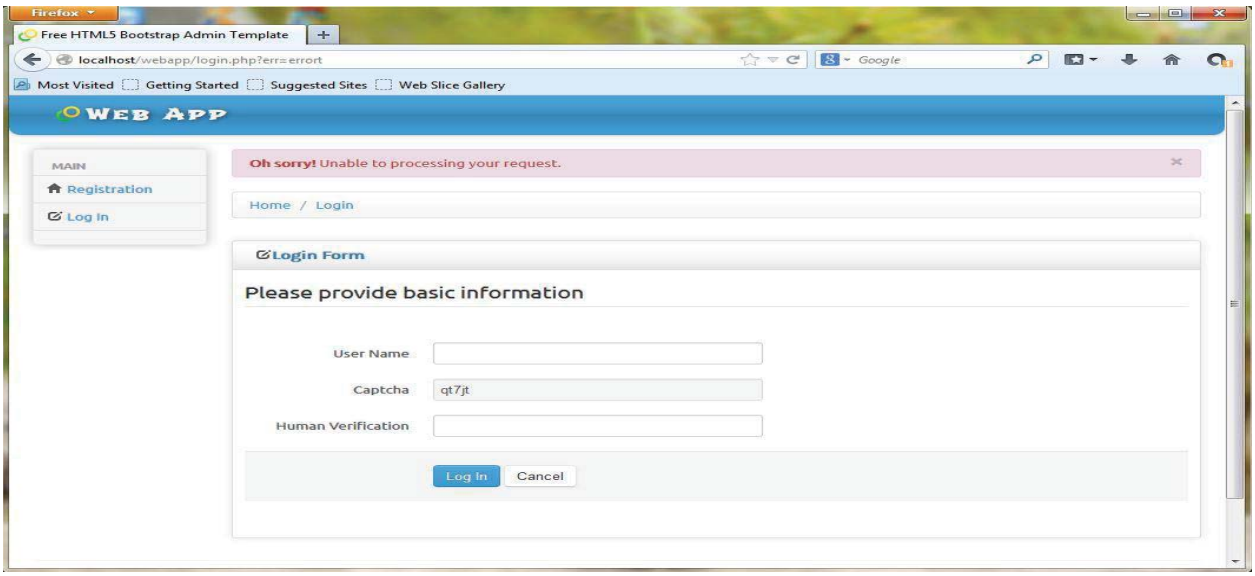


Fig: 15d Login process step 4 (second time)

In the figure 15d we can see that user cannot able to complete the login process successfully. He / she will be redirected to the login page again. But in this present stage if we look at the data base then we can see that the OTP value is modified with the new OTP value- *c40b0*, shown in figure 15e.

id	name	uname	otp	email	dob

Fig 15e Data base snap shot after unsuccessful login of user Bob Biswas

12	Bob Biswas	bob.biswas	c40b0	bob@rediffmail.com	06/11/1991

4.3 Steganography Algorithm:-

In this section I will show the output of steganography [7] algorithm which I have implemented in this paper.

- Embed data into picture:-

The user interface for embedding message is shown in figure 16. After typing the message and opening the target image for embedding the message, pressing the button named Embed writes the message into the image (figure 17). The resulting image containing the message is shown on the right side panel of the user interface (figure 18). Pressing Save button offers to save the image in either PNG format or BMP format, because these two formats offer loss-less compression (figure 19).

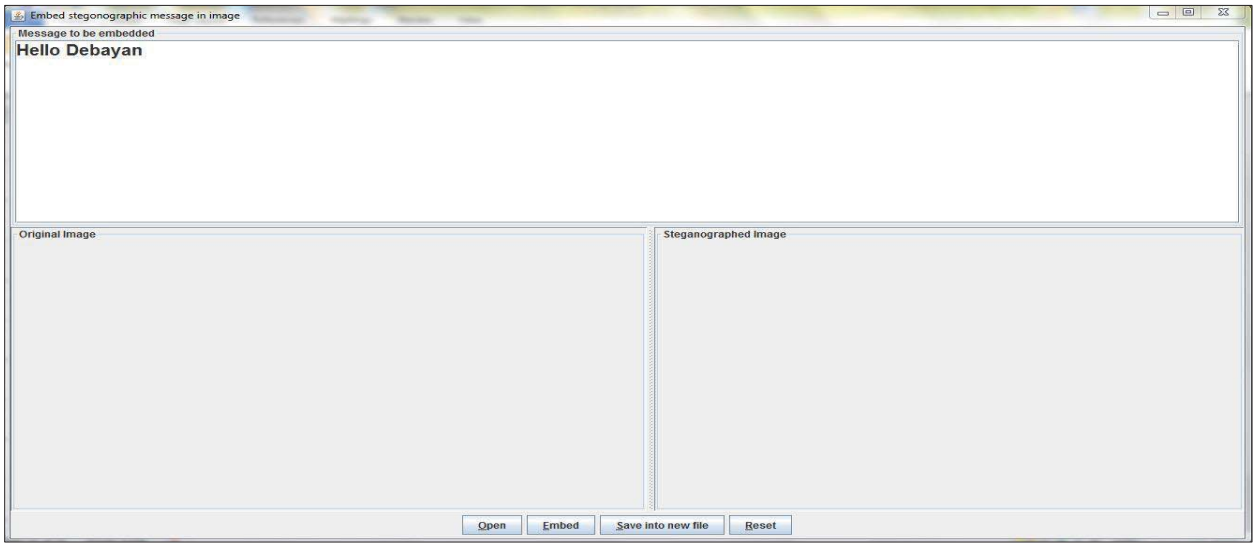


Fig: 16 Interface for Embedding Message

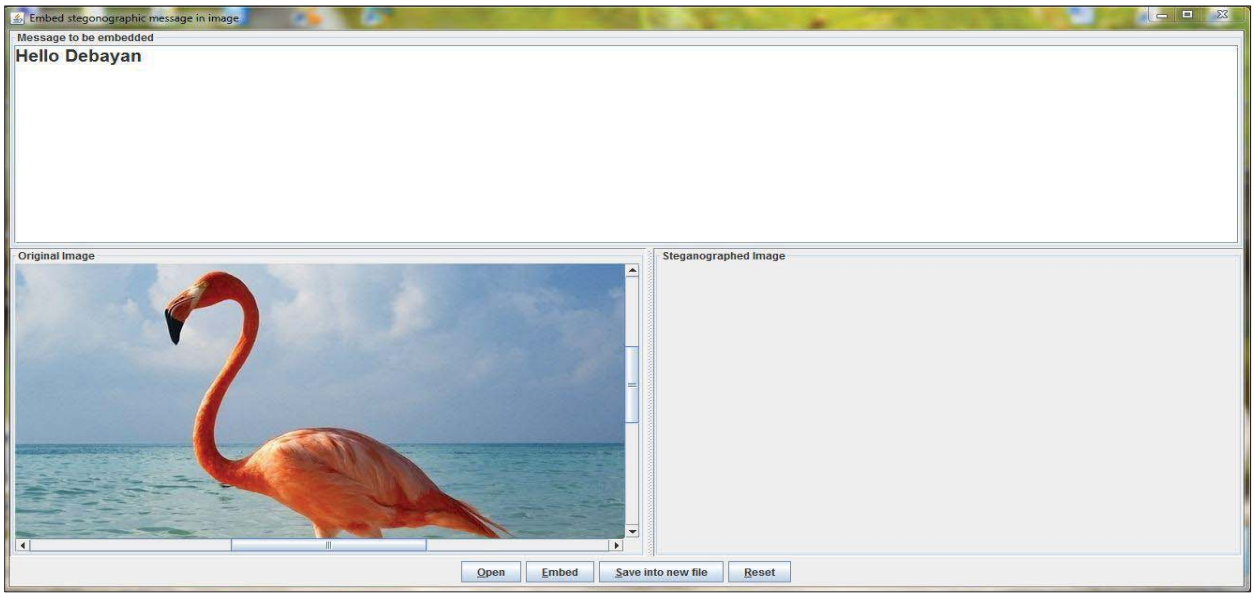
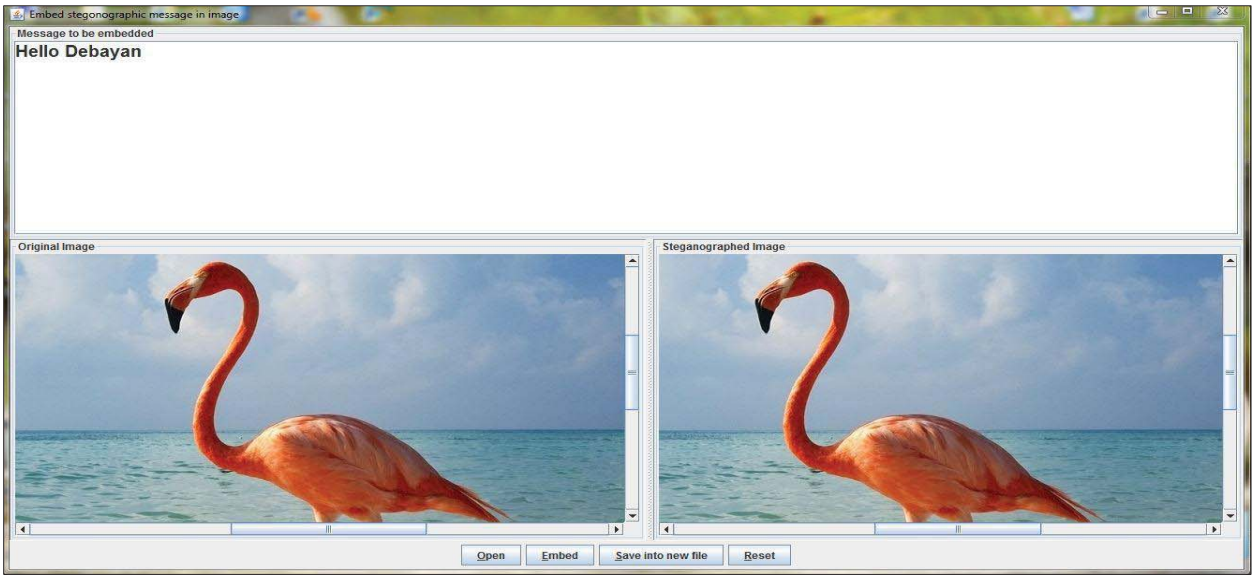


Fig: 17 Message and The Target Image (on the left side)
Fig: 18 Image Containing the Embedded Message (on the right side)



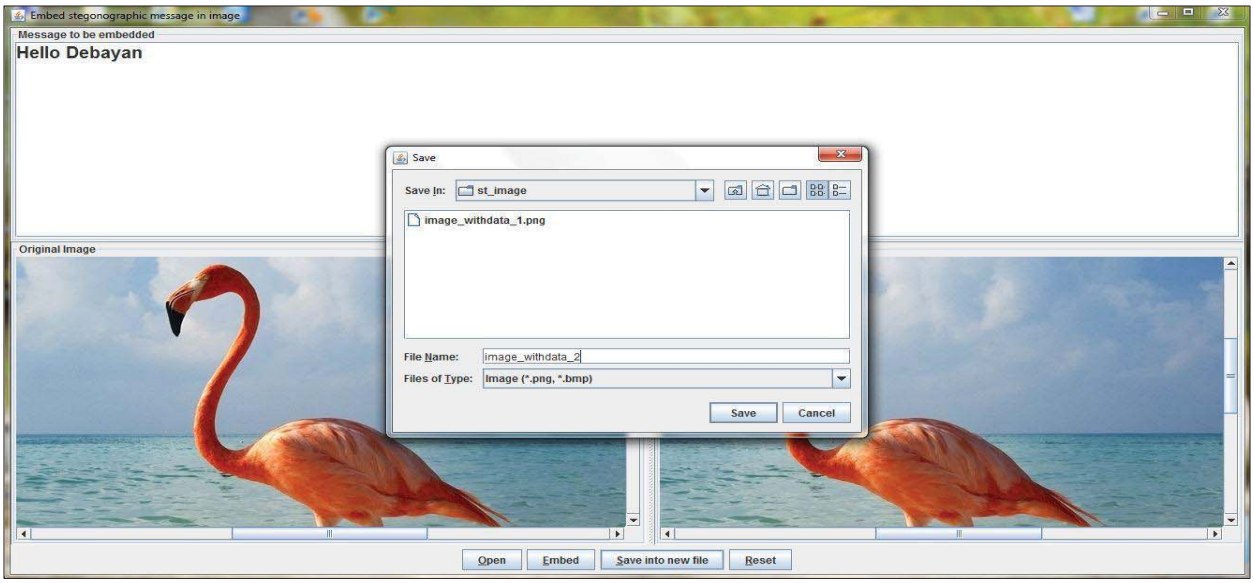


Fig: 19 Save Dialog for Image with Message (accepts only PNG & BMP formats)

After completion of embedding process if we closely look at the two pictures (the original picture named- original_1.jpeg and the steganographed picture named- image_withdata_1.png) we can see that there is no difference between this two pictures. So in naked eye we cannot make any distinguish between the original and the steganogrphed picture. These two pictures are show below (figure 20 to 21).

Fig. 20 original_1.jpeg

Fig. 21 image_withdata_1.png

- Decode data from the steganographed image:-

The user interface for extracting message from an image is shown in figure 22. The image previously saved with an embedded message is opened (figure 23 to 24). The message extracted from the image is shown in figure 25.

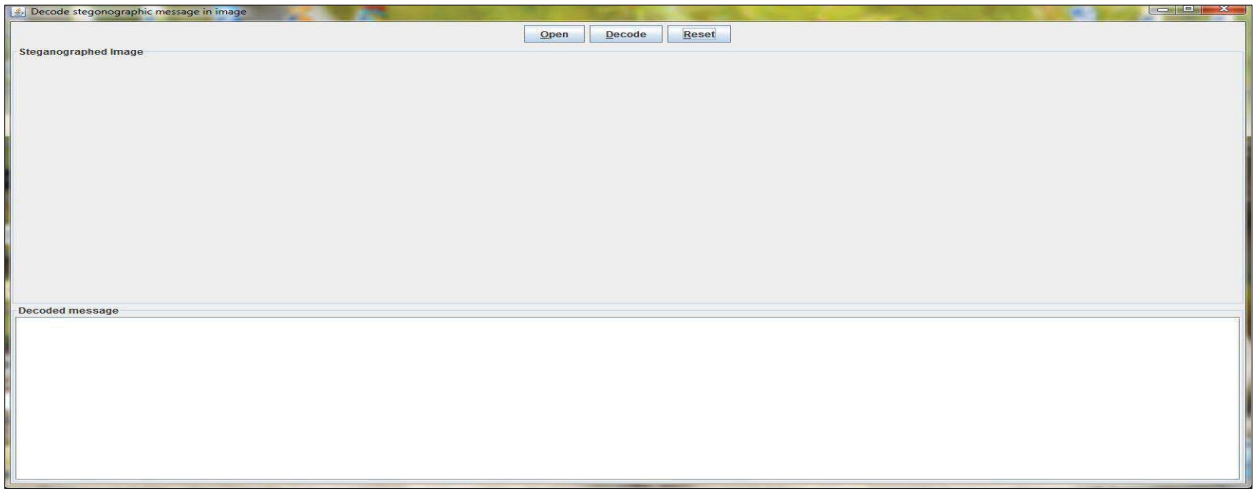


Fig: 22 User Interface for Extracting Message form Steganographed Picture

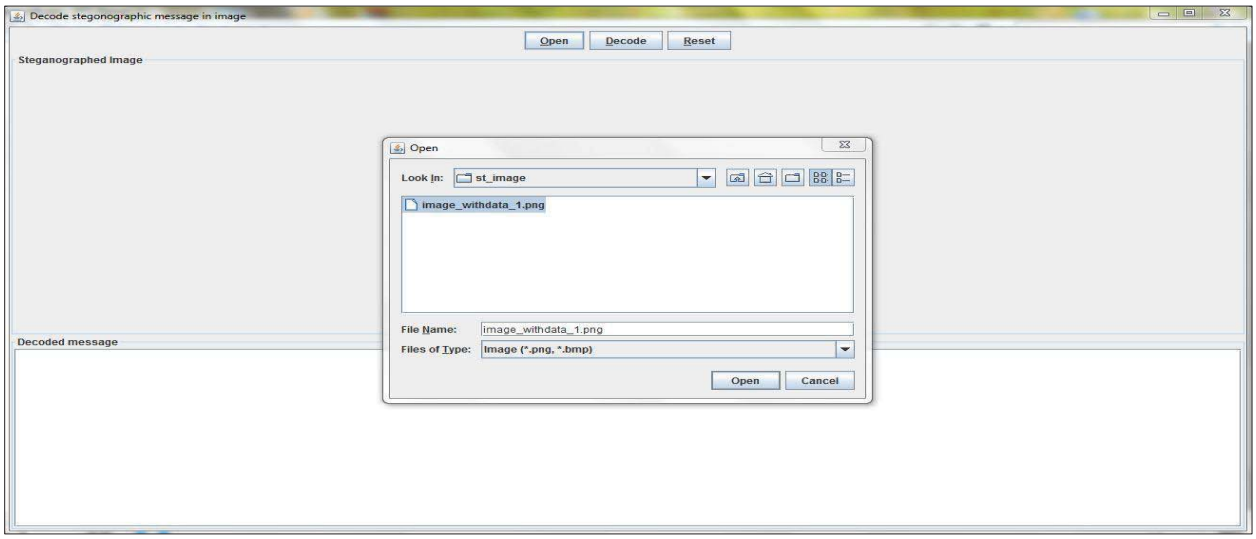


Fig: 23 Select the Picture by clicking the Open button

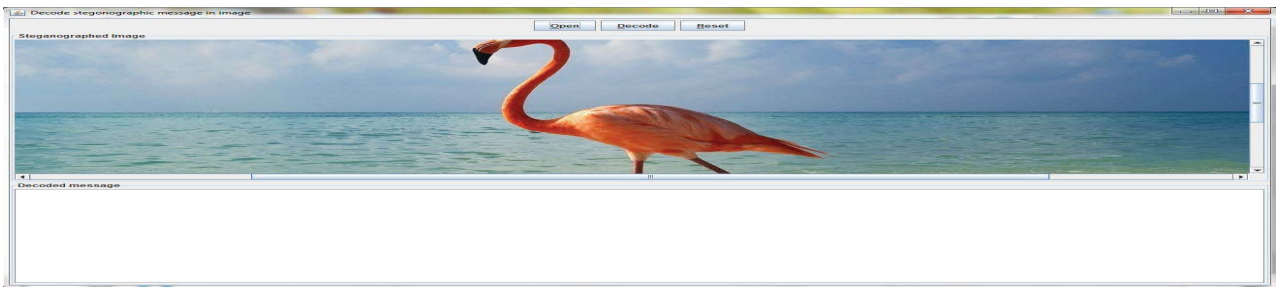


Fig: 24 Image Containing a Message (Ready for Extraction)

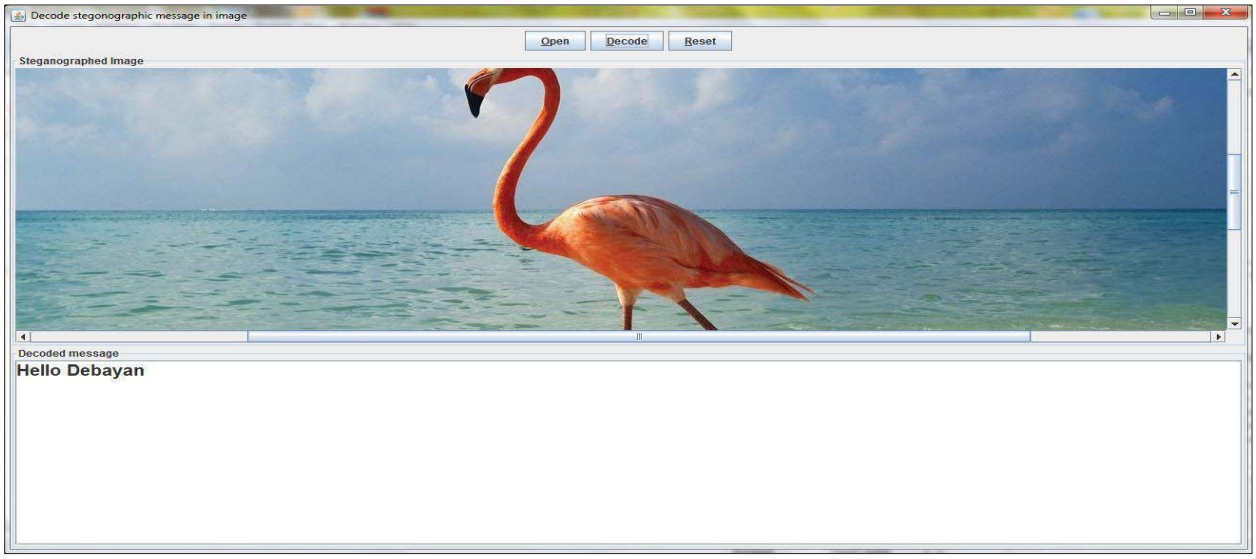


Fig: 25 Message Extracted from the Image

So in this way we can embed the OTP value into any image and send it to the user via email.

V. CONCLUSION

This study have focused on the current security situation in cloud computing. In a few cases static passwords have been used when logging in, and in other cases two-factor authentication with OTPs. In this paper I have proposed a ways to securely and easy login to a cloud service using OTPs without use of any type of an authentication device. Furthermore, a proposal for registration new users to the cloud service has been made, that is secure and easy to use. The best secured network connection (SSL [8]) is used in cloud services; it provides high security for the users while it is also easy to use. It provides benefits over the current security solutions for authentication that is used today. The big advantage from solutions with one time passwords is that the passwords in this solution are only valid for one time only, which gives great advantage in security. The whole solution is based on open source code; it has advantages over other cloud provider's. Since cloud services is used by millions of users, the security must be very good in order to protect private data, and also be fast, flexible and easy to use for all of the different users with different technology skills. With the authentication, registration and Steganography [7] method proposed and implemented in this thesis, all of those factors are accomplished.

Here I only implement my first proposal (for more details see chapter 3.1.1) and from second proposal (for more details see chapter 3.1.2) I only show the Steganography [7] algorithm but I cannot be able to embed the OTP into image. For future work I will try to embed the OTP which is generate in the cloud server into an image which I said into proposal 2 (for more details see chapter 3.1.2) and send this data embedded image to the user. I will also try to implement this logic into cloud system.

REFERENCES

[1] aws.amazon.com/mfa/ Accessed 26/09/2010
[2] B.Soh & A. Joy, ―A Novel Web Security Evaluation Model for a One – Time- Password System‖, WI. Proceedings. IEEE/WIC, 2003.
[3] D. Florencio and C. Herley, ―One-Time Password Access to Any Serverwithout Changing the Server‖, Microsoft Research, One Microsoft Way, Redmond, WA, T.-C. Wu et al. (Eds.): c_Springer-Verlag Berlin Heidelberg ISC 2008, LNCS 5222, pp. 401–420, 2008.
[4] http://en.wikipedia.org/wiki/CAPTCHA
[5] http://en.wikipedia.org/wiki/MD5
[6] http://en.wikipedia.org/wiki/RGBA_color_space
[7] http://en.wikipedia.org/wiki/Steganography
[8] http://en.wikipedia.org/wiki/Transport_Layer_Security
[9] http://googleenterprise.blogspot.com/2010/09/more-secure-cloud-for-millions-of.html Accessed 30/09/2010
[10] http://www.google.com/recaptcha/captcha
[11] I. Das, R. Das, ―Mobile Security (OTP) by Cloud Computing‖, ,IJIET, Vol. 2 (4), 2013.
[12] Markus Johnsson & A.S.M Faruque Azam ―Mobile One Time Passwords and RC4 Encryption for Cloud Computing‖, Technical Report IDE1108, School of Information Science, Computer and Electrical Engineering Halmstad University, 2011

[13] S.Zhang & S.Zhang & X.Chen, “Analysis and Research of Cloud Computing System Instance”, Future Networks. ICFN’10. Second, 2010.
[14] Wayne A. Jansen, “Cloud Hooks: Security and Privacy Issues in Cloud Computing”, Proceedings of the 44th Hawaii International Conference on System Sciences, NIST, 2011.
[15] Z. Shen & L. Li & F. Yan & X. Wu, ―Cloud Computing System Based on Trusted Computing Platform‖, Intelligent Computation Technology and Automation (ICICTA), International Conference: 11-12 May of 2010. page(s): 942, 2010.