International Journal of Computer Science and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 3, Issue 4, July 2014, 69-78
© IASET

LOCATION BASED APP MANAGEMENT WITH CAPTCHA SECURITY

VIJAYALAKSHMI S HALLIKERI[1], SUNAY A B[2], KOUSHIK N GODBOLE[3] & MAHESH M[4]

[1]Assistant Professor, Department of Computer Science & Engineering, Bapuji Institute of Engineering & Technology, Davangere, Karnataka, India

[2,3]B.E, Department of Computer Science & Engineering, Bapuji Institute of Engineering & Technology, Davangere, Karnataka, India

[4]Lecturer, Department of Computer Science & Engineering, S.J.M.I.T Chitradurga, Karnataka, India

ABSTRACT

This paper discusses the implementation of CAPTCHA on mobile device (client side) and app management based on location, in android based mobile devices. As a solution to security on smart phones based on Android OS platform, to stop automated data transfer between mobile device and web server, we implement generation and authentication of CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) on the android mobile itself. This app also shows how applications installed on mobile devices can be sorted for our use based on location, by integrating the GPS (Global Positioning System) in an application.

KEYWORDS: CAPTCHA, GPS, Location Based App Management

INTRODUCTION

There are two issues in this paper. One is CAPTCHA generation for security on mobile devices and the other is location based app management. So, we are going to discuss them separately.

Security on Mobile Devices

Security of information on mobile devices is of major concern in the present day situation. As such now there are no known methods to stop automated data transfer from mobile (client) to server. To address this issue we implement CAPTCHA on mobile.

A CAPTCHA is a kind of Turing test [1] the test is designed for humans; here computer will be the judge. With this test computer can decide whether the tested user is a human or a computer boot(simulated user).

Use of Internet is continuously increasing day by day, hackers are exploiting internet resources for their own purposes. For this it is necessary to distinguish between Valid Users and Invalid Computer bots. User Login and Password can allow only valid user and CAPTCHA can help computer to find whether user is a human or bot. [2]

Location Based Application Management

In present days, more and more users of mobile are switching from traditional basic handsets to smart phones. Android smart phone users have increased exponentially over the last few years. It has become a trend as well a necessity to have more and more applications to ease our daily work related to various fields.

As the number of applications in mobile is increasing daily, it very important to have some criteria to sort the applications based on time and location. So that if you are in a particular location like office, we can have only Department of CS&E those applications that we need in office, sorted for our use. Android mobile devices have built in GPS receiver. It can be used to find the location. Location based app management is one of the location based services. [4]

OVERVIEW OF CAPTCHA

A CAPTCHA is a program that can generate and grade tests that humans can pass but current computer programs cannot. Actually they are colourful images with distorted text. For example, humans can read distorted text as the one shown below, but current computer programs can't:

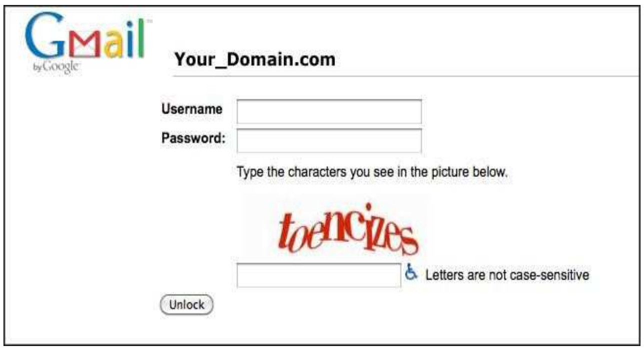

GMail by Google
Your_Domain.com
Username
Password:
Type the characters you see in the picture below.
toencizes
Letters are not case-sensitive
Unlock

Figure 1: How CAPTCHA Looks

They were found out in 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper and John Langford of Carnegie Mellon University. At the time, they developed the first CAPTCHA to be used by Yahoo. Even Hotmail, PayPal and many other popular Web sites use them to prevent automated registrations, and they work because no computer program concurrently read distorted text as well as humans. The construction of a CAPTCHA based on a text domain such as text understanding or generation is an important open problem for the project. Modern Text based CAPTCHAS are designed in such a way that they require the simultaneous use of three separate. Abilities, (1) invariant recognition, (2) segmentation, and (3) parsing to correctly complete the task with any consistency. Invariant recognition refers to the ability to recognize the large amount of variation in the shapes of letters. There are nearly an infinite number of versions for each character that a human brain can successfully identify. The same is not true for a computer and teaching it to recognize all those differing formations is an extremely challenging task. Segmentation or the ability to separate one letter from another is also made difficult in CAPTCHAs, as characters are crowded together with no white space in between. Lastly, context is also critical. A complete holistic view of the CAPTCHA must be taken to correctly identify each character. For example, in one segment of a CAPTCHA, a letter might look like “m” It is only when the whole word is taken into consideration that it becomes clear that it is actually a “u” and an “n.” Each of these problems pose a significant challenge for a computer even in isolation. The presence of all three at the same time is what makes CAPTCHAs so difficult to solve.

APPLICATION OF CAPTCHA

Preventing Comment Spam in Blogs: Most bloggers are familiar with programs that submit bogus comments, usually for the purpose of raising search engine ranks of some website (e.g., "buy penny stocks here").

This is called comment spam. By using a CAPTCHA, only humans can enter comments on a blog. There is no need to make users sign up before they enter a comment, and no legitimate comments are ever lost!

Protecting Website Registration: Several companies (Yahoo!, Microsoft, etc.) offer free email services. Up until a few years ago, most of these services suffered from a specific type of attack: "bots" that would sign up for thousands of email accounts every minute. The solution to this problem was to use CAPTCHAs to ensure that only humans obtain free accounts. In general, free services should be protected with a CAPTCHA in order to prevent abuse by automated programs.

Online Polls: In November 1999, http://www.slashdot.org released an online poll asking which the best graduate school in computer science was. As is the case with most online polls, IP addresses of voters were recorded in order to prevent single users from voting more than once. However, students at Carnegie Mellon found a way to stuff the ballots using programs that voted for CMU thousands of times. CMU's score started growing rapidly. The next day, students at MIT wrote their own program and the poll became a contest between voting "bots." MIT finished with 21,156 votes, Carnegie Mellon with 21,032 and every other school with less than 1,000. Can the result of any online poll be trusted? Not unless the poll ensures that only humans can vote.

Preventing Dictionary Attacks: CAPTCHAs can also be used to prevent dictionary attacks in password systems. The idea is simple: prevent a computer from being able to iterate through the entire space of passwords by requiring it to solve a CAPTCHA after a certain number of unsuccessful logins.

Search Engine Bots: It is sometimes desirable to keep web pages unindexed to prevent others from finding them easily. There is an html tag to prevent search engine bots from reading web pages. The tag, however, doesn't guarantee that bots won't read a web page; it only serves to say "no bots, please." Search engine bots, since they usually belong to large companies, respect web pages that don't want to allow them in. However, in order to truly guarantee that bots won't enter a web site, CAPTCHAs are needed.

Worms and Spam: CAPTCHAs also offer a plausible solution against email worms and spam: "I will only accept an email if I know there is a human behind the other computer." A few companies are already marketing this idea.[7]

OVERVIEW OF LOCATION BASED SERVICES

A Location Based Service (LBS) is a mobile application that is dependent on the location of a mobile device, like mobile phone. It is possible because of wireless communication and location positioning technologies.

LBS have two major actions, that is:

- Obtaining the location of user
- Utilizing this information to provide a service. Needed components for LBS are mobile devices, applications, communication

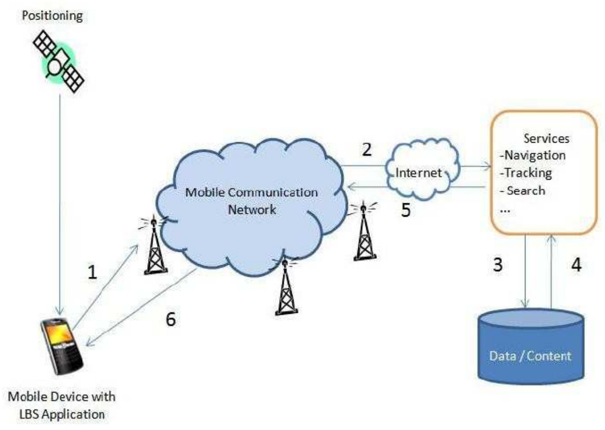


Figure 2: LBS Components and Services

Step 1: First, user sends a service request using the application running on mobile device.

Step 2: The service request, with user's current location information obtained from the positioning component (in this example, GPS data), is sent to service server via the mobile communication network.

Step 3, 4: The service server requests geographic database and other related database to get required information.

Step 5, 6: At last, the requested information is sent back to user's mobile via mobile communication network. [6]

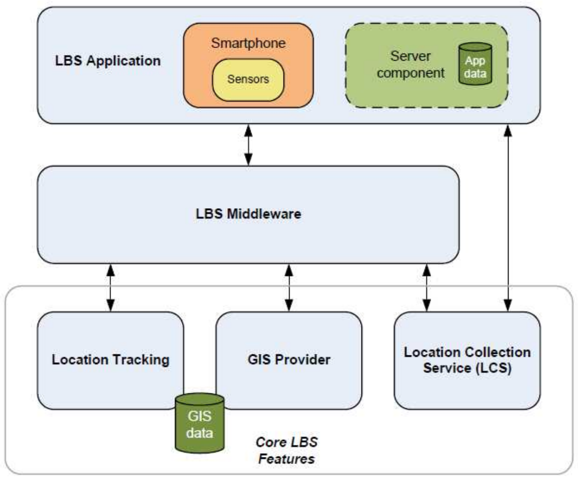


Figure 3: LBS Components

LBS Application is user application. Here it is our specific application that manages apps depending upon location of mobile device. Location tracking component stores the location trace of individual users. LBS Middleware interfaces LBS application with all its features. GIS (Geographic Information System) Provider provides map information, map visualization and directory services. Google Maps with its API can be considered a GIS provider. Location Collection Service component performs location collection to get a latitude and longitude for a specific user.

Android provides access to the above components to facilitate the implementation of LBS services through the help of following classes;

- Location Manager
- Location Provider
- Geocoding
- Google-Map

Location Manager manages all components of LBS system. Location Provider provides physical location. Geocoding provides a way to convert geographical coordinates (longitude, latitude) into street address and forward geocoding provides a mean to get geographical coordinated from street address. Google Map in Android provides a number of objects to handle maps in LBS system like Map View which displays the map. To handle this Map Activity class is there. Moreover, sufficient provisions are there to zoom the map, localize the map by means of Map Controller. Our application consumer service group of LBS applications. [6]

EXISTING SYSTEM

In the existing system CAPTCHA is generated on server side and then displayed to the user (client) to enter the text. The entered text is then sent from client to server where it is authenticated. This adds overhead for the server to process the information and the user has to wait until he is successfully authenticated, to proceed to next step.

Now a day‟s as the popularity of the android smart phones are increasing day-by-day there are millions of applications available depending upon the user requirement, but still now no application is present in android, which can sort the applications that are pre- installed in the mobile depending upon the particular location, as per the user needs to ease his work of app searching among many apps.

PROPOSED WORK

CAPTCHA is used to stop/control the automated data transfer between client and server. Generated CAPTCHA is on client side instead of server side, this reduces the overhead of the server by eliminating the authentication process performed by the server. The time required for the authentication of the client is reduced by generating CAPTCHA on client side thus the server can perform other useful task. By implementing CAPTCHA on mobile we can always guarantee to the server that the communicating process on the client side is human controlled.

As the number of applications in mobile is increasing daily, it is very important to have some criteria to sort the applications based on time and location. To address this issue in this application we allow the user to create a menu that contains list of application‟s that he requires at a particular location, so that when he is in that location he can use monitor option to see the apps that are important for him in that location which he would have specified earlier. To manage the menu list we provide delete menu option.

IMPLEMENTATION AND METHODOLOGY

Algorithm: CAPTCHA Generation and Verification

Inputs: Text entered by user in text box.

Output: Entered text authentication result (Success or Failure).

Step 1:-Create a shader that draws a linear gradient along a line, using Linear Gradient class.

Step 2: Create an object of Paint class and use its, set Shader()‟ method with the object created in Step 1 as one of its parameters. This paints the shader to a specified layout.

Step 3: Create a mutable bitmap and associate with a Canvas class object.

Step 4: Use the, draw Rect( ) ‟method of the Canvas object created in Step 3, and pass it the paint object created in Step 2 as one of its parameters. This displays a rectangle on the screen filled with paint object.

Step 5: For a specified length of CAPTCHA text, randomly generate either lower case alphabet or upper case alphabet or digit and store them in a variable, displayed Captha”.

Step 6: Create a new paint object to draw the text in, displayed CAPTCHA ‟on the screen and use its, set Text Skew( )” and„ set Color( ) ‟to rotate and color the text in„ displayed CAPTCHA” variable.

Step 7 : Create a new Canvas object with paint object, object created in Step 6 as one its parameter and use Canvas object‟s„ draw Text( ) ‟method to draw the characters on the display screen.

Pseudocode for Sorting Applications Based on Location

Create Menu

```
CreateMenu ( ) {
    If (GPS enabled and GPS location is fetched)   { Enter the menu name;
    Select the apps for present location, from app list;
    Get location details like Longitude, Latitude and Altitude from GPS;
    Store app list and location details in database;
    }
    else {
        Display error to the user with suitable message;
    }
}
```

Monitor Apps

```
MonitorApps ( ) {
    Fetch GPS location;
    If (GPS location Fetched) {
        Search database for app list with location details same as fetched GPS location;
        If (app list for fetched GPS location present) {
            Display apps in an
                app list for the user;
} else {
}
}
Display as „apps not present for present location‟;
}
```

Delete Menu

```
DeleteMenu ( ) {
Display name of app lists‟ present in the database to the user;
When user selects a list name to delete { Delete the app list form the database.
}
}
```

APPLICATION’S WORKING

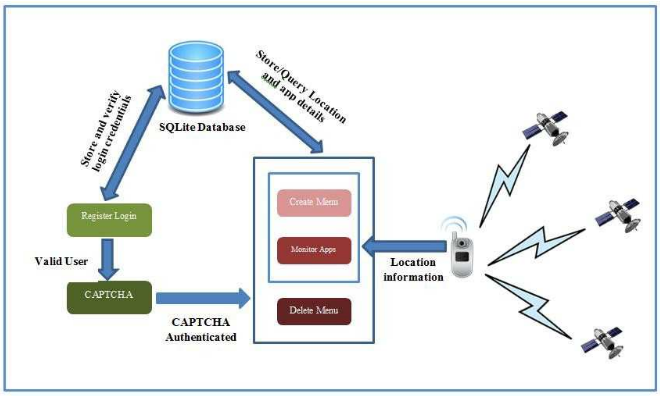


Figure 4: Architecture Diagram

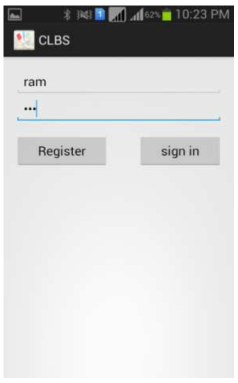
- After launching the application user will get a Login Menu.
- Once entering the valid username and password the user is registered.
- If he is already an existing user he can directly click on Login button to go to CAPTCHA menu.
- After successful login, CAPTCHA is generated for the validation of the user, to check whether user is a human or computer bot.
- Once the user enters the valid CAPTCHA he will get main menu page.
- Delete menu allows the user to delete the early created menu.
- In main Menu the user will get three choice Monitor, Create and Delete.
- Create menu option allows the user to create new menu with list of apps as per his requirement at that particular location.

- Monitor menu displays all the applications selected for that particular location.

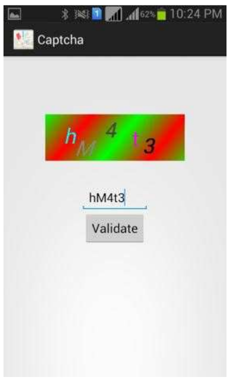
RESULTS

Screen shots taken while running the application developed.

Screen Shots of CAPTCHA Generation and Authentication



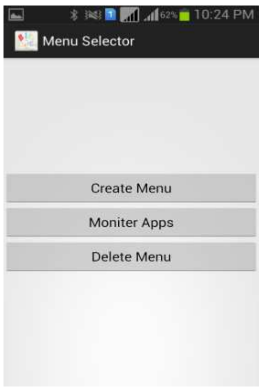
ram
•••

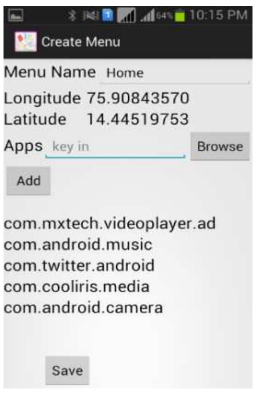


hM4t3

Figure 5(a): Login Page Figure 5(b): CAPTCHA

Screen Shots for Sorting of Apps Depending on Location

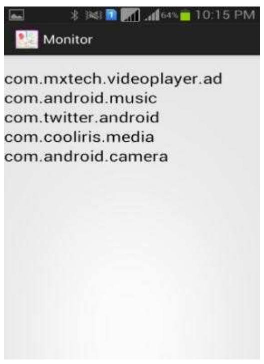




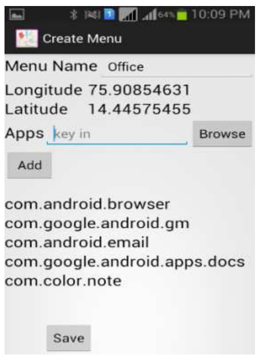
Menu Name Home
Longitude 75.90843570
Latitude 14.44519753
Apps key in

com.mxtech.videoplayer.ad
com.android.music
com.twitter.android
com.cooliris.media
com.android.camera

Figure 6(a): Main Menu Figure 6(b): Menu Created at Home



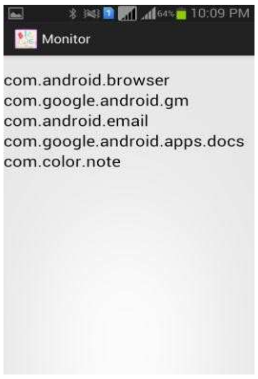
com.mxtech.videoplayer.ad
com.android.music
com.twitter.android
com.cooliris.media
com.android.camera



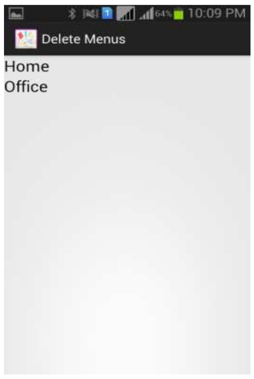
Menu Name Office
Longitude 75.90854631
Latitude 14.44575455
Apps key in

com.android.browser
com.google.android.gm
com.android.email
com.google.android.apps.docs
com.color.note

Figure 7(a): Monitor Result Figure 7(b): Menu Created at Home at Office


Monitor

com.android.browser
com.google.android.gm
com.android.email
com.google.android.apps.docs
com.color.note


Delete Menus

Home
Office

Figure 8(a): Monitor Result **Figure 8(b): Delete Menu at Office**

CONCLUSIONS

CAPTCHA can enhance the security in mobile devices. It can protect mobile devices from hacking by invalid users or by bots. But many of the mobile application services (e.g. like retrieving GPS location of user, phone call details, contact and email information, sending messages etc.) do not even need any user credentials for automated data transfer, if once installed.

The present CAPTCHA module developed is not actually stopping automated data transfer, it is helping the application developed to know whether the user is Human or Bot. To avoid automated data transfer, CAPTCHA module must be implemented and called at Kernel level.

Location based app management made possible by integrating GPS with the application can help the user to safeguard his profession or privacy related apps more efficiently by common user.

REFERENCES

1. Lin, R., Huang, S-Y, Bell, G.B. and Lee, Y-K (2011) a new CAPTCHA interface design for mobile devices. In: 12th Australasian User Interface Conference, AUIC 2011, 17 – 20 January, Perth, Western Australia.
2. Telling Humans and computers apart by Luis von Ahn, Manuel Blum, and John Langford, February 2004/Vol. 47, No. 2 COMMUNICATIONS OF THE ACM
3. Re: CAPTCHAs – Understanding CAPTCHA- Solving Services in an Economic Context by Marti Motoyama, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M. Voelker and Stefan Savage University of California, San Diego AND RELIABILITY SOCIETIES
4. Location-Based Services: Time for a Privacy Check-In, A publication of the ACLU of Northern California Available online at www.DotRights.org
5. Google Android-A Coprehensive Security Assessment1540-7993/10/$26.00 © 2010 IEEE, MARCH/APRIL 2010 COPUBLISHED BY THE IEEE COMPUTER
6. Location Based Services using Android Mobile Operating System International Journal of Advances in Engineering & Technology, Mar 2011, ISSN: 2231-1963 by Amit Kushwaha, VineetKushwaha
7. www.CAPTCHA.net/

8. http://researchrepository.murdoch.edu.au/13406
9. http://www.oracle.com/technetwork/java/javase /downloads /index.html [10] http://www. eclipse.org/
10. http://developer.android.com /sdk/index.html
11. http://developer.android.com/tools/adk/eclipse- adt.html
12. http://developer.android.com/tools/sdk/ndk/index. html