Accessing Multiple Social Networks Android Application

R.Sriramkumar
Kings College of Engineering, Pudukkottai, Tamil Nadu, India.

J.Jegan
Kings College of Engineering, Pudukkottai, Tamil Nadu, India.

D.Sivakumar
Kings College of Engineering, Pudukkottai, Tamil Nadu, India.

Abstarct - In this android application is based on application manager to manage the sensitive private application like (Facebook, whatsapp, twitter, etc.). This application is a collection of applications for manage the security and privacy of that sensitive application. This application has one password to sign all the application, after sign out from this application then other signed application automatically logout from the network. We use touch screen based password to access this application. We can provide the authenticated person to identify the person if any one try to unlock application in the smart phone. We add if any one try wrong pattern then our app capture that third party image and store it in the SD card. In this android application is developed for applications security for authenticated user mobile. However this process is not give full security to the user. It provide only proper authentication but not detect the unauthorized user. The android application is to manage the other sensitive application but only it lock and unlock the application. This application show in menu of the android mobile, but it lock the application if any one open the application get master password or pin to unlock. They used the password, pin and pattern for unlock authentication.

KEYWORDS: Smart Phone, Pin, Pattern

1. INTRODUCTION

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touch screen mobile devices such as smart phones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touch screen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface.

Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics. Android has the largest installed base of all operating systems of any kind. Android has been the bestselling OS on tablets since 2013, and on smart phones it is dominant by any metric. Initially developed by Android, Inc., which Google bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. Google releases the Nexus phones and tablets to act as their flagship Android devices, demonstrating Android's latest software and hardware features. From 2013 until 2015, Google offered several Google Play Edition devices over Google Play. While not carrying the Google Nexus branding, these were Google-customized Android phones and tablets that also ran the latest version of Android, free from manufacturer or carrier modifications. From 2010 to 2013, Hugo Barra served as product spokesperson, representing Android at press conferences and Google I/O, Google‟s annual developer-focused conference. Barra's product involvement included the entire Android ecosystem of software and hardware, including Honeycomb, Ice Cream Sandwich, Jelly Bean and Kit Kat operating system launches, the Nexus 4 and Nexus 5 smart phones, the Nexus 7 and Nexus 10 tablets, and other related products such as Google Now and Google Voice Search, Google‟s speech recognition product comparable to Apple‟s Siri. In 2013, Barra left the Android team for Chinese smart phone maker Xiaomi. The same year, 5

Larry Page announced in a blog post that Andy Rubin had moved from the Android division to take on new projects at Google. He was replaced by Sundar Pichai who became the new head of Android and Chrome OS, and, later, by Hiroshi Lockheimer when Pichai became CEO of Google. In 2014, Google launched Android One, a line of smart phones mainly targeting customers in the developing world. In May 2015, Google announced Project Brillo as a cut-down version of Android that uses its lower levels (excluding the user interface), intended for the "Internet of Things" (IoT) embedded systems

2. TOUCH BASED PASSWORD

Android's default user interface is mainly based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a virtual keyboard. Game controllers and full-size physical keyboards are supported via Bluetooth or USB. The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user. Internal hardware, such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a racing game by rotating the device, simulating control of a steering wheel.

Android devices boot to the home screen, the primary navigation and information "hub" on Android devices that is analogous to the desktop found on 6 personal computers. (Android also runs on regular personal computers, as described below). Android home screens are typically made up of app icons and widgets; app icons launch the associated app, whereas widgets display live, auto-updating content, such as the weather forecast, the user's email inbox, or a news ticker directly on the home screen. A home screen may be made up of several pages, between which the user can swipe back and forth, though Android's home screen interface is heavily customizable, allowing users to adjust the look and feel of the devices to their tastes. Third-party apps available on Google Play and other app stores can extensively re-theme the home screen, and even mimic the look of other operating systems, such as Windows Phone. Most manufacturers, and some wireless carriers, customize the look and feel of their Android devices to differentiate themselves from their competitors.

Applications that handle interactions with the home screen are called "launchers" because they, among other purposes, launch the applications installed on a device. Along the top of the screen is a status bar, showing information about the device and its connectivity. This status bar can be "pulled" down to reveal a notification screen where apps display important information or updates, such as a newly received email or SMS text, in a way that does not immediately interrupt or inconvenience the user. Notifications are persistent until read by tapping it, which opens the relevant app, or dismissed by sliding it off the screen. Beginning on Android 4.1, "expanded notifications" can display expanded details or additional functionality; for instance, a music player can display playback controls, and a "missed call" notification provides buttons for calling back or sending the caller an SMS message. Android provides the ability to run applications that change the default launcher, and hence the appearance and externally visible behavior of 7 Android. These appearance changes include a multi-page dock or no dock, and many more changes to fundamental features of the user interface.

3. SDK

Applications ("apps"), which extend the functionality of devices, are written using the Android software development kit (SDK) and, often, the Java programming language that has complete access to the Android APIs. Java may be combined with C/C++, together with a choice of non-default runtimes that allow better C++ support; the Go programming language is also supported since its version 1.4, which can also be used exclusively although with a restricted set of Android APIs. The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Initially,

Google's supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) plugin; in December 2014, Google released Android Studio, based on IntelliJ IDEA, as its primary IDE for Android application development. Other development tools are available, including a native development kit (NDK) for applications or extensions in C or C++, Google App Inventor, a visual environment for novice programmers, and various cross platform mobile web applications frameworks. In January 2014, Google unveiled a framework based on Apache Cordova for porting Chrome HTML 5 web applications to Android, wrapped in a native application shell. Android has a growing selection of third-party applications, which can be acquired by users by downloading and installing the application's APK (Android application 8 package) file, or by downloading them using an application store program that allows users to install, update, and remove applications from their devices.

Google Play Store is the primary application store installed on Android devices that comply with Google's compatibility requirements and license the Google Mobile Services software. Google Play Store allows users to browse, download and update applications published by Google and third-party developers; as of July 2013, there are more than one million applications available for Android in Play Store.

As of July 2013, 50 billion applications have been installed. Some carriers offer direct carrier billing for Google Play application purchases, where the cost of the application is added to the user's monthly bill. Due to the open nature of Android, a number of third-party application marketplaces also exist for Android, either to provide a substitute for devices that are not allowed to ship with Google Play Store, provide applications that cannot be offered on Google Play Store due to policy violations, or for other reasons.

Since Android devices are usually battery-powered, Android is designed to manage processes to keep power consumption at a

minimum. When an application is not in use the system suspends its operation so that, while available for immediate use rather than closed, it does not use battery power or CPU resources. Android manages the applications stored in memory automatically: when memory is low, the system will begin invisibly and automatically closing inactive processes, starting with those that have been inactive for longest. Life hacker reported in 2011 that third-party task killers were doing more harm than good.

4. ANDROID APPLICATIONS

Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as Google Play, SlideME, Opera Mobile Store, Mobango, F-droid and the Amazon Appstore.

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

4.1 Android - Architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.

4.2 Linux kernel

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

5. SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs)

Fig. 1 Android Architecture

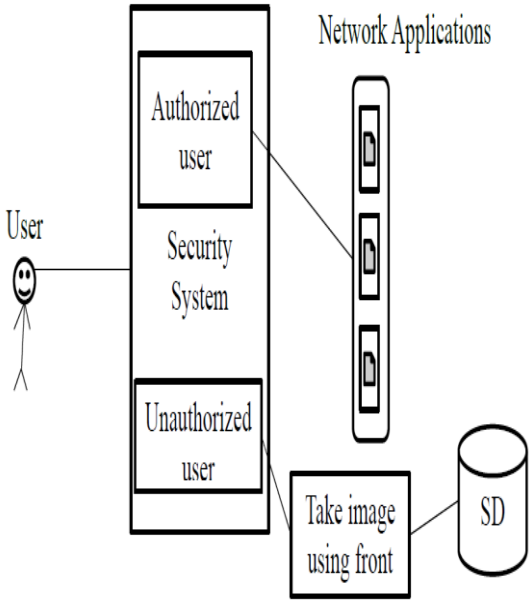

Fig 2 System Architecture diagram

6. SQLITE

SQLite is a relational database management system contained in a C programming library. In contrast to many other database

management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program. SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity. SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. SQLite has bindings to many programming languages.

Unlike client–server database management systems, the SQLite engine has no standalone processes with which the application program communicates. Instead, the SQLite library is linked in and thus becomes an integral part of the application program. The library can also be called dynamically. The application program uses SQLite's functionality through simple function calls, which reduce latency in database access: function calls within a single process are more efficient than inter-process communication. SQLite stores the entire database (definitions, tables, indices, and the data itself) as a single cross-platform file on a host machine. It implements this simple design by locking the entire database file during writing. SQLite read operations can be multitasked, though writes can only be performed sequentially. Due to the server-less design, SQLite applications require fewer configurations than client-server databases. SQLite is called zero-conf because it 27 does not require service management (such as startup scripts) or access control based on GRANT and passwords. Access control is handled by means of File system permissions given to the database file itself.

Databases in client-server systems use file system permissions which give access to the database files only to the daemon process. Another implication of the server less design is that several processes may need to be able to write to the database file. In server-based databases, several writers will all connect to the same daemon, which is able to handle its locks internally. SQLite on the other hand has to rely on file-system locks. It has less knowledge of the other processes that are accessing the database at the same time. Therefore, SQLite is not the preferred choice for write-intensive deployments.

SQLite began as a Tcl extension. Hipp based the syntax and semantics on those of PostgreSQL 6.5. In August 2000, version 1.0 of SQLite was released, with storage based on gdbm (GNU Database Manager). SQLite 2.0 replaced gdbm with a custom B-tree implementation, adding transaction capability. SQLite 3.0, partially funded by America Online, added internationalization, manifest typing, and other major improvements. In 2011 Hipp announced his plans to add an UnQL interface to SQLite databases and to develop UnQLite, an embeddable document-oriented database.

7. MODULES

7.1 Register Touch co-ordinates

1. Use Motion event class to get touch coordinates.

2. Get users touch coordinates.

3. Use android SQLite database to store touch co ordinates

7.2 Touch Based Authentication

1. Retrieve values get pixel coordinates from user login.

2. Match coordinates with stored coordinates.

3. If match sign in to the application menu.

7.3 Application List

1. Create List view control in menu

2. Use application manger to get all installed app

3. Arrange in the list view with its logo.

7.4 Sign In and Sign out

1. Sign in to the application manager and sign in to social network application.

2. If exit from application manager then sign out from all the application.

3. Use android SQLite database to store the user log.

8. GENETIC ALGORITHM

Genetic algorithm (GA) is a kind of evolutionary technique that emulates biological theories that are useful in solving optimization problems. According to Darwin‟s survival of the fittest evolutionary theory, only the most potential elements in a population are likely to survive generate offspring. The operation of GA begins with a population of random strings the design variable. Each string is evaluated to find the fitness function. The three main GA operators - reproduction, crossover and mutation are applied on the random population to create new population. The population is evaluated and tested until the termination criterion is met, iteratively altered by the GA operators.

Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators. Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection. The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation.

9. SIMULATION RESULTS

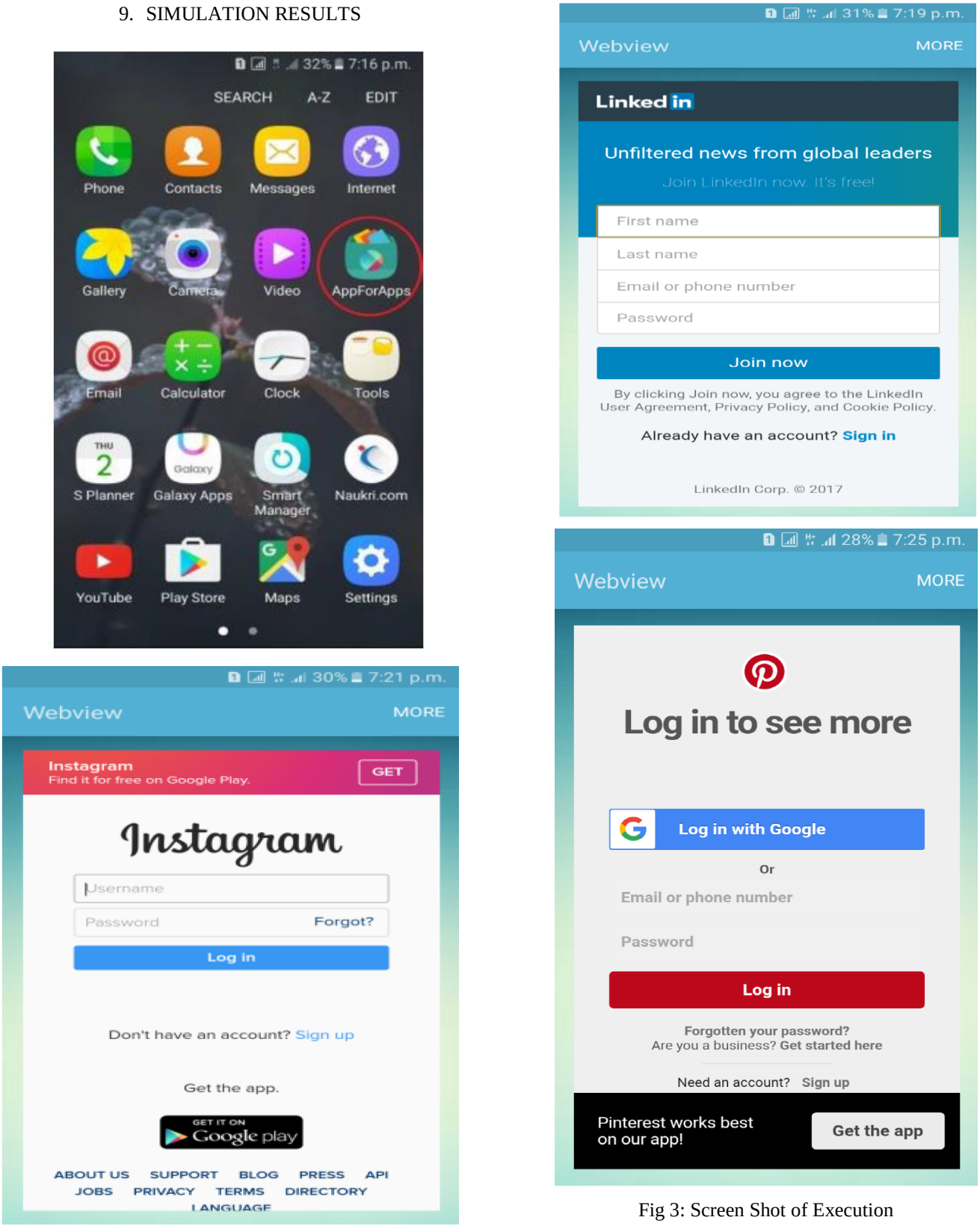

Fig 3: Screen Shot of Execution

10. CONCLUSION

In this system achieves user friendly environment to access social network application in mobile. Also maintain privacy to protect the application. It provides perfect security to access social network in android platform. Also it provides easy access to multiple applications simultaneously. We can provide the authenticated person to identify the person if any one try to unlock application in the smart phone. We add if any one try wrong pattern then our app capture that third party image and store it in the SD card. In this android application is developed for applications security for authenticated user mobile. However this process is not give full security to the user. It provide only proper authentication but not detect the unauthorized user. This system concentrate on fully maintain the security of the applications. Authentication of the application is more secure than other type of authentication. Use Motion event class to get touch coordinates and get users touch coordinates. This system manages the authentication of the other applications and it hides from the menu of the android mobile. The new password system use to authenticate that touch screen based password authentication. Retrieve values get pixel coordinates from user login and match coordinates with stored coordinates using genetic algorithm. we have fully maintained the other sensitive application with privacy. The proposed android application is collection of sensitive application with its access rights.

REFERENCES

[1] M. Motoyama and G. Varghese, “I seek you: searching and matching individuals in social networks," in Proceedings of the eleventh international workshop on Web information and data management, ser. WIDM, 2009.

[2] C. Grier, K. Thomas, V. Paxson, and M. Zhang, “spam: the underground on 140 characters or less," in Proceedings of the ACM conference on Computer and communications security, ser. CCS, 2010.

[3] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, “Detecting spammers on Twitter," in Proceedings of the Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS), 2010.

[4] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, “Who is tweeting on Twitter: human, bot, or cyborg?" in Proceedings of the 26th Annual Computer Security Applications Conference, ser. ACSAC, 2010.

[5] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Gon_calves, “Detecting spammers and content promoters in online video social networks," in Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ser. SIGIR, 2009.

[6] A. Charland and B. Loux, “Mobile application development: web vs. native,” Common. ACM, vol. 54, no. 5, pp. 49-53, May 2011.

[7] R. Padley, “HTML5-bridging the mobile platform gap: mobile technologies in scholarly communication," Serials, vol. 24, pp. 3239, 2011.

[8] Joe, M. Milton, and Dr B. Ramakrishnan. "A survey of various security issues in online social networks." *International Journal of Computer Networks and Applications* 1.1 (2014): 11-14.

[9] Joe, M. Milton, B. Ramakrishnan, and R. S. Shaji. "Prevention of losing user account by enhancing security module: A facebook case." *journal of emerging technologies in web intelligence* 5.3 (2013): 247-256.