New Orleans JAZZED about Engineering Education

Paper ID #14386

Integrating Security Education into a CS Curriculum - Practices and Experience

Prof. Yi Pan, Georgia State University

Dr. Yi Pan is a Distinguished University Professor of the Department of Computer Science and Associate Dean of Arts and Sciences at Georgia State University. He is also a visiting Changjiang Chair Professor at Central South University in China. Dr. Pan received his B.Eng. and M.Eng. degrees in computer engineering from Tsinghua University, China, in 1982 and 1984, respectively, and his Ph.D. degree in computer science from the University of Pittsburgh, USA, in 1991.

Dr. Pan's research interests include parallel and cloud computing, wireless networks, and bioinformatics. Dr. Pan has published more than 330 papers including over 150 SCI journal papers and 50 IEEE Transactions papers. In addition, he has edited/authored 40 books. He has received many awards from organizations such as IEEE, NSF, AFOSR, JSPS, IBM, ISIBM, IISF and Mellon Foundation. Dr. Pan has served as an editor-in-chief or editorial board member for 15 journals including 7 IEEE Transactions and a guest editor for 12 special issues for 10 journals including 2 IEEE Transactions. He has organized numerous international conferences and workshops and has delivered over 40 keynote speeches at international conferences around the world.

Dr. Pan is a "Great Master Face-to-Face" Series Speaker (2012), an IEEE Distinguished Speaker (2000-2002), a Yamacraw Distinguished Speaker (2002), a Shell Oil Colloquium Speaker (2002), and a senior member of IEEE. He is listed in Men of Achievement, Who's Who in Midwest, Who's Who in America, Who's Who in American Education, Who's Who in Computational Science and Engineering, and Who's Who of Asian Americans.

Dr. Michael Weeks, Georgia State University

Michael Weeks is an associate professor of computer science at Georgia State University. His areas of interest include digital signal processing, embedded systems, and computer science education. He has a Ph.D. in computer engineering.

Prof. Yanqing Zhang, Georgia State University

A full Professor of the Computer Science Department at Georgia State University, Atlanta, USA. He received the Ph.D. degree in computer science from the University of South Florida in 1997. His research interests include hybrid intelligent systems, computational intelligence, machine learning, data mining, bioinformatics, brain informatics, health informatics, computational web intelligence, green computing, granular computing, Yin-Yang computation, nature-inspired computing, security, cloud computing. He is a member of the Bioinformatics and Bioengineering Technical Committee of the IEEE Computational Intelligence Society. He received Outstanding Academic Service Award at IEEE 7th International Conference on Bioinformatics & Bioengineering (IEEE BIBE 2007), Achievement Award of the 2007 World Congress in Computer Science, Computer Engineering and Applied Computing, and 2005 IEEE-Granular Computing Outstanding Service Award at 2005 IEEE International Conference on Granular Computing.

Integrating Security Education into a CS Curriculum - Practices and Experience[1]

Zhongli Ding, Michael Weeks, Yanqing Zhang, and Yi Pan
Department of Computer Science
Georgia State University
Atlanta, Georgia 30302

Abstract

Cybersecurity is important for many applications in both research and education. Currently, graduates in cybersecurity are in short supply because few universities have sufficient courses in this area. An interesting and practical hands-on labware can help students obtain knowledge in security. We have been working on integrating security education into Georgia State University's computer science curriculum since the project was funded by NSF in Sept. 2013. We focus on adding security teaching activities into four courses: (1) Operating Systems, (2) Embedded Systems, (3) Computer Networks, and (4) Web Programming. This project aims to teach selected mobile security topics in computer science courses based on several labwares. So far, we have designed an Android application software system for a student to learn and understand mobile security. The new client-server mobile security system was developed to identify a person's voice, and store the spoken password in a cloud server, disallowing another user or malware to access the device. Upon unauthorized access, an email will be received very quickly through Java Mail service. The user interface has to be popped out in order to take a picture. Based on this open source platform, students can use their creative ideas to implement their own system or improve it. Based on pre-evaluations and post-evaluations on the four courses, we received positive feedback from students. Almost all the students have their own mobile devices and feel comfortable working with them. The newly developed mobile security system on Android and Java made the students easily learn how mobile security systems work via several hands-on exercises. In the Operating Systems class, many students also did research projects related to mobile security, and gave class presentations to share their research results in the class. Most students agree that the labware on mobile security can help them learn faster and better. In the future, we plan to design more security hands-on labware to teach security and introduce them into more courses in our curriculum.

Keywords: cloud server; Android; voice recognition; location; Java Mail; distributed system; mobile security system; labware; evaluation

I. INTRODUCTION

Mobile devices bring convenience to the daily lives of people, and make many tasks more efficient. However, there are still many potential issues to be solved urgently, especially in the area of mobile security. In this paper, we discuss and analyze the background of mobile security in order to build a architecture to solve security problems such as the unsafe practice of storing passwords on the device itself. As a result, the proposed application, Mobile Security System, would be available to install on Android devices to enhance the protection of owners' data. We can also allow the owner of a stolen device to get the device back. Furthermore, mobile security is a relatively weak area in most schools' computing curriculum [5]. For the purpose of promoting the interests and knowledge of students about mobile security, we include the manuals of the application, both for developers and users, which would certainly be helpful when students try to use and develop new features on the application. Finally, we list future improvements of the application, to further protect the data and more easily get a stolen device back. The labware has been used in several classes at GSU and the feedbacks from the students demonstrate that the labware is very useful for the students to get deep insights into the mobile security problem.

[1] This research has been supported by NSF under grants 1244665 and 1303359.

The owner of a lost mobile device often feels panicked, because of all the important documents and private information stored inside. In this paper, we propose a way to keep such information safe and even allow the owner of a lost mobile device to get it back in the future.

With the development of the Internet and associated technologies, mobile devices can perform many of the tasks that people need to do in their everyday routines. Because of this, mobile devices have become an important part of life for people in United States. An investigation shows that the number of Internet users with mobile devices has taken over that of desktop Internet user in 2014. Most of the mobile devices contain very important information of individuals or even the companies, such as contact information, and bank account information. As a result, if a loss or theft occurs, the individuals or companies could lose much more than just the cost of replacing the device.

It is estimated that; "every year In the United States, about 70 million cell phones and smart phones are lost or stolen" [1]. These devices contain sensitive messages, personal and business emails and some other critical information, which could be used by fraudulent people to do some malicious things. People may not care about the mobile device itself very much, but the data inside of the device may be priceless. So, if loss and theft cannot be prevented easily, it is imperative to keep the data safe and not allow access by others. Of course, people hope that their lost or stolen devices could come back one day.

Although mobile security is such an important topic, most people do not have the awareness or ability to do anything about this, including students. Though the Android platform has been used for several years, few universities have related classes for it, especially for mobile security. There is no such mobile security platform for students to study and develop on.

For the reasons listed above, in this paper, our mobile security system will be discussed: Section 2 describes the problem and proposes an architecture of the system in order to protect data by preventing unauthorized login and obtaining the location of the device; Section 3 explains the result of this project; and Section 4 provides the manuals both for developers and users, which would help students get started and develop new features on this platform; Section 5 summarizes this paper, and introduces the future improvements for this project.

II. PROBLEMS and SOLUTIONS

There are some critical issues underlying the current system of mobile devices, which can be used by malware to potentially grab the data inside of the devices. According to a recent report from digital forensics and security firm viaForensics (recently renamed to NowSecure), 76% of applications running on mobile devices store usernames and passwords as plaintext, which poses a risk to users because mobile devices are frequently lost or stolen [6]. In this project, some effective solutions would be adapted to protect the devices and the data inside.

1. System Architecture

For current mobile systems, including both Android and iOS, the usernames and passwords are saved in local storage. As a result, once the devices are lost, stolen,or accessed by malware, others can easily get the user information and login to the mobile devices.

In this Mobile Security System, Client-Server architecture is used as Fig-1. After creating the password for the device, the username and password would be transferred to the cloud server side by calling the RESTful web service API and stored in the cloud server. Consequently, even though the device is lost, stolen, or the device itself is accessed by malware, the attacker still cannot get the user information to login to the device. What’s more, in cloud server side, the username and password can also be encrypted before saving the information into database in server side.

Fig-1

2. Plaintext VS Voice Recognition

The survey emphasizes that the password in most of the mobile device systems are stored locally as plain text, which can be input by the keyboard embedded in mobile system. The password would be easily cracked through continuous attempts.

In Mobile Security System, the keyboard would be disabled and the user can only input the password by the voice recognizer. As a result, it becomes much more complex for attackers to crack the password by trying multiple times. In another word, the voice recognizer enhances the safety of the mobile device.

3. Retrieve Devices

For the traditional mobile devices, it is impossible to get them back once they are lost or stolen. It's a big pain to lose all the data inside of the devices because the person who acquires the lost or stolen devices, or a malware program, can gain root access on the devices, and do some evil. The worst thing is that all of this behavior cannot be traced.

With the application talked about in this paper, all the problems listed above would become much better. It not only prevents others from logging in to the device, but also gets the current location of the device and the information of the fraudulent user, which would definitely help the owner to get the device back. First, if others repeatedly failed to login the device, the system would call the API of Google to obtain the current location, such as latitude and longitude, and convert the location to a detailed real address including building number. Second, the camera embedded in the device would be invoked and take a photo of the fraudulent user. Finally, all the information would be sent to a special email address configured by the owner using the technique of Java Mail. With the detailed address information and the photo of the fraudulent user, it would be much easier to retrieve the lost or stolen devices.

III. RESULT

The Client-Server architecture is implemented in this project as shown in Fig-2.

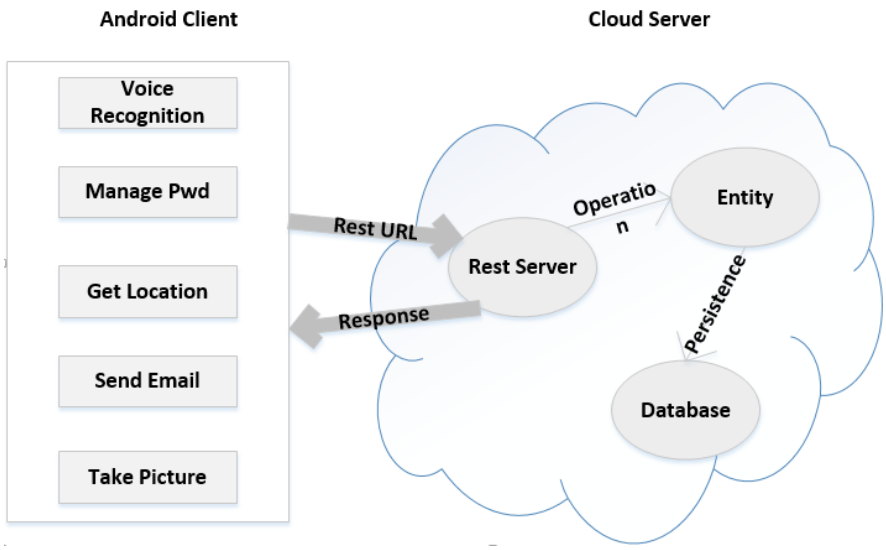

Fig-2

This architecture makes the device and data inside much safer. The voice recognizer can identify the voice accurately, and the identified password is stored in cloud server, where the fraudulent user or malware cannot reach with the device. Normally, detailed geographical location is available for the system. However, the house number will be missed when the system tries to get the address. An email will be received very quickly through Java Mail service. So far, the user interface has to be popped out in order to take a picture.

Such a platform raises the interest of students in mobile security system and provides a stable environment for students to develop on.

IV. USER MANUAL

1. Developers' Manual

a. Development environment

The developer tools for Mobile Security System are listed in table-1 as below. Both of Android Studio and Jetbrains which are the most popular, develop tools for Android and Java, come from company IntelliJ. Maven and Gradle are integrated in them, and either of them can be used to manage the libraries of your project. However, you can also use Eclipse as substitute. MySQL is used in this project as database to store user information. Refer to official guidance if not familiar with installation of MySQL.

Table-1

Android Studio	MobileSecuritySystem_Client	Latest version
IntelliJ IDEA	MobileSecuritySystem_Server	
MySQL	Database	

b. Debug projects

In order to run and debug the primary source code in our local environment, the only thing we need to do is import the projects MobileSecuritySystem_Client and MobileSecuritySystem_Server to Android Studio and IntelliJ IDEA ("File" -> "Import Project…" from the location of projects). Then create a pretty simple data table "users" with two fields, name and password.

After import the projects, MobileSecuritySystem_Server can be deployed immediately. However, before debugging MobileSecuritySystem_Client locally, there are some details we should think about. First, the IP address has to be changed in RestClient class by the IP where the server side is running on (Fig-3). Second, an Android Virtual Device (AVD) is required to run or debug MobileSecuritySystem_Client. Click "Tools" -> "Android" -> "AVD Manager" to create a new AVD as Fig-4. Now you can run or debug the project with the AVD created just now, but confirm that the server is running at the same time. Congratulations! Now you can see the homepage of the project.

Notice: Voice recognition may not be supported in AVD. Therefore, if you want test the related functions, you have to do the test by installing the application in a real Android device.

```
private static final String BASE_URL = "http://10.250.70.56:8080";
```

Fig-3

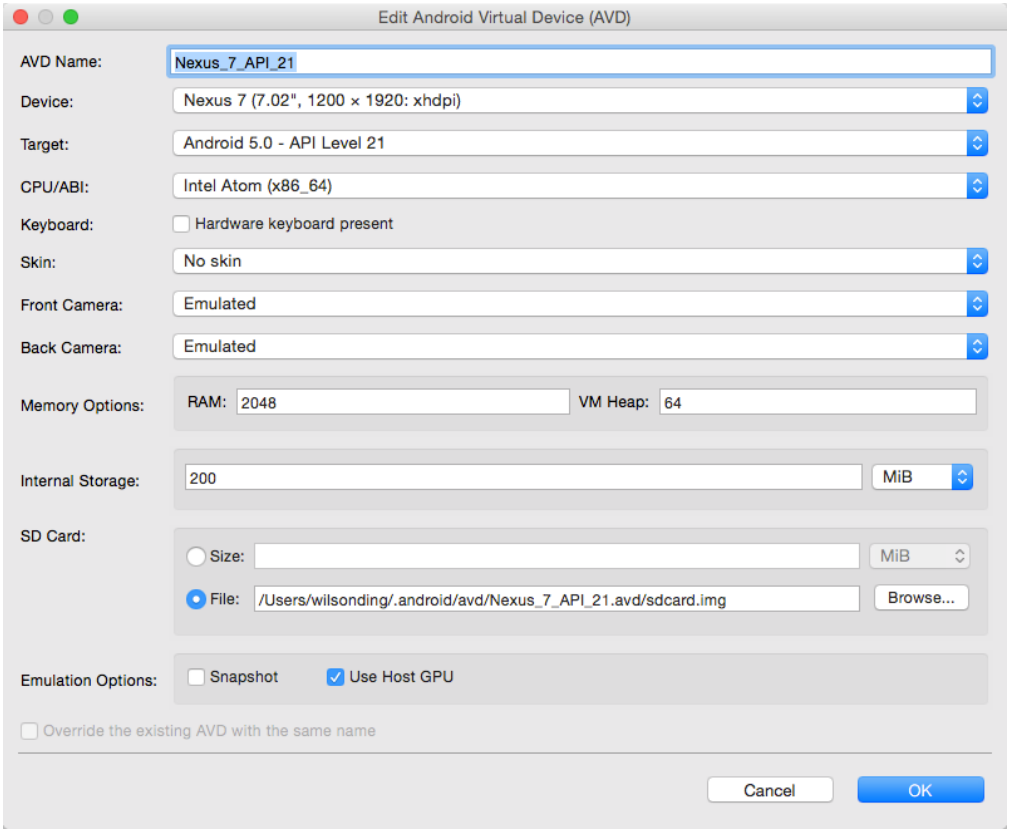

Fig-4

c. Make modification base on primary source code

It is allowed to develop new features based on the original projects, both in server side and client side. If not familiar with the project structure of Android, please refer to http://developer.android.com/training/index.html.

Take resetting password function as an example. First, if we want to reset the password, a new intent is required to input the old and new password. So we have to create new intent in Android Studio by clicking “File” -> “New…” -> “Activity” -> “Blank Activity”. Then you will see Fig-6. After naming the Activity and clicking Finish, a new Activity and Layout would be available to use. Fig-5 indicates what we need in the layout for this case.

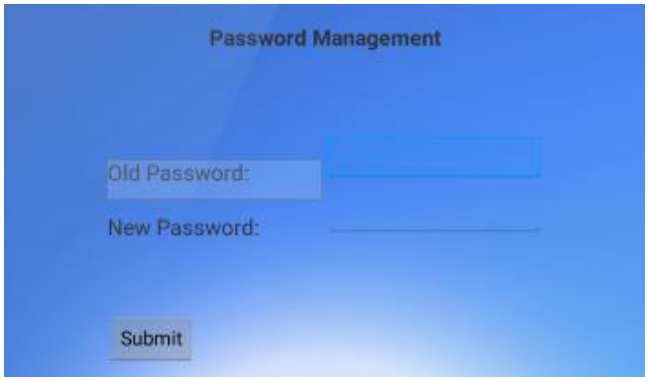

Fig-5

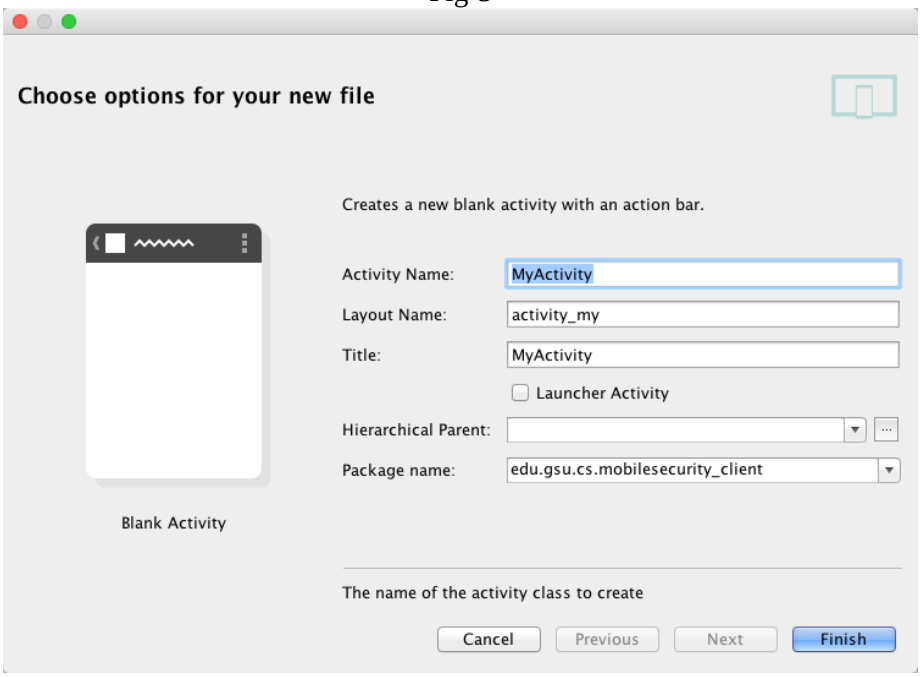

Fig-6

We have to set an onClick event for the "Submit" button as Fig-7. When it is clicked, method "onSubmit" will be called. Fig-8 shows the code for operation in onSubmit method. In this method, client side has to call restful API in server side by RestClient.put(…). Therefore, we have to add RestClient.put(…) method in RestClient class (Fig-9) and restful API in server side(Fig-10).

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvNewPassword"
    android:layout_marginTop="50dp"
    android:layout_marginLeft="100dp"
    android:text="Submit"
    android:id="@+id/btSubmit"
    android:onClick="onSubmit"/>
```

Fig-7

```
url = "/managepassword/updatepassword";
RestClient.put(url,params, new TextHttpResponseHandler() {
    @Override
    public void onFailure(int statusCode, Header[] headers, String responseString, Throwable throwable) {
        Toast.makeText(SetPassword.this,"Failed to create password!",Toast.LENGTH_LONG).show();
    }

    @Override
    public void onSuccess(int statusCode, Header[] headers, String responseString) {
        if(responseString.equals("success")){
            Toast.makeText(SetPassword.this,"Password is created successfully!",Toast.LENGTH_LONG).show();
        }else{
            Toast.makeText(SetPassword.this,"Old password is not correct!",Toast.LENGTH_LONG).show();
        }
    }
```

Fig-8

```
private static AsyncHttpClient client = new AsyncHttpClient();
public static void put(String url, RequestParams params, TextHttpResponseHandler responseHandler) {
    client.put(getAbsoluteUrl(url), params, responseHandler);
}
```

Fig-9

```
@PUT
@Path("/updatepassword")
public String updateUser(@FormParam("oldPassword") String oldPassword, @FormParam("newPassword") String newPassword) {

    String result="fail";

    EntityManagerFactory entityManagerFactory= Persistence.createEntityManagerFactory("NewPersistenceUnit");
    EntityManager entityManager= entityManagerFactory.createEntityManager();
    entityManager.getTransaction().begin();
    UsersEntity user= entityManager.find(UsersEntity.class,"iwil");
    if(user.getPassword().equals(oldPassword)){
        user.setPassword(newPassword);
        result="success";
    }

    entityManager.persist(user);
    entityManager.getTransaction().commit();
    System.out.print(user);
    entityManager.close();

    return result;
}
```

Fig-10

By this way, the password of the user name “iwil” will be reset, and stored in the persistent database.

2. Users’ Manual

a. Installation of the application

In order to install Mobile security in real Android devices, the first thing to do is copy the installation file (app-debug.apk) from the directory “/MobileSecurity_Client/app/build/outputs” of the Android project to a real device or upload it to Google Drive and run it directly from Internet. Then execute the installation file by clicking app-debug.apk. After clicking “install” and “open” showed in Fig-11 and Fig-12, this application would be installed and run.

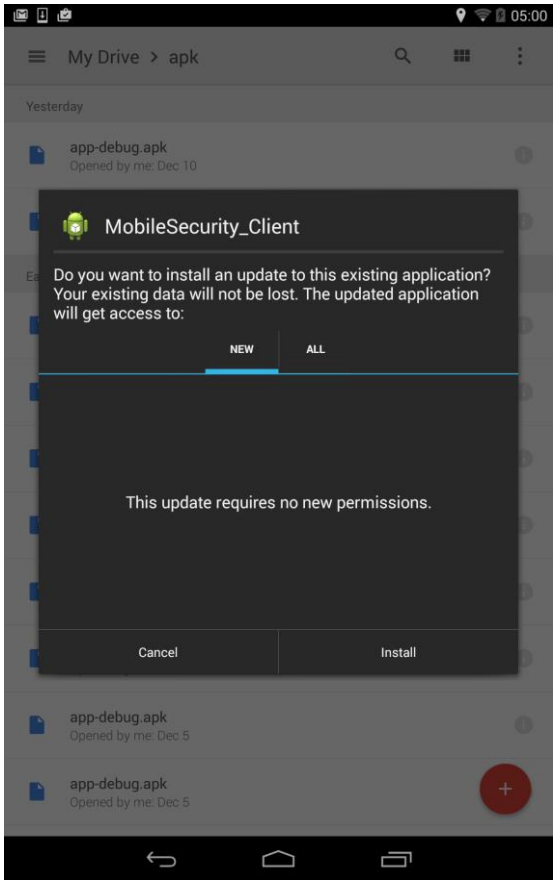

Fig-11

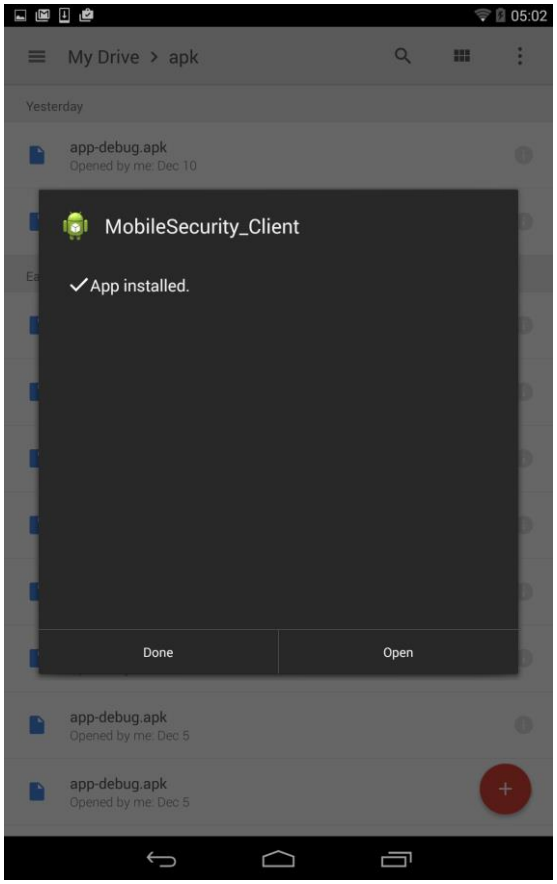

Fig-12

b. Password management module

After running Mobile Security System for the first time, the service would run automatically in the background. The user interface of this application would pop up when starting up the device, or waking the device up from sleep. The application will be full screen and disable other keys embedded in the device, as Fig-13.

If the device is a brand new one, it is required that a password has to be set at the very beginning. By clicking the “Set your password”, password management would come out, which contains one edit text element used to input your new password, as showed in Fig-14. Click edit text area, the voice recognizer will be available waiting for your voice. After saying your password, click the “Submit” button to set the password, and there will be a notice at the bottom, which says, “Password is created successfully”.

Once the device has already had a password, similar steps can change the password but both old password and new password are required as showed in Fig-15.

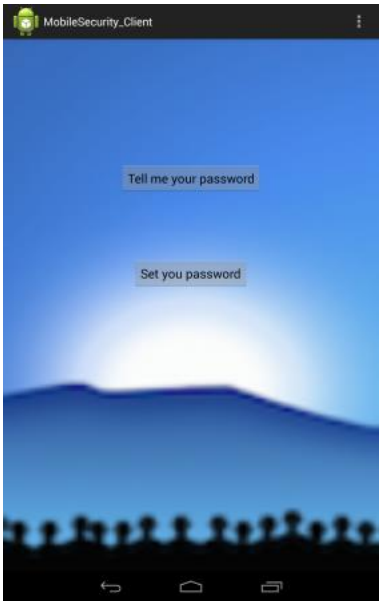

Fig-13

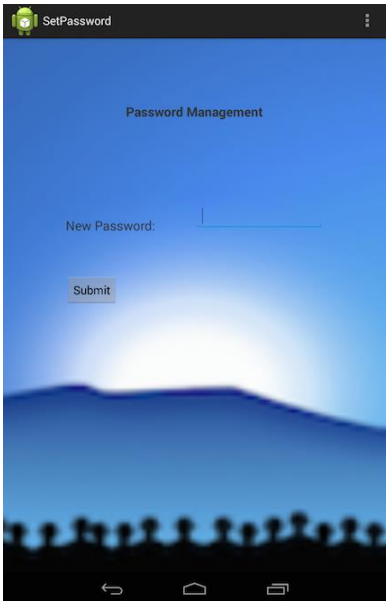

Fig-14

Fig-15

Fig-16 indicates the entire workflow for password management function module.

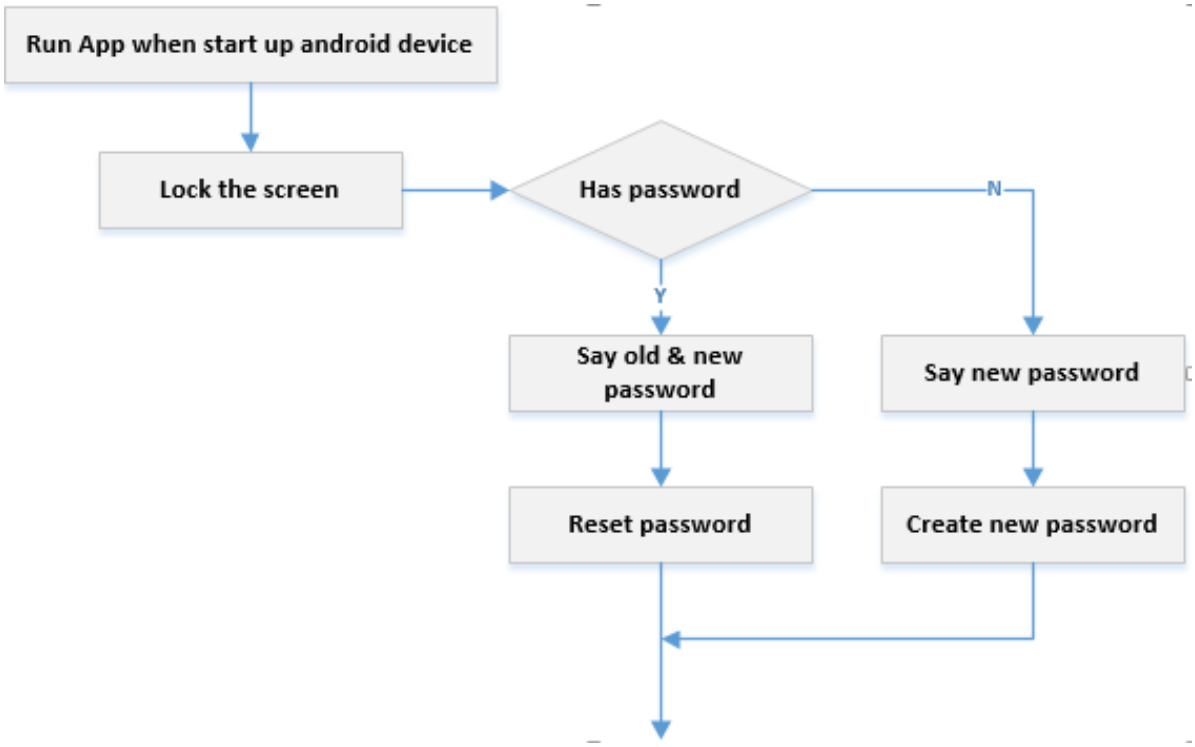

Fig-16

c. Login module

By clicking “Tell me your password” button, voice recognizer would come up to accept your voice. Fig-17 describes the page in Mobile Security System.

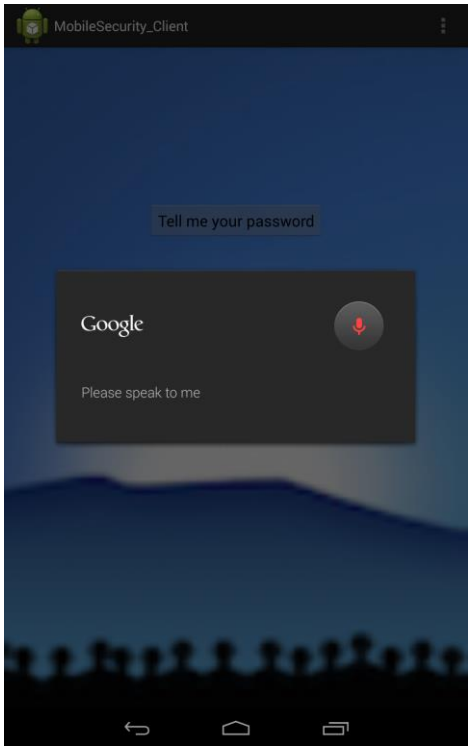

Fig-17

After saying password to the device, if the password were correct, the screen would be unlocked automatically, and then the device can be used as usual; otherwise, the embedded camera would be activated to take a picture, and then there would be a message, said “Password is incorrect”. At the same time, the specific email set previously would receive a notice with the information of current location of the device (Fig-18).

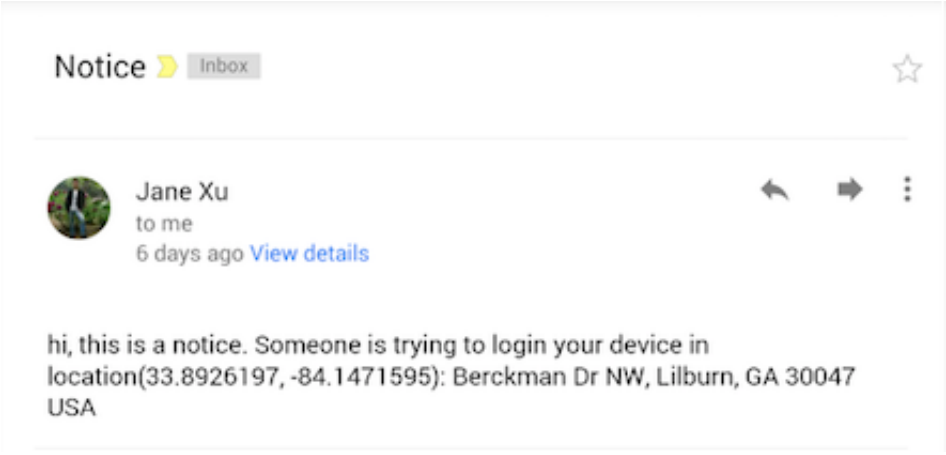

Fig-18

Fig-19 shows the whole workflow of login function module.

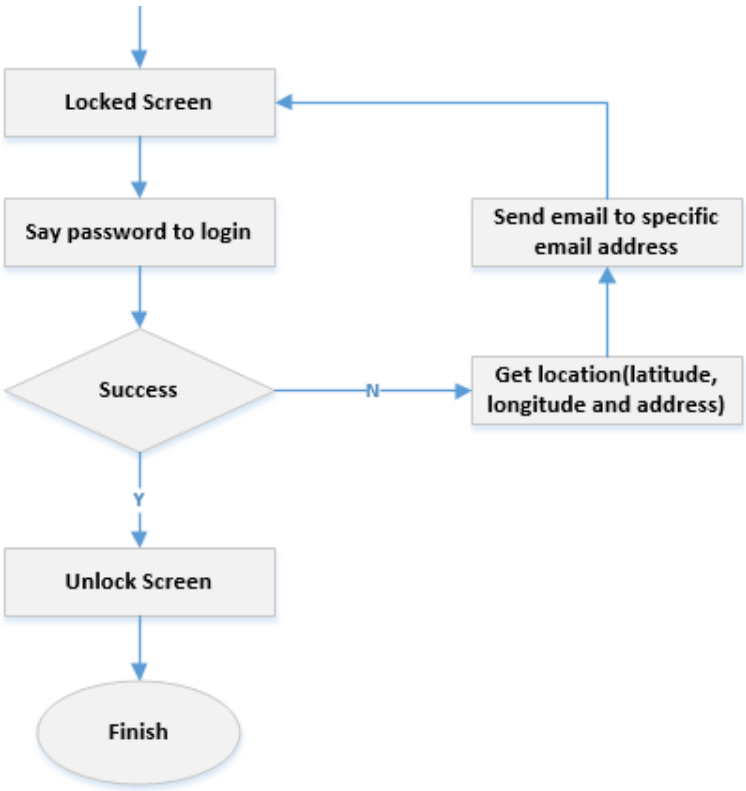

Fig-19

V. EVALUATION

We have conducted pre-evaluation and post-evaluation of our students so that we can measure the learning experiences and outcomes in these 4 courses. Below are the evaluation results.

Pre-Evaluation

All the participants are students from the computer science department at Georgia State University. The assessment is divided into three parts:

- Work experience with computer and programming language used (written response)
- Knowledge of operating system (choice question)
- Study experience of PC and different ways to learn (choice question)

The diversity in the nature of question reflects both the understanding of students about the operating system and the best way for the students to learn it effectively.

Written response – Operating System:

Work Experience	**YES (%)**	**NO (%)**
Have computer work experience	42.9	57.1

Written response – Computer Network:

Work Experience	**YES (%)**	**NO (%)**
Have computer work experience	73.3	26.7

Written response – Web Programming:

Work Experience	**YES (%)**	**NO (%)**

Have computer work experience	60.4	39.6

The table above shows that more than half of the students have computer work experience during their campus life. The participants are also asked to write down all the programing languages they have used. More than 71% of them write down JAVA as their first choice; and above 14% of the participants take JAVA as their second programing language following C/C++. Obviously, JAVA is widely used in the participants of this survey.

Knowledge of computer – Operating System:

Knowledge of computer	0(%)	1(%)	2(%)	3 or more(%)
Computer systems at home	7.1	14.3	7.1	71.4
Computer system at school	21.4	35.7	28.6	14.3
Years using a personal computer	--	--	--	100.0
Years working on special OS	35.7	21.4	7.1	35.7
Years programing	--	7.1	28.6	64.3

Knowledge of computer – Computer Network:

Knowledge of computer	0(%)	1(%)	2(%)	3 or more(%)
Computer systems at home	--	13.3	26.7	60
Years programing	--	--	33.3	66.7

Knowledge of computer – Web Programming:

Knowledge of computer	0(%)	1(%)	2(%)	3 or more(%)
Computer systems at home	14.6	22.9	20.8	39.6
Computer system at school	6.4	50.2	27.7	14.9
Years using a personal computer	6.4	4.3	--	89.4
Years working on special OS	50	4.3	19.6	26.1
Years programing	10.6	14.9	23.4	51.1

In this part, all the participants have worked with office software in a personal computer for more than 3 years, and almost of them have used more operating systems at both home and school.

Study experience – Operating System:

Study Experience	SA(%)	A(%)	N(%)	D(%)	SD(%)
Had mostly positive experience using PC	57.1	35.7	7.1	--	--
Had mostly positive experience programing	14.3	50.0	35.7	--	--
Comfortable using computer	85.7	7.1	7.1	--	--
Learn better by hands on lab work	50.0	35.7	14.3	--	--
Learn better by listening to lecture	7.1	28.6	35.7	28.6	--
Learn better by examples	50.0	42.9	7.1	--	--
Learn better by reading material	14.3	14.3	57.1	14.3	--
Learn better by learning system with feedback	21.4	57.1	21.4	--	--

Study experience – Computer Network:

Study Experience	SA(%)	A(%)	N(%)	D(%)	SD(%)
Had mostly positive experience using PC	60	40	--	--	--
Had mostly positive experience programing	33.3	60	6.7	--	--

Comfortable using computer	73.3	20	--	--	6.7
Learn better by hands on lab work	53.3	40	6.7	--	--
Learn better by listening to lecture	6.7	26.7	53.3	6.7	6.7
Learn better by examples	66.7	33.3	--	--	--
Learn better by reading material	26.7	20	26.7	20	6.7
Well know secure communication	6.7	20	33.3	33.3	6.7
Well know cryptography	6.7	13.3	26.7	46.7	6.7
Well know SSL	--	13.3	40	40	6.7

Study experience – Web Programming:

Study Experience	**SA(%)**	**A(%)**	**N(%)**	**D(%)**	**SD(%)**
Had mostly positive experience using PC	59.6	34.0	6.4	--	--
Had mostly positive experience programing	27.7	55.3	10.6	6.4	--
Confortable using computer	76.7	14.9	2.1	--	6.4
Learn better by hands on lab work	68.1	21.3	10.6	--	--
Learn better by listening to lecture	4.3	12.8	51.1	23.4	8.5
Learn better by examples	63.8	27.7	6.4	--	2.1
Learn better by reading material	12.8	36.2	23.4	27.8	--
Learn better by learning system with feedback	34.0	38.3	14.9	12.8	--

SA=Strong Agree; A=Agree; N=Neutral; D=Disagree; SD=Strong Disagree

This table indicates that majority of the participants feel comfortable to program using a personal computer. More than 70% of the students agree on that they would learn better by hands on lab work. Similarly, 92.9% of them insist that examples would help them to learn something more effectively than any other way, such as listening to lecture and reading material.

The students in class of Embedded System also completed such a questionnaire. There are totally 26 responses to the initial questionnaire (pre-survey). In the initial questionnaire, the vast majority (24 of 26) indicated "3 or more" embedded systems at home. The majority of responders indicated that they have one computer at school. While 15 of 26 reported working on embedded systems for less than a year, 19 said they have been programming 3 or more years. For the top three languages, 24 students reported knowing Java, 16 have programmed in C, and 10 have experience with C++. Of the responses to the question about work experience, 9 were blank, 4 answered "none", and another 3 indicated that their job was as a student, for a total of 16. Five responded with limited work experience, such as less than 1 year. Another seven reported significant work experience lasting over 1 year. Of those with significant experience, two stand out from the rest: one has 9 years experience while another has over 35 years experience. The vast majority (21) reports that they are very comfortable using computers. A solid majority (16 strongly agree plus 7 agree) has mostly positive experiences when using a computer. A smaller majority (8 Strongly agree plus 11 Agree) has mostly positive experiences when programming. Most responders either strongly agree (14) or agree (8) that they learn better by hands-on lab work. Similarly, 16 strongly agree with personally doing or working through examples, with another 6 agreeing. A learning/tutorial system that provides feedback has roughly two-thirds agree or strongly agree, while another one-third is neutral. In contrast, 12 are "neutral" about listening to lectures, with 5 agreeing and 8 disagreeing. Students were divided about reading the material on their own.

Post Evaluation

The post evaluation is composed by two sections of questions:

- The study experience with labware
- Written response

Study experience with labware – Operating System:

Experience with Labware	**SA(%)**	**A(%)**	**N(%)**	**D(%)**	**SD(%)**	**N/A(%)**

Like being able to work with labware	22.2	22.2	33.3	--	--	22.2
Labware on mobile security learn more	11.1	22.2	33.3	--	--	33.3
Understand security better with labware	11.1	44.4	22.2	--	--	22.2
Labware improving learning experience on operating system	11.1	33.3	33.3	--	--	22.2
Labware helps apply learned knowledge	22.2	22.2	11.1	--	--	44.4
In class demonstration to learn the topic	33.3	33.3	11.1	--	--	22.2

Study experience with labware – Computer Network:

Experience with Labware	**SA(%)**	**A(%)**	**N(%)**	**D(%)**	**SD(%)**	**N/A(%)**
Like being able to work with labware	32.1	50.0	17.9	--	--	--
Complete labware of computer network	X	75.0	X	25.0	X	X
Labware on mobile security learn more	22.2	51.9	11.1	--	--	14.8
Understand security better with labware	25.9	48.1	11.1	--	--	14.8
Labware improving learning experience on computer networks	33.3	51.9	11.1	--	--	3.7

Study experience with labware – Web Programming:

Experience with Labware	**SA(%)**	**A(%)**	**N(%)**	**D(%)**	**SD(%)**	**N/A(%)**
Like being able to work with labware	8.8	44.1	32.4	5.9	2.9	5.9
Labware on mobile security learn more	14.7	44.1	26.5	8.8	--	5.9
Understand security better with labware	11.8	50.0	26.5	5.9	--	5.9
Labware improving learning experience on operating system	3.0	39.4	39.4	12.1	--	6.1
Labware helps apply learned knowledge	11.8	44.1	26.5	11.8	--	5.9
In class demonstration to learn the topic	8.8	44.1	32.4	8.8	--	5.9

SA=Strong Agree; A=Agree; N=Neutral; D=Disagree; SD=Strong Disagree;
N/A= does not apply to me; X=no such answer

From the response of this section, majority of the students like being able to work with labware and believe that the labware would be helpful to apply their learned knowledge in class so that they could learn and understand the mobile and PC security more easily. However, there are also some students think such a labware has nothing to do with them while others show their interest in it.

Written response:

Although some of the students didn't add additional comments in this part, some others expressed their interest in learning with such a wonderful labware for the courses. In their opinion, the class contains more fun with the hands on activities, from which they would learn better of the related knowledge.

There are also 25 responses to the final questionnaire (post-survey) from Embedded System class. 17 strongly agree that they like being able to work with the labware, with another 6 agreeing. The majority (23) agrees or strongly agrees that the labware on mobile security help them learn more. The majority (20) agrees or strongly agrees that the labware helped them understand computer security. Similarly, 20 also agree or strongly agree that the labware helped them understand embedded systems. Ten respondents are neutral when asked about the in-class demonstrations of embedded system security, with 11 agreeing or strongly agreeing. When asked for additional comments, the majority (18) left no response. Of the other 7 responses, two indicated that would have liked more labs.

VI. CONCLUSION

In this paper, potential mobile security problems are described and analyzed, and a solution is provided. Storing passwords in a cloud server makes it much more complex for fraudulent users or malware to access the data inside of devices than in a traditional architecture. A notice email with location information and a picture

will be available immediately when someone tries to login the device. But the application needs improvement to hide user interface of the camera when taking the picture, so that a malicious user cannot notice it. Based on the above ideas, an original Client-Server platform is built in this project, which can be used for students to work on and make further improvements. Creating new features based on this application will increase the awareness of students about mobile security. We have tested the labware in four classes: Operating Systems, Embedded Systems, Computer Networks and Web Programming. The preliminary results and feedback from the students in these classes indicate that the labware is useful for them to gain the knowledge in mobile security systems. Based on pre-evaluations and post-evaluations on the four courses, we have got positive feedback from students. Almost all the students have their own mobile devices and feel comfortable working with them. The newly developed mobile security system on Android and JAVA made the students easily learn how mobile security systems work via several hands-on exercises. In the Operating Systems class, many students also did research projects related to mobile security, and gave class presentations to share their research results in the class. Most students agree that the labware on mobile security can help them learn faster and better. Our experiments using the labware to teach security in 4 computer science courses indicate that we can seamlessly integrate security education in the computer science curriculum.

VII. REFERENCES

[1] Fogarty Kevin, *"Call carriers launch anti-theft effort they could have started in 1996"*, Oct. 2012.

[2] Nampalli Mari Heiser. "*Mobile Security: How to Secure, Privatize, and Recover Your Devices*", September, 2013.

[3]Kai Qian, Chia-Tien Dan Lo, Minzhe Guo, P. Bhattacharya, Li Yang, *"Mobile security labware with smart devices for cybersecurity education"*, 2012.

[4] N. Penning, M. Hoffman, J. Nikolai, Yong Wang, *"Mobile malware security challenges and cloud-based detection"*, 2014.

[5] P. Bhattacharya, Li Yang, Minzhe Guo, Kai Qian, *"Learning Mobile Security with Labware"*, Jan.-Feb. 2014.

[6] http://www.informationweek.com/mobile/most-mobile-apps-fail-password-security-test/d/d-id/1099451?

[7] http://developer.android.com/guide/index.html