SPECIFICATION BASED TESTING OF ON ANDROID SYSTEMS

Yujian Fu[1] and Wichien Choosilp[2]

1,2Department, USA of EE&CS, Alabama A&M University,4900 Meridian Street, Normal AL

ABSTRACT

With the surging of mobile applications, mobile security draws more and more attentions from researchers in various areas. Due to the lack of quality assurance approaches in mobile computing, many mobile applications suffer the vulnerabilities and security flaws. In this paper, we proposed a model based unit testing approach on the android security properties using JUnit. Both behavior and structure model of the android application were developed on the Unified Modeling Language (UML) – behavior is described in state diagram, while structure is described in class diagram. Our approach focus on two common security groups – the access control and authentication properties. Both groups are represented in the operations defined in the class diagrams and dynamic behaviors are captured (partially) in the state diagram. A set of well defined test cases is developed to validate the desired properties based on the class diagram. All properties on the class diagram and state diagram are described in Object Constraint Language (OCL) – a formal specification language on the first order logic and set theory.The results of this research will provide a sound foundation towards the specification based unit testing on mobile security.

KEYWORDS

Software testing, security properties, malware analysis

1. INTRODUCTION

We are experiencing a dramatic increase of mobile users as well as mobile applications. According to the LA Observer website [9], 91% of U.S. adults own cell phones and 56% own smart phones. Android OS is one of the most popular mobile platforms. In October 2012, there were approximately 1,000,000 apps available for Android, and the estimated number of applications downloaded from Google Play, Android's primary app store, was 50 billion as of September 2013. Due to the proliferation of mobile applications, mobile phone plays an important role in our daily life to achieve various functionalities. More and more mobile users are now paying more attention to protect their privacies since lots of personal data are now generated and stored in mobile phones. Unfortunately, mobile platform has less protection against botnet and malware, therefore becomes the targets of malicious users and attackers.

There is an increasing number of malicious mobile applications, called mobile malware, targeting on mobile devices and platforms. Nowadays, mobile malware have reached a new level of maturity. For instance, mobile malware has been used to steal personal contacts [28]. Even the launch of anti-malware Android Bouncer [11] to scan apps from Android Market for evidence of malware cannot prevent the increasing threat of malware for mobile platforms [10]. Many mobile

DOI: 10.5121/ijwmn.2017.9501

users have not been aware of the risks of mobile malware, and they may download Android application from third-party markets that can easily purvey malicious applications [32, 31]. Even worse, attackers have already found out ways to evade detections [12]. Threats targeting smart phones and tablets are beginning to pose meaningful challenges to users, enterprises, and service providers alike. The number of instances of just one family of malware can be in the thousands.

The largest proportion of malware is targeting on the Android mainly due to the dominant market share of the Android platform and its open market policy.

However, many of the smart phone and tablet users have, for the most part, not been aware of the risks of mobile malware, of which there are many. However, it seems thatthe attackers have found ways to evade detections [12]. To make matters worse, there are other, third-party markets that are available for downloading Android applications that can easily purvey malicious applications [32,31].

Intensive research has been done in order to protect mobile users from the severe threats of Android malware. One research area is to investigate the ways malicious behavior are triggered. For example, a group of malwares can stay dormant until the occurrence of an specific event [24]. Some events are independent of user interactions with applications (i.e. existence of network, etc), yet some others are based on user inputs [30,25]. Recently security testing has gained popularity and has proved its importance [8] in this research area.

Security testing on mobile applications is challenging due to the nature of mobile platform and heterogeneity of the technologies [2]. The foundation of security testing remain same to traditional software testing regardless of the scope and devices. Two primary methods are being employed for security testing on mobile application:white-box approach and black-box approach. Black-box testing is to reveal the relation betweenapplication inputs and outputs as a function: inputs and outputs are viewed as parameters (variables)and image (co domain), respectively. On the other hand, whitebox testing is to reveal internal execution stepwise by investigating the static structure of source code. Fuzz [27, 20], a pioneer tool in the software testing, generates random inputs to the command line to leverage the security vulnerabilities. The output data of Fuzz is used as the inputs for many software testing tools.

In contrast, activity centered graphical user interface design is heavily adopted inAndroid applications. Therefore, it is highly possible to get large number of objects for the same class (activity) once running the application for some time. The consequence of this phenomenon cause analyzing and figuring out a threat inaccurately by gathering information needed. Among numerous execution paths in an application only a small number are covered by merely starting or running the application. Since dynamic analysis checks the executing codes behavior, to provide better coverage the testing tool has to provide Graphical User Interface (GUI) input so that more path scan be covered. This paper presented a complete process for the unit testing on the Android app with several credentiality properties, some of them focus on the GUI testing. Our previous work has presented an implementation of SMS message in Android [33].

The rest of the paper proceeds as follows. In Section 2, SMS system architecture and design model are introduced; in addition, the cryptographic algorithms used in the SMS message system are presented with our implementation. Based on those, in Section 3, we describe security requirements on the design model which are specified by Object Constraint Language – OCL - for SMS. These properties are defined before perform unit testing. After that, in Section 4, the

testing modules of security properties are introduced by focusing on the unit testing and unit testing tool JUnit is adopted as the typical testing tool. We conclude in Section 5, as well as discuss the limitation and the future work.

2. ARCHITECTURE AND DESIGN

In this work, we propose an approach to test SMS message in Android apps using Object Constraint Language (OCL) and UML model. UML models have been used in model based security engineering, such as Juerjens's work in [14,16,15] to develop security-critical software intensive systems where security requirements (such as secrecy, integrity, authenticity) on the system environment can be specified either within a UML specification or within the source code.

Short Message Service (SMS) is a radio based service for transferring short alphanumeric messages among mobile phones on GSM and UMTS cellular networks[29]. SMS allows mobile GSM phones to send and receive text messages.SMS is a widely used service for brief communication. Data sent through SMS services is confidential and should not be disclosed to a third party. A sent SMS message is stored at an SMS Center (SMSC) until the receiver's phone receives it. The receiver can identify the sender by his/her telephone number that is included in the message itself. To allow interconnection with different message sources and destinations, SMS supports several input mechanisms including voice-mail systems, Web-based messaging and E-mail integration. SMS service can also be used to provide some additional services for mobile information services such as mobile electronic commerce, mobile transactions, news, sports, and entertainment services.

2.1. SMS Architecture

Short Message Service (SMS) was introduced in wireless networks and GSM standards at the beginning of 1990's. SMS refers to a wireless, radio based service for transferring short alphanumeric messages among mobile phones on GSM and UMTS cellular networks [29]. It is a two way transmission service that uses an SMS Center (SMSC) as a store-and-forward unit for messages. The acknowledgement will be sent to the sender to notify for a success or failure of the transmission. In addition, a mobile handset is able to receive or send a message anytime, even when an active call is in progress. The message is kept in the network until the destination is available when some failures occur. It also has special features of out of band packet delivery and low-bandwidth transferring of message. The format of an SMS contains a byte array with a message header and body – where the details are attached in the header and the actual message that need to be sent along is in the body with the message. The SMS message can be converted to 70 ASCII characters using Base64 encoding with up to 160 alphanumeric.

SMS messaging is a mobile and stronger version of two characteristics: "anytime and anywhere availability", in comparison with the current other mobile ISP services. A switched-on mobile device is able to receive or send a message at anytime regardless of whether a voice or data call is in progress. Messages sent to a switched-off phone are guaranteed to deliver when the handset is on again because SMS messages are users. One application is in the selective advertising business for promotional purpose. For example, restaurant operators can entice customers by sending them advertisements and promotional information messages when they are in the vicinity of restaurants.

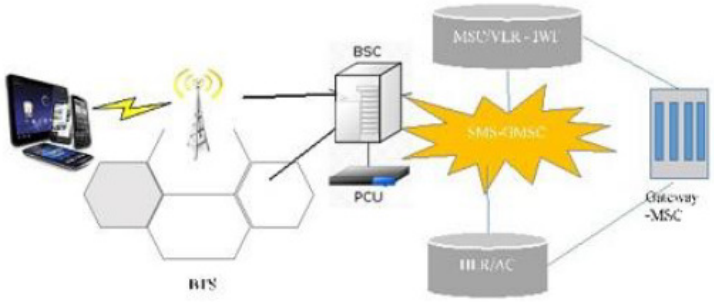

Figure 1. SMS Architecture.

Fig. 1 shows a simplified SMS architecture. Mobile Application Part(MAP) layer using the SS7 service is used to provide a platform for the various nodesin GSM mobile and GPRS core networks and GPRS core networks to communicate with each other. A Short Message Entity (SME) is an entity (mobile/cell phone) that connect to the SMS and can send or receive messages. Subscribers compose and send messages from a mobile station to the nearest Base Transceiver Station (BTS).

The SMS is transmitted to the BTS through a standard On-The-Air (OTA) interface which is used in wireless devices for transmission and reception [1]. The BTS then forwards the SMS to the mobiles home SMSC through the Base Station Controller(BSC) and Mobile Switching Center (MSC) over Signaling System (SS7), which send data grouping that is used by SMS service [1]. Repackaging the message intoa short message peer to peer (SMPP) format – a standard telecom protocol – if the SMS is exchanged between subscribers in two different operators. The sender will forward it over private or public networks that are connected to the recipients SMSC. Also, the OTA interface is used to forward the SMS to the destined SME by BTS. The notification message path from the recipient is done in the reverse order following the same path. A Home Location Register (HLR) is a set of devices to route the SMS to the destination by existing registered address. The SMS Gateway Mobile Switching Center (SMS/GSMC) receives an SMS from SMSC, was routed through HLR,and delivers the message to the expected recipient's MSC of the mobile station. HLR keeps the information regarding previously initiated delivery attempts to a specified destination for both successful and unsuccessful. If the previous unreachable attempts was recognized as reactive network, the HLR will inform the SMSCto reattempt to the undelivered [20].

2.2. Cryptographic Algorithm in SMS Architecture

AES algorithm is the most widely accepted algorithm for data encryption. Many SMS applications are developed based on the variant or improved AES. AES allows users to encrypt messages before they are transmitted over network. One of the main advantages of AES is the very low memory space requirement with efficient performance. This was the main reason why AES algorithm was selected for encryption and decryption [13].

There are few SMS applications on Google Play which encrypts the SMS using(advanced) AES algorithm. The Advanced Encryption Standard comprises three block ciphers, AES-128, AES-192 and AES-256 based on the key block size of 128,192, or 256 bits. The block-size has a maximum of 256 bits, but the key-size has no theoretical maximum. The user's input is given bya

state matrix [22]. To get the cypher text from plain text, a number of encryption rounds will be generated, where each output of a round is the input to the next round. The cypher encryption procedurecan be described symbolically as

$$C = E(ks(K),M),$$

Where E denotes two-variable encrypting function, K is the secret key chosen byuser, M is the message to be encrypted, k sis the explicit function.

Our application is programmed with simple GUI interface and encryption algorithm by meticulously considering various factors which might benefit the user. It provides functionality like conversation view, message sending, and restore with a standard SMS application. Our current SMS encryption subsystem is very a simple app, very easy to operate and understand. Thus the encryption and decryption of SMS is carried out very efficiently. A snapshot of secret key generation is shown in following.

```
byte[] returnArray;
// generate AES secret key from user input
Key key = generateSecretKey(secretKeyString);
// specify the cipher algorithm using AESCipher c = Cipher.getInstance("AES");
// specify the encryption mode
c.init(Cipher.ENCRYPT_MODE, key);
returnArray = c.doFinal(msgContentString.getBytes());
```

2.3. Design Model

In our work, we represented the security properties of SMS in the Object constraint language (OCL), which is a formal specification language that is based on first order logic and set theory.OCL is defined by set and operations on the various of sets, and embedded in the UML now. Upon the introduction of the OCL, UML diagrams can be formal reasoned and many properties can be specified precisely. To realize high assurance and trusted software intensive systems, OCL can be integrated to the software testing tool for the well developed test suites. JUnit is an automatic testing tool that apply unite testing on Java programming language. In the system design architecture, the OCL formula is associated with the design model – class diagram – in this case is used to represent the static structure and topology of the system. JUnit tool is widely applied in many applications, including military systems, information systems and embedded systems.

Many mobile phone manufacturers offer some type of support for developing applications on their products, but by far the most standardized and also the most portable solutions are based on Java. A more restricting requirement is the need to be able to send and to receive SMS messages by the application. There are Java solutions that support this functionality to some extent. Obviously, demands on restricted processing power as well as memory consumption are tackled with tools of software engineering and are somewhat independent on the selection of the development environment. With these things in mind the decision to choose Java as the tool for developing this application was made.

This work starts with the class model that simply generated for the developed Android system. The class module aims at introducing the working mechanisms of mobile malware and a

traditional cryptographic defense method that is employed. Our work focuses on the sequence of the design, security enforcement implementation, security specification and testing based on the framework of the model driven security engineering approach. Without considering the Linux kernel, we generated the class models from the SDK in eclipse. Our OCL model specifies the permission and chain of permission for the access control of SMS. Moreover, the notion of permission is to release capabilities, whereas the sandbox is to provide restrictions. Accordingly, we concentrated on the releasing features, assuming that the sandbox mechanism is working properly. We only include security-related features in our specification sin order to keep the specifications to a manageable size. Modeling the permission mechanism of the Android system goes through several steps, which are similar to the Goguen - Meseguer program described by Cristia [9] – list the security needs; describe the system; describe the conditions; prove security conditions are satisfied.

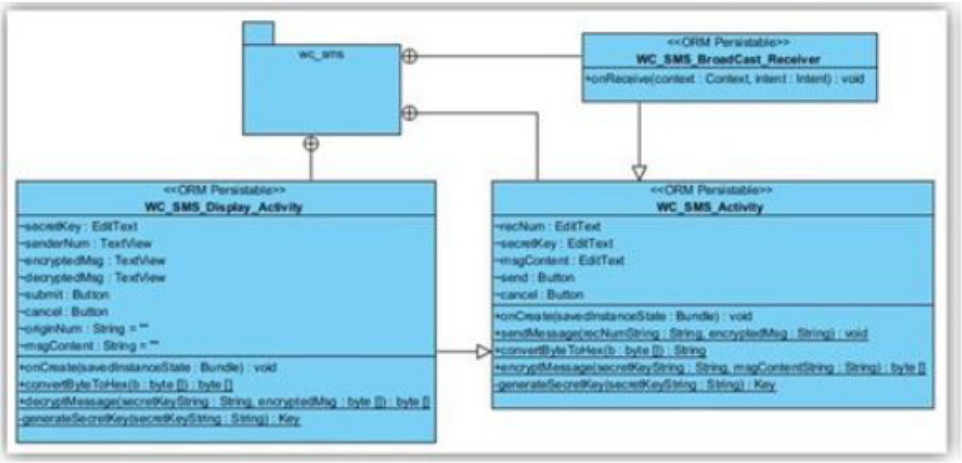

Figure 3. Class Model of Android System.

In our work, the UML class diagram in Figure 3 serves as an introductory modeling.The diagram represents the simplest of all possible communications messagesthat go through two communicating Android phones. The diagram depicts three classes that are responsible for the encoding and decoding, sending and receiving activity, text and message display, and broadcast receiver respectively. The display class accept the message and key from the user with the generated key. Sending and receiving message and decoding the text using the generated key are handled by the activity class. The application is used to describe the SMS hacking using the Android-apk tool.

3. SECURITY SPECIFICATION IN OCL

Our work focuses on the assurance of application oriented security, and userinterface design at the API level for the Android SMS apps. Several APIs are providedin the ADT bundle. In terms of API user interface design, Activity class is thekey of the apps. An activity is a single, focused thing that the user can do. Almost allactivities interact with the user, so the Activity class takes care of creating a windowfor you in which you can place your UI with setContentView(View). While activitiesare often presented to the user as full-screen windows, they can also be used in otherway, such as floating windows or embedded inside of another activity (using ActivityGroup).

There are two methods almost all subclasses of Activity will implement:

1. On Create(Bundle) is where you initialize your activity. Most importantly, here you will usually call setContentView(int) with a layout resource defining your UI, and using find View By Id (int) to retrieve the widgets in that UI that you need to interact with programmatically.
2. OnPause() is where you deal with the user leaving your activity. Most importantly, any changes made by the user should at this point be committed (usually to the Content Provider holding the data).

In terms of the cryptographic development, SDK provides a set of classes to implement the typical AES algorithm. javax.crypto.spec is a package that supports the classes and interfaces needed to specify keys and parameter for encryption. Keys maybe specified via algorithm or in a more abstract and general way with ASN.1. For example, SecretKeySpec() A key specification for a Secret Key and also a secret key implementation that is provider-independent. It can be used for raw secret keys that can be specified as byte.

To express the properties of SMS in Android apps, it can be identified by a sequence and/or group of properties in OCL. There are two main groups of properties to ensure the system meets the security requirement – identification and data integrity. In addition, the connection of two properties from the two groups can be chained and form a one property to ensure the security is maintained through the communication.

In our implementation, identify can be located by recognize number, secret key and message content, which we specify follow identity validation formula in OCL:

contextWC_SMS_Activityinv:
self.findViewById(R.id.recNum)
contextWC_SMS_Activityinv:
self.findViewById(R.id.secretKey)
contextWC_SMS_Activityinv:
self.findViewById(R.id.msgContent)

The group of identity validation concentrates on the single message identity that is needed before sending a SMS message. Once it is established, operations of sending and receiving the message will be the main focus. By encryption and decryption, all the data are transferred in the cypher text. How to ensure the final data used for the apps is the same as the original message transferred is a key issue now.

In our implementation, the operation invariants can be specified by a set of formula eto ensure the above two invariants key conversion and operation invariants. Some operation invariants are listed in the following formulae in OCL format:

contextWC_SMS_Activity :: sendMessage(String, String) :
inv :self.encryptedMsg.size() <= 160
self.recNumString != NULL
Some key conversion invariant regarding to our application can be:
contextWC_SMS_Activity::convertByteToHex(byte[] encryptedMsg)
post: result = self.toHexString(encryptedMsg)

Consequently, we can define the data integrity invariant as the conjunction of operation invariant and key conversion invariant.

4. UNIT TESTING ON ANDROID SECURITY

The objective of the testing is to confirm whether all transitions satisfy the security property occurring over states based on the above OCL specification and class diagram. The general foundation of validation approach is always same it does not matter the difference and variants of the systems. It is widely admitted that automated unit testing saves time by generating test cases and performing many tests. Automate disconsidered as a quick way to validate new builds-in the form of build acceptance or smoke tests. Finally, unit testing can be an effective way to verify each area of functionality within the application performs as expected across a wide range of devices in a systematic and reproducible fashion.

The Android SDK supports JUnit forautomated unit tests.

```
public void testActivityVariables() {
                assertNotNull(getActivity());
                assertNotNull(recNum);
                assertNotNull(secretKey);
                assertNotNull(msgContent);
                assertNotNull(send);
                assertNotNull(cancel);
}

        @SmallTest
public void testViewVisible() {
                super.assertEquals("Receiver Number is not empty", "",
recNum.getText().toString());
                super.assertEquals("Secret Key is not empty", "",
secretKey.getText().toString());
                super.assertEquals("Message content is not empty", "",
msgContent.getText().toString());

}
```

5. CONCLUSIONS

This paper describes an approach to testing the SMS message in Android apps based on the OCL and UML model. While several automated testing frameworks have been proposed and developed for Android and mobile platforms, developers still need formal approaches and corresponding tools to generate test cases for conformance testing efficiently and effectively. To

address this issue, we have proposed a novel framework to enable rigorous conformance testing for the Android platform. Our frame work adopted a model-based approach which utilizes formal techniques – OCL– to automatically generate test cases. In addition, we have demonstrated the feasibility of our approach with Android SMS application. As part of our future work, we would explore an approach for directly constructing model from the requirements, leveraging the capability of formal verification approach. Moreover, we would apply our approach to other Android modules.

ACKNOWLEDGEMENTS

We would like to thank for Lockheed Martin and the Deans office for the support of this project. This project is also partially supported by NSF sub award DUE-1202690, NSF award DGE-1419295, DUE-1225654, and DUE-1525414, and Air Force Research Lab award FA8750-15-2-0106. We would like to thanks for all valuable reviews.

REFERENCES

[1] A. Medani, A. Gani1, O. Zakaria, A. A. Zaidan and B. B. Zaidan. Review of mobile short message service securityissues and techniques towards the solution. Scientific Research and Essays, 6:1147–1165, March.

[2] D. Amalfitano, A. Fasolino, and P. Tramontana. A GUI crawling-based technique for android mobileapplication testing. In Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEEFourth International Conference on, pages 252–261, March 2011.

[3] M. Battey and A. Parakh. An efficient quasigroup block cipher. International Journal of WirelessPersonal Communication, 73(1):63–76, 2013.

[4] D. E. Bell and L. J. LaPadula. Secure computer systems: mathematical foundations. Research Report2547, MITRE, 1973.

[5] Y. Bertot, P. Castran, G. i. Huet, and C. Paulin-Mohring. Interactive theorem proving and program development: Coq'Art : the calculus of inductive constructions. Texts in theoretical computer science.Springer, Berlin, New York, 2004.

[6] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A.-R. Sadeghi, and B. Shastry. Towards taming privilegeescalationattacks on android. In 19th Annual Network & Distributed System Security Symposium(NDSS), volume 17, pages 18–25, 2012.

[7] E. D. Chris Wysopal, Lucas Nelson and D. D. Zovi. The Art of Software Security Testing: IdentifyingSoftware Security Flaws (Google eBook). Addison-Wesley Professional;, 1e edition, November 2006.

[8] M. Cristi. Formal verification of an extension of a secure, compatible UNIX file system. PhD thesis,2002.

[9] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner. A survey of mobile malware in the wild.In Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices,SPSM '11, pages 3–14, New York, NY, USA, 2011. ACM.

[10] D. Fisher. Researchers find methods for bypassing googles bouncer android security. Availablefrom https://threatpost.com/enus/blogs/researchers-find-methods-bypassing-googles-bouncer-androidsecurity-060412, June 2012.

[11] D.Halliday.Security alert: Cleanedout. Available from https://blog.lookout.com/blog/2013/02/07/security-alert-cleanedout, Febuary 2013.

[12] M. Hassinen and K. Hypponen. Strong mobile authentication. In Wireless Communication Systems,2005. 2nd International Symposium on, pages 96–100, Sept 2005.

[13] J. Juerjens. Secure Systems Development with UML. SpringerVerlag, 2003.

[14] J. J¨urjens. Sound methods and effective tools for model-based security engineering with uml. InICSE, pages 322–331, 2005.

[15] J. J¨urjens and P. Shabalin. Tools for secure systems development with UML. International Journal onSoftware Tools for Technology Transfer, 9(5-6):527–544, 2007.

[16] M. Lacter. Stunning stat: 91% of U.S. adults own cell phones. Available from: http://www.laobserved.com/biz/2013/06/stunning stat 91 of.php, June 2013.

[17] A. M. Memon, I. Banerjee, and A. Nagarajan. GUI ripping: Reverse engineering of graphical userinterfaces for testing. In Proceedings of The 10th Working Conference on Reverse Engineering, Nov.2003.

[18] A. M. Memon and Q. Xie. Studying the fault-detection effectiveness of GUI test cases for rapidlyevolving software. IEEE Trans. Softw. Eng., 31(10):884–896, 2005.

[19] P. Miller. Fuzz project web page. Available from: http://pages.cs.wisc.edu/ bart/fuzz/.

[20] W. Nyamtiga, A. Sam, and L. S. Laizer. Security perspectives for USSD versus SMS in conductingmobile transactions: A case study of tanzania. International Journal of Technology Enhancements andEmerging Engineering Research, 2013.

[21] R. R. Priyanka Pimpale and S. Upadhyay. Modifications to aes algorithm for complex encryption.International Journal of Computer Science and Network Security, 11(10):183–186, October 2011.

[22] T. Project. Coq proof assistant, 2009.

[23] S.-H. Seo, K. Yim, and I. You. Mobile malware threats and defenses for homeland security. InG. Quirchmayr, J. Basl, I. You, L. Xu, and E. Weippl, editors, Multidisciplinary Research and Practicefor Information Systems, volume 7465 of Lecture Notes in Computer Science, pages 516–524. SpringerBerlin Heidelberg, 2012.

[24] M. Szydlowski, M. Egele, C. Kruegel, and G. Vigna. Challenges for dynamic analysis of iOS applications.

[25] In Proceedings of the 2011 IFIP WG 11.4 International Conference on Open Problems in NetworkSecurity, iNetSec'11, pages 65–77, Berlin, Heidelberg, 2012. Springer-Verlag.

[26] T. Takala, M. Katara, and J. Harty. Experiences of system-level model-based gui testing of an androidapplication. In Software Testing, Verification and Validation (ICST), 2011 IEEE Fourth InternationalConference on, pages 377–386, March 2011.

[27] Takanen, J. DeMott, and C. Miller. Fuzzing for Software Security Testing and Quality Assurance.Artech House, Inc., Norwood, MA, USA, 1 edition, 2008.

[28] S. Thurm and Y. I. Kane. Apps are caught stealing privatedata stored on smartphones (android and iOS).Available from: http://online.wsj.com/article/SB10001424052748704694004576020083703574602.html.

[29] Q. Wu, Y. Semenova, P. Wang, and G. Farrell. High sensitivity sms fiber structure basedrefractometer–analysis and experiment. Opt. Express, 19(9):7937–7944, Apr 2011.

[30] A. Zheng, S. Zhu, S. Dai, G. Gu, X. Gong, X. Han, and W. Zou. Smartdroid: An automatic systemfor revealing ui-based trigger conditions in android applications. In Proceedings of the Second ACMWorkshop on Security and Privacy in Smartphones and Mobile Devices, SPSM '12, pages 93–104, NewYork, NY, USA, 2012. ACM.

[31] W. Zhou, Y. Zhou, X. Jiang, and P. Ning. Detecting repackaged smartphone applications in third-partyandroid marketplaces. In Proceedings of the Second ACM Conference on Data and Application Securityand Privacy, CODASPY '12, pages 317–326, New York, NY, USA, 2012. ACM.

[32] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang. Hey, you, get off of my market: Detecting malicious appsin official and alternative android markets. In Proceedings of the 19th Annual Network and DistributedSystem Security Symposium, pages 5–8, 2012.

[33] W. Choolsip, Y. Fu. A Case Study Of malware Detection and Removal in Android Apps. International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol. 4, No.2, April 2014.

AUTHOR BIOGRAPHY

Dr.Yujian Fu is an associate professor at department of Electrical Engineering and Computer Science at Alabama A&M University. Her research interests fall in formal verification, mobile security, cyber physical system and real-time embedded system design and verification. Dr. Fu's projects are supported by NSF and AFRL.

Mr. WichienChoolsip is a graduate student at department of Electrical Engineering and Computer Science at Alabama A&M University. His research focuses on the mobile security design and applications.