ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 7 July, 2014 Page No. 6876-6879

Mobile Application Protection System: A Secured Architecture

B Meenakumari[1] ,Sandeep Karnam[2]

[1]M.Tech, Department of Computer Science and Engineering,
SET-JU, Jain Global Campus, Jakkasandra, Kanakpura(T), Ramanagara(D),
Karnataka, bmeenareddy7@gmail.com, 562112

[2]Assistant Professor, Department of Computer Science and Engineering,
SET-JU, Jain Global Campus, Jakkasandra, Kanakpura(T), Ramanagara(D),
Karnataka, karnam.sandeep@gmail.com, 562112

Abstract -***Mobile application market is gaining a lot of attention with the introduction of increase in speed, accessibility and capacity to download huge amount of software applications from internet to the mobile devices. Mobile application protection is a critical issue for mobile network operators, content providers and all players who involved in the mobile software business chain. This paper proposes an architecture where trusted mobile software environment is designed to control the execution of mobile application. A software ID which is dynamically updated and shared key are created and used for authorization of mobile application execution requests. This solution can prevent many mobile application crack issues which include copy of the mobile application to unauthorized mobile devices and modification of mobile application.***

Keywords –Mobile Application, User Equipment, Application Store, Open ID.

I. Introduction

Mobile computing is a significant contributor to the pervasiveness of computing resources in modern western civilization. In concert with the proliferation of stationary and embedded computer technology throughout society, mobile devices such as cell phones and other handheld or wearable computing technologies have created a state of ubiquitous and pervasive computing where we are surrounded by more computational devices than people. Mobile computing is all about portable and small computers, which includes PDAs (Personal Digital Assistants) like mobile phones, palmtops, laptops etc. In this growing technological world, people are much bound to work on computers. Mobile computing can be defined as the ability to use technology that is not physically connected to any static network. Nowadays, most laptops and personal digital assistants (PDA) all have wireless cards or Bluetooth interfaces built into them for very Good mobile internet access.

Mobile Computing is an emerging technology with fast improvement in capacity of mobile devices and deployment of mobile communication 3G/4G networks. When we consider security issues about mobile computing a lot of things will be noticed such as authentication, confidentiality, application protection, data security and etc. With this type of issues mobile users are facing a lot of problems. Now a day, mobile applications have become the backbone of the mobile communication system and mobile devices have open software platforms which provide capability of downloading a huge variety of applications from internet .thus mobile application product market is gaining considerable attention. Copyright attacks on mobile applications are becoming dangerous situation which impacts both over-the-top (OTTs) corporations and mobile network operators. Hence mobile application software protection is focused by industry.

In this paper, we present a solution for mobile application protection system that establishes a safe environment which includes a set of technologies**.**A trusted mobile software environment is designed to control the execution of the mobile application. A dynamic Software ID and security key are created and deployed on both the application store and mobile device for authorization of software execution requests. The Software ID will be updated each time the mobile application is executed and will be stored on the server side. Mobile application needs to be downloaded over a mobile network and

run in memory and power-limited mobile device. Therefore, mobile application is required to be smaller and power-efficient, which are characteristics not supported by existing software protection solutions. Mobile network operators can be deeply involved in software protection because their infrastructure is usually built on 3G security architecture.

II. Previous Work

There are many tools that are readily available on the internet which can be used to crack the mobile software. Breaching the mobile software's security is easy and causes distribution of mobile application illegally. All mobile application protection techniques have been cracked within weeks of their market launch. Software piracy, reverse engineering, tampering are three types of software copyright attacks. Software watermarking, code obfuscation, tamper-resistance and runtime execution limits are technologies which are designed to prevent these copyright attacks.

Software watermarking is a Software theft, also known as software piracy, is the act of copying a legitimate application and illegally distributing that software, either free or for profit. Software watermarking involves embedding a unique identifier within a piece of software, to discourage software theft. Watermarking does not prevent theft but instead discourages software thieves by providing a means to identify the owner of a piece of software and/or the origin of the stolen software. The most difficult problem to solve is keeping the watermark hidden from attackers while, at the same time, allowing the software owner to efficiently extract the watermark when needed. Code obfuscation is obfuscated code that is difficult for humans to understand. Programmers obfuscate code to hide its purpose or its logic. It is a promising defense technology which secures software in a way that makes cost of reverse engineering high. There are Obfuscation methods such as lexical transformations, control transformation, data transformation, anti-disassembly, anti-debugging.

There are many software protection technologies are there such as DRM trust model but this model is based on a public key infrastructure.so agents have to get a unique private or public key pair and a certificate which are high requirements for a mobile device and there are many inexpensive copy protection removal software are there and to solve these issues, a trusted computing environment is needed. Mobile application should be smaller, power efficient and run in memory and power limited mobile devices.in the proposed solution, we use the architecture to enable user equipment authentication and key sharing between the user equipment and the server.

III. Proposed Work

We present a solution that uses a 3G architecture and key agreement and the Open ID integration mechanism to authenticate the user equipment and Application Store(AS) and then creates a security association for user equipment, AS and the mobile application. To achieve this security process, a dynamic software ID is configured on both the server side and client side and this software ID is used to authenticate the application when user triggers it. A trusted mobile software environment is designed to avoid cracking issues.

1. ARCHITECTURE

Figure1 shows architecture of mobile application protection system which contains client and server .Here client is user equipment and Application Store is server and Open ID provider provides shared key for client and server.

Mobile Software environment function in user equipment is that it calculates the message digest with the software ID and software package and then send it to the application store.it receives an instruction from the application store to run or not to run the user triggered software. A record of software ID and shared key is maintained at client side. Message digest authentication process will be done at server side and sends instructions to the client to run or not to run an application.

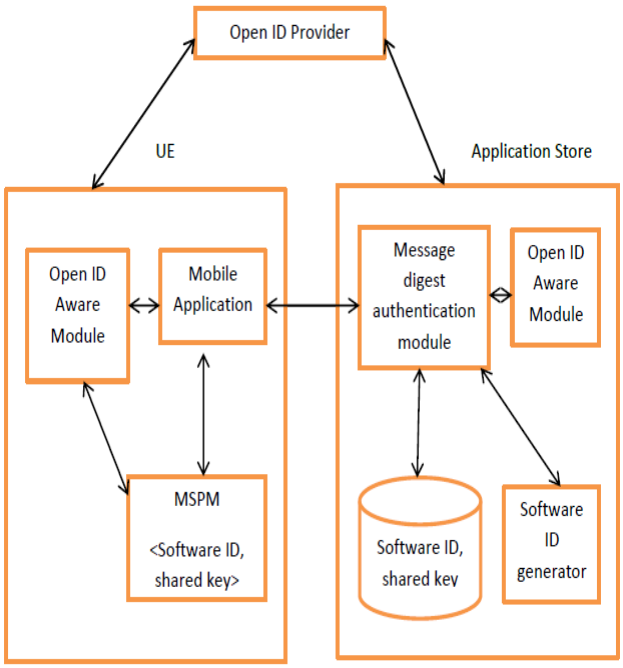


Figure 1: Architecture of Solution

IV. Implementation

Mobile Application Download:

When users try to download mobile application from application store, authentication process will be done and shared key will be generated and it will be stored at both client and server side.Application store produces a software ID based on shared key and it sends the application package and software ID to user equipment. Client stores a copy of software ID in the database.

Procedure for running mobile application:

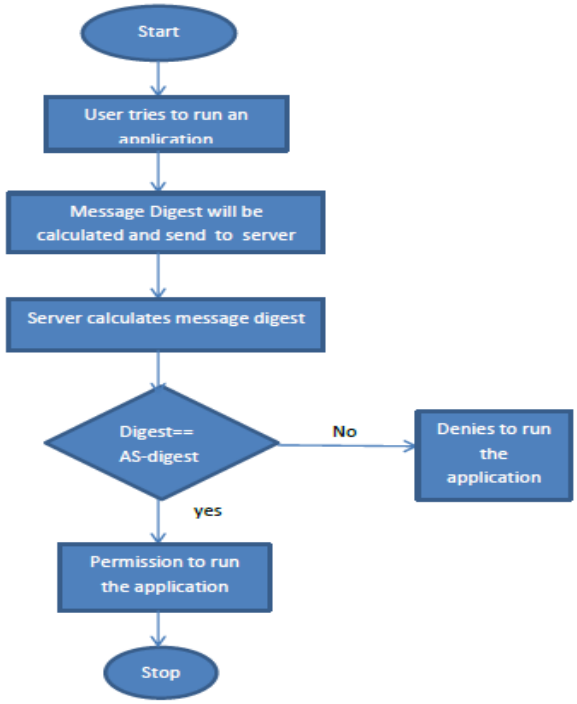
When users try to run the application, message digest will be calculated at client side and sends < digest, software ID> to the application store for authentication.AS checks for matching shared key based on received software ID and calculates message digest and compares both digest values. If digest value is equal, it sends an instruction to the client to run the application otherwise, sends an instruction to the client not to run the application.

Process Flow:

The user tries to run the mobile application, mobile software runtime environment gets the software ID according to the application and then it tries to retrieve a record of < Software ID ,shared key> in the database. If a record is not found, request for mobile application run is denied. If there is a record, message digest will be calculated and <digest, software ID > will be sent to the application store.

Application Store checks for matching shared key based on received software ID. If it is not found, then it will send an instruction not to run the mobile application. If shared key is found, Application Store calculates message digest and compares both digest values. If digest value is equal, it sends an instruction to the client to run the application otherwise, sends an instruction to the client not to run the application.

Flowchart



Analysis:

Mobile application should be downloaded over a mobile network and run in memory and power-limited mobile device. Therefore, mobile application is required to be smaller and power-efficient, which are characteristics not supported by existing software protection solutions. Mobile network operators can be deeply involved in software Protection because their infrastructure is usually built on 3G security architecture.

Here we present a solution for mobile application protection system that establishes a safe environment which includes a set of technologies**.**A trusted mobile software environment is designed to control the execution of the mobile application.Mobile Software environment function in user equipment is that it calculates the message digest with the software ID and software package and then send it to the application store.

It receives an instruction from the application store to run or not to run the user triggered software. A record of software ID and shared key is maintained at client side. Message digest authentication process will be done at server side and sends instructions to the client to run or not to run an application.

Copy of the application to unauthorized devices will be avoided .If user copy an android application to another mobile, it can't be run in another mobile because a record of<Software ID, shared key> will not be found and software ID is initiated when user download an mobile application from application store.

We can prevent modification of the mobile application with this procedure. If cracker tries to modify the mobile application such as translating it into other language, he can't run the application .when cracker runs the application ,he can get a record of<software ID, shared key> but message digest authentication will not be done .So he can't run the mobile application. The Application Store will send a permit instruction to terminate the run operation.

V. Conclusion

In this paper, we presented a secured architecture for mobile application protection system that integrates 3G architecture and key agreement and Open ID provider technologies to provide secure association between application store and the user equipment and this solution ensures that mobile application can only be downloaded through a mobile device and then run only on it. This procedure prevents distribution of mobile application illegally and running of unauthenticated software on a mobile device.

References

1.C. S. Collberg and C. Thomborson, "Watermarking, tamper-proofing, and obfuscation - tools for software protection," *Software Engineering.*

2. J.-q. Zhu, “The key problem research on softerware protect,” Ph.D. dissertation, Journal o f Jilin University, 2009.

3. M. P. Paul England, John L. Manferdelli, “software application protection by way of a digital rights management (drm)system,” Patent US 7 680 743 B2, 2010.

4. jimgilmour1. (2009) How to protect wrt widgets with omadrm 1.0.[Online]. Available: http://www.developer.nokia.com/Community/Wiki/ How to protect WRT widgets with OMA DRM 1.0.

5. M. Karnick, J. MacBride, S. McGinnis, Y. Tang, and R. Ramachandran, “A qualitative analysis of java obfuscation.”

M. P. Paul England, John L. Manferdelli, “software application protection

by way of a digital rights management (drm)system,” Patent US

7 680 743 B2, 2010.

6 jimgilmour1.(2009) How to protect wrt widgets with omadrm 1.0.

[Online]. Available: http://www.developer.nokia.com/Community/Wiki/

How to protect WRT widgets with OMA DRM 1.0

[7] S. O. Hwang, “How viable is digital rights management?” *Computer*,

vol. 42, no. 4, pp. 28 –34, 2009.

[8] C. staff. (2009) Drm removal. [Online]. Available: www itunesm4pconverter.com/drm-removal/

A. Majumdar, S. J. Drape, and C. D. Thomborson, “Slicing

obfuscations: design, correctness, and evaluation,” in *Proceedings of the 2007 ACM workshop on Digital Rights Management*, ser.

DRM ’07. New York, NY, USA: ACM, 2007, pp. 70–81. [Online].

Available: http://doi.acm.org/10.1145/1314276.1314290

[9] M. Karnick, J. MacBride, S. McGinnis, Y. Tang, and R. Ramachandran,

“A qualitatie analysis of java obfuscation.”

[10] “3g security; security architecture,” 3rd Generation Partnership Project,

2011. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/33

series/33.102/33102-b00.zip

[11] H.-L. L. Igor Faynberg, “Identity management services provided by network operator,” Patent US 2010/0 262 703 A1, 2009.