**American Journal of Mobile Systems, Applications and Services**
Vol. 1, No.1, 2015, pp. 54-58
http://www.aiscience.org/journal/ajmsas

# Construction of Trusted Computing Platform Based on Android System

**Hui Zhang, Defeng Sun, Yanli Xiao, Lei Peng***

College of Information Engineering, Taishan Medical University, Taian, China

## Abstract

With the widespread use of Android mobile phones, the problems of security of phone become increasingly prominent. The Java technology architecture for trusted computing is a trend to solve the above problems. Through the analysis of the current Java platform trusted computing architecture and the security of Android operating system, the trusted platform architecture based on Android and Java is proposed. The mobile trusted modules are implemented by combining the trusted chip and related software such as protocol and algorithm, trusted chip API, computing library API, and computing function library. In this architecture, by analyzing the feasibility of each module, the mobile platform to be verified is trusted.

## Keywords

Trusted Computing, Android, Mobile Terminal, Mobile Trusted Module

Received: June 25, 2015 / Accepted: July 5, 2015 / Published online: July 22, 2015



## 1. Introduction

With the widespread use of smart phones, the problems of security of mobile phone become increasingly prominent. Viruses that appear on the computer also appear in the mobile phone. The information in the phone is mostly personal privacy. Once stolen more dangerous. These make people more and more worry about the safety of mobile phones. Android is an open source system based on Linux kernel. The openness and freedom played an important role for its popularity and promotion, but also bring a security risk. Because the source code of Android can easily be got, the security risks can easily be found and exploited. In order to prevent security problems for Android, it is necessary to analyze and study the security system of Android. It need to analysis of the internal security mechanism and the secure communication between each module. Then find the security risks of Android. Security risks refer to the defects and deficiencies of the implementation in hardware, software and protocols, and the system security policy. From analysis and studies of the security of Android, it can be drawn that now the problem of security is: rights abuse result in malicious software is rampant, user's privacy is stolen and phone charges are absorbed. However, there are seldom harms of virus and attack for Android system itself. [1,2]

From the current software technology point of view, it is a tendency through Java or similar software architecture to achieve trusted computing technology. Today, almost every mobile phone operating systems support JVM (Java Virtual Machine). Users can run the same application on different hardware platforms by Java. Traditional software runs directly on the operating system. Thus first the trust of the operating system is must be verified, then the running platform of application software is considered to be verified trusted. However the operating system verification involves modifications of BootLoader, which is difficult for current mobile terminals. Java applications are running on JVM. JVM is verified to be trusted, and then the running environment of the Java applications can be considered to be trusted. The difficulty of verifying the trust of the platform is reduced. It becomes possible to modify the current mobile terminal and make it to be the trusted computing terminal. [3,4]

* Corresponding author
E-mail address: pengleisd@163.com (Lei Peng)

To solve these problems, in this paper proposed construction of trusted computing platform based on Android system. The trusted computing platform is based on MTM (Mobile Trusted Modules). Without losing the advantages of Java, it can meet the requirements of trusted computing, and can be implemented in a mobile terminal. More critical is that this platform can be constructed without changing the current architecture, and make the mobile terminal to be partially trusted computing platform. To build a trusted computing platform based on Android allows most current mobile terminal platform can become trusted platform.

## 2. Related Work

Debbali et al. [5] detailed analyzed the security of the architecture of J2ME (Java 2 Micro Edition) CLDC (Connected Limited Device Configuration) in Java mobile terminal. Debbali et al. compared J2ME and J2SE (Java 2 Standard Edition) which is proved security architecture. Respectively, from access control, security policy, storage, protection and other aspects of the J2ME architecture they carried out a detailed study, and had drawn the conclusion that J2ME needs improvement in the aspects of access control and memory protection. Meanwhile, Debbali et al. pointed out that the security defect of the J2ME architecture can be compensated by increasing security expansion modules, so that to improve the security of J2ME. However, Debbali et al. did not give a specific implementation. For the security of the J2ME architecture, Dietrich [6] presented the trusted computing architecture based on J2ME embedded devices. Dietrich first proposed the trusted computing architecture based on Java, and then give this architecture implementation of each module, in order to confirm that this architecture can be realized. According to Dietrich's conclusion, realization of trusted computing in J2ME is feasible. However, Dietrich did not implement the overall architecture. Further, since in this architecture the local C program is used to complete the trusted computing, including calculation and protocol etc., which involved all the functions and application programming interfaces. However, due to security reasons, J2ME did not give the references of local C program. Such architecture encountered a technical bottleneck. Dietrich used the JavaCard technology to solve this problem, but did not give a specific implementation. Bichsel et al. [7] gave the anonymous authentication implemented in the JavaCard. However, the JavaCard technology will inevitably lead to changes to the current mobile terminal architecture.

On the other hand, due to the presence of Java technology cannot solve bottlenecks in the code execution speed. The researchers of Java trusted computing are constantly optimizing current certification programs to accelerate the verification speed. Then that existing certification scheme can be applied in Java. For this reason, according to DAA (Direct Anonymous Attestation) [8] presented by Brickel, Camenisch and Chen, and the standard proposed by international TCG (Trusted Computing Group), Dietrich performed DAA performance evaluation for different embedded devices through the Java technology [9]. To accelerate the speed of anonymous authentication, Wachsmann et al. [10] on the basis of DAA proposed lightweight anonymous authentication scheme further, and realized in Java. The speed of this scheme is up to about 115ms, but a large number of calculations were given to the C program to execute. However the C program was unable to get transplants in J2ME architecture.

## 3. Trusted Computing of Java Platform

Java platform proposed two architectures for PC and mobile terminal, J2SE and J2ME architecture. J2SE architecture for the PC terminal covers all operations on the terminal, that is, developers can implement all control of the computer by J2SE. Then from feasibility perspective, J2SE architecture can achieve the requirements of trusted computing. In addition, The J2SE architecture has been proven to be a safe architecture [5]. However, J2SE is architecture for PC terminal, not for embedded mobile terminals. In this regard, Java platform proposed J2ME architecture for embedded mobile terminal. J2ME can be considered as a subset of J2SE. J2ME architecture allows executing Java code in a mobile terminal without modifying the core code. However, for the reasons of performance and security, J2ME architecture is cropped part of J2SE function, which affects the construction of trusted computing platform as follows:

(1) Without JNI (Java Native Interface). Due to the defect of execution speed, Java language is not suit for apply to calculation. To solve this problem, Java platform proposed JNI technology that allows Java to perform C program. Developers can place the calculation functions in the C program. Java can directly call the functions to improve the execution speed of the code. However, the J2ME architecture does not support JNI technology. Thus a large number of computing tasks during trusted computing platform certification process must be directly implemented in Java layer. According to Dietrich's test [9], the average time for DAA certification is about 20s in Java layer, while it is 115ms by C program, Increased nearly 200 times.

(2) Do not allow users to customize an existing class. Dynamic authentication protocol must be implemented by custom class or by covering, removing, reconfiguring existing class. However J2ME does not support custom

class. Developers must call API (Application Programing Interface) provided by J2ME to implement limited functions. Undoubtedly, these affect to transplant relevant protocol of architecture the trusted computing to J2ME.

The architecture proposed by Dietrich partial solved the above problems. Dietrich directly placed C program in the operating system layer, and made SATA (Security and trust services API) interact with the C program by SmartCard. At the same time, the custom protocols are placed in MTM abstraction layer, and are transplanted to the J2ME architecture. However, the introduction of SmartCard will inevitably lead to changes in the existing mobile terminal architecture. That is to say, the mobile terminal must access SmartCard to achieve the above architecture. The mobile terminal can only access via miniUSB interface or TD SmartCard slot. For security reasons, Operating system of mobile terminals strictly limited the access devices for USB devices and TD card. Implementation of MTM abstraction layer is not clear. On one hand, for the reason of computing speed, MTM abstraction layer cannot implement by Java, on the other hand, for the reason of without JNI technology, it cannot implement by C program. Therefore, although Kurt Dietrich's architecture seems to be realizable, but there are some difficulties in the implementation.

In addition, Java, such simple and practical software technology platform, does not provide MTM. The applications running on the Java platform neither have the correlation function call interface in TSS (Trusted Software Stack), nor have a viable technical solution for interacting with MTM [6]. This problem can be solved through software extensions. Reference TCG (Trusted Computing Group) standards on MTM and TSS, MTM and TSS can be achieved through software or external hardware approach.

# 4. Security of Android System

The introduction of Android mobile operating system makes it become a reality that completely using Java technology achieves the running of applications in mobile terminal. In the Android mobile operating system, the applications completely run in the JVM namely Dalvik. The defect of Java code execution speed low can be solved by Android NDK (Native Development Kit) technical. Android operating system allows many problems in current Java platform trusted computing architecture can be resolved technically.

Android architecture contrast with J2SE and J2ME is shown in Figure 1. As can be seen in Figure 1, the architecture frameworks of Android and J2SE are very similar. Each module in J2SE has a corresponding module in Android. In addition, Android operating system has a user-defined class loading module and a native application interface calling module, which J2ME system does not have. Therefor Android allows developers to customize classes and call the local C program by NDK technology. [11] Thus the bottleneck of trusted computing in J2ME can be broken in Android. However, the reason why J2ME remove the two modules is for security reasons.

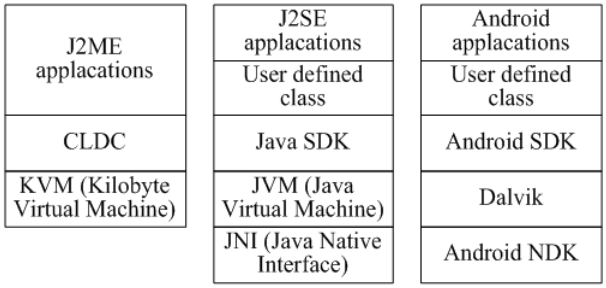


**Figure 1.** J2ME, J2SE, and Android architecture comparison and analysis.

## 4.1. Security of Android Custom Class

In Android system, in addition to the definition of internal classes, developers can cover, remove, or reconfigure all classes. The system internal classes, which cannot be modified, are defined by Android's C / C ++ layer. The classes, which can be modified, are defined by Android's Java layer. With this, on the basis of without destroying the hardware modules, Android maximize allows users to modify the class. Consider from the trusted computing platform, it does not aim to destroy the hardware modules to modify the class. Therefore, such structure meets the needs of the construction of trusted computing platform.

## 4.2. Security of Android NDK

In the J2SE structure, The C program of JNI can complete any operation on terminal. The introduction of JNI is bound to introduce hardware platform security risks, and therefore does not introduce JNI technology to J2ME. In the Android NDK allows developers to execute local C program. However, Android also restricts the local C program. Local C program cannot directly operate the hardware, only to meet the computing requirements. We try to transplant libUSB technology makes Android operating system can call USB devices locally, but without success. In the trusted computing platform, the intention of introducing the C program is to speed up code execution speed, not to operate the hardware. Then under the premise of meet the safety conditions, Android NDK can satisfy the requirements of trusted platform architecture.

## 4.3. Security of Android External Module Access

In Android 3.1 and above, the USB operation classes

(Android.hardwear.usb) are introduced. The developers are allowed to operate USB devices. The USB operation classes make it become possible to operate the mobile terminal external devices in Java. However, for security reasons, there are strict restrictions to access USB devices in Android. The parameters ProductID and VentoriID of the USB devices must be explicitly declared firstly in the application of operating external USB devices. Before operating the USB devices, authorization of allowing users operating the USB must be achieved.

In summary, under the condition of meeting the security, the custom class modules and the local function interface calling modules, which the J2ME are not available, are introduce in Android system. That makes it possible to build a trusted computing platform in Android.

# 5. Trusted Platform Based on Android

According to Android operating system security analysis, we can build a trusted computing platform architecture based on Android. Android trusted computing platform architecture are shown in Figure 2. The platform consists of seven parts, trusted computing applications, protocol and algorithm, trusted chip API, computing library API, computing function library, USB interface, and trusted chip.

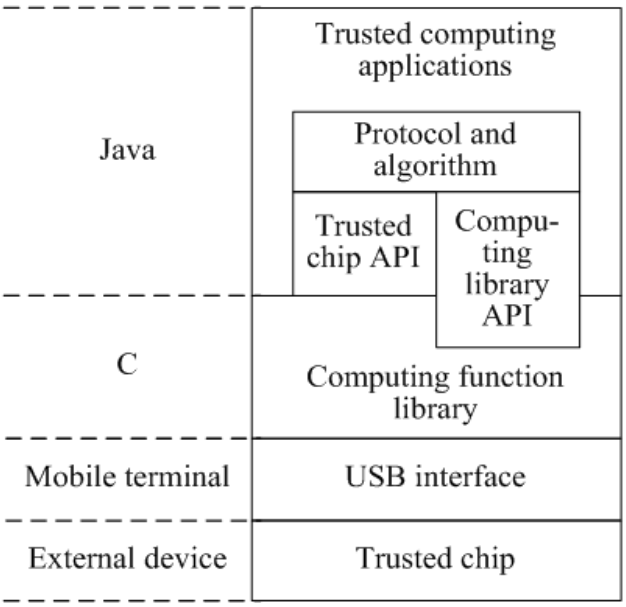


**Figure 2.** Android trusted computing platform architecture.

Top of the architecture is the trusted computing applications. They can call each library function provided by TSS and TPM to provide users the trusted computing services. The protocol and algorithm are developed by Java. They call the packages trusted chip API and computing library API, and provide protocol and algorithm support for the trusted computing applications. The trusted chip API, which is the packaged Java code, is the trusted chip application program interfaces. It encapsulates the trusted chip-related functions, such as key generation, key management, cryptographic algorithms, etc., for the user to call. The computing library API, which is developed by Java and C, packages the native C program to calculate the correlation function for users to call. The computing function library use Android NDK technology writing the consuming calculation functions to native applications and provide the computing function to developers. The trusted chips are external devices. They connected with the mobile terminal via USB, and perform related operations about trusted chip, through USB operating classes provided by Android system.

## 5.1. Mobile Trusted Module

First question to be considered is how to achieve mobile trusted module. TCG's standard does not give a specific implementation method, but described the necessary characteristics of MTM. The purpose is to enable designers to achieve more flexibility for implementing MTM. Taking into account the characteristics of current mobile terminals, the approach of implementing trusted module by modifying the hardware architecture is not feasible. It is a viable option to implement MTM by current software library or trusted chip. The first method is to implement MTM through pure software. It is obvious that the advantages of this approach are to implement the MTM without doing anything to the hardware, and the higher portability and modifiability. The shortcoming is that the trusted verification of MTM software module itself is difficult. However, today's embedded processors provide a safe zone, such as the ARM platform TrustZone. To store the MTM software modules to the safe zone can guarantee the MTM software modules not to be maliciously modified and destructed, so ensure the safety of MTM. In addition, all libraries are stored as dynamic-link library files (so files) in Android system, so the trusted verification of software library functions can be completed by checking the check value of the so files. That approach is realizable. Another method is that the MTM can be realized by the trusted chip. The trusted chip is responsible for providing protection fields and hidden fields, to protect itself from destruction and malicious modification, while providing the necessary trusted authentication service to upper layer. Android 3.1 or later edition already provides support for USB-Host. Any mobile terminal with Android 3.1 or later edition can perform USB operation by host identity through USB OTG (On-The-Go) mode. This makes the external trusted chip MTM possible. On consider of the trusted chip cannot meet the needs of trusted authentication protocol, finally the MTM is achieved by the form of combining the chip and software.

### 5.2. MTM Abstract Interface

The TCG's standard did not give a specific implementation for how to call MTM. The MTM abstract interface layer proposed by Kurt Dietrich [6] provides the service of packaging MTM for TSS to call it. However, it did not give specific implementation of the MTM abstract interface layer. Since the computational complexity, MTM should be placed in the C language layer, or placed in the trusted chip in order to improve code execution speed. The computing library API is packaged in the C language layer, and the trusted chip API is packaged in trusted chip layer.

## 6. Conclusion

In this paper, proposed the trusted platform architecture based on Android and Java, and analyzed the feasibility of each module. In this framework, after verification the mobile platform can be considered to be trusted. Further, the prior art solutions to meet the technical requirements have been trusted computing platform. Further, the existing technology can meet the requirements of the trusted computing platform. As the mobile terminal operating system, Android is constructed as trusted computing platform by the way of adding external TPM module and transplant trusted computing library. Now the trusted chip API, the computing library API and the computing function library have been implemented. In the next step, based on this framework, design and assessment of existing protocols, further propose embedded terminal authentication and access protocol based on trusted platform.

## References

[1] Bugiel S, Davi L, Dmitrienko A, et al. Poster: the quest for security against privilege escalation attacks on android[C]//Proceedings of the 18th ACM conference on Computer and communications security. ACM, 2011: 741-744.

[2] Enck W, Ongtang M, Mcdaniel P. Mitigating Android software misuse before it happens[J]. 2008.

[3] Barrera D, Kayacik H G, van Oorschot P C, et al. A methodology for empirical analysis of permission-based security models and its application to android[C]//Proceedings of the 17th ACM conference on Computer and communications security. ACM, 2010: 73-84.

[4] Fuchs A P, Chaudhuri A, Foster J S. Scandroid: Automated security certification of android applications[J]. Manuscript, Univ. of Maryland, http://www. cs. umd. edu/avik/projects/scandroidascaa, 2009, 2(3).

[5] Debbabi M, Saleh M, Talhi C, et al. Security Evaluation of J2ME CLDC Embedded Java Platform[J]. Journal of Object Technology, 2006, 5(2): 125-154.

[6] Dietrich K. An integrated architecture for trusted computing for java enabled embedded devices[C]//Proceedings of the 2007 ACM workshop on Scalable trusted computing. ACM, 2007: 2-6.

[7] Bichsel P, Camenisch J, Groß T, et al. Anonymous credentials on a standard java card[C]//Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009: 600-610.

[8] Brickell E, Camenisch J, Chen L. Direct anonymous attestation[C]//Proceedings of the 11th ACM conference on Computer and communications security. ACM, 2004: 132-145.

[9] Dietrich K. Anonymous credentials for java enabled platforms: a performance evaluation[M]//Trusted Systems. Springer Berlin Heidelberg, 2010: 88-103.

[10] Wachsmann C, Chen L, Dietrich K, et al. Lightweight anonymous authentication with TLS and DAA for embedded mobile devices[M]//Information Security. Springer Berlin Heidelberg, 2011: 84-98.

[11] Wering Liu, Jianwei Liu. Android OS trusted computing platform architecture[J]. Wuhan University: natural science edition, 2013, 59 (2): 159-164.