Improved Key Distributed Storage Mechanism Based on Android System

Yuxi Lin and Xingjun Wang
Department of Electronic Engineering, Graduate School at Shenzhen, Tsinghua University, China
Email: {linyuxide, wangxingjun2015}@163.com

***Abstract*—Mobile devices are playing a more and more important role in people's daily lives, mobile payment, watching video on mobile is increasingly common. Android system has occupied more than eighty percent of the mobile devices. Copyright protection of products and privacy protection of Android users is becoming a matter of growing concern, Researchers continue to improve the security of Android, and Digital Right Management (DRM) is an effective program, which private key protection is the most important part. This paper presents an improved key distributed storage solution to protect the private key based on Android system.**

***Index Terms*—private key, distributed storage, DRM, Android, security**

I. Introduction

The security of Android system is major in two parts: the security of the system and the security of the applications running on it, people has proposed a variety of techniques to improve the security of Android system and application, among them, Trusted Computing (TC) and DRM are two very representative solution, Trusted computing uses a trusted computing platform based on hardware security module. DRM protects the copyright of digital content, preventing from illegal copying of digital content, which means only authorized users can use digital content.

Specifications of DRM have been launched these years, such as Marlin DRM [1], OMA DRM 2.0 [2], Widevine DRM and China DRM (CDRM) [3], etc. Many of them have got a wide range of use. However, none of the existing solutions are perfect in the face of growing security challenges. Researchers have put forward their DRM security scheme to address different security challenges. Xie Lanchi has proposed a centralized home network content protection system in [4]. Yao Zhang incorporated DRM mechanism into P2P network architecture to put forward a scheme to ensure efficient content sharing and media protection in [5].

DRM works mainly as follows: first, a digital content authorization center is built, and then, a digital content will be encrypted and packaged, an encrypted and packaged digital content has its own ID and encryption key. Third, a user who wants to access the digital content will have to apply for authorization center and purchase the content authority, the authority information includes decryption key and use restrictions, etc. Only after acquiring the authority can the user decrypt and use the digital content.

Private Key is the most important factor in DRM to protect the communication between user and server, if private key is stolen by hacker, the security of the communication is lost. So, the way to protect the safety of private key is what we focus on in this paper.

This paper is structured as follows: the next section describes some technology that our work concerns, the third section proposes our solutions, and the last section with conclusions.

Manuscript received September 9, 2015; revised November 27, 2015.

II. Related Work

In this section we will first introduce the establishment of security channel based on China DRM Technical specification. And after that, we will introduce a key distributed storage technology.

A. Secure Channel Establishment

According to the China DRM Technical specification, there are three parts of establishing a security channel between server and client: certificate exchange, key agreement and challenge response.

1) Certificate exchange

Client sends its communication request and device certificates to server.

Server checks the reliability of the source and content of the certificate and makes sure that it's not in the Certificate Revocation List (CRL), then, server sends its certificate to the client.

Client checks the certificate in the same way as server does.

2) Key agreement

Client generates a 2048 bits random number R_0 and encrypts R_0 with server's public key, and then sends the encrypted number $R_0^{'}$ to server.

Server receives $R_0^{'}$ and decrypts it then gets R_0. Server generates a random number R_1, encrypts it with client's public key and sends it to client. Server calculates the logic OR of R_0 and R_1 to get the result $K^{'}$, as in (1).

$$K^{'} = R_0 \,||\, R_1 \tag{1}$$

After that, server makes a SHA-1 hash of $K^{'}$ to obtain the session key.

doi: 10.18178/lnit.3.2.70-74

Client obtains R_1 and calculates the logic OR of R_0 and R_1 to get the result K, as in (2).

$$K = R_0 \parallel R_1 \tag{2}$$

Then, client makes a SHA-1 hash of K to obtain the session key.

3) Challenge response

Client generates a 2048 bit random number R_2 and makes a SHA-1 hash of $K \parallel R_2$, the result is c, as in (3).

$$c = Hash(K \parallel R_2) \tag{3}$$

After that, client sends c and R_2 to server.

Server makes a SHA-1 hash of $K' \parallel R_2$, obtaining the result e, as in (4).

$$e = Hash(K' \parallel R_2) \tag{4}$$

Then, server checks if e equals to c, if not, server will quick the authentication.

Server generates a 2048 bit random number R_3 and makes a SHA-1 hash of $K' \parallel R_3$, the result is f, as in (5).

$$f = Hash(K' \parallel R_3) \tag{5}$$

After getting the result f, server sends f and R_3 to client.

Client makes a SHA-1 hash of $K \parallel R_3$, obtaining the result g, as in (6).

$$g = Hash(K \parallel R_3) \tag{6}$$

Then, client checks if g equals to f, if not, client will quick the authentication.

By now, we can find that the most important factor of the protocol presented above is private key, which is what we concerned in this paper. Methods of protecting private key are proposed by researchers. Storing the private key in unreadable hardware such security chips is a good method for some special devices like Set-Top Boxes (STB), but not suitable for devices such as mobile, pad, etc. Tan has proposed a CA system in network computer environment based on server-end private key storage mechanism to solve the conflict between private-key storage demands of the end-entity and the no storage character of network computer systems in [6]. Yilong Zhao proposed a one-time application protocol in reference of USB Key which is based on root of trust [7]. Tian has proposed the method of key distributed storage technology in [8], here in section B we will introduce his method.

B. Key Distributed Storage Technology

In Tian's solution, 20% of client's private key is stored in server, and the other 80% private key message is stored in a 50k file in client. If client needs to get its private key to start service, it apply to server to get the 20% private key and the DEX code of private key collection that is used to extract the 80% private key from the 50k file. A host program is setup in the Client, the host program can decrypt, load, and run the DEX file from the memory [9].

The protocol occurred in the establishment of secure channel process which is after certificate exchange and before key agreement, the protocol can be descripted as follows:

- Client sends user name, password and client device certificate to server.
- Server verifies user identity and device certificate, and then sends server device certificate.
- Client verifies server device certificate and then sends message to server to request synthesis private key.
- Server gets the private collection DEX code and sends it to client.
- Client decrypts the code and then runtime executes the private key collection DEX code.
- Client collects the 80% private key message, and synthesis the private.
- After Client collects the whole private key message, the process of key agreement and challenge response are then executed, until the security channel is established.

Tian's solution can be expressed as Fig. 1.

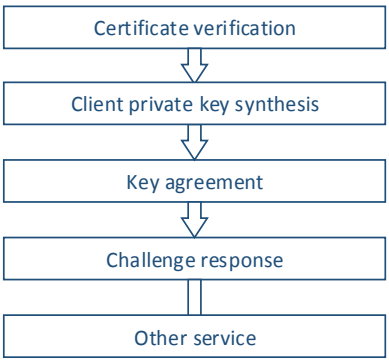


Figure 1. Tian's key distributed storage solution.

Tian's solution stores the private key information distributed in the server and client. Client will have to send its private synthesis request to server to realize client private key collection. Hackers will have to steal the 20% private key from server and 50k file and private key collection DEX code at the same time if he wants to steal client private key. Tian's solution makes a great inspiration to us in private key security, however, two points we need to pay attention to: firstly, by the time client request for private key collection code, client has not get its private key, and thus, client and server cannot negotiate a session key to encrypt this DEX code. Tian did not give a secure solution for the transmission of DEX code from server to client. And secondly, as is known to us: the client has a low reliability than server, if the file that contains part of private key message is stored in the client for a long time, the security of the private key may faces more risk.

III. Our Solution

In our solution, we will improve the key distributed storage method as mentioned above with two important areas. Firstly, we will propose a secure protocol to protect the transmission of private key collection DEX code from server to client. And then we propose a solution to change

the 50k file in the client by the time client request for private key synthesis.

A. Secure Transmission Protocol of Private Key Collection DEX Code

As analyzed in previous section, the private key collection DEX code may face security threat in the process of transmission without a security scheme to protect the transmission process.

Here, we will present our solution, the protocol is as follows:

- C→S $\{m_1, N_c, PW_c\}PuK_s$
- C→S DC_c
- S→C DC_s
- C→S $\{r_1, TK\}PuK_s$
- S→C $\{r_2, H(r_2)\}TK, \{r_3\}moblie$
- C→S $\{H(r_2 \| r_3)\}PuK_s$
- S→C $\{DF\}SK$

Among them, m_1 is used to identify the request to synthesis client private key, N_c is user name, PW_c is the password of user, PuK_s is the public key of server, the form of $\{m_1, N_c, PW_c\}PuK_s$ means m_1, N_c, PW_c are encrypted by PuK_s, DC_c is the device certificate of client, DC_s is the device certificate of server, r_1 is a random number generated by client, TK is the key generated by client to be used to encrypt r_2, and the form of $\{r_1, TK\}PuK_s$ means r_1, TK are encrypted by server public key PuK_s, r_2 and r_3 are random numbers generated by server, $H(r_2)$ is the hash of r_2, the form of $\{r_3\}mobile$ means r_3 is transmitted by mobile channel, $\{H(r_2 \| r_3)\}PuK_s$ means the hash of $r_2 \| r_3$ is encrypted by server public key PuK_s, DF is the private key collection DEX code to be transmitted from server to client, SK is the session key to encrypt DF, and SK is calculated as in (7).

$$SK = Hash(r_1 \| (r_2 \| r_3)) \tag{7}$$

The form of $\{DF\}SK$ means DF is encrypted by key SK.

Here one point needs to be specified, all accounts are required to bind a mobile number. Above-mentioned protocol is described in specific steps as follows:

- Client sends user name and password to server to request for private key synthesis, server authenticates user.
- Client sends device certificate to server and server checks client device certificate.
- Server sends device certificate to client and client checks server device certificate.
- Client generates a random number r_1, and a temporary key TK, after that, r_1 and TK are encrypted by server public key and then transmitted to server.
- Server generates random numbers r_2 and r_3, r_2 is encrypted by TK and then transmitted to client, and r_3 is transmitted to client by mobile channel.
- Client receives r_2 and r_3, and calculates the hash of $r_2 \| r_3$ as in (8) to get the result h. Then send the value of h to server after it is encrypted by server public key [10].

$$h = Hash(r_2 \| r_3) \tag{8}$$

- Server checks the value of h, if wrong, server sends an error message to client and then stops the session; otherwise, server calculates the session key SK as in (7), encrypts the DEX file with SK and send it to client.
- Client calculates the session key SK as in (7) to decrypt the DEX file, client executes the DEX code to obtain the private key message from the 50k large file, and then synthesizes the private key.
- After the private key message is collected, the session process of key agreement and challenge response is then executed, details of key agreement and challenge response are presented in Section II.

In our protocol, messages from client to server is encrypted by server public key, and the random numbers generated by server send to client one by mobile channel, another is encrypted by TK. And the session key to encrypt private key collection DEX code is generated by r_1, r_2 and r_3 among which, r_1 is generated by client, r_2 and r_3 are generated by server. We will analyze the security of the protocol in Section IV.

B. Change the 50k Large File

The security of 50k file that stores part of private key message may face challenge if it is stored in client for long time. But if we change the 50k file every time after client request for private key synthesize, even if the private key collection DEX code is stolen, hackers cannot get the 80% private key, because the 50k file is no longer the same. Here is solution.

After the process of key agreement and challenge response, a session key is created between client and server, then, we present a large file change solution using the session key:

- Server regenerates a 50k file that stores 80% of private key message and a corresponding private key collection DEX code that can be used to extract the private key information from that file. Server replaces the old private key collection DEX code with the new one.
- Server encrypts the 50k file with the session key and sends it to client.

- Client decrypts the 50k file and replaces the old 50k file with the new one.
- After the processes above are executed, other services are then started.

By now, we can show the improved key distributed storage system as a flow chat as Fig. 2. The blue modules in Fig. 2 are the parts of our security improvements.

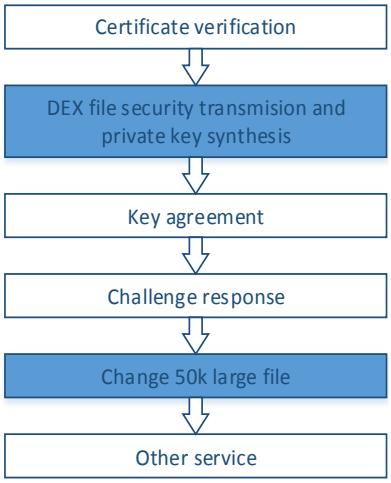


Figure 2. Flow chat of improved key distributed storage system.

IV. Conclusion

In DRM system, the security of private key is the most important part. Here we put forward a secure DEX file transmission scheme in the case of client has not acquired its private key based on the proposed scheme of Tian. In our scheme, the private key collection DEX code that is used to extract private key information is transfer to client with a designed secure protocol, the analysis of the security of the protocol is in the next paragraph. And, the 50k file with private key information in the client is updated in time.

Now, we will analyze the security of our security protocol.

A. Network Monitor

In our protocol, user name, password and random number and other information that transfer from client to server are encrypted by server public key, and messages from server to client is encrypted by password (PW_c , SK). The network interceptor can't get useful information from this method. And server uses mobile channel to transmit some information, this increase the difficulty of network monitor.

B. Replay Attack

In our protocol, each communication generates three different random numbers, which means, it's not possible for hackers to get information by replay attack.

C. Man-in-the-Middle Attack (MITM Attack)

MITM attack intercepts data of normal communication to access to data or tamper with the data, and the communication parties cannot sense the attack. In our protocol, messages from client to server is encrypted by server public key, and server send part of message through mobile channel, so that hacker cannot attack the communication through MITM attack.

By the analysis of the above, we can sum up that our protocol can effectively resist the existing main network attack: network monitor, replay attack, and MITM attack. And the method of changing the 50k large file in client each time before service started, we can prevent the possibility of the case hackers steal the private collection code and get the 80% private key by running or analyzing it.

With our two improvements, we have made the key distributed storage system more secure, and be able to resist stronger network attacks.

Acknowledgment

Here I want to take this chance to thanks to my tutor: Xingjun Wang, In the process of composing this paper, he gives me many academic and constructive advices, and helps me to correct my paper.

At the same time, I would like to appreciate Chen Tian senior, who guided me to start my subject from zero, and direct me to complete the paper patiently.

At last, I am very grateful of my dear friends, Zhiyong Li, Chao Cheng, who offer me the confidence and discuss with me about my paper.

References

[1] Marlin Developer Community. (2007). Marlin architecture overview. [Online]. Available: http://www.marlin-community.com/public/MarlinArchitectureOverview.pdf

[2] OMA2.0 DRM Specifications, Open Mobile Alliance, OMA-TS-DRM-DRM-DRM-V2_1-20070724-C, 2007.

[3] China DRM Forum Home Network Standard Architecture, China DRM Forum, unpublished V2.0, 2009.

[4] L. Xie, C. Tian, and X. Wang, "A device management and credit evaluation system in home network domain," *Lecture Notes on Information Theory*, vol. 1, no. 3, pp. 104-108, 2013.

[5] Y. Zhang, C. Yuan, and Y. Zhong, "Implementing DRM over peer-to-peer networks with broadcast encryption," in *Proc. 8th Pacific Rim Conference on Multimedia*, Hong Kong, 2007, pp. 236-245.

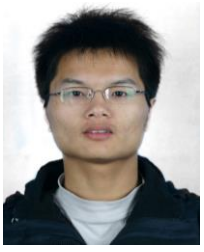
[6] T. Zhiyong, S. Tiange, and D. Yiqi, "CA system in network computer environment based on server-end private-key storage mechanism," *Journal of Tsinghua University (Science & Technology)*, vol. 47, no. 7, 2007.

[7] Y. Zhao, "An implementation of trusted computing based on trusted application," 2013.

[8] C. Tian, L. Xie, and X. Wang, "Combination of DRM and mobile code: A practice to protect TV contents and applications on Android smartphone," in *Proc. 4th International Conference on Networking and Distributed Computing*, 2013, pp. 89-93.

[9] Jack_jia. [Online]. Available: http://blog.csdn.net/androidsecurity/article/details/9674251

[10] Salted Password Hashing - Doing it Right. [Online]. Available: https://crackstation.net/hashing-security.htm



Yuxi Lin was born in Fujian, China, in 1987. He received the B.S. degree in electronic and information engineering from Tsinghua University, Beijing, China, in 2012. He is currently pursuing the M.S. degree in information and communication engineering from Tsinghua University, Beijing, China. Since 2012, he has been a Research Assistant with Shenzhen Key Lab of Information Security and Digital Content Protection

Technologies, Graduate School at Shenzhen, Tsinghua University. His research interests include digital right management, Android security and network security.

Xingjun Wang was born in Liaoning, China, in 1962. He received the M.S. degree in communication and electronic system and the Ph.D. degree in information and signal processing from Tsinghua University, Beijing, China, in 1991 and 1994, respectively.

From 1994 to 1996, he was a Postdoctoral Researcher with Department of Electrical Engineering, University of Michigan-Dearborn. From 1996 to 1997, he was a Senior Software Analyst with AT&T Corporation, Canada. From 1997 to 2000, he was first a System Software Designer with Department of Optical Communication and then a Senior System Architect with Department of Broadband Wireless Access, Nortel Networks Corporation, and Canada. From 2000 to 2003, he was a Vice President with Legend Silicon Corporation, USA. Since 2003, he has been a Research Fellow with Department of Electronic Engineering, Tsinghua University, and Beijing, China. He is currently an Associate Director of DTV Technology R&D Center, Tsinghua University. He also serves as Directors of Tsinghua-Renesas Joint Integrate Circuit Research Center and Shenzhen Key Lab of Information Security and Digital Content Protection Technologies. He has published over 40 peer-reviewed journal and conference papers. He holds three PCT and over 15 Chinese patents. His research interests include information security, digital content protection, image processing, HFC network and TV white space. Dr. Wang has many awards, including, most recently, the Recruitment Program of Global Experts Professorship at Tsinghua University from the Ministry of Education of China and Guangdong 100 Elites-Fellowship.