Environmental Key based Smartphone Data Encryption

Kun-Lin Tsai[1]*, Fang-Yie Leu[2]†, Jung-Chun Liu[2], and Chia-Yin Ko[2]
[1]Dept. of Electrical Engineering and [2]Dept. of Computer Science
Tunghai University, Taichung 40704, Taiwan

Abstract

Recently, smartphones have become one of the most important electronic devices in human being's everyday lives. Currently, a huge amount of private data is stored in smartphones, mobile devices, and cloud storages. However, unencrypted data may be hijacked in its transmission or unauthorized accessed by storage and/or delivery service provider. To secure private data for smartphones, in this paper, we propose a novel data encryption method, named the Environmental Key based Smartphone data Encryption (EKSE), with which private data or private photos stored in personal devices can be encrypted by an encryption key derived from one environmental matching key and several public matching keys, where an environmental matching key is a key derived from user's password and personal device's parameters, and the corresponding public matching key is generated by using elliptic curve encryption based on the environmental matching key. Our analyses show that the EKSE is able to resist machine-specific data leakage, insider threats, and impersonation attacks.

Keywords: Smartphone Security, Cloud Security, Environmental Key, Insider Threat

1 Introduction

In the past decade, with the development of smartphone, many people store their private data, such as photos, credit card numbers, customers' and family members' information, etc., in their smartphones or portable devices. Some data are even stored in cloud storages, e.g. Dropbox [1], Apple iCloud [2], and Asus Webstorage [3]. So users can retrieve the data from these storages to their own personal devices or deliver data from their own devices to others', meaning that the users can share data with someone who needs it. Although sharing data among different devices presents a high level of convenience, without data encryption, it will conduct high risk of data leakage. Besides, the encrypted data or photos inside personal devices may also be invalidly duplicated by service providers when the personal device has some problem and needs to be repaired.

Over the past few years, a considerable number of studies have been made on mobile device's security [6], [9], [7], [5], [4]. In [6], Polla *et al.* discussed mobile malware and categorized known attacks against smartphones, especially at an application level. They also reviewed security solutions for smartphones, focusing on existing mechanisms based upon intrusion detection and trusted mobile platforms. Wang *et al.* [9] showed smartphone threats and attacks, including sniffing, spam, attacker spoofing, phishing, pharming, vishing, and data leakage. To keep data confidential, users should employ encryption techniques and avoid storing sensitive information in plaintext in a smartphone. Therefore, in this paper, a smartphone data encryption method, named the Environmental Key based Smartphone data Encryption (EKSE) is proposed. According to the concepts of elliptic curve cryptography and asymmetric encryption, the environmental matching key and its public matching key are used to generate the file encryption key where an environmental matching key is a key derived from user's password and personal device's parameters, and the corresponding public matching key is generated by using elliptic curve encryption

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 2, Article No. 4 (July 31, 2016)
*Corresponding author: Email: kltsai@thu.edu.tw
†Corresponding author: Email: leufy@thu.edu.tw

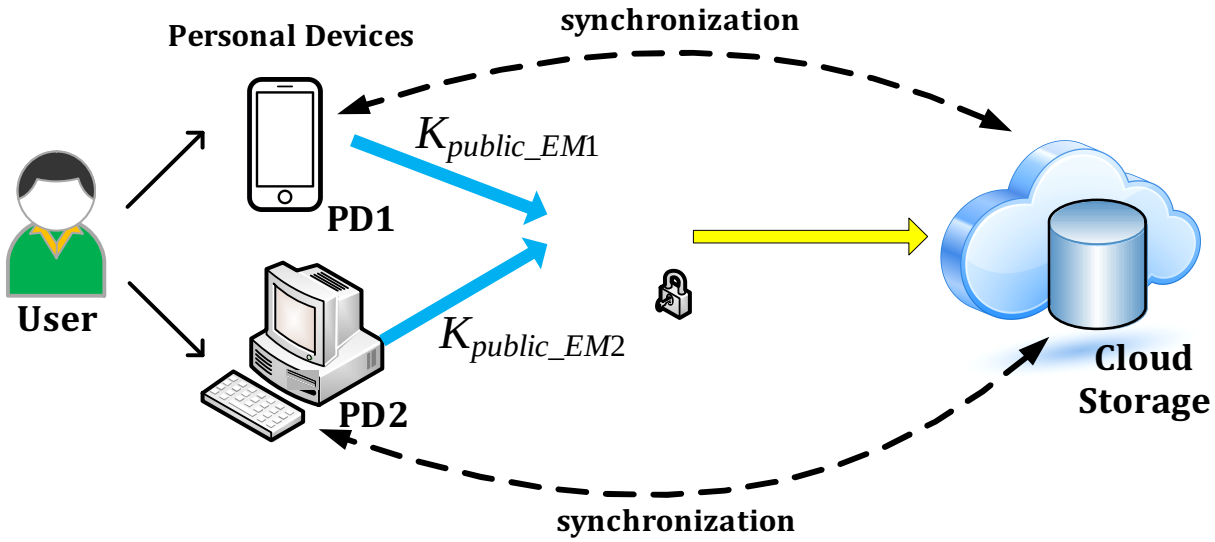

Figure 1: System architecture of the EKSE.

based on the environmental matching key. By using the EKSE, private data can be encrypted in an authorized device, and decrypted in another authorized personal device. The EKSE is able to resist not only machine-specific data leakage, but also insider threats, and impersonation attacks.

2 System Architecture and Environmental Key

2.1 System Architecture

The EKSE architecture, as shown in Figure 1, consists of cloud storage, a user, and his/her personal devices, such as smartphones, personal computers, tablets, etc. Before storing data into cloud storage, personal private data or photos are encrypted by using an encryption key which is derived from all of the public matching keys, which will be defined later. After data encryption, both the encrypted data and all the public matching keys are uploaded to cloud storage. Before accessing a file, the user firstly synchronizes his/her personal device with a cloud storage so that the newest file can be used in user's personal device, and then, decrypts the encrypted file by using the private matching key.

2.2 Initial Phase

In the initial phase of the EKSE, the user defines system parameters and selects a large prime number $q \approx 2^r$, $r > 160$. After that, the user

(1) determines F_q, which is a finite field of characteristic 2 and the order of q;

(2) determines the elliptic curve $E : y^2 \equiv x^3 + ax + b \pmod{q}$, where $a, b \in F_q$ and $4a^3 + 27b^2 \neq 0 \pmod{q}$;

(3) selects a base point P of order n on E, and $q^k \neq 1 \pmod{n}$ for any $k, 1 \leq k < 100$;

(4) generates a cyclic additive group G by P, the order of which is n;

(5) chooses two secure hash functions $H_1 : U \rightarrow Z_n^*$ and $H_2 : U^w \rightarrow Z_n^*$;

(6) publishes the system parameters $\{E, P, H_1, H_2\}$.

2.3 Environmental/Public Matching Key Generation

The idea of environmental matching keys is first introduced in [8]. A personal device's parameters, which can be IMEI code, CPU ID, Mac address of NIC, etc., are used to create an unique environmental matching key, which is then utilized to generate a public matching key. By using the unique environmental matching key, an encrypted file can only be decrypted by authorized personal devices. Assume that an authorized personal device *PDi* has w parameters, numbered from 1 to w. The procedure of the environmental matching key generation is as follows.

(1) Set the user's password key k_{PW} as the input of hash function H_1;

(2) use $H_1(k_{PW})$ to generate a parameter sequence, e.g., 1, 2, 1, ..., 9, 3, in which an element, e.g., j, of the sequence is a number between 1 and w;

(3) follow the numbers of the parameter sequence to access device numbers of the corresponding device, as the inputs of hash function H_2, to generate the environmental matching key $k_{EMi} = H_2(IMEI \| CPUID \| IMEI \| \ldots \| MAC)$ where H_2 is defined as $H_2 : U^w \rightarrow Z_n^*$;

(4) employ environmental matching key k_{EMi}, elliptic curve equation E, and base point P to generate the public matching key $K_{public\_EMi} = k_{EMi} \cdot P$;

(5) upload the public matching key to key management center.

Note that the length of parameter sequence in step (2) of this procedure may vary according to the required system's security level. The longer the sequence, the higher the security level.

3 File Encryption and Decryption

After the environmental matching key and public matching key are generated, the user can encrypt his/her private files, upload the encrypted files to cloud storages, and decrypt the files on different personal devices. Based on the number of personal devices that a user has, there are three cases of file encryption and decryption.

Case 1 (only one personal device):

When only one personal device is available, the files are encrypted by using public matching $keyE_K = K_{public\_EM1} = k_{EM1} \cdot P$. After that, the files can be uploaded to cloud storages. When the user needs the files, he/she firstly downloads the encrypted files from cloud storages or directly retrieves the files from somewhere to his/her personal device, and then, uses his/her private key k_{EM1} to decrypt the files.

Case 2 (two personal devices):

When the user has two personal devices, e.g., PD1 and PD2, the procedure of file encryption by using PD1 is as follows. The user

(1) accesses the public matching key $K_{public\_EM2} = k_{EM2} \cdot P$ of PD2;

(2) calculates $E_K = k_{EM1} \cdot K_{public\_EM2} = k_{EM1} \cdot k_{EM2} \cdot P$ where k_{EM1} is the environmental matching key of PD1;

(3) encrypts private files by using E_K, and uploads encrypted files to cloud storages.

If the user would like to decrypt the files with PD2, he/she

(1) acquires the encrypted files from cloud storages;

(2) obtains the public matching key $K_{public\_EM1} = k_{EM1} \cdot P$ of PD1 from key management center;

(3) calculates file decryption key $E_K = k_{EM2} \cdot K_{public\_EM1} = k_{EM2} \cdot k_{EM1} \cdot P$;

(4) decrypts encrypted files by using E_K.

Case 3 (more than two personal devices):

When the user has m personal devices, e.g., PD1 PDm, $m > 2$, the procedure of file encryption in PD1 is as follows. The user

(1) obtains the public matching keys $K_{public\_EMi} = k_{EMi} \cdot P$ of all personal devices from key management center;

(2) calculates $E_K = k_{EM1} \cdot \sum_{\forall i,\ i\neq 1,\ i\leq m} K_{public\_EMi} = k_{EM1} \cdot (\sum_{\forall i,\ i\neq 1,\ i\leq m} k_{EMi} \cdot P) = k_{EM1} \cdot (\sum_{\forall i,\ i\neq 1,\ i\leq m} k_{EMi}) \cdot P^{i-1} = (\sum_{\forall i} k_{EMi}) \cdot P^{i-1}$ where k_{EM1} is the environmental matching key of PD1;

(3) encrypts private files by using E_K, and uploads the encrypted files to cloud storages.

If the user would like to decrypt the files with PDx, $1 \leq x \leq m$, he/she

(1) acquires/synchronizes the encrypted files from cloud storages;

(2) obtains the public matching keys $K_{public\_EMi} = k_{EMi} \cdot P$ of all personal devices except PDx, i.e., $i/neqx$, from key management center;

(3) calculates the decryption key $E_K = k_{EMx} \cdot \sum_{\forall i,\ \text{expect PD}x} K_{public\_EMi} = k_{EMx} \cdot (\sum_{\forall i,\ \text{expect PD}x} k_{EMi} \cdot P) = k_{EMx} \cdot (\sum_{\forall i,\ \text{expect PD}x} k_{EMi}) \cdot P^{i-1} = (\sum_{\forall i} k_{EMi}) \cdot P^{i-1}$;

(4) decrypts encrypted files by using E_K.

4 Security Analysis

4.1 Machine-specific data leakage

In the key generation procedure, the environmental matching key is derived from the user's password and personal device's environmental parameters. To our knowledge, system parameters, like IMEI code, CPU ID, MAC address of NIC, etc., are individually unique among all devices having been produced in the world. Hence, it is almost impossible for users to generate the same k_{EMi} in different personal devices given the same parameter sequence. As a result, even an unauthorized person copies the files or photos to other device or computer, e.g., an untrusted computer X, from the authorized personal device, he/she is still unable to decrypt them on X, meaning that the EKSE well protects data from leakage.

4.2 Insider threat

An insider threat is a malicious person who comes from service provider or device manufacturer, such as employees or device repairer. They have inside information concerning the user's private photos and data. Without encryption, the photos and data in personal devices can be easily accessed by insider. In the EKSE, the photos and data are encrypted by using an environmental matching key and public matching keys. In fact, the environmental key is derived from user's password key k_{PW} and device's parameters. Even the insider can acquire the device's parameters; he/she cannot obtain the user's password key. Thus, the environmental key is still unknown by the insider.

4.3 Impersonation attack

When a hacker Z captures messages from the underlying wireless environment and the number of captured messages is large, Z can extract sensitive information, such as user's password, from them. In the EKSE, all sensitive files are encrypted by using the environmental matching key and public matching keys with ECC operation. The public matching keys are not private and can be accessed by any user from key management center. However, it is very hard to decrypt the files after applying ECC operation on the environmental matching key and public matching keys since ECC is based on discrete logarithm problem. Even though Z has captured a large amount of messages from the network, he/she is still unable to extract environmental matching key from the messages. Hence, the EKSE is able to thwart the impersonation attack.

5 Conclusion and Future Work

To secure private data from leakage for smartphones, in this study, the EKSE is used to encrypt secret files so that these files can only be accessed in a user's authorized personal devices. The environmental matching key of one personal device and several public matching keys from other personal devices are employed to deliver file encryption key. During the decryption procedure, the user can use the environmental matching key of his/her personal device and public matching keys from key management center to compute the file decryption key by using ECC addition operation. According to our security analyses, the EKSE is able to resist machine-specific data leakage, insider threats, and impersonation attacks.

In the near future, we would like to improve the key generation procedure as well as the file decryption procedure. Besides, we would also like to derive the reliability and behaviour models for the EKSE so that users can predict the system reliability and its behaviour before using it. These constitute our future studies.

References

[1] `https://www.dropbox.com/`.

[2] `https://www.icloud.com/`.

[3] `https://www.asuswebstorage.com/navigate/`.

[4] E. Chin, A. Felt, V. Sekar, and D. Wagner. Measuring user confidence in smartphone security and privacy. In *Proc. of the 8th Symposium on Usable Privacy and Security (SOPUS'12), Washington, DC, USA*, pages 1–16, July 2012.

[5] J. Ninglekhu, R. Krishnan, E. John, and M. Panday. Securing implantable cardioverter defibrillators using smartphones. *Journal of Internet Services and Information Security*, 5(2):47–64, May 2015.

[6] M. L. Polla, F. Martinelli, and D. Sgandurra. A survey on security for mobile devices. *IEEE Communications Surveys & Tutorials*, 15(1):446–471, February 2013.

[7] B. Rashidi and C. Fung. A survey of android security threats and defenses. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 6(3):3–35, September 2015.

[8] K. Tsai, J. Tan, F. Leu, and Y. Huang. A group file encryption method using dynamic system environment key. In *Proc. of the International Conference on Network-Based Information Systems (NBIS'14), Salerno, Italy*, pages 476–483, September 2014.

[9] Y. Wang, K. Streff, and S. Raman. Smartphone security challenges. *IEEE Computer*, 45(12):52–58, December 2012.

Author Biography

Kun-Lin Tsai received the B.S. degree in computer and information science from TungHai University, Taichung, Taiwan, in 1999, and the M.S. degree in computer science and information engineering from National Taiwan University, Taipei, in 2001, and the Ph.D. degree in electrical engineering from National Taiwan University in 2006. He was a postdoc at National Taiwan University of Science and Technology in 2007. He is currently an Associate Professor and the chairman of Department of Electrical Engineering of TungHai University. His research interests are low-power system design, information security system, and VLSI design. He is also a member of IEEE Circuits and Systems Society.

Fang-Yie Leu received his B.S., M.S. and Ph.D. degrees from National Taiwan University of Science and Technology, Taiwan, in 1983, 1986 and 1991, respectively, and another M.S. degree from Knowledge Systems Institute, USA, in 1990. His research interests include wireless communication, network security, Grid applications and Chinese natural language processing. He is currently a workshop organizer of CWECS and MCNCS workshops, a professor of TungHai University, Taiwan, the director of database and network security laboratory. He is also a member of IEEE Computer Society and one of the editorial board members of at least 6 international journals.

Jung-Chun Liu received his B.S. degree in electrical engineering from National Taiwan University in 1990. He received M.S. and Ph.D. degrees from the Department of Electrical and Computer Engineering at the University of Texas at Austin, in 1996 and 2004, respectively. He is currently an assistant professor in the Department of Computer Science at the Tunghai University, Taiwan. His research interests include cloud computing, embedded systems, wireless networking, network security, artificial intelligence, and wireless sensor networks.

Chia-Yin Ko received her M.S. in Computer Science from the University of Miami, Florida, USA in 1984. In 2013, she received her Ed D in Education from Seattle Pacific University, Washington, USA. Currently, she is an assistant professor in the Computer Science Department of TungHai University, Taiwan. Her research interests include online and blended learning, computer-supported collaborative learning, healthcare, wireless communication, wireless and sensor networks.