Title	Secure end-to-end browsing system with mobile composition
Author(s)	Jiang, ZL; Fang, J; Hui, LCK; Yiu, SM
Citation	The 2011 International Conference on Internet Technology and Applications (iTAP 2011), Wuhan, China, 16-18 August 2011. In Conference Proceedings, 2011, p. 1-4
Issued Date	2011
URL	http://hdl.handle.net/10722/152010
Rights	International Conference on Internet Technology and Applications (iTAP). Copyright © IEEE.

Secure End-to-end Browsing System with Mobile Composition

Zoe L. Jiang
School of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, China
zoeljiang@gmail.com

Junbin Fang, Lucas C.K. Hui, S.M. Yiu
Department of Computer Science
The University of Hong Kong
Hong Kong, China
jbfang,hui,smyiu@cs.hku.hk

***Abstract*—To fix the more and more serious leakage problem in remote access to confidential data, the paper designs and implements a secure end-to-end browsing system with mobile composition. It enables mobile-authenticated users to browse confidential files stored at server side using their personal computers securely. The authentication function is in real-time such that the system can stop the browsing function once it detects that the authenticated mobile is out of the communication range of user's personal computer.**

Data leakage; Secure browsing; Mobile phone; Java applet

I. INTRODUCTION

Mobile office infrastructure facilitates people's daily life by enabling them to access to data anytime and anywhere with "carry-nothing" philosophy, which, however, accompanies with a series of security problems as well, such as data leakage during transmission and processing when they connect to the servers and Intranet via WiFi, fixed connections in cybercafés, hotels and airports. Actually in recent years, we are continuously reported events of leaking confidential and privacy-related data, during the remote data access and local data processing procedures by careless users, especially from government [1-3]. Such trouble could be happened even more easily if some popular P2P sharing software is installed in the terminal devices (personal computer, notebook or PDA), since anyone connected to the Internet can search and download the data shared by foxy [4,5]. We cannot force people to operate their terminal devices as carefully as professional technicians, but to keep designing secure end-to-end data access and processing infrastructure without sacrificing convenience.

Consider that a police department runs a content server to handle with confidential data collected, including documents, pictures and videos. Since people are increasingly relying on the web for various operations, the problem is how to guarantee an authorized employee to browse such data with his notebook installing a generic/normal web browser? In this scenario, end-to-end security requires to protect data throughout the entire lifecycle, including the storage at server side in database, transmission through the Internet, processing and sanitization at client side in web application level. Due to the popular use of mobile phone as a convenient terminal device with constrained resources and small screen, it can help to authenticate users and provide some sensitive operations, although it has its own security problems [6].

In the paper, a mobile-aided secure end-to-end web browsing system is proposed for authorized users to access and browse confidential data anytime and anywhere.

A. System Design

The paper aims to design an end-to-end secure data access and browsing system, which keeps the rich browsing capability and strong computing resources of personal computers, as well as enjoys the mobile office advantage and trustworthy of mobile phone.

The role of the server is to authenticate user's identity and provide authenticated user confidential data requested. In particular, the system provider, a government department or a company, provides generic/normal user ID and password authentication for the registered users to download encrypted confidential data to their personal computers through the unsecure Internet. Then it depends on any one of the wireless communication service providers, such as China Mobile in China, to authenticate the registered user's mobile phone. The decryption-related information (the partial key) will send to the verified mobile phone for later decryption use. Here, the partial key means it cannot be used alone (without the corresponding mobile phone) to decrypt data even if it leaks out to the third party. The proposed two-level authentication scheme can highly enhance server-side authentication function without making server-side design and configuration complicated. Furthermore, all data are in encrypted form during transmission, which avoids unnecessary worry about data leakage.

At client side, the user should prepare a personal computer with a personal mobile phone connected to it through blue tooth or USB. Decryption key is retrieved by the mobile phone and sent to the personal computer to decrypt data. Here, a carefully-designed web application is required to protect decryption key and decrypted data from leakage or modification when browsing them, and sanitize them cleanly after usage.

978-1-4244-7255-0/11/$26.00 ©2011 IEEE

B. Related Work

End-to-end web application security on personal computer has been highly focused on since rich web application also suffers from various security attacks. For example, the most prominent attack on web application, script injection, can be easily deployed to web applications providing cross-site scripting [7]. Ulfar et al. proposed to enhance client-side security enforcement by specifying security policy [8]. There are also some similar work to enforce client-side security policies [9,10]. However, they all focus on pure security on personal computers at application level. Another method to secure data privacy is to use digital watermarking technique [11].

In 2006, Richard et al. proposed to secure mobile computing via public terminal [12]. They designed an application and thin-client server run on trusted smart phone, where security critical operation is processed. Non-security critical operation is performed by smart phone interacting with public terminal. They further designed a secure web browsing system [13] where public terminal can forward encrypted data to smart phone to decrypt and display. Such a system requires each public terminal installed with a remote device communication agent, and it is designed for secure multi-step online transaction.

Zhuang et al. proposed a novel architecture to provide trusted computing on public endpoints connecting to a piece of chip, called Mobile Trusted Platform Control Module, owned by individual user [14]. However, it intends for trusted computing but not secure browsing. Besides, from practical point of view, people are reluctant to bring another hardware (the chip) around for security-related operation.

C. Paper Organiztion

The paper first describes the system infrastructure in Section II. Then introduce the system implementation in Section III. Section IV is the conclusion.

II. System Infrastructure

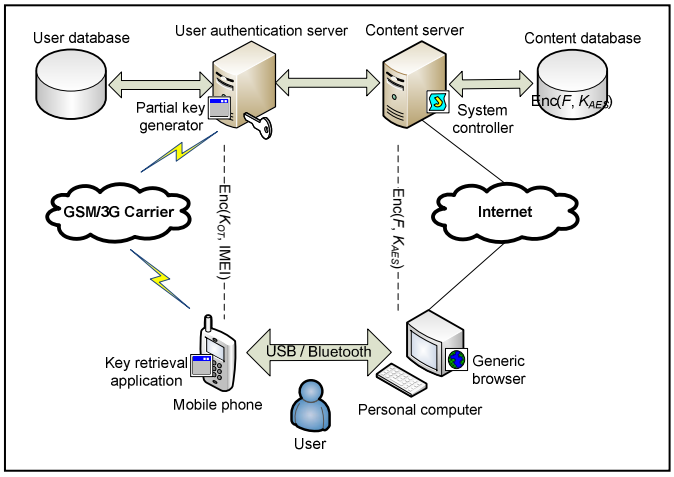

Figure 1. Secure end-to-end browsing system

As illustrated in Figure 1, the secure browsing system mainly includes (1) the content server (CS) with the system controller, (2) the user authentication server (US) with the partial key generator, (3) the mobile phone (MP) with the key retrieval application, (4) the personal computer (PC) with the generic browser. The first two components are executed at the server side for data storage and user authentication, the last two components are run at the client side for key retrieval, decryption and display usage.

NOTATIONS:

- F: the plaintext of a file
- e_F: the ciphertext of the file F
- K_{AES}: an AES key to encrypt/decrypt a file
- e_K_{AES}: the ciphertext of the key K_{AES}
- K_{OT}: a one-time pad key (or a session key)
- e_K_{OT}: the ciphertext of the key K_{OT} (or the partial key)
- Enc(*data*, *key*) and Dec(*e_data*, *key*): a pair of encryption and decryption operations to encrypt *data* using *key* and to decrypt *e_data* using *key*. It satisfies that *data* = Dec(Enc(*data*, *key*), *key*).

A. Content Server with System Controller

The content server (CS) is prepared by the system provider, such as a government department, to provide encrypted confidential data which are stored in the content database. In particular, the server (CS) generates a unique secret key (K_{AES}) for each file (F), encrypts it using AES algorithm [14], i.e.,

$$e\_F = \text{Enc}(F, K_{AES}), \quad (1)$$

which is stored in the content database. All K_{AES} keys are considered as permanent and should be safely stored. CS also provides normal user ID and password authentication for registered users as the first level authentication.

The system controller is a Java applet which actually starts up the whole session from the client point of view when it is downloaded and executed automatically at the client side. How it controls the user authentication, key distribution, as well as file downloading and decryption will be introduced in Section III.

B. User Authentication Server with Partial Key Generator

The user authentication server (US) is responsible to register its content-providing service to a wireless telecommunication service provider and obtain a service number. Then users should come to register to US with the triple (ID, mobile phone number, International mobile equipment identity (IMEI)) for the service. The triple is stored in the user database.

The partial key generator will generate one-time pad key (K_{OT}) for each session mentioned in II.A. which is controlled by the system controller. Once a specific file is requested, its

corresponding encryption key, K_{AES}, will be further encrypted by

$$e\_K_{AES} = \text{Enc}(K_{AES}, K_{OT}) \quad (2)$$

using AES algorithm/XOR. e_K_{AES} will be passed to CS and sent to PC together with e_F. To relate K_{OT} to the session initiator (the user), K_{OT} will be further encrypted by

$$e\_K_{OT} = \text{Enc}(K_{OT}, IMEI) \quad (3)$$

using XOR. IMEI is corresponding to the user's mobile phone which can be checked with the user database. We call e_K_{OT} as user's partial key since without knowing IMEI except the mobile phone itself, no one else can retrieve the integrated key K_{OT}.

C. Mobile Phone with Key Retrieval Application

At client side, the mobile phone (MP) whose IMEI number is registered to US should be ready for use. Besides, MP should be smart enough to run J2ME applications. Once the session starts up, MP is activated to request for partial decryption key through a short message. US replies to the request by sending e_K_{OT} to MP.

The key retrieval application is designed to re-calculate the decryption key K_{AES} when MP receives the replied short message including e_K_{OT} from US and downloads e_K_{AES} from PC by calculating

$$K_{OT} = \text{Dec}(e\_K_{OT}, IMEI) \quad (4)$$

and

$$K_{AES} = \text{Dec}(e\_K_{AES}, K_{OT}). \quad (5)$$

K_{AES} is passed through USB/Bluetooth to the system controller designed by ourselves to keep K_{AES} from leakage.

D. Personal Computer with Generic Browser

To run the whole system, the user should prepare a personal computer (PC) with a generic browser installed which supports Java applet. It is responsible to display confidential files for users. Note that the encrypted files are decrypted by the system controller by

$$F = \text{Dec}(e\_F, K_{AES}). \quad (6)$$

SECURITY ASSUMPTIONS. Assume that the connections between the user database and US, US and CS, CS and the content database, as well as the USB/Bluetooth connection between MP and PC, are secure. Here, "secure" means any third party cannot eavesdrop or tamper the communication between the above two parties. Since the wireless telecommunication network between US and MP is unsecure, we cannot expect the channel of sending/receiving short messages is secure. That is why only a partial key is sent to MP. However, we assume that the authentication function provided by US to authenticate MP is guaranteed. The Internet connection between CS and PC is unsecure, which only transmits encrypted data.

III. SYSTEM IMPLEMENTATION

In the section, a key distribution protocol is described first briefly, followed by introducing a MP-PC communication interface.

A. Key Distribution Protocol

Since all confidential files are stored and transmitted in encrypted form, what we should carefully pay attention to is the design of the key generation and distribution for encryption and decryption operations. There are two kinds of keys we design, K_{AES} as file encryption key and K_{OT} as one-time pad key. Besides, IMEI is used as the third key to protect K_{OT}. The details of how to generate and use such keys have been described in Section II as the equations (1)-(6).

B. MP-PC communication interface

Since all confidential files are stored and transmitted in encrypted form, what we should carefully pay attention to is the design of the key generation and distribution for encryption and decryption operations. There are two kinds of keys we design, K_{AES} as file encryption key and K_{OT} as one-time pad key. Besides, IMEI is used as the third key to protect K_{OT}. The details of how to generate and use such keys have been described in Section II as the equations (1)-(6).

First of all, the user logs into the system using his ID and PWD as usual. A list of filenames will be shown if he passes the verification by CS.

The system controller downloaded, a Java applet, will drives MP to send a short message asking for the partial key. US replies by generating K_{OT}, searching for the corresponding IMEI of MP from the user database and calculating the partial key e_K_{OT}, which is sent to MP as a short message. The key retrieval application in MP is used to calculate K_{OT}. All the above operations are automatically executed.

Once the user chooses the file he wants to browse, US should be aware of the user's action and calculate e_K_{AES}, which is downloaded to PC together with the corresponding encrypted file e_F. e_K_{AES} is further delivered to MP to retrieve K_{AES}(see equation (5), which is sent back to PC. The system controller will decrypt the file F and display it using a generic browser. The above actions can be repeated by the user as he likes.

At last when the user closes the window or logout to end the session, all sensitive data including K_{AES} and F in PC and K_{OT} in MP will be automatically clear!

IV. CONCLUSION

The paper designs a mobile-aided end-to-end secure browsing system to enable users access to confidential data anywhere and anytime securely. It protects confidential data from leakage at server side and during the transmission by

encrypting it. At client side, only mobile-authenticated users can get the decryption key to decrypt ciphertext. After that, the system will carefully erase the decryption key as well as the plaintext. Moreover, the authentication function provided by mobile phone is real-time such that once user's MP is away from the communication range with PC, the system will detect such event and resist further browsing by the user. As future work, it is desired to further discuss the security proof in implementing such system to real environment. Besides, the security of Java applet should be explored because we need to guarantee the security of the system controller which is developed using Java applet.

ACKNOWLEDGMENT

The work described in this paper was partially supported by the General Research Fund from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. RGC GRF HKU 713009E), the NSFC/RGC Joint Research Scheme (Project No. N_HKU 722/09), and HKU Seed Fundings for Basic Research 200811159155 and 200911159149.

REFERENCES

[1] http://irdial.com/blogdial/?p=1076

[2] http://articles.yuikee.com.hk/newsletter/2008/05/d.html

[3] http://www.cw.com.hk/content/privacy-commissioner-investigate-police-data-leakage

[4] http://www.cw.com.hk/content/report-foxy-exposes-hk-police-confidential-documents-again

[5] http://www.cs.hku.hk/cisc/event/20080827_FoxyPCO/FoxyPCO_20080827.pdf

[6] J. Lo, J. Bishop, and J. Eloff, "SMSSec: An end-to-end protocol for secure SMS," Computers & Security, vol 27, 2008, pp. 154-167.

[7] CGI Security. The cross-site scripting FAQ. http://www.cgisecurity.net/articles/xss-faq.shtml.

[8] U. Erlingsson, B. Livshits, and Y. Xie, "End-to-end Web Application Security," Proceedings of the 11th USENIX workshop on Hot topics in operating system, HOTOS'07, Berkeley, CA, USA, 2007

[9] E. Kirda, C. Kruegel, G. Vigna, and N. Jovanovic. Noxes, " A client-side solution for mitigating cross-site scripting attacks," *ACM Symp. on Applied Computing*, 2006.

[10] T. Jim, N. Swamy, and M. Hicks, "Defeating script injection attacks with browser-enforced embedded policies," Proceedings of the 16th international conference on World Wide Web, *WWW*, 2007.

[11] S. Wohlgemuth, I. Echizen, N. Sonehara, and G. Muller, "On Privacy-compliant Disclosure of Personal Data to Third Parties using Digital Watermarking," Journal of Information Hiding and Multimedia Signal Processing, Vol. 2, No. 2, pp. 270-281, 2011.

[12] R. Sharp, J. Scott, and A. Beresford, "Secure Mobile Computing Via Public Terminals," PERVASIVE 2006, LNCS 3968, pp. 238–253, 2006.

[13] R. Sharp, A. Madhavapeddy, R. Want, and T. Pering, "Enhancing Web Browsing Security on Public Terminals Using Mobile Composition," *MobiSys'08*, June 17–20, 2008, Breckenridge, Colorado, USA.

[14] L. Zhuang, C. Li and X. Zhang, "A novel architecture for trusted computing on public endpoints," 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing, pp. 232-235. 2010.

[15] AES, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard