Computers and Electrical Engineering 35 (2009) 370–375

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A novel software key container in on-line media services

Neng-Wen Wang, Yueh-Min Huang *

Department of Engineering Science, National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan 701, Taiwan, ROC

ARTICLE INFO

Article history:
Available online 15 August 2008

Keywords:
IP right
Encryption
Decryption
Key container
Human-trapdoor function

ABSTRACT

Due to the explosive growth of the Internet and the pervasion of multimedia, protection of IP rights of digital content in transactions induces people's concerns. For fee-based media services, data encryption may be the best solution for protection of the media. The encryption (decryption) keys placement may be a trivial but crucial issue for users. It is a significant issue that how to practically protect user's key with the password-based cryptographic scheme and at different security levels. Nowadays, key container storing user's key can be implemented by hardware or software-only. Unfortunately, the hardware key containers require expensive infrastructure; On the other hand, the software-only key containers are either insecure or impractical. Moreover, both of the hardware and software just store user's key with the single security level. To solve these problems, we propose a novel software key container in on-line media services that can provide an adaptively secure and practical solution to protect user's key. We use a human-trapdoor distortion function and symmetric cipher to protect user's key in our key container so that it is computationally infeasible to break the system by using machine attack alone. The idea is to ensure that people must participate to verify each guessed password in the attack. User can adjust the security level of container according to the security requirement. Therefore, the attacker cannot succeed to extract user's key within a reasonable time and budget.

1. Introduction

The explosive growth of the network and the pervasion of multimedia have encouraged many flourishing media services on the Internet. However, the protection of IP rights of digital content in transactions induces people's anxiety. Inexpensive tools with easy manipulation have deteriorated the circumstance. Current security requirements and copyright protection mechanisms need to work on-line.

For media service systems in the Internet, user's authentication is most essential in association with the access control of the media system. On some subscribed pay-media services in the Internet, additional data encryptions are also adopted in protection of the media. For decades, password has been the major means for user authentication on computer systems. Password-based authentication mechanism is the most extensively used authentication mechanism in the Internet and mobile communication systems. However, those weak passwords are prone to dictionary attacks. Nowadays, other alternative methods are possible for user authentication [1]. Some people use smart card and identification card to store their secret tokens [2]. Such methods usually need special sensing devices. Moreover, theft and counterfeit are serious threats to these systems. To have a token does not necessarily imply to possess a legitimate ownership. Other people may use biometric methods. Biometric methods identify individuals based on distinguishing human features [3,4]. Counterfeit and theft are generally more expensive than they are with other methods since these biometric features cannot be easily substituted. However, biometric methods generally require more costly and specialized hardware.

* Corresponding author. Tel.: +886 6 275 7575x63336; fax: +886 6 276 6549.
E-mail addresses: nwwang@cc.kyu.edu.tw (N.-W. Wang), huang@mail.ncku.edu.tw (Y.-M. Huang).

0045-7906/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2008.06.013

Data encryption may be the best solution to protect on the media for fee-based services. It may be a trivial but crucial issue for the placement on encryption (decryption) keys. The user's authentication can be automatically solved while the protected media are encrypted by keys. To use user's password as the key may be the simplest way. However, it is vulnerable for the dictionary attack since users' passwords are usually low entropy. Hardware smart card can securely store user's key. It may be an excellent solution for key storage. Nevertheless, it requires expensive infrastructure.

To solve those problems, we propose a novel software key container in on-line media services that can provide an adaptively secure and practical solution to protect user's key. It can prevent both on-line and off-line attacks on public environments. Our scheme is both practical and effective with only a little bit cost of software. We use simple hash operation and symmetrical encryption system on user's machine. If user's machines are implemented by microprocessor-based system with some simple peripheral devices such as LCD and keypad, the software container can be also embedded easily. We also describe the security of our scheme in detail. The analysis indicates that our scheme significantly increases the ability to protect user's secret key from disclosure.

2. The related works

The standard PKCS #5 [5] provides a method of key storage that encrypts a plain-textual private key with a secret key derived from a password. This method is vulnerable for the dictionary attack since the plaintext is verifiable and users' passwords are usually low entropy. Other methods following PKCS #5 also suffer from the dictionary attack. However, it is used extensively because of its convenience and low cost.

A hardware smart card can securely store user's key. However, it requires expensive infrastructure. In 1999, Hoover and Kausik [6] proposed a software smart card which applied a cryptographic camouflaged technique to protect a private key under some constrains. These constraints include that a signature must be encrypted; only asymmetric keys are protected; and a camouflaged private key must be used in a closed public-key infrastructure. Just as a hardware smart card, the software smart card can securely protect a private key at the same security level without the problem of expensive infrastructure. Nevertheless, it is impractical because of these constraints.

For password-based authentication systems, Wang et al. [7] had divided them into two modes. The first mode is the simply password-transfer systems. The user simply submits his password to the server. For security reasons, the server (for examples, in most Unix-like systems) stores related password-verification data (PVD), which is generally derived by the hash function of user ID, password and salt. In the secure password-transfer approach, the password should be encrypted before it is submitted for security considerations. The communication channel can be set up by the Transport Layer Security Protocol (TLS) or it predecessor, Secure Socket Layer (SSL) protocol. This simply password-transfer system is notoriously vulnerable to the dictionary attack. The second mode is to show user's possession of the password without sending it. The challenge/response authentication system belongs to this mode. However, it is still vulnerable to eavesdropping attacks. A new category of protocol paradigm following this path is called password-authentication key exchange (PAKE). The client authentication is fulfilled by establishing an authentication session key with the server. The session key cannot be set up if the client does not have the password or the server does not the related PVD. PAKE protocol realizes its security goals through the usage of public-key exchange techniques. The PAKE protocol operates on the basis of some facts. (1) a PAKE user possesses a password only; (2) user uses the client program to login the system by system parameters only (such as the g and q for Diffie-Hellman) and no secrets (say, a private key) are hard coded into it. Many significant researches in this category have been developed. Some protocols use the Diffie-Hellman key exchange algorithm. For example, the Encrypted Key Exchange (EKE) [13,14], Secure Password Exponential Key Exchange (SPEKE) [15,16], Simple Remote Password (SRP) [17], the PAK protocol [18] and the KOY01 protocol [19]. Other protocols use the RSA algorithm, such as the BPR00 protocol [20] and the SNAPI protocol [21].

Both the secure password-transfer approach and the PAKE approach thwart off-line dictionary attacks from the network very well. The secure password-transfer approach and most of the existing PAKE protocols use a single server to store users' PVD. The exposure of server seems to be inevitable. If the attacker compromises the centralized server and steals the PVD, he could simply guess a password (from his dictionary), compute the corresponding PVD and verify the correctness of the password. To prevent the compromise of a single server, Wang et al. [7] employs multiple (say n) servers to store PVD. Among the multiple PVD servers a user's PVD is shared and the shared PVD is never reconstructed during user authentication. The described system is intrusion-tolerant in the sense that compromising up to $(t-1)$, $2 < t < n$, where t is the threshold number. In the paper [8], Kwon has proposed "Virtual Software Tokens" to secure PKI roaming. Kwon uses the similar idea to run RSA algorithms with multiple servers. His basic is to hide a real ID and split a password as well as a private exponent over multiple servers. The multiple servers will generate signatures or decrypt messages via virtual software tokens.

In this paper, we propose a novel software key container in on-line media services to meet user's different security requirements. It can provide an adaptively secure and practical solution to protect user's key. Unlike the usage of multiple servers in [7,8], we emphasize the enhancement of security on a single server. Our system is carried out in a single server. Accordingly, it will be more efficient without lots of overhead on the communication. In our implementation, we use a human-trapdoor function. The user's private key is stored in its twisted form distorted by this function. This function is used to protect user's key in our key container so that it is computationally infeasible to break the system by using machine attack alone. User can adjust the security level of container according to the security requirement. Therefore, the attacker cannot succeed to extract user's key within a reasonable time and budget.