Information Sciences 211 (2012) 36–47

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Dynamically generate a long-lived private key based on password keystroke features and neural network

Ting-Yi Chang

Master program in e-Learning, Department of Industrial Education and Technology, National Changhua University of Education, No. 1, Jin-De Road, 500 Changhua City, Taiwan, ROC

ARTICLE INFO

Article history:
Received 25 July 2011
Received in revised form 23 February 2012
Accepted 4 April 2012
Available online 14 May 2012

Keywords:
Biometric
Cryptography
Key store
Keystroke feature
Neural network
Password guessing attack

ABSTRACT

It is well-known that the protection of long-lived private keys in cryptographic schemes is one of the most important issues for information security. Any cryptographic scheme that reveals private keys will soon have its security absolutely disintegrate. For example, in digital signature systems, anyone who obtains the victim's private key, authenticity and non-repudiation can no longer be claimed or proven. Because the private key is a long random bit string and should be stored securely, some special cryptographic hardware such as an IC (Integrated Circuit) card is needed to store and protect the private key. Unfortunately, the security of private keys solely depends on the vulnerable passwords. This study proposes combining a neural network technique and password keystroke features to dynamically generate a long-lived private key rather than statically stored in a storage unit. Compared with other traditional methods, even if the storage unit is lost or the password is revealed, the probability of exposing the private key is reduced.

1. Introduction

The rapid growth of networks, in terms of both number and size, encourages and forces the linking together of more computers in order to share various kinds of data and exchange huge amounts of information. The PKI (abbr. *Public Key Infrastructure*) is a collection of technologies, processes and organizational policies that support public-key cryptography applications to verify the relative authenticities. The PKI also provides various mechanisms to ensure the trusted relationships are established and well-maintained, as well as to certify the foundations of confidentiality, authentication, integrity and non-repudiation. With citizen digital certificates in the PKI, we can enjoy convenient and highly secure application services on the Internet, provided by diverse governmental agencies such as personal tax declarations and land administration services.

Cryptographic keys used for signatures and decryptions within a PKI environment can be generated in a *centralized* or *decentralized* manner. Under a centralized approach, the keys are generated and stored on a central server, and the keys are transmitted to the individual system when required. Hence, the central server is a trusted third party. It does, however, present one drawback that the third party can become a bottleneck for the entire system. Under a decentralized approach individual computers generate and store the keys locally. That the private key must remain confidential at all times and stored securely is a critical concept common to all PKIs that must be understood and enforced. Because the long-lived private key in public-key cryptography for decrypting ciphertext or signing messages is a long random binary string (RSA keys are typically 1024–2048 bits long) it cannot be memorized by human beings. It therefore must be permanently stored somewhere for future use. This storage area is generally referred to as a *key store*. This rests on the assumption that the one person who has the ability and permit to access the key store is the owner of that private key.

E-mail address: tychang@cc.ncue.edu.tw

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.04.009

In most software implementations, the user's key store *key store* is protected by encrypting it with a norm symmetric encryption $E_k(\cdot)$ using a key k taken from the hash $H(\cdot)$ of a password pw, denoted as $C = E_{H(pw)}(key\ store)$. When the user wants to access his/her key store, s/he will be prompted for the same password pw, which will obtain the key store by decrypting C and allow him/her access to the secret key. Unfortunately, in software implementations, passwords are relatively unsecure because they come from a rather limited set of possibilities and therefore they are vulnerable to the password guessing attack [3,20,28]. The password guessing attack is a kind of brute force attack, that is, the attacker has a list of words from a dictionary for example to mount the brute force attack. The attacker obtains the storage to access the ciphertext C of key store, even if the symmetric key $k = H(pw)$ is taken from the hash of pw, the attacker can verify the correctness of the guessing password pw' by checking whether $C \stackrel{?}{=} E_{H(pw')}(key\ store)$ holds or not. After revealing the victim's password, the attacker is able to freely access his/her key store. At this point authenticity and non-repudiation can no longer be claimed or proven.

Ellison et al. [7] used the concept of secret sharing to split up the key store into several parts. Each part is then encrypted separately, with a password independent of all the others. The attacker should successfully guesses some of these passwords to decrypt the key store. Obviously, the security key store is based on multi passwords rather than a single password. The attacker should pay more attention to reveal multi passwords. However, since the password is sample and short enough to be remembered, then the attacker can look over someone's shoulder to get passwords (shoulder-surfing attack).

Alternatively, there are other special cryptographic hardware implementations such as an IC (*Integrated Circuit*) card (so-called smart card, or chipped card) that can be implemented within a PKI to hold users' private-key information. A key pair, i.e., public and private keys, can be created within hardware modules. Similarly, the user should be required to provide a PIN (*Personal Identity Number*) to authenticate his/her identity. In order to avoid an applicant losing his/her IC card, the cryptographic modules in an IC card should be internally generated and validated to meet at least the criteria specified by the FIPS (*Federal Information Processing Standards Publication*) 140-1 or 140-2 standards. The FIPS 140-1 and 140-2 standard was created by the NIST (*National Institute of Standards and Technology*, [22]) and specifies requirements for the proper design and implementation of products that perform cryptography. Even if the PIN is known in advance, the private key stored in an IC card cannot be exported or copied for unauthorized purposes after generation. The processes for generating signatures or decrypting cipher texts require inserting an IC card into a reader attached to a computer and providing an authorization code to access the IC card. IC cards usually provide a higher level of protection compared to holding the key in software, because they more tamper-proof in nature. Compared to holding the key in software implementation, IC cards are able to provide a higher level of protection. However, the powerful abilities of attackers cannot be ignored. In other words, the attacker may directly access the IC card chip with no password authentication. For example, as reported in [10], experts successfully cracked the encoding scheme with little effort to read out data stored in the IC card.

A private key is a crucial and fundamental component of any PKI implementation. In this paper, the neural network technique and password keystroke features are combined to dynamically generate the long-lived private key. This scheme overthrows the traditional ways of protecting private keys. Even if the adversaries obtain the storage device or the password, the probability of revealing the user's private key remains very difficult. The proposed scheme is able to reduce the vulnerable password-based schemes when the password is revealed or the storage storing the private key is lost. In summary, a target output of 2048-bit randomized binary integer vector, i.e., the "private key" and a particular input real number and integer vector, i.e., the "password keystroke features" with the corresponding user will be used to train the layered neural network. After adjusting and training the layered neural network, only the weights and biases of the connections between neurons and the transfer functions used in neurons are stored for generating the private key. When a valid user enters his/her password, the password keystroke features are fed into the trained layered neural network and then to generate the user's private key. Note that the private key is not statically stored in the storage.

The organization of this paper is as follows. Section 2 introduces the basic types of keystroke features and how a layered neural network is able to learn the relationship between a particular input and output pair. Section 3 proposes the dynamic 2048-bit private key method based on keystroke features and the layered neural network. Attack results from human tests are presented in Section 4 to explain how the proposed method is able to protect the private-key even if the corresponding password is revealed. Section 5 shows the computer test brute force attack. The proposed method can reduce the successful probability of mounting brute force attacks even if some information on the feature ranges are known. At the same time, the proposed scheme's performance is evaluated. Discussions and conclusions are presented in Section 6.

2. Technical backgrounds

The technical backgrounds used in the proposed scheme are introduced in this section.

2.1. Keystroke features

The password-based authentication is the simplest and most commonly used approach in identity verification. However, the security is broken and fragile when the password is stolen or weak. Keystroke dynamic-based authentication is based on typing biometrics, which was proposed by observing that the keystroke features of a user's password are repeatable and distinct from those of other users. This provides additional security in password-based authentication which is vulnerable to