SCIS'99 The 1999 Symposium on Cryptography and Information Security
Kobe, Japan, January 26-29, 1999
The Institute of Electronics, Information and Communication Engineers

Software Protection Using Public Key Infrastructure

Byoungcheon Lee *
sultan@icu.ac.kr

Kwangjo Kim *
kkj@icu.ac.kr

Abstract— In this paper we propose a new software protection system using Public Key Infrastructure(PKI) and introduce the concept of User Dependent Software(UDS). Certificate is an electronic document which combine an user's identity and public key and is signed by a Certificate Authority(CA), so it can be used to represent user's social identity. UDS includes user specific information encrypted with user's public key appearing in user's Certificate. To use the software user has to present the private key corresponding to the Certificate. So UDS is for only one user.

It is expected that PKI will be set up throughout the society and Certificate will be used for a large number of applications in the near future. We propose to use PKI for software protection. The private key corresponding to a Certificate is private secret. Copying other's UDP is of no use without the legitimate user's private key, so illegal software usage is prevented. Using this technology, software venders can sell and deliver software via online service, which will reduce a large amount of cost for manufacturing and deliverying software products.

Keywords: Public Key Infrastructure, Software Protection, Copy Protection, Copyright Protection, User Dependent Software.

1 Introduction

The advances in computer and information society are changing our society greatly and are lead by software developers in many parts. But one of the headaching problems regarding the software industry is the unauthorized use and distribution of software. Copyright laws protect developer's right in legal sense, but are hardly enforced causing great loss to the software companies. Software protection is a term to prevent illegal use of software and protect developer's copyright.

Software protection can be divided into two categories, one is copyright protection and the other is copy protection. For legal protection of copyright, developers can register their products to the authorized organization and they can argue their right later when illegal copying of their idea or code occurs. As a technical tool for copyright protection, there have been extensive researches on watermark technology. By adding a unique watermark on multimedia data, illegal copying and use of the protected data can be detected later. Recently many useful technologies to add developer's ID to software were developed, for example by using dummy variables or by using two equivalent operations selectively[1, 2, 3].

For copy protection, J. Gosler addressed many technologies such as signatures on floppy disks(magnetic and physical) and Hardware Security Devices(HSD), and also introduced the concept of Software Analysis Denial(SAD) which is to prevent attacker from analyzing the software[4]. A. Herzberg proposed Public Protection of Software(PPS) which required a new CPU architecture[5]. O. Goldreich[6] and R. Ostrovsky[7] proposed software protection schemes where encrypted program runs on a protected chip and the values stored in the general-purpose memory are hidden using encryption. These schemes seem to be ideal protection systems, but does not seem to be practical because they require basic architectural change from current computer systems.

In software industry, many ad-hoc methods have been used. For example, requiring license number or access code when installing a software, using hardware key lock, requiring vender supplied special floppy disk, using IC card where secret key is saved, generating a software only for specified system, and on-line registration when installing network application have been used. These methods do not provide general and sufficient protection mechanism. One of the main reason is the lack of convenient means for authenticating user's identity. In this sense the forthcoming PKI trust model is an efficient and useful candidate for software protection.

It is expected that PKI trust model will change the society drastically. Until now there have been many CA systems implemented using the X.509 standard[8], and Public Key Infrastructure based on X.509(PKIX)[9] is under standardization process. It is expected that in the near future each personnel in society will use Certificate issued by CA for a broad range of applications such as Electronic Commerce(EC), e-mail transfer, mobile communications, etc, and the private key will be protected in a secure hardware device such as IC card. The operation using private key is user's responsibility and cannot be repudiated, so users have to be careful enough to protect their private key. When CA issues a

* School of Information and Computer Eng., Information and Communications Univ., 139 Kajong-dong, Yusong-gu, Taejon, 305-350, Korea

Certificate, it guarantees user's identity and legal usage of keys.

In this paper we propose a new software protection system using PKI and introduce the concept of User Dependent Software(UDS). The software provider generates and provides UDS which is compiled with user dependent information including user's Certificate. For a user to access the UDS, he or she has to present the private key corresponding to the Certificate.

In section 2, we describe our basic approach and two proposed schemes for constructing UDP. The special features of the proposed systems appear in section 3. A possible scenario of software sale and delivery is discussed in section 4. Conclusion and further works are described in section 5.

2 Proposed schemes

2.1 Our approach

The basic lines of our approach are as follows.

- Software provider generates and provides a User Dependent Software(UDS) which is compiled with user specific information encrypted with user's public key appearing in user's Certificate.
- An user has to present his or her private key corresponding to the Certificate to use the software.

We represent the basic structure of the proposed software protection system in Figure 1. The protected software is divided into two subparts, User Independent Part(UIP) and User Dependent Part(UDP). UIP is a group of subprograms where most of the functions of the software are implemented, but cannot be executed alone and is accessed only by UDP. The Dynamic Link Library(DLL) files in Windows are good examples. UDP is an executable program and is compiled with user specific information encrypted with user's public key. UDP has the role to control access to UIP. To use the software, user has to present his or her private key to UDP to prove that he or she can decrypt the encrypted information successfully. The only user having the private key can use the software, so the protected software is for only one user. The reason dividing the software into UIP and UDP is to reduce the online compile time for generating UDP and to distribute UIP freely in advance. Our main interest is how UDP is constructed using user's public key and how user's private key is used to prove the authenticity. In following sections, we propose two possible designs of UDP. One is using encrypted data block and the other is using encrypted executable module.

2.2 UDP with encrypted data block

In this method UDP is an access control application compiled with the 5 user-dependent data blocks as depicted in Figure 2a. When a customer(user) orders a copy of software, software provider asks user information including user's Certificate. Software provider generates the 5 user-dependent data blocks as follows and its flow diagram is depicted in Figure 2b.

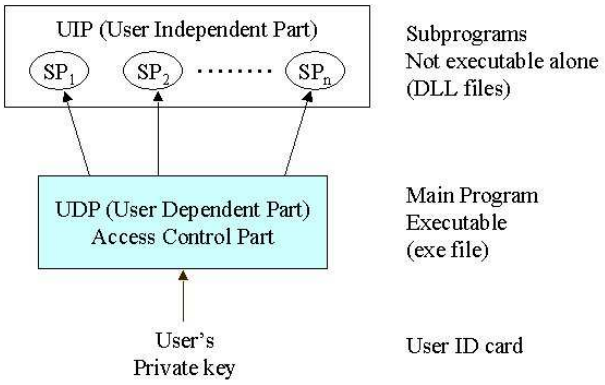


Figure 1: Basic structure of the proposed software protection system.

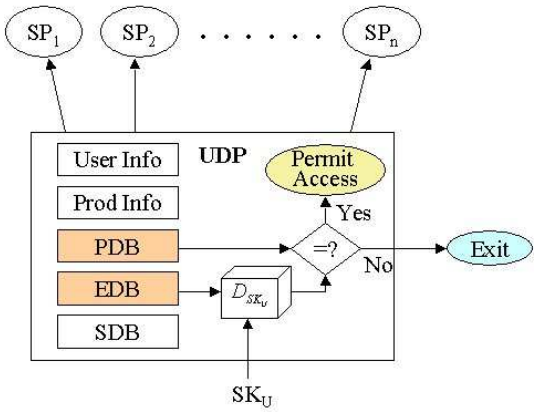
- UserInfo : user Name, supplier address, user's Certificate, and purchase date, etc.
- ProductInfo : provider name, provider address, provider's Certificate, product number, and term of validity, etc.
  $PDB = H(UserInfo||ProductInfo)$
- EDB(Encrypted Data Block) : encrypted block of PDB using user's public key
  $EDB = E_{PK_U}(PDB)$
- SDB(Signed Data Block) : signed block on PDB with provider's private key
  $SDB = S_{SK_P}(PDB)$

UserInfo includes user name, user address, user's Certificate, and purchase date, etc. ProductInfo includes provider name, provider address, provider's Certificate, product number, and term of validity, etc. Plain Data Block(PDB) is the hashed value of UserInfo and ProductInfo. PDB is encrypted with user's public key PK_U to produce Encrypted Data Block(EDB) and the provider signs on PDB with his private key SK_P to produce Signed Data Block(SDB). The provider generates UDP by compiling access control module with these data blocks. The 5 user-dependent data blocks are embedded in the access control application in binary form, so these data blocks will not be accessible in other way.

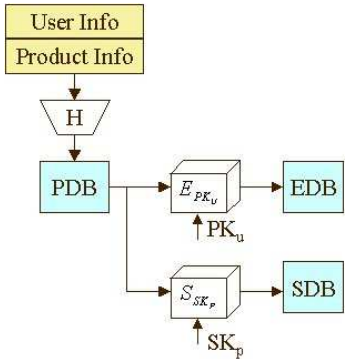
When user tries to use the software, he or she has to present the private key SK_U to UDP. When EDB is successfully decrypted to PDB using the private key, UDP permits access to the subprograms in UIP. By verifying the provider's signature on SDB, user can confirm the validity of the software. If user's private key SK_U is saved in a hardware device such as IC card, the decryption process can be designed to operate in the hardware device for the user's private key not to be exposed outside.

2.3 UDP with encrypted executable module

The method described above has a possible weak point that the 5 user-dependent data blocks and access



(a)



(b)

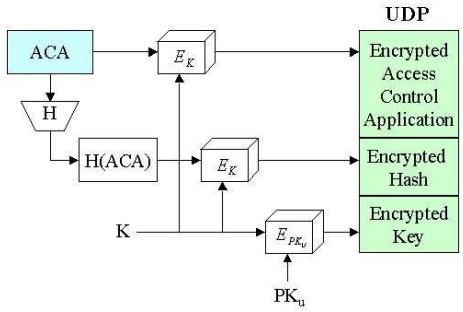
Figure 2: UDP with encrypted data block. (a) Structure and operation of UDP, (b) Generation of user-dependent data blocks

control module are embedded in the binary executable application in available form(not in encrypted form). A powerful attacker may possibly try to analyze the software, for example using software debugger, and try to modify the data blocks or bypass the access control module. So we propose another method of constructing UDP where the access control module is encrypted.

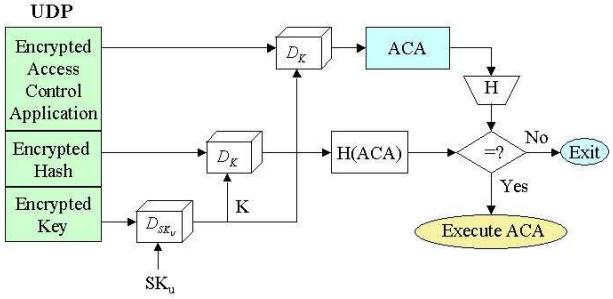
Figure 3a represents the generation process of UDP which includes encrypted executable module. Access Control Application(ACA) is an executable module which can access UIP. ACA is encrypted with a randomly generated session key K to generate Encrypted ACA. Hashed value of ACA, $H(ACA)$, is also encrypted with K to generate Encrypted Hash. Finally the random session key K is encrypted with user's public key to generate Encrypted Key. These three parts are concatenated to form UDP.

When user tries to use the software, he or she has to present the private key SK_U to UDP. As depicted in Figure 3b, the random session key K is recovered by decrypting Encrypted Key with SK_U. Using the symmetric key K, ACA and $H(ACA)$ are recovered from Encrypted ACA and Encrypted Hash respectively. If the newly computed hash $H'(ACA)$ and recovered $H(ACA)$ coincide, the execution of ACA is permitted.

In this model, the executable ACA is embedded in UDP in encrypted form. So possible attacker has to decrypt UDP first to start any possible analysis of the software.



(a)



(b)

Figure 3: UDP with encrypted executable module. (a) Generation of UDP, (b) Execution of UDP

3 Features of the proposed software protection systems

The proposed software protection systems can provide software providers with essential protection mechanism of software developer's copyright and prevention of illegal copying of software. Software is sold only to users authenticated by CA. The software sold to a customer is only for one user who has the private key corresponding to the Certificate. If the private key is protected in hardware device such as IC card, illegal copying of private key will be very difficult. If the Certificate is used for many important applications such as EC, e-mail transfer and mobile communication service, illegal copying of private key is user's responsibility.

If it is assumed that PKI is widely used in society, no more cost is needed for implementing the proposed software protection mechanism. If most individuals have their own Certificate and own identification card where the private key is saved, they can easily order the soft-

ware with the Certificate and use it with the identification card.

The proposed software protection systems provide with special features that other existing protection systems do not provide with.

- The software can be installed and used in as many computer systems as user wants. To use the software, user identification card is required. So using in many computer systems is not violating the copyright laws.
- The same private key can be used to protect many software packages in the same computer system.
- The software provider can produce software for many legitimate users. In most simple way users can have their own UDP sharing the UIP in a computer system. In other way the software provider can produce UDP which includes many user's user-dependent information or can produce UDP with a group key.
- Installing software is very simple in this model. Software providers can distribute UIP freely in advance to users through hardware manufacturers or internet services. Because UIP is already installed in the computer system, user only has to buy UDP through network service.
- Using the public key and private key of user, the data produced using the software can be encrypted and/or signed to protect the data.

4 Online scenario of software sale and delivery

Current study on EC is focused mainly on online payment and electronic money. In current EC model, online delivery of digital product is not used although online payment is executed. If concrete software protection mechanism is provided as proposed in this study, the production, sale and delivery of software can be executed via online service. A possible scenario of software sale and delivery is represented in Figure 4.

* Preparation

- Software developer develops software that is composed of UIP and UDP.
- UIP is distributed freely in advance through hardware manufacturers or online services.
- Software provider prepares online service system for generation of UDP, electronic payment and software delivery.
- User acquires UIP downloading from online service or buying computer system from hardware manufacturer.
- Software provider and user get their Certificates from CA.

* Online sale and delivery of software

- Customer connects to software provider system and orders a copy of software with his or her Certificate. In this stage, customer and provider are mutually authenticated using public key technology.
- Online electronic payment is done.
- Provider system generates UDP using customer's Certificate.
- UDP is delivered to customer system through network.

* Using the software

- The software is installed in user system simply by setting the connection between UDP and UIP.
- User has to present his or her private key to use the software.

The above scenario is an example of software sale and delivery using the proposed software protection system. Many other scenarios are also possible depending on PKI environment and customer condition. If the proposed scenario is widely used in software industry, huge amount of cost for manufacturing physical software package, delivering product and maintaining vender network will be reduced, lowering the price of software.

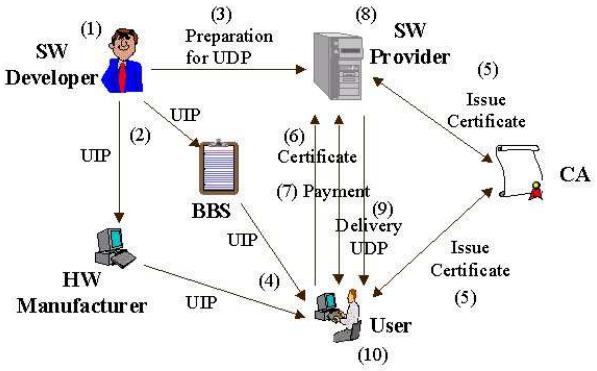


Figure 4: A possible scenario of software sale and delivery using the proposed software protection system.

5 Conclusion

A new software protection system using Public Key Infrastructure(PKI) and User Dependent Software(UDS) was proposed. Using the Certificate, the identity of customer is clearly authenticated during purchase process. An user has to present his or her private key to use the software. Since UDS is for only one user, copied UDS is of no use without legitimate user's private key. This system can be adapted immediately with no additional cost if PKI is widely established and used. Besides of the essential protection mechanism,

the proposed system provides with many characteristic features such as multiple software protection with the same key, free distribution of UIP in advance, installation in multiple computer systems, etc. Adapting the proposed software protection system, it is expected that the software industry will be changed very much.

In this paper we assumed the situation that PKI was widely spread over the society. Before the settlement of PKI, we can temporarily use another key management system such as saving private key as electronically protected key file. Still other way is that the software provider roles as a Certificate Authority(CA) and issues Certificate for customers. The security analysis of the proposed UDS mechanism and software distribution protocols will be further a challenging job for cryptanalysts.

References

[1] N. Hirose, et al., "A proposal for software protection", SCIS'98-9.2.C, The proceedings of the 1998 Symposium on Cryptography and Information Security, Shizuoka, Japan, Jan. 28-31, 1998

[2] A. Monden, et al., "A watermarking method for computer programs", SCIS'98-9.2.A, The proceedings of the 1998 Symposium on Cryptography and Information Security, Shizuoka, Japan, Jan. 28-31, 1998

[3] T. Kitagawa, et al., "Digital watermark for Java programs", SCIS'98-9.2.D, The proceedings of the 1998 Symposium on Cryptography and Information Security, Shizuoka, Japan, Jan. 28-31, 1998

[4] James R. Gosler, "Software protection: Myth or reality?", Advances in Cryptology – CRYPTO '85, 140-157

[5] A. Herzberg and S. S. Pinter, "Public protection of software", Advances in Cryptology – CRYPTO '85, 158-179

[6] O. Goldreich, "Towards a theory of software protection and simulation by oblivious RAMs", STOC 87

[7] R. Ostrovsky, "An efficient software protection scheme", Advances in Cryptology – CRYPTO '89, 610-611

[8] ITU-T Recommendation X.509, The Directory: Authentication framework, 1993

[9] Public-Key Infrastructure(X.509), http://www.ietf.org/html.charters/pkix-charter.html