A Software Protection Method Based On Computer Fingerprint And Asymmetric Encryption

Huashan Tan
School of Computer and Information Science
Chongqing Normal University
Chongqing, 400047, China
E-mail: 6510388@qq.com

Yang Yang
School of Economics & Management
Chongqing Normal University
Chongqing, 400047, China
E-mail: lilyangyy@163.com

***Abstract*—Software protection is an important research area in computer field. Combined with the traditional asymmetric encryption algorithm which provides the measures of protection, computer fingerprint was employed to identify the protection object. This novel software protection method is called AEA-CF, which means asymmetric encryption algorithm based on computer fingerprint. The computers used AEA-CF should be running on Internet so that the authentication information could be provided by server. The experiments show that this method could protect software effectively.**

Keywords- software protection; computer fingerprint; RSA

I. INTRODUCTION

Software protection is challenging of pirate because of it is easy to copy. Except the software protection law should be executed strictly, technical measures should be taken also.

To resist copying and running illegally is the main tasks of software protection from technical consideration. For a un-authority software, if it could not be executed correctly in a computer, then it is not significant to copy it. Therefore, software protection technique is to resist program running correctly in a manner, under the situation of un-authority user.

The traditional software protection techniques could be classified to software technique and hardware technique. Such as protections based on varied register codes are software techniques[1-2]. Such as protections based on varied softdogs are hardware techniques[3-5]. The main principle of these software techniques is to identify whether the user is authority. If it is authority, then the program continue to execute. Otherwise, the program should be terminated.

Some disadvantages are existed in traditional software techniques. Firstly, if pirate got the register code illegally, then the software which is not permitted to run can execute in any computer. For example, the pirate could get the register code from legal user. In addition, one legal user could run his/her authority software unrestrictedly almost. Secondly, even the techniques which register code combined with computer fingerprint is employed, the software protection may not be successful. Actually, different computer has different register code in this situation. The legal user's register code could not be used in any other computer. But the legality is judged on local computer, so that pirate has the chance to penetrate the register code. Such methods of static analysis and dynamic track could build illegal register virtual machine. The machine then could be used to find register code[6]. Finally, even the register code is encrypted, the software protection may no be successful also. After an encryption transformation is employed to register code, it's difficulty to build the legal register code from encrypted code. But methods of static analysis and dynamic track could find the protection module in the program. Furthermore, the protection module could be skipped, and the software could run illegally.

AEA-CF is proposed because of two reasons. One is the local authority check needs changing to network check. Through the checking by software developer's network server, problems such as serial number reveal and illegal register machine could be solved. Therefore, the first and second disadvantage described above could be avoided. Through the key module encryption, the problem of software illegal copying could be solved. In traditional protection, the key module of software isn't encrypted. And the form of key module appeared to all users is the same. After key module encrypted based on PC fingerprint, the key module should be download from developer's server. Therefore, the third disadvantage described above could be avoided. On the other hand, Internet is used widely today. Network checking is available.

II. RELATED WORKS

A. *Protected by register code*

Register code is the essential software protection mechanism. When the software needs to install or run, the authority module in the software will check the user input register code. If the input code is legal, then the software continues to run. Otherwise the software will terminate.

One improved method of this mechanism is to add username to register code. That is, the register code should be calculated by a function of username as below.

$$\textit{register code} = \textit{F1}(\text{username}) \qquad （1）$$

The function *F1* is an encryption algorithm. And the calculation is finished by local computer.

This mechanism and its improved have the first disadvantage above. Once you get the register code and username, you can copy or run the software in any computer unrestrictedly.

Published by Atlantis Press, Paris, France.

B. Protected by computer fingerprint

To solve the problem that *F1* is not related to the computer. *A* technique called "one PC-one register code" is proposed. This technique is based on computer fingerprint.

Computer fingerprint is called machine fingerprint also. It's the special information to identify this computer from others. Such information includes CPU ID, hard disk serial number and MAC address of network card, etc. All the computer fingerprints could be gotten by API functions provided by operating system.

"One PC-one register code" is essentially the improving of register code mechanism described in II.A. Through an encryption algorithm *F2*, different computer could get different register code. Its formula description illustrates below.

$$\textit{register code} = F2(\text{computer fingerprint}) \tag{2}$$

Not as *F1*, *F2* makes one PC has one code. Even pirate get other PC's register code, this code could not match his/her own PC. Therefore, the software could not run correctly.

But, *F2* could not resist the attack of "register virtual machine" technique. Due to the authority checking is performed in the local computer, pirate could penetrate the checking algorithm in the software. The penetrating result is the pirate gets the relationship between computer fingerprint and register code. According to this relationship, a program module called virtual register machine could be produced. After that, the virtual machine could build his/her own register code as pirate likes. From the view of this section, it's easy to know that *F2* has the second disadvantage described above.

Otherwise, *F2* has the third disadvantage also. Through static analysis and dynamic track, the protection module of the software could be found. Since the checking point could be found, therefore the pirate could skip this protection module, and run the software correctly.

C. Protected by softdog

Softdog is a hardware protection mechanism. It is an additional hardware attached to computer I/O interface. Generally, the interface is COM or USB. When the software is running, the software will check the softdog. Whether a user is legal will be judged by the communication between software and softdog.

Actually, softdog plays as the actor of register code. Because of softdog is a hardware, it's difficulty to copy. So this protection mechanism overcomes the first and second disadvantages describe above. But this mechanism has the third disadvantage described above. Through intercepting the communication, pirate could get the important communication data and get the principles of protection. Utilizing the modification of operating system interrupt, pirate could construct a virtual softdog. This virtual softdog can return the data that true softdog returned. Therefore, the software can go on to run due to this false softdog deceiving.

III. AEA-CF

From the discussion above, we could obtain two essential attributes of traditional software protection. The first attribute is the identity authentication. The goal of this authentication is to determine whether the software user is legal. If the authentication is successful, the software could go on to run. If the authentication is fail, the software will be terminated. The methods pirate attack this authentication is how to pass the authentication illegally. Another attribute is that the authentication module runs in local computer. Because of pirate could control local PC easily, therefore the level of attacking difficulty decreases heavily. In other words, pirate has opportunity to get the information which key protection module wants.

A. Algorithm of AEA-CF

AEA-CF proposed here is different essentially from these methods mentioned above. Its installation and running flowchart are illustrated by Fig. 1 and Fig. 2. There is difference between S3 and P3. User should login software developer's server using a legal account number in S3. But in P3, it's unnecessary to login. In S5, common key *K2* is calculated by RSA algorithm. In S6, the reason of using *K2* to encrypt and using *K1* to decrypt is that the calculation time of *K2* through *K1* is cost too much. If *K1* disappeared duo to some network server errors, the decrypt key could be build quickly again through algorithm F3 and PC fingerprint *A*.

AEA-CF avoids the three disadvantages pointed out in the introduction section.

Firstly, AEA-CF combines computer fingerprint and asymmetric encryption of protection module together. The secret key saved in network server and not saved in local computer. When the software is opened, it's necessary to get the key from network server. In this process, PC user doesn't care such works as key storage and register code input. User's identity is checked by network server when the software is installed. Therefore, the first disadvantage is avoided.

Secondly, key is not the same as computer fingerprint directly. *A* transformation algorithm *F3* is employed to obtain the key illustrated below.

$$K1 = F3(A) \tag{3}$$

Where *A* is the computer fingerprint. This transformation is finished in network server. It corresponds to the step 4 in the figure 1. Because of *F3* is saved and ran in network server, therefore local PC doesn't have any information about *F3*. Further, pirate is difficulty to get *F3* without the control of the server. *A* is extracted automatically and *F3* is produced in server. Decryption of protection module is finished by an asymmetric encryption algorithm called RSA. From this view, the second disadvantage is avoided.

The next, the third disadvantage could avoid also. AEA-CF encrypts core module using RSA when the software is installing. AEA-CF decrypts when the software is running. The encryption operation is finished by network server, and decryption operation is accomplished by local PC. Due to there isn't any authority authentication codes in local PC, so pirate could not run the software by skipping the authentication codes.

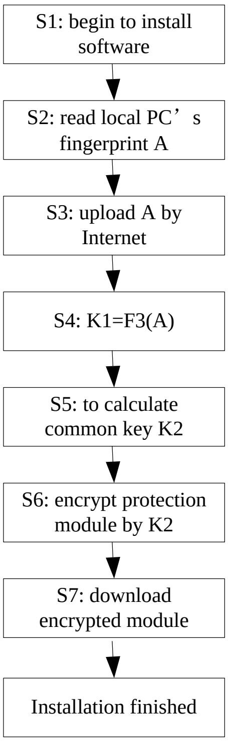
S1: begin to install software
S2: read local PC's fingerprint A
S3: upload A by Internet
S4: K1=F3(A)
S5: to calculate common key K2
S6: encrypt protection module by K2
S7: download encrypted module
Installation finished

Fig.1 Software installation and register flowchart

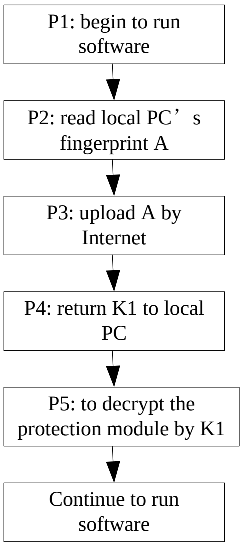
P1: begin to run software
P2: read local PC's fingerprint A
P3: upload A by Internet
P4: return K1 to local PC
P5: to decrypt the protection module by K1
Continue to run software

Fig. 2 Software running flowchart

Fourthly, AEA-CF solves the problem of illegal copying. When the core module needs to encrypt, the *F3* algorithm is employed to transform *A* to *K1* firstly, then common key *K2* is calculated by RSA secondly, and finally *K2* is used to encrypt the core module. Because of every PC has its own fingerprint *A*, thus *K1* obtained is different. This results directly that different PC has different encrypted core module. Furthermore, due to RSA is asymmetric, different encrypted core module can only decrypt by its own *K1*. When the pirate copied the installed software to another PC, the software can not run correctly due to another PC needs a different *K1*. Therefore, the illegal copy does not have any significance.

The next, AEA-CF increases friendship between software and user. In S3 of Fig. 1, user should submit legal login account to network server when the software is installing. The legal account may obtain through buying the copyright. When user login, the server will check user's identity. For example, whether the account is bought? Whether the software installed number exceed a certain value? Only user identity checking is OK, the software will go to next installing step. In other hand, pirate does not have legal account in server. So pirate can not install the software. Furthermore, only software installation needs user to input legal account. This account doesn't need to input when software is running. The account is stored in network server. Therefore, the security probability of legal account increased, and the user friendship of software increased.

Additionally, AEA-CF let software developer could do some copyright limits by network server. For example, one legal license could install the software one time. When the same license is used to install the software to any other PC, the *K1* corresponding to previous PC will be invalid automatically. When the previous PC runs the software, the PC could not get the original legal *K1* as before. In this way, AEA-CF ensures one license can only run the software in one PC at a time.

Finally, more than one core modules could be defined in the software for increasing protection level. *K2* will be used to encrypt these core modules. When the software is going to run, *K1* in network server is needed to check every time the core modules loaded.

B. Attributes of AEA-CF

To resist static tracking, AEA-CF may be improved. When the software is installing, the original S7 is "download encrypted module", it could be changed to "server stored encrypted module". When the software is running, the original P4 is "return *K1* to local PC", could be changed to "return *K1* and encrypted module according PC fingerpringt *A*". This improvement makes the core encrypted module should be download to local PC on line. The identity authentication codes don't save in local PC. Pirate could not run the software through static tracking. But, the core encrypted module needs to download every time when the software is running. This will bring three results. One is that the software's running needs Internet support with a higher speed of download. The next is that the open time of software may cost too much time. The final is that the network server should have enough space to store core encrypted module for every user.

To decrease the calculation time of server, AEA-CF may be improved. When user installs software, network server's calculation works includes: calculation of *K1* by F3, calculation of *K2* by RSA, and the encryption time of core module by RSA. It's necessary to finish these works as quickly as possible so that the software installation could continue to go on. Two measures could be employed to decrease the cost. One is large amount of *K1* and *K2* key pairs were calculated in advance. Of course, the calculation includes core module encryption. After that, relationship between *K1*, *K2* and core encrypted module is stored in server's database, the core encrypted module is stored in server's hard disk. When a PC needs to install the software, theses information use to identify user are existed already. Another measure is the definition of F3. One PC fingerprint *A* could correspond to one *K1*, or a group of many fingerprint *A* could correspond to one *K1*. When the fingerprint *A* was submitted to network server, *K1*, *K2* and the core encrypted module could be found quickly.

To utilize AEA-CF without Internet is possible. When the software installs, S3 and S7 may delivery PC fingerprint and encrypted core module by other media such as U disk and CD. When the software runs, *K1* will be calculated in local PC according to fingerprint *A*. Then *K1* is used to decrypt the core module. But if pirate could get this PC's fingerprint, pirate could run the software through simulating *A* in other computer.

IV. CONCLUSION

Compare to tradition software protection techniques, AEA-CF proposed has three advantages. One is that the traditional identity authentication in local PC is changed to network server. The next is that the core module isn't encrypted is changed to encrypted. The final is that one PC corresponds on encrypted module whatever the traditional module is encrypted or not. These advantages make software protection more security than before.

The disadvantage of AEA-CF is that Internet is needed when the software is installing and running. With Internet development, most PCs connect to Internet generally. Therefore, AEA-CF hits the strike nowadays.

ACKNOWLEDGEMENT

This work was supported by the Science Technology Researching Fund of Chongqing Education Committee (KJ100623, KJ110629).

REFERENCES

[1] SUN Yongqing, GU Yujie, ZHAO Ge. Software Encryption and Protection Based on Dynamic Register Code. Computer Engineering. June 2007, Vol.33, No.12, pp:183-184. In Chinese.

[2] Samuel Shu Kin Kwan, Jeevan Jaisingh, Kar Yan Tam. Risk of using pirated software and its impact on software protection strategies. Decision Support Systems, Volume 45, Issue 3, June 2008, pp:504-516

[3] Olga Gelbart, Eugen Leontie, Bhagirath Narahari, et al. A compiler-hardware approach to software protection for embedded systems. Computers & Electrical Engineering, Volume 35, Issue 2, March 2009, pp:315-328.

[4] Boneh D, Franklin M. Identity-based Encryption from the Weil Pairing. SIAM Journal of Computing, 2003, 32(3), pp:586-615.

[5] HAO Yao-hui, LIU Hong-bo, ZHENG Li, et al. Software Piracy-proof Method Based on USB Encryption Lock. Computer Engineering. December 2010. Vol.36, No.23, pp:119-120, 123. In Chinese.

[6] Matt Pietrek. An In-Depth Look into the Win32 Portable Executable File Format. MSDN Magazine, 2002.