Multi-User License Authentication and Validation System to Protect Against Man in the Middle Attack

G.Syam Prasad  
B.Tech,M.Tech,(Ph.D)  
Associate Prof & HOD CSE Dept  
Usha Rama college of Engineering & Technology, Telaprolu, vijayawada

G.Samuel Vara Prasad Raju  
M.Tech.,Ph.D  
Professor in CSE Department  
Andhra University, Visakhapatnam

Abstract –

Multi-user Software Authentication plays a vital role to authenticate and access software’s through online. Multi-user means multiple users are trying to access the software after purchasing through online with specified number of copies. Here copies indicate number of users to install and access the software through online process. In the existing work three tier process is used to authenticate multiple users and to validate users. In the three tier process, first tier is used to enter each user credentials for validation purpose. In the second tier user entered credentials are validated in the Tele-communication medium. After each user validation in the second tier, user registration details are sent to server for final authentication process. In this process each user is authenticated using three tier process which will take time and must posses Tele communication number. Tele communication framework is too expensive to implement. To overcome these problems a new two phase multi-user authentication framework is proposed. In the first phase, multi-user key registration process is proposed to each client registration verification and validation using proposed hash algorithm. In the second phase multi user activation code is verified to access software using proposed hash and encryption algorithm.

Keywords – NID, PRIVACY PRESERVING.

I. INTRODUCTION

Software Piracy is generally known as the illegal use of commercial software product, in other words using of copyrighted work. It can be a product for which the user is not just rewarded, and is understandably a major touch upon in the computer market. Even though the threat of legal action is possible against corporate pirates. But it’s so not so effective against persons pirates the personal computer software. The primary objective of all protection plans is to increase the costs of pirates to break the security. For that reason the higher the cost for the pirates to break the software security, vice versa the higher the security level of the application. Software is an intellectual property, protected from illegal users in an effort to ensure that the existing earning flows. Software piracy keeps growing as global threat as it is cheap and easy to use. Increasing in software piracy is devastating as it not only reduces revenues but also results in less R&D, and in less investment in marketing and other channels. A method is proposed in this work for software protection, treated as most popular types of software piracy based on using cryptographic techniques.

Different Types of Software Piracy

The Business Software Alliance explains five well-known types of software piracy:

- End-user piracy happens when an end-user reproduces duplicates of software without ever having authorization. It can present itself in one of the following forms:
  - A user acquires single licensed copy and makes uses of it to install the application on multiple computers.
  - The disks designed to install the software are copied multiply and distributed.
  - Without having previous version users buys upgraded one and uses.
  - Within a business environment, company employees use software with an academic license.
  - Client server piracy happens whenever program is installed on a network and at the time used by multiple users than the licensed entitled.
  - Internet piracy happens when unauthorized/duplicate copies of licensed software is made available in the internet sites for free as torrents, Peer-to-peer network sites which enable the transfer of illegal software or by exchanging uploaded programs.
  - Software piracy happens when licensed software is illegally replicated and sold same as the original.
This can further categorized as two types - unintentional and intentional pirates. Unintentional pirates happens when individuals purchases applications without aware of licensing and registration issues. Individuals with less software development practices and who don't understand the ethical practices of using software are part of unintentional pirates. It can be the employees of a corporate companies, uses licensed version of company in his or her personal computer for free. Sometimes the employees of the organization uses pirated software in the company computer because of not having an active security policy in the company, by putting their employers at risk for using unlicensed software.

[1]A multilevel security (MLS) system has two primary goals: first, it is often created to prevent unauthorized personnel from accessing information at higher classification than their authorization. Second, it can be created to prevent personnel from declassifying information.

Multilevel authentication model applied by sensitive applications. Moreover, this product is definitely one that belongs and applies multilevel security. As we know, any sensitive application includes confidential and secret information and must be used effectively in complicated and authenticated procedures. Suppose that the applying involves a variety of different users U=u1,u2,….un , so these users must do business in different authentication sensitive levels L=l0,l1,….,lm .

Lot of research has been investigated a variety of technology to reduce the multi user credentials in the user authentication procedure. Since humans are more adept in remembering graphical passwords than text passwords [2], many graphical password schemes were designed for single and multi user authentication process to address human’s password recall problem[3]–[6]. Using password management tools is an alternative [6]–[9]. These tools automatically generate strong passwords for each website, which addresses password reuse and password recall problems.

II. RELATED WORK

One of the simplest and most popular protection mechanisms consists in a password or key check that enables installation of the software. If the check fails the software is not installed or it works in demo mode with restricted functionality. This mechanism is very popular in shareware. The password (or key) validation function is, evidently, included in the software. Therefore, it is possible to find it using reverse engineering. As a consequence it is frequent that key generation programs are produced by dishonest users and also that authentic passwords are published in certain Internet sites. Sometimes the software is personalized to be used in one computer, for example, extracting information from some of the hardware devices (hard disk, network adapter, etc.) or from the operating system configuration. During its execution, the protected software checks that the computer is the one it was personalized for. This check, as the previous ones, can be bypassed. Also, this mechanism is inconvenient for the users because changes in the hardware or in the operating system may result in the need to get a new license and reinstall the software.

More recently, Aura and Gollman presented in [AuGo99] an interesting scheme based on smart cards and digital certificates that solves the card juggling problem and provides mechanisms for license management and transfer. In addition, a compilation of countermeasures against attacks are reviewed. Unfortunately, as their proposal is focused on the check of the presence of the smart card, it is vulnerable to the code modification attacks as shown below fig 1.

![Fig. 1. Code modification to bypass the check of the presence of the token.](image)

A secure software protection scheme can be designed using just smart card technology. In this scheme some sections of the software to be protected can be substituted by functionally equivalent sections to be processed in the smart card. In this way, the protected software is divided and will not work unless it cooperates with the right card. Code modification attacks will not succeed in this case. In fact, the only possible attack is to analyze the data transmitted to and from the card trying to guess the functions that the card performs. If we include enough functions, with enough importance in the main code, and enough complexity, the attack described could become impractical.

This scheme needs one card per application and the quantity and complexity of the protected functions are limited by the capacity of the card. Moreover, this scheme does not allow the distribution of the protected software using Internet because the cards must be distributed with the software. With the purpose of avoiding the aforementioned problems we will introduce the cryptography as the second building block of our software protection scheme.

In the authorization phase (equivalent to the personalization phase of the previous scheme), a new license is produced containing the random symmetric key used to encrypt the protected sections, information
about conditions of use (i.e. time limits, number of executions, etc.), the identification of the software (ID, version number, etc.) and finally the identification of the license. All this information is encrypted with the card public key. When the license is received by the client it is stored in the card. The functionality of the previous scheme is maintained in this new one, but the efficiency is improved because decryption of the protected sections is now much faster. The definition of the license structure permits a high degree of flexibility. Furthermore, as each application has its own key, we can manage them individually.

III. PROPOSED SYSTEM

Multi User Activation Key Registration:

**Input:** Hardware parameters, Number of licenses.

Step 1: Root User requests multi users id, copy id using root user hardware system information. Client sends E((Licenses no,Hardware information),k) to server.

Step 2: Server sends Multi User-ID and Copy-ID to the Root user through SMS or Mail.

Step 3: Root user sends Multi User-ID and Copy-ID to the server for validation.

Step 4: Server validates both ID’s and if both are valid then it generates root client activation code as software key for installation. If either one is invalid then installation fails.

Step 5: Root user gets activation key along with second user activation details like activation code, one time random nonce and Copy-ID in encrypted form.

Step 6: Root user enters activation key for software installation. If the entered activation key is valid then installation process succeeded else activation key already exist or invalid.
Step 7: This process is repeated sequentially to all other users.

VI RESULTS:

This experiment is carried out using Microsoft windows 7 and web server acts as a commercial server. Client registration process and installation front end is developed using Java swings.
Fig 4: User Request Key to Server

Fig 5: User enters his MultiUserid and Copy ID

Fig 6: Activation key is generated from the server to client.

Fig 7: Root user enters his Activation code for installation

Fig 8: Other license users interface this form for installation and validation

Fig 9: Successfully register five users list in company database.

SYSTEM 1

Activation code generation time (s) = 16.671238248
Activation code generation time (s) = 9.677896172
Activation code generation time (s) = 14.512361521
Activation code generation time (s) = 8.647310742
Activation code generation time (s) = 8.044746958

SYSTEM 2

Activation code generation time (s) = 23.884053735
Activation code generation time (s) = 11.664914169
Activation code generation time (s) = 9.032172435
Activation code generation time (s) = 6.515497585
Activation code generation time (s) = 7.439875624
SYSTEM 3:

Activation code generation time (s) = 18.855267182
Activation code generation time (s) = 6.946740224
Activation code generation time (s) = 13.190916348
Activation code generation time (s) = 8.520650722
Activation code generation time (s) = 7.542168130

Performance Analysis:

<table>
<thead>
<tr>
<th>CLIENTS ACTIVATION CODE TIME (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIENT1</td>
</tr>
<tr>
<td>SYSTEM1</td>
</tr>
</tbody>
</table>

Performance of each client for activation code generation and installation

V. CONCLUSION

In this research work, multi-user license management is performed with less time. This approach is successfully performed in two operations i.e multi-user registration and multi-user validation. Experimental results show each client access time is less and performed well to access the software. This approach successfully overcomes the middle in the man type of attack as we proposed robust hashing approach which is easy to identify the each user integrity. This system uses proposed homomorphic algorithm for transmitted data between client and server.

REFERENCES


