Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 66 (2014) 20475-20477

ISSN: 2229-712X

Watermark, Hardware Parameters and License Key: An Integrated Approach of Software Protection

Nishant Gupta, Shubhnandan S. Jamwal and Devanand

Department of Computer Science & IT, University of Jammu, India.

ARTICLE INFO

Article history:

Received: 24 May 2013;

Received in revised form:

19 December 2013;

Accepted: 31 December 2013;

Keywords

Security,
Watermarking,
Hardware Extraction,
Software Piracy,
Fragile Watermarking.

ABSTRACT

Software protection is an area of active research in which the software industry is encountering a number of threats. Software piracy is one such threat, which proves detrimental in protecting the intellectual property rights. There have been a variety of techniques developed to address the issue like software watermarking, code obfuscation, and tamper-proofing. In the current research we address the issue of software piracy through a prevention technique known as software watermarking which aims at providing copyright protection and authorized access of commercial software. In this paper fragile software watermark is used to embed personal information into the software. Then this personal information is merged with the hardware parameters of the client machine extracted during the process of installation and License key provided by the vendor. This combined string (Watermark + Hardware parameters + key) is send to the server for registration. This process is implemented and tested on different machines and the accuracy of the proposed model is found to be 99%. The proposed model will be beneficial in combating software piracy and securing the software from redistribution.

Introduction

In this technology-driven era, revolutionary progress under the domain of hi-tech scientific and industrial arena, we can observe massive improvements such as advent of internet and high end computing resources (such as Grid) have broadened the scope of research. But at the same time, they have lead to the issues like software piracy. Currently, software piracy is a major problem for software developers. Techniques are being developed and employed to control software piracy [2], [3], [7], [8], [9], [10], [11], [12]. Various schemes have been proposed and put in operation to minimize the impact of software piracy by limiting unauthorized modification. But these techniques have a minimal impact on proliferation of software piracy. Due to the proliferation of software piracy, software protection is increasingly becoming an important requirement for software development [1]. Also, the problem of protecting software from illegal copying and redistribution has been the focus of considerable research motivated by billions of dollars [13] in lost revenue each year (Fig. 1).

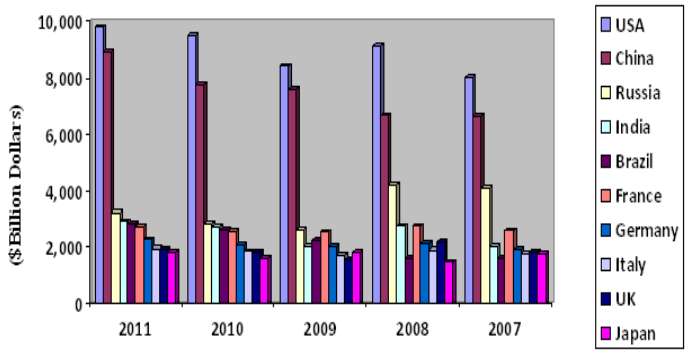

Fig. 1 Commercial Value of Pirated Software of Top 10 Economies

In the unfortunate event that software is illegally redistributed or an important algorithmic secret is stolen, an owner would like to be able to take action against the theft. This requires demonstration of ownership and/or identification of the source of the illegal redistribution. A technique which enables such action is *software watermarking* which has been the core aspect of this paper.

In this paper, we proposed a new model for securing the software from being pirated. It has been divided in four phases. First phase includes embedding a watermark in the software, second phase involves the extraction of hardware parameters, third phase is registering the software with server and the fourth phase is concerned with the continuous checking of software for verification and validation.

Software Watermarking is currently the most demanding technique to prevent the software from piracy [5]. From the last few decades, the threat of piracy and illegal use has become a key concern for software industry and many algorithms and approaches were designed to overcome this problem. We have shown through this paper that the goal can be achieved with software watermarking technique and propose an efficient privacy enhanced software registration scheme.

Related Work

Many approaches are proposed and implemented to prevent software piracy. Some people gave the concept of robust watermarking technique to prevent software piracy [2]. The SMS based gateway technique was used in this paper where an automation process is required as a manual response for each software. S.Mumtaz et. al [3] in her paper has relied on the extraction of hardware characteristics of client machine while registering for software which is not well enough for illegally distributing the software invariably. The paper [6] was based on static software watermarking techniques which are highly susceptible to semantics preserving transformation attacks and are

Tele:
E-mail addresses: nis.decent@gmail.com

therefore easily removed by an adversary. This paper had future inclinations towards the use of dynamic software watermarking algorithms.

Z.Jian-qi et. al. [7] presented a novel robust dynamic watermarking scheme based on STBDW that first utilizes the Shamir Threshold Scheme to split the watermark number into pieces, which help to retrieve the original watermark with partial information and increase resilience, then the encryption is done and self isomorphic mapping are embedded into dynamic branch structure of the program, which can resist most semantics-preserving attacks. However, these techniques are susceptible to statistical attacks. J. ZHU, J. Xiao, and Y. Wang [8] introduced Fragile Watermarking Algorithm which was implemented on Software Content Management. This paper solves the defects and problems of Software Version Control and Software tamper-proofing, still there is more need of perfection while embedding the watermark with compiler program.

In [9] and [11], the authors have used the watermarking techniques limited to tamper-proofing and copyright protection only. Y.Zhang et. al. [12] proposed a software-splitting technique which encrypt the extracted contents from the software by a key relating to the hardware characteristics and then decrypt them dynamically during the main program running. This is rather complex for developer and programmer too. Future scope demands for the use of watermark with the source codes so as to make it more secure. F.Donglai et.al. in his paper[10] has limited its research of software protection to the software programs written in jME API. Future research inclines towards the use of watermarking technique according to the type of software.

Software Watermarking

Software watermarking is used to embed additional information in a piece of software in order to encode identifying information. However, for software watermarking to be useful it must be resilient against a variety of attacks, e.g. semantics-preserving code transformations and program analysis tools. Software watermarking takes the approach of discouraging piracy through a program transformation which embeds a message (the "watermark") into the program. Each watermarking algorithm is categorized based on a set of characteristics. Software watermarking algorithms [6] and [7] are classified in different classes depending upon their goals, extraction, execution and implementation. These watermarking algorithms are further classified into [2] robust and [8] fragile techniques. Of these, Fragile Watermarking technique has been used in this paper for embedding the watermark in the software.

Proposed Technique/Model (WHLK)

The current research proposes the design, development and implementation of a model for controlling the software piracy (Fig. 2). The design and development of this model is discussed in four phases

Phase 1: Embedding a watermark

The identification of client which comprises of Name (n), Affiliation (af) and Social Security No./ID (ssn) is inserted as watermark (W) into the software (S).

$W = n + af + ssn$

The fragile software watermarking technique is used for embedding the client identification into the software which validates the copy of the software for a client. This technique is preferred because in other techniques the water marks once introduced can be modified. The technique used in our model clearly states that if semantics preserving transformations are performed on the software (S) the watermark (W) becomes invalid which means that once the transformation has been made, no other change is possible in any way.

$SW = W + S$

Once this process is complete, the watermarked software (SW) is delivered to the client.

Phase 2: Fetching the unique hardware parameters of the client machine

When the process of installation starts on client machine the hardware parameters (HW) of the client machine are extracted. These parameters includes BIOS (Basic Input Output System) information (BIOSP) such as Name (BN), Manufacturer (BM), Date (BD), Serial Number (BS), MotherBoard Information (MP) such as Installed Date (MD), Manufacturer (MMU), Model (MM), Serial Number (MSN), Version (MV) and Other Parameters (OP) like Processor ID (PID), HardDisk ID (HDSN), Media Access Control Address (MAC).

$BIOSP = BN + BM + BD + BS$

$MP = MD + MMU + MM + MSN + MV$

$OP = PID + HDSN + MAC$

On combining (1), (2) and (3)

$HW = BIOSP + MP + OP$

Then the License Key (LK) of the software is introduced during the process of installation.

The extracted hardware parameters are merged with the watermark and License Key of the software to create Registration Code.

$RGC = W + HW + LK$

Phase 3: Registering the Software at Server

After installation, RGC is sent to the activation server for registration and activation. On submitting this watermarked information, the activation message is acknowledged to the client. The information is stored at server and whenever the software is re-installed on the machine, it will check the information of that machine and match it with the existing data at server for validation.

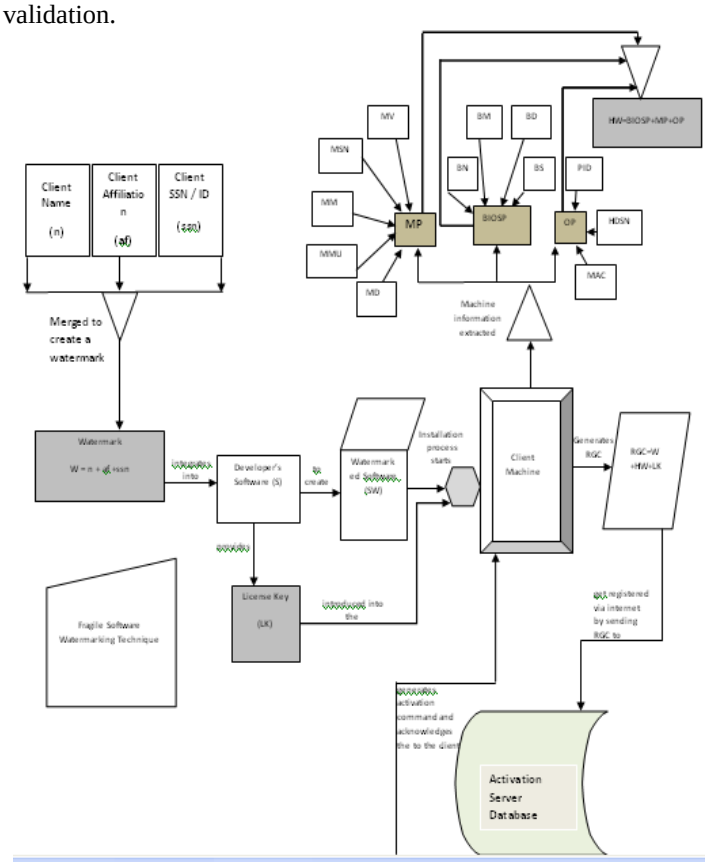

Figure 2 Proposed Model - Whlk

Phase 4: Checking the software after sixty days

This process is mandatory when the client changes his/her machine or one/few components of the existing machine. This

process is required in case of hardware failure of the client machine because the copy of the software remains the proprietary of the client.

While registering the software on the new machine, the same process is applied as registering the software for first time. The SW lies with the client only. The new hardware parameters (NHW) are extracted and the LK is introduced during the installation of SW. A new registration code (RGCN) is generated on merging the extracted hardware parameters with the watermark and the license key

RGCN = W + NHW + LK

The algorithm designed is explained in the following steps:

1. Input Watermark W=n+af+ssn in the software S.
2. SW=W+S.
3. Install SW on the client machine.
4. Fetch the hardware parameters HW=BIOSP+MP+OP of the client machine.
5. Introduce the License No. (LK) of software.
6. Create a string RGC=W+HW+LK for registration at the server end.
7. Submit RGC to the server.
8. An activation message was acknowledged to the client.

Implementation and experimentation

The designed process is tested on Intel Core (TM) i5 CPU 650 with Bus Speed of 3.20 GHz., 500 GB Secondary Memory with 5400 RPM, 2 GB Primary Memory and operating system of 32 Bit Windows 7 Professional. The process has been tested on 10 different machines with the same configuration for its reliability and accuracy. On these machines the process is executed 10 times and the average of the observations is found. The accuracy rate of the test is 99%. Inaccurate results are found on those machines on which this application could not be properly installed due to some hardware problems on the client machines.

The analysis of the results shows that the methodology/algorithm used for curbing the software piracy has been proved to be correct, accurate, and secured.

Conclusion And Future Scope

The technique proposed in this paper was tested, verified and implemented on number of machines. The rate of accuracy of our model is found to be 99%. The SW purchased cannot be installed on client machine without the verification and validation of the watermarked information. If anybody wants to pirate the copy of software of the client on its machine, the proposed technique does not allow him/her to do so, if implemented. This has given an opportunity to client to purchase the software and use it without the risk of redistribution of software to others. By doing so, intellectual property of the developer and value for money of the client, both are protected.

References

[1] Falcarin P, Collberg C, Atallah M, Jakubowski M. Software Protection Guest Editor Introduction; IEEE Software: IEEE Computer Society; 2011; 28: 24-27.

[2] Nehra A, Meena R, Sohu D, Rishi O P. A Robust Approach to Prevent Software Piracy. Students Conference on Engineering and Systems: IEEE; March 2012; 1-3.

[3] Mumtaz S, Iqbal S, Hameed I. Development of a Methodology for Piracy Protection of Software Installations. 9th International Multitopic Conference: IEEE; Dec. 2005; 1-7.

[4] Collberg C, Thomborson C. Watermarking, Tamper-Proofing, and Obfuscation-Tools for Software Protection. Transactions on Software Engineering: IEEE; Aug. 2002; 28 (8): 735-746.

[5] Jamal S, Zaidi H, Wang H. On the Analysis of Software Watermarking. 2nd International Conference on Software Technology and Engineering: IEEE; Oct. 2010; 1: VI-26-VI-30.

[6] Hamilton J, Danicic S. A Survey of Static Software Watermarking. World Congress on Internet Security: IEEE; Feb. 2011; 100-107.

[7] Jian-qi Z, Yan-heng L, Ke Y, Ke-xin Y. A Robust Dynamic Watermarking Scheme based on STBDW. World Congress on Computer Science and Engineering: IEEE; 2009; 7: 602-606.

[8] ZHU J, Xiao J, Wang Y. A Fragile Software Watermarking Algorithm for Software Configuration Management. International Conference on Multimedia Information Networking and Security: IEEE; Nov. 2009; 2: 75-78.

[9] Shengbing C, Shuai J, Guowei L. Software Watermark Research Based on Portable Execute File. 5th International Conference on Computer Science and Education: IEEE; Aug. 2010; 1367-1372.

[10] Donglai F, Gouxi C, Qiuxiang Y. A Robust Software Watermarking for jMonkey Engine Programs. International Forum on Information Technology and Applications: IEEE; July 2010; 1: 421-424.

[11] Shao-Bo Z, Geng-Ming Z, Ying W. A Strategy of Software Protection Based on Multi-Watermarking Embedding. 2nd International Conference on Control, Instrumentation and Automation: IEEE; 2011; 444-447.

[12] Zhang Y, Jin L, Ye X, Chen D. Software Piracy Prevention: Splitting on Client. International Conference on Security Technology: IEEE; 2008; 62-65.

[13] Business Software Alliance. BSA Global Software Piracy Study. Available from: http://globalstudy.bsa.org/2011/downloads/study_pdf/2011_BSA_Piracy_Study-Standard.pdf.