Design and Implementation for Piracy Protection Software

Moaz Mahmoud Adam & Yasir Abdelgadir Mohamed
Collage of Computer Science and Information Technology, Karary University

Abstract: Recently, software piracy has raised up as a major issue for software industries as they have to pay a lot to protect their applications. There are several assurance schemes for software protection. But they all are very costly and provide services for a limited period of time. This research will demonstrate a protector technique which can be developed in any platform very easily so that the vendor can protect the software not only for being pirated but also capable of limited period expiration of software. This strategy will provide easy and low cost protection which is more effective than MAC address authentication. This technique is very useful for desktop applications which use internet for their internal working, as it provides server based authentication, but is also feasible for those applications that do not use internet as it has client side authentication also.

1. Introduction
Piracy has become so prevalent over the Internet that poses a major threat to ecommerce sites. With the help of malicious codes and programs, hackers or an intruder can gain access to the system and steal the information [1]. Hence there arises a need to protect the information and products from being plagiarized [2]. This research is developed for the same purpose to protect the software's ownership of copyright and make transactions securely. Traditional methods for software protection are not enough effective [3]. Because of the possibility to use the same key twice or by using a key generator etc. this can make the software license to work in more than one device. The main aim of this research is to maintain software copyright protection and assures that it is being accessed only by the authenticated users. Making the software more secure upon cracking, (1) Utilize the MAC address of the device and one of the encryption algorithms and java, a demo program will be initiated and a license class will be created. The class will use the MAC address of the device to create the license. In other word license generation will depend on the MAC address of the user PC to make the application safeguard and protected against software crackers.

Positive Impacts for the proposed approach are, Protection of stealing the copyright ownership, Increases of making revenues of the software industries, No one can copy the software or share as it requires an activation code that is different for each individual, highly robust and secure system. The system can be used by any online software dealer. It can also be used by organizations and business owners to promote their product and at the same time protect their copyright.

2. Background
Licenses are important tools for setting specific terms on which software may be used, modified, or distributed. Based on the copyright protection automatically granted to all original works, a software license essentially, a set of formal permissions from the copyright holder may include specific “conditions” of use, and are an important part of the legally binding contract between program author (or rights owner) and end-user. Without a license agreement, software may be left in a state of legal uncertainty in which potential users may not know which limitations owners may want to enforce, and owners may leave themselves vulnerable to legal claims or have difficulty controlling how their work is used. This is equally true for software that is commercialized and offered for a fee, and software that is made available without cost to others. While end-users often balk at overly restrictive software licenses, the uncertainty caused when no license is given can also discourage those wishing to make use of a piece of code. It is important to note that licenses can be used to facilitate access to software as well as restrict it.

For a license to be valid it must be granted by the owner of the work's intellectual property (IP) rights. Under the policies of most academic and research institutions, researchers who have created a piece of software are unlikely to own full rights to their works. Instead, the institution generally holds or shares legal right to developed software. Institutions' policies on IP ownership vary, but in most cases your institution will be the legal “rights owner,” and will
be the entity that actually grants the license you choose for your software. Although many types of licenses, especially of the “free and open source” variety, are simple enough for the non-legal expert to understand and apply, it is generally necessary to consult your institution’s before imposing a license. See below for more information about working with your institution in applying a license.

The intellectual property rights cover three distinct sets of rights: trademark, patents, and copyright. From the view of the law (Besen & Raskind, 1991). Copyright refers to types of merchandise information/intellectual property goods, having artistic features [4]. Information goods have two important public goods characteristics. First, their consumption is originally non-rival. That is the use that one person makes of a piece of information does not decrease the possibility of use by others [5]. Second, information and intellectual property goods may be non-excludable in the sense that the producer of the intellectual property goods is often unable to make an exception of nonpayers from consuming goods without due authorization (Varian, 1998). The law of Intellectual property responds to this problem by giving producers certain exclusive rights that exclude nonpayers from certain uses of their intellectual property goods [6]. Although, assigning IPRs is not the only way to deal with exclusion (for example, bundling) [7]. IPRs law recognizes that no exclusion would create insufficient motives for the creation of IP goods. But the problem is that permanent intellectual property rights would lead to the standard deadweight loss of a monopoly. For that a proper IP system must ensure a fair balance between these two conflicting objectives [8]. As regards intellectual property protection, one of the biggest concerns for copyright holders is piracy [9]. The piracy can be defined as is the unauthorized use of copyrighted goods. Anyone wants to use the original product legally must pay [10]. For example when an individual break the software protection and distribute it that is a type of piracy [11]. Even though piracy occurs for all types of intellectual property and can take many forms depending on the access type and intellectual property mechanism (Watt, 2001), the piracy of software applications is considered one of the biggest issues at the present time [12]. Business software piracy has been related to economic growth (Andrés & Goel, 2012), shadow economy (Goel and Nelson, 2012) scientific output (Asonou, 2014), innovation (Banerjee & Chatterjee, 2010), and industry profits (Gomes et al., 2013). It has been calculated that for each authentic copy distributed there are up to 10 illicit copies downloaded from internet or copied from friends or members of family (Reavis & Rumelt, 1991). According to BSA (2011 p. 1) "the global piracy rate for PC software hovers at 42 percent the commercial value of this shadow market of pirated software climbed ... to $63.4billion in 2011". Consequently, investigating the determinants and effects of software piracy has been paramount object of empirical studies over the last decade (see e.g. Andrés, 2006a,b; Goel & Nelson, 2012; Bezmen et al., 2006; Chen et al., 2010; Arai, 2011; Boyce, 2011)1. By and large, related studies show that economic development, institutional arrangements, political regimes and cultural proxies are determinants of ‘softlifting’ behavior on a cross-country level. Our research offers a quite different avenue in understanding the cross-national variations in software piracy rates. We depart from a celebrated article by Lynn & Vanhanen (2002 p. 194) who claim that “national [intelligence levels] are a causal factor responsible for the differences in economic development”. Based on conclusions formulated by Lynn & Vanhanen (2002) we conjecture that intelligence may be important antecedent of software piracy through which it has impact on economic growth and innovation. Notably, we presume that there are a number of channels through which intelligence is related to software piracy, the first of which is economic development. Related literature reports that economic development is one of the most robust predictors of software piracy rates. Economic wealth, proxied by GDP per capita, is statistically significantly and negatively associated with cross-national piracy rates (e.g., Andrés, 2006b; Andrés & Goel, 2012; Bagchi et al., 2006; Robertson et al., 2008). On the other hand, in their celebrated articles devoted to the understanding of intelligence Lynn & Vanhanen (2002, 2006) suggest national IQ as an explanation for cross nations variations in per-person gross domestic product (GDP) and other country level economic outcomes. Similarly, Ram (2007), using data for 98 nations, reports that IQ has statistically significant effect on economic growth. As cognitive abilities have positive effect on economic development, we may conjecture that intelligence will be inversely related to software piracy rates. More recently, Meisenberg (2012) p. 103 concludes that "high IQ is associated not only with high per-capita GDP ... but also with more equal income distribution". The general categories of software protection: 2.1. Software watermark Software watermarking can be defined as the process of embedding additional information into software, without interrupting the functionality of the software itself. The earliest patents were filed in 1994, based on the concepts of software watermarking. The watermarking are methods for, identifying unauthorized copies and; to provide a method for identifying the source of unauthorized copies. In another patent in the same year, an assignee claiming a method and apparatus for serializing and validating copies of software, and
thus, possibilities of disabling the functionalities of the software whenever an unauthorized copy is found.

2.2. Fingerprinting

Fingerprinting is basically the same as watermarking, except that fingerprinting embeds unique identifier information on each distribution copies of software. This may not only detect an occurrence of software violation copies, but also able to trace the violator. A fingerprint may include vendor, product or customer information.

2.3. Software birthmark

One of the less popular methods on securing a copy of software is called software birthmark. It has quite a different approach compared to software watermark. The general concept of software birthmark is the same that is found in the computer virus signature concepts; to produce a unique identification of software. There are two important characteristics that differ between the software watermark and software birthmark;

In software watermark, it is often necessary to embed external information or data or code within carrier software, whereas it is not required in software birthmark;

Birthmark could not be used to identify ownership, or source of distribution but rather to confirm that software or code whether it is in partially or in fully, is a reproduction of others.

Table 1. Different software piracy protection techniques

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Change to original content</th>
<th>Concept</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOFTWARE WATERMARK</td>
<td>Yes except zero watermark</td>
<td>Embeds additional information</td>
<td>Copy protection, ownership proof, detection, integrity check</td>
</tr>
<tr>
<td>FINGERPRINTING</td>
<td>Yes</td>
<td>Embeds unique identifiers to each</td>
<td>Tracing violator, identifying source of distribution</td>
</tr>
<tr>
<td>SOFTWARE BIRTHMARK</td>
<td>No</td>
<td>Generate unique identifiers to</td>
<td>Proof ownership, To determine whether a software or a code either as fully or partially, is a reproduction of others</td>
</tr>
</tbody>
</table>

Choosing an appropriate licensing scheme is crucial for the success of a software product. When choosing the wrong licensing scheme, potential customers don’t even bother licensing your software, even if your product is superior to competitive solutions. Furthermore, an appropriate licensing scheme can pave the way for interesting and successful marketing strategies. So, one can say that the success of a software business rises and falls with the decision of how to license its software.

When we thought about the licensing scheme for SmartInspect, we naturally decided to license it per developer, as SmartInspect is a software development tool. But as we thought about it, we had an idea about an alternative license that now proofs to be quite successful. But more about that in a minute. Why You Should Care although the decision of which licensing scheme to use is only one aspect of many when selling software, it can greatly influence other parts. Tasks like marketing or choosing your market are not independent from the choice of your license. They affect each other.

As an example, depending on your licensing scheme, your software product might be either attractive or completely useless for an entire market. And in the latter case, if this market is the one you wanted to reach in the first place, then your software product probably won’t have a bright future and you might have a serious problem.

Additionally, you should be aware of the fact that you need to choose your licensing scheme early in the development process. You need to know all the technical consequences of your license. For example, you need to know exactly if and how you want to enforce your license when you plan the application.

3. Common Licensing Schemes

There are several well-established licensing schemes for software. Most of them depend on the software’s architecture and the intended market. This section presents the most common schemes including a short description

3.1.1. Licensed per named user

Licensing per named user means that the license is tied to a particular person. This license is popular for web applications and also for many shrinked-wrapped desktop applications. For desktop applications, it is usually allowed to install the software on multiple computers at once, as long as the software will only be used by one person.

3.1.2. Licensed per installation/computer

This license allows the installation and usage on one computer. You need multiple licenses if a person is using the software on more than one computer. On the other hand, it is usually allowed that multiple persons use the software as long as they use it on the same computer.

3.1.3. Licensed per client

This concept is usually used in client/server architectures. You normally need to acquire a license for each client (called Client Access License or CAL in short) and in most cases the server requires an additional server license. This licensing scheme can be used with the Microsoft SQL Server, for example. If I understand it correctly, you need an access license not only for each client connecting directly to

Imperial Journal of Interdisciplinary Research (IJIR)  
Vol-3, Issue-3, 2017  
ISSN: 2454-1362, http://www.onlinejournal.in
the SQL Server, you need one for each client you store data about.

3.1.4. No license fees at all

This is the often case when dealing with open-source or freeware software. It is also increasingly popular when selling support for a particular application. Selling the software is far more profitable than selling the software itself.

Free software applications can be very useful for marketing. For example, it is one part of the marketing strategy for SmartInspect and this is the idea I just talked about. We decided to offer a feature-reduced Personal edition of SmartInspect and it’s already a success.

We hope that the Personal edition will have two effects: On the one hand, we hope that we get a bit more buzz about SmartInspect and that the word of mouth effect makes it more popular. We hope that users who like the Personal edition and need the more advanced features will upgrade to SmartInspect Professional.

3.1.5. Licensed per developer

Licensing per developer makes sense if you are selling development tools or components/libraries. You need to acquire a license for each developer using the tool or component. When you deploy your application that uses the component, end-users normally do not need a license. This is also called royalty-free.

We decided to use this licensing scheme for the Professional edition of SmartInspect. This makes it very easy and affordable to license for small and big software vendors alike.

3.1.6. Site license

A site license means that a software product can be used by all persons on the same ‘site’. A site can mean all persons in the same building, the same physical address or all persons in the company. This license scheme is often not used because of possible discounts like many people would think, it is used to make the licensing and license control easier for the purchasing party.

Imagine you have a software product that is licensed per user. You now need to license all the sales staff in your big corporation. It can be quite hard to find out exactly how many licenses you need to get and how this number changes over time. One way to make sure that you licensed all your users is by using some sort of license management software, just buy more licenses than you need or simply get a site license.

3.1.7. Floating license

A floating license means that you are free to use the software on multiple computers by multiple users. You just have to make sure that only one user uses the license at the same time. A floating license can make sense if you have a very expensive software product that the users only need from time to time. Floating licenses are usually offered in addition to other licenses and are more expensive in most cases.

3.1.8. Royalties per item sold

This license scheme is only appropriate for software products that are part of other software applications or are sold as a part of a package. The licensee has to pay royalties to the license issuer for each end-user. For example, take the embedded market. Imagine a robot for the automobile industry that has a micro controller and a special software application. When the vendor sells one of these robots, he has to pay a royalty to the original software vendor.

In these special cases it can make sense to license software with royalties, because there are only a limited amount of items sold which are highly specialized and expensive. For more general software components, this might not work so well.

For example, if you collect royalties for your software development component, many software shops simply won’t buy it. They do not want to pay you money for each license they sell. On the other hand, there are some areas where royalties are accepted and work (embedded market, highly specialized and expensive components etc).

3.1.9. Marketing strategies

It makes sense to think about possible marketing strategies when designing a licensing scheme. In the shrinked-wrapped software market it is not too uncommon to offer a feature-reduced version of your application or an edition without support for free to generate word of mouth marketing. You may want to consider this for your own software product.

Another possible marketing strategy mainly concerns software applications which are based on a client/server architecture. Depending on the targeted market, it might be a good marketing strategy to offer the server and maybe one or two client licenses for free. Small businesses, which initially use your software for free, are then likely to continue using your software when they grow and buy additional client access licenses.

4. Methodology

The operation will be done on a trial application in this research and the application will have a license class with two license a trial license and a full license upon with the MAC address. On the client side, the program reads the registry to verify whether the user use full is or trial license if the license were full the application will run with the full features, if not the application will run trial version if the user wants to get the full version just click in the activate tap.
In the activation tap the license class will encrypt the trial license with device’s MAC address with transposition cipher and pass it to the user to register and purchase from the application website. After the user register the website gave the full license also encrypted and bring it back to the application and enter the full license the application will decrypt the license and check the MAC address from the license with the MAC address of the device and license key with the full license key. If they match; the application will write the license in the registry for the next time, otherwise, that will considered as illegal operation and the application will close.

On the server side the client register to the server and buy the full license with the trail license that generate by the application. The server will decrypt the license and extract the MAC address and encrypt it again with full license this time and give it back to the client.

5. Implementation

This section presents the design of the software piracy protection, whose purpose is to meet the requirements defined in the methodology part.

The main idea of this program is to protect the software from being cracked by crackers.
The program has a calculator application interface that will provide a free capabilities and paid features as shown in Figure 3.

**Figure 4. Registration process**
Registration process should primarily initiated with the intention of achieving the full capabilities.

**Figure 5. Code generation**
As illustrated in Figure 4, the program generates a code that combines trial license and the MAC address encrypted by Transposition cipher. This code is copied to the software website as shown in Figure 5.

To complete the registration process, the product is activated by purchasing the full license by pasting the free license that was generated by the program (as shown in Figure 6). After the registration completion the used gets the message illustrated in Figure 7.

**Figure 7. Registration completion**
To verify the registration success the user can check the registration tab in main menu, the status should be disabled as shown Figure 8.

**Figure 8. Registration success**
In Figure 9 and Figure 10 the user can make an extra verification step by checking the registry editor.
6. Results

After applying these tests on the project we come to following results:

- The protection prevent to copy and distribute the software and thus prevent to steal or violate copyright ownership because of using the MAC address along with license encrypted with it which makes it more challenging to break or to crack it.

- This system is useful in making revenues because it prevent loss of revenue in software industries because software industries invest a huge amount of money and resources to develop programs and systems that should be paid for.

- No one can copy the software or share as it requires an activation code that is different for each individual because as we mention of using the MAC address make no chance to anyone to use other license because each license depends on the MAC address.

- The software is extremely robust and secure, as well as scalable another variables might be added to this equation along with license and the MAC address such as the serial number of every software or a date.

- The system can be used by any online software dealer and also can easily modified to the systems that does not user internet.

- It can also be used by organizations and business owners to promote their product and at the same time protect their copyright.

7. Conclusion

All in all, an attempt has been made to find a way out for the piracy problem by using the MAC address and a generated code using a new software. To make this software perfect we recommend the following:

- Enhance the encryption algorithm to make it more complex to break.

- This technique can be used to protect not only software, nevertheless it can be used to protect any digital file like e-books, movies and so on by using MAC address

- Since this technique use the MAC address of the device then this technique can work on any programming language and in any type of operation systems.

References


