New Trends and Issues Proceedings on Humanities and Social Sciences

Volume 6, Issue 6 (2019) 081-091

www.prosoc.eu

Selected Paper of 6th Global Conference on Contemporary Issues in Education 29-31 August 2019, St. Petersburg, Russia

Development of application for software piracy protection from hackers attacks

Ramiz Salama*, Department of Computer Engineering, Near East University, Nicosia, Cyprus
Ayman Okal, Department of Computer Engineering, Near East University, Nicosia, Cyprus
Krell Chiparausha, Department of Computer Engineering, Near East University, Nicosia, Cyprus

Suggested Citation:
Salama, R., Okal, A. & Chiparausha, K. (2019). Development of application for software piracy protection from hackers attacks. *New Trends and Issues Proceedings on Humanities and Social Sciences.* [Online]. *6*(6), pp 081–091. Available from: www.prosoc.eu

Selection and peer review under responsibility of Prof. Dr. Huseyin Uzunboylu, Near East University, North Cyprus

Abstract

Recently, programming theft has been a serious issue for programming ventures and extremely huge costs were required to secure their applications. As per Business Software Alliance, the worldwide programming theft rate in 2013 was 43% and the business estimation of unlicensed programming establishments was $62.7 billion, which brought about a large loss in income and a number positions in programming organisations. This paper will exhibit that 'programming robbery insurance framework' is mostly used to secure the theft of the framework. Presently, a progressive number of clients download the product without having the consent of the product's proprietor since the product has the item key which can be located/used by an obscure individual to utilise that product. Our methodology will utilise 'Macintosh-based confirmation' and create an item key, which checks or compares the item key against a unique MAC address on each machine.

Keywords: Copyright protection, software piracy prevention, identification, authentication, intellectual property protection, diversity, tailored updates.

* **ADDRESS FOR CORRESPONDENCE: Ramiz Salama,** Department of Computer Engineering, Near East University, Nicosia, Cyprus. *E-mail address:* ramiz.salama@neu.edu.tr / Tel.: + 90 533 841 81 42

1. Introduction

The goal of this paper is to use a framework to solve programming theft issue. It is utilised for the online procedure. This framework only permits the approved individual, since it checks the MAC address. The MAC address is the special location of the framework, so the MAC address isn't utilised for utilisation. It is highly helpful for the proprietor of the product or item. Overall, the income misfortunes because of programming theft were $48 billion in that equivalent year, an increase of 20% from the first year. Programming theft, otherwise called copyright encroachment of programming, is the unapproved duplication or circulation of copyrighted PC programming. Albeit most PC clients today know that unapproved use and duplication of programming is illicit, many demonstrate general negligence for the significance of regarding programming as profitable Intellectual Property

Software copy protection is a never-ending topic among developers. While it is true that perfect software copy protection is almost a dream given today's operating system and hardware infrastructure, if you are careful and use the right tools and techniques, you can achieve a good degree of protection for any application (Bagriyanik & Karahoca, 2016; Kocakoyun & Bicen, 2017).

- This study is intended to maintain software copyright protection and assure that it is being accessed only by the authenticated users.
- Piracy has become so prevalent over the Internet which poses a major threat to e-commerce sites.
- With the help of malicious codes and programs, hackers or an intruder can gain access to the system and steal the information.
- Hence, there arises a need to protect the information and products from being plagiarised.
- This project is developed for the same purpose to protect the software's ownership of copyright and make transactions securely.

The objective of this paper is to use a system to overcome this problem; it is used to protect the software from being pirated (Iqbal, Khan & Minhas, 2018; Wright & Akgunduz, 2018). It is used for online processes. This system only allows the authorised person, because it checks the MAC address. The MAC address is the unique address of the system and is not used for any other user. It is more useful for the owner of the software or product. Worldwide revenue loss due to software piracy was $48 billion in that same year, an increase of 20% from the preceding year. Software piracy, also known as copyright infringement of software, is the unauthorised duplication or distribution of copyrighted computer software. Although most computer users today are aware that unauthorised use and duplication of software is illegal, many show a general disregard for the importance of treating software as valuable Intellectual Property.

2. Related research studies

Information Economics and Policy, 32 (2015); *Telemetric and Informatics* (2017); *Electronic Commerce Research and Applications, 28* (2018); Solving a Global Software Piracy Problem (August 2017); *The Journal of Systems and Software, 150* (2019); Software Piracy Prevention Through Diversity, SIIA Anti-Piracy – What is Piracy (2018); Effective Anti-Piracy Methods to Employ in Software Development, *Imperial Journal of Interdisciplinary Research (IJIR)* (2017); and *Telecommunication Policy, 39* (2015).

3. Software piracy and deterrent solutions

Most trade organisations face software piracy problems. As a result, there are many developed systems available in markets to deal with this problem, but, in fact, most of these systems do not offer an appropriate solution. To tackle software piracy, a variety of solutions have been proposed. These solutions can be classified as either deterrent or preventive. Deterrent solutions reply to the fear of

the consequences of getting caught. The solution is successful if an individual abstains from criminal behaviour due to the perceived threat or fear of sanctions. Preventive solutions make use of current technology to increase the cost of the actual act of piracy. A deterrent solution relies on an individual fear of getting caught and does not directly increase the cost of the actual act of pirating. It is a mechanism put in place to discourage the act of piracy by imposing sanctions if the act is carried out and detected. The United States is the first country in the world to deal with deterrent solutions through several intellectual property rights laws.

To tackle software piracy, a variety of solutions have been proposed. These solutions can be classified as either deterrent or preventive. Deterrent solutions reply to the fear of the consequences of getting caught. The solution is successful if an individual abstains from criminal behaviour due to the perceived threat or fear of sanctions. Preventive solutions make use of current technology to increase the cost of the actual act of piracy. These solutions can either be hardware-based or software-based and include technologies such as tamperproof CPUs and software encryption. Deterrent and preventive solutions will be further explored in the following sections. The issues associated with software piracy are not obvious to everyone, which could be partially due to the non-exclusionary nature of a computer application. To illustrate, suppose Alice has a copy of a popular video game on her computer. Alice can make a copy of the game and give it to Bob so he can play it on his computer. Now, both Alice and Bob own the copies of the game which makes the computer game non-exclusionary. On the other hand, Alice's and Bob's computers are exclusionary objects because only one of them can own each computer at a time. The exclusionary nature of the physical computer makes it clear to whom the property belongs to. This is not the case with intellectual property such as software. Even though the unethical nature of software piracy might not be obvious, the concerns are certainly not new. One of the major concerns with published literature, such as articles or books, is plagiarism. Within the software industry, plagiarism is also a concern, but identifying and proving that a section of an application is stolen is far more difficult than with a published piece of literature. The difficulty in detecting software theft can mainly be attributed to the format in which software is distributed. For example, in the case of source code theft, the stolen code could be compiled using a different compiler which will yield an executable that looks different from the original. In addition, the economic impact for the company whose application was stolen can be severe. Software companies often make a significant portion of their revenue prior to the release of a competitor product. If a portion of their application is stolen, the competitor could decrease the production time and enter the market sooner. The second ethical issue is the illegal redistribution of the software. It is generally the case that pirated copies of software are distributed at a significantly discounted price, while still including all of the original functionality. Again, there can be an economic impact associated with this act. The ramifications associated with piracy propagate throughout the software industry. The obvious victims are the software companies themselves. However, the more peripheral victims are also not often recognised. Many pirates are undeterred by reports of financial losses suffered by software producers due to piracy. This could be because they do not see the trickle effect of the monetary losses. Many think of software producers as large companies which generate significant revenue, forgetting that the individuals who work for those companies feel the effects of piracy through decreased job opportunities or even lost jobs.

A deterrent solution relies on an individual's fear of getting caught and does not directly increase the cost of the actual act of pirating. It is a mechanism put in place to discourage the act of piracy by imposing sanctions if the act is carried out and detected. The United States is the first country in the world to deal with deterrent solutions through several intellectual property rights laws. The question of how these laws can be used to protect software has been debated for many years. The difficulty in devising the proper protection is rooted in categorising software which can be a product, a service or even a combination of both. Currently, four intellectual property rights laws can be applied in the protection of software. These laws include copyright, patent, trademark and trade secret.

3.1. Prevent software piracy

Software piracy is a major issue affecting companies and developers. Consequently, companies need to implement anti-piracy protection systems on their software-based products.

3.1.1. Legal protection

Most companies make sure their software is protected legally by a user agreement. Letting consumers know that making unauthorised copies is against the law will help prevent people from unknowingly breaking piracy laws.

3.1.2. Product key

The most popular anti-piracy system is a product key, a unique combination of letters and numbers used to differentiate copies of the software. A product key ensures that only one user can use the software per purchase.

3.1.3. Tamperproofing

Some software programs have built-in protocols that cause the program to shut down and stop working if the source code is tampered with or modified. Tamperproofing prevents people from pirating the software through the manipulation of the program's code.

3.1.4. Watermarking

Watermarks, company logos or names are often placed on software interfaces to indicate that products are legitimately obtained and are not illegal copies.

3.2. About software protection methods

There are many ways to try to protect your software from piracy and none of them is 100% effective.

Most protection schemes are usually composed of the following parts:

Check if the user has a license to run the software

This point can be accomplished by various means, like dongles, software licenses tied to one particular machine, LAN/Internet activation and original media checking.

Make difficult for a malicious user to break the protection scheme.

This point can be achieved by various means, from simple code obfuscation, which makes debugging/disassembly difficult, up to code encryption, which makes disassembly virtually impossible if the user doesn’t have a program license.

The first step is the easiest to achieve, and its security degree depends almost only on the authentication media chosen.

In my opinion, a good dongle or, even better, an internet authentication method is the most secure; in particular, internet auth, which has the obvious caveat of the need of a connection available to run the application, can be the most flexible one. The second step presents most of the challenges. If it's quite easy to make it impossible for a user to unlock a protected software without having a correct license, it's virtually impossible to have 100% protection against a malicious user with a software license; every software that can be run on a computer can also be copied. The solution proposed here doesn't want to be a commercial-grade protection scheme; its purpose is just to make it impossible for a user without a license to run the software, even if he has good skills in software debugging, and make it somehow difficult for a user in possession of a license to break the protection scheme.

This is accomplished by encryption of some software parts and on-the-fly decryption with a license key obtained in any usable way.

4. Proposed system and overview

Official licenses are a very important way of making money in the making of applications. Only you or your association should have the alternative to make license keys and you should in all probability execute using each grant key on one PC (or different PCs that you pick). Various designers' hotel to hiding a dim key age and endorsement computation in the applications, or scrambling license keys with symmetrical encryption count and after that hiding the encryption/unravelling the keys into the applications for grant key unscrambling and endorsement. The systems are off base and delicate and if a toxic get-together is really enlivened by your prepared application, the unscrambling keys or disentangling/encoding count will be removed from your application in few days after the release.

- This endeavour is proposed to keep up programming copyright protection and ensures that it is being brought simply by the checked customers.
- Piracy has ended up being so unavoidable over the Internet that speaks to an essential hazard to electronic business regions.
- With the help of malevolent codes and ventures, programs or intruders can get to the system and take the information.
- Hence, there develops a need to shield the information and things from being duplicated.
- This endeavour is delivered for a comparable motivation to guarantee the item's duty and make trades securely.
- Programming copy protection is an interminable point among architects. While the realities show that perfect programming copy confirmation is practically a dream, given the present working structure and the hardware found, if you are mindful by using the right gadgets and strategies, you can always achieve a respectable dimension of security for your applications.
- This adventure is relied upon to keep up programming copyright security and ensures that it is received in a simple way by the checked customers.
- Piracy had and has ended up being so unavoidable over the Internet that speaks to a significant risk to electronic business areas.
- With the help of malevolent codes and tasks, a software engineer or a gatecrasher can get to the structure and take away the information. Subsequently, there develops a need to shield the information and things from being duplicated. This venture is created for a similar reason to ensure the product's responsibility and make exchanges safely.

5. Modules and their description

This system is having nine modules:

- Online registration
- Payment for buying software
- Download
- PC Id Reader
- Product Id Generation
- Key Generation
- Data matching and Authentication
- Authentication
- Not supported on another PC

Description:

1. ***Online enrolment:*** Users need to initially enlist themselves into the framework.
2. ***Payment for purchasing programming:*** They need to initially purchase the product for getting to it by means of a secure online instalment office accessible in the framework.
3. ***Download:*** After paying instalments to the client, the required program could be downloaded. Alongside the product, a sequential key will be given for later use.
4. ***PC Id Reader:*** The product peruses your pc Macintosh ID.
5. ***Product Id Generation:*** The framework produces an extraordinary client ID by applying a calculation on the procured Macintosh ID.
6. ***Key Generation:*** The client may now demand a sequential key. The user has to send the produced key to the client. The key is produced by applying encryption on the created unique client ID.
7. ***Data coordinating And Authentication:*** Admin applies the encryption to the client ID and sends the encoded key. Indeed, even programming creates a key by encryption and afterwards coordinates the key given by the client and produces the key.
8. ***Authentication:*** If the key matches, the product fulfils the requirements or is secured.
9. ***Not upheld on other PC:*** Since if a similar key is connected on programming in another pc since the client is created by Macintosh id on another pc is extraordinary so the key for that pc will be unique.

❖ **Problem with the current scenario**

Unintentional privateers are people who buy applications and are unconscious of authorising and enlistment issues. People who are either not presented to programming advancement rehearses or don't comprehend the moral commitments of utilising programming involve most of these inadvertent privateers. Traditionally, there were no such frameworks to recognise pilfered programming which are introduced utilising the phony framework created key. For phony key age, numerous products are utilised to make programming enlisted as a permit key.

To stop the theft of unlicensed programming, there is a need to execute programming which would have the capacities to recognise phony produced keys.

6. Project design

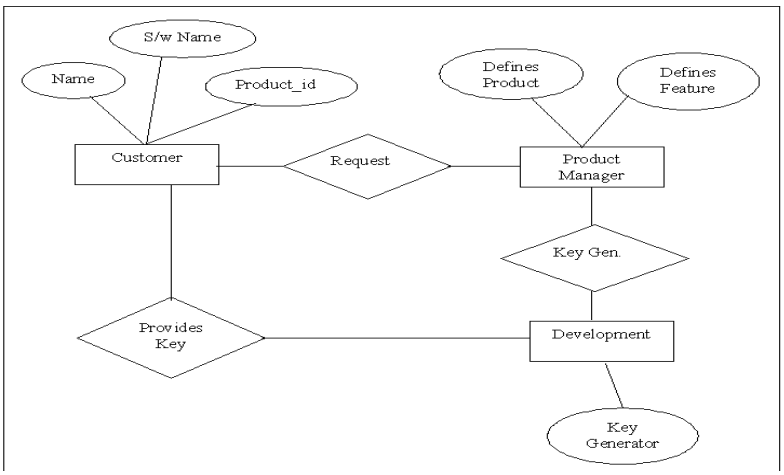

Figure 1. E-R Diagram

7. Data flow

A Data Flow has just a single heading of a stream between images. It might stream in the two bearings between a procedure and an information store to demonstrate a read before an update. Later, it generally demonstrated anyway by two separate bolts since these occur at various positions.

1. A join in DFD implies that the very same information originates from any of the two distinct procedures information store or sinks to a typical area.
2. An information stream can't go legitimately back to a similar procedure that it leads. There must be in any event where one different procedure that handles the information stream produces some other information stream to restore the first information into the starting procedure.
3. Information stream to an information store intends to refresh (erase or change).
4. An information flow from an information store intends to recover or utilise.

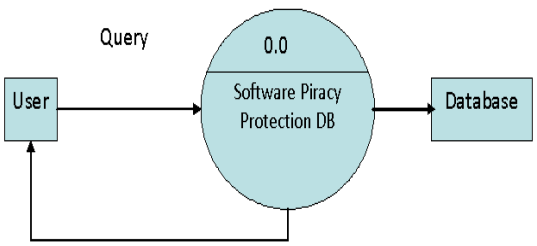

Figure 2. Database detail

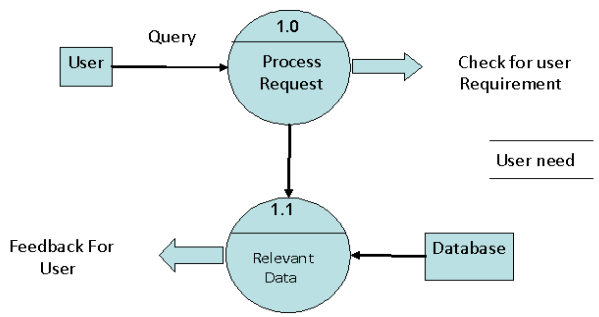

Figure 3. Level 1 DFD

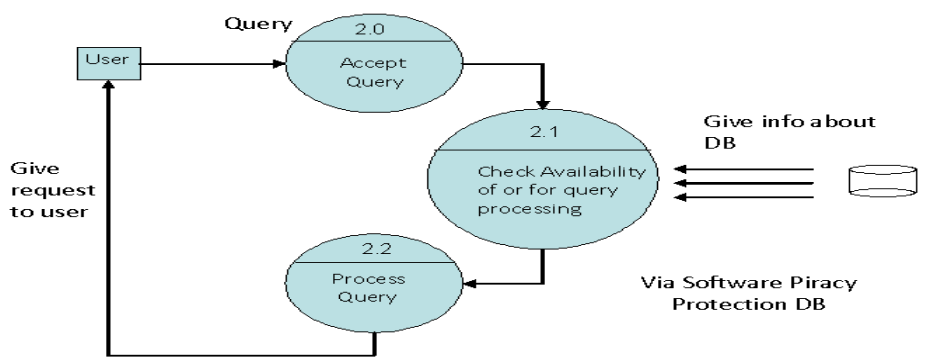

Figure 4. Level 2 DFD: prediction

The Steps of Using the Application of Software Piracy Protection:

Step 1: Home page of the system (Software Piracy Protection):

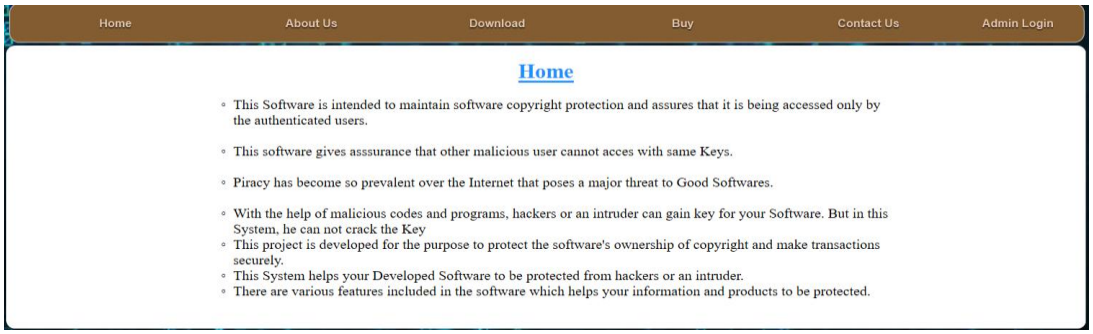

Figure 5. Step 1

Step 2: User Application: Buy Page: user fills the form to buy the product and to "submit" the information to the admin of the system

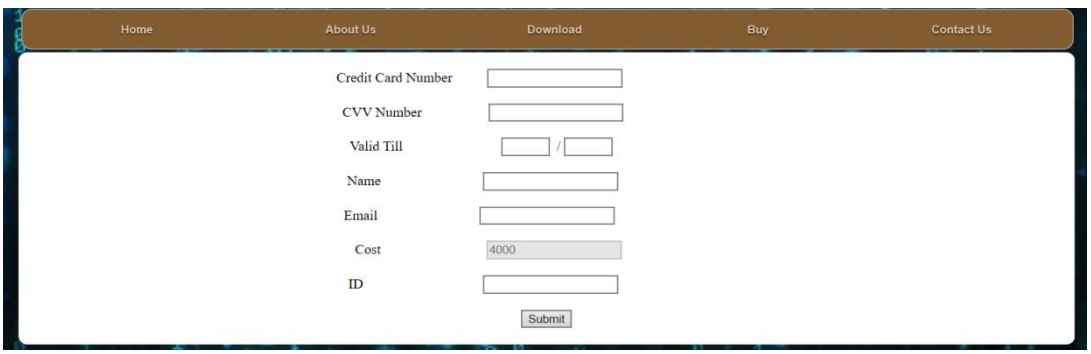

Figure 6. Step 2

Step 3: Admin Login: Admin enters the system to see the details of the customer's Name, Email and ID.

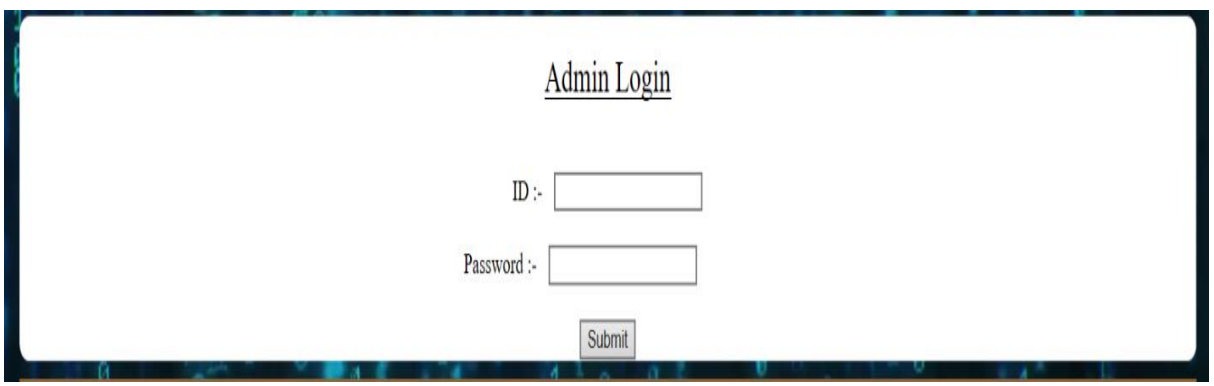

Figure 7. Step 3

Step 4: Customer list: The admin checks the information of the customer list.

Customer List

Name	Email	ID
test1	test1@gmail.com	2002
test1	test1@gmail.com	982860868625
ayman okal	aymanokal997@gmail.com	2002
test1	test1@gmail.com	2002
test1	test1@gmail.com	2002

Figure 8. Step 4

Step 5: Key generator: Admin inserts the ID number of the customer to get the code and send it to the customer's e-mail.

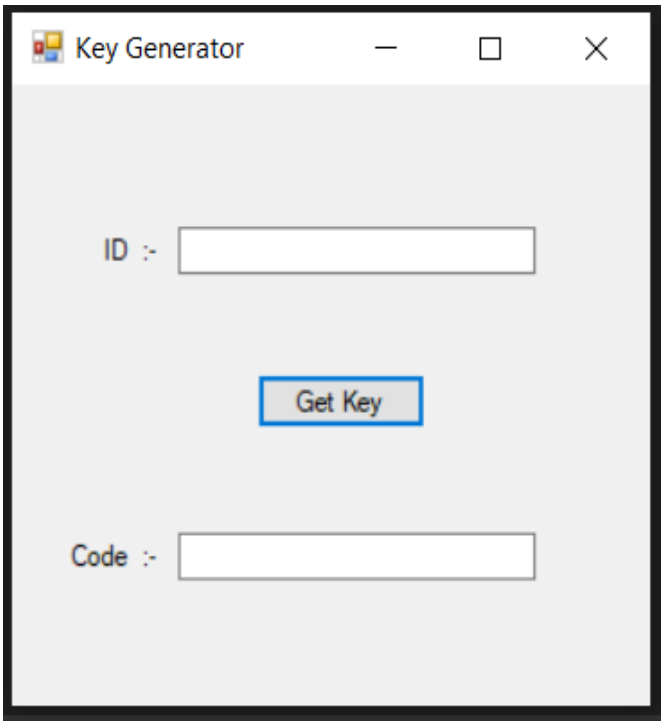

Figure 9. Step 5

Step 6: Receiving the code. In this step, the customers can receive the code of the product via e-mails sent from the admin.

8. Advantages of project

- On the off-chance that the item gets stolen, at this point, it would provoke loss of pay and can unfavourably influence the association consequently such that this structure is profitable in making wages.
- No one can copy any item or offer to someone, as it always requires an activation code that is different for every user and owner.
- Exceptionally solid and very secure structure.

9. Applications

- The framework can be utilised by any online programming seller.
- Likewise, it can be utilised by associations and entrepreneurs to advance their item and in the meantime ensure their copyright.

System is:

Burden Balancing: Since the framework will be accessible only if the administrator signs in, the measure of burden on the server will be restricted to the timeframe of the administrator.

Simple Accessibility: Records can be effectively acquired and stored along with other data individually.

Easy to use: The Website will give a very easy-to-use approach for all clients.

Proficient and solid: Maintaining a verified database on the server which will be available as indicated by the client's necessity with no maintenance cost will be effective when compared to putting away all the client information on the spreadsheet or in physically in the record books.

Simple support: Software Piracy Protection System is planned as a simple way. Along these lines, support is additionally simple.

10. Conclusion

Programming theft is a significant issue that affects the primary concern for programming designers. By executing a security plan for programming insurance, programming engineers can gain the advantages of assurance from theft by simply acquiring the capacity to actualise extra permit for the models. A Security usage plan that adjusts time and assets the ideal result is conceivable, given the wide scope of security alternatives. Engineers can also pick a staged way to deal with security usage if time or assets are compelled for time being.

Software piracy is a serious issue that impacts the bottom line for software developers. By implementing a security plan for software protection, software developers gain the benefits of protection from piracy as well as obtain the ability to implement additional license models. A security implementation plan that balances the time and resources with the desired outcome is possible, given the wide range of security options. Developers can additionally choose a phased approach for security implementation if time or resources are constrained in the short term.

References

Bagriyanik, S. & Karahoca, A. (2016). Big data in software engineering: A systematic literature review. *Global Journal of Information Technology: Emerging Technologies, 6*(1). https://doi.org/10.18844/gjit.v6i1.397

Iqbal, J., Khan, M. & Minhas, N. (2018). Are project managers informally following capability maturity model integration practices for project management? *Global Journal of Information Technology: Emerging Technologies, 8*(3), 86–94. https://doi.org/10.18844/gjit.v8i3.4048

Microsoft Developer Network (MSDN). Retrieved from http://msdn2.microsoft.com/en-us/default.aspx: This is a valuable online resource, and is a must for any developer using Microsoft tools; http://www.asp.net/: This is the official Microsoft ASP.NET web site. It has a lot of: tutorials, training videos, and sample projects;

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=403604&queryText%3DSoftware+piracy+protection+project;

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6830939&queryText%3DSoftware+piracy+protection+project;

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=403607&queryText%3DSoftware+piracy+protection+project;

Kocakoyun, S. & Bicen, H. (2017). Development and evaluation of educational android application. *Cypriot Journal of Educational Sciences, 12*(2), 58–68. https://doi.org/10.18844/cjes.v12i2.1938

Wright, B. & Akgunduz, D. (2018). The relationship between technological pedagogical content knowledge (TPACK) self-efficacy belief levels and the usage of Web 2.0 applications of pre-service science teachers. *World Journal on Educational Technology: Current Issues*, *10*(1), 52–69. https://doi.org/10.18844/wjet.v10i1.3351