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

Information Technology

Elixir Inform. Tech. 110 (2017) 48186-48190

ISSN: 2229-712X

# A Hybrid Model for Curbing Software Piracy

Ajayi, Olusola Olajide and Ezekiel, Olabode Johnson

Department of Computer Science, Faculty of Science, Adekunle Ajasin University, Akungba-Akoko, Ondo State, Nigeria.

**ARTICLE INFO**

**Article history:**

Received: 25 January 2017;

Received in revised form:

30 August 2017;

Accepted: 8 September 2017;

**Keywords**

Software,
Piracy,
License,
Copyright,
Infringement,
Unauthorised User,
Serial key,
Image Splitting.

**ABSTRACT**

The issue of copyright infringement spans across different sectors of the economy: ranging from the film industry, music, literary, as well as academics (plagiarism); just to mention but few. Software industry is another sector that is plagued with the menace of the software piracy. The practice started date back to late 1970s, and has eaten deep into the industry. The losses recorded by the software developers and the benefits users are been deprived of, prompted many researchers to suggest different measures (legal, physical and technical) to controlling this illegality. Employing means of battling this unscrupulous act has two-edged benefits. On the side of the developers, their intellectual properties are guarded, and the gains of their labors are not been reap by another. For the users, the originality of the product is assured and the users can benefits from regular update and maintenance routines. In this paper, our focus is on the technological control of this 'unwanted stranger'. Previous models were studied and adequately examined. The existing serial key and image splitting models were adopted to form a hybrid model type. The outcome of our design, testing and implementation, shows a reduction in the level of possible access by pirates or unauthorized users.



## Introduction

According to SIIA (2000), software piracy is defined as the unauthorized copying of computer software which constitutes copyright infringement for either commercial or personal use. It is the unauthorized reproduction of copyrighted software (Laurie and Kenneth, 2002). The BSA describes software piracy as the act of making unauthorised copies of software or downloading unauthorised copies of software via the internet. (BSA, 2001a). Software piracy is described as an act where there is an unauthorised use, duplication or sale of commercially available software (Moores and Dhillion, 2000). People installing software undertake risk of infecting their computers with harmful viruses and malwares, which can cause damage and loss of data (BSA, 2009). The copying, downloading, sharing, selling, or installing multiple copies onto personal or work computers are considered software piracy. Majority of software today is purchased as a single-user license, implying that it can be used by exactly a single authorised user in one or more machines provided the same licensee is the user (Khadka, 2015). Even in cases where a software comes in form of 2-users, 3-users etc. the licensee's status must still be maintained, otherwise, it is nothing but piracy. This increasing illegal act has been recognised in the industry for decades and has grown with the computer age (Wood, 1998).

## Statement of Problem

As could be observed from the reviewed articles, many approaches have been used by different authors and applied through different means to reduce software piracy, but the menace is yet to be dealt with. Examining the lapses or weak points of most techniques used, we opined therefore to adopt a hybrid approach that combines the strengths of most of the adopted techniques in one solution. This will hope will greatly alleviate the potency of piracy in the industry.

This research work is motivated by the gaps created in the reviewed literature as well as the established fact from the review that, piracy of software is still eating deep into the industry. We are bound therefore through this work, to contribute to the 'chasing away' of this 'unwanted guest' from the industry. The research aim to apply a hybrid technique that minimizes the penetration of software piracy.

## Research Methodology

In order to actualize the goal of this study, the study therefore adopts a hybrid technique of the existing and tested piracy control measures. We are poised to hybridize the various techniques into one singular solution to ensuring the effectiveness of combating the practice of pirating.

In this study, the outlined pattern was followed for the methodology:

### Pattern 1

Object Oriented Design Approach was used for the development of our demo-application.

### Pattern 2

Serial key technique was deployed at the loading point of the application to prevent any unauthorised access or intrusion.

### Pattern 3

Image splitting technique was adopted and applied at the deployment level of the application deployment. Image splitting technique; in which image is sliced into two parts, one part will be with the client and the other part will be with the trusted server. When user wants to activate the software; both parts of the image are made available to the server as well as the machine fingerprint (Hash codes of serial numbers of 'n' devices) of the users machine are sent to the server.

Tele:
E-mail address: olusola.ajayi@aaua.edu.ng

**Review of Related Literatures**

We present some previous research efforts directed at controlling software piracy activities, their strengths and noted limitations.

Shweta (2013): "Software Piracy Prevention Using Image Splitting". The study aimed at software piracy prevention techniques via image splitting technique". In this technique, image is sliced into two parts, one part will be with the client and the other part will be with the trusted server. When user wants to activate the software; both parts of the image are made available to the server as well as the machine fingerprint (Hash codes of serial numbers of 'n' devices) of the users machine are sent to the server. Using image splitting techniques. The author was able to achieve the result and the outcome of the implementation shows that image splitting technique is also suitable to cub software piracy to some extents. However, the technique has no extension retrieval capability.

Petar et al (2009): "Software piracy prevention through Digital Right Management". The goal of the work was preventing software piracy through copy protection techniques based on physical and intangible tokens. Copy protection techniques prevent unauthorized and illegal copying of software products. Common form of prevention is copy prevention techniques based on physical tokens. Physical tokens provide better protection from authorized copying than intangible one, the protected digital content become unsuitable for online distribution. The author used copy protection techniques. The result shows that copy protection techniques reduce online software piracy. Due to characteristic of online survey and the focus on user of sequencer software, the results cannot be easily generalized.

Gareth (2002) "A Taxonomy of Methods for Software Piracy". The research aimed at reducing software piracy using code obfuscation techniques. Code obfuscation is the deliberate altering of program code, whether at the source, object or machine code level. The idea is to hide the very purpose of the code, thereby making it more difficult to understand and alter (Collberg and Thomborson, 2002, where the author used code obfuscation technique with water-marking). The outcome of Gareth's empirical study shows that code obfuscation is also a means in which software piracy can be reduced**.** The major setback however is that code obfuscation allows a pirate to duplicate more easily by analysing the code for protections against duplication and circumvent it.

Lee and Kim (1999) "Copyright Protection of Software using Public Key Infrastructure". Here, the aim is to reduce software piracy where code to be executed is encrypted in some way and requires the correct key and subsequent decryption to run. Lee and Kim proposed a system based on the World Wide Web Consortium's Public Key Infrastructure (Public Key Infrastructure 2002). Users of software must possess their own unique public key certificate. The software provided by the distributor is encrypted with the user's own public key. Only the user possesses the private key and is therefore the only person who can execute the software. The outcome shows that using encryption techniques reduces software piracy. This system is of course vulnerable to key loss and assumes that all users of a given piece of software have a public key certificate issued by a trusted certification authority.

**Proposed Model**

Shweta (2013) prevent software piracy using image splitting techniques. This technique curbed or reduced software piracy to some extent but cannot be used for all software and also the technique has no extension retrieval capability. Petar et al (2009) make used of copy protection technique as a means of preventing online software piracy but due to characteristic of online survey and the focus on user of sequencer software, the results cannot be easily generalized. Gareth (2002) Reduced software piracy using code obfuscation technique .The idea of this technique was to hide the very purpose of the code, thereby making it more difficult to understand and alter by hackers but Code obfuscation allows a pirate to duplicate more easily by analysing the code for protections against duplication and circumvent it. Lee and Kim (1999) make used of Encryption techniques as a way of reducing software piracy but this technique is of course vulnerable to key loss. On this note a hybrid technique of image splitting and serial key model is adopted in the implementation of this work.

**System Models and Analysis**

**The Proposed System Model**

System modelling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. System modelling means representing the system using some kind of graphical notation, which is now almost always based on notations in the Unified Modelling Language (UML)

There are different types of modelling; of which a graphical modelling language such as Unified Modelling Language (UML) helps in developing, understanding, and communicating the different views.

The different types of modelling are;

- Use Case Modelling: - the functional requirements of the system are defined in terms of use cases and actors.
- Static Modelling: - This provides a structural view of the system.
- Class Modelling: - These are defined in terms of their attributes, as well as their relationships with other classes. Dynamic modelling provides a behavioural view of the system.

The use cases are realized to show the interaction among participating objects. Object interaction diagrams are developed to show how objects communicate with each other to realize the use case. The state-dependent aspects of the system are defined with state charts. This study is modelled using combination of different types of model that best suit the concept.

**The Models**

**Conceptual Model**

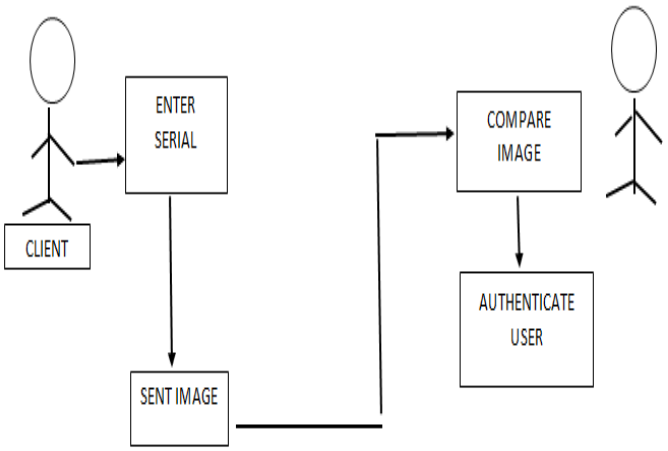



**Fig 1. Conceptual model of the system.**

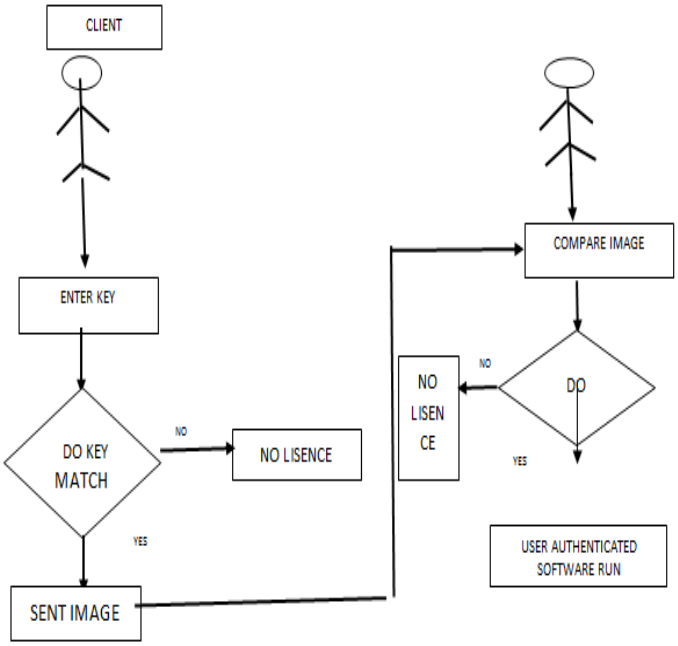



**Fig 2. Interaction Model.**

**Process Flow**

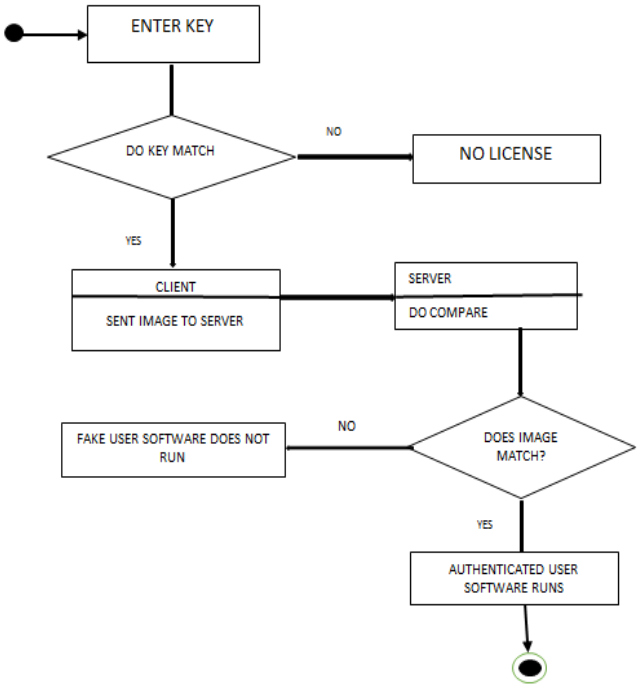



**Fig 3. The Process Flow.**

**The Architectural Model**

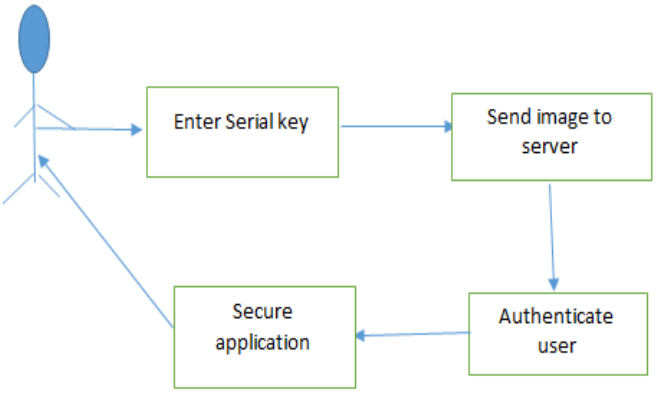



**Fig 4. The Architectural Model.**

**Authentication Framework**

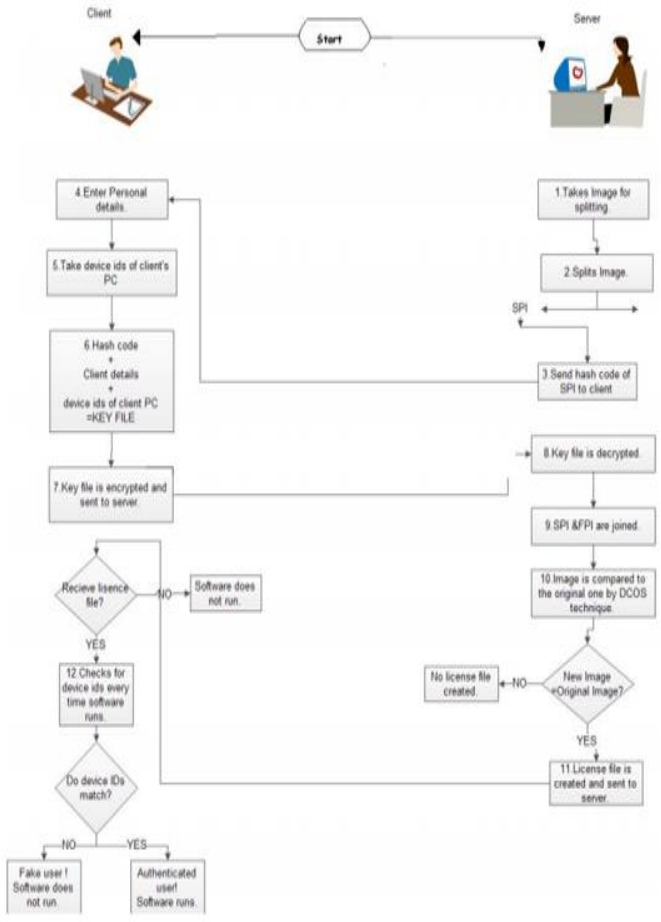



**Fig 5. The Authentication Framework.**

**The Design Interface**

System design pattern of this research work include the user of the software also known as client and the server which in this research work we refer to them as developer. Before a client can be authenticated or given access to make used the application such client must pass through the following design interface.

**Client Design Interfaces**

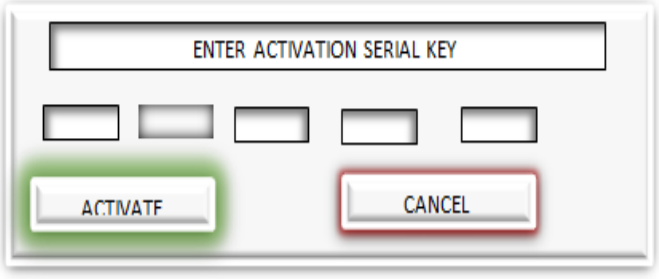



**Fig 6. Serial Key Activation.**

**Server Design Interfaces**

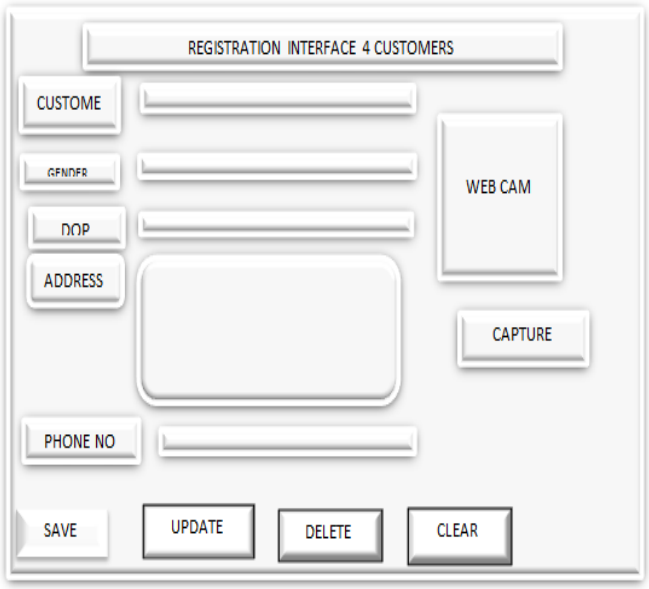



**Fig 7. Registration Interface.**


**Fig 8. Server Authentication Interface.**

**The Implementation**

**The Algorithm**

Here are the algorithms for implementing some of the logical modules.

**Admin Registration**

Enter customername, Gender, name, date of birth, address, phone.

If data entered valid

Save successful to database.

**User Installation| Client Installation**

During software loading

Prompt "Enter serial key"

If serial key is correct

Prompt "Send Image to server"

Else

Prompt "Wrong serial key"

**Server Authentication**

Retrieve image using user id from database

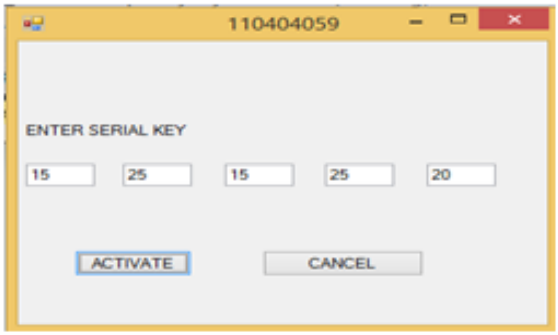
If image send by client==image in developer database

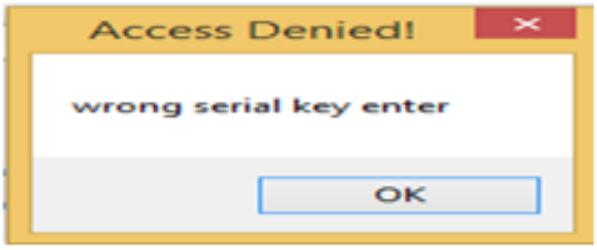
Prompt "user have access to secure application"

Else

Prompt "user cannot install software"

**The Output Interface**




**Conclusion and Recommendation**

This study uncovered the problem of software piracy and various methods some researchers have used to tackle the problem as well as their visible limitations. Adopting a hybrid approach to solving the problem of software piracy, the study presented an enhanced method of tackling the menace of software piracy. The implemented technique was to a large extent, able to prevent unauthorised copying or duplication of software.

In furtherance on the work, it is suggested that more works could be done on the server interface to allow for automated authentication process.

**References**

Business Software Alliance (2001a). http://www.bsa.org/usa/antipiracy/

Business Software Alliance (2009). Sixth Annual BSA and IDC Global Software Piracy Study. http://global.bsa.org/globalpiracy2008/studies/globalpiracy2008.pdf

Collberg, C. S. & Thomborson, C. (2002). Watermarking, Tamper-Proofing, and Obfuscation. IEEE Transactions on Software Engineering. Vol. 28 Issue 8, pg. 735-746

Gareth, C. (2002). A Taxonomy of Methods for Software Piracy. Department of Computer Science, University of Auckland, New Zealand gareth@cronin.co.nz

Khadka, I. (2015). Software Piracy: A Study of Causes, Effects and Preventive Measures. Undergraduate Thesis, Helsinki Metropolia University of Applied Sciences. pg. 6

Laurie, E. M., and Kenneth, T. F. (2002). Software Piracy: A Study of the Extent of Coverage in Introductory MIS Textbooks. Journal of Information Systems Education, Vol. 13(4).

Lee, B., Kim, K. (1999). Copyright Protection of Software using Public Key Infrastructure. Proc. of SCIS99.

Moores, T. T., and Dhillion, G. (2000). Software Piracy: A Review from Hong Kong. Communications of the ACM. 43(12), pg. 30.

Shweta, K. (2013). Software Piracy Prevention Using Image Splitting. International Journal Information and Computation Technology, Vol. 3, No 8, pg. 833-840. http://www.irphouse.com /ijict.htm

Wood, W., Behling, R., and Ang, A. Y. (1988). Software Piracy: Issues and Perceptions of Australian University Students. Journal of International Information Management, 7(2), pgs. 17-27.