International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 4 Issue: 11 84 - 90

Software Piracy Root Detection Framework Using SVM Based On Watermarking

Harshali Jagtap
Student of BE Computer Engineering
BVCEO & RI, Nashik, India
University of Pune.
harshalijagtap55@gmail.com

Samruddhi Rawool
Student of BE Computer Engineering
BVCEO & RI, Nashik, India
University of Pune.
rawoolsamruddhi93@gmail.com

Priyanka Waghmode
Student of BE Computer Engineering
BVCEO & RI, Nashik, India
University of Pune.
priyankapw94@gmail.com

Kalyani Malve
Student of BE Computer Engineering
BVCEO&RI, Nashik, India
University of Pune
malvekalyani92@gmail.com

Kavita Kumavat
ME Computer Engineering
BVCEO & RI, Nashik, India.
University of Pune.
kavitakumavat26@gmail.com

Abstract:- Software root piracy detection is tool to use for detect the owner of java software project or unauthorized copy of jar file. Existing system content the licensing mechanism for protecting our software from piracy but by skipping license or cracking that key piracy is done. The proposed system java based piracy detection software tool to overcome from this problem of piracy and find the offender. Proposed system use 'Watermarking' is a technique which attempts to protect the software by adding copyright notices or unique identifiers into software to prove ownership. We evaluate the existing Java watermarking systems and algorithms by using them to watermark byte code files. We develop the piracy root detection mechanism in this system. The advantage of this technique is that software watermarking is handled as the knowledge embedded into support vector machine and is closely associated with the program logic. It makes watermark more impossible to be destroyed and removed. We have to apply the watermarking content to the jar files of java software in this system the invisible watermarking is use. The results of the experiment further indicate that the proposed technique is a lightweight and effective software watermarking scheme.

General Terms
Embed module and retrieval module algorithms: Add method algorithm, constant string algorithm, block rendering algorithm, graph coloring algorithm.

Keywords
Watermarking, Software piracy.

*****

1. INTRODUCTION

With the improvement and expanding utilization of the Internet, replicating a computerized record is so natural and monetarily reasonable that advanced theft is uncontrolled over the globe. As indicated by a report of the Business Software Alliance (BSA), world Software theft brought about lost income of billions of dollars. The theft rate is high. Therefore, Software insurance has turned into an essential issue in momentum PC industry and a hot exploration point.

1.1 Watermarking

Software watermarking is a technique for Software assurance by installing mystery data into the content of programming. We embed such mystery data to claim responsibility for programming. This empowers the copyright holders to set up the responsibility for Software by removing this mystery message from an unapproved duplicate of this product when an unapproved utilization of this product happens. Advanced watermarking has gained significant ground and turn into a prevalent procedure for copyright insurance of sight and sound data. With the improvement and expanding utilization of the Internet, replicating a computerized record is so natural and monetarily reasonable that advanced theft is uncontrolled over the globe. Research on Software watermarking began in the 1990s. Among the methods that can shield Software from Piracy, Software watermarking is special in that it doesn't plan to keep Software theft from happening, however rather intend to show proof of a Piracy occasion. Sight and sound watermarking fills a comparative need in overcoming media Piracy, for example, securing the copyright of motion pictures in DVD design. It is as of now a well known exploration point in software engineering. Software watermarking is still a generally new region and we trust it merits more consideration. In spite of the fact that the objectives of interactive media watermark and Software

watermarking are comparable in that they embed some additional data into digitally-encoding objects. In Software watermarking, when a watermark is embedded into a project, the working semantics of the system must be saved. In most media watermarking, there is no fundamental working semantics layer in which to embed the watermark. Watermarking is the way toward concealing extra data inside Software codes, computerized information, (for example, picture, sound, and video) and records in a manner that it is about imperceptible. Computerized watermarking procedure suggests to the way toward installing the given watermark data, (for example, proprietorship data, name, logo, signature, and so forth.) in the defensive data, (for example, picture, sound, video, or content) and taking the given watermark data from the defensive data, which is not saw by human perceptual framework. At the end of the day, watermarking is a procedure of implanting a computerized watermark or sign containing data special to the copyright proprietor in the item (message, picture, sound, or video) which is should have been secured. An advanced watermark is characterized as an obvious or imperceptible recognizable proof code that is for all time installed in the information, to transmit shrouded information. It stays present in the information even after the decoding procedure. It typically gives copyright insurance of protected innovation.

Static Watermarking: Static watermark is put away in the Software projects executable itself. It can be put away in the information area of the article as static information like strings or it can likewise be inserted into code for application. For instance, consider 'n-way case' proclamation, by different stages of these n explanations a watermark of length lg(n!) can be put away. Static watermark are profoundly helpless to distortive assault. A straightforward distortive assault may re permute all case proclamations or/and break all strings into sub-strings.

Dynamic Watermarking: Dynamic watermark is put away in the condition of the product. On being food with some suitable info, the product may create particular yield which recognizes the watermark, or a few information structure may contain the watermark, hence recovery of this information structure gives the watermark. Additionally watermark can be installed into the execution hint of the product. Dynamic watermark can be effortlessly evacuated once the impossible to miss include succession is known. Additionally there are a few jumbling changes which change the dynamic state and make watermark acknowledgment incomprehensible.

1.2 Software Piracy:

Software piracy is the principle danger to Software industry because of significantly expanded use of the Internet and correspondence innovation. It has turned out to be increasingly difficult issue for Software industry. The effortlessness of downloading the product from the Net appears to urge individuals to utilize Software without approval. The product theft is characterized as the unlawful use, dissemination, duplication and offering of programming, ensured by copyright laws or secured under applicable permit assertion. Each nation is losing a huge number of dollars consistently because of the Piracy.

2. LITERATURE SURVEY

Software piracy is the significant sympathy toward Software gives regardless of the numerous safeguard systems that have been proposed to counteract it, It empowers a fine pick up control over dispersed software. It's methodology depends on diversity, each introduced duplicate is remarkable and upgrades are custom-made to work for one introduced duplicate just making of non indistinguishable duplicate's indistinguishable duplicate's and tailor overhauls is impressively all the more complicated [1].Watermarking inserts a mystery message into a spread message. In media watermarking the mystery is typically copyright notice& the spread an advanced picture or a sound or video production [2]. Software piracy is the demonstration of making unapproved duplicates of PC Software. In this we will consider Software theft where it is performed for profit. Consideration won't be given to PC clients who offer Software for no budgetary reward. In that overview some present technique for Piracy anticipation and endeavor to arrange them into a scientific classification of theft avoidance. There is some vagueness presented when encryption is considered as an unmistakable class [3]. In Markov models, we investigate a technique for recognizing Software piracy. A changeable produces is utilized to makes morphed duplicates of a base bit of software. A shrouded Markov model is prepared on the opcode succession removed from these transformed copies [4]. In Robust methodology the procedure which is used, that is known as software piracy aversion through sms gateway. The targeted methodology depends on sms entryway administration to introduce Software on a system, but the strategy left a few issues untouched i.e. issue identified with Macintosh address, time counterbalance and man in center attack [5].The first dynamic birthmark was proposed by Myles et al. To detect the program, they explored the complete control flow trace of a program execution. They proved that their method can oppose to any kind of attacks by code obfuscation. There is a drawback that their work is sensitive to different loop transformations. Besides, the program path traces are large and hence it is not feasible to scale this technique further [6].

Software theft, also known as software piracy, is the act of copying a legitimate application and distributing that software illegally, either free or for profit. The global revenue loss due to software piracy was estimated to be

more than $50 billion in 2013 [7]. Watermarks can be classified as either static or dynamic. Static watermarks are included in the code and/or data of a system program, whereas dynamic watermarking techniques store a watermark in a program's execution state [8],[9].

3. SYSTEM OVERVIEW

The proposed system has two modules:-

1) Embed module

2) Retrieve module

3.1 Embed module

As shown in following figure, Embed module used to embed the watermarking content in the byte code of the jar file. For that: let give the input as any jar file & watermarking content to the watermarking algorithm. The watermarking contents are less than 50 to 70 words. Watermarking algorithm will apply on input and then run the algorithm as an output we prepare the watermark jar file.Following figure shows embed module.

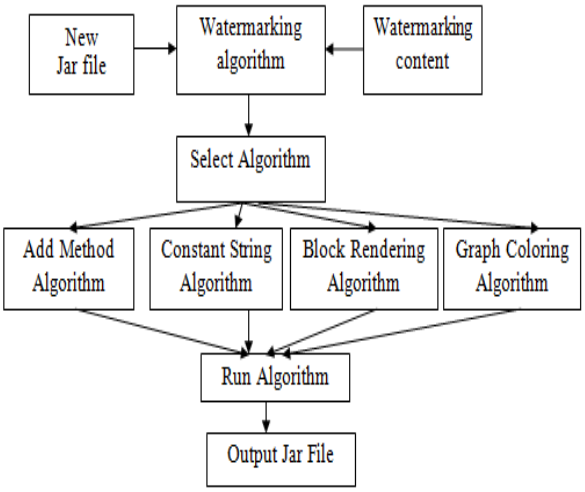

Fig 1: Embed module

Description: -

As shown in figure 1 a JAR (JavaArchive) file contains the class, image, and sound files for a Java application or applet assemble into a single file and possibly compressed. When programmer gets a Java program development kit, a small program called "jar" is included. The jar utility let the programmer to create, list, or extract the individual files from JAR file. In many enterprise, a Java application can started with a set of JAR files for use during execution. An off-the-shelf open source package can be deploy as a JAR file and run with XML data.

Watermarking algorithm: - These are the algorithms that are used to add the watermarking content into jar files. Watermarking contents: - These are the string, numbers, variables that are add as a watermark data into the jar file in hidden form. These contents are not fixed it is depend on the developer that which data is used as a watermarked contents. Select algorithm: - Here we have used 4 different algorithms so it depend on user that which algorithm he is selecting. Add method algorithm: - In this algorithm we use a string as a watermarking content and a key for retrieving contents. Block rendering: - In this algorithm we use a number as a watermarking content also use a line count to retrieve module. Constant string: - In this algorithm we use a constant string as a watermarking content and a key for retrieving contents. Graph coloring: - The user enters a watermark, a string or an integer. A string is converted into integer. Using this number a graph is generated and puts secret input sequence for retrieve Run algorithm: - Here we have to run or executes the selected algorithm from above algorithms on the jar file. Output jar file:- Now it is a composed and watermarked jar file.

3.2 Retrieve module

As shown in above figure (2), the second retrieve module. This module is used when we have to check that the copies of java software are Authorize or to retrieve the contented of Watermarking to find the root of piracy. Here we give input as java/jar file which is watermarked java file as shown in fig (2) then apply the Retrieval algorithms on that jar file & run the algorithm and get the watermark content from that file.Following figure shows Retrieve module.

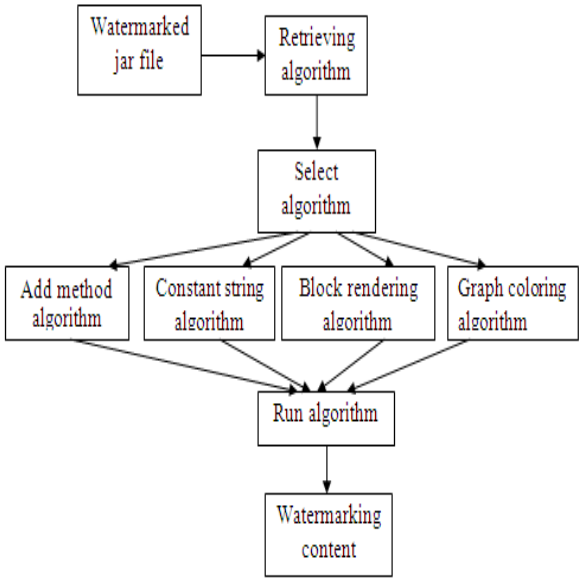

Fig 2: Retrieve module

Description:-

Watermarked jar file:- Watermarked jar file is a file which is already watermarked by using the embed module. Retrieving algorithm: - These are the algorithms that used to retrieve watermarked contents. There are 4 methods for

retrieving algorithms we have to select the one of them.. Select algorithm: - Here developer can try some input methods to identify that which algorithm is used in the jar file. These algorithms are explained in the detail as follows. Add method algorithm: - In this algorithm developer use to a key as an input to check that add method algorithm is used. Block rendering method: - In this, use line count to retrieve module and check whether the block rendering method is use. Constant string method: - In this algorithm developer use to a key as an input to check that add method algorithm is used. Graph coloring method: - In this method, during retrieval module the application is run with the secrete input sequence. To check that whether the graph coloring is use for jar file.
Run algorithm: -In this, we have to run or executes the selected algorithm on the current to get the output jar file. Watermarking content: - Here the watermarked contents are retrieving from the jar file.

3.3 System architecture

As shown in figure 3, the input file as jar file is give to the proposed system and the system will un-jar that input file & decompose that into the modules. Then embedding the watermark contents and parsing include watermark techniques as static watermarking or dynamic watermarking. Include the watermarking content using embed algorithm and compile that file. Then build the jar file which is newly created and generate output. The real file exactly similar to the newly formed jar file and they are executed similarly.

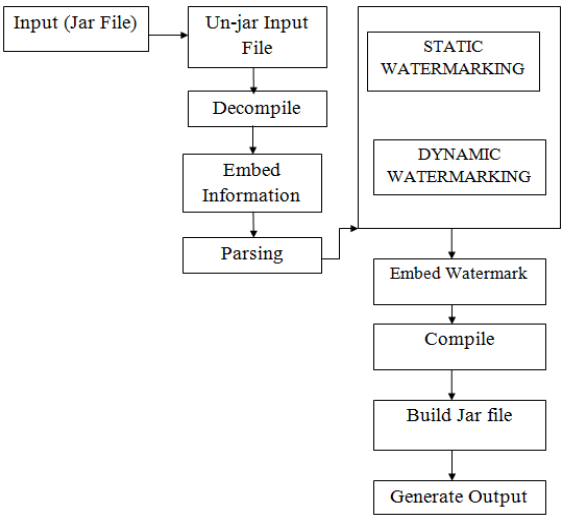

Fig 3: System Architecture

System description:-

Input jar file:-A JAR (Java Archive) file contains the class, image, and sound files for a Java application or applet assemble into a single file and possibly compressed. When programmer gets a Java program development kit, a small program called "jar" is included. The jar utility let the programmer to create, list, or extract the individual files from JAR file. In many enterprise, a Java application can started with a set of JAR files for use during execution. An off-the-shelf open source package can be deploy as a JAR file and run with XML data.
Unjar input file:-Unjar or unpack the jar fileusing jar tool.
Decompile: - To decompile is to convert object code program into some form of higher-level programming language so it can be read by humans.
Embed information:-
Parsing: - The task of the parser is basically to determine if and how the input can be resulting from the start symbol of the grammar.
Static watermarking is watermark that stored directly in data or codesections of a resident executable or class file.
Dynamic watermarking iswatermark that stored in run-time program structures.
Embed watermark:- In this we have to embed watermarking contents into jar file.
Compile built jar file:- now this jar file is then compiled to check the proper syntactical facts.
Generate output:- Generate the desired output as a watermarked jar file.

3.4 Algorithms

We have to develop the module to demonstrate how embedding and retrieving algorithms could use in our software watermarking system. The process of embedding module:

1. Construct the watermarked content W.
2. Decompose the jar file using jar tool.
3. Select the suitable algorithm from following methods for embedding the watermarking contains.
   - Add method.
   - Block rendering.
   - Constant string.
   - Graph coloring.
4. Embedded that contain into the given jar file.
5. Finish.

Other algorithms are as follows that each algorithm is operates in two different modes: 1. Embed mode 2. Retrieve mode. The algorithms are :-

1) Add method :

1. Input jar file.
2. Input watermark contents.
3. Decompile jar.
4. Divide watermark into two parts.
5. Reverse parts and join-rwatermrk
6. Find some class from jar.
7. Insert new method rwatermrk() using sendmarkapi.
8. Add name of watermarking method and place.
9. Assemble jar file.

2) Block rendering:

1. Input jar file.
2. Input watermark contents.
3. Decompile jar.
4. Convert watermark to number using sendmark string to number.
5. Count digit of number-C.
6. Find some class from jar which contains no of lines equal to count-C.
7. Embed each no digit in given line.
8. Add name of watermarking method and place.
9. Assemble jar file.

3) Constant string method:

1. Input jar file.
2. Input watermark contents.
3. Decompile jar.
4. Divide watermark into two parts.
5. Reverse parts and join-rwatermrk
6. Find some class from jar.
7. Insert new method rwatermrk() using sendmarkapi.
8. Add name of watermarking method and place.
9. Assemble jar file.

4) Graph coloring:

1. Input jar file.
2. Input watermark contents.
3. Decompile jar.
4. Convert watermark to number using sendmark string to number converter.
5. Count digit of number-C.
6. Find some graph component class from jar.
7. Embed each number digit in graph data.
8. Add name of watermarking method and place.
9. Assemble jar file.

The process of retrieve module:

1. Give jar file as a input.
2. Find method of watermarking.
3. Find location.
4. Retrieve using reverse sequence of appropriate method-add method, block rendering method constant string method or graph coloring method.
5. Display watermark.

3.5 ATTACKS ON SOFTWARE WATERMARKING

Watermarked software might be liable to assaults that have the goal of finding, mutilating, or evacuating the watermark. The nature of a watermarking framework is related with how much the watermarked software is impervious to assaults. As of not long ago, most ways to deal with programming watermarking have focused on opposing assaults by semantics-safeguarding program changes, including assemblage, advancement, obscurity, decompilation, and dead-code evacuation. Such changes don't change the conduct of a program, however they do change the type of the program. All things considered, they may effortlessly expel or bend watermarks that are implanted in the structure of the program, for instance, in remarks, in string constants, in the request of guidelines:

/* My software, version 1.0 */

- a information string: printf ("My software, version1.0");

To show these ideas we will expect the accompanying situation. An Alice watermark a host objects O with watermark W and key K, and after that offers O to Bob. Before Bob can offer O on to Douglas he should guarantee that the watermark has been rendered pointless, or else Alice will have the capacity to demonstrate that her licensed innovation rights have been damaged.

Subtractive assault: If Bob can identify the nearness and (inexact) area of W, he may attempt to yield it out of O. A viable subtractive assault is one where the trimmed question has sufficiently held unique substance to in any case be of esteem to Bob.

Distortive assault: If Bob can't find W and will acknowledge some corruption in nature of O, he can apply distortive changes consistently over the protest and, henceforth, to any watermark it might contain. A viable distortive assault is one where Alice can no longer recognize the debased watermark, however the corrupted protest still has esteem to Bob.

Added substance assault: Finally, Bob can expand O by embeddings his own particular watermark W0 (or a few such stamps). A compelling added substance assault is one in which Bob's stamp totally supersedes Alice's unique check with the goal that it can never again be extricated, or where it is difficult to identify that Alice's check transiently goes before Bob's.

Software Watermarking:

The key motivation behind a software watermark is to install data in an application. We give the accompanying formal expressions to applicable ideas: Let P be a PC program that is accessible for control and Ikey be some substantial info grouping to P. Let a number ω be the watermark we wish to implant in

P.The work E is known as a watermark inserting capacity and has the property that

E (P, Ikey, ω) = P!

Where the yield of E is a watermarked program, P

The capacity R is known as a watermark acknowledgment work and has the property that

∀P! : R (P!, Ikey) = ω

4. Advantages

- Software watermarking, which can be utilized to recognize the protected innovation proprietor of piece programming.
- Larger amount of security: Security classification of the installed data is given by a mystery key.
- Without this key the watermark can't be gotten to or modified. Watermarks can be outlined in a manner that the embedded data is still retrievable even after the bearer medium changed.
- The advantages of watermarking are that the result of inserting procedure is still a computerized medium. Customers can do everything with a stamped medium that they can do with an unmarked one.
- Watermark medium can be played with no limitation.
- Software theft persistent to be a major monetary sympathy toward organizations and associations.
- There is no additional expense for equipment and Software.
- Software watermarking requires one and only time cost for establishment.

It is easy to understand and productive. It makes watermark harder to be obliterated and expelled. There is no wastage of memory or space required for making watermarks, input records and so on. Difficult to split the watermark by unauthorized individual.

5. CONCLUSION

- Thus we have proposed the new system which is the software piracy root detection tool use to find the pirated copy of Java software. In this we have applied the watermarking technique to the Java software or Jar files & by using the invisible watermark. The watermarking content is applied and detect by using this software, we use watermarking but the size and execution time of the java software is does not increase. Also it do not get complicated.
- In the existing system the prevention algorithm is used but we have used the detection algorithm in this system. These algorithms used to embed & retrieve the watermarks which are impossible foroffender to remove and it is unknown to the end user also. So it is detected that given copy of software is pirated or not.We have implemented and experimented with a watermarking system for Java based on the ideas of James Hamilton. Our experiments show that watermarking can be done efficiently with moderate increases in code size, execution times, and memory space usage, while making the watermark code resilient to a variety of program transformation attacks. For particular representation of watermarks, the retrieval time of watermark is on the order of one minute per megabyte of memory space. Our implementation is not designed to resists all possible attacks; to do that it should be combined together with other protection methods such as obfuscation and tamper proofing.

Software watermarking is process of embedding a large number in a program such that :1)the number can be reliably retrieved after the program has been subjected to semantic preserving transformation 2) the embedding is imperceptible to an adversary ,and 3) the embedding does not degrade the performance of the program.

This is challenging problem that, to the best of our knowledge has not previously been addressed in academic literature. The some published account of which we are aware (mostly software patents) all describe trivial schemes in which copyright notices are embedded in the object code of the program. None of these methods are resilient to even the simplest program transformation.

In this paper we have constructed a taxonomy of software watermarking technique based on how marks are embedded, retrieved and attacked. We have furthermore provided a formalization of software watermarking technique that we believe will form the basis for further research in the field

Future Scope

System supports only for transformation of jar files. But in future we can make it supportable for any other type of data files.

6. REFERENCES

[1] BetrandAnckaert,Bjorn De Sutter,Koen De Bosschere. Software piracy prevention through diversity,Electronics

[2] & information systems deparments, Ghent University,Sint-Pietersnieuwstraat 419000 Ghent,Belgium.

[3] Collberg&Thomborson, watermarking Tamper,Proofing& Obfuscation tool for software protection.

[4] Gareth Cronin, Dept. Of computer science , university of Auckland, Newzealand "A Toxonomy of methods for software piracy prevention."

[5] [ShabanaKazi "Hidden morkov models for software piracy detection."

[6] Ajay Nehera, RajkiranMeena, Deepak Sohu, Omprakash Rishi "Robust Approach to prevent software piracy", March 2012.

[7] G.Myles and C. Collberg. Detecting software theft via whole program path birthmarks. In Inf. Security 7th Int. Conf. (ISC2004), pages 404–414, Palo Alto, CA, September 2004.

[8] CollbergThomborson, principles of Softwarelanguages in 1999, POPL'99.

[9] A practical method for watermarking Java programs,Monded,Lida,Matusmoto,Lnoue,Toril,Compsac 2000

[10] Softaware watermarking through register allocation ;implementation,analysis& attacks, Myles, Collberg, ICISC 2003.

She is student of computer engineering at Brahma Valley College of engineering and Research center Nashik. Underthe University of Pune. Her interest in software security.

She is student of computer engineering at Brahma Valley College of engineering and Research center Nashik. Under the University of Pune. Her interest in security.

She is student of computer engineering at Brahma Valley College of engineering and Research center Nashik. Under University of Pune. Her interest in security and networking.

She is student of computer engineering at Brahma Valley College of engineering and Research center Nashik. Under the University of Pune. Her interest is in thesoftware development and security.

K. S. Kumavat, Ph.D Appearing, ME, BE Computer Engineering. Was educated at the Pune University. Presently she is working as Head of Information Technology Department of Brahma Valley College of Engineering and Research Institute, Nasik, Maharashtra, India. She has presented papers at National and International conferences and also published papers in National and International Journals on various aspects of Computer Engineering and Networks. Her areas of interest include Computer Networks Security and Advance Database