사전 소프트웨어 워터마크 기술 기반

지샨 빠르웨즈,이승룡, 구교호
경희대학교
e-mail : {zeeshan, sylee}@oslab.khu.ac.kr
yklee@khu.ac.kr

Dictionary Based Software Watermarking Technique

Zeeshan Pervez, Sungyoung Lee, Young-Koo Lee
Dept. of Computer Science, College of Electronics and Information, Kyung Hee University

Abstract

As software is becoming increasing important to the human society, so does the effort to produce them is also increasing.. All of these efforts can be at risk when source code of the application is reverse engineered by the software pirate. Many attempts have been made to protect Intellectual Property Rights (IPR); one of the newest attempts to protect IPR is software watermarking. It is used to prove ownership right when IPR are violated, and also prevents the bandit from altering the code for his own use. In this paper we are presenting a new technique for software watermarking know as Dictionary Based Software Watermarking (DBSW). DBSW works by embedding dummy instructions in source code with the help of predefined mapping already available in the dictionary. These instructions are identical to the normal instructions of the program and are hard to identify or to extract from the watermarked program. With the help of DBSW we not only can stop source code alteration but can also identify the buyer how has distributed the pirated copy.

1. Introduction

Software industry is making progress by leaps and bounces. The way it is making progress not only introduces new opportunities to the new comers but it also introduces new threats to the software venders.

As man hours put to develop software is increasing so does the concerns on how to manage the IPR. Software piracy is causing software industry with the loss of 12 billion dollars per years [1]. As described by the AIFRS (Australian Equivalent International Financial Reporting Standard) any software which is not integral part of computer's operations is an intangible asset of a company, they have included application software in it [2]. Tangible assets are being protected by different measure like security locks, regular patrolling and many more. Unfortunately these protection measures don't apply to the software. Due to the intangible in nature they can be easily copied and can be concealed in any data storage disk.

2. Software Piracy Threats

As software industry is growing so does the threats to it. Software piracy is one of the similar threat that some malicious user does by tampering with software code so as to replace or wreck its original ownership or authorship mark [1][3]. Pirates usually gain the advantage by reselling the pirated copies by modifying the application logic. We can divide these threats into two categories

- Software Piracy
- Source Code Piracy

3. Dictionary Based Software Watermarking Technique (DBSW)

The proposed approach can work with any type of language and application. DBSW approach is divided into following major stops.

- String Generator
- String Validator
- Instruction Generator
- Instruction Embedder

Fig. 1 explains the working to DBSW approach.

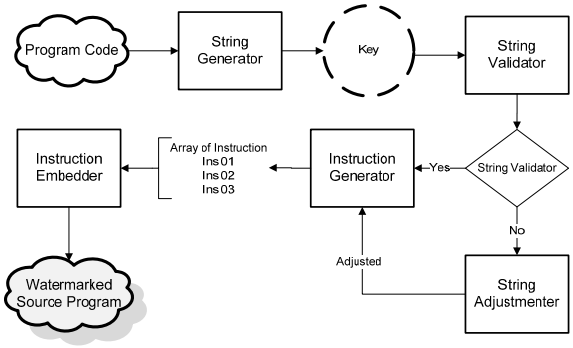

Fig. 1 - Abstract View of DBDW

3.1 String Generator

The first step in DBSW is unique string generation (Key). String generation process works by taking the program or part of a program like individual classes, and manipulate it to

generate a string. The core point is that we want to represent the source code with a much smaller string as compared to the original source code. Any technique can be used to get the representation string until unless it is ideally impossible for the attacker to mimic.

3.2 String Validator

The output of first step (String Generator) is a Key. Next step is string validator it checks whether the Key conforms to the dictionary we have. Basically dictionary represents the dummy instruction which will be inserted in the source code. Every entry in the dictionary maps a letter to the dummy instructions.

Dictionary can contain as many entries as possible but few points should be considered first before making any instruction part of the dictionary.

- It should be similar as to the regular instructions.
- It should be not give any idea about the watermarking.
- It should not affect the overall functionality of the program.

3.3 String Adjuster

As DBSW uses the dictionary to map the Key generated from the source code to the dummy instruction present in the dictionary. There is need to validate the Key that all of the letters within the Key has any entry in the dictionary. For that reason we validate the Key using string validator function described in the previous step. If Key is not a subset of the Power Set of dictionary then we have to change the Key, using String Adjuster function.

3.4 Instruction Generator

Instruction Generator basically maps the Key output by the string validated function to the dictionary. For every letter present in the Key, instruction generator checks the dummy instruction available in the dictionary and insert that dummy instruction to the array. Once for every letter dummy instruction is inserted into the array, then that array is forwarded to the instruction embedder.

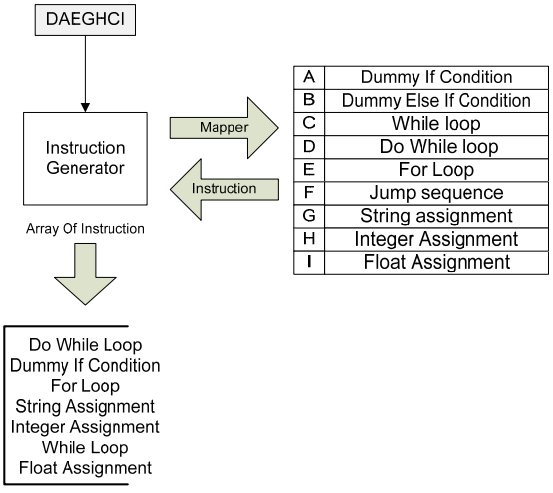

Fig. 2 - Instruction Generator

3.5 Instruction Embedder

Once we have the array of instruction which represent the Key verified by the previous steps; then we can embed these instructions into the source code. This array of instruction represents the watermark.

Before embedding the watermark we have to keep in the view that these dummy instructions should not destroy the actual functionality of the program. There could be many ways in which program can be watermarked

- Watermark the whole class
- Watermark the every methods
- Watermark selected methods

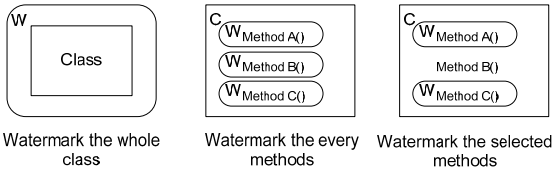

Fig. 3 – Watermark Insertion Type

Fig. 4 explains the way in which source code can be watermarked. After successfully inserting the software watermark the resultant program will the watermarked which can perform the same functionality and ideally in equal interval of time.

4. Conclusion

DBSW is an approach which can be easily adopted by any programming paradigm. What is mandatory for this technique is the knowledge of low level representation of a language like Java has Bytecode. Moreover DBSW does offer easy embedding of watermark without effecting the data and control structure of original program. Since it uses dummy instructions which are almost identical to the original program it is difficult to extract or to recognize the watermark in the program. DBSW can be made more complex by dynamic mapping as discussed in previous section, and also by extending the entries in the dictionary. More dummy entries in the dictionary tend to generate more concrete watermark.

4. Acknowledgement

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2009-(C1090-0902-0002)). Also, it was supported by the IT R&D program of MKE/KEIT, [10032105, Development of Realistic Multiverse Game Engine Technology].

References.

[1] Christian S. Collberg and Clark Thomborson “Watermarking, Tamper-Proofing and Objuscation- Tool for Software Protection”. In IEEE Transaction on Software Engineering Vol 28 in August 2002.

[2] ACT Accountancy Policy - Software http://www.treasury.act.gov.au/accounting/download/AP_05.pdf in February 2002.

[3] G. Naumovich and N. Memon. Preventing Piracy, Reverse Engineering, and Tampering. In IEEE Computer, Vol 36, No. 7, pp. 64--71, July 2003.