# **VERSION BASED SOFTWARE WATERMARK**

# D. Seetha Mahalaxmi<sup>1</sup>, S. Viswanadha Raju<sup>2</sup>, A. Vinay Babu<sup>3</sup>

<sup>1, 2, 3</sup>Associate Professor, Professor and HOD of CSE, Principal, Dept. of CSE, JNTUHCEH Jagityal, JNTUH JNTUHCEH lakshmi.dsml@gmail.com, SVRAJU.JNTU@gmail.com, avb122@gmail.com

#### Abstract

A key element of any software is to protect the software from various attacks and in case if any theft has occurred then the developer must be able to prove their ownership. Software Watermarking is one of the technique which helps in proving the authentication of the developer. A lot of research was done in Software watermarking but all this work is discussing on various embedding techniques of a software watermark into a program and the software watermark which was used for embedding is a serial number or a unique identification of the developer. In this paper a new methodology to obtain the software watermark, which is known as Version Based Software Watermark, is given. The remaining part of the paper discusses on the importance of the Version Based Software Watermark, properties of VBSW and evaluation results of the VBSW.

\*\*\*

Keywords: Software Watermark, VBSW – Version Based Software Watermark, LOC, version number

## **1. INTRODUCTION**

Software is a collection of data structures, algorithms, user interface, and implementation details. Release of such software can be used for various application, for example software written for the calculator can be used for various applications such as for mathematical calculations, scientific purpose and etc. In this scenario, there is a possibility that an attacker wants to get hold of the entire software or the logic of the software so that it can be used for their own purpose. This gives rise to dispute, and in this scenario the developer must be able to prove their ownership. Software Watermarking is a one of the techniques which gives the developer an opportunity to prove their authentication.

### 2. WATERMARK

Watermark is a technique which embeds a secret message into cover message. There are various types of watermarking technique such as digital watermark [2] which has emerged as an enabling technology for protecting intellectual property rights of digital information, Software Watermark and etc.

#### SOFTWARE WATERMARK

Software Watermark[1], "embed a structure W unto a program P such that W can be reliably located and extracted from P even after P has been subjected to code transformations such as translation, optimization and obfuscation".

#### **Embedding techniques of Software Watermarking**

A lot of work has been done on embedding software watermark into a program. Some of them are dynamic graph watermark [1], graph coloring problem [5], relative frequencies of instructions [4].

In dynamic graph watermarking [1] the watermark is embedded into the topology of a dynamically built graph structure. Qu and Potkonjak [5] proposed embedding of a into a register reference graph. In [5], stern et al. embeds the watermark in the relative frequencies of instructions throughout the entire program using a spread spectrum technique and some of the other techniques are Dynamic Path Based [3], embedding a watermark using graph problem[5][6]

#### **3. MOTIVATION**

The above techniques are discussing on the different types of embedding a given watermark into the program. All these techniques uses a watermark which is a unique number represented in a binary form, or a serial number. In the proposed work a new method of evaluating the Software Watermark known as Version Based Software Watermark is given. The main objective of the Version Based Software Watermark is to act as a copyright protection and to protect the software from the Version attack and LOC attack.

#### **PROBLEM STATEMENT**

Version Based Software Watermark is one of new approach of a Software Watermark. The Version Based Software Watermark is calculated using the LOC and Version number of the software. This is useful not only to prove the ownership of the software but also can be protected from version attack and lines of code attack. The goal of the work is that it must be difficult to retrieve the watermark by the attacker under different conditions.

#### Lines of Code

The Lines of Code is the number of lines that have been used to develop the software. While counting this number of lines the dummy lines i.e the authors name, date of development, is not counted for lines of code.

#### Version Number

Version number is the latest software release. Every release of the software has a new feature or a set of new features with new version number.

#### Version Watermark

Version Watermark, V, is a watermark which is computed using the values of LOC l, version number z, and assuming the probability of success that the attack is not possible is 0.8.

$$V = \frac{(l * 0.8)}{z}$$

#### 4. PROPERTIES OF VERSION BASED

#### SOFTWARE WATERMARK

#### **Toughness of a Version Based Software Watermark**

Version Watermark that is embedded must be tough to reasonable set of de-watermarking attacks [1] and it must be possible to recognize the watermark even after it is subjected to various attacks.

$$T: A(S) \rightarrow V$$

The Toughness of the Version Watermark is such that the Software S is subjected to various types of attacks A(S) but it is possible to retrieve the original Version Watermark, V.

#### Magnitude Of A Version Based Software Watermark

The Magnitude of Version Based Software Watermark is defined as the magnitude of the Version Watermark must be very small when compared to the magnitude of the software.

$$M(V) \ll M(S)$$

Where M(V) is the magnitude of the Version Watermark and M(S) is the magnitude of the Software.

#### **Covertness Of A Version Based Software Watermark**

The Covertness of a Version based Software Watermark is the insertion of watermark does not change the statistical properties of software.

$$C(S') \cong C(S)$$

Where C(S') is the Covertness of the Version Watermark and C(S) is the Covertness of the Software.

### **5. EVALUATION**

The evaluation of Version Based Software Watermark can be defined on the two parameters Correctness of extraction, which was obtained as:

#### **Accuracy of Extraction:**

The watermark extraction function must have the following characteristic:

$$\forall S^{\prime\prime\prime} : X(S^{\prime\prime\prime}) = V$$

Where  $S^{\prime\prime\prime}$  is the watermarked Software with transformations such as tamper-proofing and obfuscation, V is the Version Watermark, and CX is the function, Correctness of Extraction.

Programming Language	С	C++	Java
Accuracy of Extraction	90%	90%	90%

# CONCLUSIONS

The Version Based Software Watermark is one of the new approach of Software Watermarking when it makes it makes use of the LOC and version number of the software which are the heart of any software. In this paper a detail report on Version Based Software Watermark and its properties are discussed.

#### REFERENCES

[1] Christian Collberg, Clark Thomberson, "Software Watermarking: Models and Dynamic Embedding", In ACM SIGPLAN – IGACT Symposium on Principles of Programming languages (POPL98), San Antonio, Texas

[2] Jian Zhao and Chengui Luo , Digital Watermark Mobile Agents, Fraunhofer Center for Research in Computer Graphics , Inc, 321 South Main Street, Providence, RI 02903

[3] Collberg C., Carter E., Debray S., Huntwork A., Linn C., Stepp M., Dynamic Path Based Software Watermarking, Department of Computer Science University of Arizona Tucson, AZ 85721, USA [12] Cousot P., and Cousot R., "An Abstract Interpretation-Based Framework for Software Watermarking", in POPL, 2004.

[4] Stern J.P., Hachez G., Koeune F., and Quisquater J., "Robust Object Watermarking: Application to Code", in Information Hiding, pages 368,378.

[5] Qu G., and Potkonjak M., "Analysis of Watermarking Techniques for Graph Coloring Problem", in IEEE/ACM International Conference on Computer Aided Design, pages 190{193, November 1998.

[6] Venkatesan R., Vazirani V., and Sinha S., "A graph Theoretic Approach to Software Watermarking", in 4th International Information Hiding Workshop, Pittsburgh, PA, April 2001.