

Design of VBSW



Engineering

KEYWORDS : VBSW, Toughness, Magnitude, Covertness

D.Seetha Mahalaxmi	Associate Professor, Dept. of CSE, JNTUHCEH
Dr. S. Viswanadha Raju	Professor and HOD of CSE, Jagityal, JNTUHCEH
Dr. A. Vinay Babu	Principal, JNTUHCEH

ABSTRACT

Software Watermark is a technique which helps in proving the ownership of the developer. In this paper the design of VBSW is shown. In the remaining part of the paper the design of VBSW, properties of VBSW, and comparison of VBSW model with previous work is given.

1. INTRODUCTION

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[1][2][9][10][11][21]. Software Watermarking is one of the technique which helps in proving the authentication of the developer in which the developer embeds a secret message into cover message[6][7][8][13][14]. 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 [3][4][5][12][15][20] and the software watermark which was used for embedding is a serial number or a unique identification of the developer [16][17][18][19]. 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.

2. Problem Statement

Version Based Software Watermark is one of new approach of a Software Watermark. It is making use of LOC and Version number of software which are the heart of any software. 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 is the latest software release. Every release of the software has a new feature or a set of new features with new version number. 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.

3. Design of Version Watermark

The flow chart for the calculating the Version Watermark:

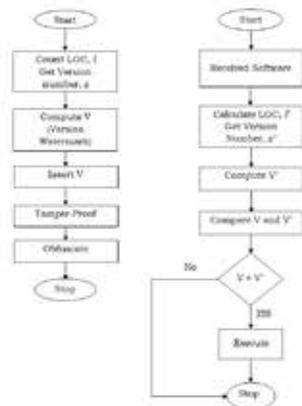


Figure: Owner

Figure: Client

The design of Version Watermark system consists of various phases, in brief it can be explained that the version watermark V is calculated, and inserted into the software, the watermark is subjected to various transformations such as Tamper-proofing and Obfuscation and the watermarked software is uploaded into the Server. The client downloads the software from the Server, and computes the new Version Watermark from the received software and compares it with the received Version Watermark, using a Recognizer function and co

4. Properties of Version Based Software Watermark

4.1 Toughness of a Version Based Software Watermark

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

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

A(S) is the attack function where S is subjected to various attacks, and it is possible to retrieve the original Version Watermark, V.

4.2 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.

4.3 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. COMPARISON

1. In the previous Watermarking techniques various embedding techniques are identified. In these techniques Watermark is a piece of secret information or an identity which was embedded in a binary form into the software.
2. In our technique we have gone a new way of calculating the watermark, which makes use of the LOC and version number of the software. This watermark known as Version Watermark is inserted into the software.
3. In the previous work, the embedding algorithms are making use of three parameters in which a key value is included and it is assumed that the key value is shared between developer and the user.
4. In the proposed technique the Insertion algorithm is con-

sisting of two parameters, and it is not making use a key value. Therefore distribution of key is not a problem in the technique.

5. All the other techniques that are discussed are static or dynamic, but the proposed one is a semi-dynamic in nature.
6. Apart from the exiting attacks such as additive attack, subtractive attack, collusive attack, and distortion attack, two

new attacks are proposed related to the watermark known as version attack and LOC attack, and how to overcome from these attacks are also discussed.

6. CONCLUSION

In this paper the concept of VBSW, design of VBSW, properties of VBSW, and comparison with previous work were studied.

REFERENCE

1. R.J Anderson and F.A.Peticolas , On The Limits of Steganography, IEEE J-SAC, 16(4), pages 474-481, May 1998. | 2. G.Arboit, "A method for watermarking java programs via opaque predicates", in the Fifth International conference on Electronic Commerce Research (ICECR-5), 2002. <http://citeseer.nj.nec.com/arboit02method.html> | 3. P.Cousot and R.Cousot, "An abstract interpretation-based framework for software watermarking", in Principles of Programming Languages 2003, POPL'03, 2003, pp. 311 – 324.er | 4. C.Collberg, E.Carter, S.Debray, A.Huntwork, C.Linn, and M.Stepp, "Dynamic path based software watermarking", in SIGPLAN'04 Conference on Programming Language Design and Implementation, June 2004. | 5. D.Curran, M.O.Cinneide, N.Hurley, G.Silverstre, "Dependency in Software Watermarking", in First International conference on Information and Communication Technologies: from Theory to Applications, 2004, pp. 311 – 324. | 6. Colleberg and C.Thomberson, Watermarking, Tamper-proofing, and Obfuscation – tools for software protection, technical Report 170, Department of Computer Science, University of Auckland, 2000 | 7. C. Colleberg, C. Thomberson, and D. Low, Manufacturing cheap, resilient, and stealthy opaque constructs., in Principles of Programming Languages 1998, POPL'98, San Diego, CA, Jan.1998. | 8. C. Colleberg, C. Thomberson, and D. Low, Breaking Abstractions and Unstructuring data structures, in IEEE, International Conference on Computer Languages, ICCL'98, Chicago, IL, May 1998. | 9. Hirose, N., Okamoto, E., Mambo, M., "A proposal for software protetion", in proc. 1998 Symposium on Cryptography and Information Security, SCIS'98-9.2.C, Jan. 1998. (in Japanese) | 10. K. Holmes, "Computer software Protection", US Patent 5,287,407, assigne: International Business Machines, Feb. 1994 | 11. A.B.Kahng, J. Lach, W.H.Mangoine-Smith, S. Mantik, I.L.Markov, M. Potkonjak, P.Tucker, H.Wang, , and G.Wolfe, "Watermarking Techniques for Intellectual Property Protection", Proc. 35th ACM/ IEEE DAC design Automation Conf. (DAC'98), pp. 776-781, June 1999. | 12. S. Kannan and T.A.Proebsting, Register allocation in structured Programs, Journal of Algorithms, 29(2); 223-237, Nov.1998. | 13. A.Monden, H.Iida, and Kichi Matsumoto, "A practical method for watermarking Java programs", in The 24th Computer Software and Applications Conference, 2000, pp.191 – 197. | 14. A.Monden, H.Iida, and Kichi Matsumoto, K.Inoue, and K.Torii, "Watermarking Java Programs", in International Symposium on Future Software Technology'99, Ocotober 1999, pp. 119 – 124. | 15. G.Myles and C.Collberg, "Software watermarking through register allocation: Implementation, analysis, and attacks", in LNCS 2971, 2004, pp. 274 – 293. | 16. G.Qu and M.Potkonjak, Analysis of watermarking techniques for graph coloring problem, in IEEE/ACM International Cnference on Computer Aided Design, pages 190-193, Nov. 1998. | 17. G.Qu and M.Potkonjak, "Hiding Signatures in graph coloring solutions", in Information Hiding Worshop '99, pp.348 -367. | 18. T.Sahoo and C.Collberg, "Software watermarking in the frequency domain: Implementation, analysis, and attacks", in Computer Science department, University of Arizona (USA), Technical Report, vol. TR04 07, March 2004. | 19. J.Stern, G.Hachez, P.Kouene, and J.J.Quisquater, "Robust object watermarking: Application to code", in Information Hiding Workshop'99, 1999, pp.368 – 378. | 20. R.Venkatesan, V.Vazirani, and S.Sinha, "A Graph Theoretic approach to Software watermarking", in 4th International Information Hiding Workshop Pittsburg, PA, 2001. | 21. X. Zhang, R.Gupta, "hiding Program Slices for Softwrae Security", in First annual IEEE/ACM International Symposium on Code Generation and Optimization, IEEE, 2003, pp. 325 -33