http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (6): 1236-1240, 2010
ISSN 1812-5638

The Software Watermarking for Tamper Resistant Radix Dynamic Graph Coding

Zhou Ping, Chen Xi and Yang Xu-Guang
Guilin University of Electronic Technology, Guilin541004, Guangxi, China

Abstract: This study mainly based on Radix-K dynamic graph software watermark. Aiming to the weak robustness against attacks showing in the existing Radix-K dynamic graph, we propose a software watermark scheme of the tamper resistant radix dynamic graph coding. Considering of the coding efficiency, we introduce constant coding in the program and make the constant coding depend on software watermark coding structure. When attacks tamper software watermark structure, the extracted constants are not correct value and lead the program to fail, then we can effectively protect the software watermark information what is embedded into program. At last, the new dynamic graph watermark scheme is on SandMark system and selects different encoding methods to embed into watermark graph, we analysis of its coding efficiency, robustness, program overload and so on.

Key words: PPCT graph, constant coding, sandmark system, graph topology structure, enumeration encoding graph

INTRODUCTION

As an important branch of information hiding technology, software watermark had attracted more and more people's attention (Myles and Collberg, 2004). According to extraction method, software watermark technology has many categories which can be divided into static and dynamic watermark Venkatesan *et al*., 2001) this also is more mentioned in currently classification methods. At present, more new software watermarking technology research is mostly focusing on the dynamic watermark.

Software watermark is a interdiscipline which involves in cryptography, software engineering, algorithm design and graph theory. The first software watermark study on graph theory is proposed by Venkatesan *et al.* (2001) and is called VVS which is a static watermarking algorithm. Collberg and Thomborson (2002) put forward a dynamic figure software watermark algorithm: CT algorithm the core idea is that embed watermark information into dynamic topology structure, however, the relevance of pointer structure make it difficult to automatic analysis software code. Therefore, Collberg *et al.* (2004) developed a variant algorithm of CT that is a constants coding algorithm (Collberg *et al*., 2004). This study based on the special structure and proposed a software dynamic watermarking scheme on the radix K coding.

DYNAMIC GRAPH SOFTWARE WATERMARK CODING TECHNOLOGY

Dynamic graph software watermark technology (Chen *et al*., 2009; Luo *et al*., 2008) is a scheme that embeds watermark information into software by program dynamically building topology diagram. Major existing dynamic graph topology encoding method is: pareto chart coding, Radix-K list coding and PPCT (Planted Plane Cubic Tree) enumerate coding etc. Pareto chart coding has a better performance against attack and lower coding efficiency. Radix-K list coding (Myles and Collberg, 2005) has the highest coding efficiency and the worst anti-attack performance. However, PPCT enumerate coding has the worst coding efficiency and the best anti-attack performance. The more details analysis and comparison of those coding schemes can refer to the study (Anckaert *et al*., 2004; Stern *et al*., 2000). Thus, you can get a better dynamic map software watermark scheme if you could make full use of the Radix-K list coding and improve its anti-attack performance.

TAMPER RESISTANT RADIX-K CODING DYNAMIC GRAPH

In order to improve the anti-attack performance of Radix-K coding, we introduce a tamper resistant program (Collberg and Thomborson, 2002) what can be immediately perceived tamper to terminate the program

Corresponding Author: Zhou Ping, Guilin University of Electronic Technology, Guilin541004, Guangxi, China

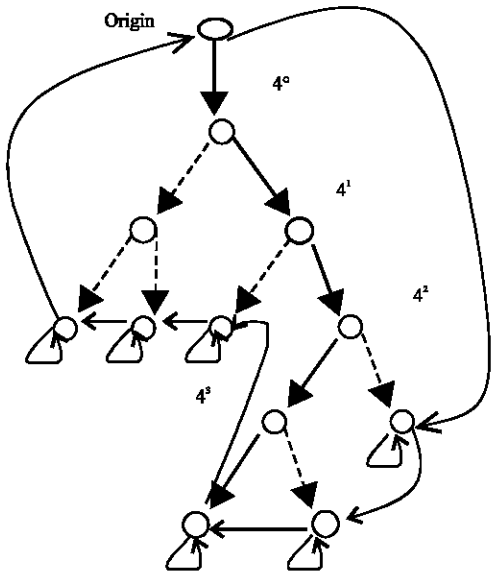

Fig. 1: Radix K = 4 tamper resistant coding, N = 82 = 2×4° +0×4¹+1×4²+1×4³

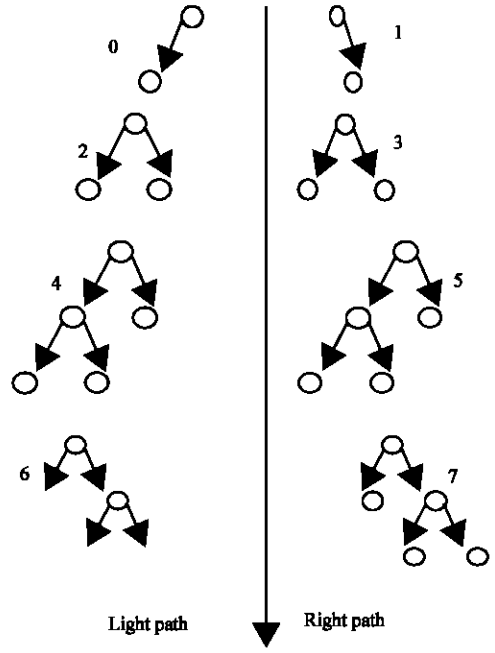

Fig. 2: Coefficient coding

running, thereby protecting the watermarking information. There are many methods about tamper resistant, in this study, we base on the coding graph and combine with PPCT graph form a new tamper resistant radix coding graph. Given coding schemes are as follows:

Coding thinking: According to the radix formulas of watermarking, we ranged each coefficient from small to large and formed the path vectors (Collberg *et al*., 2004), and used the path vectors to encode the coefficient expression area and index lists, then we can make use of the specified path vector to express the follow-up coefficients. Finally, all the leaf nodes' left pointer point to itself and its' right pointer to the next leaf node. Besides, the right pointer of the most left leaf node pointed to Origin generating node, whose right pointer pointed to the most right leaf node that can shape leaf list and the left pointer point to the first index node. So, the graph is same as PPCT graph. As shown in Fig. 1 , in accordance with the number of the radix is 4 for encoding we can get a graph style of the N = 82.

The coefficient coding method: Considering the requirement of coding efficiency and tamper resistant, we know that PPCT graph is the best offensive in currently dynamic graph watermarking (Kommerling and Kuhn, 1999), therefore, PPCTis introduced to represent the coefficients. However, the number of enumeration leaf nodes in PPCT graph is limited, since a number radix coding may have several coefficients, in order to express more coefficients and less nodes in the tamper resistant radix coding graph, use the following rules to encoding the coefficient, if the value is 0 or 1, then use one leaf node PPCT graph and distance to distinguish with its location in index node right or light; if the value is 2 or 3, then use two leaf nodes PPCT graph to distinguish with its location in index node right or light; greater value of coefficients are by analogy. The expression of enumeration coefficient is shown in Fig. 2.

Through the Fig. 2, we can compute the coding coefficient in the following formulas:

$$X(y)=S(y)+2\times IPPCT(y)\times PT(y)\times PT(y)+2\times(Pl(y)\times PT(y)+Pr(y))+d \quad (1)$$

In the Eq. 1, X(y) is the value of coding coefficient whose coefficient coding graph has y numbers of leaf nodes. The IPPCT(y) is the index value of coefficient coding graph. PT(y) is the total arrangement. The PL(y) is the arrangement value of leaf nodes' light indexs. Pr(y) is the arrangement value of leaf nodes' right indexs. d = 0 shows that coefficient coding graph is the right one. d = 1 shows that coefficient coding graph is the light one. S(y) is a Recursive formula, as follow:

$$S(y)=S(y-1)+2\times IPPCT_T(y-1)\times PT(y-1)\times PT(y-1) \quad (2)$$

Tamper resistant methods: In the dynamic graph watermark, in order to prevent attackers from tampering the watermark, graph topology structure has tamper resistant program, if there exists a certain dependence between the host program and watermark, the embedded watermark can be better protected. The purpose of constant coding is to be able to establish the dependence between host program and watermark graph, not only to put their codings into a software. Dynamic graph has there advantages (Xue-Mei and Jie, 2005; Zhu *et al*., 2009), as follow:

- The alias analysis of heap structure is difficult
- Easy to design the watermark data structure
- Easy to use the internal structure of the watermark map to implementing the tamper resistant features

Therefore, the constant coding scheme is a better tamper resistant strategy of dynamic graph watermark.

According to the constant coding, a complete dynamic graph watermark tamper resistant process can be described as follows:

Step 1: Analysis of the candidate program, extract the key constants what can be a variety of basic types or reference types

Step 2: Produces a large watermark figure what is the product of two large prime numbers and encoded the figure into a watermark graph structure with the n-leaf nodes

Step 3: Corresponding to constant integer, we find a watermark graph sub-structure in the watermark. Here sub-structure not only can be a part of the watermark graph also can be a full branch

Step 4: Produces the reference information of constant sub-structure and describes how to find the constant sub-structure in watermarking tree

Step 5: Generated the coding function of the constant sub-structure (which is the main part of the constant decoder), to decode these critical constants when the program is running

Step 6: Produces object code of coded function (i.e., byte-code program)

Step 7: Embeds constant decoder and encode function calls methods into host software

In the above constant tamper resistant coding process, step 4 need to provide some reference information of the constant sub-structure, for this study dynamic graph coding scheme, such informations include of the sub-graph structure's location, boundaries and the base node parameters.

SCHEME IMPLEMENTATION

For the implementation process of dynamic graph constant coding tamper resistant, in order to verify the feasibility (Myles and Collberg, 2004) the scheme is realized on SandMark system platform. In the implementation process, we focuse on the two aspects: the implementation of enumeration graph encoder and decoder and enumeration graph tamper resistant.

Implementation of enumeration coding graph is based on the design rules and coding theory (Qi and Yan-Yan, 2008), converts a watermark to sandmark. Util newgraph. Graph class, we need to write encode method of codes; the other is just a reverse process what needs a decode method of codes. These two methods are the main parts of codec. The new designed codec is called sandmark.util.newgraph.codec. Eradix class, through the interface of SandMark, we design a new enumeration encoding graph codec and realize the dynamic embedding and extract process of watermark number. Then we can finish the implement of enumeration graph code and decode.

In order to enhance the concealment of the watermark, this paper we use pseudo-code watermark to achieving constant coding, the method is to put constant coding into SandMark. The major functions of constant coding components are structural components of pseudo-watermark map, search for constants, constant decomposition and the formation of small-regular quantum map, search for constant sub-map in pseudo-watermark map, construct and embed decoder.

PERFORMANCE ANALYSIS

In this study, we give some types of dynamic graph encoding such as anti-offensive, coding efficiency, time and space overload, then compare the performances of a number differences mainly from the perspective of comparative analysis, which mainly include:

Anti-attack analysis of enumeration graph: Anti-attack analyse of various enumeration graphs, can be summed up in Table 1 in the conclusions.

From Table 1, we can get the following conclusions: base coding is weak in perceiving the attack of add,

Table 1: Four kinds of coded graph of the attacks on perceived ability and comparison of fault-tolerant

Enumeration encoding map	Add or reduce attacks	Distorted attack	Fault-tolerant
Base coding	Weak	Weak	General
Arrange coding	Stronger	General	Weak
PPCT coding	Strong	Strong	Strong
This study coding	Strong	Stronger	Stronger

reduce and distort, but it has the ability to fault-tolerant; arrange coding has the strongest ability to perceive the add and reduce attack, but its fault-tolerant performance is the worst; PPCT coding is good at all of the performances, however, this paper coding method is better than PPCT coding at fault-tolerant and perceiving distort attack.

Constant coding tamper resistant performance analysis: Constant coding in the program can have many constants C, the attacker is difficult to ensure that there are enough test cases to make program execute to all of the encoding functions. Embedding a constant decoder into the host process is not just a watermarking graph, but also bring a certain procedure space overload. When you choose many constants to encode, although only one pseudo-watermark graph could be encoding to many constants. As the number of parameters which embed into constants decoding increase, the value of space overload will be a corresponding increase. Therefore, considering anti-tamper requirements, we also should weigh the impact of space overload.

Analysis of the coding efficiency in enumeration graph: The coding efficiency of enumeration graph scheme is between the base coding and PPCT coding. When the number of watermark is tend to large, compared to PPCT coding, this coding scheme can show a better coding efficiency.

Analysis of the enumeration graph time and space overload: For the characteristics of dynamic graph watermark (Venkatesan *et al*., 2001; Chen *et al*., 2009) the effect of program execution time is very small. If you didn't enter the key sequence of extracting watermark before program running, then the program is still running according to original track, therefor the program can't build a watermark graph structure during the process running, In order to analyze the program space overload of enumeration graph scheme, we compared the differences of watermarked file size. Base coding, PPCT coding and this study coding scheme can produce different effects on space overload: the smallest is the base coding, this paper coding scheme and PPCT number is small and not differ greatly. When the watermark number is large, the differences is more obvious.

DISCUSSION

This study is a dynamic data structure watermark scheme which based on temper resistant Radix-K coding. In order to verify the feasibility of the program, we use the interface SandMark tool platform interface expending method. Experimental results and analysis showed that this watermark scheme had relatively modest coding efficiency and batter robustness and less affect on the program overload.

CONCLUSION

This study was a new scheme on software scheme. For the existing characteristics of base coding are weak against attack, the program planed to design a hybrid coding on PPCT and base coding and introduced a constant coding to implement tamper resistant. Through the experiment, we found that the new coding scheme on dynamic graph had lower computational complexity and easier implement.

ACKNOWLEDGMENT

This research project was fully sponsored by the Educational Office of Guangxi Province Research Project Fund with grant number 200808MS008.

REFERENCES

Anckaert, B., B.D. Sutter and K.D. Bosschere, 2004. Software piracy prevention through diversity. Proceedings of the 4th ACM Workshop on Digital Rights Management, Oct. 25-25, ACM, Washington DC, USA., pp: 63-71.

Chen, X.J., D.Y. Fang and J.B. Shen, 2009. A dynamic graph watermark scheme of tmper resistance. Inforam. Assurance Security, 1: 3-6.

Collberg, C.S. and C. Thomborson, 2002. Watermrking, tamper-proofing and obfuscation tools for software protection. IEEE Trans. Software Eng., 28: 735-746.

Collberg, C., E. Carter, S. Debray, H. Huntwork, J. Kececioglu, C. Linn and M. Stepp, 2004. Dynamic path-based software watermarking Proceeding of the ACM SIGPLAN 2004 Conference on Programming Language Design and Implementation. June 9-11, ACM, Washington DC, USA., pp: 107-118.

Kommerling, O. and M.C. Kuhn, 1999. Design principles for tamper-resistant smartcard processors. Proceedings of the USENIX Workshop on Smartcard Technology, May 10-11, USENIX Association Berkeley, Chicago, Illinois, USA., pp: 9-20.

Luo, Y.X., J.H. Cheng and D.Y. Fang, 2008. Dynamic graph watermark algorithm based ion the threshold scheme. Inform. Sci. Eng., 2: 689-693.

Myles, G. and C. Collberg, 2004. Software Watermarking Through Register Allocation: Implementation, Analysis and Attacks. In: Information Security and Cryptology-ICISC 2003, Lim, J.I. and D.H. Lee (Eds.). LNCS., 2971, Springer-Verlag, Berlin, Heidelberg, ISBN-13: 978-3-540-21376-5, pp: 274-293.

Myles, G. and C. Collberg, 2005. K-gram based software birthmarks. Proceedings of the 2005 ACM Symposium on Applied Computing, March 13-17, IEEE Computer Society, Santa Fe, New Mexico, pp: 314-318.

Qi, T. and L. Yan-Yan, 2008. Research and application of dynamic graph watermarking algorithm. Comput. Appl. Technol., 3: 57-60.

Stern, J.P. and G. Hachez, F. Koeune and J.J. Quisquater, 2000. Robust Object Watermarking: Application to Code. In: Information Hiding, Pfitzmann, A. (Ed.). LNCS., 1768, Springer-Verlag, Berlin, Heidelberg, ISBN-13: 978-3-540-67182-4, pp: 368-378.

Venkatesan, R., V. Vazirani and S. Sinha 2001. A Graph Theoretic Approach to Software Watermarking. In: Information Hiding, Moskowitz, I.S. (Ed.). LNCS., 2137, Springer-Verlag, Berlin, Heidelberg, ISBN-13: 978-3-540-42733-9, pp: 157-168.

Xue-Mei, B. and L. Jie, 2005. A transaction scheme of tamper-proofing software watermarks based on dynamic gragh. Network Secuity Appl., 7: 1013-1013.

Zhu, J., Y.H. Liu and K.X. Yin, 2009. A noval dynamic graph software watermark scheme. Technol. Comput. Sci. 3: 775-780.