Watermarking Essential Data Structures for Copyright Protection

Qutaiba Albluwi and Ibrahim Kamel

Department of Computer Engineering
University of Sharjah, UAE
qutaiba@sharjah.ac.ae, kamel@sharjah.ac.ae

Abstract. Software watermarking is a new research area that aims at providing copyright protection for commercial software. It minimizes software piracy by hiding copyright signatures inside the program code or its runtime state. Prior proposals hide the watermarks in dummy data structures, e.g., linked lists and graphs that are created during the execution of the hosting software for this reason. This makes it vulnerable to subtractive attacks, because the attacker can remove the data structure without altering the operation or the semantic of the software program. In this regard, we argue that hiding watermarks in one or more data structures that are used by the program would make the watermark more robust because removing the watermark would alter the semantic and the operations of the underlying software. However, the challenge is that the insertion of the watermark should have a minimal effect on the operations and performance of the data structure.

This paper proposes a novel method for watermarking R-tree data structure and its variants. The proposed watermarking scheme takes advantage of the redundancy in the way the entries within R-tree nodes are ordered. R-trees do not require ordering the entries in a specific way. Node entries are re-ordered in a way to map the watermark. The new order is calculated relative to a "secret" initial order, known only to the software owner, using a technique based on a numbering system that uses variable radix and factorial base. The addition of the watermark in the R-tree data structure neither affects the performance nor increases the size of the R-tree. The paper provides a threat model and analysis to show that the watermarked R-trees are robust and can withstand various types of attacks.

Keywords: Software watermarking, copyright protection, data hiding, indexing, multimedia database.

1 Introduction

Software piracy is one of the main threats targeting software development. A recent study [18] shows that 36% of the software programs used nowadays are pirated. Watermarking is a technique used to provide copyright protection for intellectual properties. Recently there have been a lot of interests in securing and protecting databases [32] [4] [21] [35].

D. Pointcheval, Y. Mu, and K. Chen (Eds.): CANS 2006, LNCS 4301 , pp. 241 – 258, 2006.

Watermarking means the embedding of digital information into the original work [27]. Watermarks are commonly known in copyright protection of multimedia objects, e.g., images [16] [14], video [20] [23], and audio [11],[29]. In addition to copyright protection, watermarking is used in authentication and privacy protection. Unlike authentication applications, watermarks in copyright protection and privacy applications need to be robust and invisible.

Recently, there has been a lot of interest in applying watermarking techniques to protect the copyrights of the data and the software. Unfortunately, most of the work in this area are trade secrets and are not published.

Technically, most of the software watermarking techniques fall under two categories: *static* watermarking [17] [39] and *dynamic* watermarking [6][30]. In *static* watermarking the watermark is stored in the source code, either in the data section or in the code section. For example, a watermark can be stored in the values of the constants or in debugging information. On the other hand, *dynamic* watermarking stores the watermark in the program's execution state. Static watermarking techniques are considered more fragile as they can be easily attacked by code optimizers or obfuscators [6]. For example, a static watermark that is saved in data strings can be easily attacked by breaking up all strings into substrings scattered over the executable. Unlike static watermarking, in dynamic watermarking the watermarks are generated during the program execution. Usually the watermark is a large integer number[1]. To make the watermark more credible legally, the large integer is chosen to be the multiplication of two large prime numbers [6] [31]. In general, dynamic watermarks are more robust than static ones and can withstand more sophisticated attacks [7]. Our proposed technique falls under the dynamic watermarking category.

Prior techniques in dynamic watermarking hide the watermark in data structures that are built specially for this purpose during the execution of the program. The fact that the data structure is built specifically to house the watermark and it is independent of the application semantic makes the watermark susceptible to subtractive (or removal) attacks. The operation and semantic of the host program will not be affected by removing the data structures that hide the watermark.

We argue that hiding watermarks in data structures, which are used by the program, would make them more robust; because tampering with these data structures would affect the program correctness and/or performance. Software products usually use a number of both memory-based and disk-based data structures e.g., binary trees, linked lists, graphs, B-trees, and R-trees. In general, each data structure requires a watermarking technique that is different from the others. The diversity in the watermarking techniques used in one program will definitely increase the robustness of the whole system and thus, decreases the likelihood of successful attacks.

Notice that disk-based data structures are easier to attack than memory-based data structures because an adversary does not need to execute the application program to attack the watermark. Rather, the adversary can invoke an off-line attack with an independent code. With our proposed algorithm the owner can prove his/her claim by inserting enough records into the data structure to rebuild the watermark.

[1] This does not limit the scope of these techniques because any text watermark can be mapped to a numeric watermark.