Chapter 39
Scaling Up Software Birthmarks Using Fuzzy Hashing

Takehiro Tsuzaki
Graduate School of Kyoto Sangyo University, Kyoto, Japan

Teruaki Yamamoto
Kyoto Sangyo University, Kyoto, Japan

Haruaki Tamada
Kyoto Sangyo University, Kyoto, Japan

Akito Monden
Okayama University, Okayama, Japan

ABSTRACT

To detect the software theft, software birthmarks have been proposed. Software birthmark systems extract software birthmarks, which are native characteristics of software, from binary programs, and compare them by computing the similarity between birthmarks. This paper proposes a new procedure for scaling up the birthmark systems. While conventional birthmark systems are composed of the birthmark extraction phase and the birthmark comparison phase, the proposed method adds two new phases between extraction and comparison, namely, compression phase, which employs fuzzy hashing, and pre-comparison phase, which aims to increase distinction property of birthmarks. The proposed method enables us to reduce the required time in the comparison phase, so that it can be applied to detect software theft among many larger scale software products. From an experimental evaluation, the authors found that the proposed method significantly reduces the comparison time, and keeps the distinction performance, which is one of the important properties of the birthmark. Also, the preservation performance is acceptable when the threshold value is properly set.

DOI: 10.4018/978-1-7998-3016-0.ch039

1. INTRODUCTION

Till today, software theft has been causing serious damage to software industry. From the BSA global software survey 2016[1], 39% of software installed on computers in the world is not properly licensed. Also, violation of open source software (OSS) licenses, such as GPL[2], by unexpected and unaware reuse of OSS source code[3] has now become a serious problem for both software companies and OSS developers (Monden et al., 2011). Software birthmark methods have been proposed against such software theft to enable us to detect the theft (Tamada et al., 2004), (Tamada et al., 2005). A software birthmark is a set of characteristics which a program originally possesses. It is extracted from a binary code and used to evaluate the similarity between one program and another (extraction and comparison phases). Various types of birthmarks have been proposed, each focusing on different characteristics in a program. Different extraction methods and comparison methods have also been defined for each type of birthmark and have been evaluated according to those definitions.

Software birthmarks are designed to search of large amounts of software to detect suspected copies; hence, their use requires high-speed, large-volume software repository searches. However, the software birthmark has the one essential problem in the practical use case. That is, the scale of the target software was not assumed. Figure 1 illustrates the problem in use of the software birthmarks. The developer can examine for detecting the copy of p_0 from the target set programs p_1 to p_n. However, the many unchecked programs are still existing in the Internet. The most important issue of the software theft is to detect suspected copies. The programs p_{n+1} to p_{n+m} in the Figure 1 are never investigated because memory constraints, vast amount of time consumed for comparison, and the enormous computational complexity. However, almost programs are innocent and quite different. Therefore, to detect the software theft requires more simple and casual algorithm for huger target set.

Therefore, we proposed the method for the software birthmark procedure to narrow the defendants with compressing birthmark information and simplifies comparison algorithms. Figure 2 shows the difference of the conventional and the proposed birthmark procedures. Form Figure 2, we insert two phases, compression phase and pre-comparison phase, between the conventional phases. The compression phase compresses the birthmark information for the next phase. The pre-comparison phase compares compressed birthmarks by simple algorithm and computes similarity. Then, remains of the pre-comparison are still defendants, then, the remains are the inputs for the comparison phase.

This remainder of this paper is organized as following. Section 2 reviews the related works. Section 3 describes the proposed method and illustrates the novel procedure of the birthmark system. Section 4 represents the empirical studies of our method. Section 5 shows conclusion and some future works.

2. RELATED WORKS

Birthmarks are a concept that was proposed by Tamada et al. as a method for detecting software theft (Tamada et al., 2004), (Tamada et al., 2005). Characteristics unique to a program that are contained in the program's binary code are extracted as birthmark information and used to measure similarity. Unlike software watermarks, there is no need for prior information embedding; characteristics unique to the program are taken from the compiled binary and defined as the birthmark. Several different types of birthmark that focus on a different program characteristics have been proposed (Chan et al., 2012), (Choi et al., 2009), (Jhi et al., 2011), (McMillan et al., 2012), (Schuler et al., 2007), (Park. et al., 2008).