Synergy 2025: A Multidisciplinary Forum for Collaborative Research and Innovation
January 2025
ISBN Number: 978-93-95305-78-5

REVIEW PAPER ON WATERMARKING ALGORITHM FOR PROTECTING SOFTWARE CODE AGAINST CYBER-ATTACK

Bharati Devidas Patil
Research scholar
Department of computer science and Engineering,
P.K. University, Shivpuri(MP),
India bharatip175@gmail.com
Prof. Dr. Rohita Yamaganti Research
Supervisor
Department of computer science and Engineering,
P.K. University, Shivpuri (MP), India

Abstract

We identify three types of attack on the intellectual property contained in software and three corresponding technical defenses. A defense against reverse engineering is obfuscation, a process that renders software unintelligible but still functional. A defense against software piracy is watermarking, a process that makes it possible to determine the origin of software. A defense against tampering is tamper-proofing, so that unauthorized modifications to software (for example, to remove a watermark) will result in nonfunctional code. We briefly survey the available technology for each type of defense. In the past, being able to download open-source code or clone it and then subsequently modify it, encouraged development of software without authorization. Often, developers and organizations mentioned the copyright of the open source, at the most.

1. Introduction

In today’s digital era, software is integral to our daily lives, powering everything from our smartphones to critical infrastructure. However, with the widespread distribution of software, concerns about copyright infringement and unauthorized modifications are growing. Traditional methods like code obfuscation often fall short in providing comprehensive protection. This is where source code watermarking comes into play, offering a robust solution to enhance software security.

In the past, being able to download open-source code or clone it and then subsequently modify it, encouraged development of software without authorization. Often, developers and organizations mentioned the copyright of the open source, at the most.

1.1 What is Source Code Watermarking?

Source code watermarking involves embedding a unique, covert marker within the code. This marker serves multiple purposes, including copyright assertion, software traceability, and tamper detection. Unlike obfuscation, watermarking does not alter the functionality of the code, making it a more reliable and transparent method for protecting intellectual property.

1.2 Need for Source Code Watermarking

The primary reasons for adopting source code watermarking are:

- **Copyright Protection**: Watermarking asserts ownership and helps in resolving disputes related to intellectual property.
- **Software Traceability**: It helps track the distribution and usage of software, identifying unauthorized copies.
- **Tamper Detection**: Watermarks can reveal unauthorized modifications, ensuring the integrity of the software.

2. The Process of Watermarking

Implementing watermarking involves several steps:

1. Embedding: Inserting the watermark into the source code. This can be done at various stages of the development lifecycle.
2. Compilation: Converting the watermarked source code into executable form.
3. Distribution: Distributing the software with the embedded watermark.
4. Verification: Detecting and verifying the watermark in the distributed software to ensure integrity and ownership.

Following figure 1 shows watermark embedding Process

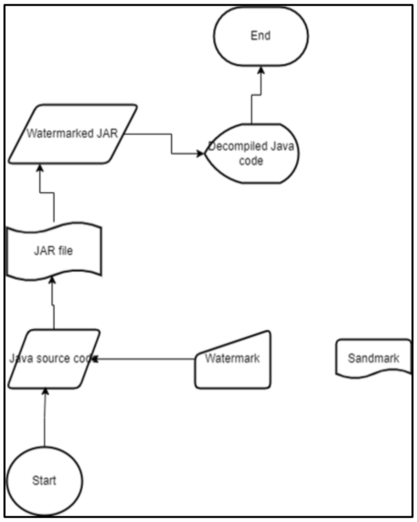

Fig 1 watermark embedding process

Following figure 2 shows watermark recognition process which uses input file and key to recognize watermark.

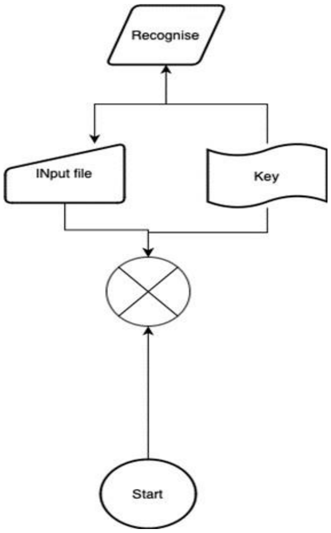

Fig.2 Process of recognizing watermark

3. Literature Review

There are several watermarking techniques, each with its own strengths and applications. Watermarking techniques can be based on three major categories, based on robustness & embedding techniques.

Here we shall consider based on embedding techniques, these are the following:

3.1 Static Watermarking:

Embeds the watermark into the code structure, making it a part of the compiled program. This method is highly resilient against reverse engineering. This technique hides a watermark within the source code by introducing a fake expression, assigned to a new local variable. The watermark is identified by this variable's unique name [1,2].

How It Works:

- The watermark is represented as a numeric value, divided into two-digit segments.
- These segments are embedded into the program by modifying existing constants in the code.
- The watermark is encoded in the sum of these altered constants.

3.2 Zero-Digital Watermarking

Zero watermarking is a unique method that protects code without altering it. Instead of embedding visible marks, it analyzes the code to identify key features like its structure and meaning and mathematically converts these features into a “watermark.” This watermark is stored separately, leaving the original code untouched [3].

KeySplitWatermark

The KeySplitWatermark algorithm is a cutting-edge software protection method that uses blind zero watermarking to embed ownership information within source code without altering its functionality. Does not modify the original code but uses external metadata to assert ownership. This method is highly secure but can be complex to implement [4,5].

EmbeddingPhase:

The algorithm operates in two stages: embedding and extraction. During embedding, the code is analyzed to identify frequently used keywords and characters. These are used to segment the code, and a unique key is generated based on these segments and code properties. This key, along with the original watermark, can be registered with a trusted third party, such as a Certification Authority (CA), for ownership verification.

ExtractionPhase:

In the extraction phase, the code is re-partitioned using the registered keywords. Statistical analysis is conducted to identify the most frequent characters in each segment, forming a characteristic list. The watermark key is then used to extract the embedded watermark, which is compared to the registered one to verify ownership.

3.2 Dynamic Watermarking:

Inserts the watermark during program execution. It adapts to runtime conditions, making it harder to detect. It embeds a watermark within a program’s runtime structure. The watermark is extracted by running the program with a specific input sequence, known as the watermark key. This dynamic watermarking method embeds the watermark into a graph structure that evolves during the program’s execution[6]. The algorithm operates through several sequential phases:

1. **Annotation:** The source code is annotated by inserting calls to the annotator.mark() function at specific locations, marking spots for potential watermark insertion.
2. **Compilation:** The annotated program is compiled and packaged into a Java jar file.
3. **Tracing:** The program is executed using a secret input sequence, generating a trace file that records the sequence of mark () calls.

4. **Embedding:** The watermark is embedded into the program using the data from the tracing phase, seamlessly integrating it into the program's runtime structure.

4. Common Types of Cyber Attacks

- **Malware:**

A malware cyberattack involves malicious code used to infiltrate and compromise a computer system, network, or device without the owner's consent. Many users are familiar with viruses, but others include worms, trojans, and ransomware. Once installed on a host device, malware is programmed to spread to other areas. Its damage can range from minor inconveniences to severe data breaches and financial losses[5].

- **Phishing and Spear Phishing:**

Spear phishing is a more targeted form in which an email is customized for specific individuals or organizations using personal details to appear more convincing. An example might be a request from the CFO to someone in accounts payable to perform a financial transaction by clicking a link or someone receiving a shared file from their boss that they are to click on to open. These tailored approaches make spear phishing attacks particularly deceptive and potentially more successful than broader phishing attempts [5].

- **Ransomware:**

The motive behind a ransomware attack is extortion. Ransomware is delivered through phishing emails or malicious downloads. The malware then encrypts the victim's files and makes them inaccessible. The attackers then demand a ransom for a decryption key to unlock the files. This cyberattack can cause significant disruption and financial damage to individuals and organizations. Ransomware has consistently ranked as one of the most concerning cyber threats for business leaders in recent years [4].

- **Distributed Denial-of-Service (DDoS):**

A Distributed Denial-of-Service (DDoS) attack aims to disrupt the regular traffic of a targeted server, service, or network by overwhelming it with a flood of internet traffic from multiple sources. Attackers typically use a network of compromised computers and devices called bots to generate this traffic. DDoS attacks can cause websites and online services to become slow or unavailable, resulting in lost business, damaged reputation, and potential data breaches [7].

- **SQL Injection:**

In a SQL injection attack, a threat actor inserts unauthorized SQL code into application queries to access database information. Attackers manipulate user input fields, such as login forms or search bars, by inserting malicious SQL code. When the application processes this input without proper sanitization, it inadvertently executes the malicious SQL commands within the database. This technique can allow attackers to bypass authentication, access sensitive data, modify database contents, or perform administrative operations.

- **Cross-Site Scripting (XSS):**

Cross-site scripting (XSS) attacks involve the injection of malicious scripts into trusted websites. These scripts are often in the form of JavaScript code. When unsuspecting users visit these compromised pages, their browsers execute the injected scripts. These scripts can steal sensitive data like cookies, session tokens, or login credentials. This allows them to perform actions on behalf of the user, such as making unauthorized transactions. XSS attacks are classified into three main types: stored, reflected, and DOM-based, with each varying in how the malicious script is injected and executed.

- **Botnets**:

Botnets are networks of compromised computers or devices that one or more attackers control. These infected machines are called “bots” or “zombies” and are unknowingly enlisted to perform coordinated malicious activities after being compromised through malware infection. Botnets can consist of thousands or even millions of devices, including computers, smartphones, and IoT gadgets. They are commonly used for DDoS attacks, spam campaigns, and cryptocurrency mining.

5. Different Strategies to Prevent Cyber Attacks

Below are some strategies outlining how to avoid cyber-attacks. There are many ways to prevent cyber-attacks, and when implemented collectively, they will help create a multilayer strategy that will significantly contribute to cyber-attack protection [7]:

- **Use Strong, Unique Passwords**: The first step in preventing unauthorized access is creating strong, unique passwords. A strong password should include a mix of upper- and lower-case letters, numbers, and special characters. It would be best to consider using a password manager to securely store and generate complex passwords to reduce the risk of password reuse.

- **Enable Multi-Factor Authentication (MFA):** Multi-factor authentication adds an extra layer of security by requiring two or more verification methods to access an account. These typically include something you know (password), something you have (smartphone or security token), and/or something you are (biometric data).
- **Regularly Update Software:** Software updates often include critical security patches that address newly discovered vulnerabilities. Your organization should have an effective patch management strategy. This starts by maintaining an inventory of all software and their versions and enabling automatic updates for all software, if possible. Regularly check for updates and apply manually when necessary. Be sure to prioritize critical security updates.
- **Implement Firewalls and Endpoint Protection:** A network perimeter firewall serves as the first line of defense against cyber threats, but you should also implement firewalls throughout the local network to isolate, segregate, and protect critical servers and sensitive data. Firewalls can be paired with endpoint protection software on all devices to create a comprehensive security barrier against malware, ransomware, and other cyber threats.
- **Reduce Privileges and Manage Data Access:** Cybercriminals very often try to render an impact on the data an organization has in its possession, either extorting the organization or abusing the data for other purposes, like getting close to the next, bigger target. Reducing privileges given to identities (accounts) in your organization, converting administrative accounts to ephemeral ones, and governing data access are interrupting the tools, tactics, and procedures used by cybercriminals [8.9].
- **Encrypt and Backup Data:** Unfortunately, achieving complete immunity from cyberattacks is unrealistic, as zero-day vulnerabilities and evolving threat landscapes are persistent challenges in the digital world. Regular backups are essential to restore systems and recover data in case of a successful attack. Encrypting sensitive and proprietary data as encryption renders the data unreadable without the proper decryption keys [10].

6. Conclusion

In this paper we have discuss different types of watermarking technique such as static, Dynamic and zero watermarking. Along with that we discuss Cyber attcks and its different types. Source code watermarking is a powerful tool in the arsenal of software security. By embedding unique markers into the code, developers can protect their intellectual property, ensure traceability, and detect tampering. In this work, we proposed KeySplitWatermark, a novel zero watermarking approach to protect software code against cyber-attacks. The algorithm is blind and adds watermark logically into the code using the inherent properties of code and provides a robust solution.

References

[1] A. R. Javed, M. O. Beg, M. Asim, T. Baker, and A. H. Al-Bayatti, "Detecting motion-based side-channel attack using smart phone keystrokes", J. *Ambient Intell. Humanized Comput.*, pp. 114, Feb. 2020.

[2] A. K. Abdulrahman and S. Ozturk, "A novel hybrid DCT and DWT based robust watermarking algorithm for color images", *Multimedia Tools Appl.*, vol. 78, no. 12, pp. 1702717049, Jun. 2019.

[3] W. Hu, R.-G. Zhou,J.Luo,andB.Liu, LSBs-basedquantumcolorimages watermarking algorithm in edge region, Quantum Inf. Process., vol. 18, no. 1, p. 16, Jan. 2019.

[4] Z. Jalil and A. M. Mirza, "An invisible text watermarking algorithm using image watermark, in Innovations in Computing Sciences and Software Engineering." *Dordrecht*, *The Netherlands*: *Springer*, 2010.

[5] C. Iwendi, Z. Jalil, A. R. Javed, T. Reddy G., "Keysplitwatermark: Zero Watermarking Algorithm For Software Protection Against Cyber-Attacks",*Ieee Acess* ,March2020 .

[6] M.E. Kumar, G.T. Reddy, K. Sudheer, M. Reddy, R. Kaluri, D. S. Rajput, and K. Lakshmanna, "Vehicle theft identification and intimation using gsm &iot", in *Proc. Mater. Sci. Eng. Conf.*, *vol*. 4, 2017.

[7] G. T. Reddy, R. Kaluri, P. K. Reddy, K. Lakshmanna, S. Koppu, and D. S. Rajput, "A novel approach for home surveillance system using IoT adaptive security", *in Proc. Int. Conf. Sustain. Comput. Sci.*, *Technol. Manage.*, vol. 3. 2019.

8] G. T. Reddy, K. Sudheer, K. Rajesh, and K. Lakshmanna, "Employing data mining on highly secured private clouds for implementing a security asa-service framework", J. *Theor. Appl. Inf. Technol.*, vol. 59, no. 2, pp. 317326, 2014.

[9] R. Raghavan, J. K. Singh, T. G. Reddy, K. Sudheer, P. Venkatesh, and S. O. Olabiyisi, "A case study: Home environment monitoring system using Internet of Things", *Int.* J. *Mech. Eng. Technol.*, vol. 8, no. 11, pp. 173180, 2017.

[10] J. Epstein, S. Matsumoto, and G. McGraw, "Software security and SOA: Danger, willrobinso", *IEEESecur.PrivacyMag.*, vol.4, no.1, pp. 8083, Jan. 2006.