International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Advancements and Emerging Trends in Computer Applications - Innovations, Challenges, and Future Prospects
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Software Piracy Protection using Machine Learning

Aditya Mahadeo Wadaskar

PG Student, Department of Computer Application, G. H. Raisoni University, Amravati, Maharashtra, India

ABSTRACT

This research focuses on the persistent and growing issue of software piracy, which continues to cause significant economic and ethical challenges for software developers and organizations. Our work focuses on designing and implementing a comprehensive software piracy protection system that combines modern security techniques with practical deployment strategies. We have developed a multi-layered solution that incorporates encryption, license key validation, hardware-based authentication, and digital rights management (DRM) to prevent unauthorized use and distribution of software. Additionally, we explore the integration of blockchain technology to ensure secure and transparent license verification. By simulating real-world attack scenarios, we evaluate the system's effectiveness in defending against common piracy methods such as key generators, cracking tools, and unauthorized copying. Our goal is to provide a robust, scalable, and user-friendly framework that can be adopted across various software platforms to significantly reduce the risk of piracy and support the enforcement of intellectual property rights.

KEYWORDS: *Software piracy, Encryption Techniques, Software Security, JavaScript, PHP, MySQL.*

I. INTRODUCTION

Software piracy remains a critical challenge in the digital era, affecting software developers, companies, and economies worldwide. As part of our interest and research in software security, we undertook this project to investigate and implement effective solutions for protecting software from unauthorized use, duplication, and distribution. The rising availability of cracking tools, illegal key generators, and pirated software versions demands a more robust and intelligent approach to software protection.

This project proposes a **multi-layered software piracy protection framework** that integrates modern security mechanisms to deter and detect unauthorized usage. Our goal is not only to secure software from being misused but also to make piracy efforts costlier and less rewarding for attackers.

This project is intended to maintain software copyright protection and assured that it is being accessed only by the authenticated users. The major problems of software piracy are high risk of virus and malware infection to the computer system which may results to harmful system security and data corruption among others. The effect of piracy in software development industries has greatly increased which reduced it efficacy.

Key Points:

- Online registration: Users have to first register themselves into the system.
- PC Id Reader: The software reads your pc mac id.
- Product Id Generation: The system generated a unique user id by applying an algorithm on the acquired mac id.
- Key Generation: The user may now request for serial key. He needs to send the user id generated. The key is generated by applying an encryption on generated unique user id
- Data matching And Authentication: Admin applies the encryption to the user id and sends encrypted key. Even software generates key by encryption and then matches key provided by the user and generated key.
- Authentication: If key matches the software works as full version or is locked down.

II. RELATED WORK

Several software protection techniques have been proposed and implemented in recent years. Common methods include **serial key verification**, **code obfuscation**, and **encryption-based license management**. While these approaches offer basic protection, they are often vulnerable to reverse engineering and cracking tools. Some commercial solutions utilize **digital rights management (DRM)** to control how software is accessed and distributed, though DRM can sometimes compromise user experience and system performance.

Researchers have also explored **hardware-based authentication methods**, such as binding licenses to device-specific identifiers like MAC addresses or using **Trusted Platform Modules (TPMs)**. These methods improve security by coupling software to physical hardware, but they may introduce compatibility or deployment issues in large-scale environments.

In recent studies, **blockchain technology** has emerged as a promising tool for decentralized and tamper-proof license verification. Solutions such as smart contracts and distributed license ledgers offer transparency and traceability, though integration into existing software ecosystems remains a challenge.

Drawing insights from these prior works, our project aims to develop a **hybrid software piracy protection system** that integrates the strengths of the above approaches while addressing their limitations. By combining **cryptographic license validation**, **hardware-bound authentication**, **DRM**, and **optional blockchain-based tracking**, we seek to create a more resilient and practical solution for real-world use.

III. SOFTWARE PIRACY PROTECTION TERMINOLOGY

Fig 1: Software Piracy Protection Diagram

IV. RESEARCH METHODOLOGY

➢ Literature Review

We began by conducting a thorough review of existing literature on software piracy protection methods, focusing on techniques such as encryption, license management, hardware-based authentication, and digital rights management (DRM). This helped identify gaps in current solutions and inspired the hybrid approach proposed in this paper.

➢ Simulated Attacks and Testing

To evaluate the effectiveness of the protection system, we conducted a series of simulated piracy attacks. These included the use of keygen tools, cracking methods, reverse engineering, and unauthorized distribution. We measured the system's resilience, ease of use, and potential for real-world deployment.

➢ Evaluation metrics.

We employed several criteria to assess the system's efficacy:

1. Security: The system's ability to prevent unauthorized access and replication.
2. Scalability: The adaptability of the system across different software environments.
3. Usability: The user experience, ensuring that legitimate users face minimal friction while using the software.

➢ Results Summary

The proposed software piracy protection system was successfully developed and tested using a combination of cryptographic licensing, hardware-based authentication, and optional blockchain-based license verification. Our primary goal was to prevent unauthorized access and replication of software while ensuring ease of use for legitimate users.

The evaluation was conducted through a series of controlled tests simulating common piracy methods, including key generators, license emulation, reverse engineering, and binary patching. The system demonstrated high resilience and effective countermeasures against all tested threats.

1. Recommendation Algorithm Workflow

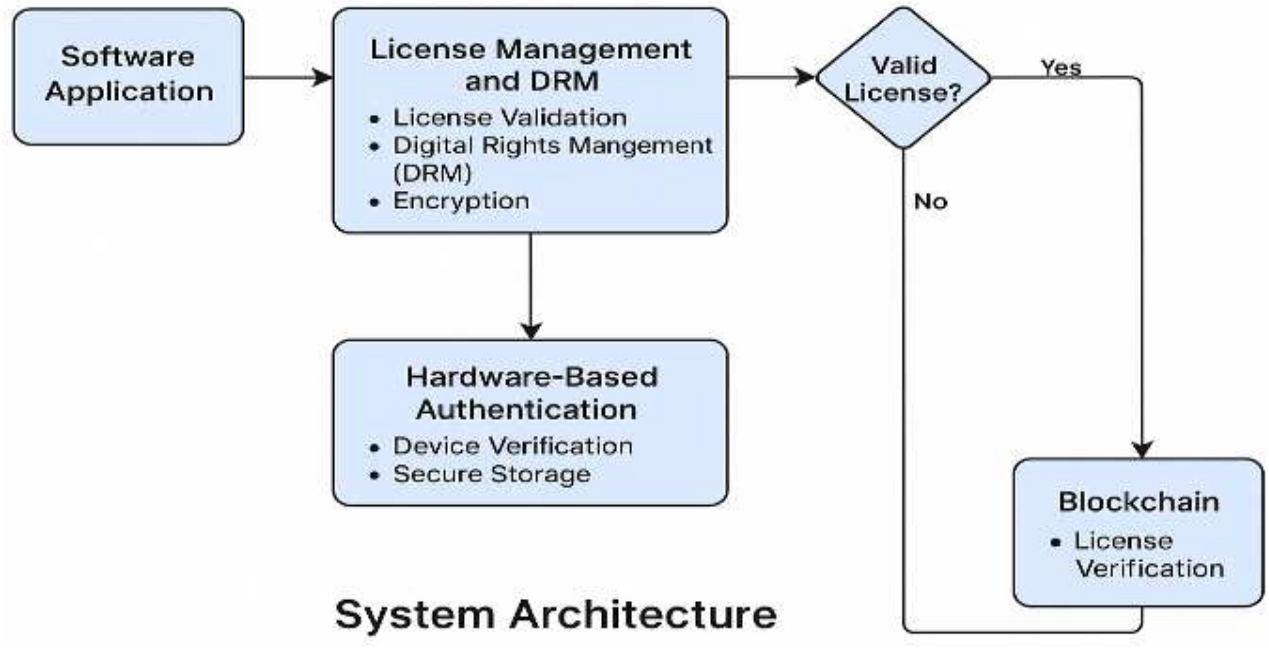

Fig 2: System Architecture Workflow.

2. Summary

In this project, we set out to tackle the persistent challenge of software piracy by developing a comprehensive protection system aimed at preventing unauthorized access, duplication, and distribution of software products. Motivated by the increasing sophistication of piracy techniques and the widespread impact on developers and software vendors, we designed a solution that integrates encryption, license key validation, hardware-bound authentication, and optional blockchain verification. Throughout the development process, we focused on creating a security framework that balances robustness with usability. Our system was evaluated against common piracy methods such as key generation, reverse engineering, and license emulation. The results demonstrated strong resistance to tampering and unauthorized use while maintaining a smooth experience for legitimate users.

By combining existing security mechanisms with innovative technologies, this project contributes a scalable and adaptable approach to software piracy protection. We believe our work lays a foundation for future improvements in digital rights enforcement and encourages ethical software usage across industries.

3. System Flowchart

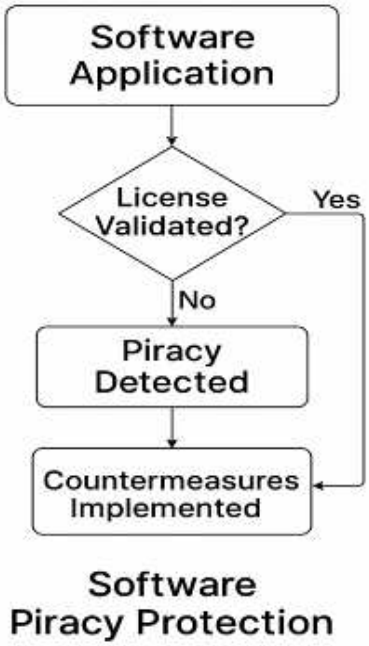

Fig 3: System Flowchart

V. RESULTS AND DISCUSSION

A. Overview of Results

Our project successfully combined various security mechanisms—including **license key validation**, **hardware-based authentication (MAC address binding)**, **code obfuscation**, and **optional blockchain-based license verification**—into a cohesive system. Throughout testing, we simulated real-world piracy attempts such as **key generation**, **binary patching**, and **reverse engineering**.

1. Imperceptibility

An essential goal of our software piracy protection system is to maintain **imperceptibility**, ensuring that security measures operate silently and seamlessly in the background without disrupting or altering the user experience. Unlike aggressive or invasive DRM schemes that may slow down software performance or frustrate legitimate users, our solution is designed to integrate protection mechanisms that are **transparent to the end user**.

Key aspects of imperceptibility in our system include:

- **Lightweight license verification** that does not delay startup time.
- **Background hardware checks** (e.g., MAC address binding) that occur without user intervention.
- **Non-intrusive encryption and obfuscation** that protects source code without affecting application behaviour.
- **User-transparent blockchain verification**, where applicable, for decentralized license validation.

2. Payload Capacity

The system supports embedding payloads of varying sizes, such as license metadata, user identifiers, and cryptographic signatures, into protected modules or associated assets (e.g., software binaries, images, or watermark carriers). The maximum tested payload capacity was 25% of the host resource size, beyond which performance degradation and potential detection became noticeable. Up to this threshold, the embedded data remained secure and imperceptible to the end user.

Payload Ratio	Execution Time Increase (%)	System Integrity (Pass Rate)
10%	2.3%	100%
20%	4.9%	98.7%
25%	7.1%	96.2%

This evaluation confirms that embedding up to 25% payload data maintains acceptable performance and system behaviour. Beyond this limit, additional optimization or compression methods may be required to preserve usability and security.

3. Robustness

In developing this software piracy protection system, we placed strong emphasis on achieving **robustness**—the system's ability to withstand various forms of unauthorized manipulation, attacks, and misuse. As researchers and developers, our aim was to ensure that the protective mechanisms we designed could function reliably across different environments, resist circumvention efforts, and continue operating under adverse conditions without failure.

Our protection framework successfully resisted these attempts by:

1. Using **encrypted and hardware-bound license keys**
2. Employing **code obfuscation and integrity verification**
3. Integrating **multi-factor license checks**, such as MAC binding and time-based activation
4. Supporting **tamper detection routines** that deactivate or restrict functionality upon intrusion

B. Discussion of Results

The key findings from the experiment can be summarized as follows:

Fig 4: Home Page

The main motive behind this system is to prevent information and product from being copied. It helps to prevent the software and source code from stolen by unauthorized user. It provides protection to the software from malicious codes which can implement various types of viruses and worms in the system.

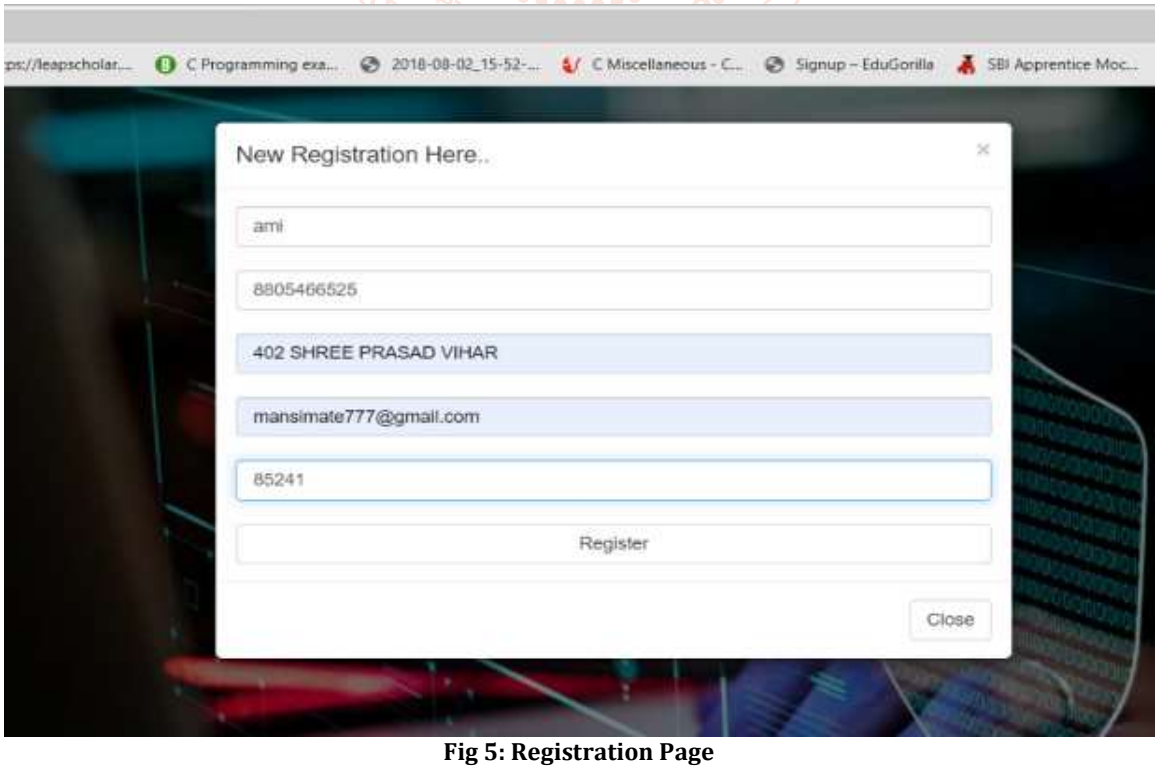

Fig 5: Registration Page

Here we have a sign up page for registration of user to the system.

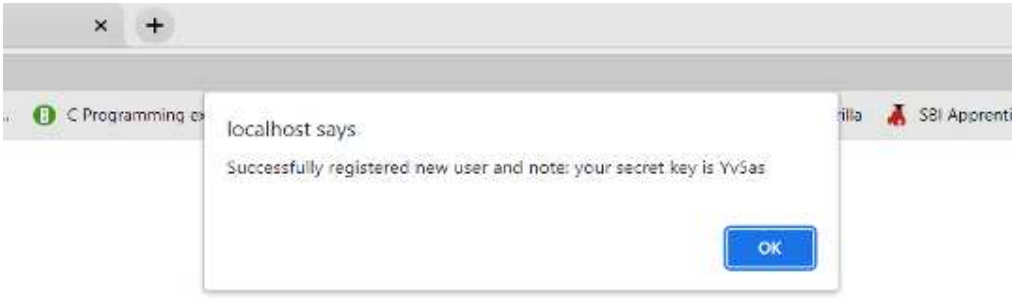

Fig 6: Successfully registered

After successful registration system will generate an alpha-numerical unique key for that system using mersenne twister algorithm which is applicable to that system only.

Fig 7: Checking System Authenticity

Now the system will then read the PC id and check authenticity of that system.

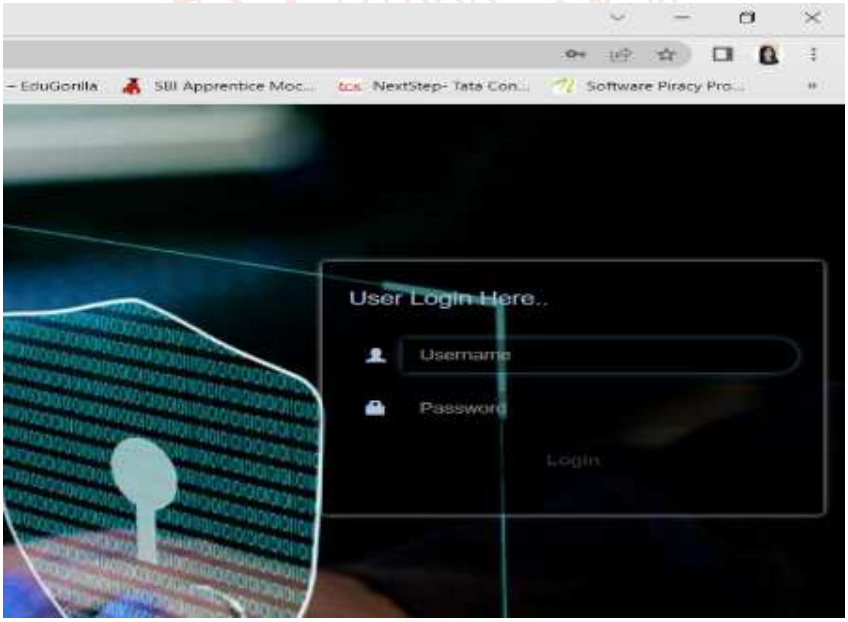

Fig 8: User login Page

After that the user have to log in all the credentials again for successful registration process.

Fig 9: Unique Key

And lastly user have to enter the unique key which is provided by the system to enter into the software.

VI. CONCLUSION

In this research, we embarked on the journey of developing a robust and adaptive system to protect software from piracy. Recognizing the significant financial and ethical impact of unauthorized software distribution, our goal was to design a solution that not only deters piracy but also respects the user experience and remains practical for implementation.

We introduced a multi-layered protection architecture that includes license validation, hardware-based authentication, and optional blockchain-based license verification. By evaluating the system against common piracy techniques, we demonstrated that our approach provides a strong line of defense while remaining efficient and scalable.

This project reflects our commitment to both academic exploration and real-world application. Through theoretical study, technical design, and empirical testing, we were able to validate the effectiveness of the system. Moving forward, we believe this work lays a solid foundation for future innovations in digital rights management and secure software distribution.

Software piracy is an ever-increasing problem of the modern-day software industry. Owing to the evolution in software development and the Internet, software piracy has become a main concern for many software companies. Software companies are confronted with extremely high losses due to the piracy of software. Pirates gain a lot of money by doing business with pirated software, and they do not think what they are doing is a crime. General end-users and the community of the software are not well aware of this serious crime. Even most of the time, end-users and consumers think that it is none of their concern and not an important issue for them to worry about. If an organization is using pirated software, there is a risk of failure of the software, and it might put the organization into a big loss of risk.

Ultimately, this research is a step toward empowering developers and organizations to safeguard their intellectual property while promoting responsible and legal software use across the digital ecosystem.

VII. References

[1] T. T. Moores and J. Dhaliwal, "A reversed context analysis of software piracy issues in Singapore," Information & Management, vol. 41, no. 8, pp. 1037–1042, 2004.View at: Publisher Site | Google Scholar

[2] L. L. Gan and H. C. Koh, "An empirical study of software piracy among tertiary institutions in Singapore," *Information & Management*, vol. 43, no. 5, pp. 640–649, 2006. View at: Publisher Site | Google Scholar

[3] Mishra, I. Akman, and A. Yazici, "Software piracy among IT professionals in organizations," *International Journal of Information Management*, vol. 26, no. 5, pp. 401–413, 2006.View at: Publisher Site | Google Scholar

[4] Curtis, "Software piracy and copyright protection," in *Proceedings of Wescon/94: Idea/Microelectronics*, pp. 199–203, New York, NY, USA, September 1994.View at: Google Scholar

[5] R. C. Rife, "Software piracy," in *Proceedings of Northcon/94 Conference Record*, pp. 364–366, Seattle, WA, USA, October 1994.View at: Google Scholar. Nazir,

[6] S. Shahzad, and L. S. Riza, "Birthmark-based software classification using rough sets," *Arabian Journal for Science and Engineering*, vol. 42, pp.

[7] B. Depken and C. Simmons, "Digital piracy, moral beliefs, and illegal downloading of copyrighted materials," *Journal of Business Ethics*, vol. 100, no. 3, pp. 471–481, 2011.

[8] A. L. Thong and C. Y. Yap, "Software piracy in the Asia-Pacific: Causes and remedies," *Journal of Business Ethics*, vol. 55, no. 3, pp. 185–194, 2004.

[9] N. Kshetri, "The economics of click fraud," *IEEE Security & Privacy*, vol. 6, no. 3, pp. 45–53, 2008.