Software Piracy Protection System: *Preventing Unauthorized Use with Unique License Key Authentication*

Amit[1], Anuj Mishra[2], Ankit Soni[3], Aman Chouhan[4]
Student, (Department of Computer Science & Engineering) Indore Institute of Science & Technology, Indore (M.P.)

Abstract- **Software piracy remains a significant challenge, affecting the profitability and integrity of the software industry. Unauthorized duplication and distribution of software lead to financial losses and security vulnerabilities for both developers and users. The software industry faces a serious problem with software piracy. This paper will show how a "software piracy protection system" serves primarily to prevent system piracy. This research presents a Software Piracy Protection System (SPPS) that prevents illegal usage through unique license key-based authentication. The system binds each software installation to the user's MAC address, ensuring that the license key is valid only for a single authorized device. The purpose of this study is to examine the value of software piracy prevention and the application of unique license key creation and verification as a means of resolving this problem. The proposed method integrates cryptographic hash functions, encryption algorithms, and a secure server-side verification process to enhance software security. The system operates with an authentication model that detects unauthorized key duplication and prevents offline misuse by validating the user's hardware identity upon every startup. Experimental results demonstrate the effectiveness of the approach in mitigating software piracy risks, ensuring compliance, and strengthening digital asset protection. Future improvements, such as AI-driven piracy detection and blockchain-based authentication, could further enhance the security framework.**

Index Terms: **Anti-Piracy Measures, Cryptographic Security, License Key Authentication, MAC-Based Registration, Privacy, Software Piracy.**

I. INTRODUCTION

Software piracy is the term for using, copying, or distributing software without permission or permission to do so. The majority of software packages are only licensed for use by one user or on one computer. Software piracy is the act of copying software to numerous computers or sharing it without multiple licenses with friends, coworkers, or customers. Software piracy has become a global concern, leading to significant revenue losses, security vulnerabilities, and legal disputes. Unauthorized software distribution and usage hinder technological advancements, affecting both individual developers and large-scale enterprises. Despite existing anti-piracy measures such as license key validation, digital rights management (DRM), and online authentication, software remains vulnerable to key generators, license tampering, and unauthorized modifications.

To address these challenges, this research presents a Software Piracy Protection System (SPPS) that integrates unique license key-based authentication linked to a user's MAC address. By binding software licenses to specific hardware devices, SPPS ensures that an authenticated copy cannot be used illegally on multiple systems. The system leverages cryptographic hash functions, encryption protocols, and periodic integrity checks to reinforce security. Additionally, server-side verification and offline mode validation prevent unauthorized usage and reduce piracy risks.

II. PROBLEM STATEMENT

Software piracy remains a persistent challenge, affecting both small and large-scale enterprises. Many users bypass legal licensing, undermining revenue models and security protocols. Attackers exploit vulnerabilities, distributing cracked software versions illegally. This study proposes a strategic framework for securing software distribution and authentication. Today, there is a lot of software piracy. Attackers will take anything they can find from movies, including an operating system, and make it usable. Users must have a fundamental understanding of all these threats as well as the software we employ.

These all are the reasons or purpose to develop a protection system for software piracy:
Risks of using pirated software:

- Using pirated software might be cheaper than buying original software, but you should be aware of the dangers that await a software pirate.
- As an unauthorized user, you will not receive any updates or customer support from the software manufacturer.
- You will face an increased risk of the unlicensed software malfunctioning or crashing.
- You will put your online security at risk because illegal and counterfeit software might infect your device with viruses, malware, or adware.
- Visiting pirating websites is a danger in itself — they contain malicious ads, let alone infected files.
- You may face legal consequences due to copyright violation, including financial penalties.
- Being familiar with the risks is step one, while step two is taking action to avoid software piracy altogether.

III. SOLUTION APPROACH

The following solution approach is going to use for this problem:
Software Piracy Protection System Using Genuine License Key:
To prevent software piracy by ensuring each installation of the software has a unique license key tied to the specific hardware (MAC address) of the user's device.
Key Components:

1. User's System
2. Software Application
3. Authentication Program
4. Hash Function (e.g., MD5, SHA-256)
5. Server and Database.

Workflow & Security Mechanisms:
Online Registration: Users have to register themselves into the system by entering it's details like Name, DOB, Mobile Number, Gmail Address, and Unique User Id and Password.
PC ID Reader: The software reads the user's PC MAC ID.
Key Generation: The user may now request for generating license key. They need to send the unique user ID. The key is generated by applying encryption to the entered unique user ID.
Data Matching and Authentication: The software applies the encryption to the user ID, MAC Address and sends the encrypted key. The software generates a key by encryption and then matches the key provided by the user with the generated key. If the keys match, the software works as a full version; otherwise, it is locked down.
Not Supported on Other PCs: Since the same key, if applied to software on another PC, will fail because the MAC ID and User ID on the other PC will be different, and thus the key for that PC will be different.

IV. METHODOLOGY & WORKING

Methodology and Working of Piracy Protection System:

Methodology Overview:
The methodology follows a multi-layered authentication mechanism leveraging unique user credentials and device-specific identification to prevent unauthorized software use. The system integrates encryption-based licensing, ensuring that software licenses remain bound to a specific user and device.

Working Process:

1. *User Registration & Credential Storage:*
    - Users register by providing personal details including Name, Date of Birth, Mobile Number, Gmail Address, and Unique User ID.
    - A secure database stores user credentials and ensures encryption to prevent unauthorized access.
2. *Device Identification via PC MAC ID:*
    - The software detects the PC's unique MAC ID upon installation.
    - This MAC ID acts as a secondary authentication factor ensuring licenses remain linked to specific hardware.
3. *License Key Generation:*
    - Once registered, users request a license key by sending their Unique User ID.

- The system encrypts this User ID to generate a unique license key using advanced encryption algorithms.
- This ensures that each user gets a personalized, tamper-proof key.

4. *Data Matching & Authentication Process:*
   - The generated encrypted key is matched against the user-provided key for verification.
   - If the two keys match, authentication is successful.
5. *Software Activation & Locking Mechanism:*
   - Upon successful authentication, the software unlocks full functionality for the registered user.
   - If authentication fails, the software remains in a restricted mode, preventing piracy.
6. *Restriction on Unauthorized Device Usage:*
   - The license key is device-specific; attempting to use the same key on another device will fail due to mismatched MAC ID and User ID.

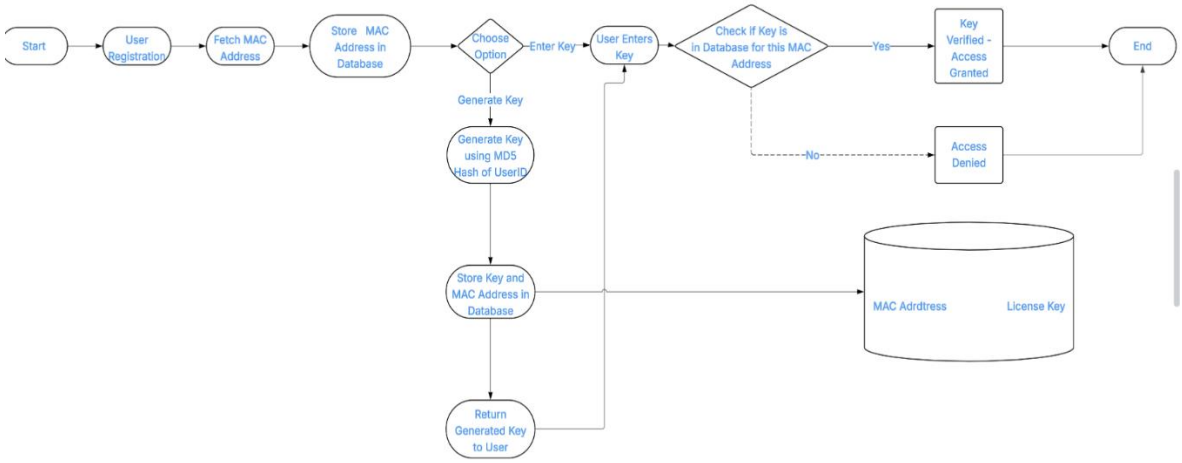

Fig. 1: DFD Diagram of Software Piracy protection System Using Unique License Key

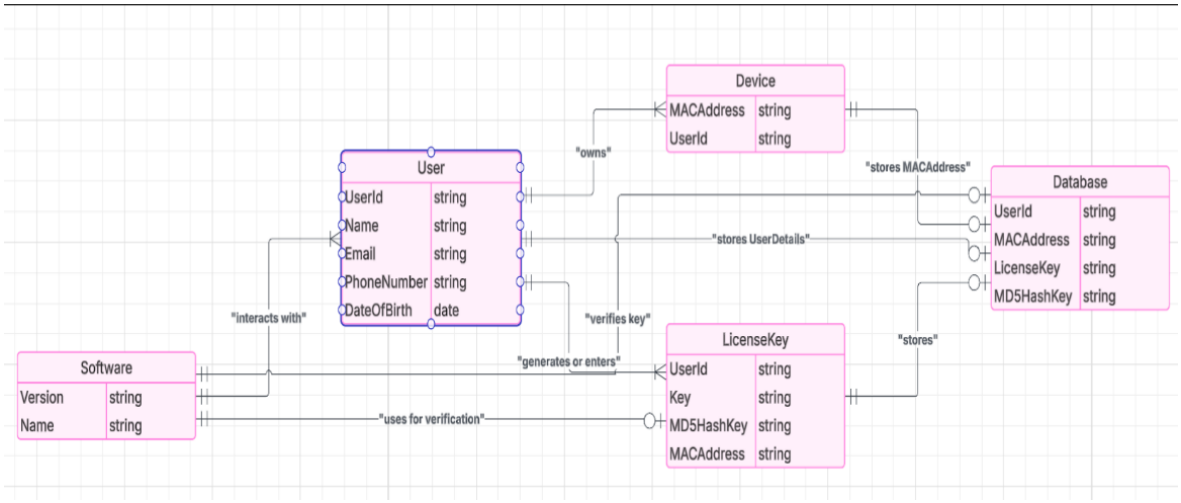

Fig. 2: ER Diagram of Software Piracy protection System Using Unique License Key

V. REQUIREMENTS

a) Functional Requirements:

The Software Piracy Protection System(SPPS) must have the following functional requirements:

Software Requirements:

These are the software requirements:

- Operating System: Windows 7, 8, 9, 10,11 .
- Language: HTML , CSS , JavaScript ,JAVA, SQL.
- Database: MySQL Server (back end)

Hardware Requirements:

- Processor: Intel core i3 or above for a stable experience and fast retrieval of data.

- Hard Disk: recommended 40GB or more.
- RAM: recommended 2 GB or more for fast reading and writing capabilities which will result in better performance time.

b) Non-Functional Requirements:

Usability: The UI should be simple, interactive, and support secure login while handling errors effectively.

Security: A secured database, multi-user authentication, and unique license keys should prevent duplication.

Performance: The system must handle a high number of license keys efficiently, with response times under 5 seconds.

Scalability: Supports growing user demands, ensuring horizontal, database, and testing scalability.

Backup & Recovery: Securely stored backups must cover all critical data, ensuring regulatory compliance and disaster recovery.

Error Handling: Prevent data loss and downtime by handling both expected and unexpected errors effectively.

Monitoring & Logging: Track system activity, license key usage, and maintain detailed logs for security compliance.

VI. NEW APPROACH -- QUANTUM-BASED SOFTWARE PIRACY PROTECTION (QSPP)

1. *Concept overview:*

Traditional cryptographic licensing systems (AES, SHA-256, MD5) rely on mathematical complexity to protect software. However, advancements in quantum computing could break current encryption through quantum algorithms like Shor's algorithm.

To counteract this, Quantum-Based Software Piracy Protection (QSPP) introduces a Quantum Fingerprint Authentication System that makes piracy mathematically impossible through quantum randomness and entanglement.

2. How QSPP Works:

QSPP ensures each software copy is uniquely linked to a user's hardware using a Quantum Fingerprint, which is generated through Quantum Random Number Generators (QRNGs).

Step-by-Step Process:

Step 1: Quantum Fingerprint Generation

- A Quantum Random Number Generator (QRNG) produces a unique fingerprint based on the user's hardware properties (CPU architecture, MAC address, memory configuration).
- This fingerprint is constantly changing, making duplication impossible.

Step 2: Quantum Secure Licensing

- When a user requests a license key, a Quantum Encrypted License Code (QELC) is created.
- The QELC is entangled with the user's Quantum Fingerprint, ensuring only their system can unlock the software.
- Even if an attacker copies the license key, it will fail on another machine because the Quantum Fingerprint mismatch will invalidate it.

Step 3: Quantum Entanglement-Based Verification

- Every time the software runs, it retrieves the user's Quantum Fingerprint and compares it with the fingerprint stored in the secure database.
  - If there is even a microscopic difference, the license key becomes invalid, preventing unauthorized cloning.
  - This ensures tamper-proof licensing, as traditional methods like key generators or software cracks cannot recreate quantum fingerprint properties.

Step 4: Protection Against Quantum Attacks

  - The system leverages Post-Quantum Cryptography (PQC) algorithms to ensure resistance to future quantum decryption methods.
  - Attackers using quantum computers cannot break the security since the encryption keys are tied to quantum-generated randomness.

3. Why This Technology is Unbreakable:

Impossible to Clone : Unlike traditional encryption, a Quantum Fingerprint cannot be duplicated, as it exists in a constantly evolving quantum state.

Immune to Cracking: Attackers cannot reverse-engineer the licensing process because of quantum uncertainty principles.

Mathematically Secure: Even if quantum computers become powerful enough to break traditional encryption, Quantum Entanglement ensures license validation beyond computational prediction.

Tamper-Proof Licensing: The fingerprint changes dynamically, meaning that even if an attacker steals the original file, the authentication system detects differences instantly.

4. Future Extensions:

Blockchain-Based Quantum Licensing: Store quantum fingerprints in a decentralized ledger, making piracy detection globally available.

AI-Driven Security Monitoring: Use AI pattern recognition to track abnormal fingerprint behavior and flag potential threats before they occur.
Quantum Cloud Licensing: Store quantum license validation data on ultra-secure quantum cloud servers to prevent tampering at any level.
Final Thoughts: QSPP is a game-changing, futuristic approach to stopping software piracy using Quantum Cryptography and Quantum Fingerprinting. With quantum entanglement-based authentication, license keys become completely unbreakable, ensuring software security for decades against even the most advanced threats.

VII. CONCLUSION

Software piracy continues to threaten the integrity and financial stability of the software industry.
This research tackles piracy with a two-pronged approach:
a practical, secure solution using unique license key authentication based on mac address and an innovative, future-proof quantum security model.
The Unique License Key Authentication System:
bound to the user's hardware (mac address), ensures that every installation remains protected against unauthorized duplication. Through cryptographic encryption, server-side validation, and hardware-specific licensing, this approach effectively prevents software piracy today.
Looking beyond conventional security methods:
Quantum-Based Software Piracy Protection (QSPP): introduces quantum fingerprinting and entanglement-based verification to completely eliminate piracy in the future. By leveraging quantum random number generators (qrngs), quantum-secured licensing, and post-quantum cryptographic validation, this model ensures tamper-proof, mathematically unbreakable software authentication.
Together, these solutions create a hybrid security framework that strengthens today's anti-piracy measures while preparing for the quantum era. The combination of traditional encryption and quantum cryptography ensures a resilient, adaptive, and forward-thinking approach to software licensing security.
Future Scope: As quantum computing evolves, integrating AI-driven piracy detection, decentralized blockchain authentication, and real-time quantum security monitoring can completely redefine software protection standards. These advancements will help transition the industry from reactive piracy prevention to proactive, predictive security, ensuring a piracy-free digital future.
Final Thought: This research not only solves piracy challenges today but also lays the foundation for next-generation cybersecurity, ensuring software protection that stands against both current and future threats.

Acknowledgment: We sincerely thank Mr. Amit Kanungo (Assistant Professor, Department of Computer Science & Engineering, Indore Institute of Science & Technology, Indore) for his invaluable guidance, continuous support, and insightful advice throughout this research.

REFERENCES:

License Key Authentication & Cryptographic Security
[1] Software Piracy: An Analysis of Protection Strategies https://pubsonline.informs.org/doi/10.1287/mnsc.37.2.125
[2] *Protection Against Software Piracy*: A Study Of Technology Adoption For Intellectual Property Rights https://doi.org/10.1080/10438590000000002
[3] A Robust Approach to Prevent Software Piracy https://ieeexplore.ieee.org/document/6199075
Quantum-Based Software Piracy Protection (QSPP)
1. Issues of Privacy and Security in Smart Cities (Relevant to Quantum Security) https://ieeexplore.ieee.org/document/6524468
2. Privacy by Evidence: A Methodology for Privacy-Friendly Software Applications https://doi.org/10.1016/j.ins.2020.06.016
3. Software Security, Privacy, and Dependability: Metrics and Measurement https://ieeexplore.ieee.org/document/7524728
4. Privacy-Oriented Software Development https://link.springer.com/chapter/10.1007/978-3-030-29238-6_2