MATHEMATICAL MODELING AND COMPUTING, Vol. 9, No. 2, pp. 326–332 (2022)

Modeling Computing Mathematical

Architecture of the License Software Manager using Blockchain technology

Liubinskyi B. B., Demediuk M. S., Topylko P. I.

Lviv Polytechnic National University,
12 S. Bandera Str., 79013, Lviv, Ukraine

(Received 28 September 2020; Revised 17 March 2021; Accepted 30 June 2021)

The paper considers Blockchain technology and the possibility of its use to prevent software piracy. There is proposed an approach using Blockchain technology. An architecture of the License Software Manager has been developed using Blockchain technology. The software code is available at `https://github.com/maksym-demediuk-clicense/clicense`.

Keywords: *blockchain, software product license, software product piracy.*

2010 MSC: 94-04, 94A05, 94A60 **DOI:** 10.23939/mmc2022.02.326

1. Problem overview and the Blockchain technology

The "*software piracy*" term refers to the copying or use of computer software that violates aspects of its licensing. The problem first arose with home computers [1, 2], but also with the use of a personal computer in business [3] and in education [4, 5]. Although software piracy can cause problems for the software vendor, Conner [6], Katz [7], and Shy and Thisse [8] argue that removing copy protection in certain circumstances may be beneficial to the vendor. Darmon [9] claims that the need for copy protection is reduced for customers who need support services from the software provider. Copyright enforcement requires significant costs for software vendors [10, 11]. Today, the level of piracy in developing countries is still consistently high at 60%. Consumers in these countries cannot legally purchase software [10–12]. The Business Software Alliance (BSA) [13] defines software piracy as the unauthorized copying or distribution of copyrighted software, including downloading, sharing, selling, or installing multiple copies of licensed software. The BSA identifies five types of software piracy, namely:

- end–user piracy, which involves the illegal use of software for their own needs;
- Client–Server architecture, which assumes an usage of an application with a centralized license for many network users;
- Internet–piracy using the Internet to obtain software;
- creating copies of programs from the hard disk, which involves the installation of unlicensed software products. It used to be a common practice to sell personal computers;
- software forgery, which involves direct copies of software and duplication of licenses.

In 2013, the BSA claimed that 43% of software installed on home computers was improperly licensed. This amount of illegal licensing amounted to 62.7 billion US dollars. It was assumed that cloud subscription models would be a means of reducing license violations. However, since 52% of user credentials are distributed among service users, this may not provide any significant impact on the level of piracy [10]. Software piracy has been found to be a significant problem in terms of industry, with losses of about $ 132 billion per year [14]. For the copyright protection of the software purposes, printed key codes can be used to activate the software (for example, on the software package), check the license on the Internet [15] and various hardware devices [16]. Smaller publishers can use activation keys, while larger publishers, such as Microsoft and Adobe, use proprietary software license validation (SLV) services that use the Internet. SLV is complicated by technological and economic changes.

The means by which several parties (software owners, multiple levels of distributors, and customers) buy software are becoming more sophisticated, new models are emerging [17], and this influences on usage patterns and applications [18]. The task of verifying the use of the software is becoming increasingly complex and expensive for all parties. Litchfield and Herbert proposed an approach to software licensing using Blockchain technology [19].

For over 30 years, personal desktops have had a variety of copyrights. Common methods include checking software, which should be inexpensive, compatible with the other systems, easy to implement [1], and provide ways to prevent illegal copying [16]. The value of copyright protection is balanced by the cost of preventing piracy, where the main methods of authorizing licensed programs are copy protection, software and hardware keys. Because of early desktops lacked capacity and were expensive, more sophisticated methods that could provide the ability to use encryption to protect licensed software were impossible. Thus, the SLV form was used instead, which was easy to distribute and deploy. When installation programs were distributed on media such as floppy disks (5.25-inch or 3.5-inch), copy protection was limited to changing disk sectors to prevent copying, and license keys were provided along with the software. With a number of tools, experienced users could easily circumvent the measures to prevent copying the disk, and copy the license key data. The Internet now provides different opportunities for other methods of testing software. Online software publishers use online authorization and verification solutions based on the purchase of licenses and license keys that issue unique keys to the customer and manage current authorization and verification of the license key. As Internet bandwidth continues to grow, the way digitally distributed software is licensed has changed. As of now, instead of making it difficult to copy software, full-featured trial software is commonplace, and the task has shifted to authorization and license verification. Some vendors, such as Microsoft, use a wide range of software licensing models including online licensing portals. This shifts the task of license management to the licensee, who in the corporate environment must provide evidence of regular and complete audit of software use. However, online license verification methods can be bypassed. For example, redirecting a domain name system (DNS) to fake authentication servers, key generators (keygens) that mimic the software vendor's own authentication system, reverse engineer and remove SLV mechanisms, or release pre-authenticated software [20]. Online application stores and vendors' trading platforms provide software vendors with an enhanced and secure way to distribute and license software through secure portals that require end-users to register, authenticate, and authorize to purchase a software license. But once the software is downloaded, protection is limited because the software is vulnerable to most pirated methods. This is most evident in the Android ecosystem, where there are no controls on the legitimacy and functionality of the downloaded application [22]. In the Apple iOS ecosystem, Apple developers must demonstrate that their product complies with Apple's policies. However, the potential for piracy is expanding on incompatible iOS devices, and recent information suggests that an unexpectedly large number of iOS applications have been counterfeited [23, 24]. Corporate licensing on mobile devices is challenging because many software vendors do not have the same level of license management as desktop.

It has also been established that the conflict between publishers and pirates continues, and attackers continue to develop tools to overcome protection against software piracy. To overcome this problem, it is planned to create a modern method of verifying software licenses to verify the user's rights to the software while using it. Software license authorization has become the main software tool to prevent software piracy. However, such tools can be expensive and complex and have limited effectiveness, as existing methods of authorizing software licenses can be relatively easily overcome. The main task of this work is an overview of Blockchain technology, analysis of the possibility of using a distributed database to combat piracy with the use of Blockchain technology. Software piracy has always been one of the most dangerous threats to the software industry, especially after the advent of the Internet and the availability of many software analysis tools. Blockchain is a distributed storage technology developed by anonymous Satoshi Nakamoto (2008) and first introduced with Bitcoin, the first cryptocurrency. Cryptocurrencies are a new form of digital currency. They are distributed

electronic currency systems being the first technology to successfully overcome the requirement for a centralized system for verifying transactions [25]. The crypto-currency and blockchain architectures provide several mixed functions that form the cryptocurrency ecosystem, including cryptographic verification of all transactions, decentralized money, bitcoin mining, and transaction processing functions. All of them are stored on publicly distributed systems within a quasi-anonymous structure. Cryptocurrencies use public-key cryptography to verify transactions between all participants, as well as digital signatures to ensure the integrity of transactions and prevent rejection. The crypto-mechanisms used by cryptocurrencies ensure the serious confidentiality, integrity of data, and non-deviating services used by businesses, government, and military organizations around the world. In a cryptocurrency ecosystem, a public key can be considered a participant's account number, while a private key represents a participant's credentials. All participants have digital wallets that are used to store private keys, as well as digital signatures representing cryptocurrency rights (bitcoins) owned by participants. Wallets can be stored privately or online on websites or in exchangers, depending on the requirements of the participant. To be resilient to threats and attacks, as well as a stable and liquid currency, cryptocurrencies have not yet proven their reliability both technologically and economically. However, as developers seek to use cryptocurrencies in more practical applications, the cryptocurrency architecture, i.e. blockchain, is the basis for developing new programs based on cryptocurrencies. Blockchain [26,27] (blockchain) is also called Distributed Ledger Technology (DLT) — a combination of components that includes: peer-to-peer (P2P) peer-to-peer networks, distributed data storage, cryptography (hashing and public-key encryption).

2. Use Blockchain to check software product licenses

An overview of the problem of software piracy shows that focusing on user identity is the key to preventing software piracy. It can be assumed that linking user identities to software rights is important to providing access to the software product. The disadvantage of user authorization is that others can share user credentials, which is not a sufficient deterrent to sharing credentials that provide access to the software. In addition, there is no protection for software that is vulnerable to pirates who attack the code structure directly to defeat the defense mechanisms. To achieve a cost-effective method to prevent software piracy, it has been suggested that the method of software license validation needs to be addressed in specific problem areas. The software license validation system must meet the following requirements:

- be global. This requirement is confirmed by the definition that the end user can be located anywhere in the world that provides universal access;
- be cost-effective. This requirement is supported by the understanding that smaller software vendors do not have the resources or capabilities to operate a software license validation system. Therefore, this system is mainly found on global providers, or through intermediaries such as Steam. Therefore, software vendors need a system with minimal overhead;
- uniquely associate user identification with software rights. The end-user must be associated with the software he is authorized to run and must not be able to distribute software that has been properly purchased;
- provide anonymity and data confidentiality to end-users. this requirement is established because the blockchain is a public repository. Transactions must be anonymous to maintain the privacy of end–users, and data must be encrypted to maintain the confidentiality of information.

The main purpose of this work is to develop a system that demonstrates that blockchain-based technologies can be used to create a method for validating software licenses. This method turns to account user identification, combined with explicitly permitted application rights and software verification, to achieve a viable method of preventing software piracy.

3. License Manager General scheme

It is proposed to use a network based on the Ethereum blockchain to implement the license verification system. Because this network is based on smart contracts, it is possible to call user code to work with software licenses. The general scheme is shown in Fig. 1.

The system consists of three parts:

- **Client** — a library that interacts with a third-party application and is responsible for licensing on the user's side. Written in C++. Additionally, a wrapper library was created in C++/CLI to enable the use of the client part in the .NET environment.
- **Server** — a control panel that allows the author of a third-party application to manage user license data.
- **Blockchain network** — a distributed database in which user license data is stored. The Ethereum platform based on smart contracts, was used as a blockchain. With smart contracts, the control panel administrator has the ability to manage user data (for example, create a new user license), and the client has the ability to connect to this database and verify the license.

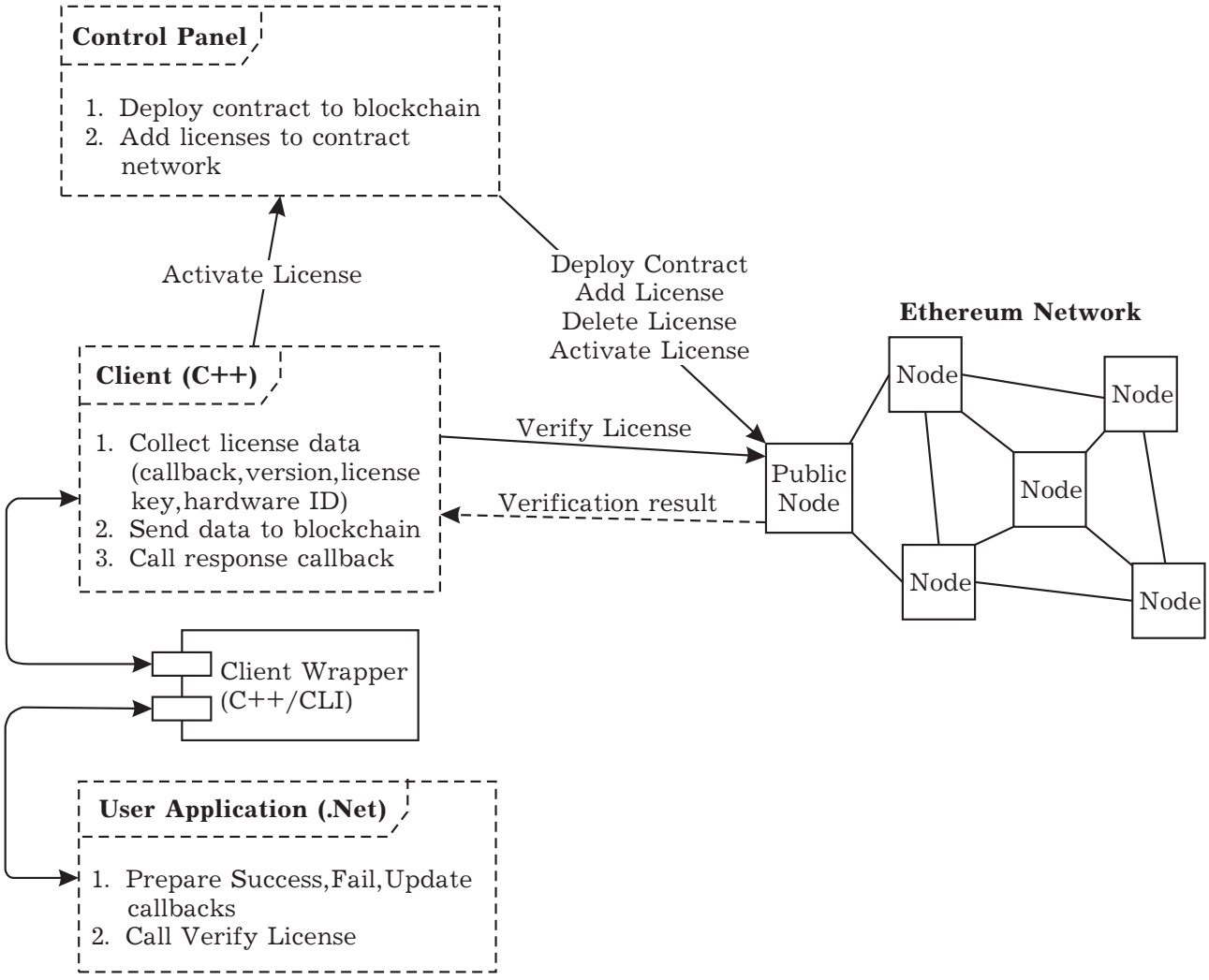

Fig. 1. License Manager General scheme.

4. The algorithm for checking license data

- The user program passes certain data (product name, version, pointers to the resulting Success, Fail, Update functions, which are called depending on the test result) to the CLicense.dll client library. If the program runs in a .NET environment, the data is transferred to the CLicenseCLI.dll wrapper library.
- The client module CLicense reads from the Windows registry private and public keys, the Ethereum network with which the program works, the address of the public network node. Collects information about computer devices (combination of CPU, RAM, HDD, network adapter, etc.) and generates HardwareID (HWID) based on this data. A request is then made to call the smart contract function to verify the encrypted data. The request is sent to a public node that is synchronized with other Ethereum nodes.
- The node receives a request. Because the license check does not require changes to the network (data is read from the distributed database without changes), the node processes user data instantly without the help of other nodes. The virtual machine executes the smart contract code, where the license key is searched and the iron identifier (HWID) is checked. Depending on the result of the check, the contract returns one of the resulting functions, such as Update, if the version of the user program does not match that registered in the blockchain network.
- The CLicense library receives a response from the Ethereum network node, decodes the data, and calls the received function.
- The control panel allows administrators to add software products, create and delete licenses, view blockchain network transactions.

You must activate the license the first time you run it. To verify the license during subsequent launches, the smart contract must know the user's iron ID. To avoid a situation where an attacker tries to activate a key (known to everyone because the data in the blockchain is public) with his data, the smart contract has a restriction — only the wallet from which the contract was created can be activated (control panel administrator). The user sends an activation request to the control panel on the first launch, handing over the public (the one stored on the blockchain) and private (receiving by e-mail) keys. A distributed Ethereum database based on smart contracts was chosen as an implementation of blockchain technology. When run on a blockchain, a smart contract becomes like a stand-alone computer program that is automatically called when certain conditions are met. On a blockchain, smart contracts allow code to work accurately, without any possibility of downtime, censorship, fraud, or third-party interference. Such conditions are quite suitable for the task of verifying license data, as one of the threats in this area is hacker attacks on the databases of program authors. For a smart contract to work, you must first add it to one of the Ethereum networks (this process is called contract deployment). A smart contract was created in Solidity to verify the licenses. The contract stores license keys and activation status, hardware IDs for activated keys, contract version. The control panel is responsible for adding the contract to the network, requests to create and delete licenses on the contract side, activation of keys. The client module connects to the Ethereum network and sends the data for verification. The main advantage of using a blockchain is that there is no intermediary between the client and the database in the form of an authorization service. If the service will not work for technical reasons, or the owner decides to stop its work, the user will be able to continue to use the program, because the data is stored on a large number of network nodes. The control panel allows the administrator to manage their own products and their licenses. When adding a new software product, the administrator selects the product name, version, blockchain network, wallet with broadcasts. The server script automatically adds the contract to the Ethereum network and notifies the panel user of the transaction result. Additionally, the control panel administrator can:

- view product information: transactions, contract address;
- add and remove licenses;
- view license information: activation status, transactions;
- retrieve the latest blockchain data (some information may change without the control panel).

When adding a new license, the administrator has the option to enter the user's e-mail to which the file containing the license data will be sent and the link to the transaction on the Ethereum network. When a user's license is removed, the transaction address will be notified and sent.

5. Conclusions

In this paper, a decentralized online service based on a blockchain (Dapps) for the Ethereum platform is described. Consistency, decentralization, and a consensus mechanism are the main characteristics of Blockchain technology, which guarantee stable operation and successful execution of such operations as inserting, updating, and storing information about the software license and its verification. The service consists of a smart contract and services that provide a simple and clear way to work with the database.

[1] Maude T., Maude D. Hardware protection against software piracy. Communications of the ACM. **27** (9), 950–959 (1984).

[2] Mooers C. N. Preventing Software Piracy. Computer. **10** (3), 29–30 (1977).

[3] Suhler P. A., Bagherzadeh N., Malek M., Iscoe N. Software Authorization Systems. IEEE Software. **3** (5), 34–41 (1986).

[4] Im J. H., van Epps P. D. Software piracy and software security measures in business schools. Information & Management. **23** (4), 193–203 (1992).

[5] Taylor G. S., Shim J. P. A Comparative Examination of Attitudes Toward Software Piracy Among Business Professors and Executives. Human Relations. **46** (14), 419–433 (1993).

[6] Conner K. R. Software piracy: An analysis of protection strategies. Management Science. **37** (2), 125–139 (1991).

[7] Katz M. L. Systems competition and network effects. Journal of Economic Perspectives. **8** (2), 93–115 (1994).

[8] Shy O., Thisse J. F. A strategic approach to software protection. Journal of Economics & Management Strategy. **8** (2), 163–190 (1999).

[9] Darmon E., Rufini A., Torre D. Back to software "profitable piracy": The role of information diffusion. Economics Bulletin. **29** (2), 543–553 (2009).

[10] Business Software Alliance. The Compliance Gap; Technical Report; Business Software Alliance: Washington, DC, USA (2014).

[11] Traphagan M., Griffith A. Software Piracy and Global Competitiveness: Report on Global Software Piracy. International Review of Law, Computers & Technology. **12** (3), 431–451 (1998).

[12] Andrés A. R., Goel R. K. Does software piracy affect economic growth? Evidence across countries. Journal of Policy Modeling. **34** (2), 284–295 (2012).

[13] Business Software Alliance. `https://www.bsa.org/files/2019-02/2018_BSA_GSS_Report_en_.pdf`.

[14] Herbert J. Solving a global software piracy problem (2017).
`https://openrepository.aut.ac.nz/bitstream/handle/10292/11021/HerbertJ.pdf`.

[15] Peyravian M., Roginsky A., Zunic N. Methods for preventing unauthorized software distribution. Computers & Security. **22** (4), 316–321 (2003).

[16] Morgan M. J., Ruskell D. J. Software Piracy — The Problems. Industrial Management & Data Systems. **87** (3/4), 8–12 (1987).

[17] Sachan A., Emmanuel S., Kankanhalli M. Efficient license validation in MPML DRM architecture. DRM '09: Proceedings of the nineth ACM workshop on Digital rights management. 73–82 (2009).

[18] Liu Z., Roychoudhury A. Relating software validation to technology trends. International Journal on Software Tools for Technology Transfer. **14**, 631–638 (2012).

[19] Herbert J., Litchfield A. A Novel Method for Decentralised Peer-to-Peer Software License Validation Using Cryptocurrency Blockchain Technology. Proceedings of the 38th Australasian Computer Science Conference (ACSC 2015). 27–30 (2015).

[20] Sigi G. Exploring the supply of pirate software for mobile devices: An analysis of software types and piracy groups. Information Management & Computer Security. **18** (4), 204–225 (2010).

[21] Han K., Shon T. Software authority transition through multiple distributors. The Scientific World Journal. **2014**, 295789 (2014).

[22] Android marketplace hit by malware. Computer Fraud & Security. **2011** (3), 3 (2011).

[23] BBC Technology Desk. http://www.bbc.com/news/technology-34338362.

[24] Davies C. Virus Scanner – or Malware? Beware App Store Fakes. https://www.cnet.com/news/privacy/virus-scanners-filled-with-malware-are-flooding-app-stores/.

[25] Trade Finance Global – Overview of Blockchain.
https://www.tradefinanceglobal.com/blockchain/history-of-blockchain/.

[26] Nomura Research Institute. Survey on Blockchain Technologies and Related Services. FY2015 Report. 1–78, (2016).

[27] Leibowitz J. Bitcoin: A 21st Century Currency Explained By a Wall Street Veteran. Coindesk (2016).

Архітектура менеджера ліцензій програмного забезпечення з використанням технології блокчейн

Любінський Б. Б., Демедюк М. С., Топилко П. І.

Національний університет "Львівська політехніка",
вул. Бандери 12, 79013, Львів, Україна

У роботі розглянуто технологію блокчейн та можливість її застосування для запобігання піратства програмного забезпечення. Побудовано підхід, який базується на використанні технології блокчейн. Розроблено архітектуру менеджера ліцензій програмного забезпечення з використанням технології блокчейн. Код програмного забезпечення доступний за адресою:
https://github.com/maksym-demediuk-clicense/clicense.

Ключові слова: *блокчейн, ліцензія на програмний продукт, піратство програмного продукту.*