ISSN 2047-3338

Online Modules: Novel Model in Serial-Based Method of Software Copy Protection

M. GhanaatPisheh Sanaei[1], H. Zamani[2], B. Emami Abarghouei[3] and A. Ghadiri Hakimi[4]

***Abstract*—One of the most important concerns of software corporations is to protect their products against unauthorized copying. Since all the researchers proposed some copy protection models that all of them have weakness to protect against unauthorized copying. The goal of this study is to propose a new model of serial-based method with more security against illegal usage. This paper offered Online Module model in serial-based method that it spilt the software in two parts. Fist part run in client and second part run in web service, when the software needs to use the second part then the software connect to the web service and the server then checks the software license. If the license has been valid then run the second part. This model compares with online activation models in serial-based method and the results show the proposed model is more secure against unauthorized copying.**

***Index Terms*— Unauthorized Copying, Serial-Based Method, Online Modules Model and Software Protection**

I. INTRODUCTION

PROTECTION of software against cracking is one branch of software security. Usually, developer software use complex codes for enhancing security in order for crackers to have problems to crack the software. But they can do it just with spending more time. In this paper popular and strong algorithm about crack proofing and bring up an idea for enhance crack proofing are reviewed. [1].

Nowadays, peoples life is better with the technology as and life without technology is unbelievable. Also, technology has caused people to have more time for other works and can avoid of duplicating work. Day by day, more people need new technology. One of the most important parts of technology is software that it can be useful for people in all fields such as business, education, entertainment, health, communication and daily routine.

For creating software, we need to spend substantial amount of money on the software developer to make powerful application for attracting more customers and earn more money. Because of this they should spend more money, time and use expert groups of software engineering for preparing the software. Unfortunately, when the software enters to the markets for selling, crackers start to crack the software in order for other users to use it without paying money. Some users buy it but others use illegal software by the cracking the code. It makes a great loss to the software developer and because of this the developers do not have enough motivation to create costly software. At first the developers estimate as to how many of license will sell with what price, after that they spend money and time for creating the software. Imagine, the users cannot use cracked software, what happens? It's clear; the developers can sell more of their software so they can spend more money for preparing software and every day we will see powerful applications come to the markets that they can be more useful for us [2].

In today's world, piracy accounts for $35 billion in lost revenues to software companies. Software developers usually have a strategy for protecting software against illegal usage of their applications. The developers often sell a license with software for activation the application. When a user enters the license code, application investigates the license validity and if the entered serial number has been correct then the application can be activated and user can use it. Crackers can open and change software codes in order to misuse the application as illegal access. A general trend that appears is software from a large company is usually more secure while smaller start-up companies seem to lack the necessary means to protect their software. But almost all models of software copy protection have some breaches to bypass licensing because crackers allow to all machine code of the software [3].

The main offer of this study is to propose the new model which is capable to improve the weakness of software copy protection models. The model doesn't allow the crackers to

[1] Mojtaba GhanaatPisheh Sanaei is the M.S Student at the Faculty of Computer Science and Information System, Universiti Teknologi Malaysia (UTM), Johor 81310, Malaysia (corresponding author phone: +60142704644; e-mail: ghanaatpisheh.m@gmail.com).

[2] Hadi Zamani, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, (e-mail: hadii.zamani@gmail.com).

[3] Babak Emami Abarghouei, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, (e-mail: emami.babak@gmail.com).

[4] Ahmad Ghadiri Hakimi has M.S degree from Universiti Teknologi Malaysia, (e-mail: ahmad.ghadiri@gmail.com).

have all the machine code of software. Another goal of this study is to research on some models' trends of software copy protections and to understand the vulnerabilities of their methods.

II. RELATED WORK

This phase discusses some types of software license. The next part reviews software copy protection methods and models. Also, compares them in the table and reviews the advantages and disadvantages. In addition, describe Symmetric-key and Asymmetric-key in cryptography that is used in software copy protection methods.

A. Type of Software Licensing

Today software developers provide a different software license and the type of software licensing are expressed below.

1) Open Source / Free Software: This is a special software licensing model in which the developer's goal is that users and everybody can access the source code and able to change and develop it. Because of this Open Source/ Free Software do not need any protection of their code and license, since it is meant to be free.

2) *Freeware:* It is similar to Open Source/ Free Software, but the source code is not open. Therefore it is normally, users are not able to change code. Same as the Free Software, this does not need to be copy-protected.

3) *Shareware:* In this type of software licensing, usually for free to all but with some limitations:

- Some software from this category can be used for free for limited period only. This type of software licensing is called free-trial. To remove restrictions, users must purchase a license.
- The another type of shareware contains the full functionalities, it's meant for full version usage and you should upgrade the software to full version and after that you can access the full version of software.

4) Node Locked License: This is a software license model which bounds software to a specific device. In this idea users must pay for every unit where the software is to be used: one license, one unit. Such examples include Copy-Protected Games which you can (normally) only play with their original CD/DVD, (High-Cost) Dongle-protected software, and so on.

5) Floating License: Is an alternative to the Node Locked Licensing, since it allows for a central management of different licenses for more machines. The idea is that you can buy a single license for software, which can be installed in a (limited) number of computers and can be used at the same time. A central server in the network is normally used to manage this type of licensing.

B. Copy Protection Methods

In this phase three types of copy protection methods are explained on some type of serial-based models.

1) *Serial-Based Protections:* This method is the most common method that software companies use it for protecting software copyright. It is simple to use in the software and it does not need special devices.

Using product serial number is one of the most common ways to verify the authenticity of legitimate users. The concept is simple: the author provides legitimate users with a serial, which is then checked by the program using a secret validation algorithm [4].

The boom of online available applications has boosted the popularity of the serial number protections enormously. To both software authors and end-users the scheme offers high flexibility and is relatively user-friendly. Smaller developers especially benefit from these features: the scheme allows the end-user to download the program free of charge. The user can try the program and, if convinced of the program's quality, buy it by an act as simple as an online registration procedure. The author has the opportunity to market his program at low distribution costs, because no physical media is required. Most of the time, the scheme is implemented by using certain restrictions on the freely available software in order to encourage registration. These include: time limits, crippled features, advertising and nag screens (e.g., A message being displayed every time the program starts) [5].

a) Non Parameter-Based Verification

The scheme, however, is not exclusively used for online distributions. In fact, it is originally used in over-the-counter software. During installation of the software, the installer asks the user to insert the serial, if it is invalid the installation process terminates. Usually such a serial is printed on something bundled with the software. In that case the key itself has a specific structure that allows a built-in key verification algorithm to decide on the user's legitimacy. In this model at first developer generate a key and send it to the client and user use it for activation software that it shows in Figure 1.

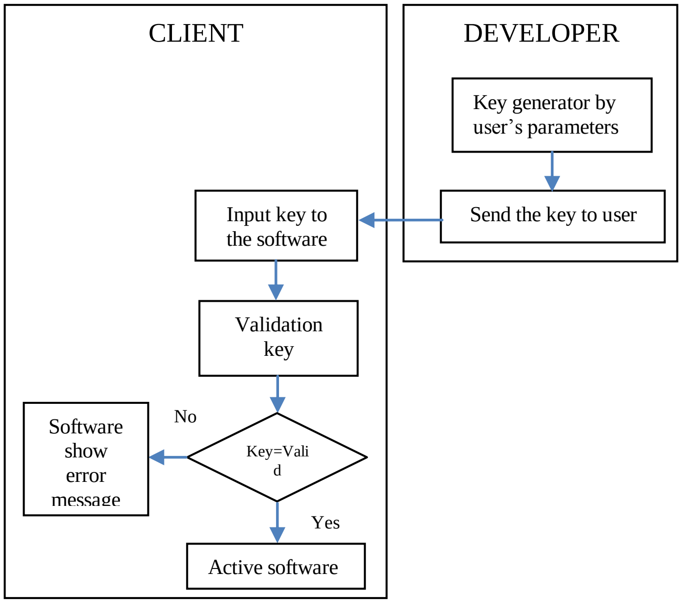
CLIENT
DEVELOPER
Key generator by user's parameters
Send the key to user
Input key to the software
Validation key
No
Software show error message
Key=Vali d
Yes
Active software

Fig. 1: Non parameter-based verification scheme

b) Parameter-Based Verification

Applications that can be registered online, the serial can be of a specific structure and can use above described scheme or it can be implemented as a user parameter-based verification scheme. In the latter case the serial is dependent on some of the user's parameters, like for instance his name. To register an application, the user then contacts the author by sending him for his name and the author provides the user with a key, created on the basis of the user's parameters. This serial was generated using the vendor's private key-generating algorithm; flowing process shown in Figure 2.

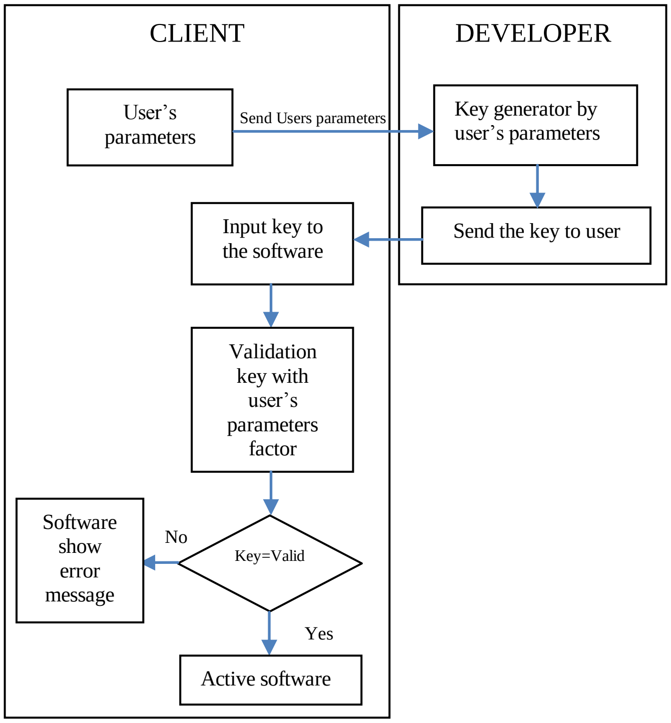
CLIENT
DEVELOPER
User's parameters
Send Users parameters
Key generator by user's parameters
Send the key to user
Input key to the software
Validation key with user's parameters factor
Software show error message
No
Key=Valid
Yes
Active software

Fig. 2: User parameter-based verification scheme

When the user enters his parameters and the key in the software's registration box, the program calculates the key by running the user's parameters through the built-in key generator and then compares the entered key with the one calculated in the background. When these two values match, the registration is successful.

It should be mentioned already that this protection, although flexible and user-friendly, has an inherent security risk, since the verification process includes generating the correct key on the end-user's machine.

In the case of parameter-based implementations, the verification algorithm can often be reversed to create a valid serial. Another weakness of serial-based protections is that there is no mechanism that prevents a same key to be used on different software installations, allowing users to share keys.

For implementing Serial-Based copy protection model, a developer use cryptography algorithm for encrypting software code on client because when crackers want to analyze the code need more time to decrypt the code. In Parameter-based verification usually during installing software encrypted by using client parameter in Public Key vs. Private Key of cryptography and then developer can make a private key to decrypt the encrypted code. Another way is using one-way cryptography algorithm. The software makes a hash code by using client parameter and developer makes a new license that both hash codes are same [6].

c) Online Activation

The challenge response mechanism is a well-known authentication protocol typically used for authenticating specific users or computers in a networking environment [7]. The mechanism has also been applied as an improvement to the serial number protection scheme. The idea is that during installation of the application, the end-user has to enter a registration number, comparable to that of the original scheme. The difference is that instead of just running this number through a verification algorithm, the installation program com-poses a unique challenge made up of the user supplied number and a unique machine identifier. This challenge is sent to the software vendor, who verifies that the serial number is legitimate. This way, the vendor exercises control over which keys can be used. Following verification, the vendor responds with a key that is fed into the target program, where it is checked to be mathematically correct. The following process is shown in Figure 3.

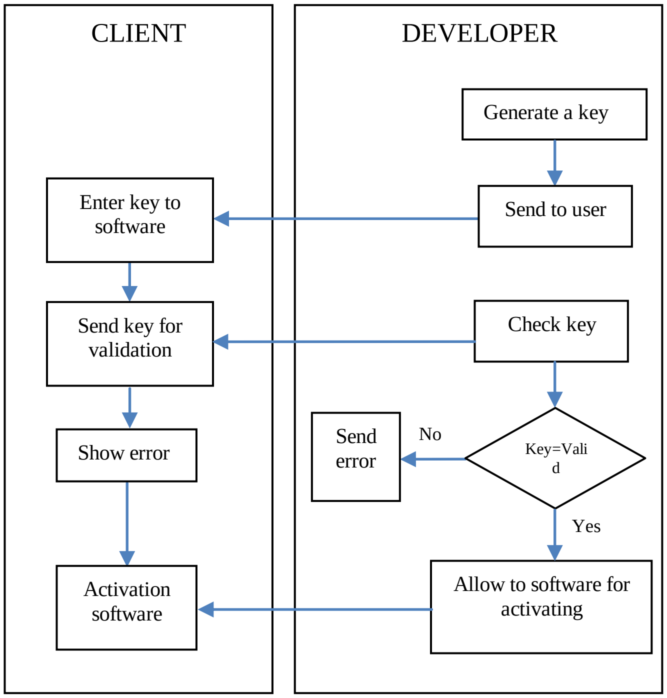

Fig.3: Workflows of Online Activation

While being slightly less flexible due to the requirement of network access during registration, this approach is definitely a step up from the conventional serial number scheme, since serials cannot be used unchecked by pirates.

2) *Hardware Tokens (Dongles) for Copy Protection*

Software comes in different types and targets different users. In cases when it contains innovative algorithms which are meant to be kept secret, this is a special kind of Intellectual Property, this need to be protected [8].

Most of the commercial software products in the market today apply some sort of copy protection. There are many technologies available for this purpose and each of them has their own implementation, security and use characteristics, but generally all of them fall into one of the two main categories: local or remote validation. The dongle-based software protection schemes fall into the former category.

Hardware-based copy protection solutions come in different forms and implementations. The main characteristic of this protection system is the use of a special piece of hardware, together with the software functionalities, to validate the given installation. Depending on when the hardware authentication is used, we can distinguish between two main types of hardware-based copy protection systems [9]:

a) Copy protection based on passive dongles:

This type of protection checks with the operating system if the required hardware device is connected to the computer during the installation or when the program starts. Figure 4 describes the flow of process in Copy protection based on passive dongles.

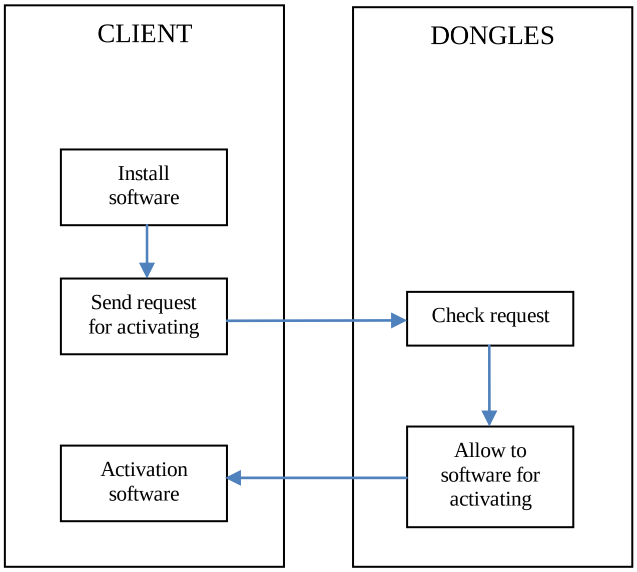

Fig. 4: Workflows of Copy protection based on passive dongles

b) Copy protection based on active dongles:

Unlike the passive mode, this model actively checks for the presence of the hardware (the dongle) to prevent software abuse. The flow of process is after the install the software when the user needs to run each process should check the dongle that it is shown in Figure 5 [10].

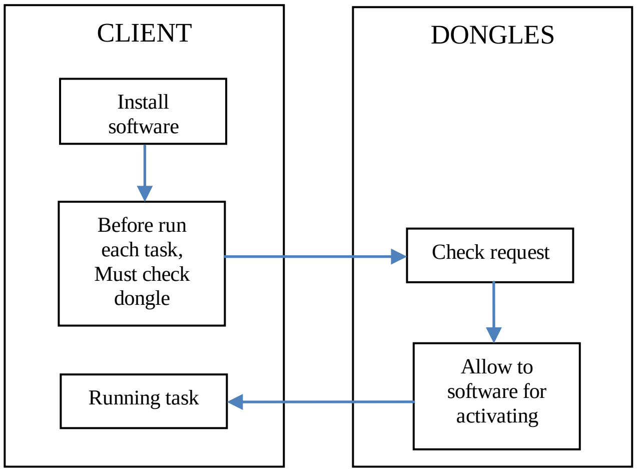
CLIENT
DONGLES
Install software
Before run each task, Must check dongle
Check request
Allow to software for activating
Running task

Fig. 5: Workflows of Copy protection based on active dongles

It is common for the hardware tokens to use the Universal Serial Bus (USB) port, but there are also other solutions that can be implemented through the Line Printing Terminal (LPT), Express Card, SD Card, PC Parallel port, Ethernet port and so on. Therefore, the term "dongle" can be used to mean the device that uses any of the ports to connect to the computer [11].

3) *Online Software*

In this method of software, software codes complete uploads on internet servers and users by an interface (Like Web Explorer) connect to the server and use the software [12]. The flow of process is at first client send a request to the server and server prepare interface and send it to client and client use it that Figure 6 shows the flow of process:

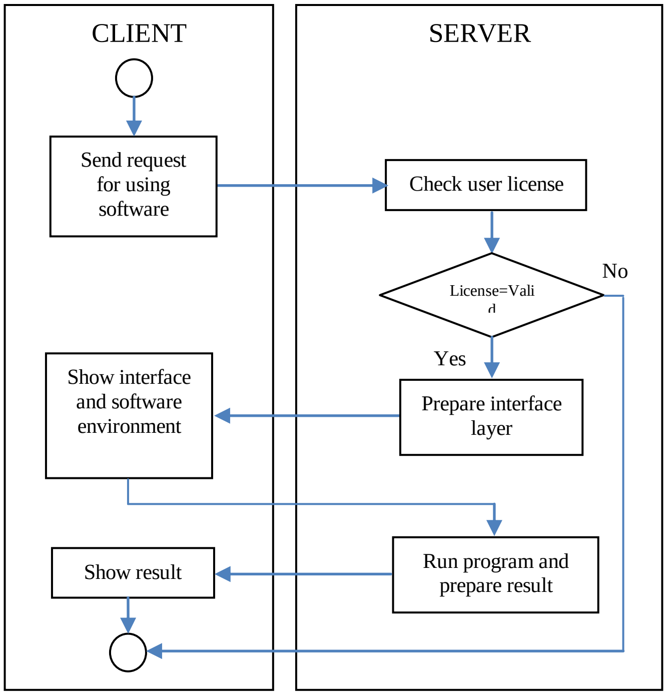
CLIENT
SERVER
Send request for using software
Check user license
License=Vali
d
No
Yes
Show interface and software environment
Prepare interface layer
Run program and prepare result
Show result

Fig. 6: Online Software Scheme

C. Security Issues about Software-Based Copy Protection

The software copy protection methods for software licensing have shown to possess weak security features, as they were broken sooner or later. As the attackers had access to the full software in the host, protection by serial numbers was circumvented either by analyzing disassembling the target program, disabling the functions that were used to connect to the validation servers or generating valid-looking serial keys which the servers accepted as authentic [11].

Some tools, such as SoftIce for Windows systems, can be used for this purpose. With this (and other similar tools), one can generate the assembler code for the targeted software and other debugging possibilities. After the extraction of the validation algorithm, it can be bypassed or a key generator can be implemented for that purpose [13].

Similar attacks can be performed on most of the software-based copy protection methods described. Therefore, a lot of efforts are being put on an alternative measure - the usage of hardware-based solutions - dongles.

D. Dongle-Based Protection Security

Dongles are pieces of hardware that are used for validating a certain copy of software. The dongle is produced and shipped together with the software package by the software vendor, thus adding to the degree of the control of the publisher over the specifics of the dongle. The security in the developed mechanisms so far has relied on the verification of the dongle presence during software execution. The software (which is installed in the computer) checks if the dongle is present in the system after it loads in the memory in order to continue its execution [14]. This is the simplest type of the dongles, but it may be circumvented using different breaking mechanisms.

Attackers have broken such systems by skipping the verification step. They have observed the call to the dongle and the respective responses using an always-true answer from an emulated dongle [14, 15] are the most implemented techniques used to break such schemes. The main weakness here is the simplicity of the operations performed in the dongle.

Other, more complex solutions to dongle-based protection systems include the possibility to perform some operations inside the dongle. The software sends a pair of input parameters to the dongle and compares the returned results to the expected one. Analyzing the calls to the dongle and dongle's response to the software, attackers have been able to break such systems [16].

Emulating dongles in software and making the software communicate with the emulated dongle, which is capable of performing the same operations as the dongle, has been a successful attack on such systems. Techniques used in this sense include reverse-engineering methods such as code debugging, obfuscation and similar are typical examples of such attacks [15]. Anti-Debugging [17] and anti-obfuscating techniques have been developed by software vendors, but it is only a matter of time until they are reverse-engineered as well.

E. Cryptography

Current copy-protection dongles, be sided the challenge-response protocol implemented, also employ cryptographic functionalities to provide another layer of security. Encryption is the process used to transform information (the plaintext) into a form which makes it unreadable, except for the person(s) who possess special knowledge to decrypt it. Normally, encryption is performed using a certain encryption algorithm and a key, while decryption is the reverse process of generating plaintext from the cipher-text (text in encrypted mode) in order to make it readable again [18].

1) Public-key vs. Private-Key Cryptography

Traditional cryptography used to work on the principle of a secret key, which the sender and the receiver of an encrypted message know and use [19]. The sender encrypts the message and the receiver is then able to decrypt it using the same key. This method is known as private key cryptography. This system works as long as the sender and the receiver are the only ones who have knowledge about the key, but the challenge for this system is agreeing on the same key to use for both parties, especially in cases when the two are far away and use electronic communication means to exchange keys. During this exchange, an adversary can intercept the exchanged keys and consequently, is able to read, modify and forge messages [20]. Therefore, managing keys in this system are a challenge (weakness).

To overcome this challenge, Public-Key Cryptography was proposed as an alternative. Introduced by Diffie and Hellman in 1976, this method was found to be useful for two primary mechanisms: privacy protection (encryption), but also for authentication (digital signatures). The concept is based on the idea that each party in the system gets a pair of keys: a private key, which is kept private from a user, and a public key, which is published and may be known to the other parties. The need for both parties to share the secret key is eliminated, as all the communication is performed on a message encrypted with a public key, while decryption can only take place if the receiver knows the secret key [21].

2) One-way Cryptographic Functions

One-way cryptographic functions1 are a very useful tool in cryptography. A one-way hash function is defined as a function F, such that it satisfies the following criterions [9]:

a) F can be applied to any argument of any size. F applied to more than one argument, F is equivalent to F applied to the bit-wise concentration of its arguments.

b) F produces a fixed size output (measured in the number of bits).

c) Given function F and an argument x, it is easy to compute F (x).

d) Given F and a "suitably chosen" (random) x, it is computationally hard to find an

$$X' \neq X$$

Such that

$$F(X) = F(X')$$

So, hash functions on a given input of any size produce an output of a fixed size (length, i.e. 56 bits). It is easy to compute the hash value of a given input, but knowing the reverse process must be computationally infeasible: knowing the hash value of an argument, it is difficult to find the original input. Randomization functions are used to encrypt the input value in such a way that small changes in the input produce big ("unpredictable") changes in the output. Therefore, these functions can be used for Integrity Checks.

Table I (appendix) compare software copy protection in some areas. The first factor to compare is protection cost; it means how much money developer should spend for running the model [22]. Next one is User-Friendly; this one reviews whether users are satisfied of the requirements or not. The third part is Copy Protection Strength; it shows the resistance for copyright protection. Model Software Uses shows which kind of software model usually uses and the last part is User Requirement; this shows users need to use software that the model implemented on it.

III. Proposed Method

This phase explains the layered architecture in client and server. For implementing this model, we created an application that a user's uses it on the client and it contains basic classes of software. On the other hand a Web Service with the tasks listed below was implemented:

1) Present secure connection between client and Web Service
2) Authenticate the user's license
3) Use special model to identify the license use by multiple computers and band the license
4) Process requests sent by the client and send processed data to the client

In the next step illustrate the technology that is used for implementing Online Module Model. Also, clarifies how application on the client and Web Service communicate together.

B. The Layered Architecture

Software architecture has emerged as an important sub-discipline of software engineering, particularly in the realm of large system development. While there is no universal definition of software architecture, there is no shortage of them, either. In this study, two layered architecture are explained below:

1) The Client Layered Architecture

This layered architecture designed for Online Modules model and this includes two layered architecture. First layer is about the application on the client. This layer has a communication with Presentation, Business and Data (This relation is shown in Figure 8). Data part has connection to local data base and server that the application in which the client gets basic data and often uses data from local data. Also, the application gets special data from the server.

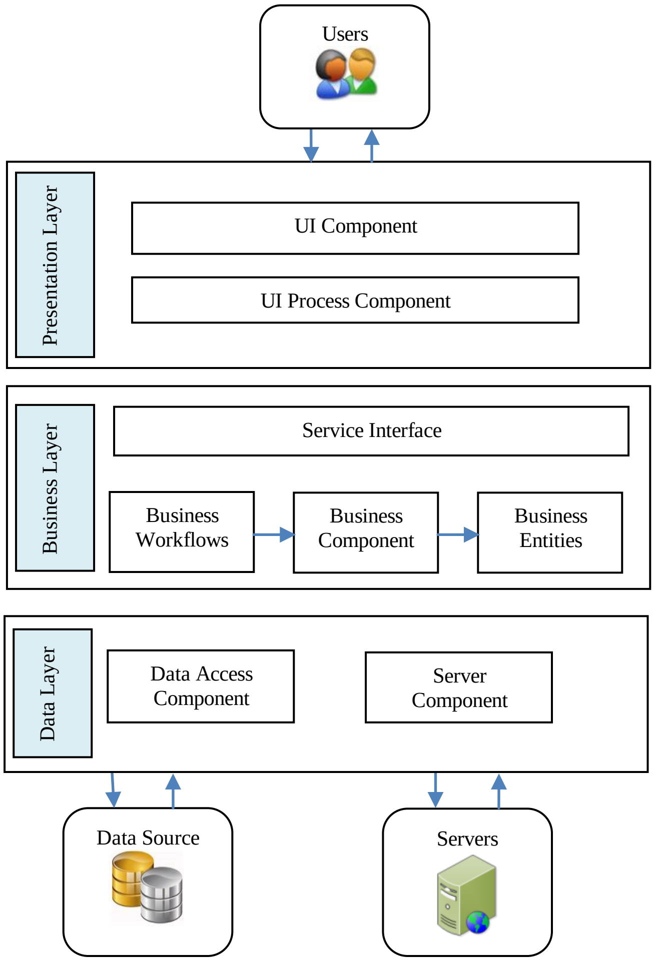

Fig. 8: The Client Layered Architecture

The client layered architecture has three layers as explained below:

2) Presentation Layer

Presentation Layer is the top layer that takes care of presenting Business Data to users and offers a way to manipulate this data, to perform Business Processes. This is the only layer that contains information about the specific technology used to create the user interface.

3) Business Layer

Business Layer is the middle layer. This is where Business Processes, Business Rules and Business Logic are implemented and where all Business Data is defined. Functionality of this layer is mainly used from the Presentation Layer.

4) Data Layer

Data Layer is the bottom layer. Its purpose is to manage the data storage and provide the upper layers with the ability to store and retrieve data. Functionality of Data Layer is mainly used from the Business Layer.

C. The Server Layered Architecture

Figure 9 shows server layered architecture of Online Module model.

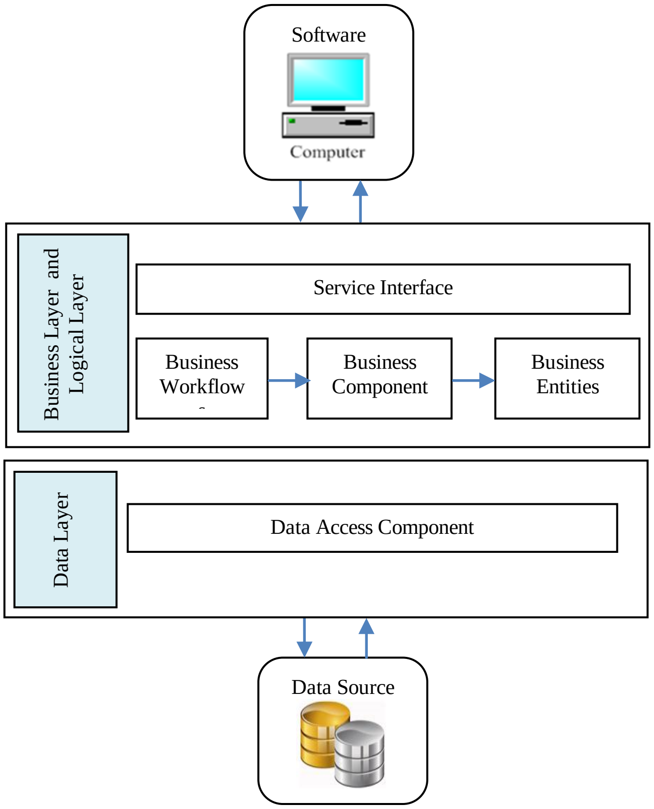

Fig. 9: server layered architecture of Online Modules model

The server layered architecture of Online Modules model is as described below:

1) Business and Logic Layer: this layer is responsible to authorize users and does their process by the data come that is sent by user.

2) Data Layer: this layer is responsible to keep user data and send it to the Business and Logic Layer.

D. Scenario

The new idea is to divide the application into two parts so that first part includes the basic modules of the application and second part includes important modules that are used for resulting. For example if assume an accounting software the first part include entering the entity and second part include the modules that create a report. Now, if we want to use this algorithm in this software we need to prepare the first part for installing on the user's computer and upload the second part on the server that connects to the internet. When the users need to report the accounting software to send some data to the module on the server then server authorize the software account if this be valid and then server processes the data and send the result to the client and at the end the software creates a report and show it to the user. Figure 10 shows flowchart about the new software copy protection model.

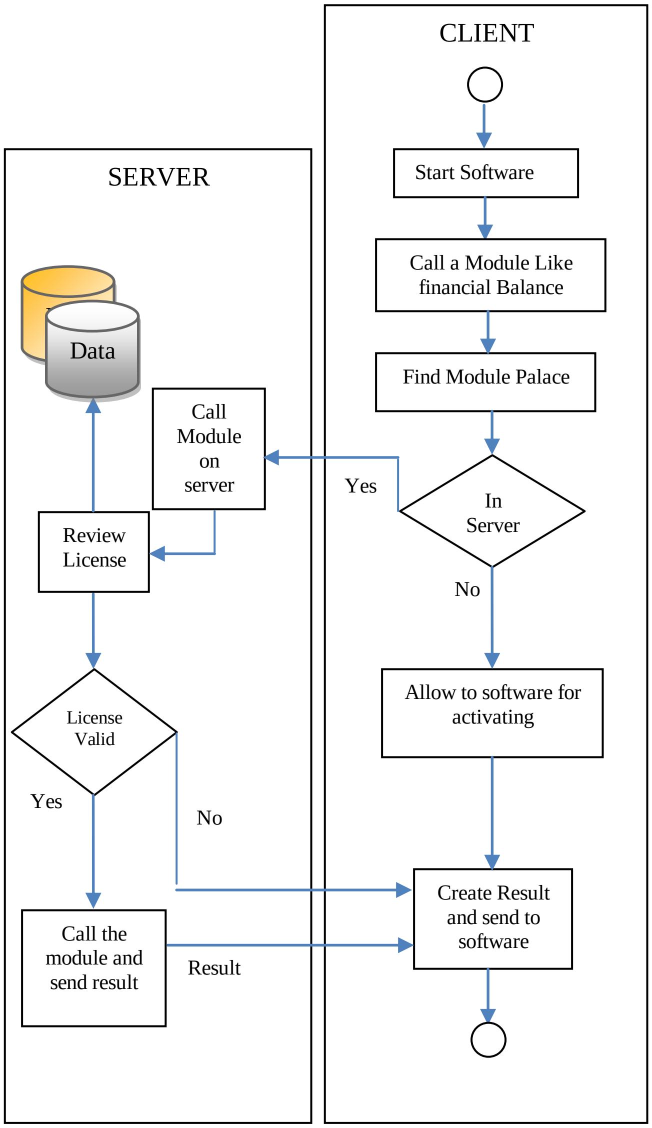
CLIENT
Start Software
Call a Module Like financial Balance
Find Module Palace
In Server
Yes
No
Allow to software for activating
Create Result and send to software
SERVER
Data
Call Module on server
Review License
License Valid
Yes
No
Call the module and send result
Result

Fig. 10: The flowchart of Online Modules Model

1) The Client parts in Online Modules Model described in below:

a) Start the software: User run the software on the client.

b) Call a modules: User response to run a part on the software

c) Find modules: The client looking for the place of modules if the place in client calls the modules on the client and if the place in server then the software sends a response to run on the server.

d) Use First Part: If the module places in client then software call the module and process the data. Finally, send the result to next step.

e) Create Result and Send to Software part: This part gets the processed data from client or server then creates a result and shows to the user.

2) The Server part in Online Modules Model described in below:

a) Call module on server: This part gets a response from the client and decrypts it. Finally sends the decrypted data to next step.

b) Review License: check the license and if it is correct and active then allows calling module. Otherwise send error message to client.

c) Call the Modules and Send the Result: this part gets the data and processes it. Finally send the result to client.

3) Frequency Password

In this algorithm that is used for implementing Online Modules model, user can activate finite number of software by one license because the algorithm does not limit a license and can install in one client for all the time but the user must use it just for one computer in limited time. Propose a solution for avoiding of this vulnerability is use Frequency Password. Frequency Password is a model that avoids using one license more than one client in this algorithm.

Frequency Password generates a new code after each communication with client and save it to its database and sends the code to the client. The next time the client sends a request for processing the data, the server checks the username, Password and the code if all of them are true then the server allows processing the data.

By use this model, if some computers active by one license, the server understands it because after each communication the Frequency password changes and other computers do not have it.

IV. CONCLUSION

The online Modules model applies strong software copy protection strength and it allows the software developer companies for protecting software copyright. Also, when this model in software is used, users are forced to purchase the software. This model aid, to the software development companies to have more sells and the company will gain more profit. Because of this, the companies can reduce the license price and users need to pay less money for buying the license.

REFERENCES

[1] Cerven, Pavol. Crackproof Your Software: Protect Your Software Against Crackers. No Starch Press, 2002.

[2] Djekic, Petar, and Claudia Loebbecke. "Preventing application software piracy: An empirical investigation of technical copy protections." The Journal of Strategic Information Systems 16, no. 2 (2007): 173-186.

[3] Jain, Ankit, Jason Kuo, Jordan Soet, and Brian Tse. "Software Cracking (April 2007)."

[4] Merckx, Gert. "Software security through targetted diversification." PhD diss., Master’s thesis, Katholieke Universiteit Leuven, 2005-2006.

[5] Usama, Mohab, and Mohamed Sobh. "Software Copy Protection and Licensing based on XrML and PKCS# 11." In Communications, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference on, pp. 856-861. IEEE, 2011.

[6] Joye, Marc. "On white-box cryptography." Security of Information and Networks (2009): 7-12.

[7] Tanenbaum, Andrew S. "Computer Networks. 4th." (2002).

[8] Nützel, Jürgen, and Anja Beyer. "Towards trust in digital rights management systems." Trust and Privacy in Digital Business (2006): 162-171.

[9] Nigurrath, S,. "Cracking with loaders: theory, general approach and a framework. ARTeam.", 2005.

[10] Eberhardt, Gergely, Zoltán Nagy, Erno Jeges, and Zoltán Hornák. "Copy protection through software watermarking and obfuscation." GYULA SALLAI–president, Scientific Association for Infocommunications ÁKOS DETREKÔI–president, National Council of Hungary for Information and Communications Technology: 2.

[11] Razeen, M., Ali, A. & Sheikh, N. M.,. "Software protection: The Last Line of Defense against Piracy.", 2003.

[12] Bobba, Jayaram, Weiwei Xiong, Luke Yen, Mark D. Hill, and David A. Wood. "StealthTest: Low overhead online software testing using transactional memory." In Parallel Architectures and Compilation Techniques, 2009. PACT'09. 18th International Conference on, pp. 146-155. IEEE, 2009.

[13] Linn, Cullen, and Saumya Debray. "Obfuscation of executable code to improve resistance to static disassembly." In Proceedings of the 10th ACM conference on Computer and communications security, pp. 290-299. ACM, 2003.

[14] Ionescu, A . "Introduction to NT Internals, Part 1: Processes, Threads, Fibers and Jobs, Relsoft Technologies.", 2004.

[15] Li, Shengying. "A survey on tools for binary code analysis." URL http://www. ecsl. cs. sunysb. edu/tr/BinaryAnalysis. doc (2004).

[16] Genov, Eugene. "Designing robust copy protection for software products." In Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, p. 49. ACM, 2008.

[17] Madou, Matias, Bertrand Anckaert, Bjorn De Sutter, and Koen De Bosschere. "Hybrid static-dynamic attacks against software protection mechanisms." In Proceedings of the 5th ACM workshop on Digital rights management, pp. 75-82. ACM, 2005.

[18] Stallman, Richard M. "the GCC Developer Community." Using GCC: The GNU Compiler Collection Reference Manual (2003).

[19] De Sutter, Bjorn, Koen De Bosschere, Bruno De Bus, Bart Demoen, and Peter Keyngnaert. "Whole-program optimization of binary executables." In Proceedings of the International Conference on Advances in Infrastructure of Electronic Business, Science, and Education on the Internet, pages CD Rom paper, vol. 50. 2000.

[20] Schneier, Bruce. "Applied Cryptography: Protocols." Algorithms, and Source Code in C, Second Edition (1995 John Wiley & Sons) (1996).

[21] Shi, Weidong, Hsien-Hsin S. Lee, Chenghuai Lu, and Tao Zhang. "Attacks and risk analysis for hardware supported software copy protection systems." In Proceedings of the 4th ACM workshop on Digital rights management, pp. 54-62. ACM, 2004.

[22] Zhao, Jianming, Nianmin Yao, and Shaobin Cai. "A new method to protect software from cracking." In Computer Science and Information Engineering, 2009 WRI World Congress on, vol. 2, pp. 636-638. IEEE, 2009.

Table I: Comparison Copy Protection Models

Copy Protection models	**Production Cost to implement**	**User-Friendly**	**Copy Protection Stability**	**Model Software Uses**	**User Requirement**
Serial-Based(Non parameter-based)	It's cheap because do not need a special requirement	Users like this model because it's simple to use software	Very Weak. Unlimited software can active just by one license	It use for common and cheap software	No need special requirement to run the protection
Serial-Based(use parameter-based)	The company needs a web server to active the license	Users do not like it because when they change their computer they will have problem	Crackers can find software copy protection code and bypass it	It use for professional software	The users must One time communication with developer
Online Activation	The company need a strong webserver and use complex algorithm	The users must one time connect to the internet	Crackers by monitoring the activation process can bypass the protection	Use for expensive common software	The users must One time communication with developer
Passive dongles	For selling each node should send a Dongle device	Users do not intersect to get the device	By on dongle can active unlimited software	Professional Software	Need Dongle
Active dongles	For selling each node should send a Dongle device	Users do not intersect to get the device	Crackers can simulate the dongle	Professional Software	Permanent use dongle device
Software Cloud	The company most has strong servers	Users must be permanent online	All the code in the server and it process the data	Light & Professional Software	Permanent use Broadband Interne

Table II: Compare Online Activation and Online Modules

	Production Cost to implement	**User-Friendly**	**Copy Protection Stability**	**Model Software Uses**	**User Requirement**
Online Activation	The company need a server and use complex algorithm	The users must one time connect to the internet	Crackers by monitoring the activation process can bypass the protection	Use for expensive common software	The users must One time communication with developer
Online Modules	The company needs some servers for supporting the model	The questioner shows about 85% of users do not have problem to implement it on the applications	This so secure because important modules of the software placed in server	It can use for all type of software	When users want to use the modules that placed in the server must connect to Internet