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

Bhawana Parihar[#1], Neha Bora[*2]
[#]Assistant Professor, Department of Computer Science, BTKIT, Dwarahat, Uttarakhand, INDIA
[#]Assistant Professor, Department of Computer Science, GBPEC, Pauri, Uttarakhand, INDIA

Abstract— *One of the most significant concerns of software corporations is protect their products against unauthorized copying. Since now the researchers proposed some copy protection models that all of them have weakness to protect against unauthorized copying. The goal of this paper is propose novel model of serial-based method that more security against illegal usage and offered Online Modules model, in serial-based method that it spilt the software to two parts as well. 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 check the software license. If the license has been valid then run the second part. This model compares with online activation model in serial-based method and we will show the proposed model has more secure against unauthorized copying.*

Keywords— *Online Modules Model, Serial-Based Method, Software Protection, Unauthorized Copying.*

I. Introduction

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

For creating software need to spend substantial money and 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 markets for selling, crackers start to crack the software in order to other users allow use it without pay money. Some users buy it but other users use illegal software by the crack. It makes a great loss to the software developer because of this the developers do not have enough motivation for create a costly software. At first the developers estimate that 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 more sell of their software so they can spend more money for preparing software and now we will see powerful applications come to markets that they can more useful for us [2].

In today's world, piracy accounts for $50 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. Most retail programs are licensed for use at just one computer site or for use by only one user at any time. By buying the software, you become a licensed user rather than an owner. You are allowed to make copies of the program for backup purposes, but it is against the law to give copies to friends and colleagues. Software piracy is all but impossible to stop, although software companies are launching more and more lawsuits against major infractions. Originally, software companies tried to stop software piracy by copy-protecting their software. This strategy failed, however, because it was inconvenient for users and was not 100 percent foolproof. Most software now requires some sort of registration, which may discourage would-be pirates, but doesn't really stop software piracy.

Some common types of software piracy include counterfeit software, OEM unbundling, soft lifting, hard disk loading, corporate software piracy, and Internet software piracy. When user enters the license code, application investigates the license validity and if the entered serial number has been correct then application been activated and user can use it. Crackers can open and change software codes in order to misuse the application as illegal access [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 to crackers to 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 used to software copy protection methods.

A. *Type of Software Licensing*

Today software developer provide different software license that the type of software licensing expressed in below.

1) Open Source / Free Software: *This is a special software licensing model which the developer goal is that users and everybody can access to 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 Open Source/ Free Software, but the source code is not open therefor it is normally, users do not able to change code. Same as the Free Software, This is not need copy-protected.*

3) Shareware: *In this type of software licensing, usually you can try it for free but with some limitations:*

a) Some software from this category used for free for limited period only. This type of software licensing called free-trial. To remove restrictions, users must purchase license.

b) The another type of shareware contain the full functionalities, it's mean for full version usage you should upgrade the software to full version and after that you access to full version of software.

2) 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.

3) 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 describes tree type of copy protection methods and explains the some type of serial-based models.

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

Using product serial numbers 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 was 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.

b) Parameter-Based Verification

In applications that can be registered online, the serial can be of a specific structure and 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; also present in the software that flow of process

For implementing Serial-Based copy protection model, developer use cryptography algorithm for encrypting software code on client because when crackers want to analyze the code need a much 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 to be send 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.

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 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 describe the flow of process in 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 user need to run each process should check the dongle

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 evice 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.

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 bypasses 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.

C. Dongle-Based Protection Security

Dongles are pieces are hardware that are used for validating a certain copy of a 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 result 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.

D. 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, 27]. 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, 25, 26]. Therefore, managing keys in this system is 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) G can be applied to any argument of any size. G applied to more than one argument, G is equivalent to G applied to the bit-wise concentration of its arguments.

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

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

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

$$X' \neq X$$

Such that

$$G(X) = G(X')$$

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 input produce big ("unpredictable") changes in the output. Therefore, these functions can be used for Integrity Checks.

It compares software copy protection in some areas. First factor to compare is protection cost, it is mean how much money developer should spend for running the model [22, 23, 24]. Next one is User-Friendly; this one review that users satisfy of the requirements or not. Third part is Copy Protection Strength; it shows the resistance for copyright protection. Model Software

Uses shows the model usually uses for which kind of software. Last part is User Requirement; this is shows what users need to use software that the model implemented on it.

III. PROPOSED METHOD

Now, we are explaining the proposed layered architecture in client and server. For implementing this model, we created an application that user uses of it on the client and it's contain basic classes of software. On the other hand implemented a Web Service that the tasks listed below:

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 request that sent by client and send processed data to the client

In the next step illustrate the technology that used for implementing Online Module Model. Also, clarified to 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 project use two layered architecture that explain it below:

1) The Client Layered Architecture

This layered architecture designed for Online Modules model and this is includes two layered architecture. First layer is about the application on the client. This layer has communication with Presentation, Business and Data (This relation shows on Figure 1). Data part has connection to local data base and server that the application on the client gets basic data and often used data from local data. Also, the application gets special data from the server.

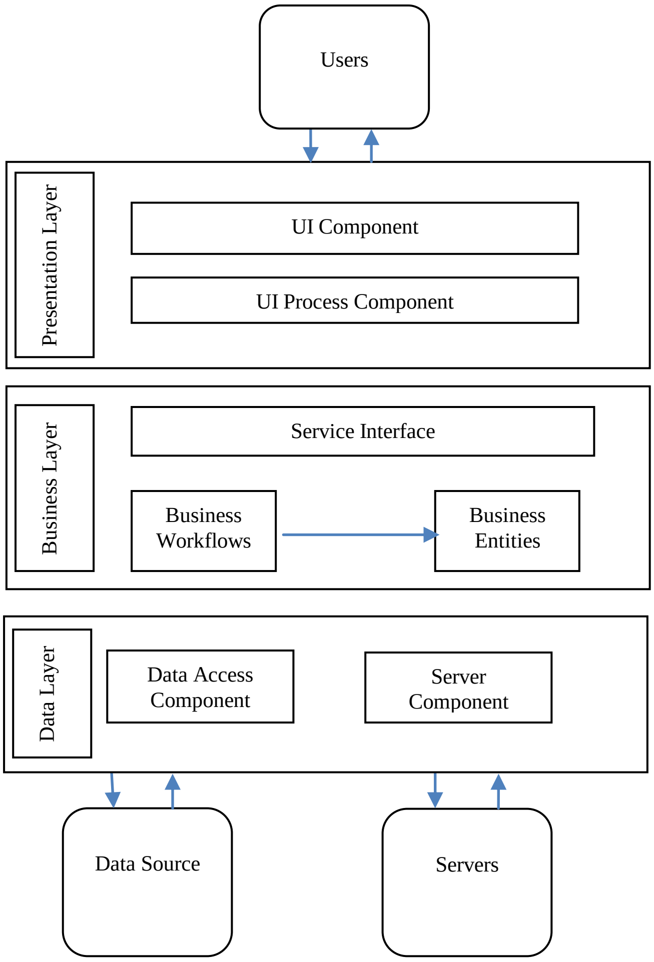

Fig. 1: The Client Layered Architecture

The client layered architecture has three layered that explain it in below:

2) Presentation Layer

Presentation Layer is the top layer that takes care of presenting Business Data to users and offering 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 2 shows server layered architecture of Online Modules model.

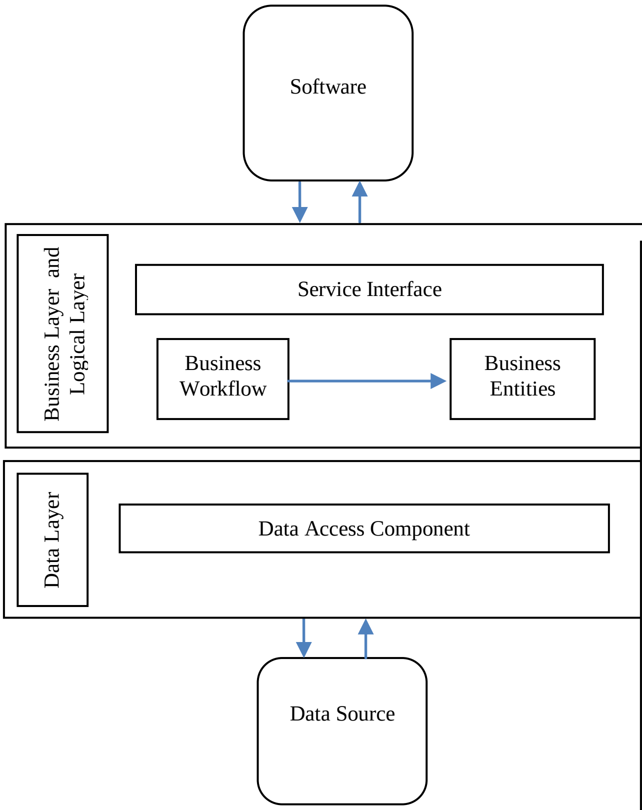

Fig. 2: server layered architecture of Online Modules model

The server layered architecture of Online Modules model described it in below:

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

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

D. Scenario

The new idea is divide the application to two parts so that first part include the basic modules of the application and second part include important modules that use for resulting. For example if assume an accounting software the first part include enter the entity and second part include the modules that create a report. Now, if we want use this algorithm for this software we preparing the first part for installing on the user computer and upload the second part on the server that connects to internet. When the users need to report the accounting software send some data to the module on the server then server authorized the software account if this be valid then server process the data and send the result to the client and the end the software create a report and show it to user. Figure 3 shows flowchart about the new software copy protection model.

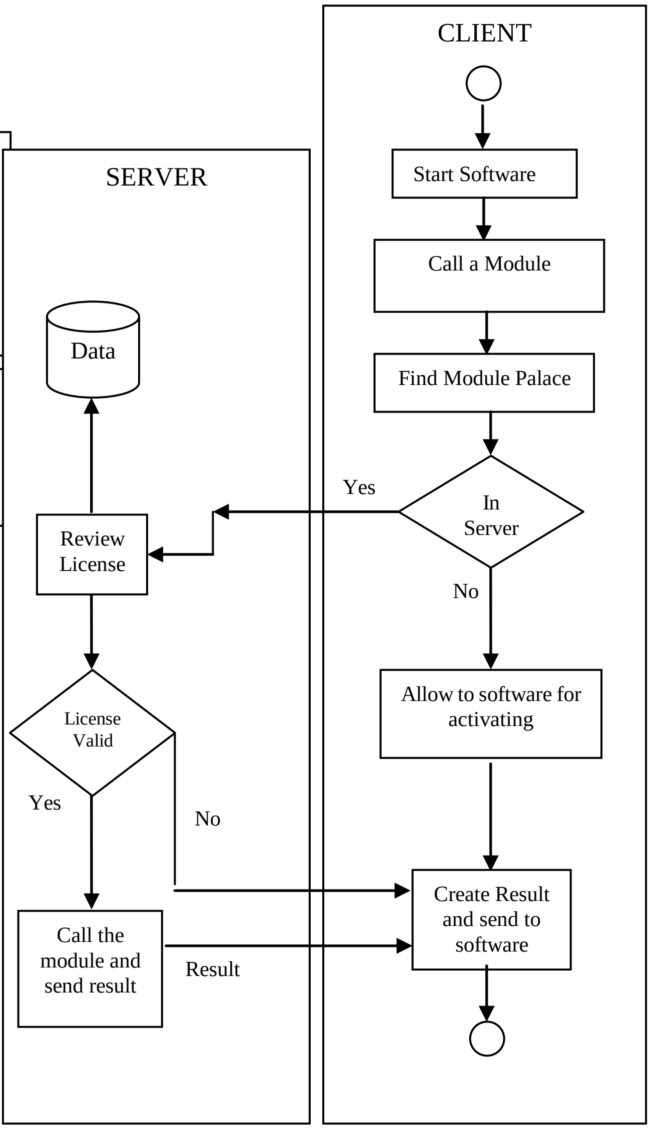

Fig. 3: The flowchart of Online Modules Model

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 user.

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

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

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

c) Request the Modules and Direct the Result: this part get the data and process it. Finally send the result to client.

3) Frequency Password

In this algorithm that used for implementing Online Modules model user can active finite number of software by one license because the algorithm do not limit that one license can install in one client for all the time but the user must use it just for one computer in limited during the 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 communicates with client and save it to its database and send the code to the client. The next time the client send a request for processing the data, server check the username, Password and the code if all of them were true then allow to the server process the data.

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

IV. CONCLUSION

Online Modules model applied strong software copy protection strength and it allows to software developer companies for protecting software copyright. Also, when use this model in software, users forced to purchase the software. This model aid to the software developers companies to have more sells and the company will gain more profit. Because of this, the companies can reduce the license price and users pay less money for buying the license.

REFERENCES

[1] Cerven, P., 2002. Crackproof Your Software. In: San Francisco: William Pollock, pp. 3-6.

[2] Djekic, P. & Loebbecke, C., 2007. Preventing application software piracy: An empirical investigation of technical copy protections. Journal of Strategic Information Systems 16, 12 july, p. 173–186.

[3] Ankit, J., Kuo, J., Jordan Soet, J. & Tse, B., 2007. Software Cracking.

[4] Merckx, G., 2006. Software Security thirough Targetted Diversification, s.l.: Katholieke University IT Leuven.

[5] Usama, M. & Sobh, M., 2011. Software Copy Protection and Licensing based. IEEE, pp. 856-861.

[6] Joye, M., 2008. On White-Box Cryptography. Security of Information and Networks,, pp. 7-12.

[7] Tanenbaum, A. S., 2002. Computer Networks. In: 4th ed. s.l.:Prentice Hall, pp. 156-165.

[8] Nützel, J. . & Beyer, A., 2006. Towards Trust in Digital Rights Management Systems. ACM workshop on Digital rights.

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

[10] Eberhardt, G. & Nagy, Z., 2006. Copy protection through software watermarking and obfuscation. Department of Measurement and Information Systems Budapest University of Technology and Economics Budapes.

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

[12] Bobba, J. et al., 2009. StealthTest: Low Overhead Online Software Testing using Transactional Memory. Appears in the Conf on Parallel Architectures and Compilation Techniques (PACT), september.

[13] Linn, C. & Debray, S., 2003. Obfuscation of Executable Code to ImprovenResistance to Static Disassembly. Department of Computer Science University of Arizona.

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

[15] Li, S., 2004. A Survey on Tools for Binary Code Analysis. Stony Brook University, pp. 37-52.

[16] Genov, E., 2008. Designing Robust Copy Protection for Software Products. International Conference on Computer Systems and Technologies.

[17] Madou, M., Anckaert, B., Sutter, B. D. & Bosschere, K. D., 2004. Hybrid Static-Dynamic Attacks against Software Protection Mechanisms. Ghent University.

[18] Vishal Gupta, Neha Bora, Nitin Arora; “Equivalence between the Number of Binary Trees, Stack Permutations and Chain Matrix Multiplication” International Journal of Advances in Engineering, Science and Technology, pp. 232-235, Vol. 2- No. 3, Aug-Oct 2012.

[19] J. Ziv and A. Lempel, "An universal algorithm for sequential Data compression", in *IEEE Transactions on Information Theory*, 1977, Vol. 23, Issue 3, pp. 337-343.

[20] Vishal Gupta; "Comparative Performance Analysis of AODV, DSR, DSDV, LAR1 and WRP Routing Protocols in MANET Using GloMoSim 2.0.3 Simulator" International Journal of Computer Applications (0975-8887), pp. 16-24, Vol. 52- No. 20, August 2012.

[21] Vishal Gupta, Neha Bora, Deepika Sharma; "A Survey of Dynamic Instruction Scheduling for Microprocessors Having Out Of Order Execution" International Journal of Computer Application, pp. 80-88, Issue 2 Vol. 5, October 2012.

[22] Suresh Kumar, Vishal Gupta and Vivek Kumar Tamta; "Dynamic Instruction Scheduling for Microprocessors Having Out Of Order Execution" Computer Engineering and Intelligent Systems, IISTE, pp. 10–14, Vol. 3- No. 4, 2012.

[23] Stallman, R., 2003. Stallman and the GCC Developer Community. Using the GNU Compiler Collection, GNU Press.

[24] Sutter, D., Bus, D., Bosschere, D. & Demoen, K., 2000. On the Static Analysis of Indirect Control Transfers in Binaries. Ghent University and Katholieke Universiteit Leuven.

[25] Schneier, B., 1996. Applied Cryptography. In: s.l.:John Wiley & Sons, pp. 39-44.

[26] Shi, W., Lu, C. & Zhang, T., 2004. Attacks and Risk Analysis for Hardware Supported Software Copy Protection Systems. College of Computingy School of Electrical and Computer Engineeringz Georgia Institute of Technology Atlanta.

[27] Zhao, J. & Yao, N., 2009. A New Method to Protect Software from Cracking. World Congress on Computer Science and Information Engineering.