DOI: https://doi.org/10.53555/nncse.v2i2.507 **Publication URL: http://nnpub.org/index.php/CSE/article/view/507**

USB KEY BASED ANTIPIRACY SOLUTION

Ketki R. Bhakare*[1], Manjiri R. Bawane[2], Madhavi S. Borkar[3], Payal R. Wawarkar[4], Sanika M. Shende[5], Heena K. Chandel[6]

*[123456]*Dept. Of Computer Science and Engg. Dr. Babasaheb Ambedkar College Of Engineering and Research Nagpur (MS), India*

[1]ketki.bhakare@gmail.com, [2]manjiri.sagi@gmail.com, [3]madhaviborkar21@gmail.com, [4]payalwawarkar11@gmail.com,[5]ssanika13@gmail.com, [6]heenachandel09@gmail.com

Corresponding Author: -
Email ID -ketki.bhakare@gmail.com

Abstract: -

USB key is a new kind of intelligent security product that comprises microprocessor and operation system. Computer software is intellectual property, and is protected by copyright law. This paper proposes a software protection method that utilizes efficient calculation ability and security space of USB key in environment, which combines protection methods, identity authentication and trusted computing technology. In this project we proposed to develop a hardware based solution to prevent software piracy. A hardware based system consist of physical device and hence cannot be shared over the internet and hence eliminates the flaws of conventional mechanisms discussed above.

Keywords: - *USB Key; authentication; software protection; physical device*

1. INTRODUCTION

Various mechanism has been developed that aim to provide a 'protection' for original software. 'Serial Key' or 'CD Key' as it is popularly known is one such mechanism. However, due to the rising popularity and availability of internet it has become very easy to obtain illegal serial keys from the internet. An additional protection in the form of 'Activation Code' has been introduced which provided greater protection, However, it is breakable

as well intellectual property provides the ability to own rights to an individual's creativity and innovation. It also allows for protection through the use of patents, trademarks, designs and copyrights. In a global economy, software systems play a dominant role, running, protecting and entertaining the world. The need for software is growing exponentially, and so is the market for pirated software.

Software piracy, the unauthorized duplication and distribution of software, is a worldwide problem that is growing at an epidemic pace. Software piracy is the illegal duplication and distribution of software packages and applications, which violate software licenses and copyright laws. Software is protected by the same laws that protect other intellectual property such as music, literature and movies. Like digital music and digital movies, software is a fairly new type of intellectual property. Software's unique existence is a challenge to those protecting the intellectual property attributed with any software.

Most users never consider stealing, but many make copies of software and, or, use pirated software every day. These users are breaking the law. In many ways, the losses from stealing software intellectual property are more severe and costly than stealing non intellectual property. Consider a car stolen from the side of a street. This car has a certain market value. But unlike a car, software can be duplicated an unlimited number of times. Each copy, contains the exact same information as the original, and is duplicated for a small price. This price is usually the cost of the blank media it is recorded on. If these were cars, one car could be duplicated many times for little cost and then resold at market value. If this were possible it would crush the car industry and economy. Counterfeit copies of software can be distributed and sold for large profits, making the software piracy industry a very damaging to the economy and a very lucrative business for software pirates.

2. LITERATURE REVIEW

There is a lot of different ways in which companies are trying to protect their software. They are buying special methods of creating the carrier just to save it from copying. Programmers are also implementing in applications the new techniques to prevent piracy.

Generally, to protect shareware applications Programming-based methods wear used, which can be used for free only for couple of days, after that user has to buy the full license. These techniques base also on checking hidden information such as license code, install date. Mentioned data are very important that's why companies put a lot of effort to encipher and hide them inside the application or in system registry. As discussed in above section there were many techniques used to prevent piracy.

To prevent piracy in [1] with the help of the key technologies including the integrity verification, access control, and sealed storage, a kind of USB key-based approach for software protection is presented at first, the integrity verification is used to protect the data consistency in USB key. secondly, the mutual authentication is performed between the USB key and software to check the validity, in which identity authentication is based on the dynamic password technology. lastly the data is encrypted between the software and USB key using triple DES, and the communication data is integrated with the MAC (message authentication code). Above all, the mutual authentication is to avoid the fake attack, the encrypted text is used for the information hiding. The MAC is adopted to protect the communication text from being tampered, the dynamic password is used to prevent the replay attack. In addition, the play-and- plug USB interface is convenient to use. [2] A lot of approaches for software tamper resistance have been proposed to ensure that the program will execute as expected. One important technique is integrity- checking which use self-hashing to check the integrity of the software, but the adversary can easily bypass the verification by locating the hash value comparison instruction. Another one is the software encryption. But most of them don't protect the keys and decryption routine as well. in this paper we propose an approach for software tamper-based resistance on USB-key. we divide the software into individual blocks and use different keys to encrypt them, a number of small code segments called security guides which are used to control the program flow and check the security of software are inserted into the blocks. And the USB-key is used to protect the keys and security self-checking.[3] Sealed storage and access control are the characteristics of the USB key integrated with smart card, a kind of USB key-based approach for virtual asset protection is presented. Virtual assets of users are stored in the hidden partition of USB KEY which is not able to read out, it would be displayed on the screen of USB_KEY after user's pin is verified. If a pin is blocked, it must be reloaded, it is designed that answering private questions and reloading key are adopted to generate a new PIN. In this scheme, line protection is used to protect the communication data from being tampered, and reloading PIN is effective to resist PIN guess, furthermore, it is convenient to use that the users are no longer to remeber all kinds of usernames and passwords.

3. METHODOLOGY

Our project will comprise of a hardware lock, which is nothing but a microcontroller-based circuit which would connect to the PC via USB port. The microcontroller in this hardware lock will be so programmed to provide a discrete unlocking key, every time the user opens the software on the PC. We call it discrete unlock key because this communication will be hidden from the user, and shall be encrypted. The user would not know what data is being exchanged and the encryption ensures that it cannot possibly be intercepted and decoded.

The idea is, that whenever the user starts the software, the software seeks this hardware lock in the the list of available removable devices attached to the system. Then it sends a long input string to the device and expects a response. The hardware lock performs a mathematical operation on this long string and returns the result. If the returned result and the expected result are the same, the software authenticates itself as a genuine version and provides full access to the user. However, if this authentication test fails the software does not start and the user cannot access it. This way, only user who have purchased the licensed version of the software, and have received this hardware can use it. Users with pirated copies do not possess the hardware key and possibly cannot attain it from the internet or any other such source and hence are unable to use it. Thus this project has the potential to become a major tool in the fight against software piracy and intellectual property protection.

The host system is the PC i.e. the system on which the protected software intended to run. The protected software is installed on this system and hardware key is inserted into the USB port of this system.

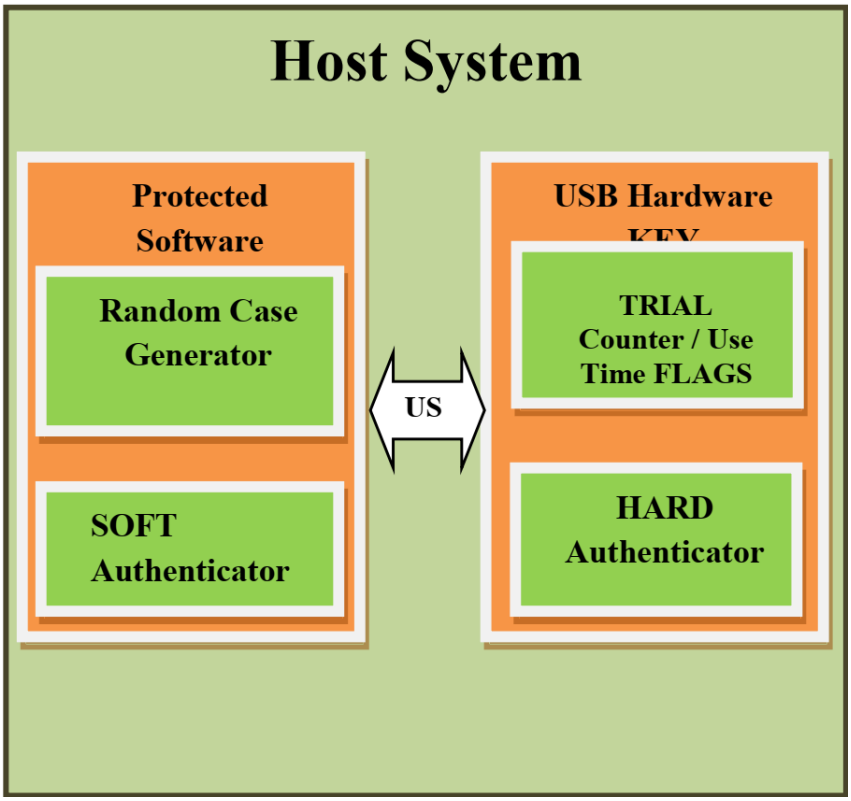

Figure: Block Diagram of the system

Our project mainly consists of two modules.

[1] Protected software Module

This is the software for which the Anti-piracy mechanism is being implemented. The protected software can be any software, developed in any language and designed to run on any platform. To implement our hardware-based piracy prevention mechanism an interface is added to the startup module of the said software. Whether the software starts or not depends on this interface. This interface is then used to control and operate the key and validate or invalidate authentication thus allowing or disallowing authentication.

Random Case Generator

For the authentication handshake to take place, the two systems viz. Antipiracy interface within the protected software and the USB key must exchange random test case strings. Then a preprogrammed function f(x) is applied over this string to generate an output. These strings must be random and be generated arbitrarily during each authentication. The random case generator module generates random test cases i.e., strings for such computations.

The Random case generator module can be seen only on the protected software side as it is the interface on the protected software that searches for the presence of the USB key and initiates the authentication process, and hence must provide test strings to the USB key.

SOFT Authenticator
The function f(x) during each level of handshake must be known to both the anti-piracy interface and the USB key. Only then can their results be compared and the authentication be allowed. The Soft Authenticator module performs these computations on the test case strings at the protected software end.

[2] USB hardware Key Module
This is the hardware module that our project uses tp authenticate the genuineness of the protected software. Only genuine users who have bought legal versions of the said software are meant to possess such a USB key. The key is intended to have a form factor such that of a pen drive, in terms of its size, and portability.

HARD Authenticator
Like the SOFT Authenticator module that performs f(x) functional computations and result evaluations at the software end, the Hard Authenticator performs the same computations computing and evaluating a result for the programmed f(x) function over the received random test strings. The hard Authenticator runs on the microcontroller built in the USB key.

Trial Counter/Use Time Flags
The USB key can also be programmed to function as a TRUE trial version enforcer. Unlike its software counterpart mechanisms to allow running of trial versions of software that use the system time and date to detect whether the particular no. Of days of the trial period have expired or not, our USB key can enforce this limitation by using inbuilt flags that can count the no. Of times a software is started, and thus refuse authentication when a preprogrammed limit has been reached.

4. CONCLUSION
Software protection based on USB key integrates functions of the hardware protection and authentication. This method prevents the attacker from analysing and cracking the software effectively.

5. REFERENCES
[1].Li MeiHong, Liu JiQiang, “USB Key-Based Approach for Software Protection”, 2009 International Conference on industrial Mechantronics and Automation.
[2].Shi-yuan Zheng, Jun Liu, “An USB-Key_Based Approach for Software Tamper Resistance”, 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE).
[3].Li MeiHong*, Zhao Yibing, Liu JiQiang*, Wang Jun**, “USB Key-Based Approach for Virtual Assets Protection”, 2010 International Symposium on Intelligence Information Processing and Trusted Computing.
[4].Jiang Yu, Chuan-fu Zhang, “Design and Analysis of a USB-Key based Strong Password Authentication Scheme”.
[5].Lei Li, Quan Liu, Xuemei Jiang, “USB Key- based Dual-factor Dynamic Authentication Scheme”, 2010 International Conference on Computational Intelligence and Security.
[6].Yang An, Bo Zhao, Yuanming Li, “Research on Software protection method based on USB Key”, 2013 Interational Conference in Computer Science and Applications.