Implementation of License File Using Cryptographic Technique

 Amol C. Dudhe

 (Student of K.D.K.C.E)

Ashutosh B. Chandanbatve Ritesh S. Yadav

(Student of K.D.K.C.E) (Student of K.D.K.C.E)

 (Department of Information Technology, K.D.K. College Of Engineering, Nagpur, India)

 **Abstract**

In earlier Software’s we used a product key which can be easily transferred to anybody and thus using this product key, a software can be install.Now, In our proposed work to eliminate the drawbacks of the earlier software we narrowed our attention towards key factor like Media Access Control(MAC) address and Hard Disk Drive (HDD) serial number (partition). In our project we are using MAC address & the HDD serial number (partition) Verification Technique which has an advantage in itself. After acquiring the MAC address & the HDD serial number (partition) of customer’s PC the License File prescribing the Vendor & Customer information. This license file along with the license reader is clubbed with the customer purchased software. Now when the Customer installs the software the license reader reads the license file contents and verifies the MAC address & the HDD serial number (partition) of customer’s machine with the MAC address mentioned in license file. If the match is found then the software is installed else the installation will be terminated. Advantage of our project over the earlier system is that cracks are not easy to develop, no serial key is generated hence there is no possibility to copying to serial key.

*Keywords: MAC Address , HDD Serial Number(partition), License Writer, License Reader*

### **Introduction**

### **What is License Genrator?**

### Whenever any customer purchase any software application (product) from vendor then customer needs serial key or product key to start application, this serial key is provided by the vendor of software application but sometimes some users cracks this serial key and uses the features of software application without any permission of vendor that is against the policy of software company. But some users are having different techniques, they purchase single license from the vendor and uses the application in multiple machines or network that is also against the policy of Software Company. To stop this cracking or illegal usage, company always tries to create unique serial key or product key but crackers cracks this serial key through some auto generated program. To solve the above problem of cracking or illegal usage, it is required to create license file that will be unique for each computer and also this license file will be associated with the machine. Thus for this approach we are creating a software which will generate license file containing all the detail of the customer. We are building this software by using advanced java technology named “UNIQUE LICENSE GENERATOR”. License generator will involve information requirement for customer details, his PC’s MAC address.

### Our project Unique license generator consist of five section which are explain below:

1. **License Reader**
2. **Problem Definition**
3. **License File Format**
4. **Literature Review**
5. **Prediction Strategy Implementation**

### **1) License Reader:**

### Vendor of product is consider as administrator and the administrator should have the power to view the license file so for retaining administrative power we have implemented “License Reader”

### .

### **a) Virtual Reader:**

### We have made virtual reader such that it will accept license file from the user as well as it will ask for MAC address and after the successful comparison of given MAC with the MAC address of license file, it will show the complete information of license file on the screen.

### Since our product will be used by the customer when he installs any software, at that time the license file’s contents i.e. MAC Address would be tested against the MAC Address of customer’s PC. Now as a vendor if we want to test this functionality then we need to implement some mechanism. For this purpose we have implemented the concept of Virtual Reader.

### Here we are going to take customer’s MAC Address and prepare a License File. Now when we are testing the software then we would go for Virtual Reader concept where we need to enter any dummy MAC Address. If this dummy MAC Address matches with the License File’s contents then the message would be displayed as “matched” else the message would be displayed as “failed”.

### **b) Real Reader:**

### Real reader will be one of the modules of our product and when the product (application software) starts running it will read MAC address from the machine and compares the machine’s MAC address with the decoded MAC address of file and after successful completion of this comparison it will allow user to access application. By implementing real reader we have given rights to vendor for manually inputting the MAC address and thereby comparing with License File.

### When the customer opts for installation of any software then automatically License Reader gets activated and it reads the License File for the MAC Address. After this it retrieves the customer’s system MAC Address and then compares this MAC Address with that of the License File. If this comparison works true then the installation is done else the installation is terminated.

### Essential condition for License Reader is that the License File should be present in the same folder as of the project with “**.lcs”** extension.

**2. Problem Definition:**

After accepting all the above information from the vendor or developer of product, writer will generate encrypted binary file known as license file. Generated license file will contain encrypted data and having custom format therefore standard application of any operating system will not able to view its contents.

During writing process, writer will generate encrypted code number from the given user information and MAC addresses and it will store that code number into the license file. Generated license file will contain all the information given by the user including photo (image) i.e. it will have ability to hold image data and binary data.

**3. License File Format:**

The License File that we have designed for our project mainly encompasses six divisions as follows:

**Key Object**: It is the key object generated by standard Key Generator class of Java.

**Image**: It is the Customer inputted Image.

**Vendor Information**: It is the Information about the Software Vendor.

**Customer Information**: It consist basic information about customer.

**Encrypted MAC address**: It’s the MAC address of Customer machine in Encrypted form..

The License File Format for our project is given below:

****

### **4. Literature Review:**

### The first software copy protection was on early [Apple II](http://en.wikipedia.org/wiki/Apple_II), [Atari 800](http://en.wikipedia.org/wiki/Atari_800) and [Commodore 64](http://en.wikipedia.org/wiki/Commodore_64) software. Game publishers, in particular, carried on an arms race with software crackers. Over time, publishers have resorted to increasingly complex countermeasures to try to stop unauthorized copying of their software.

### Unlike modern computers that use standardized [drivers](http://en.wikipedia.org/wiki/Device_driver) to manage device communications, the Apple II DOS directly controlled the step motor that moves the floppy drive head, and also directly interpreted the raw data (known as [nibbles](http://en.wikipedia.org/wiki/Nibble)) read from each track to find the data sectors. This allowed complex disk-based software copy protection, by storing data on half tracks (0 1 2.5 3.5 5 6...), quarter tracks (0 1 2.25 3.75 5 6...), and any combination thereof. In addition tracks did not need to be perfect rings, but could be sectioned so that sectors could be staggered across overlapping offset tracks, the most extreme version being known as spiral tracking. It was also discovered that many floppy drives do not have a fixed upper limit to head movement, and it was sometimes possible to write an additional 36th track above the normal 35 tracks. The standard Apple II DOS copy programs could not read such protected floppy disks, since the standard DOS assumed all disks had a uniform 35 track, 13 or 16 sector layout. Special nibble-copy programs such as *Locksmith* and *Copy II Plus* could sometimes duplicate these disks by using a reference library of known protection methods, but when protected programs were cracked they would be completely stripped of the copy protection system, and transferred onto a standard DOS disk that any normal Apple II DOS copy program could read.

### One of the primary routes to hacking these early copy protections was to run a program that simulates the normal CPU operation. The CPU simulator provides a number of extra features to the hacker, such as the ability to single-step through each processor instruction and to examine the CPU registers and modified memory spaces as the simulation runs.

### The Apple II provided a built-in opcode disassembler, allowing raw memory to be decoded into CPU opcodes, and this would be utilized to examine what the copy-protection was about to do next. Generally there was little to no defense available to the copy protection system, since all its secrets are made visible through the simulation. But because the simulation itself must run on the original CPU, in addition to the software being hacked,the simulation would often run extremely slowly even at maximum speed.

### The most common protection method on the Atari computers was "bad sectors". These were sectors on the disk that were intentionally unreadable by the disk drive. The software would look for these sectors when the program was loading and would stop loading if an error code was not returned when accessing these sectors. Special copy programs were available that would copy the disk and remember any bad sectors. The user could then use an application to spin the drive by constantly reading a single sector and display the drive RPM. With the disk drive top removed a small screwdriver could be used to slow the drive RPM below a certain point. Once the drive was slowed down the application could then go and write "bad sectors" where needed. When done the drive RPM was sped up back to normal and an uncracked copy was made. Of course cracking the software to expect good sectors made for readily copied disks without the need to meddle with the disk drive. As time went on more sophisticated methods were developed, but almost all involved some form of malformed disk data, such as a sector that might return different data on separate accesses due to bad data alignment. Products such as the "Happy Chip" became available that were hardware add-ons similar to today's game console [modchips](http://en.wikipedia.org/wiki/Modchip). However, the Happy Chip would allow the user to make exact copies of the original program with copy protections in place on the new disk. "Happy Chip" owners quickly became popular in game trading circles.

### On the [Commodore 64](http://en.wikipedia.org/wiki/Commodore_64), several methods were used to copy protect software. For software distributed on [ROM](http://en.wikipedia.org/wiki/ROM)[cartridges](http://en.wikipedia.org/wiki/Cartridge_%28electronics%29), [subroutines](http://en.wikipedia.org/wiki/Subroutine) were created that attempted to write to the ROM. If nothing happened, the presence of a ROM cartridge was verified, but if the software had been moved to [RAM](http://en.wikipedia.org/wiki/RAM), the write routine would disable the software. Because of the operation of Commodore floppy drives,

### some write protection schemes would cause the floppy drive head to bang against its stop and could cause the drive head to become misaligned. In some cases, cracked versions of software were desirable to avoid this result.

### Most of the early software crackers were computer hobbyists who often formed groups that competed against each other in the cracking and spreading of software. Breaking a new copy protection scheme as quickly as possible was often regarded as an opportunity to demonstrate one's technical superiority rather than a possibility of money-making. The cracker groups of the 1980s started to advertise themselves and their skills by attaching animated screens known as [crack intros](http://en.wikipedia.org/wiki/Crack_intro) in the software programs they cracked and released. Once the technical competition had expanded from the challenges of cracking to the challenges of creating visually stunning intros, the foundations for a new subculture known as [demo scene](http://en.wikipedia.org/wiki/Demoscene) were established. Demo scene started to separate itself from the illegal "warez scene" during the 1990s and is now regarded as a completely different subculture. Many software crackers have later grown into extremely capable software reverse engineers; the deep knowledge of assembly required in order to crack protections enables them to engineer drivers in order to port them from binary-only drivers for [Windows](http://en.wikipedia.org/wiki/Windows) to drivers with source code for [Linux](http://en.wikipedia.org/wiki/Linux) and other [free](http://en.wikipedia.org/wiki/Free_software) operating systems. With the rise of the [Internet](http://en.wikipedia.org/wiki/Internet), software crackers developed secretive online organizations.

### Most of the elite, or well known cracking groups make software cracks entirely for respect in "[The Scene](http://en.wikipedia.org/wiki/The_Scene)", not Profit. From there, the cracks are eventually leaked onto public internet sites by people/crackers who use the well protected/secure FTP release archives, and are made into pirated copies and sold illegally by other third parties. The today is formed of small groups of very talented people, who more or less compete to have the more genius crackers, and methods of [cracking](http://en.wikipedia.org/wiki/Cracking) and [reverse engineering](http://en.wikipedia.org/wiki/Reverse_engineering).

### **5. Prediction Strategy Implementation:**

### After understanding the concept of license generator, it is required to develop project by process, and this process also requires some extra study of different technology .So as per our project requirement we have distributed our project study in three basic parts as follows:

### Study of Tagged Image File (tiff) format.

### Study of swing of advanced Java.

### Study of Input Output operation in Java.

### The Step by Step process for our Project development is as follows:

### **5.1 MAC Address:**

In [computer networking](http://en.wikipedia.org/wiki/Computer_network), a [Media Access Control](http://en.wikipedia.org/wiki/Media_Access_Control) address (MAC address), Ethernet Hardware Address (EHA), hardware address, adapter address or physical address is a quasi-[unique identifier](http://en.wikipedia.org/wiki/Unique_identifier) assigned to most [network adapters](http://en.wikipedia.org/wiki/Network_adapter) or network interface cards (NICs) by the manufacturer for identification. If assigned by the manufacturer, a MAC address usually encodes the manufacturer's registered identification number.

 Sample program to read MAC address of machine. application will retrieve MAC address of Network Card which is a 6-byte number, for ex. 56 -22 -89 -40 -76-20. MAC address is generated using standard java interface called Network Interface. Encryption of MAC address by using standard algorithm of Java language and not custom algorithm to provide security. We are using DES Algorithm (Data Encryption Standard).

### The encryption is done by the standard algorithm of java on the basis of a key which is generated using machine’s time (in milliseconds since 1900) so the key will be unique and the same key cannot be generated again. decryption of MAC address by using the same standard algorithm of Java language.



Figure 1.MAC address

Three numbering spaces, managed by the [Institute of Electrical and Electronics Engineers](http://en.wikipedia.org/wiki/Institute_of_Electrical_and_Electronics_Engineers) (IEEE), are in common use for formulating a MAC address: MAC-48, EUI-48, and EUI-64. The IEEE claims [trademarks](http://en.wikipedia.org/wiki/Trademark) on the names "EUI-48" and "EUI-64", where "EUI" stands for Extended Unique Identifier.

Although intended to be a permanent and globally unique identification, it is possible to change the MAC address on most of today's hardware, an action often referred to as [MAC spoofing](http://en.wikipedia.org/wiki/MAC_spoofing). Unlike [IP address spoofing](http://en.wikipedia.org/wiki/IP_address_spoofing), where a sender spoofing their address in a request tricks the other party into sending the response elsewhere, in MAC address spoofing (which takes place only within a local area network), the response is received by the spoofing party.

A host cannot determine from the MAC address of another host whether that host is on the same OSI [Layer 2](http://en.wikipedia.org/wiki/Layer_2)[network segment](http://en.wikipedia.org/wiki/Network_segment) as the sending host or a network segment [bridged](http://en.wikipedia.org/wiki/Network_bridge) to that network segment.

In TCP/IP networks, the MAC address of a subnet interface can be queried with the [IP address](http://en.wikipedia.org/wiki/IP_address) using the [Address Resolution Protocol](http://en.wikipedia.org/wiki/Address_Resolution_Protocol) (ARP) for [Internet Protocol](http://en.wikipedia.org/wiki/Internet_Protocol) Version 4 ([IPv4](http://en.wikipedia.org/wiki/IPv4)) or the [Neighbor Discovery Protocol](http://en.wikipedia.org/wiki/Neighbor_Discovery_Protocol) (NDP) for [IPv6](http://en.wikipedia.org/wiki/IPv6). On broadcast networks, such as Ethernet, the MAC address uniquely identifies each node and allows frames to be marked for specific hosts.

### **5.2 Image Reader:**

### Reading image file name from the user.

### Display the image on the window through JAI technology.

### Display the full path from where the image is chosen.

### **5.3 Image Writer:**

### Writing image data in file.

### Writing image as well as other information in file.

### Other information contains customer information, vendor information and encrypted MAC address.



Figure 2 Vendor Information

**5.4 License Writer:**

Creating custom format license file along with image and other information with the help of above modules.

At the time of creation of license file the information can be reviewed.

Before writing the data into license file it is encrypted.Integration of all the modules.

Testing of all the modules.



 **Figure 3 License File**

**Conclusion:**

Piracy is the most hunting problem faced by the most software companies. In our project “License Generator” we have come up with some methods to curb this piracy to some extent. Since, this project involves some advanced features liability between customer & vendor are upto the utmost level. The benefits in our project are, the software purchased for one machine is limited to that machine only. This will ultimately help in reducing software piracy & it will definitely help the company to gain huge profit. Since, every project has certain drawbacks, our project also has certain shortcomings such as, if a customer shifts his LAN card to other machine he can easily run the software on that machine, etc. To eliminate these drawbacks we are trying for “Configuration Verification technique“ and we hope that this will make our project an unique application.**References:**

1. Lein Harn and Jian Ren ,” Generalized Digital Certificate for User Authentication and Key Establishment for Secure Communications”, IEEE Transaction On Wireless Communication,Vol.10, No. 7,pp 2372-2379, July 2011.
2. Ingrid Verbauwhede ,Frank Hoornaert,” Security and Performance Optimization of a New DES Data Encryption Chip”,IEEE Journal Of Solid-State Circuits,Vol,23, No3, pp.647-656 June 1988.
3. Fangguo Zhang,” Cryptanalysis of Chang *et al*.’s Signature Scheme with Message Recovery” , IEEE Communicatios Letters, Vol.9,No.4,pp.358-359 April 2005.
4. Kai-Hui Lee and Pei-Ling Chiu,”An Extended Visual Cryptography Algorithm for General Access Structures”, IEEE Transaction On Information Forensics And Security, Vol.7, No.1,pp 219-229,Februrary 2012.
5. Jingxian Wu, Geoffrey Ye Li,” Collision-Tolerant Media Access Control with On-Off Accumulative Transmission” IEEE Transaction On Wireless Communications , Vol .12, No.1,pp 50-59,January 2013