International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

PROTECTING THE SOFTWARE PRIVACY USING HARDWARE PARAMETERS AND LICENSE KEY

J KAVITHA REDDY[1], G VENUGOPAL RAO[2], C.S.BASAVARAJ[3], ASHOK AYUSHI[4], BOMMI YASHWANTH SIMHA[5], E SAI SHARAN REDDY[6]

[1] Assistant Professor, Department of CSE, Malla Reddy Engineering College (Autonomous), Hyderabad, India

[2] Assistant Professor, Department of CSE, Malla Reddy Engineering College (Autonomous), Hyderabad, India

[3] Assistant Professor, Department of CSE, Malla Reddy Engineering College (Autonomous), Hyderabad, India

[4,5,6] UG Student, Department of CSE, Malla Reddy Engineering College (Autonomous), Hyderabad, India

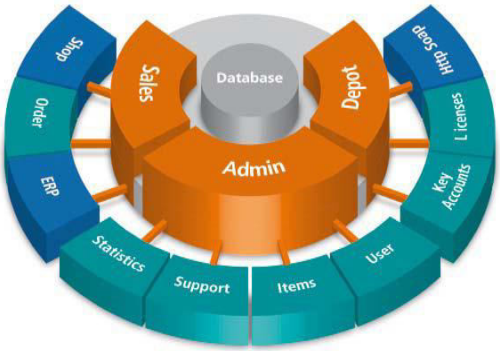
ABSTRACT:

The project aims at securing the software given to the customers such that the application runs only for particular count, up to a date, number of days starting from the installation date, runs in one machine. The application if copied to other system will not work. During first run, a license key is generated based on the Network MAC address which is unique in nature and the disk drive's serial number. The key is encrypted using Triple DES (Data Encrypted Standard) and stored in Windows Registry. In successive runs, the application looks up the value in the registry and may or may not continue. Thus the application aims in protecting the software piracy from one customer to another. The main objective of this project is securing the software given to the customers such that the application runs only for particular count, up to a date, number of days starting from the installation date, runs in one machine. The application if copied to other system will not work.

INTRODUCTION

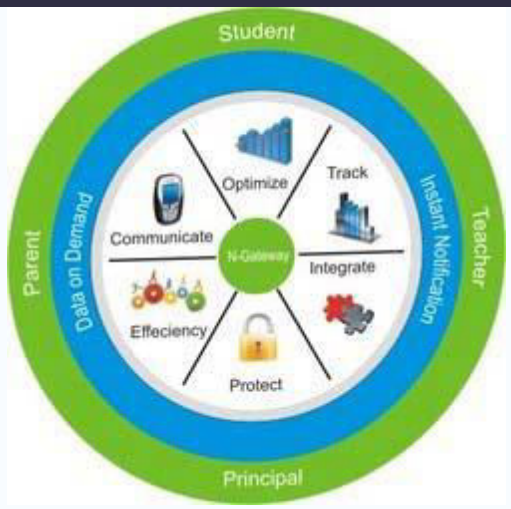
The project "Enhanced License Maker for Software Protection" is designed using Microsoft Visual Studio.Net 2003 as front end and Microsoft SQL Server 2000 as back end which works in .Net framework version 1.1. The coding language used is Visual Basic .Net. The project aims at securing the software given to the customers such that the application runs only for particular count, up to a date, number of days starting from the installation date, runs in one machine. The application if copied to other system will not work. During first run, a license key is generated based on the Network MAC address which is unique in nature and the disk drive's serial number. The key is encrypted using Triple DES (Data Encrypted Standard) and stored in Windows Registry. In successive runs, the application looks up the value in the registry and may or may not continue. Thus the application aims in protecting the software piracy from one customer to another.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org



LITERATURE SURVEY

A software license (or software license in commonwealth usage) is a legal instrument (by way of contract law) governing the usage or redistribution of software. All software is copyright protected, irrespective of whether it is in the public domain. Contractual confidentiality is another way of protecting software. A typical software license grants an end-user permission to use one or more copies of software in ways where such a use would otherwise constitute copyright infringement of the software owner's exclusive rights under copyright law.Some software comes with the license when purchased off the shelf or an OEM license when bundled with hardware. Software can also be in the form of freeware or shareware. Software licenses can generally be fit into the following categories: proprietary licenses and free and open source licenses, which include free software licenses and other open source licenses. The features that distinguishes them are significant in terms of the effect they have on the end-user's rights.



A free or open source license makes software free for inspection of its code, modification of its code, and distribution. While the software released such a license, like the GNU General Public License can be sold for money, the distribution cannot be restricted in the same ways as software with copyright and patent restrictions used by firms to require licensing fees.A license manager is a software tool used by software vendors to control where and how their products are able to run. License managers protect the software vendor from losses due to piracy and enable the vendor to offer a range of licensing models, such as trial licenses, subscription licenses, feature-based licenses, floating licensing and usage-based or metered licensing from same software package they provide to all users.A license manager is different from a software asset manager, which end-user organizations employ to manage the applications they have purchased from many vendors.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

EXISTING SYSTEM:

In Existing system, the software is not protected by giving proper security. Users can crack or hack the software and use the software unlimited no of times. Software Works by uninstalling the older or trial version or by editing the cleaning the registry

PROPOSED SYSTEM:

In Proposed system, the software is fully protected by giving proper security permissions. Users can't crack or hack the software and use the software unlimited no of times. This is don't the editing values in registry and getting details about MAC Address, System Drive Information and etc. Software does not work by uninstalling the older or trial version or by editing and cleaning the registry

MODULES AND ITS DESCRIPTION

There are 4 modules in this project. They are

1. Generate License
2. Getting MAC & Other Addresses
3. Reading & Writing Registry
4. Verifying Settings

1. Generate License

In the generate license module, we can give the installation date, number of days to be valid and end of day. The above details are to be stored in to database at the same time in database also. This information is checked at our protected software running.

2. Getting MAC & Other Address

In the second module, we are checking the MAC Address and IP Addresses for verifying system registry. The MAC address is a unique value associated with a network adapter. MAC addresses are also known as **hardware** addresses or **physical** addresses. They uniquely identify an adapter on a LAN. MAC addresses are 12-digit hexadecimal numbers (48 bits in length). By convention, MAC addresses are usually written in one of the following two formats: MM:MM:MM:SS:SS:SS

MM-MM-MM-SS-SS-SSThe first half of a MAC address contains the ID number of the adapter manufacturer. These IDs are regulated by an Internet standards body (see sidebar). The second half of a MAC address represents the serial number assigned to the adapter by the manufacturer. In the example, 00:A0:C9:14:C8:29The prefix 00A0C9indicates the manufacturer is Intel Corporation.

3. Reading & Writing Registry

In the 3rd module we are accessing the system registry for getting the protection information. This information is verified at the time software running.

4. Verifying Setting

The Registry has a hierarchal structure, like the directories on your hard disk. Each branch (denoted by a folder icon in the Registry Editor, see below) is called a Key. Each key can contain other keys, as well as Values. Each value

contains the actual information stored in the Registry. There are three types of values; String, Binary, and DWORD - the use of these depends upon the context.

Security architecture/design analysis

Security architecture/design analysis verifies that the software design correctly implements security requirements. Generally speaking, there are four basic techniques that are used for security architecture/design analysis.

Logic analysis

Logic analysis evaluates the equations, algorithms, and control logic of the software design.

Data analysis

Data analysis evaluates the description and intended usage of each data item used in design

FUTURE ENHANCEMENT

The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Improvements can be appended by changing the existing modules or adding new modules. One important development that can be added to the project in future is file level backup, which is presently done for folder level.

CONCLUSION

It is concluded that the application works well and satisfy the users. The application is tested very well and errors are properly debugged. The site is simultaneously accessed from more than one system. Simultaneous login from more than one place is tested. The site works according to the restrictions provided in their respective browsers. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one. The speed of the transactions become more enough now.Thus the application aims in protecting the software piracy from one customer to another and provides gain to the companies which are developing the product.

REFERENCES

[1] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," Proc. Advances in Cryptology(Crypto84), LNCS 196, Springer- Verlag, 1984, pp.47-53.

[2] J.C. Cha and J.H. Cheon, "An identity-based signature from

[3] gap Diffie-Hellman groups," Proc. Public Key Cryptography(P KC03), LNCS 2139, Springer-Verlag, 2003, pp.18-30.

[4] F. Zhang and K. Kim, "Efficient ID-based blind signature and proxy signature from bilinear pairings," Proc. Advances in Cryptology(Crypto03), LNCS 2727, Springer-Verlag, 2003, pp.312-323.

[5] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: Delegation of the power to sign messages," IEICE Trans. Fundamentals, ol.E79-A, o. 9, pp. 1338-1353, Sep.1996.

[6] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," Proc. International Conference on Information,

Communications and Signal Processing(ICICS97), LNCS 1334, Springer-Verlag, 1997, pp.223- 232.

[7] B. Lee, H. Kim and K. Kim, "Secure mobile agent using strong nondesignated proxy signature," Proc. 14th Australasian Conference on Information Security and Privacy(ACISP01), LNCS 2119, Springer- Verlag, 2001, pp.474-486.

[8] F. Zhang, R.S. Naini, and C.Y. Lin, "New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairings," Cryptology ePrint Archive, available at: http://eprint.iacr.org/2003/, 2003.