Abstract

Unlike any hardware component or product a software product can easily be copied from any source to any destination without involving cost, people, and effort. As there lies the fact that software can be copied and since software development is costly with reference to resources, personnel, effort, intellectual property, etc, software development companies do not allow people to copy their software setups without license. So it is necessary to devise a mechanism to ensure that whosoever possesses license or paid royalties alone should be able to access the software and that the license holders should also be legitimate users. This will ensure that the licensed software should not be run on more number of systems than specified. This has become a major challenge in the software industry and for the developers of the software products in protecting their intellectual property rights [3].

There are many means of gaining unauthorized access to software products being used. Also, to control this access to copying the software products on various systems, software developers generally restrict its operation by placing a limit on the product being used in a particular environment or making the product unusable after a predetermined period of time thereby developing a mechanism so as to generate a key to complete its installation on user’s site. Another approach is to generate a call automatically to the License Provider’s site each time the product is run for its installation. Whenever an attempt is made to install the software at client’s environment, an automatic request is sent to the License Provider’s site to verify and check whether the installation of the software product is an authorized one. If it is, the product can be installed otherwise will not be permitted.

There are many potential security breaches that must be taken into account while protecting sensitive data on the network. Now we shall go through some of the cases that the developers of the software products should be aware of. At first, consider the case where all the license generation certificates of various products are maintained by the License Provider Server in the form of files in two or more systems to which clients can connect. Suppose the License Provider Server can give users the facility to reserve licenses for a particular period of time so that the users can disconnect from the network and run the software on their machine. If at all license is to be issued to a new client, then only that part of the information from the license generation process needs to be stored on the client’s machine as a license certificate. It is required for the License Provider Server to maintain the list of all requests made by various users for licenses, so that any license that is already reserved cannot be granted to another user until the date of expiration of the certificate. Also the development organization should provide a facility to all the license holders to interrupt such requests, by informing the License Provider Server that the license is already provided to that user. To overcome these issues, the License Provider Server should also maintain the list of all requests granted and also the requests to be granted to avoid such confusion.

Consider a different case in which the purchaser of licensed software is maintaining numerous copies of the same intended software. If the user/customer/purchaser of the software comes to know where the copy of the license certificate is stored on the machine, he could create a copy of it and install the software on another system. In this manner, he would create multiple
copies of the licensed software and use the software on many other machines even though it is intended for a single machine, without the knowledge of the License provider, which is a situation that most of the software developers dread.

There can be another kind of solution where the sensitive information that is stored on the disk is captured by the intruders on the net. The problem here is this information should be prevented from being accessed by unauthorized persons. This is a major challenge for the people maintaining sensitive data. Generally the data might be disclosed due to several reasons probably from the user end by having the chance of copying the sensitive data to an unprotected area on the machine or due to accidental deletion of the data or by catastrophic means. Anything that is stored in a database can potentially be compromised intentionally or unintentionally. Even if the information is encrypted, data integrity might be at risk; for instance, data could be deleted without needing to interpret their contents. So, how to protect sensitive data from the intruders on the net? One of the best solutions is to store sensitive information on the disk in the form of hidden files of encrypted files. Even then it is easy for the user to trace out where the data is actually stored and make a copy. These techniques discussed above do not stop skilled personnel from copying the contents of a hidden file. Hence it is required to devise a mechanism to stop such activities and control the unauthorized usage of data over net or by any other means.

In this research we focus on generating NodeLocked license to the software with Multi Prime Rebalanced RSA algorithm. In software industry NodeLock Licensing concepts are not much appropriate when compared to distributed license as there would be a need of changing the hardware parts frequently. But distributed license does not meet the need or requirements for disconnected or offline software environments. Though NodeLock License Methodology is most appropriate for addressing this problem, there is no right pattern or solution available to address the specific needs of the target environment. To address this problem in implementing NodeLock License Methodologies everywhere outside connecting environments, we propose many variations of NodeLock Licensing along with target security needs by extending RSA with multi prime support called Multi Prime Rebalanced RSA. Also we suggest the best possible variations of the tolerances on hardware attributes, software features and other customizations. These NodeLock licenses are always specific to a particular node or a system in which software is to be installed and operated. During the license generation process, it is part of the installation procedure that the terms and conditions of the license should be agreed upon by the user which is present in the license certificate in an encrypted form and managed by the License Provider Server. This agreement between the software developer and the user of the software is permanent and is valid till the expiry of license as indicated in the certificate. This even introduces a security issue that addresses the duplication of the license information by copying it on another system and, hence, with the NodeLock License, the entire installation process can be controlled and gets completed in regard to the validation of encrypted hardware details as is in our algorithm. The users of the software licensed with NodeLock should be aware that the License Provider monitors the usage of the software and that any unauthorized access to the software installation procedure would not be entertained which is the main concern of the software development organization providing NodeLock licenses. This fact enforces a constraint on the users of the software that NodeLocked licensed software cannot be installed nor run on more than one system and any unpermitted access
to the software will automatically and strictly be monitored. In this research, we present NodeLocked licensing methodologies for protecting sensitive data during NodeLock licensed software installation. Generally sensitive data will be stored on disks in the form of hidden files. The RMPRSA algorithm that is defined for protecting the sensitive data during the NodeLock licensed software installation is based on considering hardware attributes of the client's machine. This sensitive data is encrypted and is stored in the License Providers' site and hence no copy of it is available on the disk in user's site. In particular the present RMPRSA algorithm is applied to a NodeLock license while generating the keys for the installation of software on user's site even though the user is working offline which is the main advantage of NodeLock license over distributed licenses.