THE DESIGN AND IMPLEMENTATION OF SOFTWARE REGISTRATION MODULE BASED ON RSA ENCRYPTION ALGORITHM

1JIQIU DENG, 2BO BI, 3QIANHONG WU, 4NA LI

1,2,3,4Key Laboratory of Metallogenic Prediction of Nonferrous Metals, Ministry of Education, Changsha, 410083, China
1,2,3,4School of Geosciences and Info-Physics, Central South University, Changsha, 410083, China
E-mail: 1djq318@163.com, 2whatisbibo@163.com, 3qhwu19@163.com, 4anlysmile@qq.com

ABSTRACT

Based upon RSA encryption algorithm and MD5 hash function, this paper designs a software license scheme which adopts the idea of “one machine one code”. Using # language in the .NET Framework 4.0 platform, the author completed the license scheme and it ran on mining management information system successfully. It not only ensures the safety of the software and convenience in registration authorization, but also makes registration code more aesthetic and effective.

Keywords: RSA Encryption Algorithm, Software License, MD5

1. INTRODUCTION

For professional software products, software developers must develop software registration authorization module in order to prevent illegal software theft and transfer. Currently, the main software encryption methods are symmetric cryptosystem and asymmetric cryptosystem. Symmetric cryptosystem mainly covers AES (Advanced Encryption Standard), DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm) and so on. Symmetric cryptosystem security depends on the confidentiality of secret key, so security is not enough. The main stream of asymmetric cryptosystem which is also called public key cryptosystem is divided into three parts: based on the difficulty of the big integer factorization, the difficulty of the discrete logarithm calculation and elliptic curve public key cryptography.


In terms of software Registration and authorization, Xu Su-juan(2006) put forward a software registration program in which Xu Su-juan encrypts the hard disk serial number, MAC address with MD5 encryption method as the registration number and encrypts the registration number with RSA encryption method for storage. As the registration number was encrypted with MD5 method, the software security was not high but the registration number was saved safely. Zhao LiPing, Shu QiLiang, Lai XiaoLiang(2011) proposed a file encryption scheme based on RSA encryption algorithm and determined whether the files had been tampered with MD5 Algorithm. The encryption and decryption method had been implemented of a single text. Liu Wentao(2012) summarized the software license protection program and proposed that not only one of the software protection technologies should be applied. Multiple encryption and verification technologies should be combined. A method that used the exe file and DLL file to protect each other and used the license file to check and provide software function was put forward. Tang Jiutao, Lin Guoyuan divided the forms of software protection into software-based encryption and hardware-based encryption. The two methods are usually used together with no
2. ALGORITHM IDEA AND MODULE DESIGN

2.1 Generate Registration Code

In consideration of the uniqueness of CPU serial number in computers, CPU serial number can be served as machine code. The user who needs software authorization should provide machine code to software provider, then software providers use RSA encryption algorithm to encrypt machine code and transform the encrypted results into the MD5 hash value. Because results have a fixed length after MD5 encryption, aesthetic and fixed length registration code can be generated through proper handling. The Registration code will be sent to the user so that the user can use the software through the registration verification. Encryption process is as follows:

In RSA encryption algorithm, it needs to find large prime numbers \( p \) and \( q \), calculate \( n =pq \), take a \( > 1 \) meeting \( \left( a,\phi \left( n \right) \right) = 1 \), then find \( d \) meeting \( da \equiv 1 \mod \phi \left( n \right) \). Encryption is obtained by \( E_a \equiv m^a \mod n \) while decryption is obtained by \( E_a^d \equiv m \mod n \). Here \( p, q, d, \phi \left( n \right) \) are all secret keys and should be remain private. In the software registration module, read the user's CPU serial number first. With a set of established secret keys, encrypt the machine code to get an encryption result \( E_a \). And the code with an arbitrary length can be operated into a fixed length value by MD5. Then turn the result \( E_a \) into MD5 hash value recorded as EEa, remove any four position value in EEa and insert “.” in the position NO.15, 10, 5 to get the final registration code. The specific implementation steps are as follows:

1) Generate large prime number \( p, q \). Big prime number can be generated through the Maple. In Maple, through the commands such as “ithprime”, “nextprime” and “prevprime”, prime number can be generated easily. For example, through the Function (1) a large prime number more than 100 bit can be generated;

\[
p = \text{nextprime}(123!)
\]  

(1)

2) Calculate \( n = pq \), \( t = p-1 \cdot q-1 \);

3) Select \( a > 1 \), such as take 67, to calculate \( t \mod 67 \), if not zero, then \( a = 67 \);

4) Calculate \( d \equiv a^{\frac{1}{\phi(n)}} \mod \phi(n) \) so to get a group secret keys;

5) In order to use c# depict secret key conveniently, first the key should be
convert into hexadecimal form. In the system, with byte[] type initialization, .NET 4.0 provides the BigInteger class to depict any big signed integer [2] and byte[] type can be forced into BigInteger type;

6) Read the CPU serial number and turn it into byte sequence (byte[]), then convert to BigInteger type as data m which needs to encrypt;

7) Use the selected secret key to encrypt m, then get Ea;

8) Convert Ea to MD5 hash value EEa;

9) Remove any four position values in EEa, and insert "." in its 15, 10, 5 locations to get the final registration code.

2.2 The Design of the Software Registration Module

Taking into account the system's ease of maintenance and scalability, let the software registration module be a project alone in Visual Studio 2010 programming environment. A class that inherits from the Installer is added to the project. Override the Install method and a form will appear during its execution. Users must enter a registration number to continue. The authentication of the registration number will execute in the form. So the authentication and authorization can be performed during the software installation process.

The registration number must be entered during the software installation process. The software cannot be installed, unless you enter the correct registration number. After successful validation, registration number will be written in a configuration file in the software installation directory. The registration number in the configuration file will be checked and verified before the system runs in order to prevent users from skipping the installing and copying the software to use directly. The machine code will be given and the software supplier has the program to turn the machine code into the registration number. It is convenient and effective to provide authorization and protect the copyright of the software.

3. The Design and Implementation of the Module Program

Based on .NET Framework 4.0 the software registration program is realized by C# programming language in the Visual Studio 2010 development environment. It can be used for authentication of the registration number. Using the same method the program of generating registration number is written in another solution.

![Installation Complete](image1)

![Write configuration file](image2)

![Initialize system](image3)

![Write configuration file](image4)

![Register system](image5)

![Authenticate registration number](image6)

![Is registration number correct?](image7)

![Read configuration file](image8)

![Register system](image9)

![Is registration number correct?](image10)

Fig. 1: Install or run the system

3.1 Machine code obtainment

In .NET Framework, the System.Management namespace provides access to a rich set of management information about the system, devices and applications. By using the namespace, it’s easy to get the CPU serial number, which can be treated as the machine code.
3.2 Realization of Encryption

Public and private keys can be designed upfront on Maple and initialized in the program by C#. Using ModPow() method of the class BigInteger can realize encryption. The result returned is a BigInteger and it can be transformed into a byte[] by the method ToByteArray(). So the method ComputeHash of the class MD5CryptoServiceProvider can be used to encrypt the result to its MD5 hash value. Finally, appropriately adjust the MD5 hash value to the formation of the "*****-*****-*****-*****" form. It is the ultimate registration number. During the registration and authentication, compare the registration number with the correct one to determine whether the software can be registered or has been registered.

In the key initialization, p and q are all more than one hundred digits large prime numbers. Initialize the hexadecimal numbers to byte[] type and then convert them into BigInteger type. Calculate their product n. a should also be initialized to the BigInteger type.

For example data is the machine code that will be encrypted. Encode data into a sequence of bytes and then convert it into BigInteger type. Encrypt it and get the value after encryption. The crucial code of the encryption is as follows:

```csharp
byte[] Endata_byte;
BigInteger rsa_data = new BigInteger(data);
BigInteger Endata_bigint = BigInteger.ModPow(rsa_a, rsa_data, rsa_n);
Endata_byte = Endata_bigint.ToByteArray();
```

Encrypt the result to its MD5 hash value and then turn it into the form of ultimate registration. encpubyte is the result of the RSA encryption and regstr is the ultimate registration number. The crucial code is as follows:

```csharp
MD5CryptoServiceProvider MD5CSP1 = new MD5CryptoServiceProvider();
byte[] registBytestr = MD5CSP1.ComputeHash(encpubyte);

StringBuilder md5str = new StringBuilder();
md5str.Append(BitConverter.ToString(registBytestr)).Replace("-", "");
md5str.Remove(6, 3).Remove(8, 3).Remove(13, 3).Remove(20, 3);
regstr = md5str.Insert(15, ".").Insert(10, ".").Insert(5, ".").ToString().ToUpper();
```

4. RESULTS AND ANALYSIS

The encryption method is tested on about 50 computers including single- and dual-core CPU computer, and the computers with different operating systems including Windows xp, Windows 7, Windows Server 2003. The result conforms to the idea of "one machine one code".

For example, we tested a computer with Dual-core CPU and Windows7 operating system. Its CPU serial number was “BFEBFBFF0001067A” and its registration number should be “C3A0E-5A3DB-553C7-17C64” (Fig.2).

![Fig. 2: Acquire machine code and generate registration number](image-url)
The computer's CPU serial number is unique and for a given machine code, there's only one registration number. The generated registration numbers are aesthetic, safe and stable. This registration method has been used in a mine information management system. A few computers have been registered without any safety problem. It is a little slow to read the CPU serial number. The serial number of other hardware devices can also be treated as the machine code. If other length of the registration number is needed, the result of the RSA encryption can be dealt with in other ways and generate the registration numbers to meet the demand.

5. CONCLUSION

Based upon RSA encryption algorithm and MD5 hash function, this paper designs a software license scheme which adopts the idea of “one machine one code”. Using c# language in the.net Framework 4.0 platform, the author completed the license scheme and ran on mining management information system successfully. It not only ensures the safety of the software and convenience in registration authorization, but also makes registration code more aesthetic and effective.

REFERENCES: