Apparatus and Method For Serializing and Validating Copies of Computer Software
Presented by Zhang Yeliang

Peter R. Samson
October 18, 1999

1 Introduction

This paper presents an apparatus and method of disabling an unauthorized copy of a computer program. A set of numbers with an uncommon mathematical property is defined. Before a copy of the computer program is attempted to run, first the computer system will check the number at a pre-defined location. Then to determine if the number has that uncommon mathematical property. The program will be disable and exit if that location has not the uncommon mathematical property.

2 Background of the Invention

Computer is ideal for its versatility, it can be programmed to perform various functions. This ability stems from the fact that a given computer can run a wide range of different types of software applications.

- In order to commercially distribute the program, typically involves expending a great deal of time, effort, and resources to write the lines of “code” for achieving the desired results.

- Most computer program involve a great deal of creativity.

- Before a computer program is marketed, the program is subjected to extensive “debugging” to ensure that it operates properly under almost every scenario.

Compare to the extensively effort for programming, it relatively cheap to duplicate copies from the original finished products. It is ordinarily a two step process. First the computer program is down-loaded onto a computer system, usually by reading it from a magnetic disk. Next, the program is written onto
a different disk, thereby creating a duplicate copy. But the typical prior art software protection mechanisms is not very ideal, because it is:

- Easy to crack, pirates find ways to bypass the protection mechanisms.
- Third parties will disclose the protection mechanism to others.
- Hard to verify the number of duplicates made by the third parties.
- Fail to provide for contingencies, wherein multiple terminals on a computer network communicate with one another to accomplish a specific task by running the same computer program.
- Once the pirate detect the serial number of the prior art method, he can replace it with a totally spurious serial number, as a result, the counterfeit copies cannot be tracked back by their spurious serial number.

3 Objects of this Invention

- One object of the present invention is to disable computer programs which were illicitly copied.
- Another object of the present invention is to allow a particular copy of a computer program to be traced to its source.
- Another object of the present invention is to allow computer programs to be duplicated anywhere in the distribution chain.
- Another object of the present invention is to verify the number of duplicates made by third parties.
- Another object of the present invention is to limit the number of programs that can be simultaneously run on a computer network.

A dedicated computer program, which is kept-in-house and proprietary by the developer, is used to generate a particular set of license numbers. These license numbers have the uncommon mathematical property. Without knowing the generating scheme, it will be extremely difficult for unauthorized parties to generate these numbers.

4 Detailed Description of the Invention

As we said in above, this invention uses some numbers with uncommon mathematical property embedded in the program. This number can be tested if it has the uncommon mathematical property but without knowing the generating scheme, it will be extremely difficult to generate the numbers. Therefore, the
embedded software is protected. \textit{CR}_L The currently preferred embodiment of the present invention involves the serialization, personalization and validation of a computer program. Serialization is the process of assigning a distinct identification number to each physical copy of the computer program. Through this serial number, we derive the license number created with the knowledge of a proprietary number \( x \). Keeping \( x \) proprietary is to make it extremely difficult for unauthorized parties from creating valid license number. License numbers are based on a mathematical “one-way trapdoor” function. \textit{CR}_L How to generate a valid license number:

- Select a prime number, \( q \), such that \( q - 1 \) has few small factors.
- Select three numbers \( a, b, \) and \( x \) which are all less than \( q \). The \( x \) shall be kept confidentially since it is used to generate future license numbers.
- Compute
  \[
  y = a^x \mod q.
  \tag{1}
  \]
- Set \( i \) equal to the 11 digits serial number.
- Compute
  \[
  m = i \times b \mod q.
  \tag{2}
  \]
- Select a number \( k \) with the property that \( \gcd(k, q - 1) = 1 \).
- Compute
  \[
  r = a^k \mod q.
  \tag{3}
  \]
- Let
  \[
  m = x \times r + k \times s \mod (q - 1),
  \tag{4}
  \]
  then the value of \( s \) can be determined.
- The license number is equivalent to \( m, r, \) and \( s \).

\textit{CR}_L The currently preferred embodiment also includes a personalization process. It should be noted that personalization is an added feature and is not critical to the implementation of the present invention. \textit{CR}_L

Each copy of computer program includes a verification procedure:

- The license number is read from the computer program.
- Check if
  \[
  m = (i \times b) \mod q.
  \tag{5}
  \]
  If not equal, an error message occurs and execution of the computer program is halted.
• Check if

\[ a^m = y^r \ast r^n \mod q. \] (6)

If not equal, an error message occurs and execution of the computer program is halted.

• If above tests are finished successfully, then the computer program begin to run.

Note that even an unauthorized party can determine how to verify the license numbers by examining the computer program, he still can not generate other valid license number. The reason is because the parameter is not presented in the code. In network environment, the verification procedure will be a little bit different. In addition to verify the running program’s own license number, it also looks for other running copies of itself on the network. If one is found, the program presents its own license number to the other program to verify. Next, the license numbers of the two programs are compared. If the programs have identical license number, it indicates either one of the programs is illicit, the error message generated and the execution of program is halted. In an alternative embodiment, the network can have up to \( z \) different license numbers incorporated with a single copy. If during network connection one of the license numbers is rejected, the other license numbers can be successively verified until one is accepted. In the end alternative embodiment, this method can be used with hardware implementation “dongles”. Dongle is a physical device which is provided along with the software. It plugs in the back of the computer. Each time the software is initialized, it interrogates that port to verify that its proprietary dongle is plugged in. We can use this invention here. When the dongle is interrogated, the computer program also check to determine whether the dongle contains a valid authorization code.

5 Duplication Process and License Number Generating

A master copy of the licensed software and a set of valid license numbers are provided to the third parties, distributor or even an end user. A licensing program is also provided to embed the license numbers in the duplicate.

In the license generating program:

• Read data from computer system.

• The program prompts for the user’s name and how many license numbers are to be generated.

• A pseudo-random number is selected.
6 Summary

This is a very easy paper with a very clever method to implement the license number to protect the software from pirating. It doesn’t say very clear on some details such as how to get the prime number q, how to validate the license number etc.