(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 April 2006 (06.04.2006)

PCT

(10) International Publication Number
WO 2006/035227 A2

(51) International Patent Classification:
G06F 1/00 (2006.01)

(21) International Application Number:
PCT/GB2005/003735

(22) International Filing Date:
29 September 2005 (29.09.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0421774.1 30 September 2004 (30.09.2004) GB

(71) Applicant *(for all designated States except US)*: **TTP-COM LIMITED** [GB/GB]; Melbourn Science Park, Cambridge Road, Melbourn, Royston, Hertfordshire SG8 6HQ (GB).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **MERSH, John, David** [GB/GB]; 21 Willow Way, Bottisham CB5 9BS (GB).

(74) Agents: **HOGG, Jeffery, Keith** et al.; Withers & Rogers LLP, Goldings House, 2 Hays Lane, London SE1 2HW (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 2006/035227 A2

(54) Title: SOURCE CODE PROTECTION

(57) Abstract: A method comprising encrypting an original plain text file and making it available to a user as a protected file (101), and issuing to said user a user program and a user licence (103) to enable said user to decrypt the protected file (101) and view an image of the original file whilst preventing the image of the original file from being copied to any file, other than as a further protected file. The image is preferably stored in a memory not backed up to the computer swap file. Preferably, the user program comprises an editor program and the user saves editorial changes to the original image in an encrypted difference file (208), separate from the original file. Both files are then used to re-create the edited image using the editor program and user licence (103). The user program may comprise any computer tool including compilers. Alternatively, the user program comprises a special editor program incorporating an obfuscator (302) which generates obfuscated code from. the image generated by the editor program, so that this is only intelligible to a compiler (305) or similar tool capable of converting the obfuscated code to an object file (306).

Source Code Protection

Technical Field

This invention relates to a method of protecting an original plain text file, and a computer tool to access an original plain text file that has been protected.

The invention is concerned with plain text file, especially computer source code, and their distribution to customers. The reason why software developers supply products in the form of source code, for example, in a high-level language such as C or C++, is that this can be readily adapted by their customers to suit their particular applications. However, in distributing source code in this manner, the developer loses control of subsequent use including copying by their customers and third parties obtaining the code from their customers, and the developer is then wholly dependent on legal enforcement of contracts and licences to protect their intellectual property rights.

Disclosure of the Invention

An object of the invention is to provide a method and means whereby a plain text file can be protected against illegal copying or use once it has been distributed to an authorised user.

This is achieved according to the invention by providing a method comprising encrypting an original plain text file and making it available to a user as a protected file, and issuing to said user a user program and a user licence to enable said user to decrypt the protected file and view an image of the original file whilst preventing the image of the original file from being copied to any file, other than as a further protected file.

Thus access to the plain text file is granted by a combination of a user licence and a special user program which makes use of the user licence and protected file to re-create, in the memory of a computer, an image of the original file, which can be displayed and edited. This image is preferably protected against copying by arranging that the memory in which it is stored is not backed up to the computer swap file so that it cannot be found by programs that might read the image file.

Preferably, the user program comprises an editor program and the user uses the editor program to edit the image of the original file and saves the changes to the original image in an encrypted form in a difference file, separate from the original file. The user re-creates the edited image of the original file from the protected file and the difference file using the editor program and user licence. If desired, the user can re-edit the edited image and then save the changes to the re-edited image in a second difference file, which is encrypted and is kept separate from the original file and the difference file. Subsequently, the user uses the editor program and user licence to re-create the most recent edited image from the original file and as many difference files as are involved.

The advantage of storing changes to the original file as a difference file or multiple difference files is that this is an auditable arrangement in which the ownership of each file can be readily identified as with the originator of each, and this identification is not lost with successive edit processes.

Also, if each of the difference files is encrypted using the same or a separate licence to that of the protected file, then each of these is similarly protected.

The user program preferably has only limited copying capabilities that allow data to be copied only into other protected files, a typical data copying capability being known as "cut and paste".

Portions of the original file may be marked as non-editable or invisible so that they can never be removed by the editor program, and thus will always be present to allow identification of the original file for licensing or other purposes or to restrict the use of the original file to that defined by the user licence.

The user program may comprise any computer tool needed by a user to access the original plain text, and includes compilers, version or configuration management tools and source level debuggers. Where multiple tools are required to access plain text source code and to

generate executable code from it, each of the tools will require the protection features of the invention.

However, in an alternative embodiment of the invention, the user program comprises a special editor program incorporating an obfuscator which generates obfuscated code from the image generated by the editor program, thereby preventing access to the source code other than by a compiler or similar tool which is capable of converting the obfuscated code to intelligible object code which is identical to that which would have been generated if the compiler had had access to the original source code.

Thus, according to another aspect, the invention consists in a computer tool for a user to access an original plain text file which has been protected by being encrypted in a protected file, the tool being adapted to decrypt the protected file once authorised by a user licence issued by an authority responsible for the protected file so as to produce an image of the original plain text file whilst preventing the image of the original file from being copied to any file, other than as a further protected file.

The tool preferably comprises an editor program and may be a special editor program including an obfuscator as already described above.

Description of the Drawings

The invention will now be described by way of example with reference to the accompanying drawings:

Figure 1 is a block diagram illustrating steps in a method according to the invention for allowing a user access to source code in a protected file including any changes to the source code;

Figure 2 is a block diagram illustrating steps in a method according to the invention for allowing a user to access source code in a protected file and to save changes made to an image of the source code in a difference file, separate from the protected file; and

Figure 3 is a block diagram illustrating further steps in the method of Figure 1 for allowing a user access to source code including obfuscation of the source code and passing of the obfuscated file to a compiler to produce an object file.

Embodiments of the Invention

Referring to Figure 1, the protected file 101 consists of a plain text file, such as source code, which is encrypted using a fast symmetric key algorithm, with the key stored at the beginning of the file in an encrypted form using public key cryptography. The whole file 101 is then protected using a digital signature algorithm.

The protected file 101 is distributed by the source code owner to a user, together with a user licence 103, which incorporates the key used in the public key cryptography to protect the file 101. The licence 103 is distributed using either the public key infrastructure PKI or a similar certificate-based mechanism.

The user receiving the protected file 101 and user licence 103, is provided with a special computer tool to access the source code in the protected file.

The tool first checks its own validity by checking a digital signature which is stored within its own executable file to ensure it has not been modified. It then opens the licence 103 and determines if it is entitled to run.

If the tool is entitled to run, then it opens the protected file 101. It then uses two decryption engines (102, 105) to generate data streams which represent the decrypted contents of the protected file and licence. These streams are then fed to a reconstruction engine 106 which combines the two streams to generate a human readable image 107. This image is held in the memory of a computer which is not backed up to the computer swap file so that the image cannot be found by other programs on the computer.

The human readable image 107 contains all of the source code lines along with flags indicating whether lines are invisible or non-editable.

The computer tool includes an editor that allows the image 107 to be edited, and this is illustrated in Figure 2 in which an edited image 201 is generated from the original image 107. The tool then allows the user to save the edited image 201 in terms of the differences compared with the original image 107. This is accomplished by a differences engine 206 which compares the images 107 and 201 and determines a set of differences which will subsequently allow the edited image 201 to be created from the original image 107.

This set of differences is then passed to an encryption engine 207 which uses a public key pair from the licence 103 to create a difference file 208. This public key pair used to encrypt the difference file is preferably different to the public key pair used to encrypt the protected file 101.

The difference file 208 is separate from but associated with the protected file 101 for use in creating the edited image, thereafter.

If, as shown in Figure 1, the difference file 208 is associated with the protected file 101 when it is accessed, then the reconstruction engine 106 receives a series of further instructions from the difference file 208, and lines of instructions are copied from either the protected file 101 or the difference file 208 as appropriate, to create the edited image 201 in place of the original image 107.

Figure 3 shows the process of compiling a protected source file using a source code obfuscator 302.

The first operation performed by the obfuscator 302 is to use the mechanism described in Figure 1 to build the human readable image 107 based on the licence 103.

Once the image 107 is available, then the obfuscator 302 can process this to generate the obfuscated source file 304. The process of obfuscation involves the removal of all human intelligible information from the file and is a well known technique (see for example Collberg et. al. US Patent 6668325). Obfuscation typically involves the removal of all comments from the source code; replacement of human-meaningful variable names with

randomly-selected names and; modification of formatting to make the code difficult for humans to read.

Once the obfuscated source file 304 is available it can be read by the conventional compiler 305 which will produce an object file 306 identical to that which would have been produced by compiling the human readable image 107.

Claims

1. A method of protecting an original plain text file which comprises the steps of :
   a) encrypting the original file and making it available to a user as a protected file (101); and
   b) issuing to said user a user program and a user license (103) to enable said user to decrypt the protected file and view an image of the original file whilst preventing the image of the original file from being copied to any file, other than as a further protected file.

2. A method as claimed in claim 1 in which the user program comprises an editor program and in which the user uses the editor program to edit the image of the original file and then saves the changes made to the image of the original file in an encrypted form, separate from the original file.

3. A method as claimed in claim 2 in which changes to the image of the original file are saved by a difference engine which re-opens the protected file using said licence (103) and compares the image of the original file with the edited image to produce a difference file (208) which is saved.

4. A method as claimed in claim 2 or 3 in which said changes are encrypted by the editor program using said user licence or a different licence key.

5. A method as claimed in any one of claim 2 to 4 in which said changes are stored in said protected file (101) or in a difference file (208) related to the protected file.

6. A method as claimed in any one of claims 2 to 5 in which the user creates the edited image of the original file from the protected file (101) and the difference file (208) using the editor program and user licence (103).

7. A method as claimed in claim 6 in which the user re-edits the edited image and then saves the changes to the re-edited image in a second difference file, which is encrypted and is kept separate from the original file and the difference file.

8. A method as claimed in any one of claims 2 to 7 in which parts of the original file are marked as non-editable, and the editor program prevents such parts being edited so that they will always be present in any image created from the original file and any difference file or files.

9. A method as claimed in any one of claims 2 to 8 in which parts of the original file are marked as invisible, and the editor program prevents such parts from being displayed in any image created from the original file and any difference file or files.

10. A method as claimed in any one of the preceding claims in which the user program comprises an obfuscator (302) that generates from the image of the original file an obfuscated output file which is intelligible to a specific software tool only.

11. A method as claimed in claim 10 in which the specific software tool is a compiler (305).

12. A method as claimed in any one of the preceding claims in which the original plain text file comprises source code.

13. A computer tool for a user to access an original plain text file which has been protected by being encrypted in a protected file (101), the tool being adapted to decrypt the protected file once authorised by a user licence (103) issued by an authority responsible for the protected file so as to produce an image of the original plain text file whilst preventing the image of the original file from being copied to any file, other than as a further protected file.

14. A tool as claimed in claim 13 which comprises an editor program that edits the image of the original file and then saves the changes made to the image of the original file in an encrypted form, separate from the original file.

15. A tool as claimed in claim 14 which comprises a difference engine which re-opens the protected file using said licence and compares the image of the original file with the edited image to produce a difference file (208) which is saved.

16. A tool as claimed in claim 14 or 15 which encrypts said changes using said user licence (103) or a different licence key.

17. A tool as claimed in any one of claim 14 to 16 which stores said changes in said protected file (101) or in a difference file (208) related to the protected file.

18. A tool as claimed in any one of claims 14 to 17 in which creates the edited image of the original file from the protected file and the difference file using the editor program and user licence.

19. A tool as claimed in claim 18 which re-edits the edited image and then saves the changes made to the re-edited image in a second difference file, the tool encrypting the re-edited second difference file and keeping this separate from the original file and the difference file.

20. A tool as claimed in any one of claims 14 to 19 in which parts of the original file are marked as non-editable, and the editor program prevents such parts being edited so that they will always be present in any image created from the original file and any difference file or files.

21. A tool as claimed in any one of claims 14 to 20 in which parts of the original file are marked as invisible, and the editor program prevents such parts from being displayed in any image created from the original file and any difference file or files.

22. A tool as claimed in any one of the preceding claims in which the user program comprises an obfuscator (302) that generates from the image of the original file an obfuscated output file which is intelligible to a specific software tool only.

23. A method as claimed in claim 22 in which the specific software tool is a compiler (305).

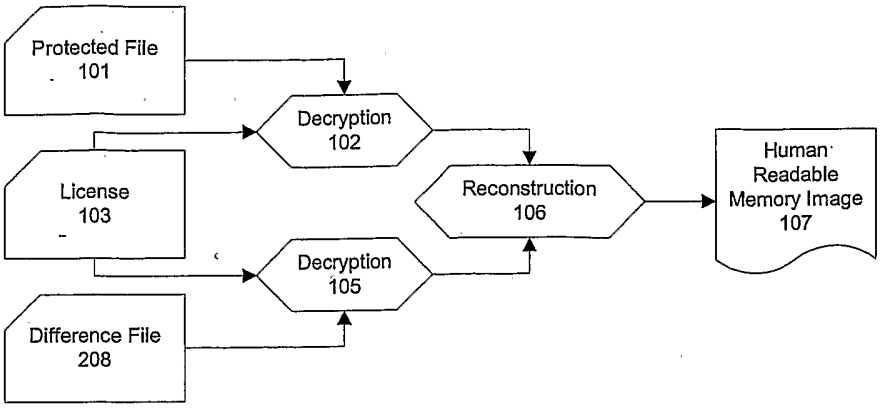

Figure 1

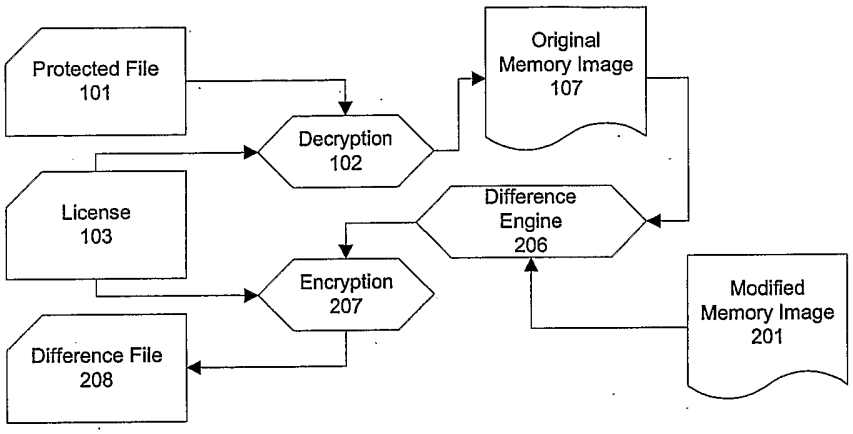

Figure 2

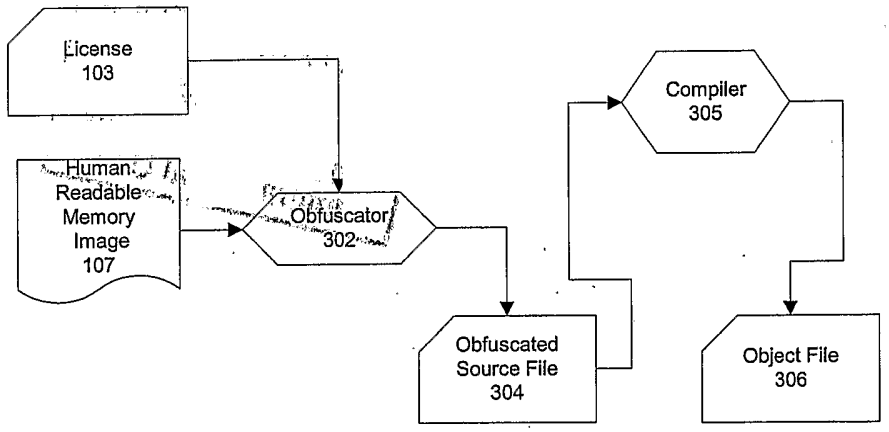

Figure 3