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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 September 2007 (27.09.2007)

PCT

(10) International Publication Number
WO 2007/107905 A2

(51) International Patent Classification:
G06F 21/22 (2006.01)

(21) International Application Number:
PCT/IB2007/050796

(22) International Filing Date: 9 March 2007 (09.03.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
06111353.6 17 March 2006 (17.03.2006) EP

(71) Applicant *(for all designated States except US)*: **KONINKLIJKE PHILIPS ELECTRONICS N.V.** [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **MICHIELS, Wilhelmus, P., A., J.** [NL/NL]; c/o Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). **ESSER, Norbert, C.** [US/US]; c/o Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). **GORISSEN, Paulus, M., H., M., A.** [NL/NL]; c/o Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(74) Agents: **GROENENDAAL, Antonius, W., M.** et al.; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(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, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

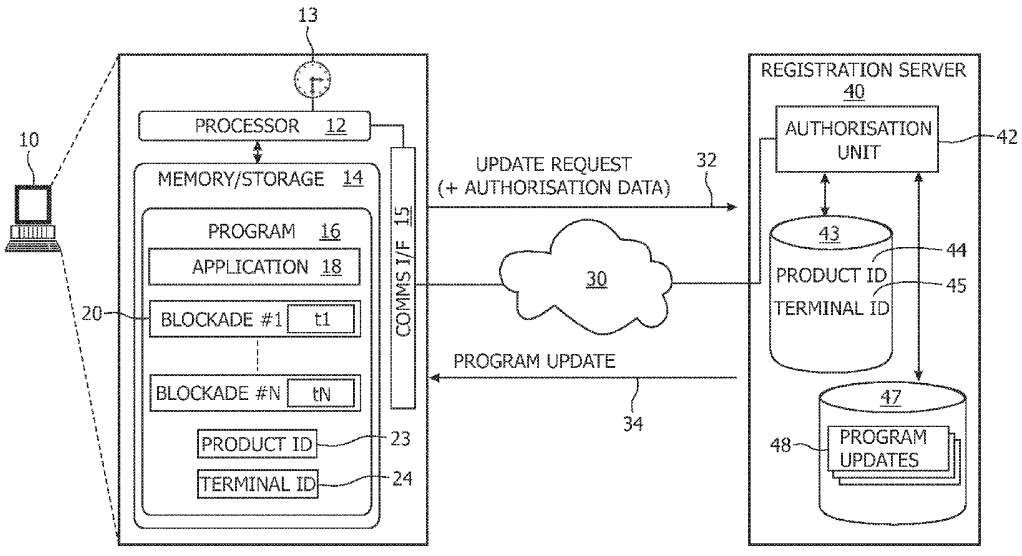
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, 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).

Declaration under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

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

[Continued on next page]

(54) **Title:** PROTECTION OF SOFTWARE FROM PIRACY



(57) **Abstract:** A method of protecting a computer program (16) running on a computer apparatus (10) from piracy. The computer program (16) includes a code module (18) which performs an application and code which defines a plurality of blockades (20). Each blockade is associated with a parameter which determines activation of the blockade. Once activated, each blockade changes the functionality of the application compared to that pertaining prior to activation of the blockade. Blockades activate at different times. Each blockade requires at least one program update to be executed to deactivate that blockade. Updates can be installed manually or automatically. Preferably, the computer program is authorised as being a genuine copy of the program before receiving a program update.

WO 2007/107905 A2

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.

Protection of software from piracy

This invention relates to protection of software products.

The computer industry loses a significant amount of money to software piracy. In software piracy a so-called 'pirate' obtains a piece of software, copies the software, and then distributes illegal copies of the software for profit. Where software includes some form of copy protection, a pirate may take steps to remove, or disable, the copy protection before making illegal copies and distributing them.

Various forms of security measures are known which attempt to protect software from piracy. In client-server systems it is possible to arrange for part of the software code to be executed by a client and part of the software code to be executed by a secure remote server. In this type of system checks can regularly occur between the client and server and in the event of a security concern with a particular client, the server can stop executing code for that client. While this is a secure solution it has a disadvantage of requiring the software supplier to provide processing capability at the server for each software user.

A majority of software is executed in a stand-alone manner by terminals. Although the terminal which executes the software can have access to an external network, such as the Internet, the terminal does not require a continuous connection to a remote server for the software to function. It is this kind of set up where it is most difficult to protect software.

It is known to provide software which requires a user to enter a security 'key' (typically a code supplied with the product) at the time of installing software. The software product is only unlocked when the correct key is given. However, this can be bypassed by ensuring that any illegal copies of a software product are accompanied with this key. An additional security measure is to ask a user to register their product with a registration authority. A user can register their product by sending details by mail, phone, fax or, more typically, by allowing their terminal to electronically exchange data with a remote server. The registration process sends a key which uniquely identifies the copy of the installed

product, in an attempt to prevent multiple registrations of the same software product installed on different terminals. The registration process may also collect details of the hardware configuration at the terminal where the software has been installed. US 6,243,468 describes a system of this kind. Registration is often not compulsory, and so it is still possible to use pirate software without registration.

It is known to provide software on a trial basis with a time-dependent blockade. After using the program for a particular time (e.g. operating the program ten times), the user is required to purchase a key to continue using the product. If the correct key is not provided, the program is rendered unusable. US 5,014,234 describes software of this kind.

In spite of these various security measures, software piracy remains a significant problem to the software industry.

The present invention seeks to discourage software piracy.

Accordingly, a first aspect of the present invention provides a method of protecting a computer program running on a computer apparatus from piracy comprising, at the computer apparatus:

executing at least one code module of the computer program to perform an application;

executing code of the computer program which defines a plurality of blockades, each blockade being associated with a parameter which determines activation of the blockade, wherein each blockade, once activated, is arranged to change the functionality of the application compared to that pertaining prior to activation of the blockade, and wherein activation of a first blockade is arranged to occur at a different time to that of a second blockade;

receiving a program update, and

executing the program update to deactivate one of the blockades,

wherein each blockade requires at least one program update to be executed to deactivate that blockade.

Providing a plurality of blockades which activate at different times forces the computer apparatus which executes the program to install program updates at different points in time to deactivate the blockades and continue using the software. Preferably a blockade, once activated, is arranged to stop the application from operating, or to cause the application

to operate with reduced or incorrect functionality compared to that pertaining prior to activation of the blockade.

Pirates are discouraged from distributing illegal software as users of illegitimate software continue to be dependent on the pirate for updates. This increases the operating costs of the pirate and, because the pirate has to keep in contact with the customers, it increases the risk of the pirate being discovered. If a pirate chooses not to support users of the pirate software, the pirate copies of the software will terminate, or operate with reduced or incorrect functionality, when the first blockade is reached.

In addition to deactivating a blockade, the program updates can include code to fix a breach of a security mechanism which may have occurred since release of the computer program. Program updates can additionally, or alternatively, include code to implement additional blockades and code to fix other bugs which have been discovered in the program. By incorporating the blockades, a user is required to install a program update if they are to continue using the program with at least the functionality pertaining originally.

The program updates can be installed manually by a user or, more preferably, the updates can be installed automatically before a blockade becomes active. This can occur as a background process which does not interrupt the normal operation of the program. The updates can be retrieved by a user or automatically by the program. In all cases, it is preferred that the copy of the computer program is verified as being an authorised copy before a blockade is deactivated.

A program update can be installed before the activation time of a blockade. This will have the effect of deactivating the blockade and will ensure that the effects which are scheduled to occur at the time of activation (application stops operating; application operates with reduced or incorrect functionality) do not occur. However, it is preferred that the update can only be installed during a limited window of time preceding the activation time of the blockade. This prevents pirates from supplying a copy of the program with updates already installed to deactivate the blockades.

Legitimate users do not need to be inconvenienced by the blockades as the software can automatically contact the server to deactivate the blockade at an appropriate time.

Preferably, the code which implements the blockades is buried within the code module (or modules) of the application to prevent the blockade code from being readily identified and altered.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.