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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 April 2006 (20.04.2006)

PCT

(10) International Publication Number
WO 2006/040757 A1

(51) International Patent Classification:
G06F 9/45 (2006.01) *G06F 11/30* (2006.01)

(21) International Application Number:
PCT/IL2005/001058

(22) International Filing Date: 2 October 2005 (02.10.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
164571 14 October 2004 (14.10.2004) IL

(71) Applicants and
(72) Inventors: **BROSHY, Yuval** [IL/IL]; Yosef Haglili 9, 52416 Ramat-Gan (IL). **ROSENTHAL, Dani** [IL/IL]; David Elazar 22, 43205 Raanana (IL). **FELDMAN, Ofer** [IL/IL]; Teena 22, 99797 karme Yosef (IL).

(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.

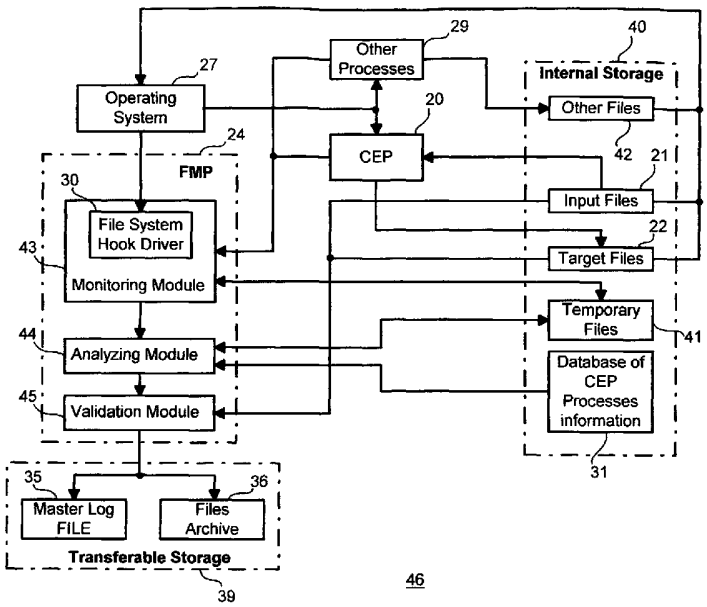
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT,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

Declarations under Rule 4.17:
— *as to the identity of the inventor (Rule 4.17(i))*
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*
— *of inventorship (Rule 4.17(iv))*

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

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.

(54) **Title:** A SYSTEM AND METHOD FOR AUTHENTICATING AND VALIDATING THE LINKAGE BETWEEN INPUT FILES AND OUTPUT FILES IN A COMPUTATIONAL PROCESS



(57) **Abstract:** Provided is an automatic solution for authenticating and validating the unique linkage between input files and output target files in computational processes, regardless of their structure and regardless of the software used to process them. In particularly it relates to automatic authentication and validation of the linkage between source code files and resulted executable files in compilation linkage processes used for software creation.

WO 2006/040757 A1

A system and method for authenticating and validating the linkage between input files and output files in a computational process

Background of the invention

Field of the invention

The present invention relates to the field of authentication and validation of input files and their unique involvement in the creation of output files by computational processes. In particularly it relates to automatic authentication and validation of the linkage between source code files and executable files in compilation processes during creation of software.

Art Background

In computational processes an input file is converted and manipulated by software program according to internal programmed instructions and the results are written to an output target file that is saved for later uses. It is a known fact among software professionals, that it is very difficult to know by looking at a data input file and a random target file whether that input file has actually been used for creating that target file and if so – whether it was the only input file used. This can be done in some simple cases if the examiner has prior technical information on how the software operates and the files are structured in a comprehensible way, but in most cases it is a tormenting task. The task approaches impossibility when the software operation involves modes of artificial intelligence, optimization, translation and encryption, as is the case in communication software, database software, compilers etc.

The compiler case is especially noticeable and important because of its unique implications on the software industry. The input source files are written in human-readable form, commonly referred to as source code, and are translated by special software, called compiler, into machine language files, commonly referred to as object code. Usually several related object code files are linked together by software, commonly referred to as linker, to form the final version of a new software package, commonly known as executables, that can be sold to customers. Through the process the ability to identify the originating source code by looking at the resulting object code or final software executables is lost and reverse engineering is close to impossible. It is known among professionals in the art that customers often require that copies of the source code and other related source files be kept in the hands of a trusted third party – for any eventuality that some calamity will happen to the software creator and he will be unable to support his software. That third party, known in the art as software-escrow agent, gets copies of the source code and of the object code or the

executable from the creator and places them in a safe place until a release event occurs as legally agreed between the parties. For the reasons explained hereinabove, the biggest problem facing the customer and the escrow agent is that they are totally depended on the software creator to provide the complete set of correct up-to-date source code files. The only practical way to check everything is to go to the software creator's office and monitor the software while it is being compiled and linked and immediately create copies of the source code files after the software has been finalized. Such practice is very expensive and labor intensive and is also subject to human errors and fraud. Another problem that faces the escrow arrangement users is that software developers are very reluctant to risk exposure of their technology by giving anybody, even a trusted entity, access to the source code files, thus increasing the uncertainty and unreliability of that approach.

In prior art, the only method available to alleviate some of the trust- or security concerns discussed hereinabove is encryption of the source code files either by the software developer or by the escrow agent to prevent technology leakage. Evidently this further hampers the customer or the escrow agent in ascertaining that all the right source files for the software are kept in escrow deposit.

It will be apparent that the same problem applies to any input file that is going through calculations by any program, or process, that outputs any target files.

The invention disclosed herein is of an automatic reliable solution for authenticating and validating the unique linkage between input files and output target files regardless of their structure and regardless of the software program used to process them.

Definitions

The following terms are used in the present application as defined hereinbelow.

Input file: Plurality of data- or source code files that contain information to be fed to the CEP computational process.

CEP: Compiling or Encoding Program; a computer program, embodying one or more software processes, used by the operator to perform computational process on input files, which results in an output target files.

Target file: Plurality of files which are the output result of the computational process done by the CEP.

FMP: File Monitoring Program, a computer program, which is part of the present invention, that monitors, authenticates and validates all activities of files and processes done by the computer at any given time during its operation.

Digital Signature: A mathematical method, well known in the art, that gives any given group of information bits a unique identifier, which enables detection of any change in the group.

Encryption: A mathematical method, well known in the art, that ciphers any message by using a cipher key.

Symmetrical key: A cipher key that can be used for both encryption and decryption of the message.

Public key and Private key: A ciphering method, well known in the art, that enables encryption of a message with publicly known public key and decryption of the message by only the holder of the matching private key.

AVP: Archive Verification Program; a computer program, which is part of the present invention, that is used to verify that a particular set of input files was used to create a given set of target files and that the whole archive was not tampered with.

Attributes: Any or all characteristics of a digital file that distinguishes it and can be used in identifying it. For example but not limited to, name, time, date, size, digital signature, storage location, structure, topology, content, its creating program, etc.

The acronyms given hereinabove were given for the sole purpose of text and description simplification, as such they should not imply of any other meaning.

Brief summary of the invention

According to the present invention, there is disclosed a system and method that monitors all activities on a computer in order to authenticate and validate the linkage between input and target files of any given software. First part of the present invention is of the monitoring program, FMP, that identifies each and every file operation and its activating process and registers it to a log file together with its attributes. Furthermore, said program makes a digital signature of each encountered file and registers it to a log file. When monitored processes have finished, the program analyzes the activities of all the relevant input and target files by checking which process used them, in any way whatsoever, according to a preset set of rules. However, some checks are done by the FMP concurrently with the main processes, i.e. "on the fly", in order to further ensure that no attempts are made to alter the files and

mislead the monitor. In addition, the program analyzes, according to a preset set of rules, whether the same processes also created the registered target files. That is done by comparing their attributes to a predefined set of attributes and conditions stored in the FMP's database. It should be noted that any deviation from the rules indicate an abnormal way in which the target files were created – for example, by illegal dummy process – with the intention of forging the target file. Such indication will cause the program to issue a fail warning. On the other hand, if everything was according to the rules, the program will archive all the input files, target files and log file, with their attributes, for safe keeping; this authenticates the linkage between the input files and target files, i.e. that they were all used and created under the watchful eye of the FMP in one session.

Verification of the archive is done, according to the second part of the present invention, by means of the AVP, which reads each file from the archive and compares its attributes and digital signature to that registered in the log file. Any discrepancies will show that an attempt was made to manipulate the archived files or that they were damaged during storage – either physically or by virus etc.

It will be appreciated that encryption can be used to enhance security of any or all the stages described hereinabove.

Brief description of the drawings

FIG. 1 is a schematic flow chart, showing a single input file being processed by the CEP, resulting in a single target file, according to prior art;

FIG. 2 is a schematic flow chart, showing a plurality of input files being processed by the CEP, resulting in target files, according to prior art;

FIG. 3A is a block diagram of a preferred embodiment of a system for creating target files through a CEP, including an FMP and files archive according to the present invention;

FIG. 3B is a block diagram of a preferred embodiment of an archived files verification system, including an AVP, according to the present invention;

FIG. 4 is a flow diagram of the monitoring, analyzing and validation processes for the system of Fig. 3A.;

FIG. 5 is similar to Fig. 4, but including encryption of all records and files;

FIG. 6 is similar to Fig. 5, but with the FMP being integrated with the CEP as one unified comprehensive software system;

FIG. 7 is a flow diagram of the files verification process of the AVP system of Fig. 3B; and

FIG. 8 is an illustration of the FMP or AVP results report after a compilation-linkage program run.

Detailed description of the invention

Described below, with reference to the block- and flow diagrams of Figures 1-8, are methods and preferred embodiments of systems according to the present invention, in various configurations. Generally, a system consists of two parts, one comprising an FMP and one – an AVP; in most cases the FMP part is used by the creator of the target files in one location and the AVP part is used by a verifying entity in another location.

Figure. 1 is a flow diagram . that illustrates a CEP software 2 manipulating the information given in a single input file 1 and outputting a single target file 3, according to prior art. The CEP may comprise a plurality of processes in sequence – for example a compiler, a linker, a ciphering program or any other computer program. However, there is no simple way for anyone who gets both the input file 1 and the target file 3 to tell that the latter is a direct result of the former being processed by the CEP. It may appear like the authentic result when looking at some file attributes, like file header, time stamp etc., but all these can be easily manipulated to mislead the examiner.

Referring now to Fig. 2, a flow diagram of prior art illustrating a situation similar to that of Fig. 1 but with a plurality of input files 10,11,12 and a plurality of target files 14,15. It will be apparent that the task of verifying that all target files are direct result of all the input files is practically impossible, because the input data are entirely altered by the CEP 13 and because data are divided and then integrated into multiple target files in an unknown manner.

Figure 3A is a block diagram, and Fig. 4 – a flow diagram, of a preferred embodiment of a first aspect of the invention in a first configuration, as will be explained in greater detail hereinbelow. Shown in Fig. 3A is a host computer system 46, having an operating system 27 and a CEP 20 for creating target files 22 from input files 21, and a File Monitoring Program (FMP) 24, creating a files archive 36 and a master log file 35. The host system has generally also other processes 29 running concurrently with the CEP and using other files

42. Shown in addition are temporary files 41 and database of CEP information 31; all files reside on an internal storage 40 within the host system – either locally or remotely via communication network. The FMP 24 comprises monitoring module 43 with its "file system hook driver" 30, analyzing module 44 and validation module 45. The resulting files archive 36 and master log file 35 are copied to a transferable storage 39 – e.g. tape, disc, DVD, removable memory device etc. – or sent via communicating network directly to a verifying entity. Said entity use the verification module (AVP) 119, shown in Figs. 3B and Fig. 7 and explained further below, to verify the integrity of the files.

The File Monitoring Program (FMP) 24 and the manner in which it authenticates and validates the input files and their linkage to the Compiling Encoding Program (CEP) will now be explained with reference to Figs. 3A and Fig. 4. The FMP monitors and analyzes all file related activities and processes done by the computer and subsequently authenticates the target files linkage to the input files. Illustrated is CEP 20 that has input of input files 21 and outputs plurality of target files 22. The CEP can be any computer program known in the art for example, compiler, linker, ciphering program, translation program, calculations program etc. The CEP is running under the control of an operating system 27, e.g. Windows, UNIX, Linux or any other system known in the art. Also, there are other unrelated processes 29 running concurrently on the same computer. In order to facilitate the monitoring and the authentication the operator first starts 25 the monitoring module 43 which activates a memory resident monitoring program 30, named "File system hook driver". Said resident monitoring program 30 constantly receives information from the operating system 27 on all file activities and processes running on the computer. It specifically but not exclusively receives file commands, like open, close, read, write, erase, copy and move, all – together with their requesting process attributes specifically but not limited to path, name, time. After the FMP has been started the "start CEP" command 26 is issued manually or automatically and the CEP 20 starts operation. The FMP constantly monitors 23 every activity of every file and every process in the computer and registers the information into temporary files 41. It also analyzes 28 each activity according to a set of rules and parameters, present in the Database of CEP Processes 31, which contains, *Inter alia,* all CEP valid attributes and processes. Furthermore, it marks every activity as approved or not approved. The analyzed results are now tested 32 for relevancy to the CEP operation; if it is not relevant, i.e. it is part of any other process running on the computer at the same time and the files used are not connected in any way to CEP activity, then the information is rejected 33 and erased from the temporary files. However, if said information is relevant to CEP activity, the FMP registers the active file attributes information into a master log file 35. At the validation stage

34 the FMP calculates for each of the input and target files its digital signature and sorts its relevant attributes before registering it into log file 35 . It will be noted that the attributes and digital signature of the target files can only be calculated and registered after they were finished and closed by the creating CEP; therefore the FMP also monitors them after the CEP has finished operation and registers them to the master log file 35 before terminating itself. Furthermore, before the FMP terminates itself, Analyzing Module 44 analyzes 28 the usage and way of creation of each registered file according to the information logged in the temporary files 41 and data from the database of CEP processes 31 for workflow consistency, which processes accessed it, when and for what operation. If the FMP reveals that processes other than those which are part of the CEP also accessed the registered files they are marked "unapproved" and they are shown in report 38 with list of all files and clear warnings that certain files were handled by other programs during or after there usage by the CEP. Stop command 37 is used to terminate the FMP monitoring operation either manually by the operator or automatically from within the program.

To further clarify, according to this invention, the FMP tests are done to prevent any distortion of the input files after they have been used and of the target files during and after their creation by the CEP. Any intervention by processes other than those of the CEP or the operating system are considered suspicious and treated as such. Final report 38, with or without warnings, based on the approval or disapproval of each action and each file, is issued after every run of the FMP for user reference and archive management. Said report contains listing of all input files that participated and list of all target files. It will be further appreciated that the prove of and validation of the linkage between input files and target files lies in the master log file 35, created during the monitoring process and containing exclusive digital signatures and attributes of all files. After receiving a clean report, which shows no indication of abnormalities in the registered input and target files, the input and target files can be archived 36 and together with the master log file 35 stored on any removable media 39 or transmitted via communication net.

According to additional feature of the invention, The analyzing- 28 and testing 32 procedures, described hereinabove can be executed concurrently with the operation of the CEP, i.e. "on the fly" or alternatively as post processing after all the information has been registered to temporary files and the CEP has finished. It is noted that in either option the FMP keeps its monitoring action until the last file has been validated and all the information has been registered to the master log file, in order to prevent any mishandling of the files.

Optionally, for saving processing time, the operator can point out the path to all the input files prior to CEP activation and the FMP will concurrently with its monitoring create digital signatures for all of them and register them directly to a temporary file. During the monitoring and analyzing processes the FMP will verify their usage and validity in same way described hereinabove.

It should be apparent to those skilled in the art that the usage of the system and method described hereinabove in Fig. 3A and Fig. 4 is suitable for an environment where there is no danger of intentional or unintentional interference or hackers attack. Otherwise, the registered information, target files and input files can be easily accessed and manipulated in ways that will render the whole authentication and validation process useless.

Referring now to Fig. 5, disclosed is an embodiment of the present invention in another configuration, comprising the FMP and an integrated two tiers encryption system, which prevents any intentional or unintentional interference with the authentication process or with the finished archive. According to an advantage of the present invention said archive can not be opened or verified by the CEP operator himself, but only by a second party holding a private key, thus allowing the latter not to be physically present in the same location where the target files are created and yet maintain absolute confidence in the results. As in the configuration of Figs. 3A and 4, the FMP program 53 monitors and analyzes all file related activities and processes done by the computer and subsequently authenticates the target files linkage to the input files. Illustrated is CEP 50 that has input from input files 51 and outputs plurality of target files 52. The CEP can be any computer program known in the art for example, compiler, linker, ciphering program, translation program, calculations program etc. The CEP is running under the control of an operating system 56 e.g. Windows, UNIX, Linux or any other system known in the art. Also, there are other unrelated processes 57 running concurrently on the same computer. In order to facilitate the monitoring and the authentication the operator first starts 54 the FMP monitoring module which activates a memory resident monitoring program 55 named "File system hook driver". It also creates an internal random symmetric ciphering key 62 and keeps it in a temporary memory location, not shown. Said resident monitoring program 55 constantly receives information from the operating system 56 on all file activities and processes running on the computer. It specifically but not exclusively receives file commands like open, close, read, write, erase, copy and move, all – together with their requesting process attributes specifically but not limited to path, name, time. After the FMP has been started the "start CEP" command 59 is issued manually or automatically and the CEP 50 starts operation. The FMP constantly monitors 60 every activity of every file and every process in the computer and encrypts the

information using symmetric key 62 and registers the results into temporary files, not shown. It also analyzes 58 each activity according to set of rules and parameters present in an encrypted database of CEP processes 61 which contains, *Inter alia,* all CEP valid attributes and processes. Furthermore, it marks every activity as approved or not approved – all encrypted with an internal hard coded key, not shown. The analyzed results are now tested 63 for relevancy to the CEP operation; if it is not relevant, i.e. it is part of any other process running on the computer at the same time and the files used are not connected in any way to CEP activity, then the information is rejected 64 and erased from the temporary files, not shown. However, if the information is relevant to CEP activity, the Validation Module 65 calculates for each of the input and target files its digital signature and sort its relevant attributes. It, furthermore, encrypts the file information, using internal key 62, and registers it into a master log file 66. It will be noted that the attributes and digital signature of the target files can only be calculated and registered after they were finished and closed by the creating CEP; therefore the FMP also monitors them after the CEP has finished operation and registers them to the master log file 66 before terminating itself. Furthermore, before the FMP terminating itself, it analyze 58 the usage and way of creation of each registered file according to the information logged in the temporary files (not shown) and data from the encrypted database of CEP processes 61 for workflow consistency, which processes accessed it, when and for what operation. If the FMP reveals that processes other than those which are part of the CEP also accessed the registered files they are marked "unapproved" and they are shown in report 73 with list of all files and warnings for these certain files that were handled by other programs during or after their usage by the CEP. Immediately after, the FMP validates each of the input files 51 and each of the target files 52 against their respective digital signature in the master log file 66 and encrypts them 67 together with the master log file 66 into a single archive file 68, using internal key 62. Thereafter, the FMP requests the operator for the second-party public key 70, which is a part of a verification keys set issued by the verifying authority. The public key is used to encrypt 69 the internal symmetric key 62 and output an encrypted key file 71, that will be given, together with the encrypted archive 68, to the verifying authority. Stop command 72 is used to terminate the FMP monitoring operation either manually by the operator or automatically from within the program. Final report 73 is created as explained for Fig 4 hereinabove.

Referring now to Fig. 6, according to still another configuration of an embodiment of the present invention, the FMP can be an internal part of any CEP, such as a compiler, a linker, a translation program etc., and operate as integral part thereof, in order to achieve maximum reliability of the authentication and validation. There is shown in Fig. 6 a CEP

software package 152 that includes a CEP module 153 and FMP modules. The operation is similar to that explained hereinabove with respect to Fig. 5, except for some minor changes, explained hereinafter. The operator first directs the CEP software to the input files 150 and inserts a verification public key 151, given to him by the verifying authority, then starts the system 156. During the operation stage, as described hereinabove the CEP, instead of directly creating the final target files, creates temporary target files 155 by its internal module 153. Only after said module has finished its activities and the FMP has finished creating the archive file, as explained hereinabove for Fig. 4 and 5, then the FMP copies the temporary target files 155 to final target files 157 for delivery to the final user. It should be noted that in this configuration of the invention all this is done within a single program (the FMP being an integral part of the CEP) resulting in security enhancement and in that real-time manipulation of processes and files is prevented to a higher degree.

According to yet another configuration of an embodiment of the invention the FMP can be integrated into the operating system itself and function in the same manner as explained hereinabove with respect to Fig 3A, Fig. 4, Fig. 5 and Fig. 6.

Referring now to Fig. 7 – a flow diagram, and Fig. 3B – a block diagram, of a preferred embodiment of the second aspect of the invention. They show a Verification Module which is the Archives Verification Program (AVP) 119 that is used by the verifying authority to verify that input files given in an archive were actually the files used by a certain known CEP to create those given archived target files. Furthermore, it is used to verify that the input files were not changed, tempered with, damaged, add to etc. The AVP 119 can handle archives like those resulted from any of the processes explained hereinabove with respect to Figs. 4, 5 and 6. The verification operator decrypts the key file 121, if any, by using his private key 120, and extracts a symmetric key, which is then fed to the AVP and starts the program 122. The AVP decrypts and extract 123 the master log file 125 and decrypts the archive file 124, extracting input files 126 and target files 130. All files 125, 126, and 130 are placed in a location within the computer's memory (not shown). Now the AVP performs verification 127 on each input file in the archive 126 by comparing each attribute registered in the master log file 125 to the actual attribute of the file in the archive. If a mismatch is encountered 128, the program registers the problematic file in a failure report 129. After all input files have been checked, the program checks in a similar way 131,132 and 133, the target files 130 in the archive. Before terminating, the AVP erases all decrypted files 134 from memory and issues a comprehensive result report 135.

It will be apparent that for unencrypted archives the process remains the same, except for the use of the keys and decryption stages.

Optionally in some cases the creator of the target files may give a copy of them directly to the customer, as, for example, in the case of the final software executables, while at the same time sending the archive file to a verifying authority. In such cases, the verifying authority can obtain the customer's copy and, by means of simple bit to bit comparison with the archived files, authenticate its origin.

In another configuration of the second aspect of the present invention, the AVP (verification module) can be integrated into the operating system itself and function in the same manner as explained hereinabove.

Referring now to Figure 8, illustrated, by way of example, a final report 180, (e.g. 38 in Fig. 4), for compiler monitoring. In said report are listed general developer information, pass/fail, extraordinary events that were detected during the run and all type of files used or created in a compiling process. Such a report gives the examiner complete picture of the participating files and their usage. A similar report can be obtained also from the AVP.

What is claimed is:

1. A system for authenticating and validating the linkage between input files and output target files of a given computational process, the system comprising:

a monitoring module, for monitoring and registering all file operations and active processes on the host computer;

an analyzing module, for analyzing the collected information relevancy and authenticating the linkage between the preferred computational process and active files; and

a validating module, for creating digital signature for each authenticated file, registering its attributes and storing said signatures and attributes, together with the input files and the output target files, in an archive.

2. The system according to claim 1, wherein the monitoring module continually receives information from the operating system regarding all current file operations on the computer.

3. The system according to claim 1, wherein the monitoring module continually receives information from the operating system regarding all current processes running on the computer.

4. The system according to claim 2 or 3, wherein the monitoring module has a memory resident driver.

5. The system according to claim 2 or 3, wherein the monitoring module resides in an auxiliary electronic circuitry.

6. The system according to claim 2 or 3, wherein the collected information is stored in plurality of temporary files.

7. The system according to claim 1, wherein said analyzing module comprises an analyzing method for analyzing the relevancy of the collected information.

8. The method according to claim 7, comprising the steps of:

retrieving file and processes information from the temporary files;

retrieving information from database of computational processes information;

matching for each file its activating process;

authenticating that the files were handled by the appropriate computational process; and

examining all the authenticated files for access by processes other then the monitored given computational process.

9. The method according to claim 7, further comprising the step of concurrently creating digital signature for said authenticated files.

10. The method according to claim 7, further comprising the step of concurrently saving digital signature and other file attributes to a file.

11. The system according to claim 1, wherein said validating module comprises a validation method for detecting illegal manipulation of the input and output target files.

12. The method according to claim 11, comprising the steps of:

retrieving information from temporary files;

creating file digital signature;

checking for each file its attributes and processes;

validating that the files were handled only by the appropriate computational process and were not changed after that; and

saving the digital signatures and other file attributes to a master log file.

13. The method according to claim 11, further comprising the step of issuing a warning report for any file accessed by processes other then the monitored given computational process.

14. The system according to claim 1, comprising an archiving process.

15. The system according to claim 14, wherein the input files, output target files and master log file are automatically archived together.

16. The system according to claim 1, 6, 10 or 14, comprising data encryption.

17. The system according to claim 16, wherein the encryption key is a random internally generated symmetric key.

18. The system according to claim 16, wherein the encryption key is encrypted by an externally imported public key for safekeeping.

19. The system according to claim 1, wherein said system is integrated into the given computational software.

20. The system according to claim 1, wherein said system is integrated into the computer operating system.

21. The system according to claim 1, wherein result reports are issued.

22. The system according to claim 21, wherein said reports contain lists of all authenticated participating files and their attributes.

23. The system according to claim 1, further comprising A verification method for verifying the consistency and validity of the archive.

24. The verification method according to claim 23, comprising the steps of:

providing an archive;

retrieving archived input files;

retrieving archived output target files;

retrieving archived master log file; and

comparing the attributes registered in the log file to the actual attributes of the retrieved files.

25. The method according to claim 23, further comprising the step of decrypting the archive files.

26. The method according to claim 23, further comprising the step of issuing results report.

27. The system according to claim 1, further comprising the variations described in the detailed description section of the present invention.

28. The system according to claim 1, further comprising the variations described in the drawings section of the present invention.

29. A method for authenticating and validating the linkage between input files and output target files of a given computational process, the method comprising:

monitoring and registering all file operations and active processes on the host computer;

analyzing the collected information for relevancy and authenticating the linkage between the preferred computational process and active files; and

creating digital signature for each authenticated file, registering its attributes and storing said signatures and attributes, together with the input files and the output target files, in an archive.

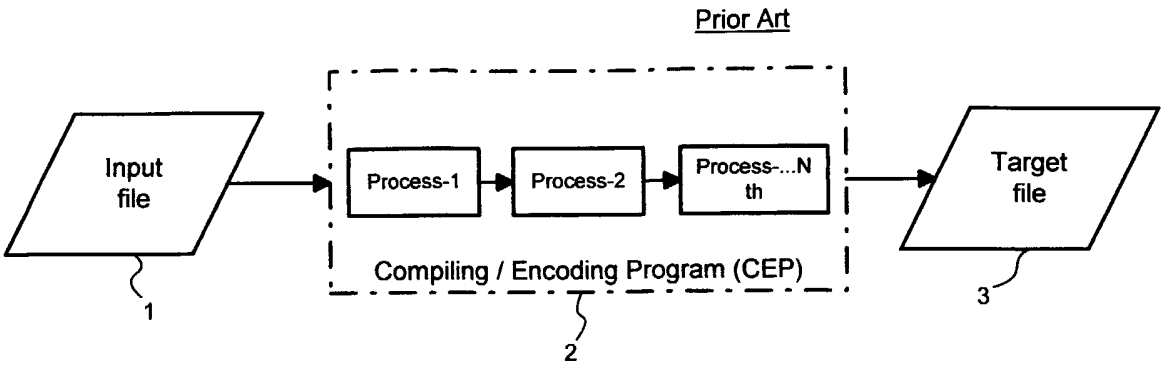

Prior Art
Input file
Process-1
Process-2
Process-...N th
Target file
Compiling / Encoding Program (CEP)
1
2
3

FIG. 1

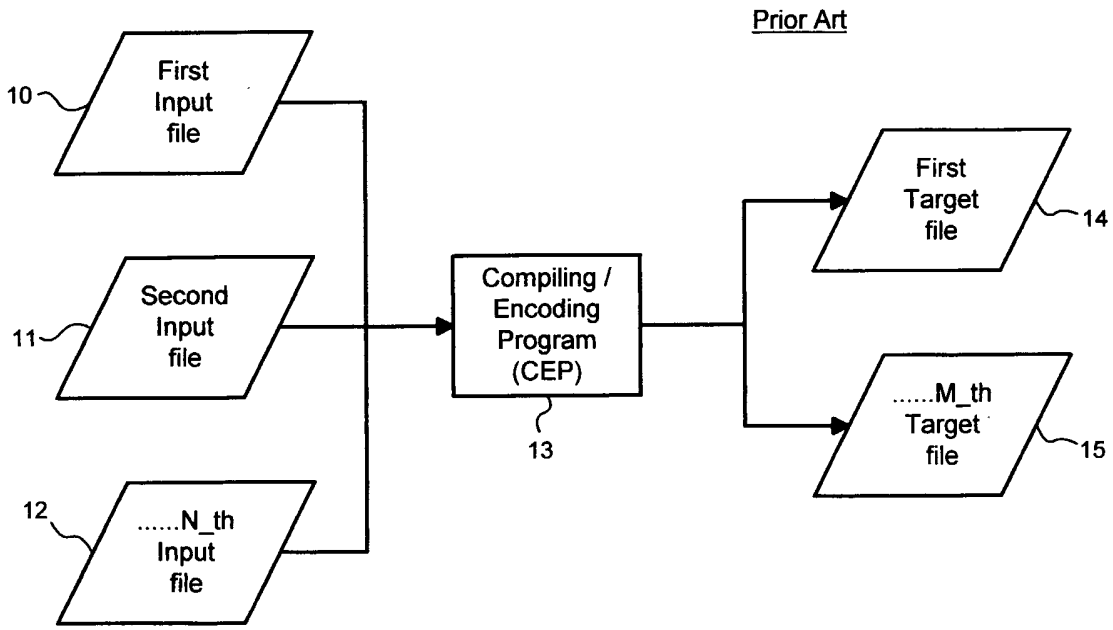

Prior Art
10
First Input file
11
Second Input file
12
.....N_th Input file
Compiling / Encoding Program (CEP)
13
First Target file
14
.....M_th Target file
15

FIG. 2

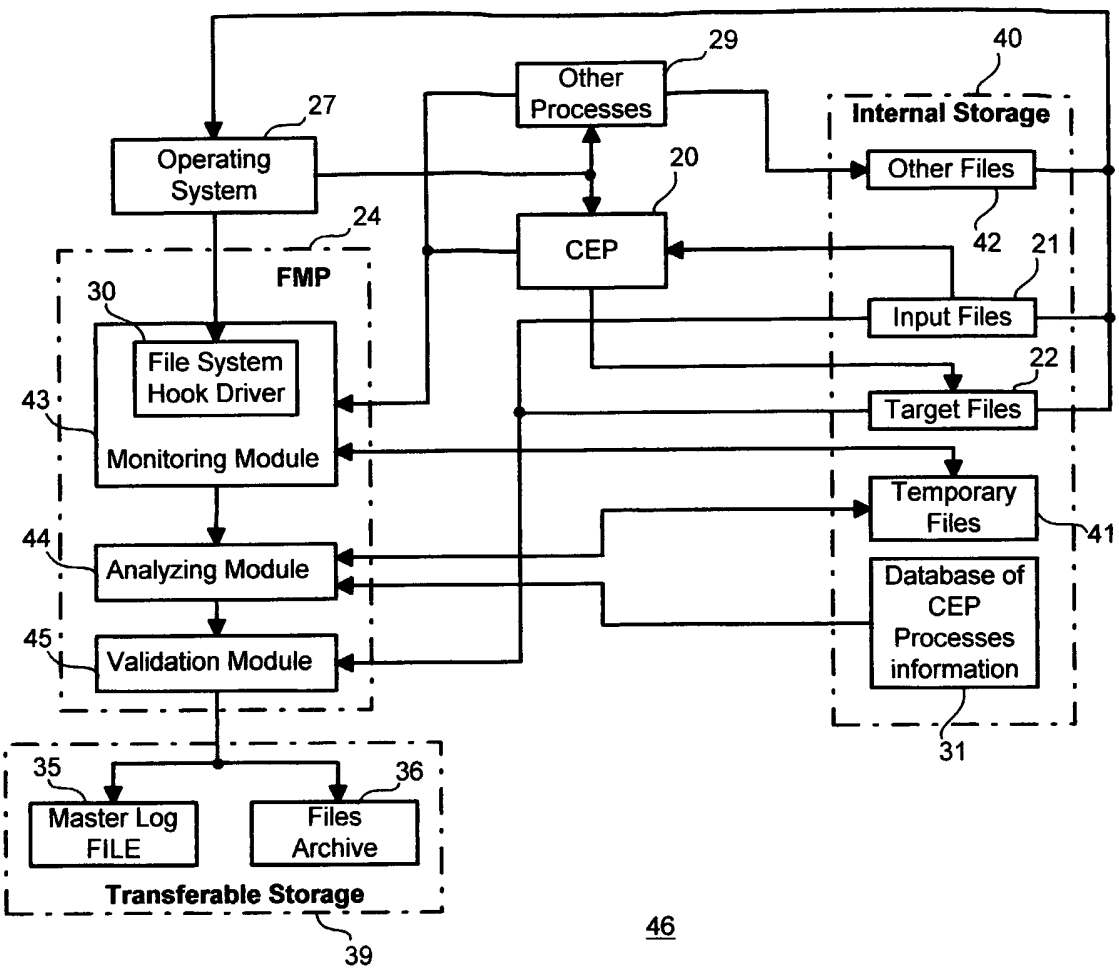

29
40
Other Processes
Internal Storage
27
Operating System
20
Other Files
24
CEP
42
21
FMP
30
Input Files
File System Hook Driver
22
43
Target Files
Monitoring Module
Temporary Files
41
44
Analyzing Module
Database of CEP Processes information
45
Validation Module
31
35
36
Master Log FILE
Files Archive
Transferable Storage
39
46

FIG. 3A

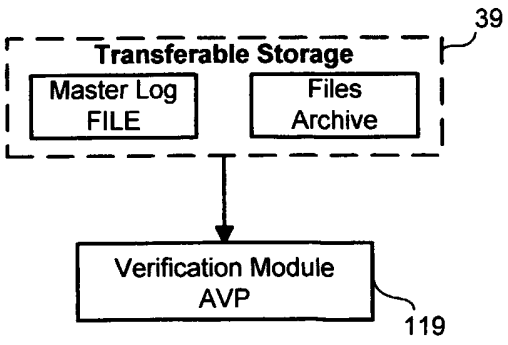

39
Transferable Storage
Master Log FILE
Files Archive
Verification Module AVP
119

FIG. 3B

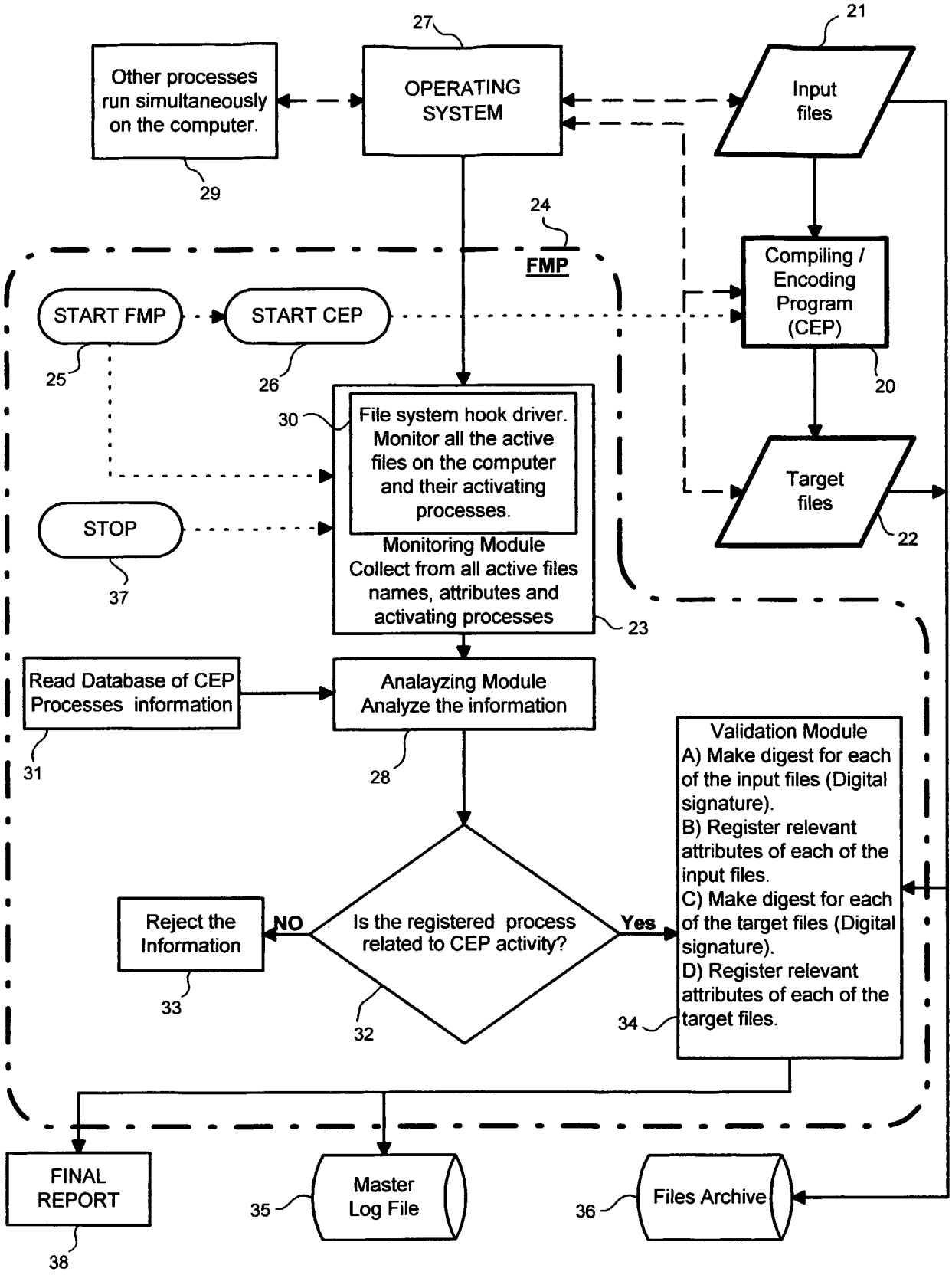

27
21
Other processes run simultaneously on the computer.
OPERATING SYSTEM
Input files
29
24
FMP
Compiling / Encoding Program (CEP)
START FMP
START CEP
25
26
20
30
File system hook driver. Monitor all the active files on the computer and their activating processes.
Target files
STOP
Monitoring Module
Collect from all active files names, attributes and activating processes
22
37
23
Read Database of CEP Processes information
Analayzing Module
Analyze the information
Validation Module
A) Make digest for each of the input files (Digital signature).
B) Register relevant attributes of each of the input files.
C) Make digest for each of the target files (Digital signature).
D) Register relevant attributes of each of the target files.
31
28
Reject the Information
NO
Is the registered process related to CEP activity?
Yes
33
32
34
FINAL REPORT
Master Log File
Files Archive
35
36
38

FIG. 4

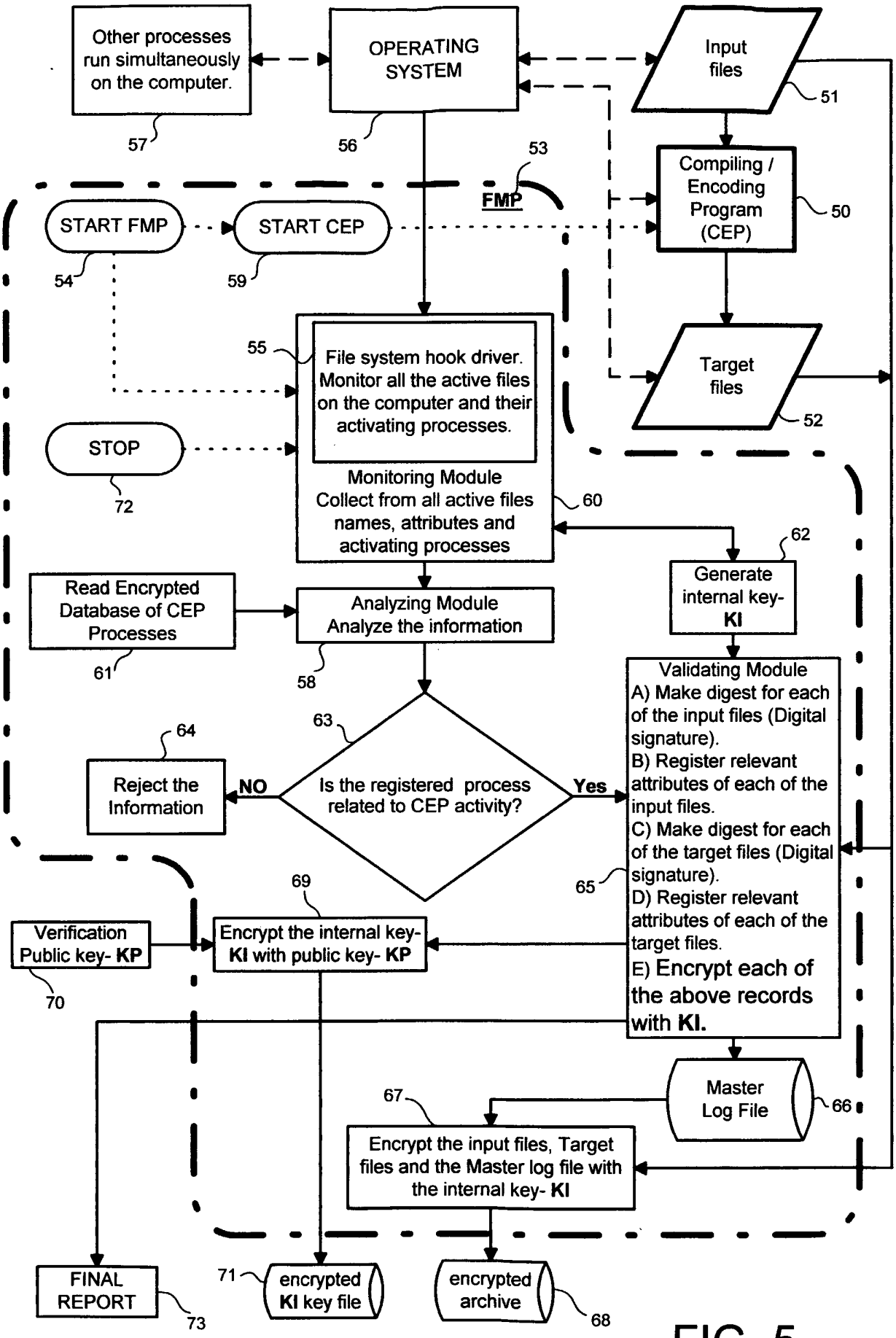

Other processes run simultaneously on the computer.
57
OPERATING SYSTEM
56
Input files
51
53
FMP
Compiling / Encoding Program (CEP)
50
START FMP
54
START CEP
59
55
File system hook driver. Monitor all the active files on the computer and their activating processes.
Target files
52
STOP
72
Monitoring Module
Collect from all active files names, attributes and activating processes
60
62
Generate internal key- KI
Read Encrypted Database of CEP Processes
61
Analyzing Module
Analyze the information
58
Validating Module
A) Make digest for each of the input files (Digital signature).
B) Register relevant attributes of each of the input files.
C) Make digest for each of the target files (Digital signature).
D) Register relevant attributes of each of the target files.
E) Encrypt each of the above records with KI.
65
63
Is the registered process related to CEP activity?
NO
Yes
64
Reject the Information
69
Verification Public key- KP
70
Encrypt the internal key- KI with public key- KP
Master Log File
66
67
Encrypt the input files, Target files and the Master log file with the internal key- KI
FINAL REPORT
73
71
encrypted KI key file
encrypted archive
68

FIG. 5

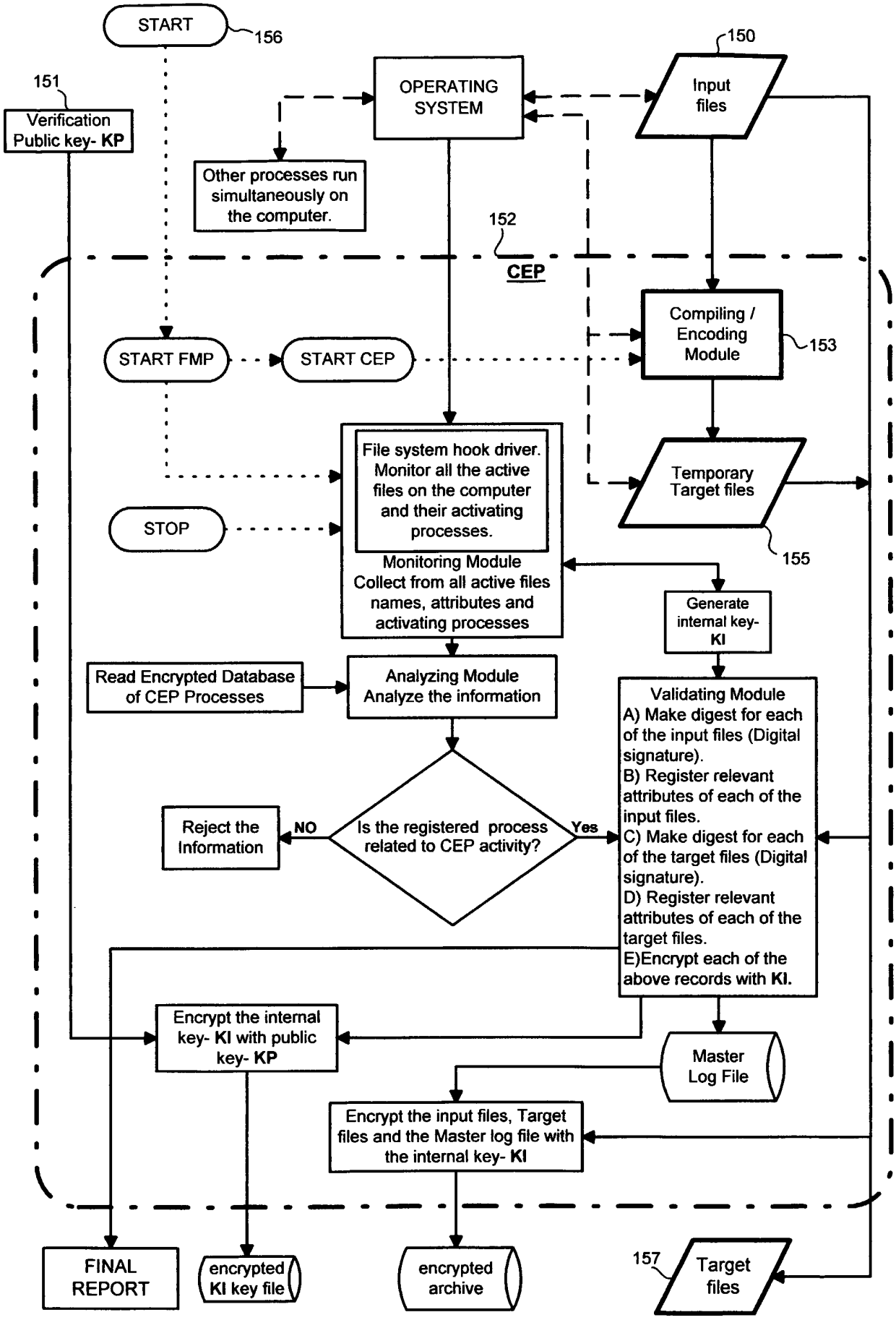

START
156
150
151
Verification Public key- KP
OPERATING SYSTEM
Input files
Other processes run simultaneously on the computer.
152
CEP
Compiling / Encoding Module
153
START FMP
START CEP
File system hook driver. Monitor all the active files on the computer and their activating processes.
Temporary Target files
STOP
155
Monitoring Module Collect from all active files names, attributes and activating processes
Generate internal key- KI
Read Encrypted Database of CEP Processes
Analyzing Module Analyze the information
Validating Module
A) Make digest for each of the input files (Digital signature).
B) Register relevant attributes of each of the input files.
C) Make digest for each of the target files (Digital signature).
D) Register relevant attributes of each of the target files.
E)Encrypt each of the above records with KI.
Reject the Information
NO
Is the registered process related to CEP activity?
Yes
Encrypt the internal key- KI with public key- KP
Master Log File
Encrypt the input files, Target files and the Master log file with the internal key- KI
FINAL REPORT
encrypted KI key file
encrypted archive
157
Target files

FIG. 6

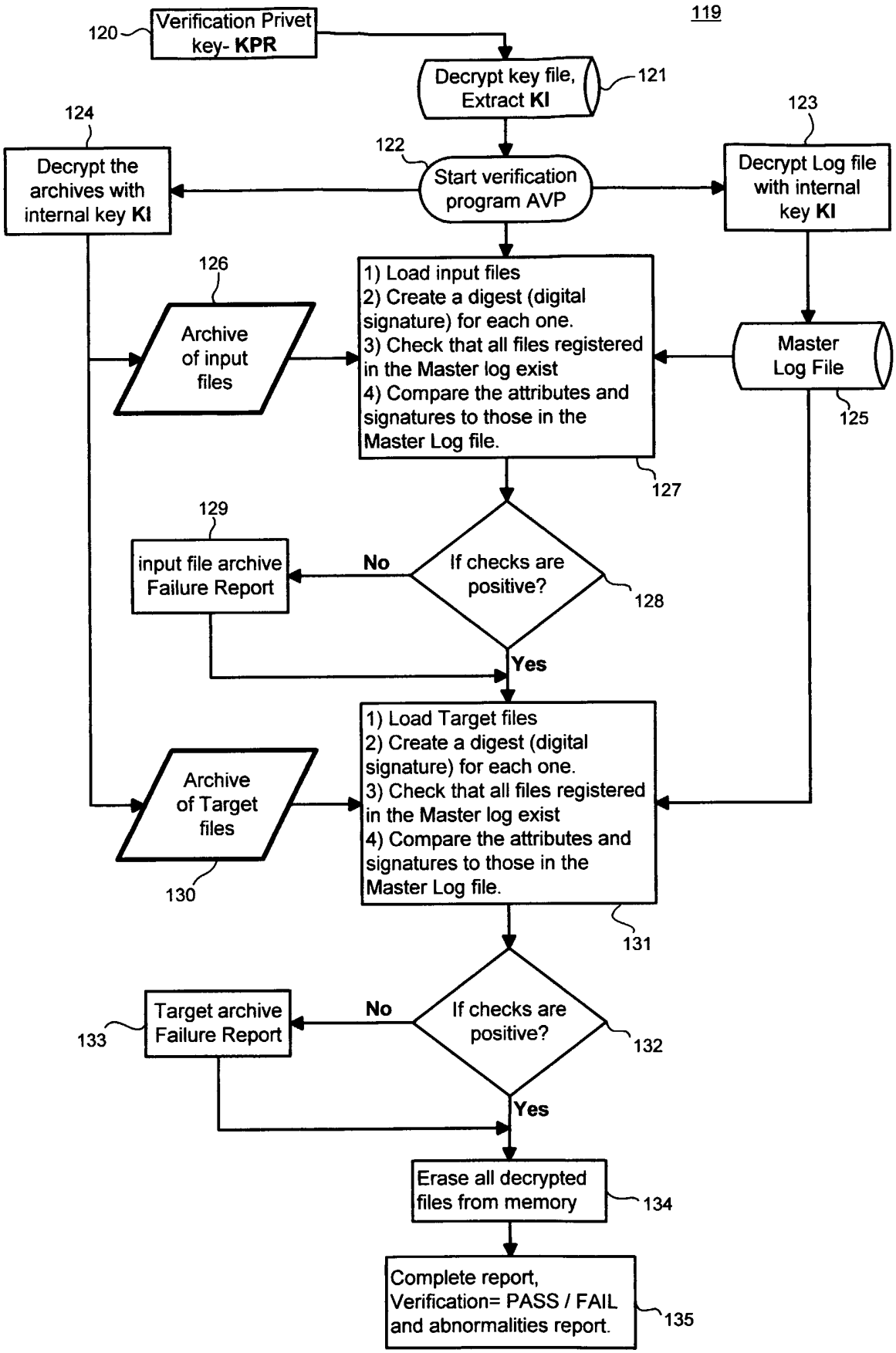

119
120
Verification Privet key- KPR
Decrypt key file, Extract KI
121
123
124
Decrypt the archives with internal key KI
122
Start verification program AVP
Decrypt Log file with internal key KI
126
Archive of input files
1) Load input files
2) Create a digest (digital signature) for each one.
3) Check that all files registered in the Master log exist
4) Compare the attributes and signatures to those in the Master Log file.
Master Log File
125
127
129
input file archive Failure Report
No
If checks are positive?
128
Yes
1) Load Target files
2) Create a digest (digital signature) for each one.
3) Check that all files registered in the Master log exist
4) Compare the attributes and signatures to those in the Master Log file.
Archive of Target files
130
131
Target archive Failure Report
133
No
If checks are positive?
132
Yes
Erase all decrypted files from memory
134
Complete report, Verification= PASS / FAIL and abnormalities report.
135

FIG. 7

FMP Pass Fail Report 180

Developer Information-
Status - Pass / failed-
Version creation date-
Version size-
Missing Source files-
Changed Source files-
Added and Changed Source files-
Source added after START or changed after BUILD-
Missing Used files-
Used But Not found Files-
Illegally changed EXE / DLL / OBJ files-
EXE / DLL Generated not in Working folder-
Third party libraries or objects-
(No available source file)
Source files in Working folder-
Source files Not in Working folder-
Used OBJ files in Working folder-
Used OBJ files Not in Working folder-
EXE / DLL list-
Unused OBJ files-
All other participating files-

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL05/01058

A. CLASSIFICATION OF SUBJECT MATTER
IPC: **G06F 9/45**(2006.01),**11/30**(2006.01)

USPC: 717/120,131,140,162,163;713/193,194
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 717/120,131,140,162,163; 713/193,194

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EAST - authenticate, validate, link, hash, message digest, attribute, file, components

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6,185,569 B1 (EAST et al) 06 February 2001 (06.02.2001)	1-29
A	US 2003/0115461 A1 (O'NEILL) 19 June 2003 (19.06.2003)	1-29
A	US 2003/0196096 A1 (SUTTON) 16 October 2003 (16.10.2003)	1-29

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
02 March 2006 (02.03.2006)

Date of mailing of the international search report
14 MAR 2006

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
Facsimile No. (571) 273-3201

Authorized officer
for
Matthew B. Smithers
James A. Matthews
Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet) (April 2005)