



(51) International Patent Classification:
G06F 21/00 (2006.01) *G06F 21/02* (2006.01)

(21) International Application Number:
PCT/EP2009/057830

(22) International Filing Date:
23 June 2009 (23.06.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
08158870.9 24 June 2008 (24.06.2008) EP
08161479.4 30 July 2008 (30.07.2008) EP

(71) Applicant (for all designated States except US): **NA-GRAVISION S.A.** [CH/CH]; Route de Genève 22-24, CH-1033 Cheseaux-sur-Lausanne (CH).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **GREMAUD, Fabien** [CH/CH]; Chemin de Crey-Derrey 116, CH-1618 Châ-

tel-St-Denis (CH). **GOGNIAT, Christophe** [CH/CH]; Place de la Gare 5, CH-1807 Blonay (CH). **BELLOC-CHIO, Marc** [FR/CH]; Les Vernettes, Bât. 8, CH-1081 Montpreveyres (CH). **FUCHS, Pascal** [CH/CH]; c/o Jorge Pena, Avenue Avant-Poste 3, CH-1005 Lausanne (CH).

(74) Agent: **LEMAN CONSULTING S.A. 284**; Chemin de Précossy 31, CH-1260 Nyon (CH).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, 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, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

[Continued on next page]

(54) Title: SECURE MEMORY MANAGEMENT SYSTEM AND METHOD

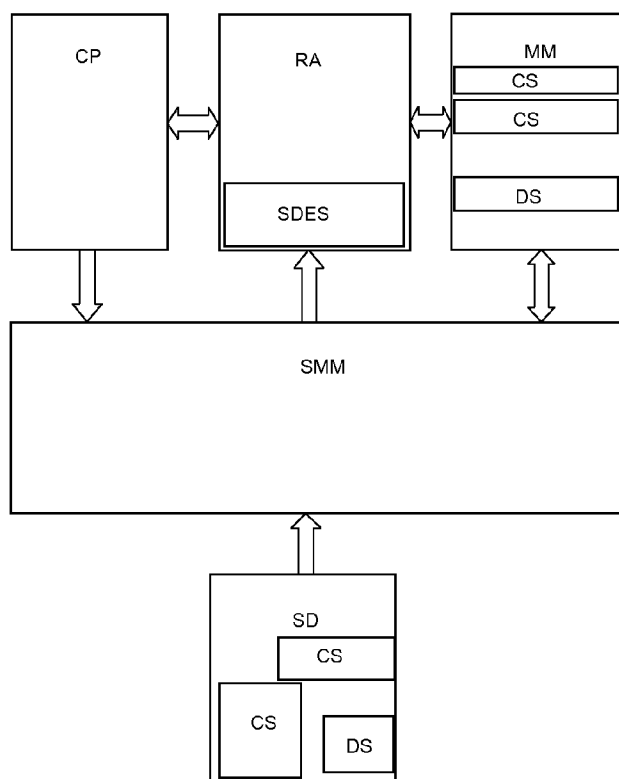


FIG. 1

(57) Abstract: The present invention describes a system and a method for securely loading digital information from a storage device into a memory module in a data processing system, said data processing system comprising at least one storage device, one memory module and at least one processor, said data processing system further comprising a memory access controller module connected between the processor and the memory module, and a secure memory management module connected to the processor, the memory module, the storage device and the memory access controller. Requests by the processor for data are passed to the secure memory management module, which loads the data from the storage device to the memory module and configures the memory access controller such that the processor will have access to the data.



(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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, 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:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

SECURE MEMORY MANAGEMENT SYSTEM AND METHOD

INTRODUCTION

The present invention relates to the domain of computer security, particularly in guaranteeing the secure loading of data or applications into volatile, working memory or in isolating concurrent applications from each other such that one application may not modify data or code intended for another application.

BACKGROUND OF THE INVENTION

A data processing system may include hardware resources such as a processing unit (CPU), volatile memory (RAM) and non-volatile memory (ROM). The data processing system may operate under the control of at least one operating system and may perform routines according to one or several software resources or applications. The applications may be stored in non-volatile memory and loaded into volatile memory when required to be executed. During the execution of an application, the data required by said application or the data which is produced by the application may be stored in the non-volatile or volatile memory or transferred from one memory to another.

With the advent of multiple connectivity options for data processing systems, including wireless connectivity, and with the huge growth in the use of mobile data processing systems, the need to protect these systems from malicious attacks has become increasingly important. Malicious attacks can be aimed at interfering with system booting, modifying the operating system, intercepting and/or modifying data produced by or utilized by some application.

Indeed, it has now become a necessary requirement to protect data processing systems against fraudulent manipulations and attacks on their integrity. Such malicious attacks may come in the form of software designed to take over a data processing system's operating system or otherwise interfere with the normal processing sequence of the data processing system without the user's knowledge or approval. Such software is generally known as malware. The presence of malware in a data processing system is generally difficult to remedy and can lead to complete system failure or even to irreparable damage to the system.

Computer viruses, worms, Trojan horses, spyware etc. are all different types of malware. The different types of malware can attack the processing system in various ways such as by intercepting data which was meant for another application or by monitoring key strokes in order to steal passwords or other information which is meant to be kept secret, modifying or otherwise altering data or corrupting files, modifying a program in order to cause it to crash or to execute some function which was not originally intended by the user.

Systems to combat against malware attacks exist and usually feature a memory management unit, which is configurable by the system's CPU. Security is thus compromised if the CPU suffers an attack from an ill-intentioned user. Because of the complexity exhibited by a modern CPU, the additional security functions which would be required in order to minimize the possibility of such an attack would lead to a significant cost increase in terms of the extra on-chip real estate necessary to implement such functions and would lead to computing overhead and therefore compromise the speed of operation. Again, due to the complexity of a typical CPU, such modifications could not offer a high level of security with an acceptable level of confidence. Accordingly, it would be desirable to have a cost-efficient and size-efficient solution providing secure management of the loading or unloading of data or applications into or out of memories in a data processing system.

SUMMARY OF THE INVENTION

The present invention describes a system and a method for securely loading digital information from a storage device into a memory module in a data processing system comprising at least one storage device (SD), at least one memory module (MM) and at least one processor (CP), said storage device (SD) having at least one segment of data as well as access and authentication data related to the segment, characterized in that it comprises a memory access controller (RA) connected between the processor (CP) and the memory module (MM), and a secure memory management module (SMM) connected to the processor (CP), the memory module (MM), the storage device (SD) and the memory access controller (RA), said secure memory management module (SMM) comprising means to receive a request from the processor (CP) for a segment stored in the storage device (SD), said segment having at least one access condition and at least one piece of authentication data pertaining

to it, said secure memory management module (SMM) further comprising means to load the requested segment from the storage device (SD) to the memory module (MM), means to authenticate the access condition using the authentication data, means to configure the memory access controller (RA) using the access condition, 5 said memory access controller (RA) having means to detect a processor status from the processor (CP) and means to compare the access condition with the processor status, the memory access controller (RA) having means to allow or to block the access to the memory module (MM) according to the result of the comparison.

The method used in the present invention to securely load data from the storage 10 device to the memory module comprises the following steps:

receiving, by the secure memory management module (SMM), a request from the processor (CP) for digital information,

locating a segment in the storage device (SD) containing the requested digital information,

15 extracting a set of access conditions and authentication data pertaining to said segment,

authenticating the access conditions with the authentication data,

determining an appropriate region in the memory module (MM) to accommodate the located segment,

20 loading said located segment into the determined region in the memory module (MM),

configuring the memory access controller module (RA) using the extracted access conditions.

The invention therefore uses a memory access controller (RA) to serve as a firewall 25 between the processor (CP) and the memory module (MM) coupled with a secure memory management module (SMM) to load the memory module (MM) and configure the memory access controller (RA). This leads to a high level of security in the system since the secure memory management module (SMM) is of simple

architecture with a reduced set of commands such as load and store, and therefore less prone to attack than would be a CPU in the case that said CPU were responsible for configuring the memory access controller module.

BRIEF DESCRIPTION OF THE DRAWINGS

- 5 The invention will best be understood by reference to the following detailed description of a preferred embodiment when read in conjunction with the accompanying drawing, wherein:

FIG.1 shows a data processing system comprising a processor (CP), a memory module (MM), a storage device (SD), a memory access controller module (RA) and a
10 secure memory management module (SMM).

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Modern data processing systems are generally memory intensive. This fact, coupled with the fact that on-chip memory can be expensive, leads to the necessity of relying on significant amounts of off-chip storage in many data processing systems. In the
15 case where security is important, there is a need therefore to be able to secure transfers of data between the off-chip or non-trusted environment and the on-chip or trusted environment. Additionally, with the significant complexity of modern CPUs (processors), it is not easy to modify the processor to be able to achieve the required goal while maintaining the required high level of security, nor is it cheap in terms of
20 on-chip real-estate. Therefore it would be better to have a dedicated piece of hardware, using a limited number of commands such as load and store, to take care of tasks related to memory access.

The present invention describes a hardware solution and a method for providing, within a data processing system, a means for secure loading of digital information
25 from a storage device into a memory module. In general the storage device is in a non-trusted environment and the memory module is in a trusted environment. The invention provides an interface between the trusted environment and the non-trusted environment, through which requests for access to the digital information must pass. The invention includes means for configuring the interface such that a processor
30 having the necessary access rights to the data stored in the memory module will

indeed have access to the required part of the memory once it has been loaded. The storage device can take the form of a flash memory, an EPROM, an EEPROM, a ROM, a hard disk, an external server or other such storage means. The memory module will usually take the form of a random access memory (RAM) i.e. a volatile memory. In the context of this document, the term digital information is used to describe data liable to be loaded into the memory module, such as executable code or information generated by executable code or used by executable code.

The secure memory management system of the current invention is integrated into a data processing system (FIG.1) comprising at least a processor (CP), a memory module (MM) and a storage device (SD) and includes dedicated hardware known as a secure memory management module (SMM) connected between the processor (CP) and memory module (MM) on one side, and the storage device (SD) on the other side, the purpose of the secure memory management module (SMM) being to manage the communication between the processor (CP) and the memory module (MM) as well as to transfer digital information between the storage device (SD) and the memory module (MM), i.e. to load and unload the memory module (MM). The secure memory management system further includes dedicated hardware known as a memory access controller module (RA) placed between the processor (CP) and the memory module (MM). The memory access controller module (RA) acts as a firewall between the processor (CP) and the memory module (MM). The digital information stored in the storage device (SD) has a set of access conditions associated with it, which are stored along with the digital information. The secure memory management module (SMM) configures the memory access controller (RA) to allow the processor (CP), given that said processor (CP) has the appropriate access rights, the correct access to the parts of the memory module (MM) which have been loaded. The secure memory management module (SMM) thus functions together with the memory access controller module (RA) to ensure that the memory module (MM) remains secure.

In the present invention, a convention known as segmentation is used i.e. the digital information is stored in segments. The segmentation convention is used for the digital information stored in the storage device as well as for the digital information stored in the memory module. A segment is made up of several blocks of digital

information comprising a predetermined number of bytes. For example a block of digital information could be 32 bytes long. Each segment has a set of attributes associated with it, such as a segment identification number, the type of data contained in the segment, the length of the segment, the address of the segment, a digital signature, an integrity figure such as a one-way function of the contents of the segment for example, the set of conditions rights describing which processes can have read access or write access or execute access to the segment. These attributes are recorded in a segment header attached to and stored with the segment. FIG.1 shows two different types of segments, namely code segments (CS) comprising executable code and data segments (DS) comprising digital information which can be used by an application or generated by an application.

Digital information which is currently being used by the processor (CP) is held in the memory module (MM). When the processor (CP) requires access to digital information which does not currently reside in the memory module (MM), the processor (CP) sends a request to the secure memory management module (SMM) for the required digital information. Upon receiving the request from the processor (CP), the secure memory management module (SMM) locates the segment, or the plurality of segments containing the requested digital information, and extracts several pieces of information from the segment header, including access conditions, a digital signature and a segment integrity figure. The secure memory management module (SMM) performs an authentication on the segment by verifying the digital signature according to a predetermined cryptographic technique. The secure memory management module (SMM) performs an integrity check on the segment by calculating an integrity figure, such as a one-way function of the contents of the segment, and comparing the calculated integrity figure with the integrity figure extracted from the segment header. The secure memory management module (SMM) determines an appropriate region in the memory module (MM) capable of accommodating the located segment and loads said segment into the memory module (MM) at the determined region, said region comprising a plurality of addressable memory module locations. In one embodiment of the current invention the integrity check could be done on-the-fly, block by block or segment by segment while loading the memory module (MM). In another embodiment of the current invention, the integrity check could be done in the memory module (MM) after having

been loaded. The secure memory management module (SMM) will also configure the memory access controller (RA) so that a processor with the necessary access rights will have the required access to the loaded data.

5 In general, a data processing system will also comprise some means to do memory mapping, whereby a block or a segment of digital information is accessed by the processor using a virtual address while said block or segment is stored at some physical address in the memory module (MM) which is different from the virtual address. The segment header may further comprise the virtual address of the segment. When the digital information is loaded into an available space in the
10 memory module, the mapping is updated to reflect a link between the virtual address and the physical address where the information was loaded. In one embodiment of the current invention this memory mapping could vary between successive loads of the memory module (MM).

15 In another embodiment of the present invention the digital information in the storage device is preferably in encrypted format. In this case the segment headers further comprises segment keys with which to decrypt the segments. The segment keys are preferably extracted from the segment headers by the secure memory management module (SMM) and the keys used to decrypt the digital information before loading into the memory module (MM).

20 The memory access controller (RA) contains a segment descriptor (SDS). The segment descriptor holds part of the segment header corresponding to each segment of digital information that has been loaded into the memory module. As each segment is loaded, the secure memory management module, having extracted and authenticated the access conditions to that segment from the segment header,
25 updates the access conditions in the segment descriptor in such a way as to allow the processor, given that said processor has the appropriate access rights, to have access to the corresponding segment in the memory module. Until the segment descriptor is updated, the processor has no access to the part of the memory module where the digital information is being loaded. In order to allow for compatibility over
30 several different operating systems, the segment descriptors could be of an "ELF" format (Executable and Linking Format), which is a standard file format for executables, object code, libraries etc.

The segment descriptor specifies which types of access are allowed by the processor. For example, a certain range of addresses may only be accessible in read mode whereas any attempt to write to that region would be disallowed. This type of information is indicated in the segment descriptor. Similarly, regions where a processor is allowed to write or to erase are indicated in the segment descriptor. The segment descriptor may also indicate regions which hold executable routines or functions so that a processor would need to have the right to execute in order to be able to fetch an instruction from a region indicated as holding executables (or certain process ids would have the right to execute certain functions).

Beside the data and the address buses, the transfer from or to peripherals is controlled by control lines (read/write) defining the access type. Other lines can be used to determine if executable code is fetched (execute mode) or if the processor is reading/writing data in a memory.

One example of the access conditions attached to a segment define the condition in read, write or execute mode.

According to another embodiment, the access conditions are defined in relation with the mode of the processor.

Whether or not a processor will have the right to access a certain piece of data depends then on the access conditions associated attached to the segment in which that piece of data is located, and on the mode in which the processor is running at the time that it requests that data - for example the processor can be in user mode or super-user mode. The mode of operation is usually indicated by a bit in a status or mode register associated with the processor. The mode in which the processor runs at any particular time can be verified by checking the state of the relevant register.

This register can be connected to communication lines with the memory access controller allowing the latter to determine in which mode is the processor. This mode can be also communicated to the memory access controller by transferring the register value via the main bus. Depending on the mode of operation of the processor at the time it makes a request for a piece of data, access can either be granted or denied according to the result of a comparison between the mode of operation of the processor and the mode required by that piece of data according to

associated access conditions held in the segment descriptor. Similarly, it is easy to imagine how a process identifier (process id) could be used, in a similar way to processor mode, to indicate which processes have a right to access particular segments of data.

- 5 The role of the memory access controller is then to receive the value of the processor mode and compare this mode with the content of the access conditions attached to the segment.

In a further embodiment of the present invention, a segment of data may have a plurality of sets of access conditions associated with it, each of the sets pertaining to a particular mode of processor operation . When the processor requests access to a piece of data in a segment for which there is a plurality of access conditions, then the memory access controller (RA) will apply the set of access conditions which are relevant to the mode in which the processor is running. The processor mode define the set of access conditions and the type of the processor (read/write/execute) will then be used to define the access to the piece of data.

The access conditions attached to a segment therefore define the mode that the processor needs to have in order to access the segment, the status including the type of access requested and the mode of operation of the processor.

In a further embodiment of the present invention, a light encryption could be used on the digital information before storing it in the memory module. According to a first embodiment, the encryption key would be based on a random number generated by the secure memory management module. This number can be generated while initializing the system so that each time the system is powered on, a new key will be generated. The digital information in a segment would be encrypted under this key and the key would be placed in the segment descriptor corresponding to that segment. The random key could alternatively be generated each time a segment is uploaded. The digital information would then be decrypted by the memory access controller at the time that the processor requests that information. In yet another embodiment of the current invention, the encryption key could be a function of the physical address (the actual address in the memory module) in which the digital information is loaded (i.e. calculating a one-way function of the physical address).

As well as ensuring the security of digital information loaded into the memory module, the current invention also allows for the processor to complete other tasks while the memory module is being loaded, since all functions related to the loading of the memory module are delegated to the secure memory management module. With
5 the high level of complexity built into modern processors, the present invention provides for the advantage of guaranteeing a high level of security by having the described memory management functions handled by a dedicated hardware system represented by the combination of the memory access controller (RA) and the secure memory management module (SMM) rather than by trying to include these functions
10 in the already complex processor.

CLAIMS

1. A data processing system comprising at least one storage device (SD), at least one memory module (MM) and at least one processor (CP), said storage device (SD) having at least one segment of data as well as access and authentication data related to the segment, characterized in that it comprises a memory access controller (RA) connected between the processor (CP) and the memory module (MM), and a secure memory management module (SMM) connected to the processor (CP), the memory module (MM), the storage device (SD) and the memory access controller (RA), said secure memory management module (SMM) comprising means to receive a request from the processor (CP) for a segment stored in the storage device (SD), said segment having at least one access condition and at least one piece of authentication data pertaining to it, said secure memory management module (SMM) further comprising means to load the requested segment from the storage device (SD) to the memory module (MM), means to authenticate the access condition using the authentication data, means to configure the memory access controller (RA) using the access condition, said memory access controller (RA) having means to detect a processor status from the processor (CP) and means to compare the access condition with the processor status, the memory access controller (RA) having means to allow or to block the access to the memory module (MM) according to the result of the comparison.
2. The data processing system according to claim 1, wherein said segment further comprises an integrity figure and said secure memory management module (SMM) comprises means to check the integrity of said segment by comparing said integrity figure with a calculated integrity figure based on the data in the segment.
3. The data processing system according to either of claims 1 or 2, wherein the access condition defines a condition for reading or writing into or for executing instructions from said memory module (MM) being determined by the memory access controller (RA) according to the type of access requested by the processor (CP).
4. The data processing system according any of claims 1 to 3, wherein the access condition comprises a plurality of sets of access data, the processor (CP) communicating a processor mode to the memory access controller (RA), said

memory access controller (RA) selecting the set of access data based on the received processor mode.

5. A method for securely loading digital information from a storage device (SD) into a memory module (MM) in a data processing system, said data processing system comprising at least one storage device (SD), at least one memory module (MM) and at least one processor (CP), said data processing system further comprising a memory access controller module (RA) connected between the processor (CP) and the memory module (MM), and a secure memory management module (SMM) connected to the processor (CP), the memory module (MM), the storage device (SD) and the memory access controller (RA), said method comprising the following steps:

receiving, by the secure memory management module (SMM), a request from the processor (CP) for digital information,

locating a segment in the storage device (SD) containing the requested digital information,

extracting a set of access conditions and authentication data pertaining to said segment,

authenticating the access conditions with the authentication data,

determining an appropriate region in the memory module (MM) to accommodate the located segment,

loading said located segment into the determined region in the memory module (MM),

configuring the memory access controller module (RA) using the extracted access conditions.

6. The method according to claim 5 characterised in that it further comprises the steps of:

extracting an integrity figure pertaining to said segment of digital information,

calculating an integrity figure for said segment of digital information,

verifying the integrity of the segment of digital information by comparing the extracted integrity figure with the calculated integrity figure.

7. The method according to either of claims 5 or 6 characterised in that the storage device (SD) is in a non-trusted environment and the memory module (MM) is in a trusted environment.

8. The method according to any of claims 5 to 7 characterised in that a mapping scheme is used whereby a virtual address is related to a physical address, said virtual address being an address used by the processor to refer to a piece of digital information and said physical address being an address of a location in the memory module (MM) where said digital information is stored.

9. The method according to claim 8 characterised in that the relationship between the virtual address and the physical address is modified at least once following a load of the memory module (MM).

10. The method according to any of claims 5 to 9 characterised in that the segment of digital information in the storage device is in an encrypted format and it further comprises the steps of:

extracting a segment key pertaining to said segment of digital information,

decrypting said segment of digital information using the extracted segment key.

11. The method according to any of claims 5 to 10 characterised in that the segment of digital information is encrypted by a key prior to loading into the memory module (MM).

12. The method according to claim 11 characterised in that the encryption key is based on a random number generated in the secure memory management module (SMM).

13. The method according to claim 11 characterised in that the encryption key is based on the physical address.

14. The method according to any of claims 5 to 13 characterised in that the memory module (MM) is a volatile memory.

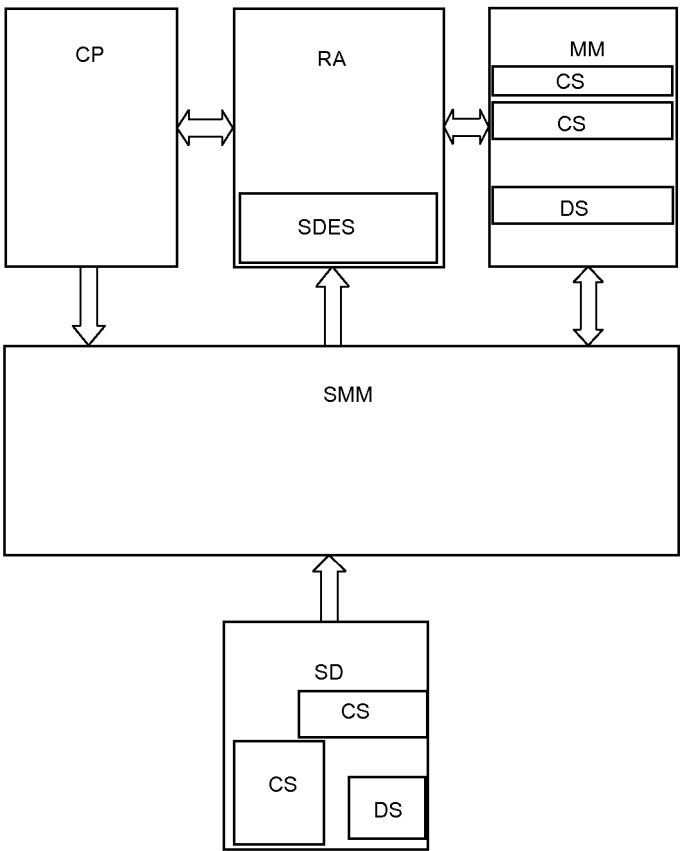


FIG. 1

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2009/057830

A. CLASSIFICATION OF SUBJECT MATTER
 INV. G06F21/00 G06F21/02

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)
 G06F

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 practical, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 825 878 A (TAKAHASHI RICHARD [US] ET AL) 20 October 1998 (1998-10-20) column 2, line 24 - column 4, line 20 figure 1	1-14
A	US 7 043 616 B1 (MCGRATH KEVIN J [US]) 9 May 2006 (2006-05-09) column 2, line 62 - column 3, line 18 column 6, line 10 - line 40 column 9, line 37 - line 57	1-14
A	WO 2006/093618 A (SILICON STORAGE TECH INC [US]; DING ZHIMIN [US]; HOLLMER SHANE [US]; B) 8 September 2006 (2006-09-08) the whole document	1-14
	----- -/--	

☒ 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 document but 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.
- *G* document member of the same patent family

Date of the actual completion of the international search

13 October 2009

Date of mailing of the international search report

20/10/2009

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer

Sigolo, Alessandro

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2009/057830

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2006/036830 A1 (DINECHIN CHRISTOPHE DE [FR] ET AL) 16 February 2006 (2006-02-16) paragraph [0035] figure 7	1-14
A	----- US 2003/200451 A1 (EVANS EDWARD K [US] ET AL) 23 October 2003 (2003-10-23) paragraph [0011] - paragraph [0014] paragraph [0032] - paragraph [0040] -----	1-14

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2009/057830

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 5825878	A	20-10-1998	NONE	
US 7043616	B1	09-05-2006	NONE	
WO 2006093618	A	08-09-2006	US 2006218425 A1	28-09-2006
US 2006036830	A1	16-02-2006	NONE	
US 2003200451	A1	23-10-2003	AU 2003230960 A1	03-11-2003
			CN 1647046 A	27-07-2005
			EP 1499976 A1	26-01-2005
			JP 4299679 B2	22-07-2009
			JP 2005523519 T	04-08-2005
			WO 03090086 A1	30-10-2003