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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 August 2006 (03.08.2006)

PCT

(10) International Publication Number
WO 2006/079139 A1

(51) International Patent Classification:
H04L 9/00 (2006.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/AU2005/001584

(22) International Filing Date: 12 October 2005 (12.10.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2004905889 12 October 2004 (12.10.2004) AU

(71) Applicant *(for all designated States except US)*: **CANON KABUSHIKI KAISHA** [JP/JP]; 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo 146 (JP).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **PARTIS, Ashley** [AU/AU]; 410/3 Warayama Place, Rozelle, NSW 2039 (AU).

(74) Agent: **SPRUSON & FERGUSON**; GPO Box 3898, Sydney, NSW 2001 (AU).

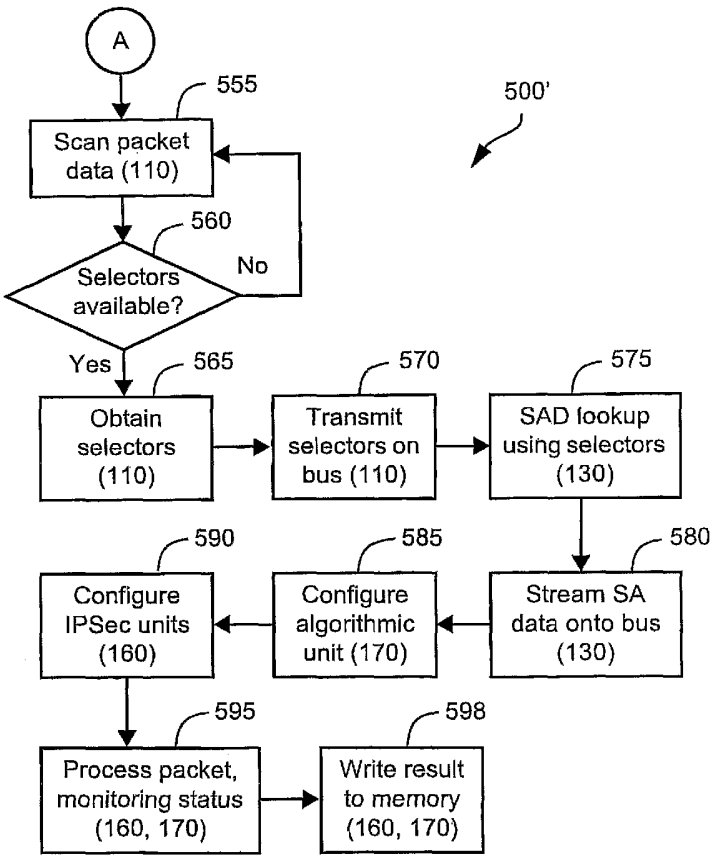
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK,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

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

Published:
— *with international search report*

[Continued on next page]

(54) Title: CONCURRENT IPSEC PROCESSING SYSTEM AND METHOD



(57) Abstract: A method of performing IPsec processing of an incoming communication packet is disclosed. The method comprises determining (560), from a received portion of the incoming packet, if sufficient information has been received to enable the IPsec processing to be commenced, obtaining (565) the necessary information from the received portion of the packet, and commencing (570, 575, 590, 585, 580, 595, 598) IPsec processing of said packet before the entire packet has been received depending upon the obtained information.

WO 2006/079139 A1

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.

CONCURRENT IPSEC PROCESSING SYSTEM AND METHOD

Field of the Invention

The current invention relates to cryptographic processing and in particular to methods of increasing the speed of processing IPsec cryptographic packets (RFC2401).

Background

As the Internet has grown, privacy has progressively become a more significant issue, with sensitive information increasingly being communicated over insecure data networks. There are many different applications requiring privacy and security over the Internet, such as banking, commercial transactions or accessing private company networks remotely. These transactions often involve sending sensitive information over insecure sections of the Internet, requiring such sensitive information to be secured.

The present data communication protocol that is generally used to communicate over the Internet, namely Internet Protocol (IP), is not secure. Consequently, third parties can potentially eavesdrop on transmitted packets, repeat transmitted packets or even divert packets off the network and replace the diverted packets with locally created or forged packets.

In order to protect against this, a method of securing Internet communication packets has been created, and called IP Security, or "IPsec". IPsec is a complex approach that provides security services for IP packets, thereby providing confidentiality, authentication and protection against "replayed" packets. IPsec is controlled by a set of rules (also known as "Security Policies") that are created prior to processing network traffic.

Several protocols are used in IPsec, each protocol having headers that must be processed separately to the base IP header. Each IPsec protocol offers a set of services. Examples of IPsec protocols include the Encapsulating Protocol (ESP), the

Authentication Header (AH), IP Compression (IPcomp) and Internet Key Exchange (IKE).

Each IPsec protocol uses one or more cryptographic algorithms to provide the services offered by the protocol. Due to the speed bottleneck presented by the cryptographic algorithms, it is difficult to process IPsec packets at faster than network (line) speeds without parallelism and pipelining. These approaches include processing multiple IPsec packets at one point in time, which is difficult to do at high speeds.

A further problem is the unpredictable amount of latency generated from a Security Association Database (SAD) lookup. This latency can be extremely small or large, depending on the type of lookup and the number of Security Associations (SA's) involved.

Summary

It is an object of the present invention to substantially overcome, or at least ameliorate, one or more disadvantages of existing arrangements.

Disclosed are arrangements, referred to in this specification as "partial packet IPsec processing" arrangements, which seek to address the above problems by starting to IPsec process a packet as soon as sufficient "selectors" have been received, without waiting for the entire packet to be received. The disclosed partial packet IPsec processing arrangements are applicable both to packets being IPsec processed for transmission, these packets being referred to as "packets for transmission", and to IPsec processing of received packets, these packets being referred to as "received packets". The term "incoming packet" refers to a packet that is input into a disclosed partial packet IPsec processing arrangement, and can be either a packet for transmission, or a received packet.

The term "selector" is described in relation to **Figs. 3A-3D** and **Fig. 4**. Prior art solutions require the entire packet to be written to memory before the selectors can be found in the packet and/or before processing can begin. This aggravates the latency of

the processing due to the need to wait for the packet to be completely received before processing of the packet can start. This latency can be unpredictable and large, as packets can range from 20 bytes to 64 k bytes, although most packets range from 64 bytes to 8 k bytes in size. The disclosed partial packet IPsec processing approach reduces the aforementioned latency by starting the processing cycle as soon as enough information is present.

According to a first aspect of the present invention, there is provided a method of performing IPsec processing of an incoming communication packet, the method comprising the steps of:

determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

obtaining said sufficient information from the received portion of the packet; and

commencing IPsec processing of said packet before the entire packet has been received depending upon the obtained information.

According to another aspect of the present invention, there is provided an apparatus for performing IPsec processing of an incoming communication packet, the apparatus comprising:

means for determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

means for obtaining said information from the received portion of the packet; and

means for IPsec processing said packet before the entire packet has been received depending upon the obtained information.

According to another aspect of the present invention, there is provided an apparatus for performing IPsec processing of an incoming communication packet, the apparatus comprising:

a processor; and

a program for directing the processor to perform the steps of:

determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

obtaining said information from the received portion of the packet; and

commencing IPsec processing of said packet before the entire packet has been received depending upon the obtained information.

According to another aspect of the present invention, there is provided a computer program product including a computer readable medium having recorded thereon a computer program for directing a processor to execute a method of performing IPsec processing of an incoming communication packet, the program comprising:

code for determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

code for obtaining said information from the received portion of the packet; and

code for IPsec processing said packet before the entire packet has been received depending upon the obtained information.

According to another aspect of the present invention, there is provided a computer program for directing a processor to execute a method of performing IPsec processing of an incoming communication packet, the program comprising:

code for determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

code for obtaining said information from the received portion of the packet; and

code for IPsec processing said packet before the entire packet has been received depending upon the obtained information.

According to another aspect of the present invention, there is provided a communication packet formed by a method of performing IPsec processing of an incoming communication packet, the method comprising the steps of:

determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

obtaining said information from the received portion of the packet; and

commencing IPsec processing said packet before the entire packet has been received depending upon the obtained information.

Other aspects of the invention are also disclosed.

Brief Description of the Drawings

One or more embodiments of the present invention will now be described with reference to the drawings, in which:

Fig. 1A is a functional block diagram of an architecture for performing the disclosed partial packet IPsec processing method;

Fig. 1B shows a functional block diagram of the IPsec processing unit of **Fig. 1A**;

Figs. 2A - 2B depict flow charts of concurrent process fragments for one example of a method for implementing the disclosed partial packet IPsec processing approach;

Figs. 3A – 3D show the typical layout of an IPv4 header, IPsec headers and an IPv4 packet;

Fig. 4 shows the typical layout of the packets processed by the partial packet IPsec processing method;

Figs. 5A - 5D depict memory utilisation during operation of the partial packet IPsec processing method;

Fig. 6 is a schematic block diagram of a general purpose computer upon which a software implementation of the partial packet IPsec processing approach arrangement described can be practised; and

Fig. 7 depicts the cryptographic algorithmic unit of **Fig. 1B** in more detail.

Detailed Description including Best Mode

It is to be noted that the discussions contained in the "Background" section and that above relating to prior art arrangements relate to discussions of devices which form public knowledge through their use. Such should not be interpreted as a representation by the present inventor or patent applicant(s) that such documents or devices in any way form part of the common general knowledge in the art.

Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and symbolic representations of operations on data which is either (a) within a computer memory and/or (b) within storage elements in a hardware or mixed hardware/software platform. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that the above and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise, and as apparent from the

following, it will be appreciated that throughout the present specification, discussions utilizing terms such as "obtaining", "commencing", "extracting", "processing", "repeating", "configuring", "identifying", "scanning", "calculating", "determining", "replacing", "generating" "initializing", "outputting", or the like, refer to the action and processes of a computer system, or similar or computationally equivalent electronic device, that manipulates and transforms data represented as physical (electronic) quantities within the registers and memories of the computer system into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

The present specification discloses apparatus for performing the operations of the methods. Such apparatus may be specially constructed for the required purposes, or may comprise a general purpose computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general and/or special purpose machines may be used with programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be appropriate. The structure of a conventional general purpose computer, and of a special purpose computation device, will appear from the description below.

In addition, the present invention also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the preferred method described herein can be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any

particular control flow. There are many other variants of the computer program, which can use different control flows without departing the spirit or scope of the invention. Furthermore one or more of the steps of the computer program may be performed in parallel rather than sequentially.

Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a general purpose computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a general-purpose computer effectively results in an apparatus that implements the steps of the preferred method.

Where reference is made in any one or more of the accompanying drawings to steps and/or features, which have the same reference numerals, those steps and/or features have for the purposes of this description the same function(s) or operation(s), unless the contrary intention appears.

Fig. 1A shows one example of an architecture for performing IPsec processing upon a given packet. A host 101 communicates with an IPsec system 102 via a pair of connections 113 and 114. The IPsec system 102 comprises a packet unloader 116 and a processing unit 118 that communicate via a connection 117. The processing unit 118 communicates with a Security Association Database (SAD) 145 via a connection 140. The host 101 communicates with a Security Policy Database (SPD) 112 via a connection 111.

All packets that are to be processed by the IPsec system 102 are first examined in order to determine which packets require IPsec processing. This determination is performed by either performing a Security Policy Database (SPD) lookup (in the case of a

packet for transmission) based upon selectors that are present in the packet, or determining the presence of an IPsec header (in the case of a received packet).

Incoming packets received via an external network 620 (see **Fig. 6**), (ie received packets), are received by the host 101 which forwards the received packets at high speed to the IPsec processing system 102. The IPsec system 102 processes these packets, at line speed or greater, and sends the processed (received) packets back to the host 101.

Packets for transmission, generated by the host 101 for transmission over the network 620, are forwarded by the host 101 at high speed to the IPsec processing system 102. The system 102 IPsec processes these packets (for transmission), at line speed or greater, and sends the processed packets to the host 101 for forwarding over the network.

Fig. 1B shows a functional block diagram of the IPsec system processing unit 118. An incoming packet 105 is directed to a Packet Loader 100 and a Selector Finder 110 by respective connections 121 and 122. The Packet Loader 100 is connected to a Packet RAM 180 by a connection 123, and to an IPsec processing unit module 160 by a connection 190. The Selector Finder 110 is connected to a SAD lookup interface 130 by a connection 120. The SAD lookup interface 130 is connected to the SAD 145 (see **Fig. 1A**) by the connection 140. The SAD lookup interface is connected to the IPsec protocol processing unit module 160 by a connection 124 and to a cryptographic algorithmic unit 170 by a connection 150. The cryptographic algorithmic unit 170 is described in more detail in regard to **Fig. 7**. The IPsec protocol processing unit module 160 is connected to the cryptographic algorithmic unit 170 by connections 165 and 166, and to the packet RAM 180 by a connection 185. The packet RAM 180 is connected to the Packet Unloader (see **Fig. 1A**) by the connection 117. The IPsec system processing unit 118 can in practice be implemented as an ASIC accelerator chip.

Figs. 2A – 2B depict concurrent process fragments 500 and 500' for performing the partial packet IPsec processing approach. The method 500 is carried out by various elements of the IPsec system 102 (see **Figs. 1A and 1B**).

Fig. 2A shows the process fragment 500 which starts with a START step 501, after which incoming packets (e.g. **105**) that arrive into the system **102** are loaded into the local packet RAM **180** (see **Fig. 1B**) via the Packet Loader **100** in a step 510. The Packet Loader **100** receives all incoming data and buffers, in a small buffer in the Packet Loader 100, the data to be written to memory **180** as the incoming packet (e.g. **105**) is streamed into the system. The incoming packet is written to memory as fast as it arrives at the Packet Loader **100**.

The Packet Loader **100** also outputs the status of the incoming packet onto a bus **190** in a step 520. This status includes the current write pointer of the incoming packet in memory and an indication of whether the whole incoming packet has currently arrived. The incoming packet **105** can arrive at the Packet Loader **100** from the host **101** in segments, resulting in delays before the whole packet has arrived. The packet status on the bus 190 is also available, as depicted by a dashed arrow 512 and a circled letter "A" to the concurrent process fragment 500' in **Fig. 2B**.

Figs. 3A - 3D depict the layout of an IP header **300**, an Authentication Header **350**, an Encapsulating Security Protocol header **360** and the layout of a typical IP packet without IPsec headers **370**, respectively. In an IPsec context, a Security Association Database (SAD) contains Security Associations (SAs). Each SA contains the information required to process an IPsec packet. The SAD must be looked up to locate an SA for each IPsec packet that is required to be processed. The selectors used to lookup a Security Association from the Security Association Database are obtained from the aforementioned headers.

Fig. 4 shows the IP header **300**, the IPsec header **410** (either the Authentication Header **350** or the Encapsulating Security Protocol header **360**) and the IP packet payloads **400** and **420** in the context of a complete IP packet with and without IPsec.

The aforementioned selectors for the Security Association lookup differ for a received packet and a packet for transmission. In the case of a packet for transmission, the selectors consist of the Source IP address **310**, the Destination IP address **315**, the Protocol **305**, the Source Port **330** and the Destination Port **335**. For a received packet the selectors consist of the Source IP address **310**, the Destination IP address **315** and the Security Parameter Index (SPI) **325**.

Returning to **Fig. 2A** the Packet Loader 100 continually tests in a step 530 whether the incoming packet is complete. If not, the process 500 is directed by a NO arrow to a step 540 which continues buffering the packet, and control returns via an arrow 511 to the step 520. If on the other hand buffering is determined by the step 530 to be complete, then the process 500 is directed by a YES arrow to a FINISH step 550.

Fig. 2B shows the process fragment 500' which runs concurrently with the process fragment 500 in **Fig. 2A.** The process fragment 500' commences from the circled letter "A" from **Fig. 2A.** As the data arrives at the Packet Loader **100** it is scanned in a step 555 by the Selector Finder **110**, which continually tests in a step 560 whether the required selectors are available. While the required selectors are not available, i.e. while the selectors are not present in that portion of the incoming packet that has been received, then the process fragment 500' is directed by a NO arrow back to the step 555. Once the required selectors are available, the process fragment 500' is directed by a YES arrow to a step 565.

In the step 565 the Selector Finder **110** is able to obtain the selectors, either from the IP header **300** and the IP payload **400**, or from the IP header **300** and the IPsec header **410** and the following payload **420** if required. A following step 570 transmits the

selectors sequentially over a bus **120** to a device which provides an interface **130** to the Security Association Database (SAD) **145** via a bus **140**. The SAD is controlled by an SAD management unit (not shown) which performs the actual SAD lookup using selectors sent by the SAD lookup interface 130. The Selector Finder **110** is informed, by a message processor (not shown) in the IPsec system 102, of whether the packet is a packet for transmission, or alternately, is a received packet, and then further determines which selectors are required and also which selectors are available so the correct selectors can be obtained from the incoming packet and provided for an SAD search **145**.

The selectors, which are described above, are used to perform an SAD lookup in a step 575 at the SAD management unit via the interface **130** and the bus **140**. As the selectors are received in the IPsec system 102 at the start of the incoming packet 105, the SAD lookup can occur while the incoming packet is still being written to the packet RAM 180.

To perform a lookup, the SAD lookup interface **130** must first gain access to the arbitrated SAD request bus **140**. Once access is granted, the interface **130** can transmit the selectors over the bus **140** in set transactions, and await the result of the SAD lookup. The composition of these set transactions vary depending on whether the incoming packet is a received packet or a packet for transmission, and the set transactions contain the selectors and any other information required for the SAD lookup.

The result of the SAD lookup by the SAD management unit **145** contains the data stored within the Security Association (SA) entry matched by the selectors. This information is transmitted in pre-defined data structures from the SAD management unit **145** to the SAD lookup interface **130**. The SAD lookup interface **130** streams the received data structures in a step 580 to the cryptographic algorithmic unit **170** and to the IPsec processing modules **160** over a bus **150**.

The cryptographic algorithmic unit **170** uses the transmitted data to configure itself, in a step 585, with the algorithms to be used, the cryptographic keys to be used, and the mode in which the selected algorithms are to run. The data that is used to configure the cryptographic algorithms within the cryptographic algorithmic unit **170** is obtained from the data structures transmitted on the bus **150** from the SAD lookup interface **130**.

The IPsec processing units **160**, which typically include an ESP unit **155** and an AH unit, configure themselves, in a step 590, with the IPsec protocol specific information and algorithmic information. This information includes whether the SA is in transport or tunnel mode, the length of the Authentication Data **345**, which IPsec processing unit **160** should process the incoming packet (eg, ESP or AH) and the block size of the algorithms. The data that is used to configure the IPsec processing units **160** is obtained from the data structures transmitted on the bus **150** from the SAD lookup interface **130**.

The IPsec processing units **160** can now start processing, in a step 595, the IPsec header and the buffered section of the incoming packet obtained from memory 180 via the link 185. All data to be processed by the cryptographic algorithmic unit **170** can now be passed to the cryptographic algorithmic unit **170** via the link **165** by the IPsec processing units **160**. The resultant data from the cryptographic and IPsec processing can be written back, in a step 598, to memory **180** via the link **185**.

Figs. 5A – 5D show, having regard to the packet RAM 180 in **Fig. 1B**, at various stages during the arrival of an incoming packet according to the disclosed partial packet IPsec processing arrangement, a memory segment **200** that has been allocated for the entire incoming packet being considered, the memory **210** that has been taken up by that proportion **210** of the incoming packet that has been received to the point of time in question, and the memory **220** to which data processed from that section **210** has been written.

The IPsec processing units **160** make sure that the memory consumed by the processed section **220** of the incoming packet does not pass the leading edge **211** of the arrived section of the incoming packet **210**. The IPsec processing units **160** ensure that this does not occur during step 595 by monitoring the aforementioned status information on the bus **190** from the Packet Loader **100**. The aforementioned information includes whether the whole incoming packet has arrived and the current write pointer of the incoming packet in memory. The whole incoming packet has arrived and can be fully processed, as shown in **Fig. 5D**, when the arrived section of the incoming packet **210** is equal to size of the allocated memory **200** or conforms to the packet total length **320** contained in the IP header **300**. As the incoming packet arrives and is processed, as shown in **Figs. 5A – 5D**, the arrived section **210** must always be greater than the processed section **220**.

Fig. 6 shows how the method of partial packet IPsec processing can be practiced using a general-purpose computer system 600, such as that shown in **Fig. 6** wherein the processes of **Figs. 2A-2B** may be implemented as software, such as an application program executing within the computer system 600. In this arrangement the IPsec system 102 of **Figs. 1A** and **1B** are implemented, as one or more software applications, in the host machine 101.

In this arrangement, the steps of method of partial packet IPsec processing are effected by instructions in the software that are carried out by the computer. The instructions may be formed as one or more code modules, each for performing one or more particular tasks. The software may also be divided into two separate parts, in which a first part performs the partial packet IPsec processing methods and a second part manages a user interface between the first part and the user. The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer from the computer readable medium,

and then executed by the computer. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer preferably effects an advantageous apparatus for partial packet IPsec processing.

The computer system 600 is formed by the host computer module 101, input devices such as a keyboard 602 and mouse 603, output devices including a printer 615, a display device 614 and loudspeakers 617. A Modulator-Demodulator (Modem) transceiver device 616 is used by the computer module 101 for communicating to and from a communications network 620, for example connectable via a telephone line 621 or other functional medium. The modem 616 can be used to obtain access to the Internet, and other network systems, such as a Local Area Network (LAN) or a Wide Area Network (WAN), and may be incorporated into the computer module 101 in some implementations.

The computer module 101 typically includes at least one processor unit 605, and a memory unit 606, for example formed from semiconductor random access memory (RAM) and read only memory (ROM). The module 101 also includes an number of input/output (I/O) interfaces including an audio-video interface 607 that couples to the video display 614 and loudspeakers 617, an I/O interface 613 for the keyboard 602 and mouse 603 and optionally a joystick (not illustrated), and an interface 608 for the modem 616 and printer 615. In some implementations, the modem 616 may be incorporated within the computer module 101, for example within the interface 608. A storage device 609 is provided and typically includes a hard disk drive 610 and a floppy disk drive 611. A magnetic tape drive (not illustrated) may also be used. A CD-ROM drive 612 is typically provided as a non-volatile source of data. The components 605 to 613 of the computer module 101, typically communicate via an interconnected bus 604 and in a manner which results in a conventional mode of operation of the computer

system 600 known to those in the relevant art. Examples of computers on which the described arrangements can be practised include IBM-PC's and compatibles, Sun Sparcstations or alike computer systems evolved therefrom.

Typically, the application program is resident on the hard disk drive 610 and read and controlled in its execution by the processor 605. Intermediate storage of the program and any data fetched from the network 620 may be accomplished using the semiconductor memory 606, possibly in concert with the hard disk drive 610. In some instances, the application program may be supplied to the user encoded on a CD-ROM or floppy disk and read via the corresponding drive 612 or 611, or alternatively may be read by the user from the network 620 via the modem device 616. Still further, the software can also be loaded into the computer system 600 from other computer readable media. The term "computer readable medium" as used herein refers to any storage or transmission medium that participates in providing instructions and/or data to the computer system 600 for execution and/or processing. Examples of storage media include floppy disks, magnetic tape, CD-ROM, a hard disk drive, a ROM or integrated circuit, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 101. Examples of transmission media include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

The method of partial packet IPsec processing is preferably implemented in dedicated hardware such as one or more integrated circuits performing the functions or sub functions of partial packet IPsec processing. Such dedicated hardware may include graphic processors, digital signal processors, or one or more microprocessors and associated memories.

Fig. 7 depicts the cryptographic algorithmic unit 170 of **Fig. 1B** in more detail. The cryptographic algorithmic unit 170 comprises a controller 250, shown in two parts, each of which is surrounded by a dashed box, and supports N sets of algorithms. Each algorithm set (numbered 1 to N in **Fig. 7**) comprises several modules, each module implementing a mode of operation of the algorithm. Illustrative modes are represented in **Fig.7** as encryption / decryption, authentication, and a generic mode labelled as "Type C". Thus, for example, a first algorithm set 254 comprises an encryption/decryption module 251, an authentication module 252, and a generic Type C module 253.

During the configuration step 585 (see **Fig. 2B**), the cryptographic algorithmic unit controller 250 receives the transmitted configuration data via the connection 150 and sets up the associated data paths within the cryptographic algorithmic unit 170. The cryptographic algorithmic unit controller 250 then sends the cryptographic keys to the selected algorithm modules and sets the selected algorithm modules to the correct mode.

Once the cryptographic algorithmic unit 170 has been configured and the data path has been set, the cryptographic algorithmic unit controller 250 controls the data flow of data from the protocol processing units 160 (see **Fig. 1B**) via the input path 165 to the appropriate algorithm module and thence back to the protocol processing units 160 via the output path 166.

Industrial Applicability

It is apparent from the above that the arrangements described are applicable to the data processing industry.

The foregoing describes only some embodiments of the present invention, and modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the embodiments being illustrative and not restrictive.

The claims defining the invention are as follows:

1. A method of performing IPsec processing of an incoming communication packet, the method comprising the steps of:

determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

obtaining said sufficient information from the received portion of the packet; and

commencing IPsec processing of said packet before the entire packet has been received depending upon the obtained information.

2. A method according to claim 1, wherein:

said information comprises sufficient selectors to support a search of a Security Association Database;

the obtaining step comprises obtaining said selectors from the received portion of the packet; and

the IPsec processing step comprises the steps of:

extracting, depending upon said selectors, a Security Association from corresponding Security Association entries in said Security Association Database; and

processing said packet depending upon the extracted Security Association.

3. A method according to claim 1, wherein the determining step is repeated for at least some successively received portions of the packet.

4. A method according to claim 1, wherein the IPsec processing step comprises the steps of:

configuring a processing device depending upon said information, and

IPsec processing said packet using the configured device.

5. A method according to claim 1 comprising, prior to the determining step, a step of (a) performing a Security Policy Database lookup dependent upon selectors in the packet or (b) determining presence of an IPsec header in the packet, to thereby determine if the packet is to be processed according to the method in claim 1.

6. A method according to claim 1, wherein the IPsec processing is performed at a speed at least equal to the line speed at which the incoming packet is received.

7. A method according to claim 1, wherein:

prior to the determining step the method comprises the step of identifying that the packet is one of a transmit packet and a receive packet; and

the determining step is performed dependent upon the identifying step.

8. An apparatus for performing IPsec processing of an incoming communication packet, the apparatus comprising:

means for determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

means for obtaining said information from the received portion of the packet; and

means for IPsec processing said packet before the entire packet has been received depending upon the obtained information.

9. An apparatus for performing IPsec processing of an incoming communication packet, the apparatus comprising:

a processor; and

a program for directing the processor to perform the steps of:

determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

obtaining said information from the received portion of the packet; and

commencing IPsec processing of said packet before the entire packet has been received depending upon the obtained information.

10. A computer program product including a computer readable medium having recorded thereon a computer program for directing a processor to execute a method of performing IPsec processing of an incoming communication packet, the program comprising:

code for determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

code for obtaining said information from the received portion of the packet; and

code for IPsec processing said packet before the entire packet has been received depending upon the obtained information.

11. A computer program for directing a processor to execute a method of performing IPsec processing of an incoming communication packet, the program comprising:

code for determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

code for obtaining said information from the received portion of the packet; and

code for IPsec processing said packet before the entire packet has been received depending upon the obtained information.

12. A communication packet formed by a method of performing IPsec processing of an incoming communication packet, the method comprising the steps of:

determining, from a received portion of the packet, if sufficient information has been received to enable said IPsec processing to be commenced;

obtaining said information from the received portion of the packet; and

commencing IPsec processing of said packet before the entire packet has been received depending upon the obtained information.

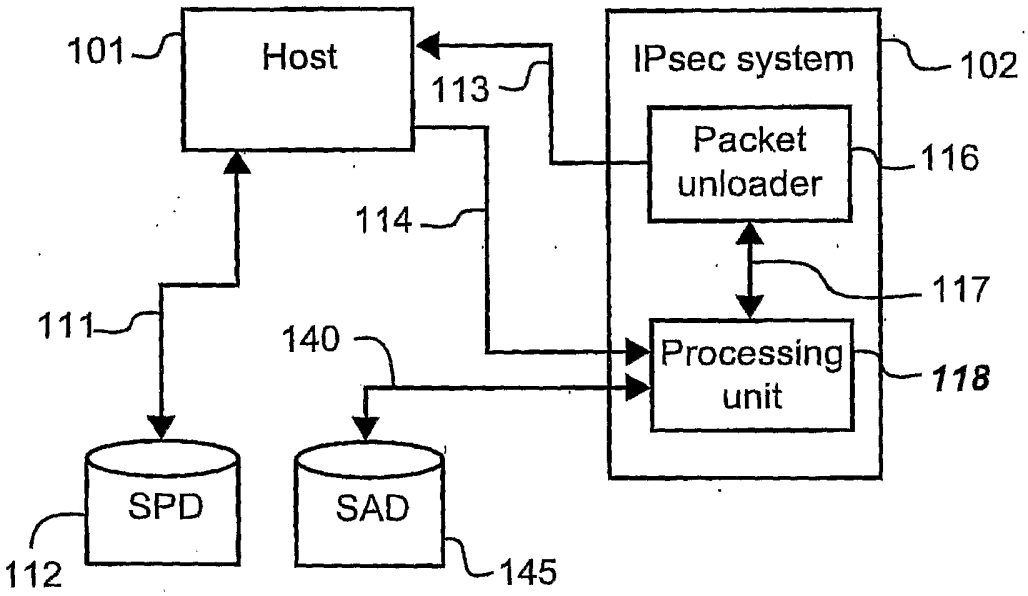

101
Host
113
IPsec system
102
Packet unloader
116
114
117
111
140
Processing unit
118
SPD
SAD
112
145

Fig. 1A

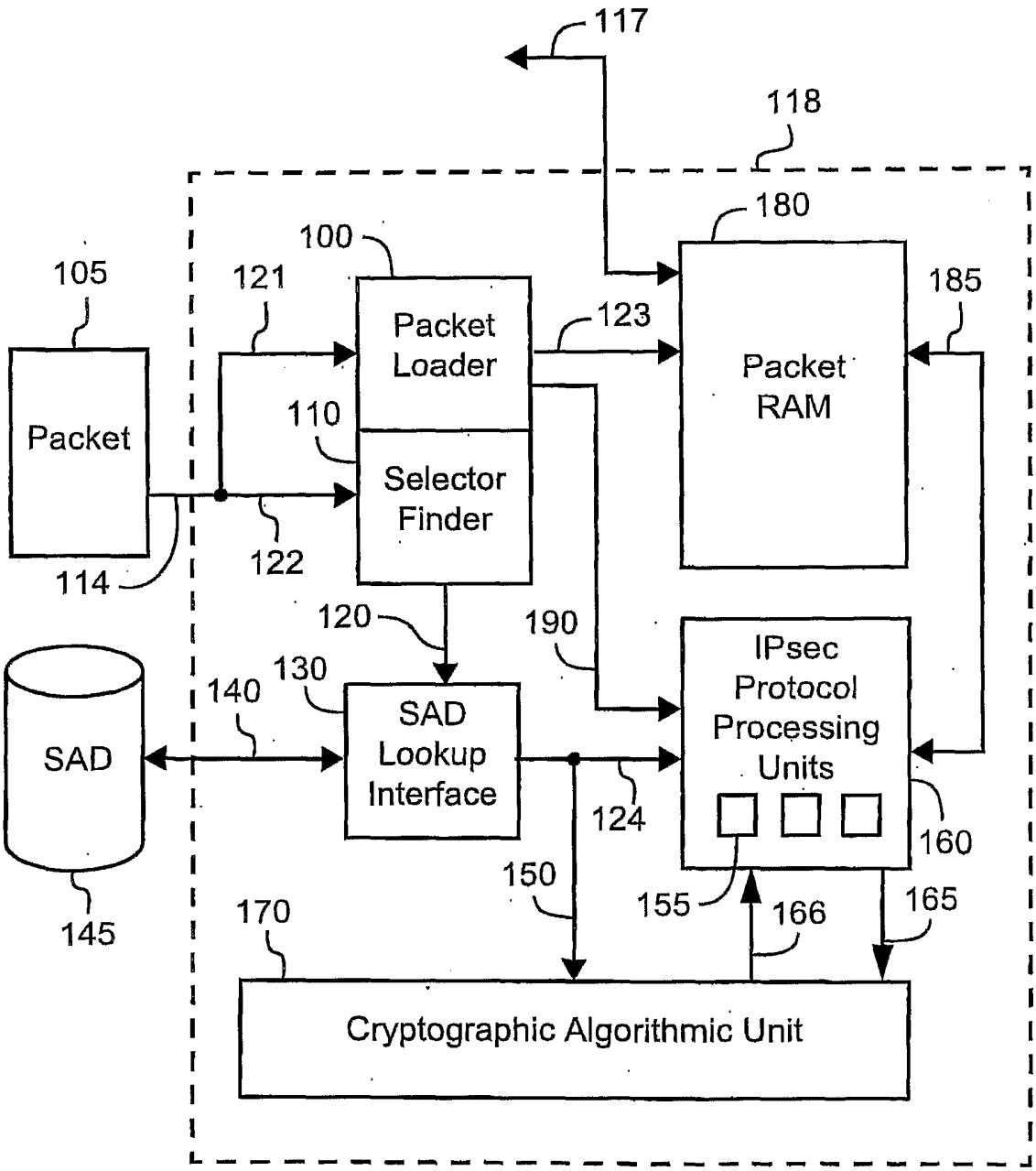

117
118
180
100
105
121
Packet Loader
123
Packet RAM
185
Packet
110
Selector Finder
114
122
120
190
IPsec Protocol Processing Units
130
140
SAD Lookup Interface
SAD
124
160
150
155
166
165
145
170
Cryptographic Algorithmic Unit

Fig. 1B

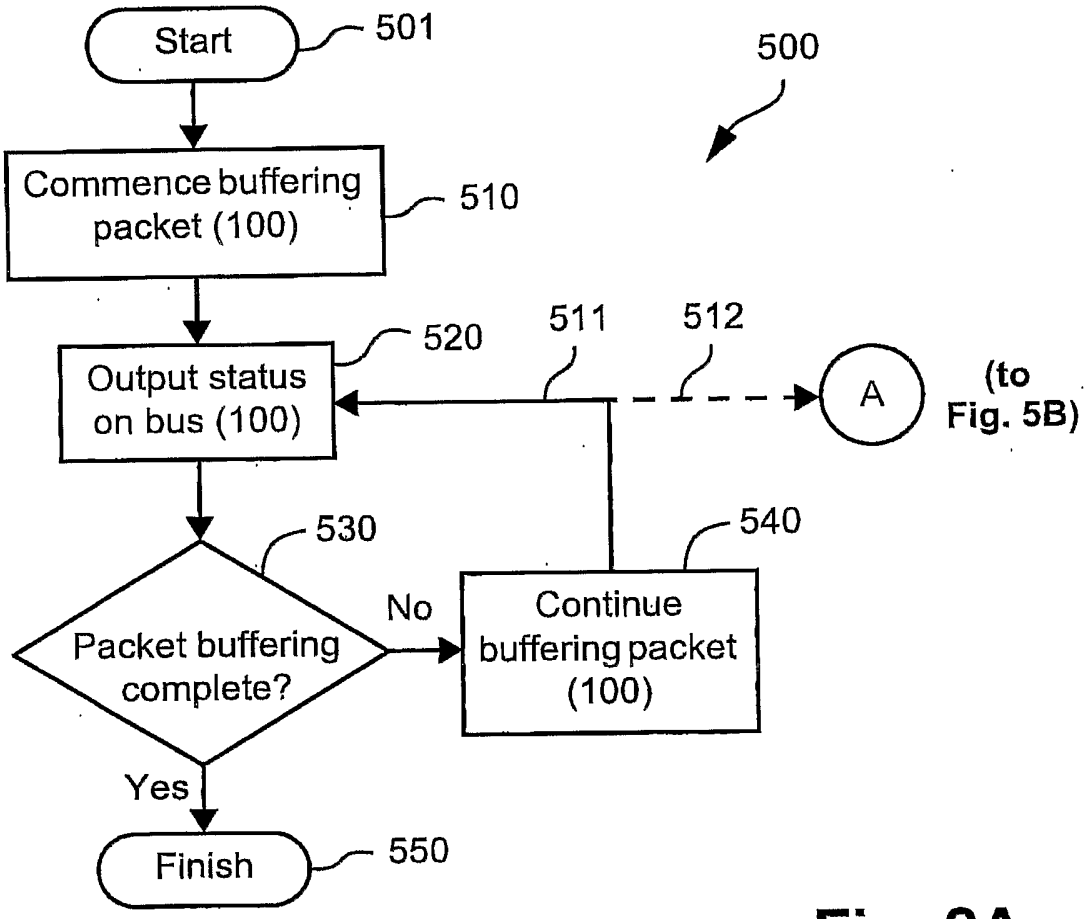

Start
501
500
Commence buffering packet (100)
510
Output status on bus (100)
520
511
512
A
(to Fig. 5B)
530
Packet buffering complete?
No
Continue buffering packet (100)
540
Yes
Finish
550

Fig. 2A

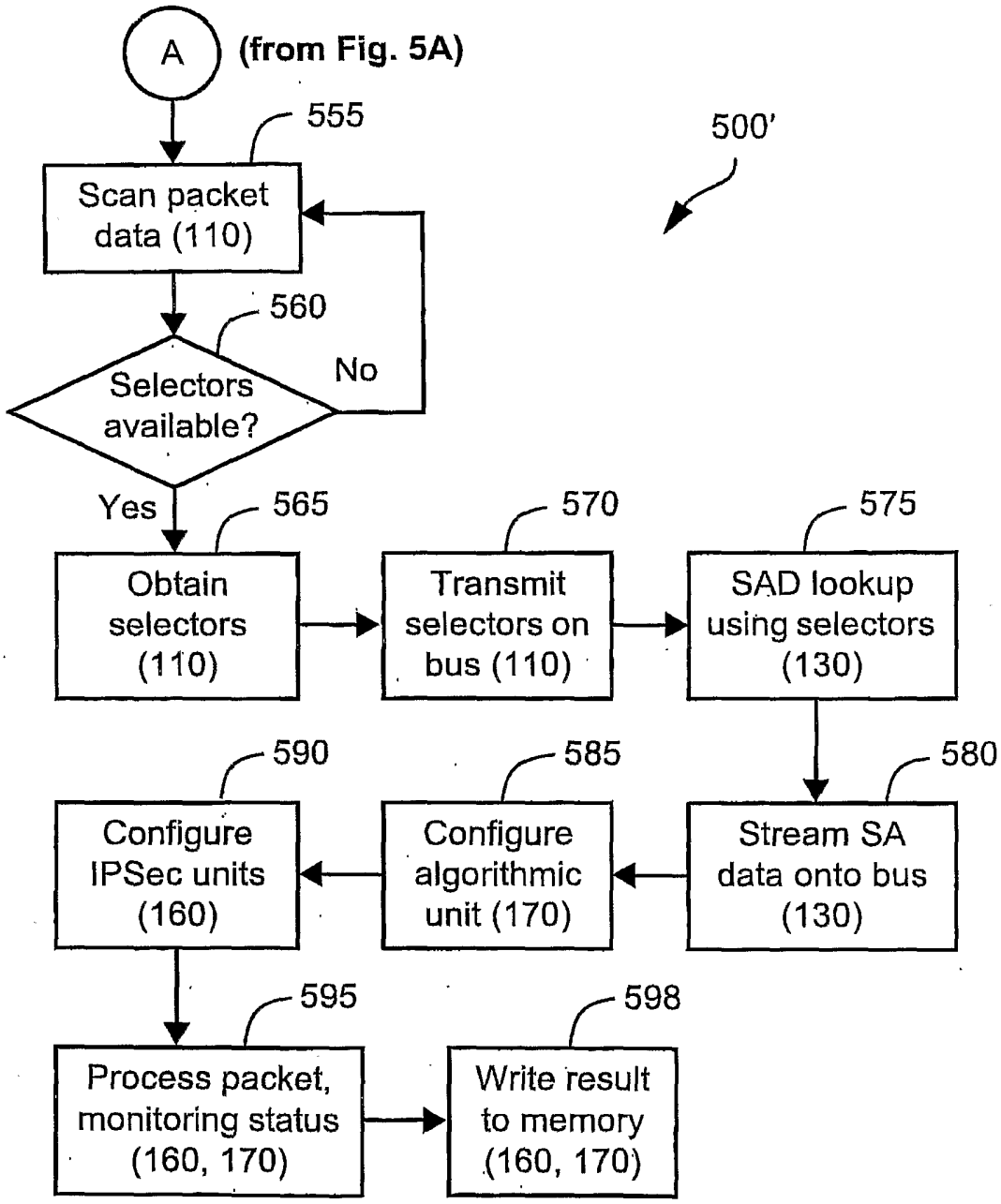

A
(from Fig. 5A)
555
500'
Scan packet data (110)
560
Selectors available?
No
Yes
565
570
575
Obtain selectors (110)
Transmit selectors on bus (110)
SAD lookup using selectors (130)
590
585
580
Configure IPSec units (160)
Configure algorithmic unit (170)
Stream SA data onto bus (130)
595
598
Process packet, monitoring status (160, 170)
Write result to memory (160, 170)

Fig. 2B

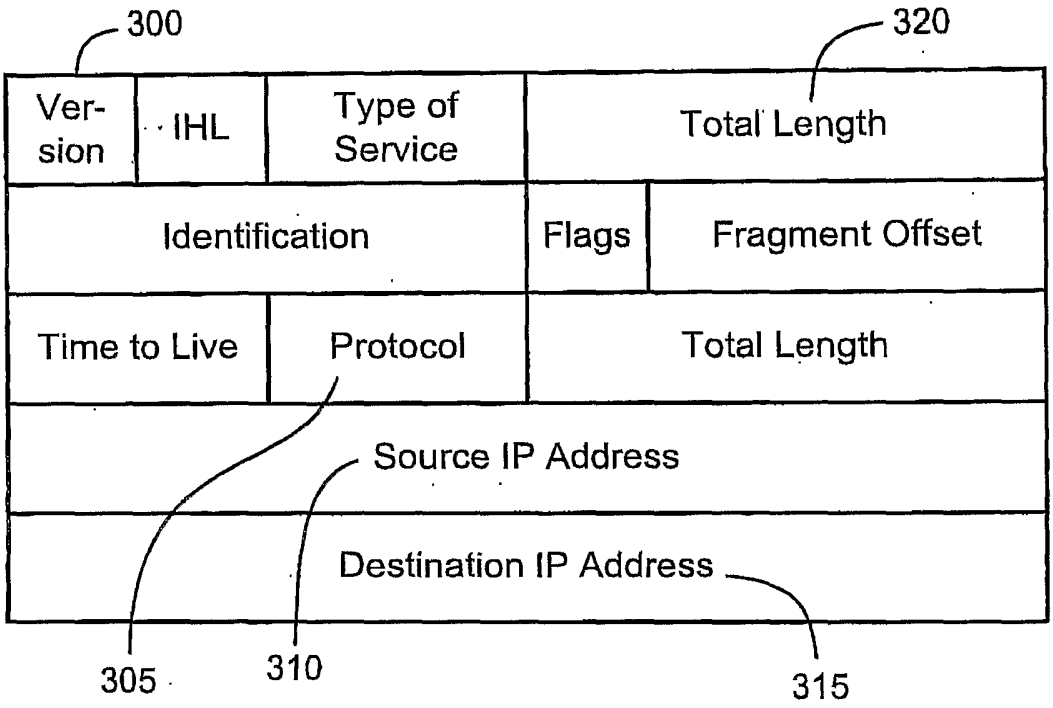

300
320
Ver-sion
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Total Length
Source IP Address
Destination IP Address
305
310
315

Fig. 3A

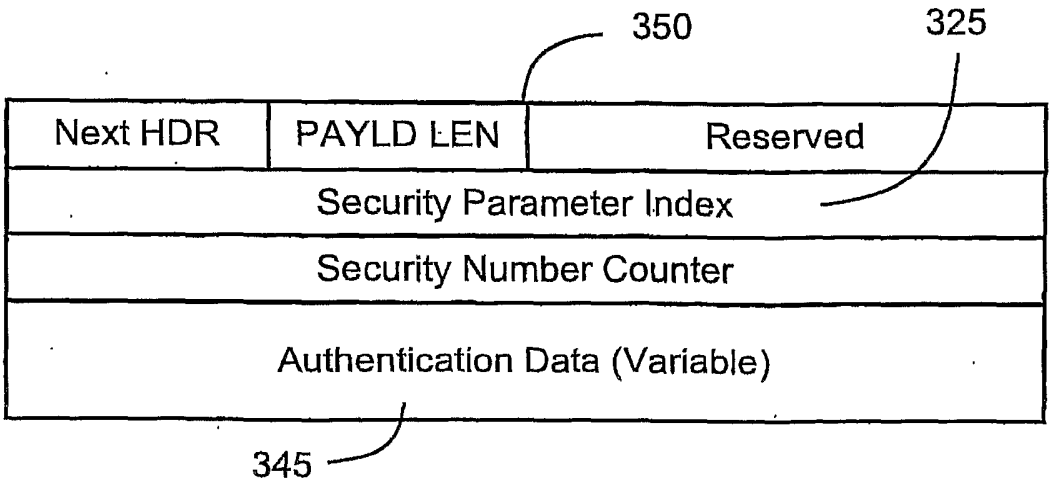

350
325
Next HDR
PAYLD LEN
Reserved
Security Parameter Index
Security Number Counter
Authentication Data (Variable)
345

Fig. 3B

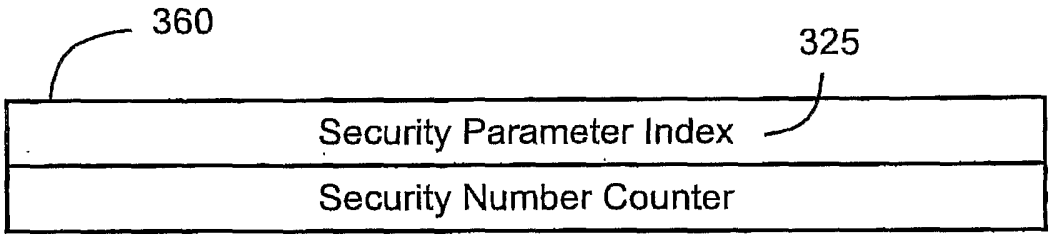

360
325
Security Parameter Index
Security Number Counter

Fig. 3C

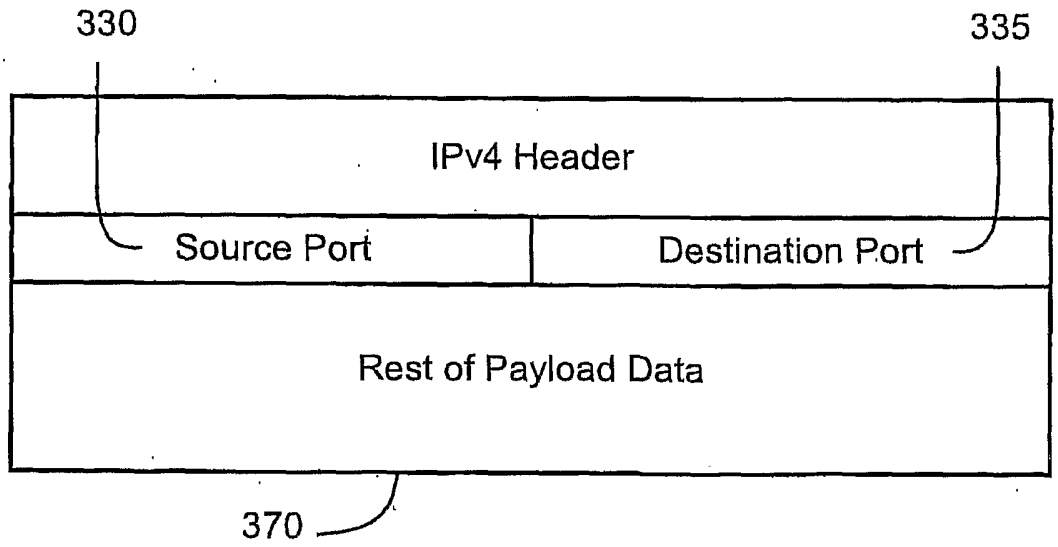

330
335
IPv4 Header
Source Port
Destination Port
Rest of Payload Data
370

Fig. 3D

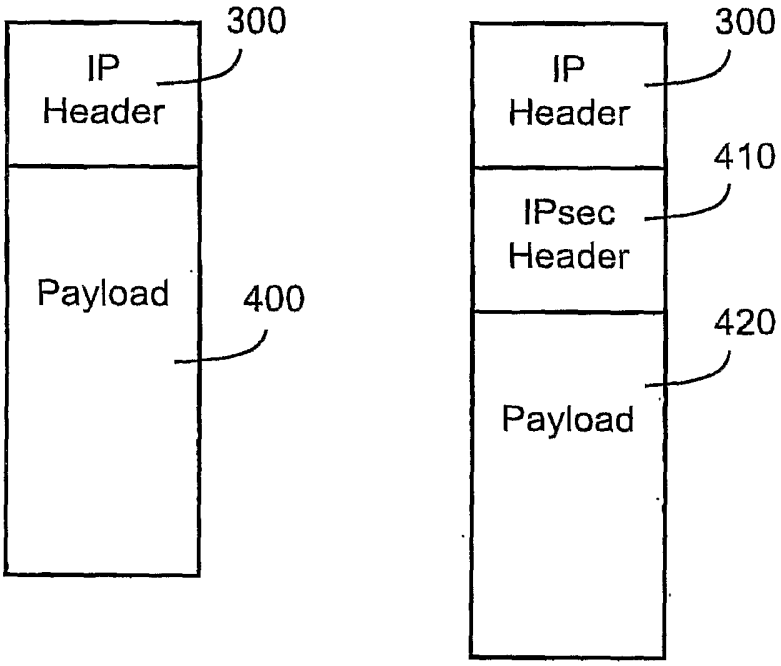

300
IP
Header
Payload
400
300
IP
Header
410
IPsec
Header
420
Payload

Fig. 4

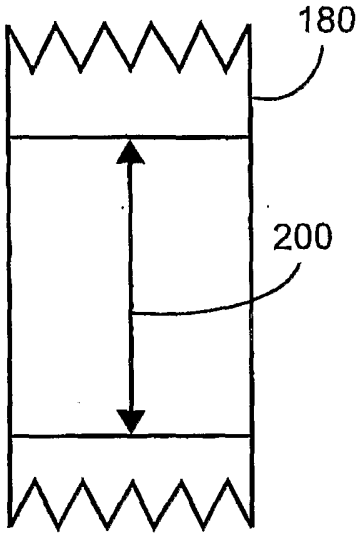


Fig. 5A

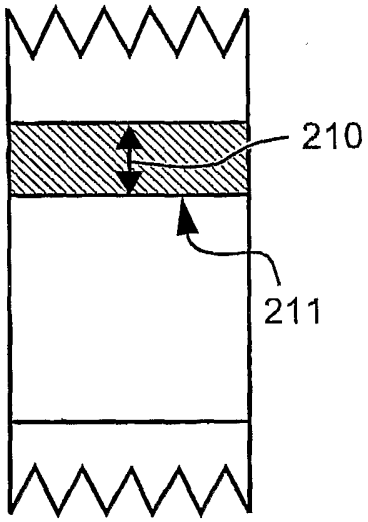


Fig. 5B

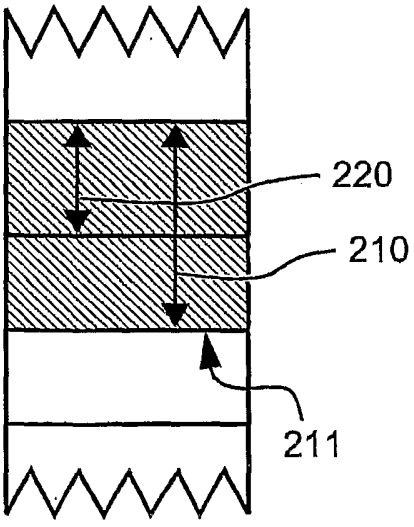


Fig. 5C

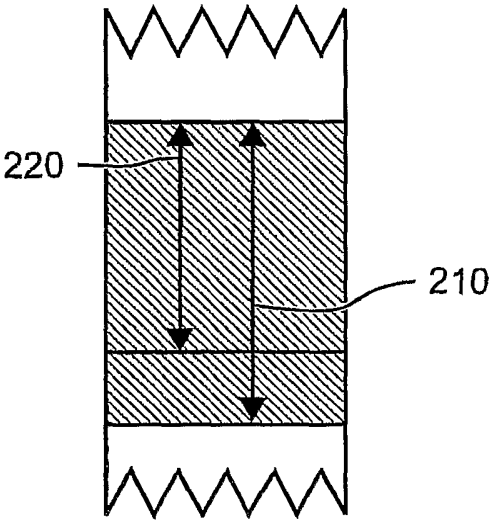


Fig. 5D

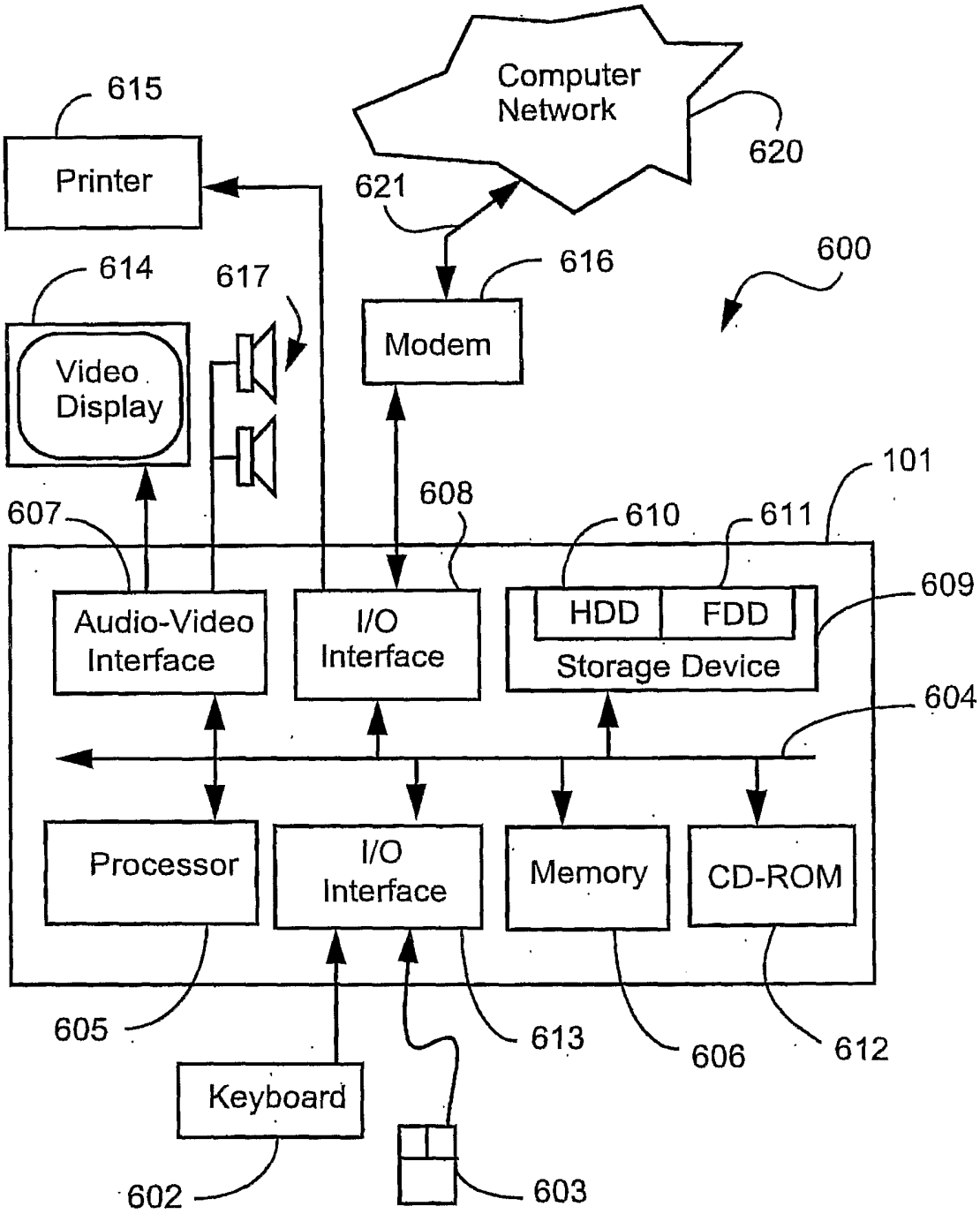

Computer Network
620
615
Printer
621
600
614
617
616
Modem
Video Display
101
607
608
610
611
609
HDD
FDD
Audio-Video Interface
I/O Interface
Storage Device
604
Processor
I/O Interface
Memory
CD-ROM
605
613
606
612
Keyboard
602
603

Fig. 6

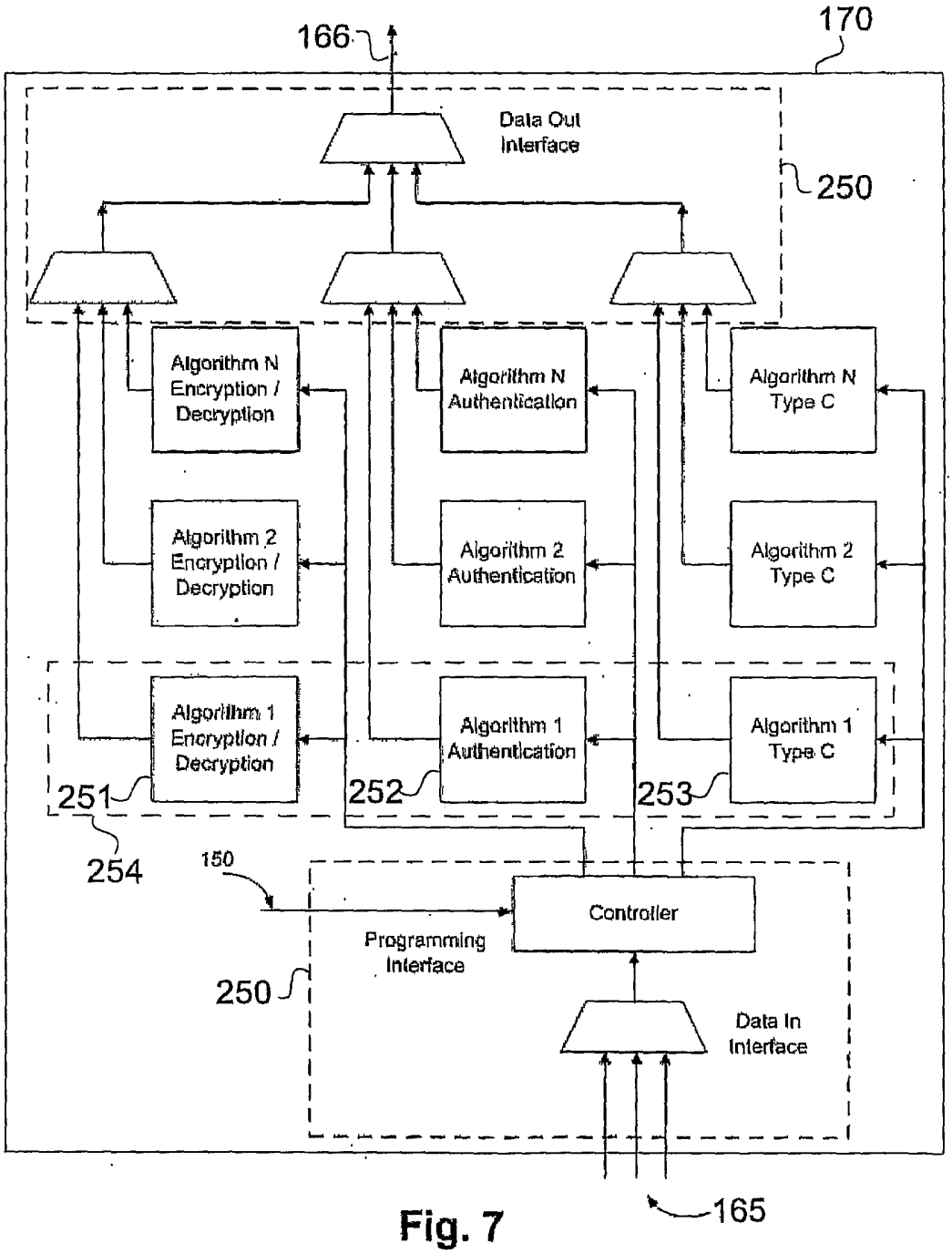

166
170
Data Out Interface
250
Algorithm N Encryption / Decryption
Algorithm N Authentication
Algorithm N Type C
Algorithm 2 Encryption / Decryption
Algorithm 2 Authentication
Algorithm 2 Type C
Algorithm 1 Encryption / Decryption
Algorithm 1 Authentication
Algorithm 1 Type C
251
252
253
254
150
Controller
Programming Interface
250
Data In Interface
165

Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU2005/001584

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl.

H04L 9/00 (2006.01) ***H04L 29/06*** (2006.01)

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)

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)
WPAT, USPTO, Esp@cenet, IEEExplore: packet, IPsec, process, receive, incoming, speed, performance, selector, partial and similar terms

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, A	US 2005/0102525A1 (AKIMOTO), 12 May 2005 whole document	
P, A	US 2004/0205332A1 (BOUCHARD et al), 14 October 2004 whole document	
P, A	US 2004/0205336A1 (KESSLER et al), 14 October 2004 whole document	

☐ 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 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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 December 2005	22 DEC 2005
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA E-mail address: pct@ipaustralia.gov.au Facsimile No. (02) 6285 3929	Authorized officer **MANISH RAJ** Telephone No : (02) 6283 2175

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/AU2005/001584

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member			
US	2005102525	EP	1523149	JP	2005117246
US	2004205332	WO	2004093373		
US	2004205336	WO	2004092930		

Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.

END OF ANNEX