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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 October 2006 (19.10.2006)

PCT

(10) International Publication Number
WO 2006/108808 A1

(51) International Patent Classification:
H04L 29/12 (2006.01) *H04L 29/06* (2006.01)

(21) International Application Number:
PCT/EP2006/061443

(22) International Filing Date: 7 April 2006 (07.04.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/907,659 11 April 2005 (11.04.2005) US

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

(71) Applicant *(for MG only)*: **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, Portsmouth Hampshire PO6 3AU (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **JAKUBIK, Patricia** [US/US]; 5704 Crutchfield Road, Raleigh, North Carolina 27606 (US). **OVERBY JR., Linwood Hugh** [US/US]; 7252 Manor Oaks Drive, Raleigh, North Carolina 27615 (US). **PORTER, Joyce Anne** [US/US]; 1007 West Saint Julian Place, Apex, North Carolina 27502 (US). **WIERBOWSKI, David John** [US/US]; 2489 East Beecher Hill Road, Owego, New York 13827 (US).

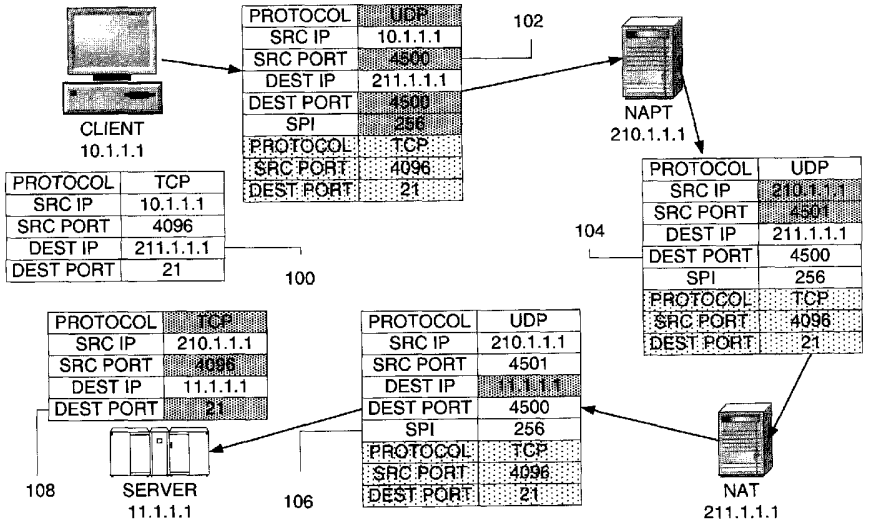
(74) Agent: **ROBERTS, Scott**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, 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,

[Continued on next page]

(54) Title: PREVENTING DUPLICATE SOURCES FROM CLIENTS SERVED BY A NETWORK ADDRESS PORT TRANSLATOR



(57) Abstract: Preventing duplicate sources on a protocol connection that uses network addresses, protocols and port numbers to identify connections that include port number translation. In response to an inbound IPsec packet from a remote source client, a determination is made as to whether or not a port number is available within a range of port numbers that comply with a security association governing the connection. If so, an available port number is assigned to the connection, thereby avoiding a possibility of a duplicate source. If a port number is not available, the packet is rejected.

WO 2006/108808 A1

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*

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.

PREVENTING DUPLICATE SOURCES FROM CLIENTS SERVED BY A NETWORK ADDRESS PORT TRANSLATOR

DESCRIPTION

Technical Field

The present invention relates generally to internet networking and specifically to addressing conflicts caused by network address and port translation.

Background

Problems and solutions are described herein in terms of the Internet and the TCP/IP protocols that form the basis of Internet communications. However, the invention can apply to other communication protocols as well, depending on the specifics of the protocols.

Internet Network Address Translation is used for several reasons. The main reason is to economize on the use of public addresses. The Internet Protocol (IP) address of a Network Address Translator (NAT) is generally a public address. That is, the NAT IP address is known to the outside world, while all of the servers or clients behind the NAT are private addresses, unknown to the outside world. In such a case, the outside world communicates with the NAT and the NAT controls the communications with the appropriate servers and clients behind it. This means that the IP addresses of devices behind the NAT only have to be unique within that family, but can be duplicative of other IP addresses in the rest of the world. NATs involve only the translation of IP addresses. There is a further type of translation known as Network Address Port Translation (NAPT) in which both IP addresses and port numbers are translated. The standards for Network Address Translation (NAT) and Network Address Port Translation (NAPT) are set forth in the Internet Engineering Task Force (IETF) RFC 3022, entitled "Traditional IP Network Address Translation".

The original Internet was not designed with security as a primary factor. In fact, the Internet was purposely made relatively open as an aid to scientific and educational communication. However, the advent of the Web and its commercial uses has increased the need for secure Internet communications. The Internet Security Protocol, commonly known as IPsec,

was defined to address these issues. For example, IPsec provides for the authentication of network devices and/or for the encryption of transmitted data. An IPsec communication between source and destination addresses is administered in accordance with a security association (SA); an SA is one or more rules that define the IPsec processing applied to the communication. IPsec is defined in RFC 2401 and other RFCs. Whether a packet is denied, permitted without Ipsec processing or permitted with Ipsec processing is determined by matching the attributes of a packet with the security rules in a security policy database (SPD). To make this determination the known art searches both static and dynamic rules in the order of most specific to least specific attributes for both outgoing and incoming packets. A set of static rules is essentially a security policy. Static rules are predefined and generally do not change very often. Dynamic rules are rules that are negotiated between nodes during IKE (Internet Key Exchange) processing as needed and added to the security policy database. U.S. patent 6,347,376 to International Business Machines describes a preferred method of searching the dynamic rules of an SPD. This patent is incorporated herein by reference in its entirety.

There are inherent incompatibilities between network address or port translation and IPsec processing. These incompatibilities are a barrier to deployment of IPsec. RFC 3715 recognizes and discusses some of these incompatibilities, but offers no general solutions. For example, Section 4.1 of RFC 3715 refers to a limited solution proposed in RFC 3456, "Dynamic Host Configuration Protocol (DHCPv4, Configuration of IPsec Tunnel Mode"), but states that a more general solution is needed. In addition, Section 5 of RFC 3948 entitled "*UDP Encapsulation of IPsec Packets*" from the IPsec working group of IETF also addresses some of the incompatibility problems. Particularly, Section 5.2 of RFC 3948 describes briefly a problem in determining what IPsec security associations to use on connections to clients served by a NAPT. This Section also describes another problem in allowing a clear text connection to a client behind a NAPT when the NAPT also handles IPsec traffic.

Thus there is a need to address the problem of avoiding duplicate sources when clients are served by a NAPT. No solutions are provided for this problem by any of the related IETF RFCs. For purposes of this specification, duplicate sources are defined as packets having the same source addresses (e.g., an IP address of a NAPT assigned to an IPsec encapsulated original packet), the same transport protocol and the same original source port number (i.e. port number in the transport header of the IPSec encapsulated packet).

Duplicate sources results in duplicate connections that breech network integrity. For example, packets can be sent to the wrong destination.

RFC 3947, entitled "*Negotiation of NAT-Traversal in the IKE*" , describes what is needed in the IKE (Internet Key Exchange) phases 1 and 2 for the NAT traversal support. This includes detecting if both ends in a packet communication support NAT traversal, and detecting if there is one or more NATs along the path from host to host. It also covers how to negotiate the use of User Datagram Protocol (UDP) encapsulated IPsec packets in the IKE Quick Mode and describes how to transmit an original source IP address to the other end if needed". The UDP is defined in RFC 768. RFC 3948, "*UDP Encapsulation of IPsec Packets*", defines methods to encapsulate and decapsulate ESP (Encapsulating Security Payload) packets inside of UDP packets for the purpose of traversing NATs. ESP is defined in RFC 2406. ESP is designed to provide a mix of security services in IPv4 and IPv6.

Summary of the Invention

The present invention is directed to preventing duplicate sources of packets in connections that use source addresses, protocols and source port numbers to identify source applications that are served by a NAPT. When a packet is received at a server, a determination is made as to whether the packet is a UDP packet that encapsulates an ESP packet whose transmission path contains a network address translator (NAPT). If these determinations are met, the original packet is decapsulated to obtain the original source port number and the original transport protocol. A determination is made as to whether a port number is available within a range of port numbers that comply with a security association governing the connection. If an available port is found, it is assigned to the connection, thereby avoiding a possibility of a duplicate source. If no available port number is available within the range of port numbers governing the connection, the packet is rejected.

Brief Description of the Drawings

A preferred embodiment of the present invention will now be described by way of example only with reference to the following drawings in which:

Fig. 1 shows a packet progressing from a client, through a NAPT and a NAT to a destination host and the changes to the packet headers and contents as the packet progresses;

Fig. 2 shows a return packet responsive to the packet of Fig. 1;

Fig. 3 shows an illustrative embodiment of the Source Port Translation Table (SPTT);

Fig. 4 shows a NAPT translated packet that encapsulates an encrypted original packet;

Fig. 5 shows the packet of Fig 4 after decryption;

Figs. 6 and 7 correspond to Figs. 4 and 5, respectively, and show a second packet on the same path as the earlier packet that represents an illegal duplicate connection caused by the inclusion of a NAPT in the transmission path;

Fig. 8 is an illustrative flowchart of the Internet Key Exchange protocol whereby a security association is defined and a range of port numbers are determined, both for installation into a protocol stack for communications between this destination and a client;

Fig. 9 is a flowchart showing options that are available when an inbound packet first arrives at a destination host;

Fig. 10 is a flowchart showing at entry A the processing of an inbound packet that both encapsulates an encrypted original packet and has passed through an NAPT; at entry B the processing of an IPsec packet that has not passed through a NAPT, and at entry D has not passed through a NAPT and is not an IPsec packet; and

Fig. 11 continues the processing of an inbound packet from Fig. 10;

Fig. 12 is a flowchart showing an alternative way of processing inbound packets that do not satisfy both conditions of encapsulation and passing through an NAPT; and

Fig. 13 shows an illustrative embodiment of an Available Source Port pool that keeps track of assigned and unassigned port numbers.

Detailed Description

This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art. Although the problems addressed exist for both transport mode and tunnel mode in Internet transmissions, the disclosed embodiment is directed primarily to transport mode. A small variation that is described adapts the transport mode disclosure for operation in tunnel mode.

Embodiments of the present invention can be implemented in software, hardware or hardware and software. As will be appreciated by those of

skill in the art, the embodiments of the present invention can take the form of an entirely hardware embodiment, an entirely software (including firmware, resident software, micro-code, etc.) embodiment, or an embodiment containing both software and hardware aspects. Furthermore, embodiments of the present invention can take the form of a computer program product on a computer-usable or computer-readable storage medium having program code means embodied in the medium for use by or in connection with a computer or any instruction execution system. In the context of this document, a computer-usable or computer-readable medium can be any means that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device. The medium can be, for example, but is not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples (a non exhaustive list) of the computer-readable medium would include an electrical connection having one or more wires, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, and a portable compact disc read-only memory (CD-ROM). Note that the computer-usable or computer-readable medium could even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via, for instance, optical scanning of the paper or other medium, then compiled, interpreted, or otherwise processed in a suitable manner if necessary, and then stored in a computer memory.

In this description, like numbers refer to like elements throughout.

IPsec processing can be used to authenticate or to encrypt the contents of packets for security purposes. Authentication and encryption can both be applied to a packet or either can be applied separately. To simplify this presentation, the description of IPsec processing discusses the encapsulation and decapsulation of the packets in terms of encryption and decryption. The processing described is equally valid if authentication is being applied either alone or in conjunction with encryption.

When IPsec processing is applied to outgoing packets from a source client, the processing encrypts the original source and destination ports and the protocol field and encapsulates this encrypted material into a UDP packet. The original client source IP address is retained in the UDP packet, but the source port number is set to 4500 as prescribed by the RFC 3948 "UDP Encapsulation of IPsec ESP packets". If the UDP packet then

passes through a NAPT, the NAPT performs further transformations. These transformations are described in detail below with respect to Figs. 1 and 2. Specifically, the NAPT substitutes it's own IP address for the client source IP address, assigns a new unique port number to the UDP header and keeps track of these translations so that return packets can be mapped to the original source. RFC 3948 describes a scheme in which the original source port number in a TCP or UDP packet is not changed by the NAPT device, since it is part of the original transport header that is now encrypted as part of the IPsec ESP payload. The port number in the UDP header that is added for UDP encapsulation is changed instead as mentioned above. When such an IPsec packet is received by a server and decrypted, the original source and destination ports of the packet are revealed. For packets that are not processed through IPsec, the NAPT device translates the original source IP address and source port. For unencrypted packets, NAPTs ensure that there are no duplicate connections (duplicate sources).

Fig. 1 illustrates the transformation of an IPsec packet along a network path as it is sent from client to server. Fig. 2 illustrates the transformation of the return packet in the reverse direction, from server to client. With reference to Fig. 1, the client at IP address 10.1.1.1 sends an encrypted packet destined for the server at IP address 11.1.1.1. The original contents of the packet before processing by IPsec are shown at 100. The left column of 100 describes a field type of the packet, while the right column shows the field contents. Note that the destination IP address at 100 is 211.1.1.1, which is the public address of the NAT in front of the real destination server 11.1.1.1. It is the responsibility of NAT 211.1.1.1 to map packets to its backend servers such as 11.1.1.1. At 100, the source and destination ports are illustratively set to 4096 and 21, respectively. The contents of the packet after IPsec processing are shown at 102. The lightly shaded portion at the bottom of the packet 102 illustrates the portion encrypted by IPsec. The heavier shaded portions of 102 (and the packet contents at other points of the transmission path) illustrate fields that have changed or have been added at that point in the transmission. At 102, the real source and destination ports are encrypted values of 4096 and 21 by IPsec and are not readable at this point. IPsec processing has added a UDP header to indicate that this is an IPsec packet that encapsulates the ports and protocol of the original client packet. The source and destination ports in the clear text UDP header added by IPsec are set to 4500 as required by RFC 3948. An SPI (Security Parameter Index) field is illustratively set to 256. The SPI field, together with a security protocol (ESP or AH) and a destination address, points to a security association between client

10.1.1.1 and server 11.1.1.1 that governs the encryption algorithm and other security parameters between these entities.

The packet at 102 is translated by the NAPT at IP address 210.1.1.1 to result in the packet shown at 104. At this point the NAPT 210.1.1.1 has changed the source IP address to reflect its own address of 210.1.1.1. The NAPT also sets a new unique source port number. In Fig. 1, the selected source port number is illustratively changed from 4500 to 4501. The NAPT 210.1.1.1 keeps track of this translation for return packets from the server 11.1.1.1 and for future outbound packets from client IP 10.1.1.1 and source port 4500.

The packet at 104 is re-translated by NAT 211.1.1.1 into the input packet for server 11.1.1.1. This input packet is shown at 106. Essentially, the destination IP address of the packet is mapped by NAT 211.1.1.1 into the real destination address 11.1.1.1 of the destination server. The IPsec processing of the packet removes the UDP header added by the IPsec processing at the source 10.1.1.1 and restores the real source and destination port numbers. The restored packet, as shown at 108 is then delivered to the destination port (21 in this example) for application processing.

For completeness, Fig. 2 shows a return packet flow from server 211.1.1.1 to the original client 10.1.1.1. There is no need to discuss this packet flow in any detail because the duplicate source problem which is addressed cannot occur for return packets.

With reference again to Fig. 1, the packet at 108 contains as a source address the address of the NAPT 210.1.1.1 and source port address 4096. However, it is very possible that another client, say 10.1.1.2, behind NAPT 210.1.1.1 is also sending packets to host 11.1.1.1 from source port 4096. Therefore, because of the presence of an NAPT in the path between client 10.1.1.1 and host 11.1.1.1, there is a possibility of an illegal duplicate source that results in a conflict.

A Source Port Translation Table (SPTT, Fig. 3) at a destination host is used to define associations between sources (called Source Port Entries in Fig. 3), which involve a NAPT, and translated port numbers assigned to such connections by the destination host (called Translated Source Port Entries in Fig. 3). The translated source ports are selected from a pool of ports that are available for assignment (not already assigned) to incoming packets that are associated with a given security association.

Each negotiated security association will have its own pool of translation ports. For a given connection, each pool of translation ports must, of course, comply with the same or equivalent security rule in the SPD that the original connection requires, as defined in part by the original client port number. By assigning available translation ports to incoming packets, the number of "duplicate" packets that are rejected by the invention should be reduced and perhaps dramatically reduced.

An illustrative SPTT is shown in Fig. 3 at 300. This table is built dynamically as needed when incoming packets arrive at the destination host. The Source Port Entries of SPTT contain four fields; 1) the source IP address of the NAPT in the path of the connection (e.g., entry 302 contains NAPT IP address 210.1.1.1); 2) a UDP source port assigned by the NAPT (e.g., 4501 of entry 302); 3) the original source port number selected by the originating client served by the NAPT (e.g. 4096 in entry 302); 4) the protocol of the original client packet (e.g. TCP in entry 302). Each Source Port Entry of the SPTT points to a Translated Source Port Entry. For example, Source Port Entry 302 points to Translated Source Port Entry 308 in Fig. 3. Each Translated Source Port Entry contains three fields; 1) the source IP address of the NAPT in the path of the connection (e.g., entry 308 contains NAPT IP address 210.1.1.1); 2) the translated source port number of the originating client served by the NAPT (e.g. 4096 in entry 302); 3) the protocol selected by an original client packet (e.g. TCP in entry 308). The Source Port Entries are used to find the Translated Source Port Entries for packets inbound to the destination host, while the Translated Source Port Entries are used to find the Source Port Entries for packets outbound from the host.

An Available Source Port Pool (ASPP), shown in Fig. 13 (Fig. 13 is on the same sheet as Fig. 3) keeps track of port numbers that are available for translation assignment and those that have already been assigned. The information in ASPP is generated dynamically when a packet is first received at a server from a new remote client. ASPP 1300 contains Remote Client Entries that point to Port Vectors. Each Remote Client Entry contains two fields, an IP address of a NAPT, taken from an incoming packet, and the original client protocol, UDP or TCP, of the packet. The original client protocol is encrypted when the packet arrives, and is available in the clear after IPsec processing at the destination host. Each of these Remote Client Entries points to a different Port Vector, each bit of which describes the available or unavailable state of a port number defined by the position of the bit within the vector.

With reference to Fig. 13, when IKE negotiates an IPsec security association that traverses a NAPT, the TCP/IP stack creates a range of port numbers that are acceptable under the security association that is negotiated. Port numbers within this range can be arbitrarily assigned to incoming packets to avoid the possibility of a duplicate source. How this is done will become clear in the discussion of Figs. 8 through 12. Only the range of ports that can be assigned according to the security association for the corresponding Remote Client Entry are addressed in any Vector, as will be seen.

Figs. 4 through 7 help illustrate the above discussion. Fig. 4 shows a packet coming from a source NAPT. The client address and port are assumed to be 10.1.1.1 and 4096 for illustration. 400 is the IP header updated by the NAPT. It contains the NAPT address 210.1.1.1 and a host destination address 11.1.1.1. 402 is the encapsulating UDP header added by IPsec processing and updated by the NAPT. Source port 4500 has been changed to 4501 by the NAPT. 404 contains the Encapsulated Security Protocol (ESP) header added by IPsec processing. The TCP transport header 406 contains the original client source and destination ports, 4096 and 21. 408 contains the payload data followed by the ESP trailer. The transport header 406 and payload 408 are encrypted in accordance with IPsec processing. Fig. 5 represents the packet of Fig. 4 after decryption at the destination. Note now that the source NAPT address 210.1.1.1 (from 500), and the client source port 4096 and protocol (TCP) are now available from 506. For inbound packets, the Source Port Entries (e.g., 302, 304) of SPTT 300 are searched using these attributes to locate a corresponding Translated Source Port Entry, if it exists. For outbound packets, the Translated Source Port Entries (e.g., 308, 310) are searched to locate a corresponding Source Port Entry.

Figs. 6 and 7 represent a second arriving "duplicate" source packet. This packet will be accepted by an embodiment of the present invention, assuming that there is an available port number for assignment to the packet from the range of allowable port numbers according to the security association that governs the packet.

This process is now explained in more detail below in association with appropriate flowcharts.

Fig. 8 illustrates the initializing of a security association during IKE negotiations. The IKE negotiation is represented at step 802. During the negotiation, step 804 sends a notification to the TCP/IP stack to

install the negotiated security association. Once the security association is known, the SPD is searched to determine the range of port numbers, if there is a range, that is available for assignment to connections. Step 806 determines these port numbers and adds them to the security association stored in the stack.

Fig. 9 begins the process of avoiding a duplicate source when a data packet arrives at the destination host. Fig. 9 begins the process of detecting a duplicate source when a data packet arrives at the destination host. Step 902 determines if the incoming packet contains an ESP packet encapsulated in a UDP header, and the source port in the UDP header is not the predefined UDP encapsulation port 4500. If the above is true, then the packet is using IPsec, either for encryption or authentication, and a NAPT is involved in the transmission path. If a packet is using a UDP protocol with a destination port of 4500 and the first four bytes contain non-zero data, then the packet is identified as a UDP encapsulated ESP packet.. If the answer to these questions is no, then there are two alternative processing options, option 1 at 904 and option 2 at 906. These are both discussed below. Assuming that the answer to both questions is yes, then step 908 continues at A in Fig. 10. In Fig. 10, step 1001 removes the UDP header from the incoming packet. Step 1002 performs the required IPsec processing to decrypt the packet. As a result, the original client source port number, such as 4096 at 302 and the original client protocol, such as TCP at 302, are obtained. The NAPT source IP address and NAPT assigned port number, such as 210.1.1.1 and 4501, respectively at 302, are known from the UDP header. Step 1004 searches the Source Port Entries of SPTT 300 on these attributes. If a matching Source Port Entry is found at step 1006, this means that a translated port number has already been assigned to this session. Step 1008 locates the port number that has been assigned. In this example, the matching Source Port Entry is 302. The corresponding Translated Source Port Entry is 308. Entry 308 contains the assigned source port number 4096, which happens to be identical to the original client source port number. This means is that when entry 308 was created, the original client source port number was available for assignment and was therefore used as the assigned port. This will be covered further below. Step 1010 replaces the source port number in the packet transport header with the translated source port from 308 and normal packet processing continues at 1012. Had the matching Translated Source Port Entry been 310 at step 1008, then the translated port number would have been 38096 according to the illustrative contents of SPTT 300.

Continuing with this example, if a Source Port Entry in SPTT 300 is not found at step 1006, then a process begins to create such an entry and the entries of the ASPP 1300 as needed. Initially, step 1016 determines if an ASPP Remote Client Entry already exists. If not, step 1020 creates the entry using the source NAPT IP address 210.1.1.1 in this example and the Protocol TCP from the decrypted packet. Step 1022 creates the corresponding Bit Vector. Initially, all bits of the Vector are set to the available state. At E of Fig. 11, step 1102 next determines if the client source port number in the packet is marked as available in the Source Port Bit Vector. If it is, the source port number in the packet is assigned at 1104. This corresponds to the example at 302 and 308, where the original source port number and the assigned port number are identical. If the original client source port number is marked unavailable in the Vector, step 1110 determines the range of ports that are allowable for assignment according to the security association. Within this range in the Port Bit Vector, step 1112 determines the next port number that is available, if any. If no port numbers are available for assignment, the packet is rejected at 1114, because it represents a duplicate source that cannot be resolved. If a port is available within the allowable range in the Vector, step 1113 creates a translated source port entry, such as 308 and 310) in SPTT 300 using a selected available port from the Vector; step 1106 marks the selected port as unavailable; step 1108 creates a Source Port Entry, such as 304, in SPTT 300 and associates it with the translated source port entry created at step 1113. Processing continues at F in Fig. 10 where the packet port number is replaced with the assigned port number at step 1010 and normal packet processing continues at step 1012.

One variation to the above at step 1110 is required if option 1 from Fig. 9 is used. In the event that step 910 determines that an incoming packet is not an IPsec packet, then there is no SA associated with this packet. Therefore, step 1110 must obtain the port bit vector containing the range of ports available for assignment directly from the SPD rather than from a SA as in the preferred embodiment.

Options 1 and 2 from Fig. 9 represent situations in which packets are sent in the clear (no IPsec processing) or there is no address translation (NAPT) in the path. As long as some security association ends on this host that also traverse a NAPT, duplicate sources are still possible in these situations. Both alternative options 1 and 2 avoid such duplicate packets in these conditions. Option 1 applies the same principles as discussed above for duplicate source avoidance, but by applying or

avoiding IPsec processing as required by the packet. Option 2 uses the filtering rules of the SPD to avoid duplicate sources. If option 1 (904) is selected, step 910 determines if the packet requires IPsec processing. If yes, processing continues at 912, which continues to B in Fig. 10; step 1002 at B in Fig. 10 performs IPsec processing as required and processing continues on at 1004. If the packet is not an IPsec packet at step 910, step 914 continues to D in Fig. 10, which merely skips the IPsec processing step at B. In this situation, step 1004 in Fig. 10 uses a UDP source port of zero (0), since there is no encapsulating UDP header in the packet.

Option 2 uses inbound IPsec packet filtering to avoid duplicate sources, if possible. Once IPsec is in place, all packets are processed through the IPsec rules table SPD, whether any packet is encrypted or not. This is to verify that clear packets on a given connection are in fact allowed by the IPsec rules. The option 2 process begins at C of Fig. 12. The incoming packet is processed through the IPsec rule table (not shown) at step 1202. An example of how this is done in a preferred embodiment can be determined from the aforementioned U.S. patent 6,347,376. This patent is incorporated by reference in its entirety. If the packet is encrypted (step 1204), then step 1206 determines if the matching IPsec rule requires encryption. Assuming that is the case, the packet is allowed at 1206. Otherwise, it is rejected at 1210. If the packet is in the clear at step 1204, then a determination is made at 1212 if the matching IPsec rule allows unencrypted packets and the packet is allowed or rejected accordingly.

In tunnel mode, the IPsec SA is not necessarily end-to-end. For example, an SA might be negotiated between a host and a gateway that serves multiple clients or servers. In tunnel mode a single NAPT address (which is the source IP address in the UDP encapsulating header) could potentially represent multiple hosts. In tunnel mode, the encapsulated, encrypted portion of a packet contains both the original IP address of the source and the TCP transport header. For the purpose of this specification, the original IP address of the source in tunnel mode is called the inner source IP address. Because the inner source IP address is not globally unique, it is not usable for packet routing or for representing the source of a connection. The original source port, such as contained in the source port entries of SPTT 300, and the encapsulating source IP address with the UDP port alone, as described above for transport mode, might not be unique. To address this, an additional field that contains the inner source IP address is added to the source port

entries (e.g., 302 and 304) of the SPTT 300 in Fig. 3. The inner source IP address (not available in transport mode) when combined with the other values of the source port entries yield a unique identifier for hosts protected by a tunnel mode IPsec SA.

Those skilled in the art will realize that the preferred and disclosed embodiment can have many minor variations that are within the intent and scope of the teaching. It is the intent of the inventor to encompass these variations to the extent possible in accordance with the state of the applicable relevant art in the field of the invention.

CLAIMS

1. A method of preventing duplicate source conflicts in a network protocol that uses network addresses, protocols and port numbers to identify connections, comprising

in response to an inbound packet on a connection at a destination host from a remote source client, determining if a port number is available within a range of port numbers that comply with a security association governing the connection,

assigning an available port number to the connection, thereby avoiding a possibility of a duplicate source, and

rejecting the packet if no port numbers are available within the range of port numbers governing the connection.

2. The method of claim 1 further comprising

maintaining a list of assignable port numbers for each remote source client and the assigned and unassigned states of each port number in the list.

3. The method of claim 2 wherein each list of assignable port numbers is a bit vector wherein bit positions identify port numbers and the bit states describe the assigned and unassigned states of the port numbers.

4. The method of claim 1 further comprising

maintaining a Source Port Translation Table that associates each client side connection with a translated port number.

5. The method of claim 4 wherein the Source Port Translation Table contains Source Port Entries and Translated Source Port Entries, wherein each Source Port Entry is uniquely associated with a Translated Source port Entry, and each Source Port Entry contains an internet source address, a UDP source port number assigned by an NAPT, a client source port number, and a client protocol identifier, and each Translated Source Port Entry contains an internet source address, a translated client source port number, and a client protocol identifier, wherein the Source Port Entries are searched on incoming packets to identify a port number previously assigned to a remote client connection, and the Translated Source Port Entries are searched on outgoing packets to identify a client port number from a previously assigned translated port number.

6. A method of preventing duplicate sources in a network protocol that uses network addresses, protocols and port numbers to identify connections, comprising

a) receiving a packet at a server,

b) determining if the packet has been translated by a network address port translator and contains an encapsulated encrypted packet,

c) if the packet has been translated and contains an encapsulated encrypted packet, decrypting the encapsulated packet to obtain original connection information,

d) determining if a port number is available within a range of port numbers that comply with a security association governing the connection,

e) assigning an available port number to the connection, thereby avoiding a possibility of a duplicate source, and

f) rejecting the packet if no port numbers are available within the range of port numbers governing the connection.

7. The method of claim 6 further comprising

maintaining a list of assignable port numbers for each connection and the assigned and unassigned states of each port number in the list.

8. The method of claim 7 wherein each list of assignable port numbers is a bit vector wherein bit positions identify port numbers and the bit states describe the assigned and unassigned states of the port numbers.

9. Apparatus for preventing duplicate source conflicts in a network protocol that uses network addresses, protocols and port numbers to identify connections, comprising

means responsive to an inbound packet on a connection at a destination host from a remote source client for determining if a port number is available within a range of port numbers that comply with a security association governing the connection,

means for assigning an available port number to the connection, thereby avoiding a possibility of a duplicate source, and

means for rejecting the packet if no port numbers are available within the range of port numbers governing the connection.

10. The apparatus of claim 9 further comprising

means for maintaining a list of assignable port numbers for each remote source client and the assigned and unassigned states of each port number in the list.

11. The apparatus of claim 10 wherein each list of assignable port numbers is a bit vector wherein bit positions identify port numbers and the bit states describe the assigned and unassigned states of the port numbers.

12. The apparatus of claim 9 further comprising

means for maintaining a Source Port Translation Table that associates each client side connection with a translated port number.

13. The apparatus of claim 12 wherein the Source Port Translation Table contains Source Port Entries and Translated Source Port Entries, wherein each Source Port Entry is uniquely associated with a Translated Source port Entry, and each Source Port Entry contains an internet source address, a UDP source port number assigned by an NAPT, a client source port number, and a client protocol identifier, and each Translated Source Port Entry contains an internet source address, a translated client source port number, and a client protocol identifier, wherein the apparatus further comprises means for searching Source Port Entries on incoming packets to identify a port number previously assigned to a remote client connection, and means for searching the Translated Source Port Entries on outgoing packets to identify a client port number from a previously assigned translated port number.

14. Apparatus for preventing duplicate sources in a network protocol that uses network addresses, protocols and port numbers to identify connections, comprising

a) means for receiving a packet at a server,

b) means for determining if the packet has been translated by a network address port translator and contains an encapsulated encrypted packet,

c) means for decrypting an encapsulated packet to obtain original connection information,

d) means for determining if a port number is available within a range of port numbers that comply with a security association governing the connection,

e) means for assigning an available port number to the connection, thereby avoiding a possibility of a duplicate source, and

f) means for rejecting the packet if no port numbers are available within the range of port numbers governing the connection.

15. The apparatus of claim 14 further comprising

means for maintaining a list of assignable port numbers for each connection and the assigned and unassigned states of each port number in the list.

16. The apparatus of claim 15 wherein each list of assignable port numbers is a bit vector wherein bit positions identify port numbers and the bit states describe the assigned and unassigned states of the port numbers.

17. A storage medium for storing program instructions, which when loaded into a computer causes the computer to perform a method of preventing duplicate source conflicts in a network protocol that uses network addresses, protocols and port numbers to identify connections, the instructions comprising

instructions responsive to an inbound packet on a connection at a destination host from a remote source client for determining if a port number is available within a range of port numbers that comply with a security association governing the connection,

instructions for assigning an available port number to the connection, thereby avoiding a possibility of a duplicate source, and

instructions for rejecting the packet if no port numbers are available within the range of port numbers governing the connection.

18. The medium of claim 17 wherein the instructions further comprise

instructions for maintaining a list of assignable port numbers for each remote source client and the assigned and unassigned states of each port number in the list.

19. The medium of claim 18 wherein each list of assignable port numbers generated by the instructions is a bit vector wherein bit positions identify port numbers and the bit states describe the assigned and unassigned states of the port numbers.

20. The medium of claim 17 further comprising

instructions for maintaining a Source Port Translation Table that associates each client side connection with a translated port number.

21. The medium of claim 20 wherein the Source Port Translation Table generated by the instructions contains Source Port Entries and Translated Source Port Entries, wherein each Source Port Entry is uniquely associated with a Translated Source port Entry, and each Source Port Entry contains an internet source address, a UDP source port number

assigned by an NAPT, a client source port number, and a client protocol identifier, and each Translated Source Port Entry contains an internet source address, a translated client source port number, and a client protocol identifier, wherein the Source Port Entries are searched on incoming packets to identify a port number previously assigned to a remote client connection, and the Translated Source Port Entries are searched on outgoing packets to identify a client port number from a previously assigned translated port number.

22. A storage medium for storing program instructions, which when loaded into a computer causes the computer to perform a method of preventing duplicate sources in a network protocol that uses network addresses, protocols and port numbers to identify connections, comprising

a) instructions for receiving a packet at a server,

b) instructions for determining if the packet has been translated by a network address port translator and contains an encapsulated encrypted packet,

c) instructions for decrypting an encapsulated packet to obtain original connection information,

d) instructions for determining if a port number is available within a range of port numbers that comply with a security association governing the connection,

e) instructions for assigning an available port number to the connection, thereby avoiding a possibility of a duplicate source, and

f) instructions for rejecting the packet if no port numbers are available within the range of port numbers governing the connection.

23. The storage medium of claim 22 further comprising

instructions for maintaining a list of assignable port numbers for each connection and the assigned and unassigned states of each port number in the list.

24. The medium of claim 23 wherein each list of assignable port numbers generated by the instructions is a bit vector wherein bit positions identify port numbers and the bit states describe the assigned and unassigned states of the port numbers.

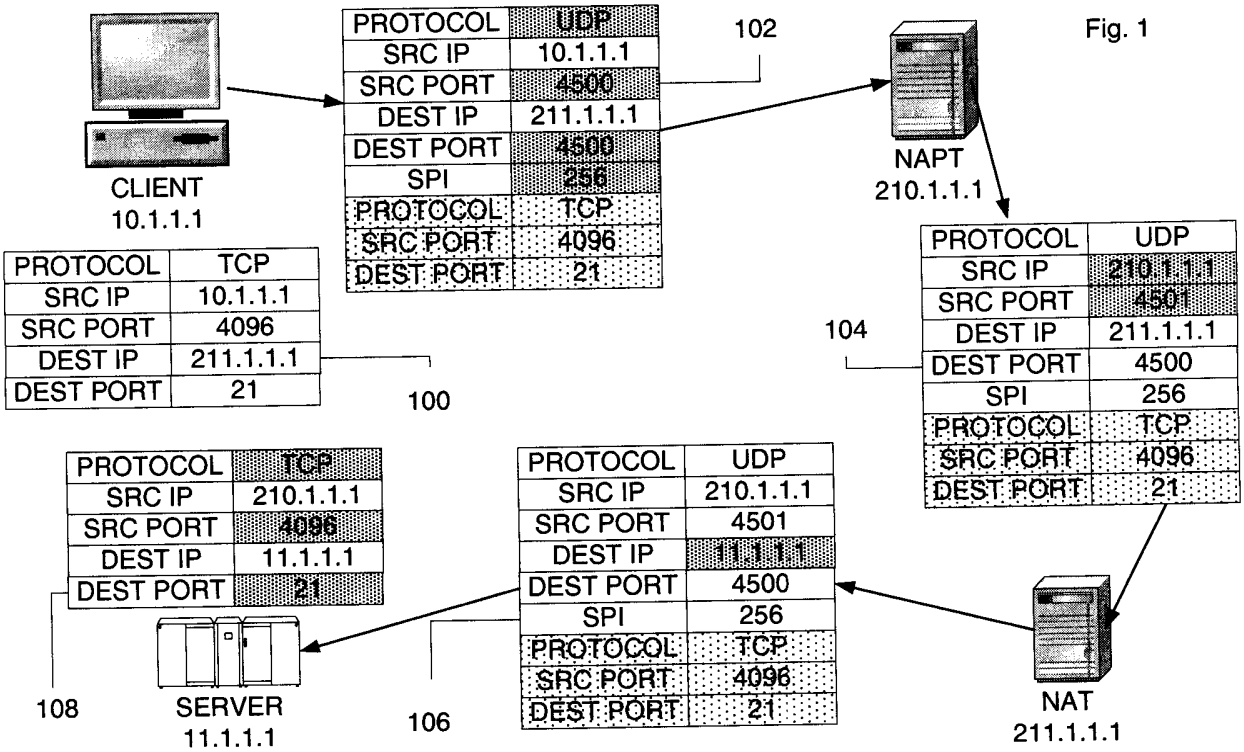

CLIENT
10.1.1.1
PROTOCOL UDP
SRC IP 10.1.1.1
SRC PORT 4500
DEST IP 211.1.1.1
DEST PORT 4500
SPI 256
PROTOCOL TCP
SRC PORT 4096
DEST PORT 21
102
NAPT
210.1.1.1
PROTOCOL TCP
SRC IP 10.1.1.1
SRC PORT 4096
DEST IP 211.1.1.1
DEST PORT 21
100
PROTOCOL UDP
SRC IP 210.1.1.1
SRC PORT 4501
DEST IP 211.1.1.1
DEST PORT 4500
SPI 256
PROTOCOL TCP
SRC PORT 4096
DEST PORT 21
104
PROTOCOL TCP
SRC IP 210.1.1.1
SRC PORT 4096
DEST IP 11.1.1.1
DEST PORT 21
108
SERVER
11.1.1.1
PROTOCOL UDP
SRC IP 210.1.1.1
SRC PORT 4501
DEST IP 11.1.1.1
DEST PORT 4500
SPI 256
PROTOCOL TCP
SRC PORT 4096
DEST PORT 21
106
NAT
211.1.1.1

Fig. 1

Fig. 2

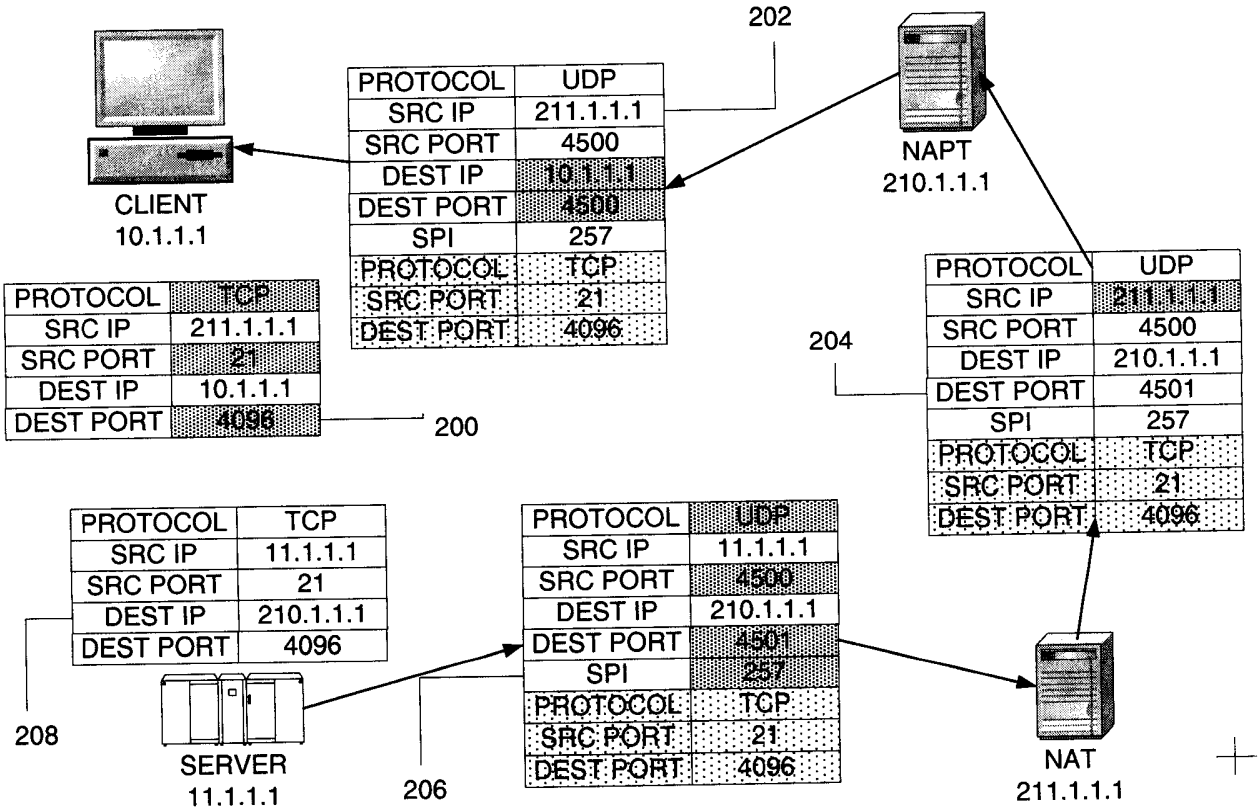

CLIENT
10.1.1.1
PROTOCOL UDP
SRC IP 211.1.1.1
SRC PORT 4500
DEST IP 10.1.1.1
DEST PORT 4500
SPI 257
PROTOCOL TCP
SRC PORT 21
DEST PORT 4096
202
NAPT
210.1.1.1
PROTOCOL TCP
SRC IP 211.1.1.1
SRC PORT 21
DEST IP 10.1.1.1
DEST PORT 4096
200
PROTOCOL UDP
SRC IP 210.1.1.1
SRC PORT 4500
DEST IP 210.1.1.1
DEST PORT 4501
SPI 257
PROTOCOL TCP
SRC PORT 21
DEST PORT 4096
204
PROTOCOL TCP
SRC IP 11.1.1.1
SRC PORT 21
DEST IP 210.1.1.1
DEST PORT 4096
208
SERVER
11.1.1.1
PROTOCOL UDP
SRC IP 11.1.1.1
SRC PORT 4500
DEST IP 210.1.1.1
DEST PORT 4501
SPI 257
PROTOCOL TCP
SRC PORT 21
DEST PORT 4096
206
NAT
211.1.1.1

FIG. 3

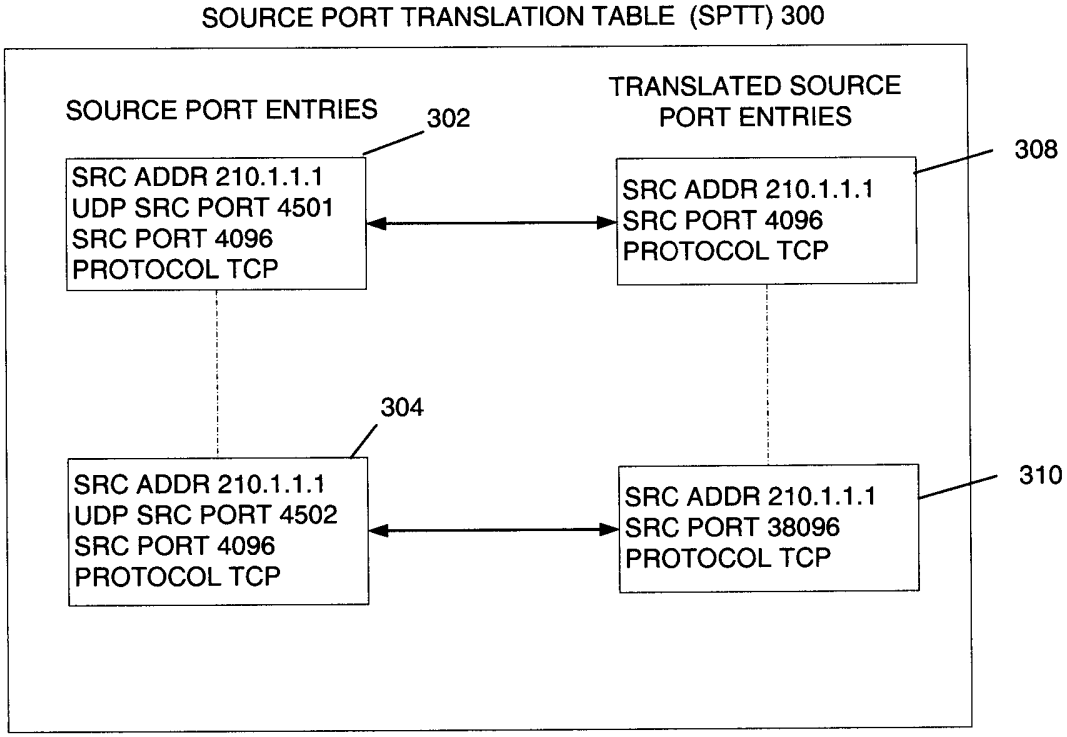

SOURCE PORT TRANSLATION TABLE (SPTT) 300
SOURCE PORT ENTRIES
302
TRANSLATED SOURCE PORT ENTRIES
308
SRC ADDR 210.1.1.1
UDP SRC PORT 4501
SRC PORT 4096
PROTOCOL TCP
SRC ADDR 210.1.1.1
SRC PORT 4096
PROTOCOL TCP
304
310
SRC ADDR 210.1.1.1
UDP SRC PORT 4502
SRC PORT 4096
PROTOCOL TCP
SRC ADDR 210.1.1.1
SRC PORT 38096
PROTOCOL TCP

FIG. 13

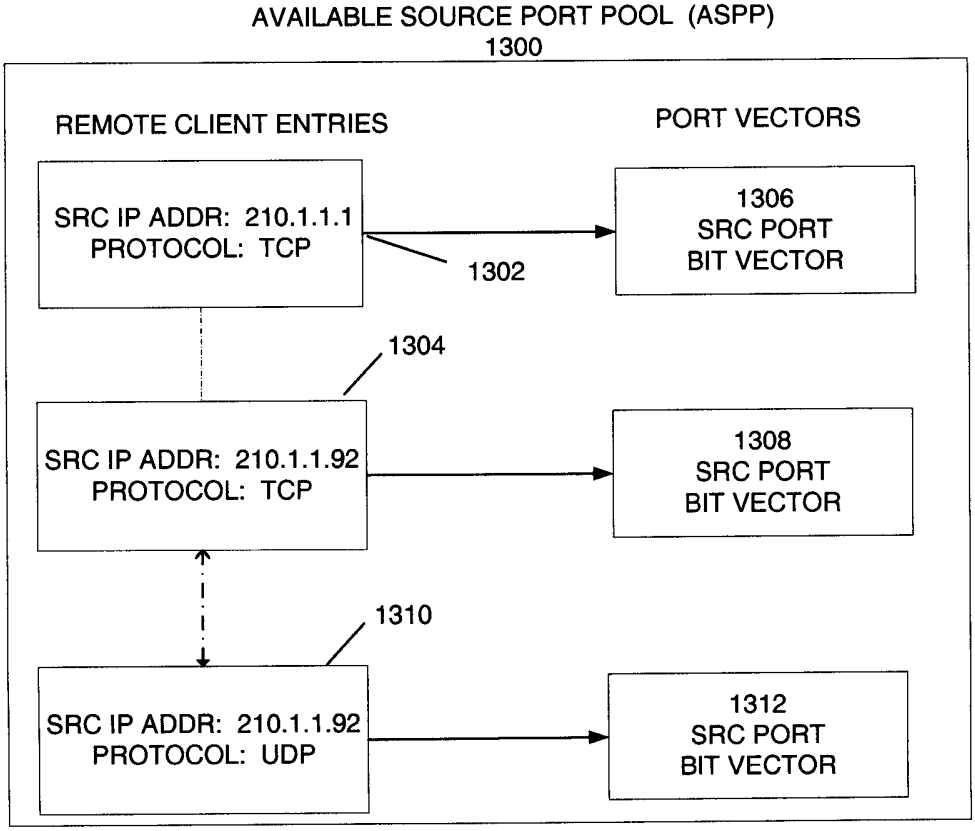

AVAILABLE SOURCE PORT POOL (ASPP)
1300
REMOTE CLIENT ENTRIES
PORT VECTORS
SRC IP ADDR: 210.1.1.1
PROTOCOL: TCP
1306
SRC PORT
BIT VECTOR
1302
1304
SRC IP ADDR: 210.1.1.92
PROTOCOL: TCP
1308
SRC PORT
BIT VECTOR
1310
SRC IP ADDR: 210.1.1.92
PROTOCOL: UDP
1312
SRC PORT
BIT VECTOR

FIG. 4 - NAPT ENCAPSULATED PACKET FROM CLIENT ADDR 10.1.1.1 PORT 4096

400 IP HEADER SRC 210.1.1.1 DEST 11.1.1.1	402 UDP HEADER SRC PORT 4501 DEST PORT 4500	404 ESP HEADER	406 PROTOCOL TCP SRC PORT 4096 DEST PORT 21	408 PAYLOAD	410 ESP TRAILER

FIG. 5 - PACKET FROM FIG. 4 AFTER IPSEC PROCESSING AT HOST

500 IP HEADER SRC 210.1.1.1 DEST 11.1.1.1	506 PROTOCOL TCP SRC PORT 4096 DEST PORT 21	508 PAYLOAD

FIG. 6 - NAPT ENCAPSULATED PACKET FROM CLIENT ADDR 10.1.1.2 PORT 4096

600 IP HEADER SRC 210.1.1.1 DEST 11.1.1.1	602 UDP HEADER SRC PORT 4502 DEST PORT 4500	604 ESP HEADER	606 PROTOCOL TCP SRC PORT4096 DEST PORT 21	608 PAYLOAD	610 ESP TRAILER

FIG. 7 - PACKET FROM FIG. 5 AFTER IPSEC PROCESSING AT HOST

700 IP HEADER SRC 210.1.1.1 DEST 11.1.1.1	706 PROTOCOL TCP SRC PORT 4096 DEST PORT 21	708 PAYLOAD

FIG. 8 - IKE NEGOTIATION

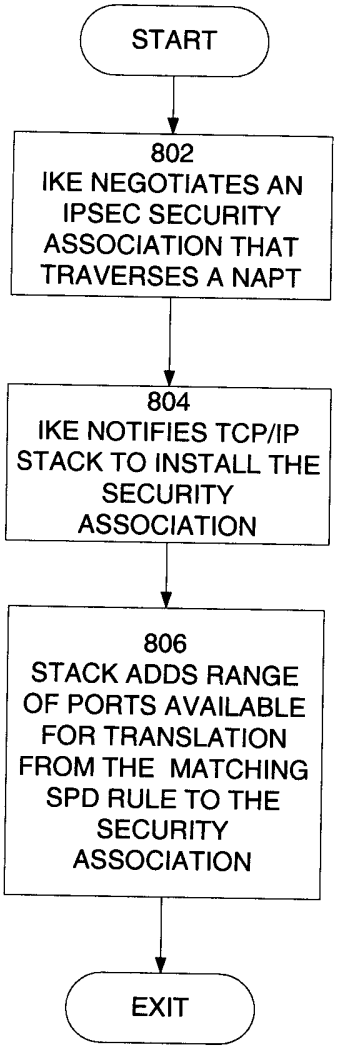


FIG. 9 - DATA PACKET ARRIVES

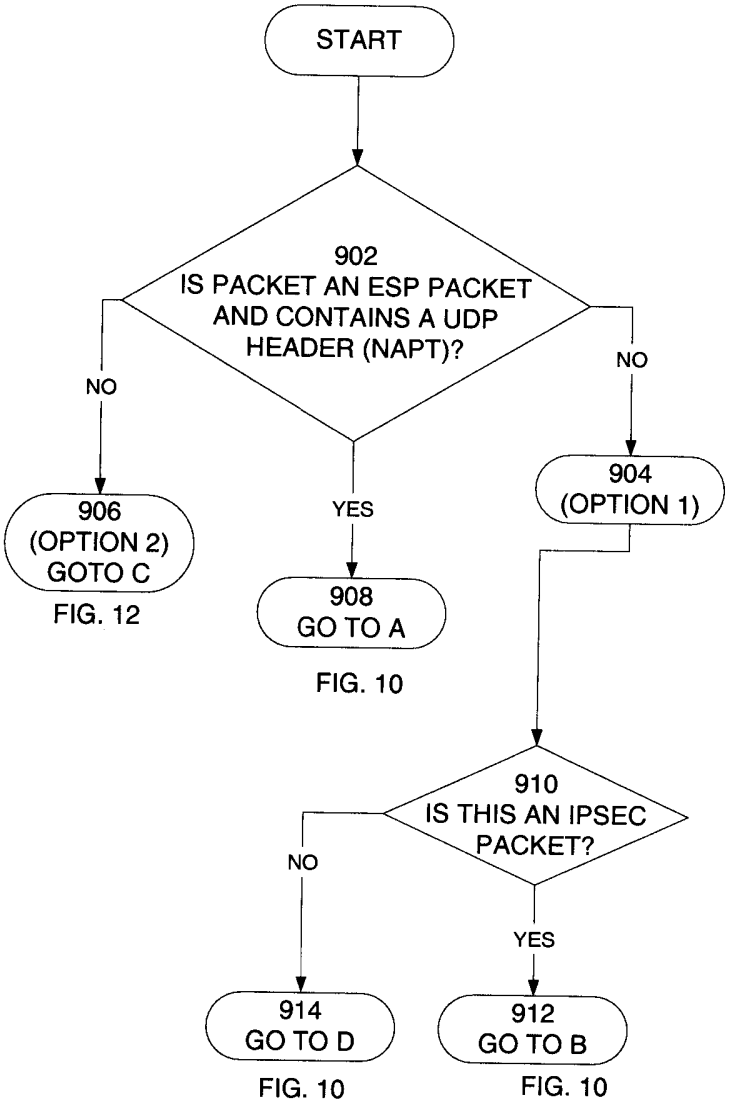


FIG. 12

FIG. 10

FIG. 10

FIG. 10

FIG. 10

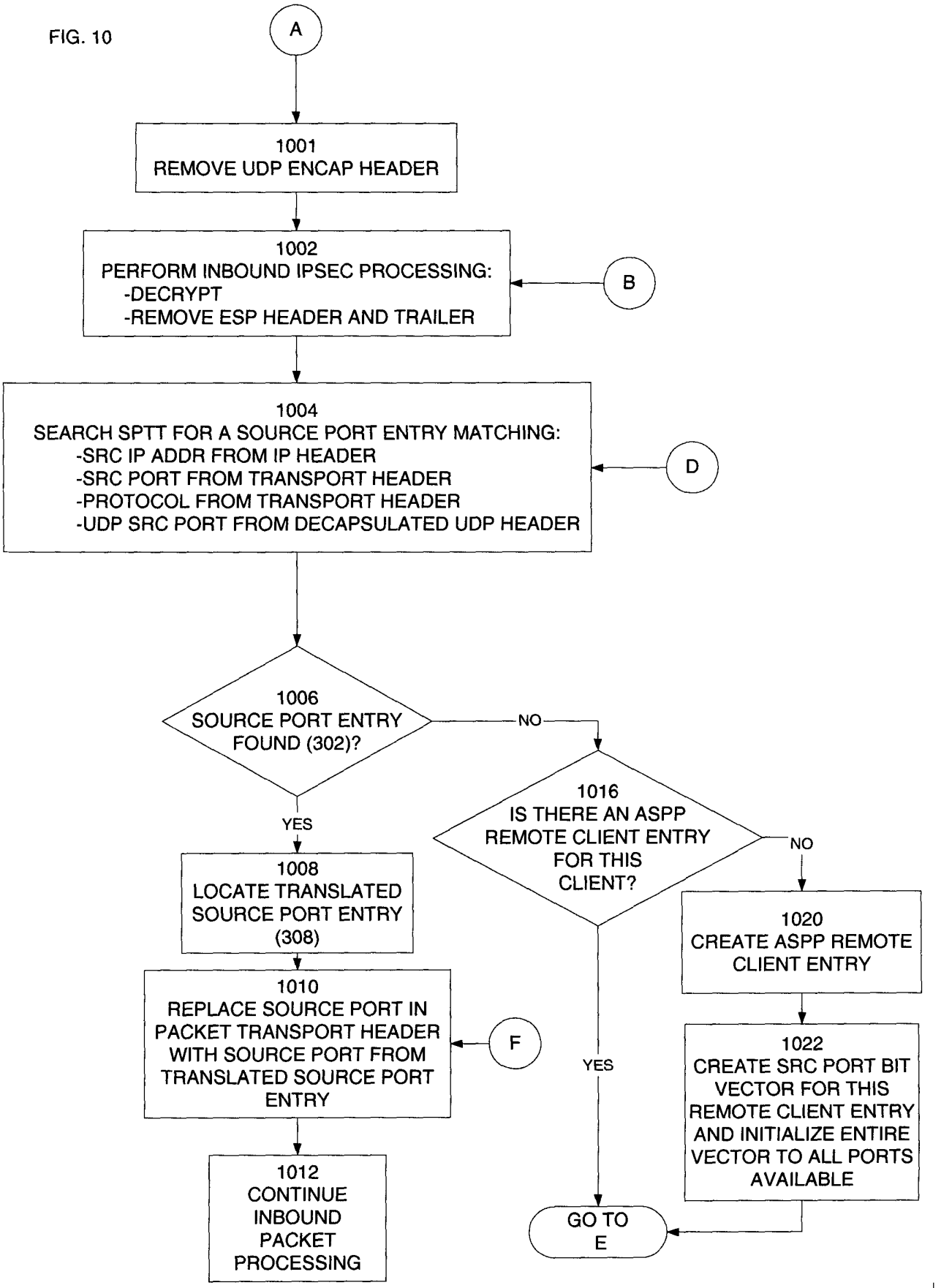

A
1001
REMOVE UDP ENCAP HEADER
1002
PERFORM INBOUND IPSEC PROCESSING:
-DECRYPT
-REMOVE ESP HEADER AND TRAILER
B
1004
SEARCH SPTT FOR A SOURCE PORT ENTRY MATCHING:
-SRC IP ADDR FROM IP HEADER
-SRC PORT FROM TRANSPORT HEADER
-PROTOCOL FROM TRANSPORT HEADER
-UDP SRC PORT FROM DECAPSULATED UDP HEADER
D
1006
SOURCE PORT ENTRY FOUND (302)?
NO
YES
1016
IS THERE AN ASPP REMOTE CLIENT ENTRY FOR THIS CLIENT?
NO
YES
1008
LOCATE TRANSLATED SOURCE PORT ENTRY (308)
1020
CREATE ASPP REMOTE CLIENT ENTRY
1010
REPLACE SOURCE PORT IN PACKET TRANSPORT HEADER WITH SOURCE PORT FROM TRANSLATED SOURCE PORT ENTRY
F
1022
CREATE SRC PORT BIT VECTOR FOR THIS REMOTE CLIENT ENTRY AND INITIALIZE ENTIRE VECTOR TO ALL PORTS AVAILABLE
1012
CONTINUE INBOUND PACKET PROCESSING
GO TO E

FIG. 11

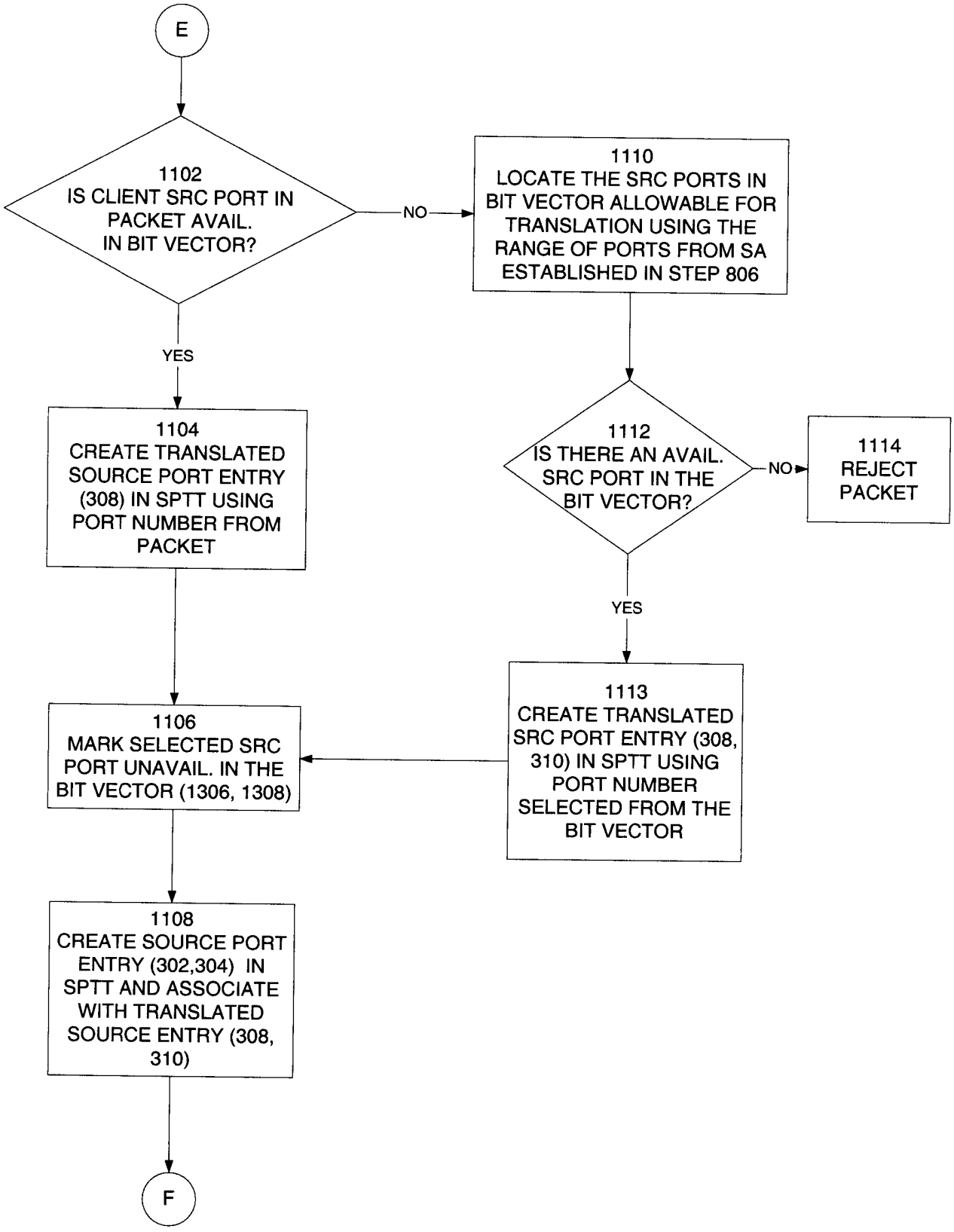
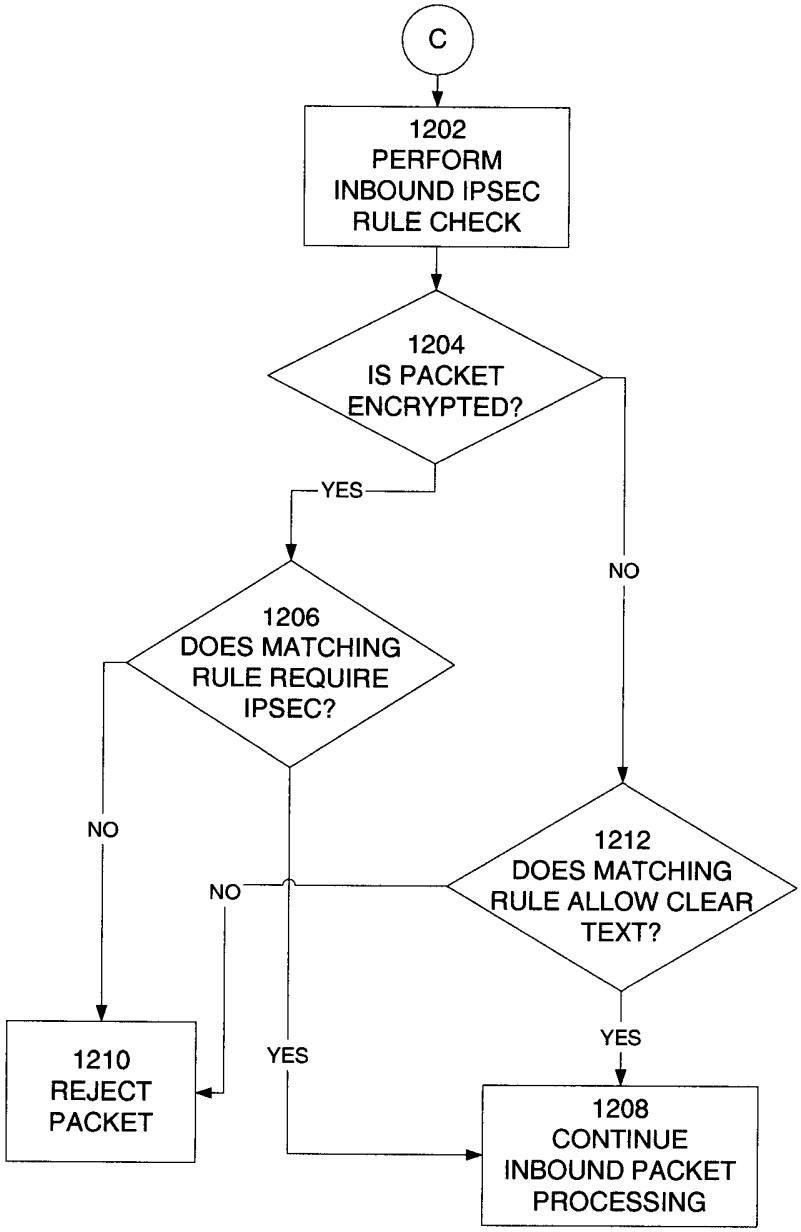

E
1102
IS CLIENT SRC PORT IN PACKET AVAIL. IN BIT VECTOR?
NO
1110
LOCATE THE SRC PORTS IN BIT VECTOR ALLOWABLE FOR TRANSLATION USING THE RANGE OF PORTS FROM SA ESTABLISHED IN STEP 806
YES
1104
CREATE TRANSLATED SOURCE PORT ENTRY (308) IN SPTT USING PORT NUMBER FROM PACKET
1112
IS THERE AN AVAIL. SRC PORT IN THE BIT VECTOR?
NO
1114
REJECT PACKET
YES
1113
CREATE TRANSLATED SRC PORT ENTRY (308, 310) IN SPTT USING PORT NUMBER SELECTED FROM THE BIT VECTOR
1106
MARK SELECTED SRC PORT UNAVAIL. IN THE BIT VECTOR (1306, 1308)
1108
CREATE SOURCE PORT ENTRY (302,304) IN SPTT AND ASSOCIATE WITH TRANSLATED SOURCE ENTRY (308, 310)
F

FIG. 10

FIG. 12


C
1202
PERFORM
INBOUND IPSEC
RULE CHECK
1204
IS PACKET
ENCRYPTED?
YES
NO
1206
DOES MATCHING
RULE REQUIRE
IPSEC?
NO
NO
1212
DOES MATCHING
RULE ALLOW CLEAR
TEXT?
YES
YES
1210
REJECT
PACKET
1208
CONTINUE
INBOUND PACKET
PROCESSING

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2006/061443

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/12 H04L29/06

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)
H04L

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, COMPENDEX, INSPEC, IBM-TDB, 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	HUTTUNEN F-SECURE CORPORATION B SWANDER MICROSOFT V VOLPE CISCO SYSTEMS L DIBURRO NORTEL NETWORKS M STENBERG A: "UDP Encapsulation of IPsec ESP Packets; rfc3948.txt;" IETF STANDARD, INTERNET ENGINEERING TASK FORCE, IETF, CH, January 2005 (2005-01), XP015009720 ISSN: 0000-0003 cited in the application page 9, line 1 - page 10, line 30 page 12, line 1 - page 13, last line -/--	1-24

[X] 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.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
14 July 2006	25/07/2006

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Lebas, Y

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2006/061443

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	STENBERG S PAAVOLAINEN T YLONEN T KIVINEN SSH COMMUNICATIONS SECURITY CORP M: "IPsec NAT-Traversal; draft-stenberg-ipsec-nat-traversal-02.txt; " IETF STANDARD-WORKING-DRAFT, INTERNET ENGINEERING TASK FORCE, IETF, CH, no. 2, 28 February 2001 (2001-02-28), XP015035694 ISSN: 0000-0004 page 12, line 1 - page 12, line 12	1-24
A	ZACCONE C ET AL: "Address reuse in the internet, adjourning or suspending the adoption of IP next gqneration ?" NETWORKS, 2000. (ICON 2000). PROCEEDINGS. IEEE INTERNATIONAL CONFERENCE ON SEPTEMBER 5-8, 2000, PISCATAWAY, NJ, USA,IEEE, 5 September 2000 (2000-09-05), pages 462-468, XP010514142 ISBN: 0-7695-0777-8 page 464, left-hand column, line 30 - right-hand column, line 23	1-24
A	MONTENEGRO SUN MICROSYSTEMS G ET AL: "RSIP Support for End-to-end IPsec; rfc3104.txt" IETF STANDARD, INTERNET ENGINEERING TASK FORCE, IETF, CH, October 2001 (2001-10), XP015008885 ISSN: 0000-0003 page 13, line 1 - last line	1-24