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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
02 January 2020 (02.01.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/005540 A1

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

(21) International Application Number:
PCT/US2019/036644

(22) International Filing Date:
12 June 2019 (12.06.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
16/022,657 28 June 2018 (28.06.2018) US

(71) Applicant: **VMWARE, INC.** [US/US]; 3401 Hillview Avenue, Palo Alto, CA 94304 (US).

(72) Inventors: **JAIN, Rahul**; 3401 Hillview Avenue, Palo Alto, CA 94304 (US). **HIRA, Mukesh**; 3401 Hillview Avenue, Palo Alto, CA 94304 (US). **WANG, Su**; 3401 Hillview Avenue, Palo Alto, CA 94304 (US).

(74) Agent: **ADELI, Mani** et al.; Adeli LLP, 11859 Wilshire Blvd., Suite 408, Los Angeles, CA 90025 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH,

(54) Title: MANAGED FORWARDING ELEMENT DETECTING INVALID PACKET ADDRESSES

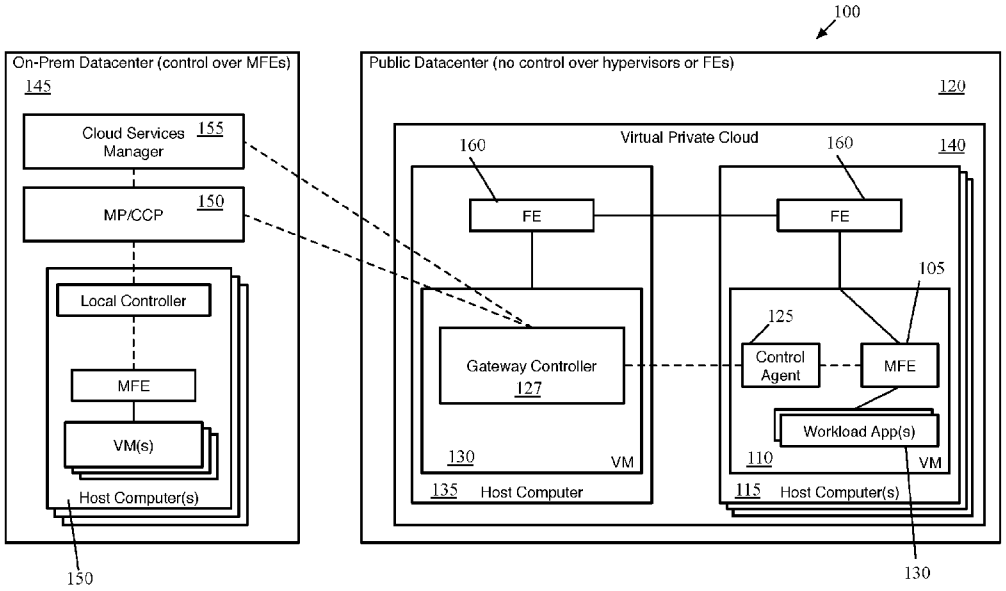

Figure 1

(57) Abstract: Some embodiments provide a method for a managed forwarding element (MFE) executing on a data compute node (DCN) that operates on a host computer in a public datacenter. The MFE implements a logical network that connects multiple DCNs within the public datacenter. The method receives a packet, directed to the DCN, that (i) has a first logical network source address and (ii) is encapsulated with a second source address associated with an underlying public datacenter network. The method determines whether the first logical network source address is a valid source address for the packet based on a mapping table that maps logical network addresses to underlying public datacenter network addresses. When the first source address is not a valid source address for the packet, the method drops the packet.

WO 2020/005540 A1

[Continued on next page]

GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

MANAGED FORWARDING ELEMENT DETECTING INVALID PACKET ADDRESSES

Rahul Jain, Mukesh Hira, Su Wang

BACKGROUND

[0001] Public datacenters provide companies with the ability to expand or move their networks out of their own private datacenters, thereby reducing the expense, administration, and other burdens required with operation and maintenance of physical servers. However, public datacenters do not allow their tenants to directly control the virtualization software, and they may not have as robust or transparent security capabilities. These limitations make tenant machines in a public datacenter vulnerable to attacks such as IP spoofing, which can evade security measures implemented by the tenant.

BRIEF SUMMARY

[0002] Some embodiments of the invention provide a novel spoofed-packet detection method for a managed forwarding element (MFE) executing on a data compute node (DCN) such as a virtual machine (VM) that operates on a host computer in a public multi-tenant datacenter. The MFE implements a logical network that connects multiple other DCNs (e.g., operating on different host computers) within the public datacenter. The host computers implement an underlying public datacenter network, also referred to as an underlay network.

[0003] The method of some embodiments initially receives a data message (such as a packet) that is directed to the DCN. The data message in some embodiments includes both (i) a logical network source address (also referred to as a logical source address) that is stored in a first header of the data message and (ii) an underlay network source address (also referred to as an underlay source address) that is stored in an encapsulating second header of the data message. The MFE determines whether the logical source address is valid by using a mapping table that maps logical source addresses to underlay source addresses. If the data message's logical source address does not match the expected logical source address listed in the mapping table for the data message's underlay source address, the MFE drops the packet.

[0004] In some embodiments, the MFE determines whether the logical source address is valid by first identifying an entry of the mapping table that corresponds to the logical source address. The identified mapping table entry specifies a valid underlay source address associated with the logical source address and the MFE determines whether the underlay source address of the data message matches the valid source address specified in the mapping table entry. If the addresses match, then the logical source address of the data message is valid, and the data message is forwarded to the logical destination address.

[0005] Within the logical network, any DCN may send data messages to any other DCN. In some embodiments, these data messages are originated from and directed to a workload application executing on the DCN with the MFE. When the MFE determines (using the mapping table) that the logical source address of a received data message is valid, the MFE forwards the data message to the workload application executing on the same DCN.

[0006] The logical source address of the data message is an address that corresponds to a workload application executing on the DCN from which the data message originated. If the MFE determines that the logical source address of the data message is valid (i.e., that the underlay

source address of the data message matches the valid source address from the mapping table), then this indicates that the data message was genuinely sent from the workload application on the source DCN to the receiving DCN. If the logical source address of the data message is invalid (i.e., the underlay source address of the data message does not match the valid source address from the mapping table), then the data message may have been sent from a DCN that has been compromised by an attacker. Such a compromised DCN may be impersonating the other DCN to direct malicious traffic to the receiving DCN, and the packet will therefore be dropped.

[0007] In some embodiments, the mapping table is distributed to the DCN by a public cloud gateway controller, also referred to as a gateway controller. The gateway controller configures the DCN to implement the logical network and to use the mapping table for source address validation. The gateway controller operates in some embodiments on a different host computer than the host computer of the DCN, and also manages other DCNs operating on other host computers in the public datacenter. The gateway controller and the DCNs form a virtual private cloud (VPC) for a tenant of the public datacenter in some embodiments.

[0008] A controller agent also executes on the DCN in some embodiments, alongside the MFE. The controller agent receives the mapping table, as well as configuration data for the logical network, from the gateway controller. The controller agent directly configures the MFE to (1) implement the logical network using the logical network configuration data, and (2) utilize the mapping table to determine the validity of packets' source addresses.

[0009] A VPC in the public datacenter thus includes a gateway controller to manage multiple DCNs, with each DCN executing (i) an MFE to implement the logical network and (ii) a controller agent to configure each MFE (in addition to one or more applications that send and receive the data messages). The logical network may also span at least one private datacenter and other VPCs in the same public datacenter or other public datacenters. A central controller in a central controller cluster, which may be located in a private or a public datacenter, sends logical network configuration data to the gateway controller for a particular VPC. In some embodiments, the central controller also has access to MFEs operating on host computers in the private datacenter(s), which implement the logical network. The central controller can directly provide configuration data to these private datacenter MFEs. However, neither the central controller nor the gateway controller has access to the forwarding elements operating on the host

computers in the public datacenter (e.g., in virtualization software of the host computers), since the administrator is a tenant of the public datacenter instead of the owner.

[0010] The preceding Summary is intended to serve as a brief introduction to some embodiments of the invention. It is not meant to be an introduction or overview of all of the inventive subject matter disclosed in this document. The Detailed Description that follows and the Drawings that are referred to in the Detailed Description will further describe the embodiments described in the Summary as well as other embodiments. Accordingly, to understand all the embodiments described by this document, a full review of the Summary, Detailed Description and the Drawings is needed. Moreover, the claimed subject matters are not to be limited by the illustrative details in the Summary, Detailed Description and the Drawing, but rather are to be defined by the appended claims, because the claimed subject matters can be embodied in other specific forms without departing from the spirit of the subject matters.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The novel features of the invention are set forth in the appended claims. However, for purposes of explanation, several embodiments of the invention are set forth in the following figures.

[0012] **Figure 1** conceptually illustrates a physical view of a logical network of some embodiments that spans (1) an on-premises private datacenter and (2) a virtual private cloud in a public datacenter.

[0013] **Figure 2** conceptually illustrates the logical topology of the logical network in **Figure 1**, as specified by an administrator in the management plane.

[0014] **Figure 3** conceptually illustrates a process performed by a managed forwarding element of a VM operating in a virtual private cloud in a public datacenter, to detect spoofed packets.

[0015] **Figure 4** illustrates an example of the control plane of a logical network that spans a private datacenter and a virtual private cloud located in a public datacenter.

[0016] **Figure 5** illustrates an example of the data plane of a logical network that spans a private datacenter and a virtual private cloud located in a public datacenter, corresponding to the logical network in **Figure 4**.

[0017] **Figure 6** illustrates an example of packets being sent to one VM from another, compromised VM in the virtual private cloud of **Figures 4-5**.

[0018] **Figure 7** conceptually illustrates an electronic system with which some embodiments of the invention are implemented.

DETAILED DESCRIPTION OF THE INVENTION

[0020] In the following detailed description of the invention, numerous details, examples, and embodiments of the invention are set forth and described. However, it should be understood that the invention is not limited to the embodiments set forth and that the invention may be practiced without some of the specific details and examples discussed.

[0021] Some embodiments of the invention provide a novel spoofed-packet detection method for a managed forwarding element (MFE) executing on a data compute node (DCN) such as a virtual machine (VM) that operates on a host computer in a public multi-tenant datacenter. The MFE implements a logical network that connects multiple other DCNs (e.g., operating on different host computers) within the public datacenter. The host computers implement an underlying public datacenter network, also referred to as an underlay network.

[0022] The method of some embodiments initially receives a data message (such as a packet) that is directed to the DCN. The data message in some embodiments includes both (i) a logical network source address (also referred to as a logical source address) that is stored in a first header of the data message and (ii) an underlay network source address (also referred to as an underlay source address) that is stored in an encapsulating second header of the data message. The MFE determines whether the logical source address is valid by using a mapping table that maps logical source addresses to underlay source addresses. If the data message's logical source address does not match the expected logical source address listed in the mapping table for the data message's underlay source address, the MFE drops the packet.

[0023] In some embodiments, the mapping table is distributed to the DCN by a public cloud gateway controller, also referred to as a gateway controller. The gateway controller configures the DCN to implement the logical network and to use the mapping table for source address validation. The gateway controller operates in some embodiments on a different host computer than the host computer of the DCN, and also manages other DCNs operating on other host computers in the public datacenter. The gateway controller and the DCNs form a virtual private cloud (VPC) for a tenant of the public datacenter in some embodiments.

[0024] **Figure 1** conceptually illustrates a network control system 100 that includes such a gateway controller for managing an MFE 105 that executes on a DCN (in this case a virtual machine (VM) 110) in a public datacenter 120 to implement a logical network. The VM 110 operates on one of numerous host computers 115 in the public datacenter 120. In practice, each

VM actually operates on the host computer's virtualization software, which is not shown for clarity. In addition to the MFE 105, a local control agent 125 and any number of workload applications 130 also execute on the VM 110 in some embodiments. The control agent 125 directly configures the MFE 105 to implement the logical network, and the workload applications 130 (e.g., a web server, application server, database server, etc.) exchange data messages such as packets with other data compute nodes in the logical network.

[0025] In some embodiments, the logical network connects numerous member DCNs that operate on the virtualization software of host computers 115 in the public datacenter 120. Each public datacenter VM that is connected to the logical network also executes an MFE, local control agent, and workload applications. A person skilled in the art would understand that any number of DCNs such as VMs can be operated on a given host computer, even though only a single VM 110 is illustrated on host computer 115 in **Figure 1**. In some embodiments, all member VMs in the public datacenter are managed by a public cloud gateway controller (PCG) 127 which executes on another VM 130 operating on host computer 135. Collectively, the host computers 115 and 135, on which the member VMs (such as VM 110) and the gateway controller 127 operate, form a virtual private cloud (VPC) 140 for the owner of the logical network, who is one tenant of the multi-tenant public datacenter 120.

[0026] In some embodiments, the logical network implemented by the MFE 105 also spans other physical locations, such as a private datacenter 145 on the tenant's premises. These locations are connected via a network external to the networks in the private and the public datacenters, such as a wide-area network (WAN) or a public network of networks such as the Internet. Logical networks that span public and private datacenters are described in further detail in U.S. Patent Publication 2018/0062917, which is incorporated herein by reference.

[0027] The network control system 100 also includes a central management plane / central control plane (MP/CCP) cluster 150 that resides in the private datacenter 145 in some embodiments, which provides data to local controllers on each of numerous private datacenter host computers 150. These local controllers directly manage MFEs that operate on the virtualization software (not shown) of the host computers 150, because the host computers 150 in the private datacenter are owned and managed by the public datacenter tenant. As shown, VMs (or other data compute nodes) operating on the host computers 150 connect to their local host computer MFEs in order to send and receive traffic over the logical network.

[0028] In some embodiments, a cloud services manager (CSM) 155 in the private datacenter directly deploys the VPC 140 in the public datacenter and configures the gateway controller 127. In some such embodiments, the MP/CCP 150 sends configuration information for the logical network to both the local controllers in the private datacenter 145 and to the gateway controller 127 in the public datacenter 120, to configure their respective MFEs to implement the logical network 100.

[0029] As mentioned, unlike the host computers 150 in the private datacenter, the tenant does not own or manage the host computers 115 in the public datacenter. Accordingly, the network control system 100 does not have direct access to the forwarding elements 160 of the public datacenter host computers 115. The VMs 110 on these host computers 115 are assigned IP addresses by the administrator of the public datacenter (separate from the logical network addresses) and the forwarding elements 160 implement an underlying public datacenter network (also referred to as an underlay network).

[0030] **Figure 2** conceptually illustrates the logical topology 200 of some embodiments of a logical network (e.g., the logical network implemented by the MFEs 105). In some embodiments, the tenant specifies this logical topology 200 at the management plane 150. In this example, four workload VMs 205-220 are connected to a logical L2 switch 225, which in turn is connected to a logical L3 router 230 that has an uplink to an external network. For simplicity's sake, only a single logical switch and logical router is shown. In some embodiments however, there may be multiple logical switches attached to the logical router 230, the logical router 230 may have multiple logical routing components, and there may be multiple tiers of logical routers. Multiple tiers of logical routers and routing components for a logical router are described in further detail in U.S. Patent 9,787,605 titled "Logical Router with Multiple Routing Components," which is incorporated herein by reference. The logical switch 225 is assigned a subnet (e.g. 192.168.1/24), and the workload VMs are assigned addresses within the subnet. The subnet and these addresses are assigned to the VMs by the administrator via the network control system 100.

[0031] **Figure 3** conceptually illustrates a process 300 to detect spoofed packets in such an environment with MFEs implementing a logical network in a public datacenter. In some embodiments, the process 300 is performed by an MFE executing on a VM that operates in a public datacenter.

[0032] As shown, the process 300 begins by receiving (at 305) a packet. In some embodiments, the MFE receives the packet from an underlay forwarding element operating in virtualization software of the host computer on which the VM resides. This packet may have been sent from another VM on the same logical network (e.g., another VM in the same VPC as the VM on which the MFE executes).

[0033] The process 300 then identifies (at 310) the underlay network source address. In some embodiments, the packet is encapsulated using the underlay network addresses assigned to the VMs by the network administrator, which allow the forwarding elements of the host computers to forward the packet. The MFEs in the VMs are responsible, in some embodiments, for encapsulating and decapsulating the packets with these underlay network addresses.

[0034] The process 300 also identifies (at 315) the logical network source address of the packet. In some embodiments, this is the inner source address of the packet as generated by a workload application executing on the source VM. The MFE on the source VM will have received the packet with logical network addresses from the workload application and added the encapsulation addresses before forwarding the packet from the source VM.

[0035] Next, the process 300 searches (at 320) the mapping table for an entry with the identified underlay source address. The process 300 then determines (at 325) whether the identified logical network source address (in the packet) matches the expected logical network source address from the mapping table entry. This determines whether the logical network source address is a valid address for the VM that sent the packet (which is identified based on the underlay address). In some embodiments, the underlay address is assumed to be valid, as the datacenter-controlled forwarding elements will have verified this address. However, if the destination VM is compromised, that VM may be able to send packets with an incorrect source address (e.g., that of a different VM on the logical network with different privileges). Other embodiments search the mapping table for an entry for the logical network source address and perform the comparison for the underlay addresses.

[0036] When the logical source address of the packet matches the expected source address for the packet's underlay source address, the process 300 continues (at #0335) processing the packet. This further processing may include logical network processing, firewall processing based on the logical source and/or destination address, and delivery of the packet to the destination workload application on the DCN.

[0037] On the other hand, when the packet logical source address does not match the expected logical source address, the process 300 drops the packet. In this situation, the packet may have been sent from a VM that has been compromised by an attacker. Such a compromised VM may be impersonating another VM to evade a firewall or other security policy and direct malicious traffic to the receiving VM. The process 300 then ends.

[0038] The process 300 will now be illustrated with a specific example by reference to **Figures 4-6**. **Figure 4** illustrates a VPC 400 located in a public datacenter, which includes three virtual machines 405-415 and a gateway controller 425 (which operates on a fourth VM in some embodiments), all managed by a MP/CCP/CSM 430 located outside the VPC (e.g., in a private on-premises datacenter). Control data connections are denoted by dashed lines in **Figure 4**. Logical network configuration data is sent from the MP/CCP/CSM 430 to the gateway controller 425, which in turn shares the appropriate logical network configuration data with controller agents 420-430 executing on the respective VMs. The controller agents 420-430 then use the logical network configuration data to directly configure their local MFEs executing on each VM.

[0039] The VPC is assigned a subnet 10.1.0.0/16 by the public cloud administrator, and the logical switch (not shown) is assigned a subnet 192.168.1.0/24 by the logical network administrator. For the sake of simplicity, VMs 405-415 are assumed to be connected to the same logical switch, but similar concepts also apply when the VMs span multiple logical switches connected by one or more logical routers. The gateway controller 425 distributes a mapping table 435 to each VM 405-415. As shown, this mapping table 435 includes entries mapping each of the underlay network addresses of the VMs (the addresses 10.1.0.1, 10.1.0.2, etc.) to the corresponding logical network addresses for those VMs (192.168.1.1, 192.168.1.2, etc.).

[0040] **Figure 5** illustrates the same network as **Figure 4**, showing data plane connections within the VPC. As shown, MFEs 505-515 execute on each VM 405-415, and each MFE has a virtual tunnel endpoint (VTEP) that uses the underlay address for that VM. The gateway controller 425 also operates a VTEP and maintains tunnels between itself and each VM. Each VM also maintains tunnels between itself and the other VMs, denoted by solid black lines (not all tunnels are shown in the figure, for clarity).

[0041] Each VM 405-415 is assigned an IP address in the logical network subnet. This is the source address for packets in the logical network that originate from the workload applications of the VM. Each VTEP associated with an MFE on is also assigned an IP address in

the VPC subnet. The mapping of each VMs source address to its corresponding MFE's VTEP source address is recorded in the mapping table 435.

[0042] In the example of **Figure 5**, the logical network configuration permits full access to VM1 (405) from VM2 (410), and only limited access to VM1 from VM3 (415). For example, if VM1 operates as a database server and VM2 operates as web server that requires access to the data in the database on VM1 while VM3 operates a different web application that does not need access to the data in the database on VM1, then the administrator could configure an access restriction for VM3 to VM1. These access restrictions may be implemented at each MFE in some embodiments using a distributed firewall, access control lists, and/or other methods of network security. In some embodiments, these access restrictions are configured within the logical network and are therefore implemented (e.g., as firewall rules) using the logical network addresses rather than the underlay addresses.

[0043] **Figure 6** illustrates two examples of packets being sent and received between the VMs 405-415 in VPC 400. As part of its configuration, VM1 (405) has received a copy 635 of the mapping table 435 from the gateway controller 425. As noted above, VM2 (410) is permitted full access to VM1, whereas VM3 (415) is only permitted limited access. These access restrictions are enforced by a distributed firewall that is configured to operate on each VM's MFE by their local control agent.

[0044] In the first example, VM1 (405) receives a packet 615 from VM2 (410). The packet 615 includes both (1) a logical network source address (192.168.1.2) as part of a first (inner) header of the data message and (2) an underlay network source address (10.1.0.2) as part of an encapsulation second (outer) header of the packet 615. The MFE on VM1 removes the underlay network address (e.g., as part of its VTEP processing) and stores this for its spoof protection evaluation.

[0045] The MFE then searches the copy of the mapping table 635 for an entry corresponding to the underlay network source address of the packet 615. The entry for the underlay address (10.1.0.2) of the packet 615 indicates that the corresponding logical network source address for this underlay address is 192.168.1.2. The MFE then compares this expected logical source address to the actual logical source address of the packet 615 and determines that these addresses match. As such, the MFE treats the packet 615 as valid for further processing

(e.g., using the distributed firewall rules to determine that the packet is allowed and distributing the packet to the workload application).

[0046] In the second example, VM1 (405) receives another packet 625 originating from VM3 (415). However, VM3 (415) has been compromised by an attacker and is attempting to send unauthorized packets to the database server on VM1 (405). Because VM3 (415) has limited network connectivity with VM1 (405), enforced by the MFE of VM1, VM3 attempts to send malicious packets spoofing VM2 (i.e., appearing as though the packets are sent from VM2).

[0047] Accordingly, VM3 (415) sends a malicious packet 625 to VM1 using the logical source address 192.168.1.2 (the address of VM2) instead of the actual VM3 address of 192.168.1.3. However, because the attacker does not have access to the host computer forwarding element, the source underlay network address needs to be correct in order for the datacenter-controlled forwarding element at the host of VM3 to forward the packet to the forwarding element at the host of VM1.

[0048] When received by VM1 (405), the malicious packet 625 includes both (1) a logical network source address (192.168.1.2) as part of a first (inner) header of the data message and (2) an underlay network source address (10.1.0.3) as part of an encapsulation second (outer) header of the packet 625. The MFE on VM1 removes the underlay network address (e.g., as part of its VTEP processing) and stores this for its spoof protection evaluation.

[0049] The MFE then searches the copy of the mapping table 635 for an entry corresponding to the underlay network source address of the packet 625. The entry for the underlay address (10.1.0.3) of the packet 625 indicates that the corresponding logical network source address for this underlay address is 192.168.1.3. In this case, the expected logical source address (192.168.1.3) does not match the actual logical source address of the packet 625, and thus the MFE drops this packet. Without this additional check, the firewall would allow the packet because the logical packet would appear as though received from VM2.

[0050] Many of the above-described features and applications are implemented as software processes that are specified as a set of instructions recorded on a computer readable storage medium (also referred to as computer readable medium). When these instructions are executed by one or more processing unit(s) (e.g., one or more processors, cores of processors, or other processing units), they cause the processing unit(s) to perform the actions indicated in the instructions. Examples of computer readable media include, but are not limited to, CD-ROMs,

flash drives, RAM chips, hard drives, EPROMs, etc. The computer readable media does not include carrier waves and electronic signals passing wirelessly or over wired connections.

[0051] In this specification, the term "software" is meant to include firmware residing in read-only memory or applications stored in magnetic storage, which can be read into memory for processing by a processor. Also, in some embodiments, multiple software inventions can be implemented as sub-parts of a larger program while remaining distinct software inventions. In some embodiments, multiple software inventions can also be implemented as separate programs. Finally, any combination of separate programs that together implement a software invention described here is within the scope of the invention. In some embodiments, the software programs, when installed to operate on one or more electronic systems, define one or more specific machine implementations that execute and perform the operations of the software programs.

[0052] **Figure** 7 conceptually illustrates an electronic system 700 with which some embodiments of the invention are implemented. The electronic system 700 can be used to execute any of the control, virtualization, or operating system applications described above. The electronic system 700 may be a computer (e.g., a desktop computer, personal computer, tablet computer, server computer, mainframe, a blade computer etc.), phone, PDA, or any other sort of electronic device. Such an electronic system includes various types of computer readable media and interfaces for various other types of computer readable media. Electronic system 700 includes a bus 705, processing unit(s) 710, a system memory 725, a read-only memory 730, a permanent storage device 735, input devices 740, and output devices 745.

[0053] The bus 705 collectively represents all system, peripheral, and chipset buses that communicatively connect the numerous internal devices of the electronic system 700. For instance, the bus 705 communicatively connects the processing unit(s) 710 with the read-only memory 730, the system memory 725, and the permanent storage device 735.

[0054] From these various memory units, the processing unit(s) 710 retrieve instructions to execute and data to process in order to execute the processes of the invention. The processing unit(s) may be a single processor or a multi-core processor in different embodiments.

[0055] The read-only-memory (ROM) 730 stores static data and instructions that are needed by the processing unit(s) 710 and other modules of the electronic system. The permanent storage device 735, on the other hand, is a read-and-write memory device. This device is a non-

volatile memory unit that stores instructions and data even when the electronic system 700 is off. Some embodiments of the invention use a mass-storage device (such as a magnetic or optical disk and its corresponding disk drive) as the permanent storage device 735.

[0056] Other embodiments use a removable storage device (such as a floppy disk, flash drive, etc.) as the permanent storage device. Like the permanent storage device 735, the system memory 725 is a read-and-write memory device. However, unlike storage device 735, the system memory is a volatile read-and-write memory, such as random-access memory. The system memory stores some of the instructions and data that the processor needs at runtime. In some embodiments, the invention's processes are stored in the system memory 725, the permanent storage device 735, and/or the read-only memory 730. From these various memory units, the processing unit(s) 710 retrieve instructions to execute and data to process in order to execute the processes of some embodiments.

[0057] The bus 705 also connects to the input and output devices 740 and 745. The input devices enable the user to communicate information and select commands to the electronic system. The input devices 740 include alphanumeric keyboards and pointing devices (also called "cursor control devices"). The output devices 745 display images generated by the electronic system. The output devices include printers and display devices, such as cathode ray tubes (CRT) or liquid crystal displays (LCD). Some embodiments include devices such as a touchscreen that function as both input and output devices.

[0058] Finally, bus 705 also couples electronic system 700 to a network 765 through a network adapter (not shown). In this manner, the computer can be a part of a network of computers (such as a local area network ("LAN"), a wide area network ("WAN"), or an Intranet, or a network of networks, such as the Internet. Any or all components of electronic system 700 may be used in conjunction with the invention.

[0059] Some embodiments include electronic components, such as microprocessors, storage and memory that store computer program instructions in a machine-readable or computer-readable medium (alternatively referred to as computer-readable storage media, machine-readable media, or machine-readable storage media). Some examples of such computer-readable media include RAM, ROM, read-only compact discs (CD-ROM), recordable compact discs (CD-R), rewritable compact discs (CD-RW), read-only digital versatile discs (e.g., DVD-ROM, dual-layer DVD-ROM), a variety of recordable/rewritable DVDs (e.g., DVD-RAM,

DVD-RW, DVD+RW, etc.), flash memory (e.g., SD cards, mini-SD cards, micro-SD cards, etc.), magnetic and/or solid state hard drives, read-only and recordable Blu-Ray® discs, ultra-density optical discs, any other optical or magnetic media, and floppy disks. The computer-readable media may store a computer program that is executable by at least one processing unit and includes sets of instructions for performing various operations. Examples of computer programs or computer code include machine code, such as is produced by a compiler, and files including higher-level code that are executed by a computer, an electronic component, or a microprocessor using an interpreter.

[0060] While the above discussion primarily refers to microprocessor or multi-core processors that execute software, some embodiments are performed by one or more integrated circuits, such as application specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs). In some embodiments, such integrated circuits execute instructions that are stored on the circuit itself.

[0061] As used in this specification, the terms "computer", "server", "processor", and "memory" all refer to electronic or other technological devices. These terms exclude people or groups of people. For the purposes of the specification, the terms display or displaying means displaying on an electronic device. As used in this specification, the terms "computer readable medium," "computer readable media," and "machine readable medium" are entirely restricted to tangible, physical objects that store information in a form that is readable by a computer. These terms exclude any wireless signals, wired download signals, and any other ephemeral signals.

[0062] This specification refers throughout to computational and network environments that include virtual machines (VMs). However, virtual machines are merely one example of data compute nodes (DNCs) or data compute end nodes, also referred to as addressable nodes. DCNs may include non-virtualized physical hosts, virtual machines, containers that run on top of a host operating system without the need for a hypervisor or separate operating system, and hypervisor kernel network interface modules.

[0063] VMs, in some embodiments, operate with their own guest operating systems on a host using resources of the host virtualized by virtualization software (e.g., a hypervisor, virtual machine monitor, etc.). The tenant (i.e., the owner of the VM) can choose which applications to operate on top of the guest operating system. Some containers, on the other hand, are constructs that run on top of a host operating system without the need for a hypervisor or separate guest

operating system. In some embodiments, the host operating system isolates the containers for different tenants and therefore provides operating-system level segregation of the different groups of applications that operate within different containers. This segregation is akin to the VM segregation that is offered in hypervisor-virtualized environments, and thus can be viewed as a form of virtualization that isolates different groups of applications that operate in different containers. Such containers are more lightweight than VMs.

[0064] Hypervisor kernel network interface modules, in some embodiments, is a non-VM DCN that includes a network stack with a hypervisor kernel network interface and receive/transmit threads. One example of a hypervisor kernel network interface module is the vmknic module that is part of the ESX hypervisor of VMware Inc.

[0065] One of ordinary skill in the art will recognize that while the specification refers to VMs, the examples given could be any type of DCNs, including physical hosts, VMs, non-VM containers, and hypervisor kernel network interface modules. In fact, the example networks could include combinations of different types of DCNs in some embodiments.

[0066] While the invention has been described with reference to numerous specific details, one of ordinary skill in the art will recognize that the invention can be embodied in other specific forms without departing from the spirit of the invention. In addition, **Figure 6** conceptually illustrates a process. The specific operations of this process may not be performed in the exact order shown and described. The specific operations may not be performed in one continuous series of operations, and different specific operations may be performed in different embodiments. Furthermore, the process could be implemented using several sub-processes, or as part of a larger macro process. Thus, one of ordinary skill in the art would understand that the invention is not to be limited by the foregoing illustrative details, but rather is to be defined by the appended claims.

CLAIMS

We claim:

1. For a managed forwarding element (MFE) executing on a data compute node (DCN) that operates on a host computer in a public datacenter, the MFE implementing a logical network that connects a plurality of DCNs within the public datacenter, a method comprising:

receiving a packet directed to the DCN, wherein the packet (i) has a first logical network source address and (ii) is encapsulated with a second source address associated with an underlying public datacenter network;

determining whether the first logical network source address is a valid source address for the packet based on a mapping table that maps logical network addresses to underlying public datacenter network addresses; and

when the first source address is not a valid source address for the packet, dropping the packet.

2. The method of claim 1, wherein the host computer is a first host computer, wherein the first host computer receives the mapping table from a network controller that (i) operates on a second host computer in the public datacenter and (ii) configures the MFE to implement the logical network.

3. The method of claim 2, wherein the network controller distributes the mapping table to a controller agent executing on the first DCN, said controller agent directly configuring the MFE to implement the logical network and to use the mapping table.

4. The method of claim 2, wherein the network controller operating on the second host computer is a first network controller that manages a plurality of MFEs operating in the public datacenter to implement the logical network, wherein the first network controller receives logical network configuration data from a second network controller operating in a second datacenter.

5. The method of claim 4, wherein:

the second network controller has access to and provides configuration data for managed forwarding elements operating in virtualization software of a plurality of host computers of the second datacenter, and

the second network controller does not have access to forwarding elements operating in virtualization software of the first and second host computers of the first datacenter.

6. The method of claim 1, wherein:

the DCN is a first DCN,

the first logical network source address is a logical network address for applications executing on a second DCN, and

the second source address is an address assigned to the second DCN by the public datacenter.

7. The method of claim 1, wherein a workload application executes on the DCN alongside the MFE, wherein the packet is directed to the workload application.

8. The method of claim 7, wherein the packet has a first logical network destination address associated with the workload application, and wherein the encapsulation includes a second destination address associated with the DCN on the underlying public datacenter network.

9. The method of claim 7, wherein when the first source address is a valid source address for the packet, the MFE forwards the packet to the workload application.

10. The method of claim 1, wherein determining whether the first logical network source address is a valid source address for the packet comprises:

identifying an entry of the mapping table for the first logical network source address;

identifying a third source address associated with the underlying public datacenter network that is a valid source address for the first logical network source address according to the mapping table entry; and

determining whether the third source address matches the second source address.

11. The method of claim 10, wherein the third source address is an Internet Protocol (IP) address assigned by the public datacenter to a second DCN, wherein the packet is received from a third DCN that has been compromised by an attacker and is impersonating the second DCN to direct traffic to the first DCN.

12. A non-transitory machine readable medium storing a managed forwarding element (MFE) which when executed on a data compute node (DCN) operating on a host computer in a public datacenter implements a logical network that connects a plurality of DCNs within the public datacenter, the program comprising sets of instructions for:

receiving a packet directed to the DCN, wherein the packet (i) has a first logical network source address and (ii) is encapsulated with a second source address associated with an underlying public datacenter network;

determining whether the first logical network source address is a valid source address for the packet based on a mapping table that maps logical network addresses to underlying public datacenter network addresses; and

when the first source address is not a valid source address for the packet, dropping the packet.

13. The non-transitory machine readable medium of claim 12, wherein the host computer is a first host computer, wherein the first host computer receives the mapping table from a network controller that (i) operates on a second host computer in the public datacenter and (ii) configures the MFE to implement the logical network.

14. The non-transitory machine readable medium of claim 13, wherein the network controller distributes the mapping table to a controller agent executing on the first DCN, said controller agent directly configuring the MFE to implement the logical network and to use the mapping table.

15. The non-transitory machine readable medium of claim 13, wherein:

the network controller operating on the second host computer is a first network controller that manages a plurality of MFEs operating in the public datacenter to implement the logical network;

the first network controller receives logical network configuration data from a second network controller operating in a second datacenter;

the second network controller has access to and provides configuration data for managed forwarding elements operating in virtualization software of a plurality of host computers of the second datacenter; and

the second network controller does not have access to forwarding elements operating in virtualization software of the first and second host computers of the first datacenter.

16. The non-transitory machine readable medium of claim 12, wherein:

the DCN is a first DCN,

the first logical network source address is a logical network address for applications executing on a second DCN, and

the second source address is an address assigned to the second DCN by the public datacenter.

17. The non-transitory machine readable medium of claim 12, wherein a workload application executes on the DCN alongside the MFE, wherein the packet is directed to the workload application.

18. The non-transitory machine readable medium of claim 17, wherein the packet has a first logical network destination address associated with the workload application, and wherein the encapsulation includes a second destination address associated with the DCN on the underlying public datacenter network.

19. The non-transitory machine readable medium of claim 17, wherein when the first source address is a valid source address for the packet, the MFE forwards the packet to the workload application.

20. The non-transitory machine readable medium of claim 12, wherein the set of instructions for determining whether the first logical network source address is a valid source address for the packet comprises sets of instructions for:

identifying an entry of the mapping table for the first logical network source address;

identifying a third source address associated with the underlying public datacenter network that is a valid source address for the first logical network source address according to the mapping table entry, wherein the third source address is an Internet Protocol (IP) address assigned by the public datacenter to a second DCN; and

determining whether the third source address matches the second source address,

wherein the packet is received from a third DCN that has been compromised by an attacker and is impersonating the second DCN to direct traffic to the first DCN.

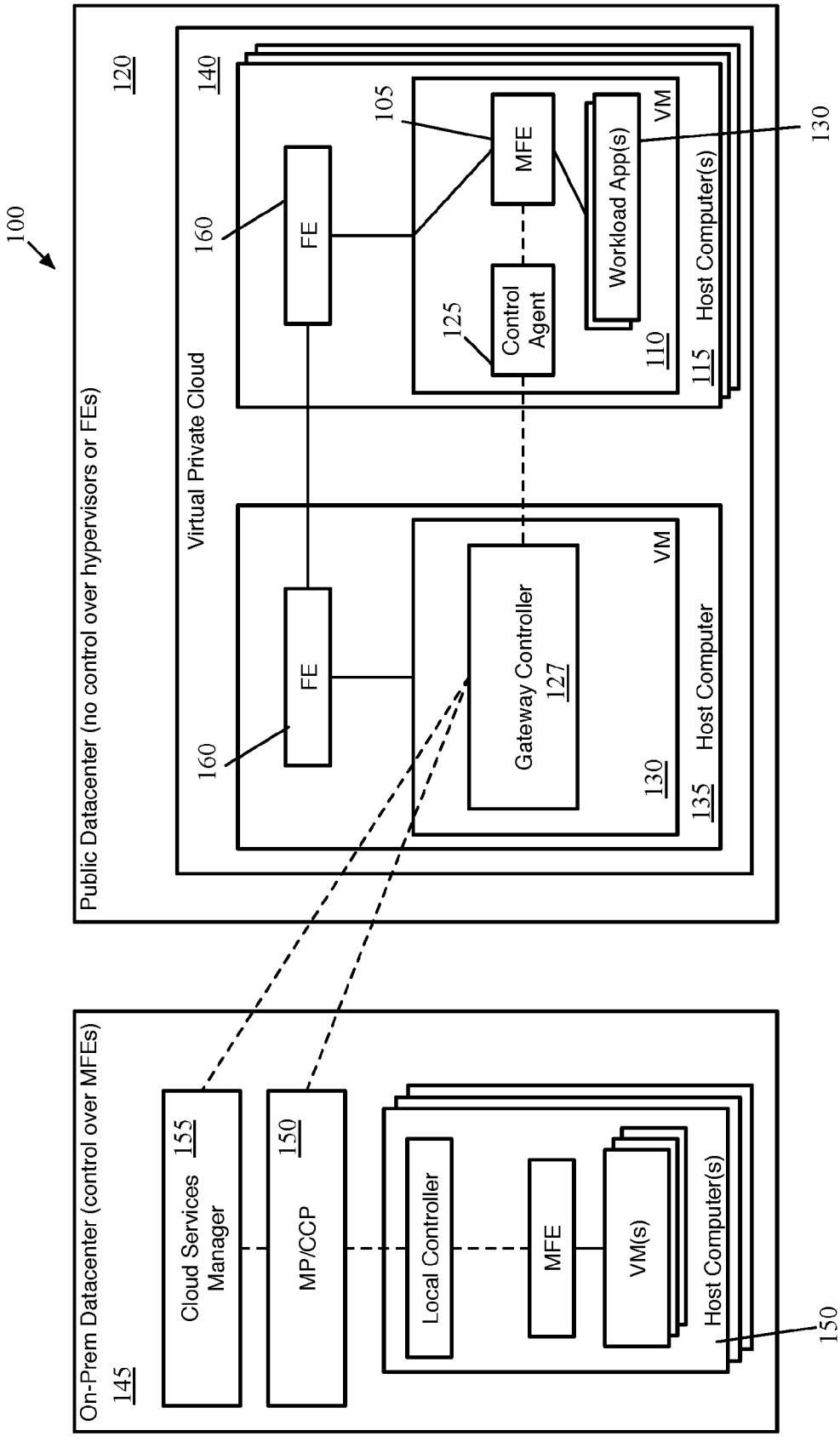
100
Public Datacenter (no control over hypervisors or FEs)
120
Virtual Private Cloud
140
160
FE
105
MFE
VM
Workload App(s)
Host Computer(s)
115
125
Control Agent
110
130
160
FE
Gateway Controller
127
VM
Host Computer
135
130
On-Prem Datacenter (control over MFEs)
145
Cloud Services Manager
155
MP/CCP
150
Local Controller
MFE
VM(s)
Host Computer(s)
150

Figure 1

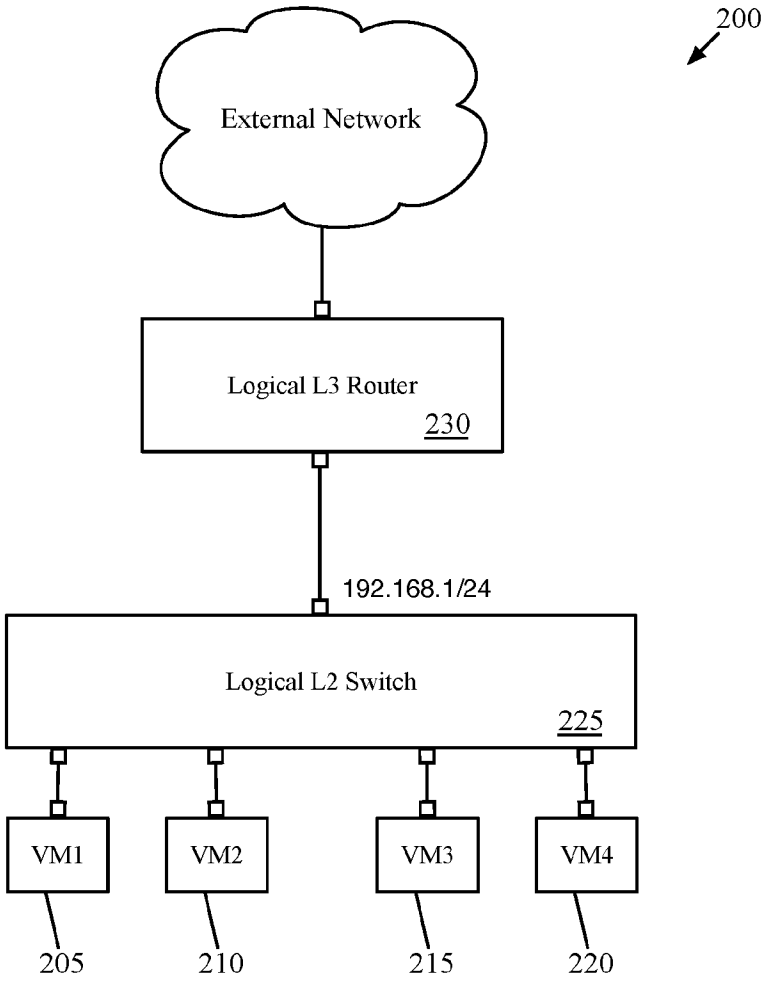
200
External Network
Logical L3 Router
230
192.168.1/24
Logical L2 Switch
225
VM1
VM2
VM3
VM4
205
210
215
220

Figure 2

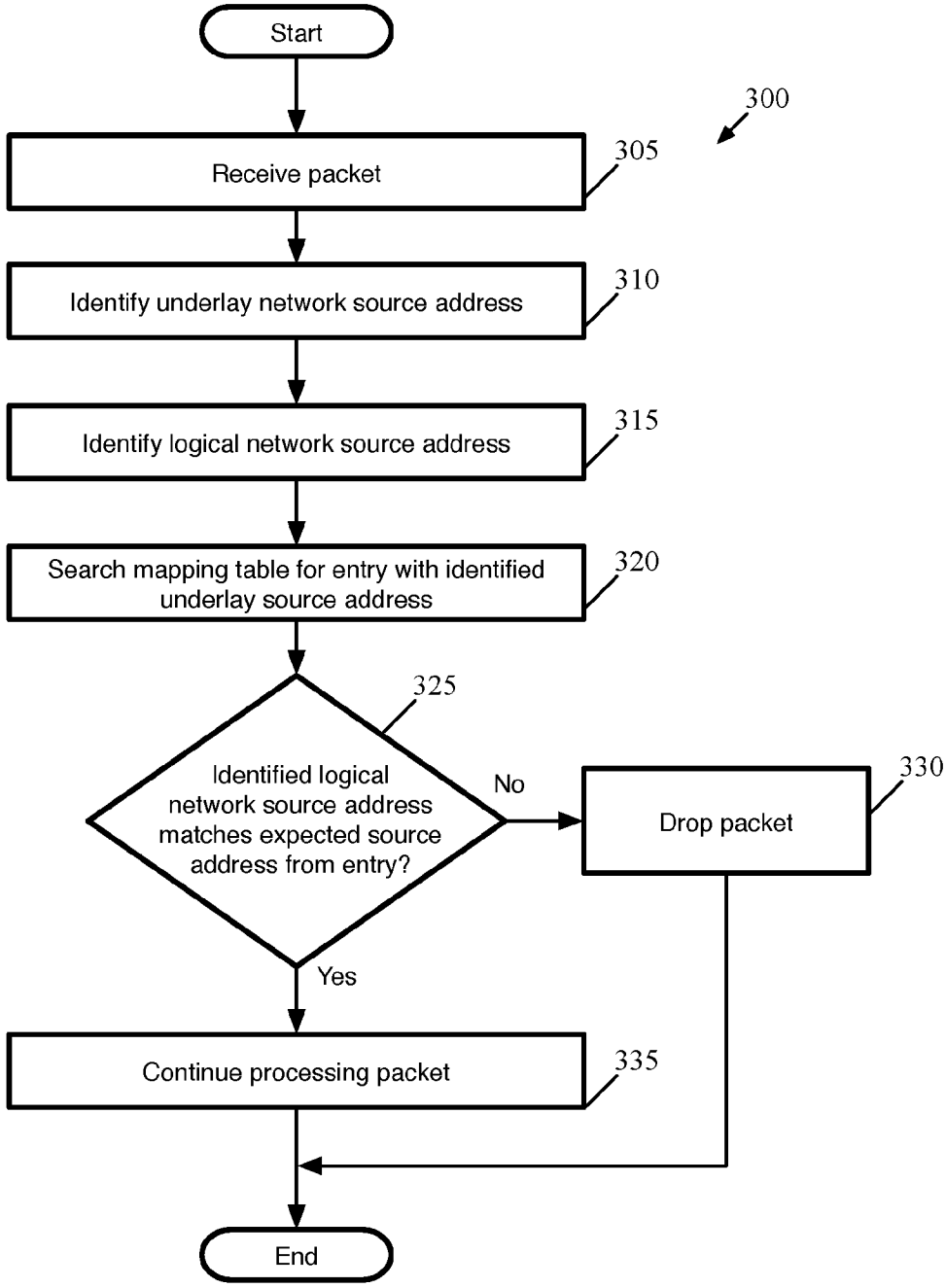
Start
300
Receive packet
305
Identify underlay network source address
310
Identify logical network source address
315
Search mapping table for entry with identified underlay source address
320
325
Identified logical network source address matches expected source address from entry?
No
Drop packet
330
Yes
Continue processing packet
335
End

Figure 3

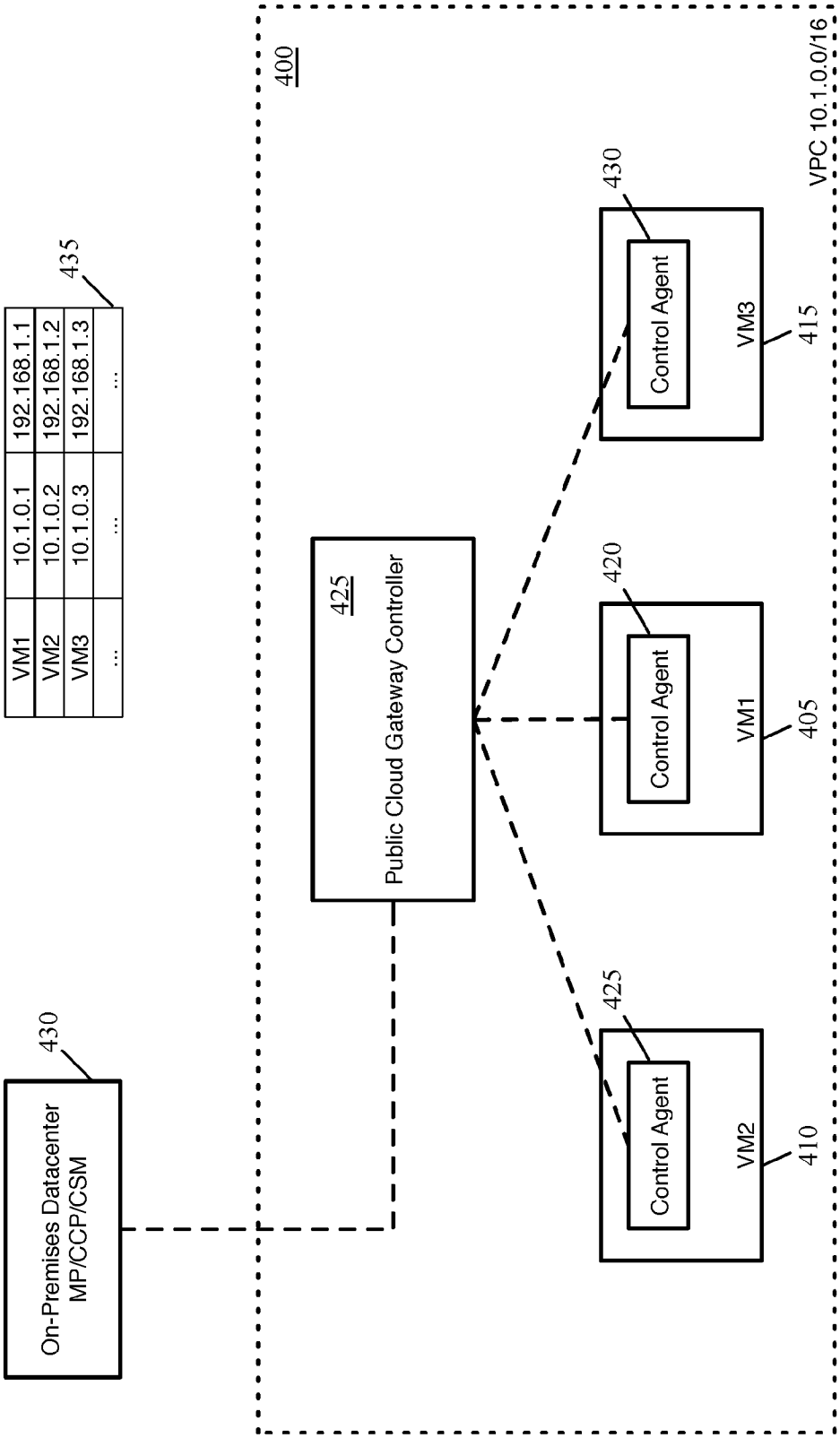
400
VPC 10.1.0.0/16
On-Premises Datacenter
MP/CCP/CSM
430
Public Cloud Gateway Controller
425
Control Agent
VM1
405
420
Control Agent
VM2
410
425
Control Agent
VM3
415
430
VM1 10.1.0.1 192.168.1.1
VM2 10.1.0.2 192.168.1.2
VM3 10.1.0.3 192.168.1.3
... ...
435

Figure 4

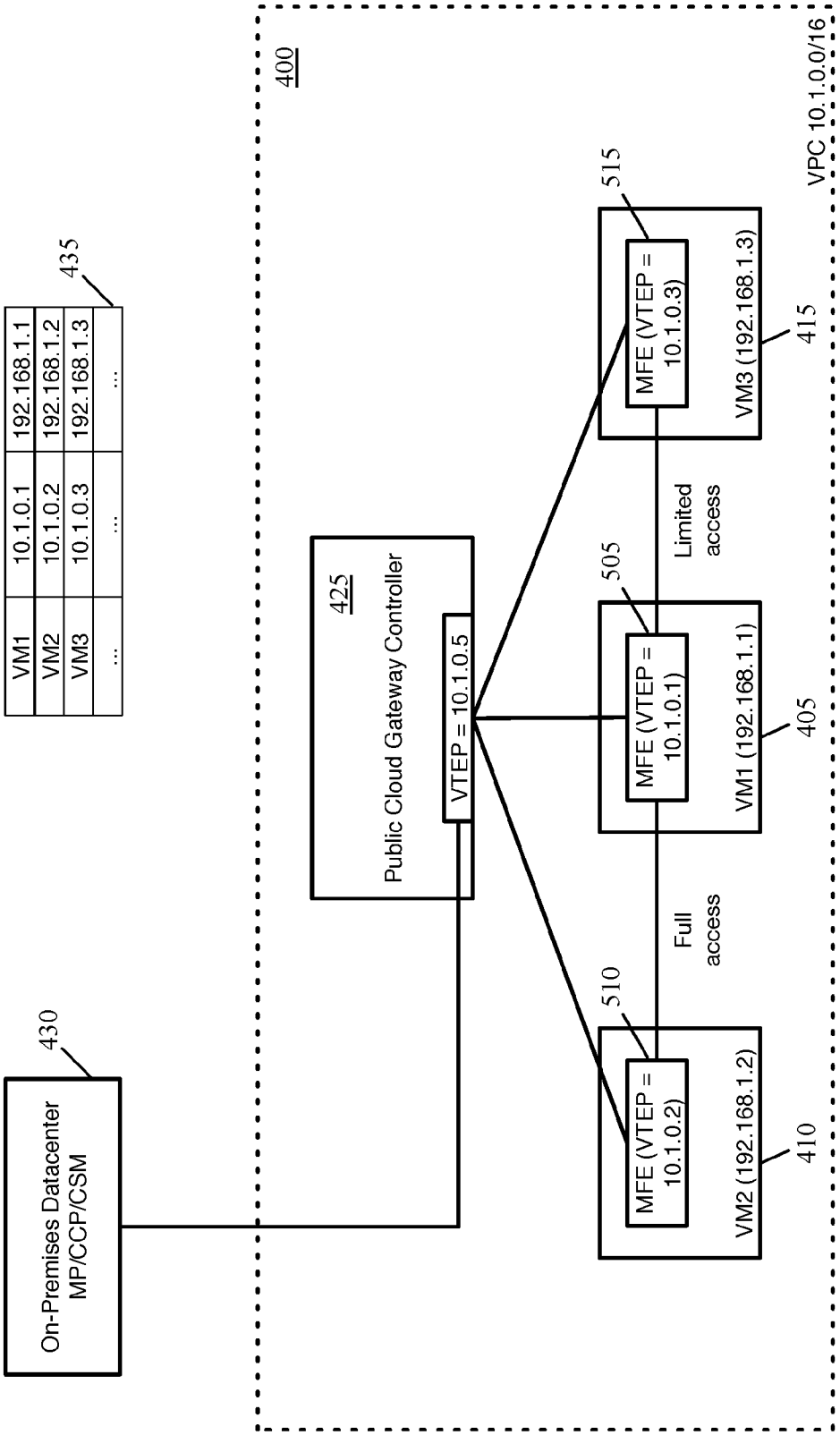
400
VPC 10.1.0.0/16
On-Premises Datacenter
MP/CCP/CSM
430
Public Cloud Gateway Controller
425
VTEP = 10.1.0.5
VM1 (192.168.1.1)
405
MFE (VTEP = 10.1.0.1)
505
VM2 (192.168.1.2)
410
MFE (VTEP = 10.1.0.2)
510
VM3 (192.168.1.3)
415
MFE (VTEP = 10.1.0.3)
515
Full access
Limited access
VM1 10.1.0.1 192.168.1.1
VM2 10.1.0.2 192.168.1.2
VM3 10.1.0.3 192.168.1.3
... ...
435

Figure 5

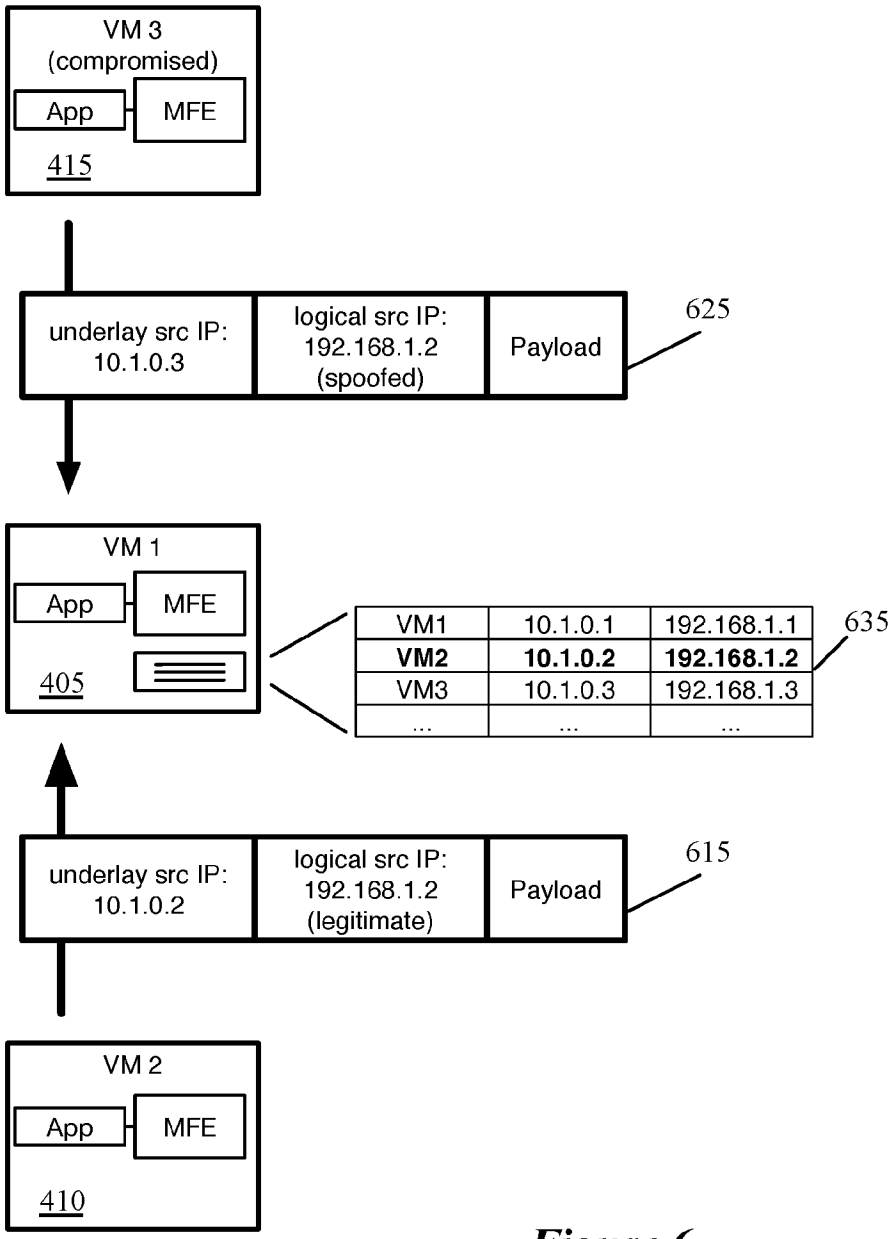
VM 3
(compromised)
App
MFE
415
underlay src IP:
10.1.0.3
logical src IP:
192.168.1.2
(spoofed)
Payload
625
VM 1
App
MFE
405
VM1 10.1.0.1 192.168.1.1
VM2 10.1.0.2 192.168.1.2
VM3 10.1.0.3 192.168.1.3
... ...
635
underlay src IP:
10.1.0.2
logical src IP:
192.168.1.2
(legitimate)
Payload
615
VM 2
App
MFE
410

Figure 6

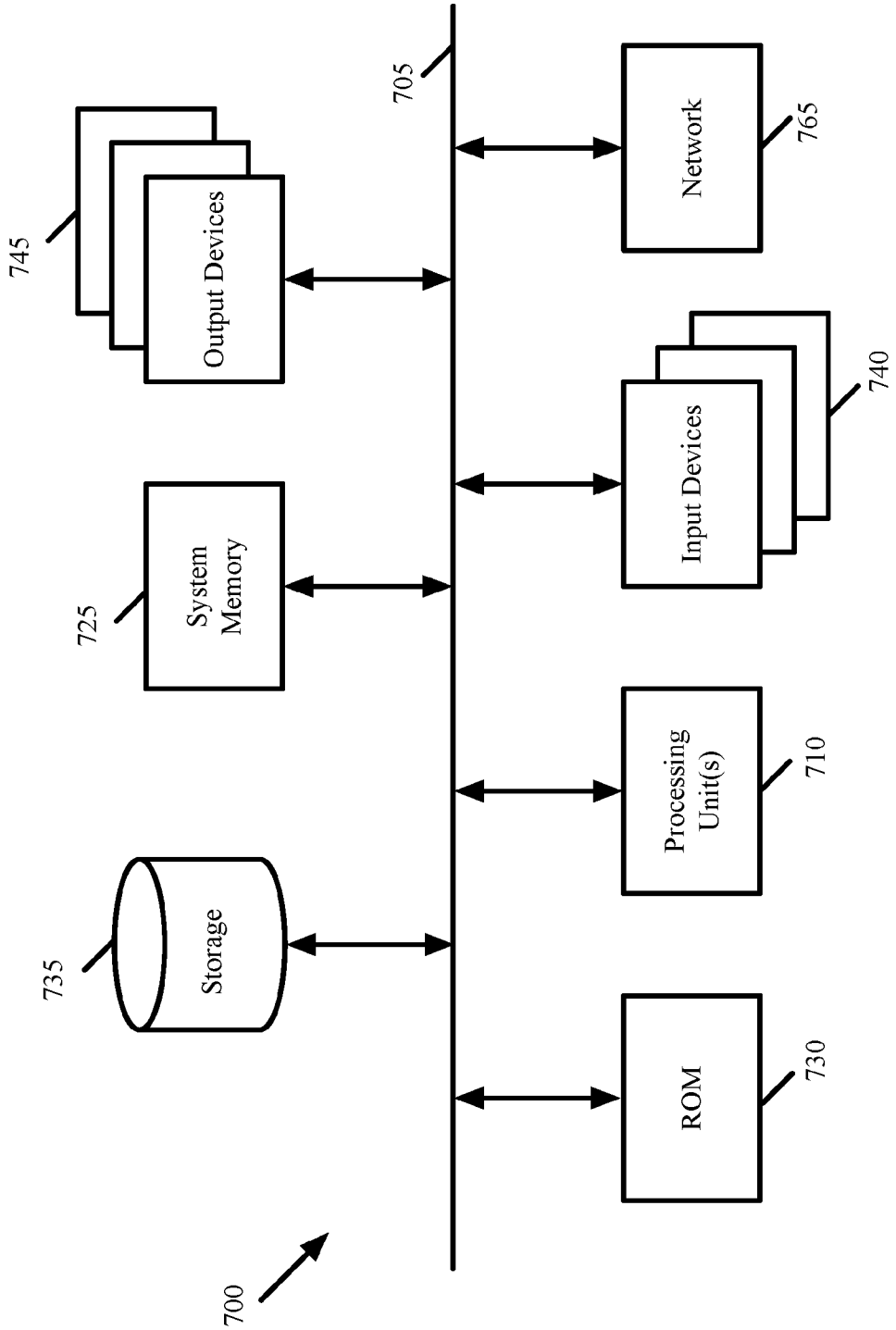
700
705
745
Output Devices
725
System Memory
735
Storage
Network
765
Input Devices
740
Processing Unit(s)
710
ROM
730

Figure 7

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2019/036644

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

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 practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 360 245 B1 (RAMACHANDRAN VIYYOKARAN R [IN] ET AL) 15 April 2008 (2008-04-15) column 2, line 45 - column 3, line 15 column 6, line 38 - column 7, line 28	1-20
X	US 7 523 485 B1 (KWAN PHILIP [US]) 21 April 2009 (2009-04-21) column 3, line 43 - column 6, line 55	1-20
A	US 2015/281274 A1 (MASUREKAR UDAY [US] ET AL) 1 October 2015 (2015-10-01) paragraphs [0004] - [0006]	1-20

☐ 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
22 August 2019	29/08/2019

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, Fax: (+31-70) 340-3016	Veen, Gerardus

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2019/036644

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7360245 B1	15-04-2008	NONE	
US 7523485 B1	21-04-2009	US 7523485 B1	21-04-2009
		US 7562390 B1	14-07-2009
		US 2009254973 A1	08-10-2009
		US 2009265785 A1	22-10-2009
		US 2009307773 A1	10-12-2009
		US 2012011584 A1	12-01-2012
US 2015281274 A1	01-10-2015	NONE	