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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 June 2018 (21.06.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/108261 A1

(51) International Patent Classification:
H04L 29/06 (2006.01) *H04L 9/32* (2006.01)
H04W 12/08 (2009.01) *H04W 36/00* (2009.01)

(21) International Application Number:
PCT/EP2016/080936

(22) International Filing Date:
14 December 2016 (14.12.2016)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **NOKIA TECHNOLOGIES OY** [FI/FI]; Karaportti 3, 02610 Espoo (FI).

(72) Inventors: **VESTERINEN, Seppo Ilmari**; Lillukkakuja 8, 90460 Oulunsalo (FI). **MAEDER, Andreas**; Erthalstr. 46, 97074 Würzburg (DE). **GODIN, Philippe**; 20 Rue de la Chaumiere, 78000 Versailles (FR).

(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, 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, 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).

(54) Title: HANDOVER IN COMMUNICATIONS NETWORK

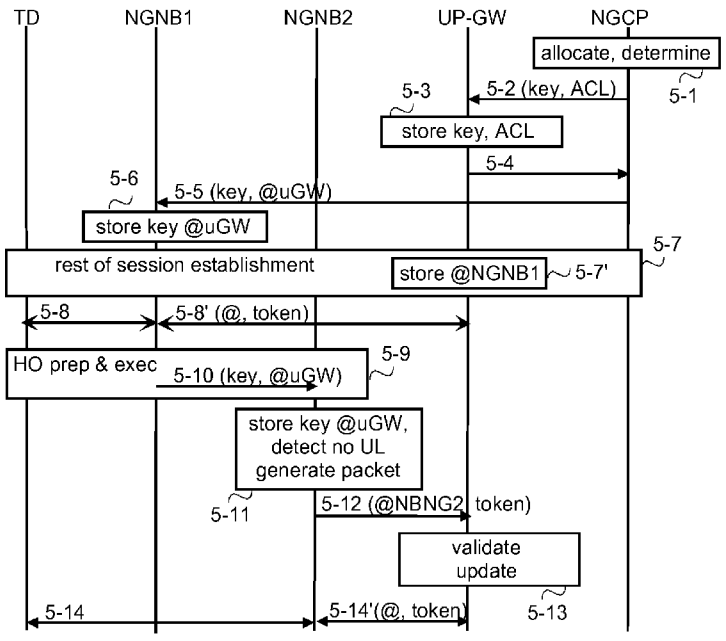

FIG.5

(57) Abstract: To validate a source node and/or path, a header of a data packet comprises a source node address, a data path identifier and a token. The token is calculated using a secret shared by the source node and a receiving node that receives the data packet. The receiving node checks whether the source node address is in a list of allowed addresses, calculates a further token by using the secret and the received data path identifier, and discards the data packet if the source node address is not in the list or if the token is different from the further token.

WO 2018/108261 A1

[Continued on next page]

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

HANDOVER IN COMMUNICATIONS NETWORK

TECHNICAL FIELD

The invention relates to communications.

BACKGROUND

In recent years the phenomenal growth of mobile services and proliferation of smart phones and tablets have increased a demand for higher network capacity. One way to increase the capacity is to use smaller cells. That in turn increases the amount of handovers. When a handover of a user device takes place between cells provided by different base stations, the handover requires several signaling messages, also to and from a control plane node in a core network. Taking into account the number of moving user devices, this leads to lots of control plane signaling.

BRIEF DESCRIPTION

The invention is defined by the subject-matter of the independent claims. Some embodiments are defined in the dependent claims.

According to an aspect, there is provided a method comprising: checking, by a network node, in response to receiving a data packet, whether or not a source node address in a header of the data packet is in a list of allowed addresses, calculating, by the network node, a token using a data path identifier in the header and a secret shared between the network and the source node, comparing the token and a token in the header, causing forwarding the data packet to further processing if the source address is in the list of allowed addresses and the token calculated is the same as the token in the header, otherwise discarding the data packet.

In an embodiment, the method further comprises: checking before calculating the token whether or not to use a token check, if the token check is to be used, performing the calculating and comparing, and if the token check is not to be used, causing forwarding the data packet to further processing if the source address is in the list of allowed addresses.

In an embodiment, wherein the data packet is an uplink packet, the method further comprises: maintaining in the network node information on the last source node address wherefrom a previous uplink data packet over the data path was received, using the source node address as a target address for downlink data packets of the data path, comparing the source node address to the last

source node address; and updating the last source node address to be the source node address, if the addresses are different.

In an embodiment the data path is a tunnel between the network node and the source node, the data path identifier is a tunnel identifier and the header is a tunnel header.

According to an aspect, there is provided a method comprising: calculating, in response to receiving in a network node a data packet to be sent over a data path to a target node, a token using a data path identifier of the data path and a secret shared between the network node and the target node, adding the token and an address of the network node to a header of the data packet, and causing forwarding the data packet with the header to the target node.

In an embodiment, the method further comprises: checking before calculating the token whether or not to use a token check, and if the token check is to be used, performing the calculating and adding the token to the header.

In an embodiment, wherein the data packet is an uplink packet, the method further comprises: causing generating, in response to the data path being handed over to the network node from another network node, an uplink data packet, calculating a token using the data path identifier of the data path and the secret, adding the token and an address of the network node to a header of the uplink data packet, and causing forwarding the uplink data packet with the header to the target node.

In an embodiment the data path is a tunnel between the network node and the target node, the data path identifier is a tunnel identifier and the header is a tunnel header.

According to an aspect, there is provided a network node comprising at least one processor, and at least one memory comprising a computer program code, wherein the processor, the memory, and the computer program code are configured to cause the network node to: check, in response to a reception of a data packet, whether or not a source node address in a header of the data packet is in a list of allowed addresses, calculate a token using a data path identifier in the header and a secret shared between the network and the source node, compare the token with a token in the header, cause forwarding the data packet to further processing in response to the source address being in the list of allowed addresses and the token calculated being the same as the token in the header, and discard the data packet in response to the source address not being in the list of allowed addresses or the token calculated not being the same as the token in

the header.

In an embodiment, the processor, the memory, and the computer program code are further configured to cause the network node to: check before calculating the token whether or not to use a token check, perform, in response to the token check being to be used, the calculating and comparing, cause, in response to the token check being to be used and the source address being in the list of allowed addresses, forwarding the data packet to further processing.

In an embodiment, the processor, the memory, and the computer program code are further configured to cause the network node to: maintain in the network node information on the last source node address wherefrom a previous uplink data packet over the data path was received, use the source node address as a target address for downlink data packets of the data path, compare the source node address to the last source node address, and update, in response to the addresses being different, the last source node address to be the source node address.

In an embodiment the network node is a user plane gateway.

According to an aspect, there is provided a network node comprising at least one processor, and at least one memory comprising a computer program code, wherein the processor, the memory, and the computer program code are configured to cause the network node to: calculate, in response to a reception of a data packet to be sent over a data path to a target node, a token using a data path identifier of the data path and a secret shared between the network node and the target node, add the token and an address of the network node to a header of the data packet, and cause forwarding the data packet with the header to the target node.

In an embodiment, the processor, the memory, and the computer program code are further configured to cause the network node to check before calculating the token whether or not to use a token check, and perform, in response to the token check being to be used, the calculating and adding the token to the header.

In an embodiment, the processor, the memory, and the computer program code are further configured to cause the network node to: generate, in response to the data path being handed over to the network node from another network node, an uplink data packet; calculate a token using the data path identifier of the data path and the secret; add the token and an address of the network node to a header of the uplink data packet; and cause forwarding the uplink data

packet with the header to the target node.

In an embodiment the network node is a base station.

According to an aspect, there is provided a network node comprising means for carrying out any one of the above-described methods.

According to an aspect there is provided a non-transitory computer readable media having stored thereon instructions that, when executed by a computing device, cause the computing device to: check, in response to a reception of a data packet, whether or not a source node address in a header of the data packet is in a list of allowed addresses, calculate a token using a data path identifier in the header and a secret shared between the network and the source node, compare the token with a token in the header, cause forwarding the data packet to further processing in response to the source address being in the list of allowed addresses and the token calculated being the same as the token in the header, and discard the data packet in response to the source address not being in the list of allowed addresses or the token calculated not being the same as the token in the header.

In an embodiment there is provided a non-transitory computer readable media having stored thereon further instructions that, when executed by a computing device, cause the computing device further to: check before calculating the token whether or not to use a token check, perform, in response to the token check being to be used, the calculating and comparing, and cause, in response to the token check being to be used and the source address being in the list of allowed addresses, forwarding the data packet to further processing.

In an embodiment there is provided a non-transitory computer readable media having stored thereon further instructions that, when executed by a computing device, cause the computing device further to: maintain in the network node information on the last source node address wherefrom a previous uplink data packet over the data path was received, use the source node address as a target address for downlink data packets of the data path, compare the source node address to the last source node address, and update, in response to the addresses being different, the last source node address to be the source node address.

In an embodiment there is provided a non-transitory computer readable media having stored thereon instructions that, when executed by a computing device, cause the computing device to: calculate, in response to a reception of a data packet to be sent over a data path to a target node, a token using a data path

identifier of the data path and a secret shared between the network node and the target node, add the token and an address of the network node to a header of the data packet, and cause forwarding the data packet with the header to the target node.

In an embodiment there is provided a non-transitory computer readable media having stored thereon further instructions that, when executed by a computing device, cause the computing device further to: check before calculating the token whether or not to use a token check, and perform, in response to the token check being to be used, the calculating and adding the token to the header.

In an embodiment there is provided a non-transitory computer readable media having stored thereon further instructions that, when executed by a computing device, cause the computing device further to: generate, in response to the data path being handed over to the network node from another network node, an uplink data packet; calculate a token using the data path identifier of the data path and the secret; add the token and an address of the network node to a header of the uplink data packet; and cause forwarding the uplink data packet with the header to the target node.

According to yet another aspect, there is provided a computer program product comprising program instructions configuring a network node to perform any of the above-described methods when the computer program is run.

One or more examples of implementations are set forth in more detail in the accompanying drawings and the description below. Other features will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

In the following embodiments will be described in greater detail with reference to the attached drawings, in which

Figures 1A and 1B illustrate exemplified wireless communication systems with schematic block diagrams of some nodes;

Figure 2 to 5 illustrate exemplified processes; and

Figure 6 and 7 are schematic block diagrams.

DETAILED DESCRIPTION OF SOME EMBODIMENTS

The following embodiments are exemplifying. Although the specification may refer to “an”, “one”, or “some” embodiment(s) and/or example(s) in several locations of the text, this does not necessarily mean that each reference is made to the same embodiment(s) or example(s), or that a particular feature only

applies to a single embodiment and/or example. Single features of different embodiments and/or examples may also be combined to provide other embodiments and/or examples.

Embodiments and examples described herein may be implemented in any communications system, wired or wireless, that are configured to support mobility of user devices by handovers, such as in at least one of the following: Universal Mobile Telecommunication System (UMTS, 3G) based on basic wide-band-code division multiple access (W-CDMA), high-speed packet access (HSPA), Long Term Evolution (LTE), LTE-Advanced, LTE-Advanced Pro, fifth generation (5G) system, beyond 5G, and/or wireless local area networks (WLAN) based on IEEE 802.11 specifications on IEEE 802.15 specifications. The embodiments are not, however, restricted to the systems given as an example but a person skilled in the art may apply the solution to other communication systems provided with necessary properties. One example of a suitable communications system is the 5G system, as listed above.

5G has been envisaged to use multiple-input-multiple-output (MIMO) multi-antenna transmission techniques, more base stations or access nodes than the current network deployments of LTE, by using a so-called small cell concept including macro sites operating in co-operation with smaller local area access nodes, such as local ultra-dense deployment of small cells, and perhaps also employing a variety of radio technologies for better coverage and enhanced data rates. 5G will likely be comprised of more than one radio access technology (RAT), each optimized for certain use cases and/or spectrum. 5G system may also incorporate both cellular (3GPP) and non-cellular (e.g. IEEE) technologies. 5G mobile communications will have a wider range of use cases and related applications including video streaming, augmented reality, different ways of data sharing and various forms of machine type applications, including vehicular safety, different sensors and real-time control. 5G is expected to have multiple radio interfaces, including apart from earlier deployed frequencies below 6GHz, also higher, that is cmWave and mmWave frequencies, and also being capable of integrating with existing legacy radio access technologies, such as the LTE. Integration with the LTE may be implemented, at least in the early phase, as a system, where macro coverage is provided by the LTE and 5G radio interface access comes from small cells by aggregation to the LTE. In other words, 5G is planned to support both inter-RAT operability (such as LTE-5G) and inter-RI operability (inter-radio interface operability, such as inter-RI operability between cmWave and

mmWave). One of the concepts considered to be used in 5G networks is network slicing in which multiple independent and dedicated virtual sub-networks (network instances) may be created within the same infrastructure to run services that have different requirements on latency, reliability, throughput and mobility.

It should be appreciated that future networks will most probably utilize network functions virtualization (NFV) which is a network architecture concept that proposes virtualizing network node functions into "building blocks" or entities that may be operationally connected or linked together to provide services. A virtualized network function (VNF) may comprise one or more virtual machines running computer program codes using standard or general type servers instead of customized hardware. Cloud computing or cloud data storage may also be utilized. In radio communications this may mean node operations to be carried out, at least partly, in a server, host or node operationally coupled to a remote radio head. It is also possible that node operations will be distributed among a plurality of servers, nodes or hosts. It should also be understood that the distribution of labour between core network operations and base station operations may differ from that of the LTE or even be non-existent. Some other technology advancements probably to be used are Software-Defined Networking (SDN), Big Data, and all-IP, which may change the way networks are being constructed and managed. For example, one or more of the below described network node functionalities may be migrated to any corresponding abstraction or apparatus or device. Therefore, all words and expressions should be interpreted broadly and they are intended to illustrate, not to restrict, the embodiment.

An extremely general architecture of exemplifying systems 100, 100' to which embodiments of the invention may be applied are illustrated in Figures 1A and 1B. Figures 1A and 1B are simplified system architectures only showing some elements and functional entities, all being logical units whose implementation may differ from what is shown. It is apparent to a person skilled in the art that the system may comprise any number of the illustrated elements and functional entities.

Referring to Figure 1A a cellular communication system 100, formed by one or more cellular radio communication networks, such as the Long Term Evolution (LTE), the LTE-Advanced (LTE-A) of the 3rd Generation Partnership Project (3GPP), or the predicted future 5G solutions, are typically composed of one or more network nodes that may be of different type. An example of such network nodes is a base station 110, such as a next generation NodeB (NGNB),

providing a wide area, medium range or local area coverage 101 for terminal devices 120, for example for the terminal devices to obtain wireless access to other networks 150 such as the Internet, either directly or via a core network 103. The base stations are configured to minimize control plane signaling in handovers in the core network by a path and/or source validation mechanism. For that purpose the base station 110 comprises a radio access path management unit (ra-pm-u) 111 and in a memory 112 there may be additional information (a.i.) for the path and/or source validation mechanism. The additional information or part of it may be received during a session set up or during a handover execution and/or the additional information or part of it may be preconfigured to the base station. Examples of the additional information in a base station include one or more allowed addresses, such as IP addresses and link layer addresses, that are known to be valid addresses of communication counterparts, a secret that is shared with a communication counterpart either for all paths or path-specifically and an algorithm to derive/calculate tokens. Examples of different functionalities of the radio access packet management unit 111 with further examples of the additional handover related information will be described in more detail below.

The terminal device (TD) 120 refers to a portable computing device (equipment, apparatus), and it may also be referred to as a user device, a user terminal or a mobile terminal or a machine-type-communication (MTC) device, also called Machine-to-Machine device and peer-to-peer device. Such computing devices (apparatuses) include wireless mobile communication devices operating with or without a subscriber identification module (SIM) in hardware or in software, including, but not limited to, the following types of devices: mobile phone, smart-phone, personal digital assistant (PDA), handset, laptop and/or touch screen computer, e-reading device, tablet, game console, notebook, multimedia device, sensor, actuator, video camera, car, refrigerator, other domestic appliances, telemetry appliances, and telemonitoring appliances.

The core network 103 comprises one or more network nodes 130 providing next generation core user plane gateway function (UP-GW) that may correspond to a serving gateway or packet data network gateway in LTE. The network nodes 130 are configured to minimize control plane signaling during handovers in the core network by a path and/or source validation mechanism. For that purpose such a network node 130, called herein simply UP-GW, comprises a user plane path management unit (up-pm-u) 131 and in a memory 132 there may be additional information (a.i.) for the path and/or source validation mecha-

nism. The additional information or part of it may be received during a session set up and/or the additional information or part of it may be preconfigured to UP-GW. Examples of the additional information in UP-GW include one or more allowed addresses, such as IP addresses and link layer addresses, that are known to be valid IP addresses of communication counterpart NGNBs, a secret that is shared with a communication counterpart, either for all data paths or path-specifically, and an algorithm to derive/calculate tokens. Examples of different functionalities of the user plane path management unit 131 with further examples of the additional handover related information will be described in more detail below.

The core network 103 comprises also one or more network nodes 140 providing next generation core control plane (NGCP) functions, such as access and mobility management function. In the architecture illustrated in Figure 1A, such network nodes, called herein simply NGCP, are configured to support the path and/or source validation mechanism. For that purpose they comprise an enhanced session set up unit (e-s-u) for providing UP-GWs and NGNBs during PDU session creation with the additional information or part of it, as will be described in more detail below, especially with Figure 5.

The exemplified system architecture in Figure 1B differs from the one illustrated in Figure 1A in that respect that an operation and maintenance center (O&M) 160 in a network management system provides UP-GWs 130 and/or base stations 110 with preconfigured additional information or at least part of the additional information, as a part of their normal configuration. More precisely, in the illustrated example the additional information in the memory 112 of the base station 110 comprises as a secret between the base station and UP-GW a key, and access control list ACL. The access control list in the base station 110 contains information on allowed IP addresses and/or link layer addresses of UP-GWs that serve base stations in the local coverage area 101 of the base station 110. Correspondingly the additional information in the memory 132 of the UP-GW 130 comprises the key as the secret between the base station and UP-GW and an access control list ACL1. The access control list ACL1 contains allowed IP addresses and/or link layer addresses of all base stations in a serving area of UP-GW.

Naturally there may be systems comprising pre-configured additional information as illustrated in Figure 1B for NGNB specific shared data paths, and further have one or more enhanced session set up units for dynamic terminal device -specific paths.

In the examples below it is assumed that a tunnel with a tunnel identifier is established between NGNB and UP-GW for a path, and that the actual packet with header information is encapsulated in a data packet according a tunnel protocol, the data packet having a tunnel header (outer header). However, it should be appreciated that for one skilled in the art implementing the examples to any corresponding connectivity path is a straightforward solution.

Figure 2 illustrates an exemplified functionality in a sending network node for the mechanism to validate a path and a source in the other end point of the tunnel. The functionality may be performed by the radio access path management unit in NGNB for an uplink packet or the user plane path management unit in UP-GW for a downlink packet. Further, in the example it is assumed that the network node is configured with the additional information, at least with the key (shared secret). The key may be a temporary seed key that may be transmitted during handover or a preconfigured key shared by the sending node and the other end point and that is not transmitted during handover. Naturally the sending node is configured with one or more algorithms which to use to calculate a token, and if there are more than one algorithm the sending node comprises information which one of the algorithms to use with the other end point. In other words, the sending node comprises configuration data defining the other end point (destination) and information how to create tunnel header content.

Referring to Figure 2, when a packet to be tunneled (i.e. sent over an existing tunnel) to the other end of the path is received in block 201, a token is calculated (derived) in block 202 using the key and the tunnel identifier. Then the token, the tunnel identifier (TNL-ID) and a source address, i.e. an IP address or a link layer address used by the network node, are added in block 203 to the tunnel header and forwarding of the packet with the tunnel header is caused in block 204.

It should be appreciated that the sending node may be configured to check before block 202 whether or not to apply the token procedure, i.e. calculate the token and add it to the tunnel header, only if the token procedure is to be applied. For example, the token procedure may be applied when a packet is first time sent from NGNB over the tunnel, or the packet is first time sent from UP-GW to that specific NGNB over the tunnel, i.e. a first packet after the tunnel has been established or after a handover. A further example is to apply the token procedure at certain intervals, for every x^{th} send packet. It should be appreciated that any other rule when to apply the token procedure may be used, and the rule may

be given in the information how to create tunnel header content or as part of the additional information.

Figure 3 illustrates an exemplified functionality of UP-GW, or more precisely of the user plane path management unit, for the mechanism to validate a path and a source as a receiving network node in the other end point of the tunnel. Further, in the example it is assumed that the network node is configured with the additional information, at least with the key (shared secret), and that it maintains information on a source address wherefrom the last received packet was received. Naturally UP-GW is configured with one or more algorithms which to use to calculate a token, and if there are more than one algorithm UP-GW comprises information which one of the algorithms to use with the other end point.

Referring to Figure 3, when an uplink packet is received in block 301 in UP-GW, a source address (sender's address) is read in block 302 from the packet, from the tunnel header (outer header). Then it is checked in block 303, whether or not the source address is an allowed address. In other words, it is checked whether or not the source address is in the ACL list of UP-GW. If it is not, the source is not a trusted source, and the packet is discarded in block 304.

If the source address is an allowed address (block 303: yes), the tunnel identifier (TLD-ID) is read in block 305 from the tunnel header of the packet to check in block 306, whether a session context associated with the tunnel identifier exists. If not, the packet is discarded in block 304. Without the session context it is not possible to perform the token verification since the information needed is stored as part of the session context.

If a session context associated with the tunnel identifier exists (block 306: yes) it is checked in block 307, whether the source address is the same address that is stored to the session context as the last source address.

If the source address is not the same as the last source address (block 307: no), the packet is the first one received from that source. That indicates that a handover has been executed. To update handover to UP-GW the last source address is set in block 308 to be the source address read in block 302. That automatically also switches the downlink destination of the associated user data.

In the illustrated example the key to be used is retrieved in block 309. Depending on an implementation the key is either retrieved from the session context or from semi-permanent configuration information using the source address indicating the NGNB wherefrom the packet was received. Further, the token is read in block 310 from the tunnel header of the packet, and another token is de-

rived (calculated) in block 311 using the retrieved key and the tunnel identifier. Then the derived token is compared in block 312 to the received token. If they are the same (block 313: yes), the token is validated, and thereby also the path is validated, and the sending NGNB is determined (verified) to be a trusted sender node, and forwarding the packet to further processing, such as tunnel-identifier associated processing, is caused in block 314.

If the derived token is not the same as the received token (block 313: no), this indicates that the sending node is not a trusted sender node, and the packet is discarded in block 304.

The token comparison (token verification) increases security since a malicious user cannot eavesdrop information required to create a proper token.

In the illustrated example the source verification process, i.e. the token comparison procedure, is performed always when there is an indication of a handover and for the other packets there may be a rule which indicates whether or not to apply the token comparison procedure. Examples of such rules are described above with Figure 2. Therefore, if the source address is the same as the last source address (block 307: yes), the process described in Figure 3 proceeds to block 315 to determine whether or not to check the validity of the token. If yes, the process proceeds to block 309 to retrieve the key. If not, the process proceeds to block 314 to cause the forwarding.

It should be appreciated that in another example no token check is performed but the process proceeds directly from block 307 to block 309 to retrieve the key if the source address is the same as the last source address.

Further, it should be appreciated that even though the blocks are described in a subsequent order, blocks 309 to 314 may be performed simultaneously or before block 308 if the source address is not the same as the last source address. That ensures, especially if no token check is performed to packets whose source address is the same as the last source address (proceed from block 307 directly to block 314 without block 315 checking), that the downlink destination of the associated user data flow is not switched to NGNB that failed the token verification.

A further example is that if the token check is performed to all packets, the process may proceed from block 306 when the associated context exists to to block 309 to retrieve the key and blocks 307 and 308 (check of source address and update if needed) can be performed after block 312, i.e. after the token verification.

Figure 4 illustrates an exemplified functionality of NGNB or more precisely of the radio access path management unit, for the mechanism to validate a path and a source as a receiving network node in the other end point of the tunnel. Basically the process is similar to the one performed in UP-GW. However, there is no need to check whether or not to update sender address to correspond address after handover. Therefore there are no blocks corresponding to blocks 307 and 308. In Figures 3 and 4 similar blocks are associated with similar numbers, and detailed explanation of a block is not repeated in vain, what is stated above with Figure 3 applies also to Figure 4.

Referring to Figure 4, when a downlink packet is received in block 401 in NGNB, a source address (sender's address) is read in block 402 from the packet, from the tunnel header (outer header). Then it is checked in block 403, whether or not the source address is an allowed address. In other words, it is checked whether or not the source address is in the ACL list of the NGNB. If it is not, the source is not a trusted source, and the packet is discarded in block 404.

If the source address is an allowed address (block 403: yes), the tunnel identifier (TLD-ID) is read in block 405 from the tunnel header to check in block 406, whether a session context associated with the tunnel identifier exists. If not, the packet is discarded in block 404.

If a session context associated with the tunnel identifier exists (block 406: yes) it is checked in block 415 whether or not to check the validity of the token. If yes, the process proceeds to block 409 to retrieve the key to be used. Further, the token is read in block 410 from the tunnel header of the packet, and another token is derived (calculated) in block 411 using the retrieved key and the tunnel identifier. Then the derived token is compared in block 412 to the received token. If they are the same (block 413: yes), the token is validated, and thereby also the path is validated, and the sending UP-GW is determined (verified) to be a trusted sender node, and forwarding the packet to further processing, such as tunnel-identifier associated processing, is caused in block 414.

If the derived token is not the same as the received token (block 413: no), this indicates that the sending node is not a trusted sender node, and the packet is discarded in block 404.

If the validity of the token is not checked (block 415: no), the process proceeds to block 414 to cause the forwarding.

Figure 5 illustrates information exchange when NGCP configures UP-GW and NGNB with the additional information. In the example it is assumed that

there is one tunnel per one session per one terminal device. The protocol used for tunneling may be any tunneling protocol, such as Generic Routing Encapsulation (GRE).

Referring to Figure 5, upon starting a session establishment for a terminal device TD, the session establishment including establishment of a tunnel between NGNB1 serving TD and UP-GW, NGCP allocates in block 5-1 a unique shared secret for the session, i.e. the key used above in token verification. In the illustrated example NGCP also determines in block 5-1 allowed IP addresses for the nodes. In the illustrated example determined allowed addresses (ACL) that are to be used by UP-GW comprise at least addresses of NGNB1 and NGNB2, and determined allowed addresses that are to be used by NGNB1 comprise only UP-GW's address. However, it should be appreciated that addresses of two or more UP-GWs may be determined to be allowable. The key, ACL and other session related information are forwarded in message 5-2, such as a create session request, to UP-GW which then stores the key and ACL in block 5-3. In other words, UP-GW is configured with the additional information. UP-GW responds to message 5-2 by sending message 5-4 to NGCP.

NGCP sends message 5-5, such as PDU session set up request, to NGNB1, message 5-5 comprising the key, UP-GW's address. Upon receiving message 5-5 NGNB1 stores in block 5-6 the key and UP-GW's address. Thereby the NGCP configures NGNB1 with the additional information.

The rest of the session establishment (block 5-7) including tunnel establishment takes place between TD, NGNB1, UP-GW and NGCP as is known in the art. During the tunnel establishment UP-GW receives NGNB1's address (@NGNB1) and stores it in block 5-7' as the address of the last received packet. For example, NGNBs may be configured to generate and send a dummy uplink packet to UP-GW on behalf of TD during the session set up so that UP-GW receives the address of the other part of the tunnel.

Once the session has been established, data packets may be sent to and from TD via NGNB1, in the tunnel (messages 5-8', only one shown) with token derived as described above, and with sender address (@), and naturally, although not illustrated in Figure, with the tunnel identifier without any affect to data packets send over the air (messages 5-8). For example GRE sequence number may carry the token.

Then in the illustrated example a handover from NGNB1 to NGNB2 is prepared and executed in block 5-9, during which the additional information, i.e.

the key and UP-GW's address, are forwarded (message 5-10) to NGNB2 with all other TD-specific context data to NGNB2 (target node).

NGNB2 stores in block 5-11 the additional information to its memory. In another implementation NGNB2, as other corresponding nodes, may be configured to update, using a pre-agreed algorithm, the key received with other information received from NGNB1. Further, in the example NGNBs are configured to generate and send a dummy uplink packet to UP-GW in case there is no uplink packet received from TD just after the handover execution phase in order to finalize the handover. In the illustrated example it is detected in block 511 that there is no uplink packet received from TD, and therefore an uplink packet is generated in block 511 by the NGNB2, and the packet is processed as if it had been received from TD. In other words, a token is generated using the tunnel identifier and the key (S_ID) as described above, and the token and sender's address (@NGNB2) are added to a tunnel header of the packet and then the packet is sent (message 5-12) to UP-GW.

Upon receiving the packet UP-GW in block 5-13 validates the address, the token and then updates in its memory NGNB2 to be the other end node of the tunnel, as described above. If NGNBs are configured to update the key because of handover, UP-GW is correspondingly configured to update, upon detecting that the source address is not the same as the previous source address, the key correspondingly.

Once the session has been established, data packets may be sent to and from TD via NGNB2, in the tunnel (messages 5-14', only one shown) with token derived as described above, and with sender address (@), and with the tunnel identifier, without any affect to data packets send over the air (messages 5-10).

As is evident from the above, no control plane signaling is required in the handover – thanks to the automatic path switching in downlink, the automatic path switching being a secured one thanks to the token and sender address validation.

Although in the above examples it is assumed that the path and/or source validation is performed to both uplink and downlink packets, that need not to be the case; it suffices that the validation is performed to uplink packets. Further, the token check may be performed to uplink packets after a handover: base stations may be configured with a rule to use token check in response to sending a first uplink packet, i.e. add to the first uplink packet the token, and UP-GW may be configured to use the token check in response to an uplink packet

containing a token or in response to the source address being different than the first source address.

The blocks, related functions, and information exchanges described above by means of Figures 2 to 5 are in no absolute chronological order, and some of them may be performed simultaneously or in an order differing from the given one. Naturally similar processes for several packets/streams may run in parallel. Other functions can also be executed between them or within them, and other information may be sent. Some of the blocks or part of the blocks or one or more pieces of information can also be left out or replaced by a corresponding block or part of the block or one or more pieces of information. For example, if no tunnel header, or corresponding outer header is used in packets, the information (source address, token, path or data flow identifier) may be added to the actual packet header. Another example includes performing only the token validation, i.e. skipping the source address validation.

The techniques and methods described herein may be implemented by various means so that an apparatus/device/network node/base station/UP-GW configured to support a path and/or source validation mechanism based on at least partly on what is disclosed above with any of Figures 1A to 5, including implementing one or more functions/operations of a corresponding network node (NGNB, UP-GW) described above with an embodiment/example, for example by means of any of Figures 2 to 5, comprises not only prior art means, but also means for implementing the one or more functions/operations of a corresponding functionality described with an embodiment, for example by means of any of Figures 2 to 5, and it may comprise separate means for each separate function/operation, or means may be configured to perform two or more functions/operations. For example, one or more of the means and/or the radio access path management unit, or its sub-units, and/or the user plane path management unit, or its sub-units, described above may be implemented in hardware (one or more devices), firmware (one or more devices), software (one or more modules), or combinations thereof. For a hardware implementation, the apparatus(es) of embodiments may be implemented within one or more application-specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), processors, controllers, micro-controllers, microprocessors, logic gates, other electronic units designed to perform the functions described herein by means of Figures 1A to 5, or a combination thereof. For firmware or software,

the implementation can be carried out through modules of at least one chipset (e.g. procedures, functions, and so on) that perform the functions described herein. The software codes may be stored in a memory unit and executed by processors. The memory unit may be implemented within the processor or externally to the processor. In the latter case, it can be communicatively coupled to the processor via various means, as is known in the art. Additionally, the components of the systems described herein may be rearranged and/or complemented by additional components in order to facilitate the achievements of the various aspects, etc., described with regard thereto, and they are not limited to the precise configurations set forth in the given figures, as will be appreciated by one skilled in the art.

Figure 6 provides an apparatus (device) according to some embodiments of the invention. Figure 6 illustrates an apparatus configured to carry out the functions described above in connection with the base station (NGNB). Each apparatus may comprise one or more communication control circuitry, such as at least one processor 602, and at least one memory 604, including one or more algorithms 603, such as a computer program code (software) wherein the at least one memory and the computer program code (software) are configured, with the at least one processor, to cause the apparatus to carry out any one of the exemplified functionalities of the base station.

Referring to Figure 6, at least one of the communication control circuitries in the apparatus 600 is configured to provide the radio access path management unit, or its sub-units, and to carry out functionalities described above by means of any of Figures 2, 4 and 5 by one or more circuitries.

Figure 7 provides an apparatus (device) according to some embodiments of the invention. Figure 7 illustrates an apparatus configured to carry out the functions described above in connection with UP-GW. Each apparatus may comprise one or more communication control circuitry, such as at least one processor 702, and at least one memory 704, including one or more algorithms 703, such as a computer program code (software) wherein the at least one memory and the computer program code (software) are configured, with the at least one processor, to cause the apparatus to carry out any one of the exemplified functionalities of UP-GW.

Referring to Figure 7, at least one of the communication control circuitries in the apparatus 700 is configured to provide the user plane path management unit, or its sub-units unit, and to carry out functionalities described above by means of any of Figures 2, 3 and 5 by one or more circuitries.

Referring to Figures 6 and 7, the memory 604, 704 may be implemented using any suitable data storage technology, such as semiconductor based memory devices, flash memory, magnetic memory devices and systems, optical memory devices and systems, fixed memory and removable memory.

Referring to Figures 6 and 7, the apparatus may further comprise different interfaces 601, 701 such as one or more communication interfaces (TX/RX) comprising hardware and/or software for realizing communication connectivity according to one or more communication protocols. The communication interface may provide the apparatus with communication capabilities to communicate in the cellular communication system and enable communication between different network nodes and in apparatus 600 depicting the base station also a communication interface between the terminal device and the different network nodes, for example. The communication interface may comprise standard well-known components such as an amplifier, filter, frequency-converter, (de)modulator, and encoder/decoder circuitries and one or more antennas. The communication interfaces may comprise radio interface components providing the network node and the terminal device with radio communication capability in the cell.

As used in this application, the term 'circuitry' refers to all of the following: (a) hardware-only circuit implementations, such as implementations in only analog and/or digital circuitry, and (b) combinations of circuits and software (and/or firmware), such as (as applicable): (i) a combination of processor(s) or (ii) portions of processor(s)/software including digital signal processor(s), software, and memory(ies) that work together to cause an apparatus to perform various functions, and (c) circuits, such as a microprocessor(s) or a portion of a microprocessor(s), that require software or firmware for operation, even if the software or firmware is not physically present. This definition of 'circuitry' applies to all uses of this term in this application. As a further example, as used in this application, the term 'circuitry' would also cover an implementation of merely a processor (or multiple processors) or a portion of a processor and its (or their) accompanying software and/or firmware. The term 'circuitry' would also cover, for example and if applicable to the particular element, a baseband integrated circuit or applications processor integrated circuit for a mobile phone or a similar integrated circuit in a server, a cellular network de-vice, or another network device.

In embodiments, the at least one processor, the memory, and the computer program code form processing means or comprises one or more computer

program code portions for carrying out one or more operations according to any one of the embodiments of Figures 3 to 10 or operations thereof.

Embodiments as described may also be carried out in the form of a computer process defined by a computer program or portions thereof. Embodiments of the methods described in connection with Figures 2 to 5 may be carried out by executing at least one portion of a computer program comprising corresponding instructions. The computer program may be in source code form, object code form, or in some intermediate form, and it may be stored in some sort of carrier, which may be any entity or device capable of carrying the program. For example, the computer program may be stored on a computer program distribution medium readable by a computer or a processor. The computer program medium may be, for example but not limited to, a record medium, computer memory, read-only memory, electrical carrier signal, telecommunications signal, and software distribution package, for example. The computer program medium may be a non-transitory medium. Coding of software for carrying out the embodiments as shown and described is well within the scope of a per-son of ordinary skill in the art.

Even though the invention has been described above with reference to examples according to the accompanying drawings, it is clear that the invention is not restricted thereto but can be modified in several ways within the scope of the appended claims. Therefore, all words and expressions should be interpreted broadly and they are intended to illustrate, not to restrict, the embodiment. It will be obvious to a person skilled in the art that, as technology advances, the inventive concept can be implemented in various ways. Further, it is clear to a person skilled in the art that the described embodiments may, but are not required to, be combined with other embodiments in various ways.

CLAIMS

1. A method comprising:

checking, by a network node, in response to receiving a data packet, whether or not a source node address in a header of the data packet is in a list of allowed addresses;

calculating, by the network node, a token using a data path identifier in the header and a secret shared between the network and the source node;

comparing the token and a token in the header;

causing forwarding the data packet to further processing if the source address is in the list of allowed addresses and the token calculated is the same as the token in the header;

otherwise discarding the data packet.

2. A method as claimed in claim 1, further comprising:

checking before calculating the token whether or not to use a token check;

if the token check is to be used, performing the calculating and comparing;

if the token check is not to be used, causing forwarding the data packet to further processing if the source address is in the list of allowed addresses.

3. A method as claimed in claim 1 or 2, wherein the data packet is an uplink packet and the method further comprises:

maintaining in the network node information on the last source node address wherefrom a previous uplink data packet over the data path was received;

using the source node address as a target address for downlink data packets of the data path;

comparing the source node address to the last source node address; and

updating the last source node address to be the source node address, if the addresses are different.

4. A method as claimed in any preceding claim, wherein the data path is a tunnel between the network node and the source node, the data path identifi-

er is a tunnel identifier and the header is a tunnel header.

5. A method comprising:

calculating, in response to receiving in a network node a data packet to be sent over a data path to a target node, a token using a data path identifier of the data path and a secret shared between the network node and the target node;

adding the token and an address of the network node to a header of the data packet; and

causing forwarding the data packet with the header to the target node.

6. A method as claimed in claim 5, further comprising:

checking before calculating the token whether or not to use a token check; and

if the token check is to be used, performing the calculating and adding the token to the header.

7. A method as claimed in claim 5 or 6,

causing generating, in response to the data path being handed over to the network node from another network node, an uplink data packet;

calculating a token using the data path identifier of the data path and the secret;

adding the token and an address of the network node to a header of the uplink data packet; and

causing forwarding the uplink data packet with the header to the target node.

8. A method as claimed in claim 5, 6 or 7, wherein the data path is a tunnel between the network node and the target node, the data path identifier is a tunnel identifier and the header is a tunnel header.

9. A network node comprising:

at least one processor, and

at least one memory comprising a computer program code, wherein the processor, the memory, and the computer program code are configured to cause the network node to:

check, in response to a reception of a data packet, whether or not a

source node address in a header of the data packet is in a list of allowed addresses;

calculate a token using a data path identifier in the header and a secret shared between the network and the source node;

compare the token with a token in the header;

cause forwarding the data packet to further processing in response to the source address being in the list of allowed addresses and the token calculated being the same as the token in the header; and

discard the data packet in response to the source address not being in the list of allowed addresses or the token calculated not being the same as the token in the header.

10. A network node as claimed in claim 9, wherein the processor, the memory, and the computer program code are further configured to cause the network node to:

check before calculating the token whether or not to use a token check;

perform, in response to the token check being to be used, the calculating and comparing;

cause, in response to the token check being to be used and the source address being in the list of allowed addresses, forwarding the data packet to further processing.

11. A network node as claimed in claim 9 or 10, wherein the processor, the memory, and the computer program code are further configured to cause the network node to:

maintain in the network node information on the last source node address wherefrom a previous uplink data packet over the data path was received;

use the source node address as a target address for downlink data packets of the data path;

compare the source node address to the last source node address; and

update, in response to the addresses being different, the last source node address to be the source node address.

12. A network node as claimed in claim 9, 10 or 11, wherein the network node is a user plane gateway.

13. A network node comprising:

at least one processor, and

at least one memory comprising a computer program code, wherein the processor, the memory, and the computer program code are configured to cause the network node to:

calculate, in response to a reception of a data packet to be sent over a data path to a target node, a token using a data path identifier of the data path and a secret shared between the network node and the target node;

add the token and an address of the network node to a header of the data packet; and

cause forwarding the data packet with the header to the target node.

14. A network node as claimed in claim 13, wherein the processor, the memory, and the computer program code are further configured to cause the network node to:

check before calculating the token whether or not to use a token check; and

perform, in response to the token check being to be used, the calculating and adding the token to the header.

15. A network node as claimed in claim 13 or 14, wherein the processor, the memory, and the computer program code are further configured to cause the network node to:

generate, in response to the data path being handed over to the network node from another network node, an uplink data packet;

calculate a token using the data path identifier of the data path and the secret;

add the token and an address of the network node to a header of the uplink data packet; and

cause forwarding the uplink data packet with the header to the target node.

16. A network node as claimed in claim 13, 14 or 15, wherein the network node is a base station.

17. A network node comprising means for carrying out the method according to any one of claims 1 to 8.

18. A non-transitory computer readable media having stored thereon instructions that, when executed by a computing device, cause the computing device to:

check, in response to a reception of a data packet, whether or not a source node address in a header of the data packet is in a list of allowed addresses;

calculate a token using a data path identifier in the header and a secret shared between the network and the source node;

compare the token with a token in the header;

cause forwarding the data packet to further processing in response to the source address being in the list of allowed addresses and the token calculated being the same as the token in the header; and

discard the data packet in response to the source address not being in the list of allowed addresses or the token calculated not being the same as the token in the header.

19. A non-transitory computer readable media as claimed in claim 18, having stored thereon further instructions that, when executed by a computing device, cause the computing device further to:

check before calculating the token whether or not to use a token check;

perform, in response to the token check being to be used, the calculating and comparing;

cause, in response to the token check being to be used and the source address being in the list of allowed addresses, forwarding the data packet to further processing.

20. A non-transitory computer readable media as claimed in claim 18 or 19, having stored thereon further instructions that, when executed by a computing device, cause the computing device further to:

maintain in the network node information on the last source node address wherefrom a previous uplink data packet over the data path was received;

use the source node address as a target address for downlink data

packets of the data path;

compare the source node address to the last source node address; and

update, in response to the addresses being different, the last source node address to be the source node address.

21. A non-transitory computer readable media having stored thereon instructions that, when executed by a computing device, cause the computing device to:

calculate, in response to a reception of a data packet to be sent over a data path to a target node, a token using a data path identifier of the data path and a secret shared between the network node and the target node;

add the token and an address of the network node to a header of the data packet; and

cause forwarding the data packet with the header to the target node.

22. A non-transitory computer readable media as claimed in claim 21, having stored thereon further instructions that, when executed by a computing device, cause the computing device further to:

check before calculating the token whether or not to use a token check; and

perform, in response to the token check being to be used, the calculating and adding the token to the header.

23. A non-transitory computer readable media as claimed in claim 21 or 22, having stored thereon further instructions that, when executed by a computing device, cause the computing device further to:

generate, in response to the data path being handed over to the network node from another network node, an uplink data packet;

calculate a token using the data path identifier of the data path and the secret;

add the token and an address of the network node to a header of the uplink data packet; and

cause forwarding the uplink data packet with the header to the target node.

24. A computer program product comprising program instructions

configuring a network node to perform a method as claimed in any of claims 1 to 8 when the computer program is run.

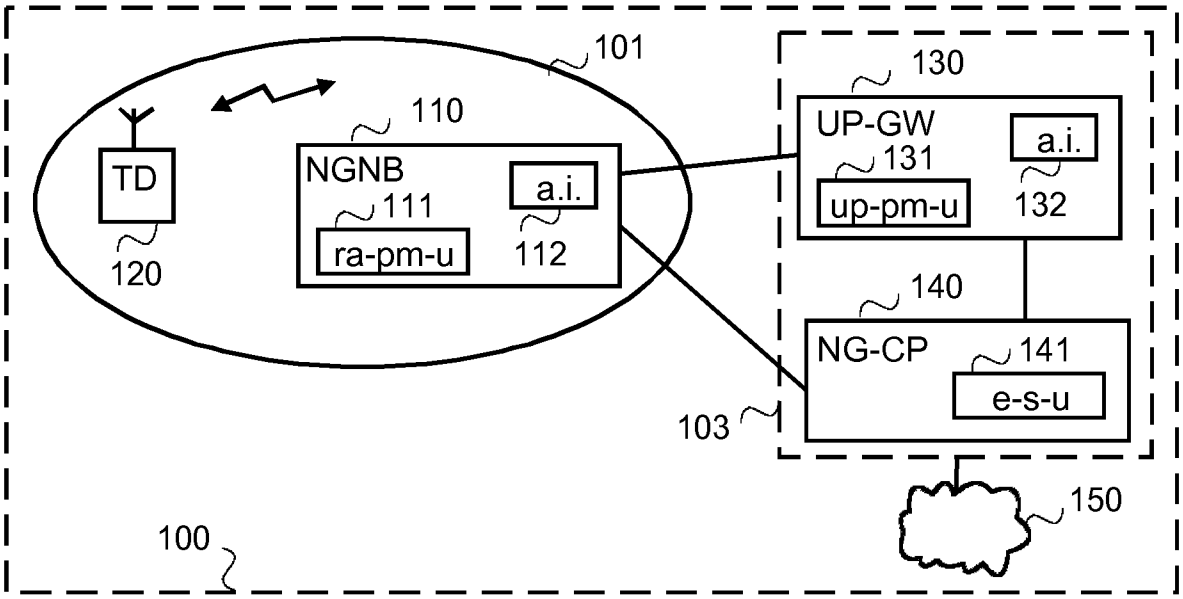

FIG.1A

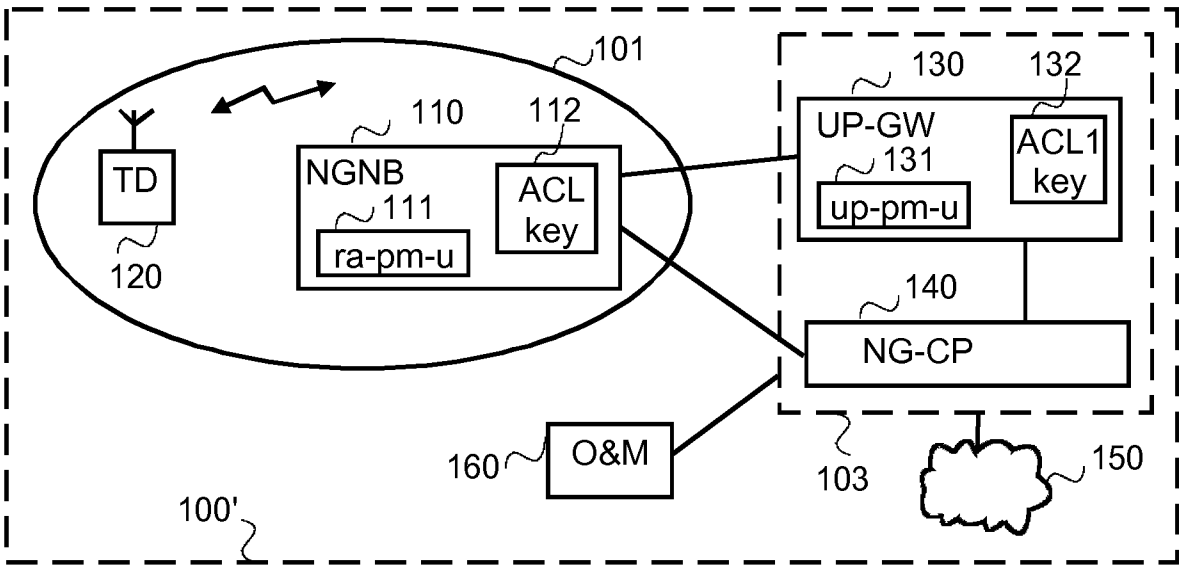

FIG.1B

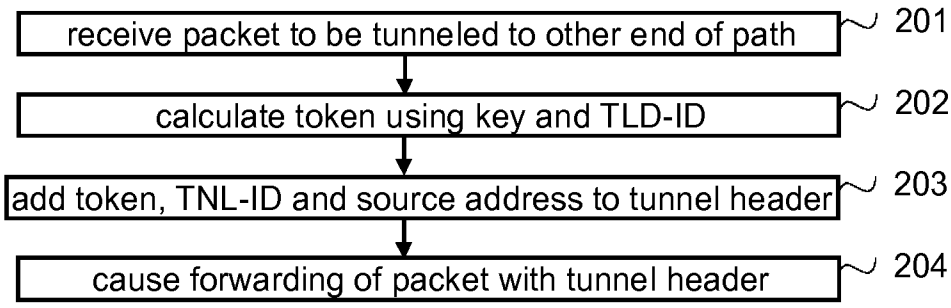

FIG.2

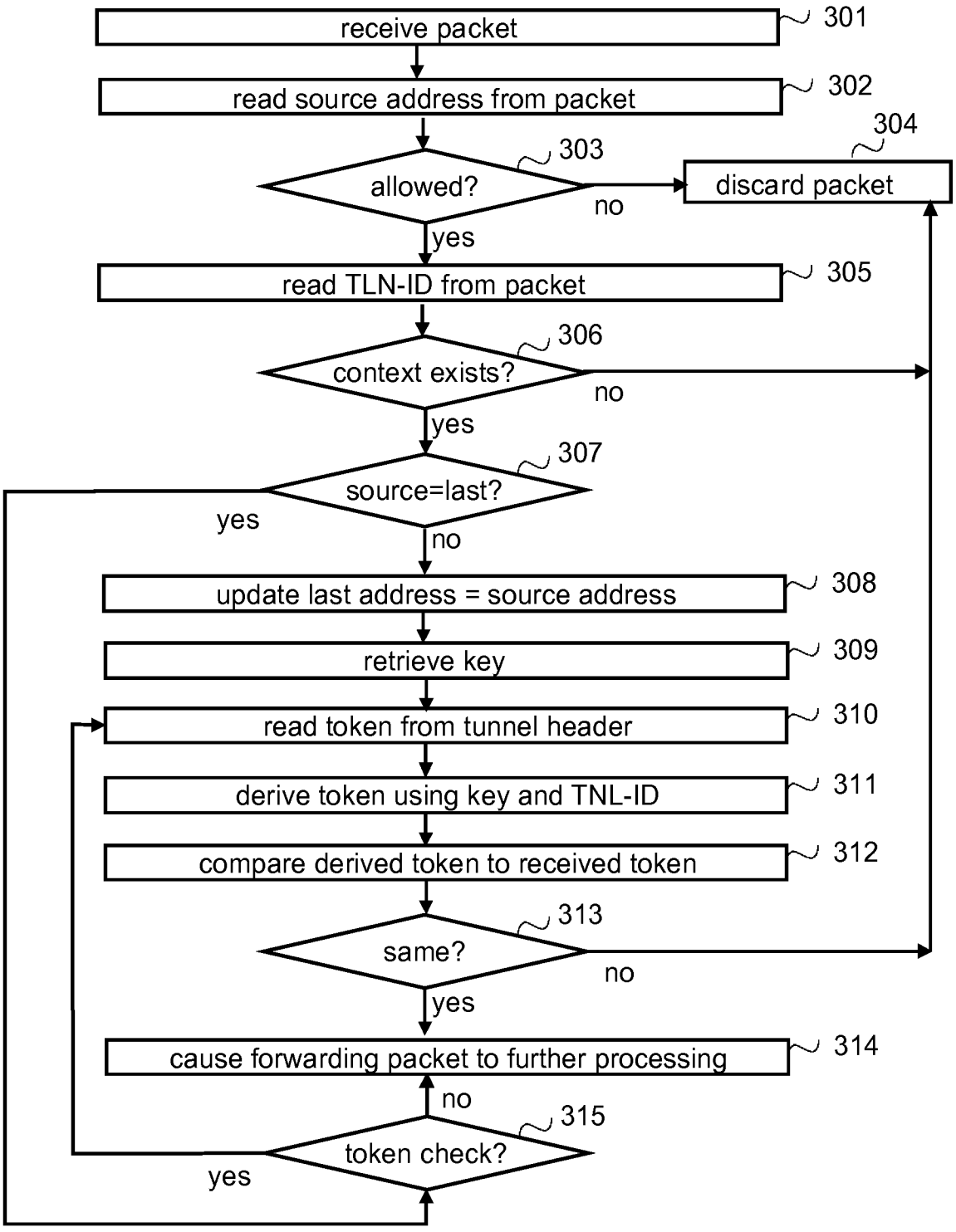
receive packet
301
read source address from packet
302
allowed?
303
no
discard packet
304
yes
read TLN-ID from packet
305
context exists?
306
no
yes
source=last?
307
yes
no
update last address = source address
308
retrieve key
309
read token from tunnel header
310
derive token using key and TNL-ID
311
compare derived token to received token
312
same?
313
no
yes
cause forwarding packet to further processing
314
no
token check?
315
yes

FIG.3

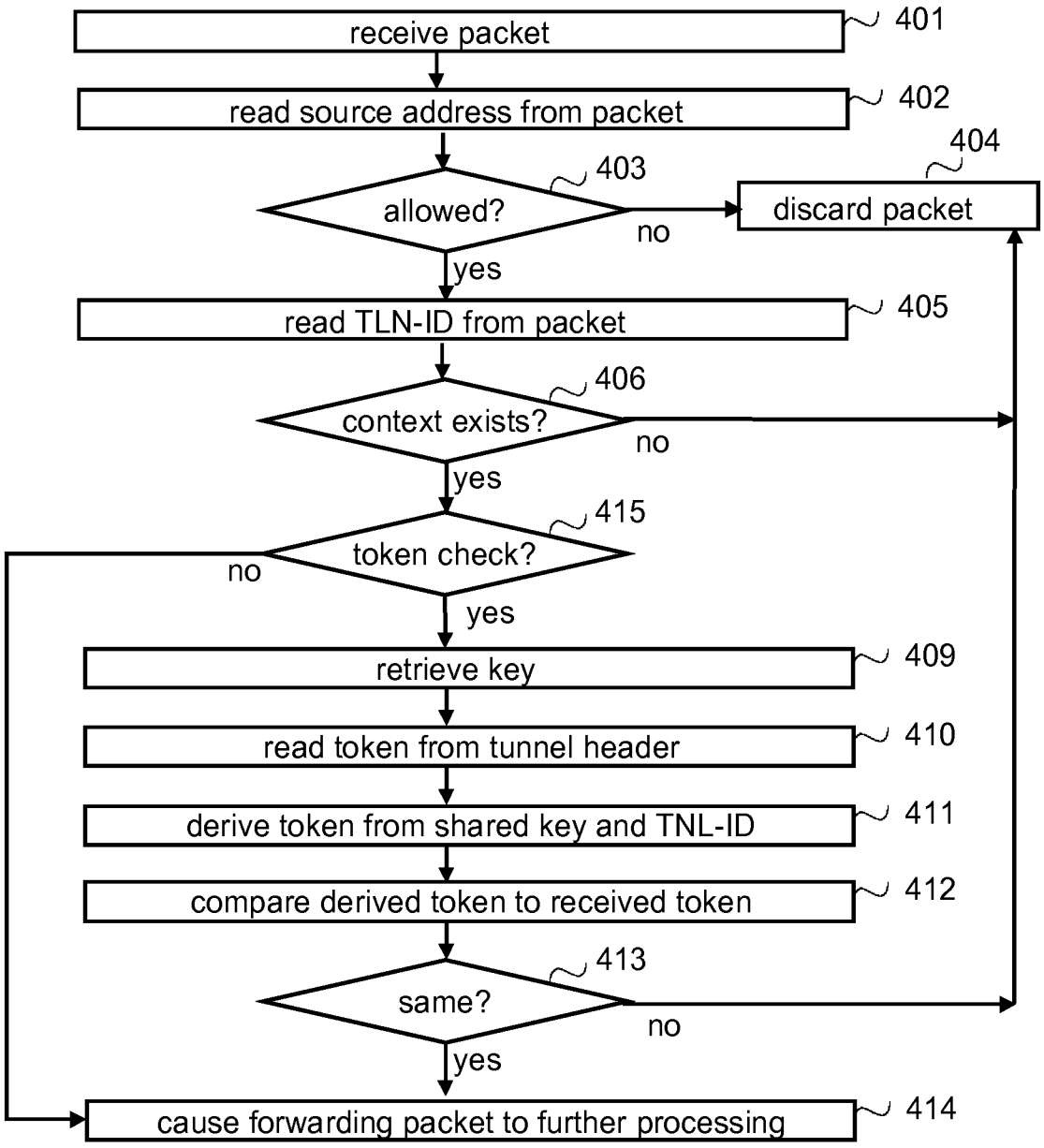
receive packet
401
read source address from packet
402
allowed?
403
no
discard packet
404
yes
read TLN-ID from packet
405
context exists?
406
no
yes
token check?
415
no
yes
retrieve key
409
read token from tunnel header
410
derive token from shared key and TNL-ID
411
compare derived token to received token
412
same?
413
no
yes
cause forwarding packet to further processing
414

FIG.4

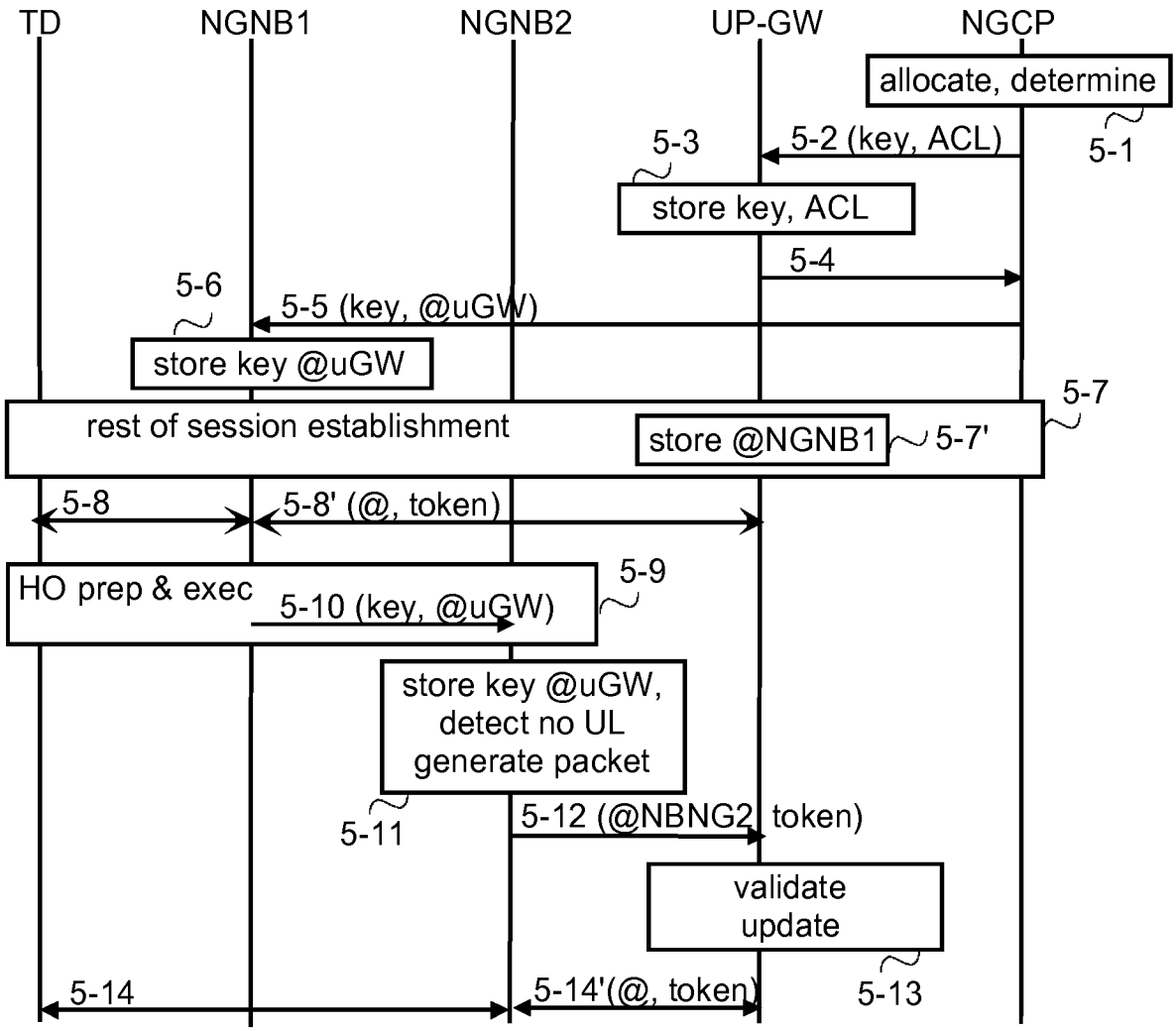

FIG.5

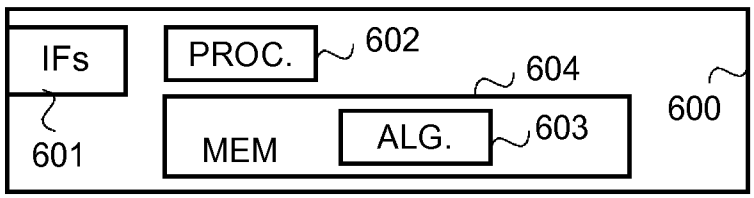

FIG.6

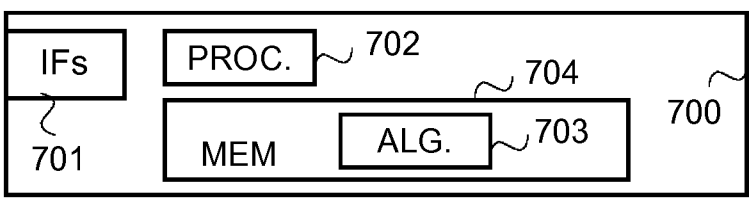

FIG.7

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/080936

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06 H04W12/08 H04L9/32
ADD. H04W36/00

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 H04W

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 2016/248686 A1 (LEE SOO BUM [US] ET AL) 25 August 2016 (2016-08-25) paragraphs [0012], [0018] - [0019], [0022], [0079], [0090], [0124], [0127], [0130] - [0132], [0160], [0162], [0184], [0190]; figures 5, 9-12	1-24
X	US 2016/344635 A1 (LEE SOO BUM [US] ET AL) 24 November 2016 (2016-11-24) paragraphs [0014], [0041], [0043], [0046], [0074] - [0075], [0089] - [0090], [0096] - [0100], [0123] - [0126], [0129], [0138], [0158]; figures 6, 8, 9	1-24

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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
3 August 2017	11/08/2017

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	Betz, Sebastian

1

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2016/080936

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2016/155896 A1 (TELEFONAKTIEBOLAGET LM ERICSSON [SE]) 6 October 2016 (2016-10-06) page 1, paragraph 4, page 5, paragraph 2, page 6, paragraph 2, page 7, paragraph 2, page 8, paragraph 6, page 11, paragraph 2	1-24
A	US 2008/233963 A1 (ALANARA SEPPO M [FI] ET AL) 25 September 2008 (2008-09-25) abstract; paragraphs [0044] - [0046], [0079] - [0080]; figure 4	1-24
A	US 2010/067489 A1 (PELLETIER GHYSLAIN [SE] ET AL) 18 March 2010 (2010-03-18) abstract; paragraphs [0007] - [0010], [0031] - [0037]; figures 4, 5, 6A, 6B	1-24

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2016/080936

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2016248686 A1	25-08-2016	TW 201644236 A	16-12-2016
		US 2016248686 A1	25-08-2016
		WO 2016175909 A2	03-11-2016
US 2016344635 A1	24-11-2016	US 2016344635 A1	24-11-2016
		WO 2016186885 A2	24-11-2016
WO 2016155896 A1	06-10-2016	US 2017104727 A1	13-04-2017
		WO 2016155896 A1	06-10-2016
US 2008233963 A1	25-09-2008	CN 101675678 A	17-03-2010
		EP 2123088 A1	25-11-2009
		US 2008233963 A1	25-09-2008
		WO 2008110996 A1	18-09-2008
US 2010067489 A1	18-03-2010	EP 2123085 A2	25-11-2009
		JP 5032598 B2	26-09-2012
		JP 2010522466 A	01-07-2010
		US 2010067489 A1	18-03-2010
		WO 2008115125 A2	25-09-2008