



(51) International Patent Classification:

G06F 21/57 (2013.01) H04L 29/06 (2006.01)

(21) International Application Number:

PCT/EP2020/052076

(22) International Filing Date:

29 January 2020 (29.01.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

19157811.1 18 February 2019 (18.02.2019) EP

(71) Applicant: **NOKIA TECHNOLOGIES OY** [FI/FI];
Karakaari 7, 02610 Espoo (FI).

(72) Inventors: **OLIVER, Ian Justin**; Hitävägen 153, 01150
Söderkulla (FI). **LIMONTA, Gabriela**; Rantaharju 10 B
42, 02230 Espoo (FI).

(74) Agent: **NOKIA EPO REPRESENTATIVES**; Nokia
Technologies Oy, Karakaari 7, 02610 Espoo (FI).

(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, ST, SV, SY, TH, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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).

Published:

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

(54) Title: SECURITY STATUS OF SECURITY SLICES

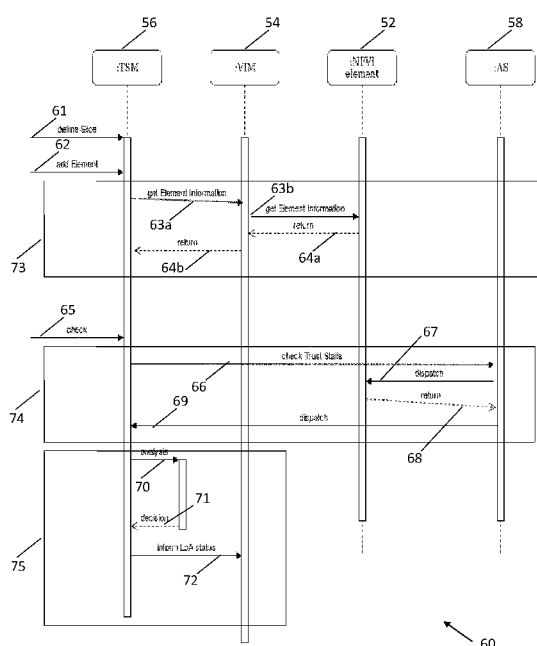


Fig. 6

(57) Abstract: An apparatus, method and computer program is described comprising: sending one or more requests to an attestation server, wherein each request requests security attributes corresponding to one of one or more network elements of a security slice of a system; receiving the requested security attributes from the attestation server; and processing the received security attributes to determine a security status of the security slice.

SECURITY STATUS OF SECURITY SLICES**Security Status****Field**

The present specification relates to a security status of a network element or a security slice comprising one or more network elements.

Background

There remains a need for alternative or improved systems for managing security status, such as levels of security assurance, in network elements security slices.

Summary

In a first aspect, this specification describes an apparatus comprising: means for sending one or more requests to an attestation server, wherein each request requests security attributes corresponding to one of one or more network elements of a security slice of a system; means for receiving the requested security attributes from the attestation server; and means for processing the received security attributes to determine a security status of the security slice. The security attributes may, for example, be properties or measurements. Some embodiments further comprise means for outputting the determined security status of the security slice.

Some embodiments comprise means for comparing the determined security status of the security slice with a required security status for the security slice.

Some embodiments comprise means for adding one or more additional network elements to the security slice. The means for adding one or more network elements to the security slice may comprise means for obtaining information relating to each of the one or more network elements to be added to the security slice. The said information may comprise at least one of availability information and/or authentication parameters and/or information regarding whether the network element(s) are suitable to be added to the slice.

Some embodiments further comprise: means for receiving, from the attestation server, security attributes (e.g. properties or measurements) of a first additional network element to be added to the security slice; means for determining an integrity level of the first additional network element based on the received security attributes of said first additional network element; and means for preventing the first additional network element from being added to the security slice in the event that said integrity level is below a required level. The integrity level may, for example, be a security status.

The means for processing the received security attributes may further comprise means for determining whether any of the network elements of the security slice fail to satisfy a defined requirement. In response to detecting a failing network element, another element
5 of the system may be informed (e.g. at least one of: the VIM, the security orchestrator and/or the SDN). Alternatively, or in addition, the identified network element may be repaired.

The apparatus may be one of a security attribute manager and a trusted slice manager.

10

A network element may comprise at least one of: a physical network element, and/or a virtualized network function, and/or a virtual machine image, and/or a virtual machine instance. Alternatively, or in addition, a network element may, for example, be a core network element, an edge device, a mobile communication device, a network function
15 virtualisation node, a virtualised network function or an Internet of Things device such as a wireless sensor. In some embodiments, a network element may comprise a top-level element, such as an NFVI element, server, edge device, IoT device, VM image, VM instance, VNF, UE (mobile device) etc. The network element may comprise a structured element comprises a number of top-level elements.

20

The requested security attributes may comprise at least one of: a measured boot capability, and/or a secure boot capability, and/or a runtime integrity measurement, and/or a virtual machine integrity level.

25

The security status of the security slice may comprise a level of assurance.

The said means may comprise at least one processor; and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the performance of the apparatus.

30

In a second aspect, this specification describes a system comprising an apparatus as described above with reference to the first aspect and further comprising: an attestation server for receiving requests for security attributes; and one or more network elements. Each network element may further comprise a trust agent for providing an interface
35 between the respective network element and said attestation server.

- 3 -

In a third aspect, this specification describes a method comprising: sending one or more requests to an attestation server, wherein each request requests security attributes corresponding to one of one or more network elements of a security slice of a system; receiving the requested security attributes from the attestation server; and processing the
5 received security attributes to determine a security status of the security slice. Some embodiments further comprise outputting the determined security status of the security slice.

The method may comprise comparing the determined security status of the security slice
10 with a required security status for the security slice.

The method may comprise adding one or more network elements to the security slice.

The method may comprise: receiving, from the attestation server, security attributes (e.g.
15 properties or measurements) of a first additional network element to be added to the security slice; determining an integrity level of the first additional network element based on the received security attributes of said first additional network element; and preventing the first additional network element from being added to the security slice in the event that said integrity level is below a required level. The integrity level may, for example, be a
20 security status.

Processing the received security attributes may further comprise determining whether any of the network elements of the security slice fail to satisfy a defined requirement.

25 In a fourth aspect, this specification describes any apparatus configured to perform any method as described with reference to the third aspect.

In a fifth aspect, this specification describes computer-readable instructions which, when executed by computing apparatus, cause the computing apparatus to perform any method
30 as described with reference to the third aspect.

In a sixth aspect, this specification describes a computer program comprising instructions for causing an apparatus to perform at least the following: send one or more requests to an attestation server, wherein each request requests security attributes corresponding to one
35 of one or more network elements of a security slice of a system; receive the requested security attributes from the attestation server; and process the received security attributes to determine a security status of the security slice.

- 4 -

In a seventh aspect, this specification describes a computer-readable medium (such as a non-transitory computer readable medium) comprising program instructions stored thereon for performing at least the following: sending one or more requests to an attestation server, wherein each request requests security attributes corresponding to one of one or more network elements of a security slice of a system; receiving the requested security attributes from the attestation server; and processing the received security attributes to determine a security status of the security slice.

In an eighth aspect, this specification describes an apparatus comprising: at least one processor; and at least one memory including computer program code which, when executed by the at least one processor, causes the apparatus to: send one or more requests to an attestation server, wherein each request requests security attributes corresponding to one of one or more network elements of a security slice of a system; receive the requested security attributes from the attestation server; and process the received security attributes to determine a security status of the security slice.

In a ninth aspect, this specification describes an apparatus comprising: a first output for sending one or more requests to an attestation server, wherein each request requests security attributes corresponding to one of one or more network elements of a security slice of a system; a first input for receiving the requested security attributes from the attestation server; and a processor for processing the received security attributes to determine a security status of the security slice. The security attributes may, for example, be properties or measurements. Some embodiments further comprise means a second output outputting the determined security status of the security slice.

Brief description of the drawings

Example embodiments will now be described, by way of non-limiting examples, with reference to the following schematic drawings, in which:

FIG. 1 is a block diagram of a system in accordance with an example embodiment;
FIG. 2 is a block diagram of a system in accordance with an example embodiment;
FIG. 3 is a block diagram of a system in accordance with an example embodiment;
FIG. 4 is a flow chart showing an algorithm in accordance with an example embodiment;
FIG. 5 is a flow chart showing an algorithm in accordance with an example embodiment;
FIG. 6 is a message sequence in accordance with an example embodiment;

- 5 -

FIGS. 7A and 7B are block diagrams of systems in accordance with an example embodiment;

FIG. 8 is a block diagram of a system in accordance with an example embodiment;

FIG. 9 is a flow chart showing an algorithm in accordance with an example embodiment;

5 FIG. 10 is a message sequence in accordance with an example embodiment;

FIG. 11 is a block diagram of components of a system in accordance with an example embodiment; and

10 FIGS. 12A and 12B show tangible media, respectively a removable memory unit and a compact disc (CD) storing computer-readable code which when run by a computer perform operations according to example embodiments.

Detailed description

In the description and drawings, like reference numerals refer to like elements throughout.

15

Computing systems, including (but not limited to) distributed or cloud computing systems, may include a wide range of hardware or other elements connected thereto. Such modules may, for example, be provided to run virtual workloads, base stations of a telecommunication network, edge devices etc. At least some of those elements may be
20 arranged in slices, as described further below.

The example embodiments disclose apparatus and methods for creating security slices for managing levels of security assurance in network slices. An attestation server may be utilized for creating interfacing with security slices (such as trust slices). Such security
25 slices are independent of network slices and may, for example, be orthogonal to network function virtualisation (NFV) slices. Some of the example uses of security slicing may include guaranteeing different levels of security assurances for critical workload deployment, such as:

- Attestation;
- 30 - Underlying hardware or software integrity (including run-time integrity);
- Virtual network function (VNF) integrity checks over life-cycle (including VNF image load, VNF instance start, suspend, migrate, and/or destroy);
- Supply-chain integrity (for example: only allowing known, trusted VNF supply chain, notarisation technologies);
- 35 - Additional monitoring and analysis (artificial intelligence or machine learning, proactive protections) features (for example: monitoring, firewalls, network encryption on demand);

- 6 -

- Software defined networks (SDN) and Management and Orchestration (MANO) software integration; and
- Extension outside Core NFV Infrastructure, such as Edge Slicing and IoT slicing

5 As defined in ETSI NFV SEC007 (2018), the overall attestation scope depends also on the exact use case and, most importantly, on the agreed Level-of-Assurance (LoA), described in SEC007. In particular, the LoAs define the sets of systems and components to be considered during attestation procedures and, thus, facilitate the determination of the overall attestation scope. An overview of the defined LoAs in relation to the attestation
10 scope is depicted in table 1.

LoA Level	LoA defined set of attested Systems and Components	Type	Affected Attestation Sub-Scope(s)	Attestation
0	all components	None	None	None
1	Hardware and Virtualization Platform	Loadtime	Platform + Hypervisor	Local
2	Hardware and Virtualization Platform	Loadtime	Platform + Hypervisor	Remote
3	VNF Software Packages	Loadtime	Virtual Machine + Application	Local
4	VNF Software Packages	Loadtime	Virtual Machine + Application	Remote
5a	Hardware and Virtualization Platform (Infrastructure Network)	Runtime	Platform + Hypervisor	Remote
5b	Virtualization Platform and VNF Software Packages	Runtime	Hypervisor + Virtual Machine + Application	Remote

Table 1: Level of Assurance to Attestation Scope Mapping

- 7 -

In some example embodiments, integrity levels (sometimes referred to herein as security status) may be determined for each network element of a system. Each network element may belong to one or more security slices. Details of how integrity levels/security statuses are used in the example embodiments are discussed later in the document. The integrity levels for security slices define properties or measurements that can be taken in order to satisfy a specific level of assurance (LoA). A level of assurance may be guaranteed based on the integrity level/security status. An example mapping of integrity level of security slices with respect to the level of assurance guaranteed by the security slice is illustrated in Table 2. For example, in order to guarantee a particular LoA, the devices must pass all the integrity checks defined in the mapping of Table 2.

LoA Level	Integrity level requirements
0	None
1	Platform secure boot level 1 OR measured boot level 3
2	Platform measured boot level 4
3	Virtual machine Image level 1 AND Virtual machine Instance level 1
4	Virtual machine Image level 2 AND Virtual machine Instance level 2
5a	Virtual machine Image level 3 AND Virtual machine Instance level 3
5b	Virtual machine Image level 3 AND Platform Runtime level 2

Table 2 - Mapping between LoAs and integrity levels

The LoAs define the desired level of trustworthiness or integrity of the platform. They also define what subdomains need to be attested to guarantee that level of assurance. On the other hand, the trusted slice integrity levels define concrete measurements that can be considered using attestation in order to guarantee the security status of the attestation scope defined by the LoA.

For better understanding of Table, 2, the integrity levels of various platforms are explained below. There may be two level hierarchies for platform integrity measurements depending on the capabilities of the platform. Many variants to the arrangements described herein are possible.

Platform with Measured Boot capabilities

For a platform with Measured Boot capabilities, there may be five integrity levels that can be achieved:

- Level 0: No integrity checks

- 8 -

- Level 1: Verification of the credentials for the RTR (TPM). This includes a validity check of the Endorsement Certificate and proof of the ownership of the trusted platform module (TPM) keys (e.g. endorsement and attestation keys).

- Level 2: Verification of the integrity of the core root of trust measurements

(CRTM) of the system, in addition to the checks of level 1. This includes ACM and BIOS code modules.

- Level 3: Verification of the integrity of the static root of trust (SRTM) of the system, in addition to the checks of level 2. This includes Host Platform Configuration, Optional ROMs, Optional ROM configuration and data, IPL code, IPL configuration and data and Launch Control Policies, if they exist.

- Level 4: Verification of the integrity of the dynamic root of trust (DRTM) of the system, in addition to the checks of level 3. This includes the kernel, OS and virtualization platform.

The checks for levels 1 to 3 can be done locally and/or remotely and level 4 can only be done remotely. Additionally, these integrity levels include both the platform and the hypervisor sub-scopes.

Platform with Secure Boot capabilities

For a platform with Secure Boot capabilities, there may be two integrity levels that can be achieved:

- Level 0: No integrity checks.

- Level 1: Verification of the firmware, bootloader and virtualization platform (OS/kernel/hypervisor), based on digital signatures or certificates.

Level 1 for a platform with Secure Boot capabilities can only be achieved locally. These checks are performed during boot time and if any of the checks fail, the platform will not start up. For a platform with Secure Boot capabilities, there is only one integrity level which includes the all the checks of the platform and hypervisor sub-scopes.

Platform that is capable of performing runtime integrity measurements

For a platform that is capable of performing runtime integrity measurements (e.g. by using a kernel module such as Linux IMA), there are four integrity levels that may be achieved:

- Level 0: No integrity checks.

- 9 -

- Level 1: Local verification of the integrity of the static filesystem (this can include configuration files, binaries, etc.) by preventing access to files that have changed, in addition to the checks from either measured boot level 4 OR secure boot level 1.
- Level 2: Remote verification of the integrity of the filesystem, in addition to the checks of level 1, in addition to the checks from level 1.
- Level 3: Verification of the deployed infrastructure network for the VNF, in addition to the checks of level 2.

Virtual machine sub-scope

For the virtual machine sub-scope, there are different integrity levels that can be achieved, depending on whether the checks are for a virtual machine image or a virtual machine instance.

Virtual machine image

For a virtual machine image, there may be three integrity levels that can be achieved:

- Level 0: No integrity checks.
- Level 1: Local verification of the signature of the image during onboarding time.
- Level 2: Remote verification of the signature of the image during onboarding time, in addition to the checks for level 1.

Virtual machine instance

For a virtual machine instance, there may be four integrity levels that can be achieved:

- Level 0: No integrity checks.
- Level 1: Local verification of the image to be launched at instantiation time, by using digital signatures.
- Level 2: remote verification of the image to be launched at instantiation time, in addition to the checks from level 1.
- Level 3: remote verification of the integrity of the virtual machine instance during runtime and throughout all the operations that the instance may go through during its lifecycle (including shutdown, suspend and migration), in addition to the checks for level 2.

Note that the levels of integrity defined for a platform can be achieved by a virtual machine instance, given that it has the capabilities of a virtual trusted platform module (vTPM).

A typical NFV system consists of any number of NFVI elements (typically servers of some form) managed by management and orchestration (MANO). Using the concept of slicing, these physical machines can be partitioned into logical blocks and purposed for specific purposes, for example, a specific operator provisioning etc. The mechanisms for slicing based on network partitioning are well known.

FIG. 1 is a block diagram of a system, indicated generally by the reference numeral 1, in accordance with an example embodiment. System 1 comprises a plurality of network elements (NE), namely NE1, NE2, NE3, and NE4. The network elements are assigned to one or more network slices and/or one or more security slices. As discussed above, security slices (SS) and network slices (NS) are created independently. Further, a network element may belong to one or more security slices at the same time. However, in many implementations a network element may only belong to one network slice at a time. In the system 1, NE1 and NE2 belong to a first network slice (NS1), while NE3 and NE4 belong to a second network slice (NS2). Also, NE1, NE2 and NE4 are assigned to a first security slice (SS1), and NE3 and NE4 are assigned to a second security slice (SS2).

Network elements may be assigned to network slices based on their uses, customers, etc. For example, NS1 may specifically contain network elements relating to Internet-of-Things (IoT) devices, and NS2 may specifically contain network elements relating to medical devices.

Network elements may be assigned to security slices regardless of their functionality and regardless of their corresponding network slicing. A security status of SS1 and SS2 may be indicated by Level of Assurance (LoA) level, which is determined based on security attributes of the network elements assigned to the security slices. For example, security status of SS1 may be dependent upon the security attributes of NE1, NE2 and NE4, and security status of SS2 may be dependent upon the security attributes of NE3 and NE4. The security status of the security slices may be indicated by LoA level. For example, if SS1 has LoA level 2 (explained in Tables 1 and 2 above) and SS2 has LoA level 1, this may mean that integrity levels of NE1, NE2, and NE4 are collectively higher than collective integrity levels of NE3 and NE4.

Network elements may be chosen for specific workloads based on the network slicing as well as security slicing. More particularly, for a specific workload, a network slice

- 11 -

requirement and a security slice requirement may be defined, and one or more network elements may be selected for the specific workload accordingly.

For example, assume that there is a requirement for a first workload for network elements relating to medical devices and the LoA level of the network elements must be at least 2. Referring to FIG. 1, NS2 is a network slice containing network elements NE3 and NE4 relating to medical devices, and SS1 is a security slice with LoA level 2. A workload manager may then determine which network elements meet the both the network slice and security slice requirements of the first workload. It may be determined that only NE4 is suitable for the first workload, as NE4 relates to medical devices, and also belongs to SS1 which has LoA level 2.

In an alternative example, assume that there is a requirement for a second workload for network elements relating to IoT devices and LoA level of the network elements must be at least 2. In this case, only NE1 and NE2 are suitable for the second workload, as they are in NS1 relating to IoT devices and they belong to SS1 which has LoA level 2.

FIG. 2 is a block diagram of a system, indicated generally by the reference numeral 10, in accordance with an example embodiment. The system 10 comprises network functions virtualization infrastructure (NFVI) 12, a virtual infrastructure manager (VIM) 14, a trusted slice manager (TSM) 16 and an attestation server (AS) 18.

The network functions virtualization infrastructure (NFVI) 12 comprises multiple network elements of a system. The network elements may take many forms, such as physical network elements (such as servers), virtualised network function nodes, a virtual machine image, a virtual machine instance, an edge node, an Internet-of-things (IoT) enabled device, a communications module etc. The network elements of the NFVI 12 may be grouped into one or more slices. One or more of those slices may be security slices, as discussed above with reference to FIG. 1.

The trusted slice manager (TSM) 16 is responsible for managing aspects of security slices. The trusted slice manager 16 may communicate with the virtual infrastructure manager 14 and the attestation server 18 to obtain security attributes corresponding to network elements of a security slice. The attestation server 18 provides such information to the trusted slice manager 16, as discussed in detail below.

- 12 -

On the basis of the security attributes corresponding to network elements of a security slice, the trusted slice manager 16 can determine a security status, e.g. a Level of Assurance (LoA), for a particular security slice that is formed from network elements of the NFVI 12. Algorithms for determining a security status of a security slice are discussed further below.

FIG. 3 is a block diagram of a system, indicated generally by the reference numeral 20, in accordance with an example embodiment.

The system 20 comprises a group of network elements 21. As shown schematically in FIG. 3, the network elements 21 may include one or more NFVI elements or physical machines 22, one or more virtual machine images 23 and/or one or more virtual machine instances 24. The system 20 also comprises a number of security slices 25. As shown schematically in FIG. 3, the security slice 25 has a security attribute 26. The security attribute 26 may include a number of parts, such as a trust Level of Assurance 27 and one or more other parts 28.

An example network element 21 may belong to any number of security slices 25 (including zero). Similarly, a security slice may include any number of network elements 21 (including zero).

FIG. 4 is a flow chart showing an algorithm, indicated generally by the reference numeral 30, in accordance with an example embodiment. The algorithm 30 shows an example arrangement for building a security slice.

The algorithm 30 starts at operation 32, where a security slice is defined. For example, the operation 32 may involve providing a name for the security slice and providing other basic information. Thus, the operation 32 may result in a security slice 25 being instantiated.

At operation 34, one or more elements are added to the slice. For example, instances of the element 21 may be added to an instance of the security slice 25 described above.

At operation 36, information regarding the added element(s) is obtained. As described further below, the information obtained in the operation 36 may be used for functions such as determining security attributes of the added element(s).

- 13 -

As described above, one aspect of a security slice is to define a Level of Assurance (LoA) that the elements in that slice must be able to guarantee. Indeed, there may be provided means for determining an integrity level or security status of the network element referred to in operation 34 based on the received security attributes of said first additional network
5 element. In the event that the integrity level or security status is below a required level, then the network element may be prevented from being added to the security slice.

FIG. 5 is a flow chart showing an algorithm, indicated generally by the reference numeral 40, in accordance with an example embodiment.

10 The algorithm 40 starts at operation 42 where a trust status or security status of one or more elements of a relevant security slice is obtained. Then, at operation 44, a level of Assurance (LoA) is determined for the security slice. An example implementation of the algorithm 40 is described further below.

15 Optionally, the algorithm 40 may include an operation 46 in which the security status determined in operation 44 is compared with a required security status for the security slice. If a security status is below a required security status, then action may be taken (as discussed further below).

20 FIG. 6 is a message sequence, indicated generally by the reference numeral 60, in accordance with an example embodiment. The message sequence 60 shows an example sequence of messages between a network functions virtualization infrastructure (NFVI) element 52, a virtual infrastructure manager (VIM) 54, a trusted slice manager (TSM) 56
25 and an attestation server (AS) 58. The NFVI element 52 may be an element of the network function virtualization infrastructure (NFVI) 12 described above. Similarly, the virtual infrastructure manager (VIM) 54, the trusted slice manager (TSM) 56 and the attestation server (AS) 58 may be similar to the VIM 14, the TSM 16 and the attestation server 18 described above.

30 The message sequence 60 starts with a define slice instruction 61 received at the trusted slice manager 56. This operation may define a slice (e.g. a security slice) in terms of identity and name. Other characteristics such as encryption keys, authentication, access etc. may also be defined. A required level of assurance (LoA) may be defined for the slice.
35 Other aspects, such as LoA failure handling may also be defined.

- 14 -

As shown in FIG. 6, an add element instruction 62 is received at the trusted slice manager 56 requesting the addition of a network element to the relevant security slice.

5 In response to the instruction 62, the VIM 54 may be contacted to obtain element information to decide upon availability, authentication parameters, suitability for inclusion in that slice etc. Elements are not necessarily restricted to traditional NFVI elements such as servers, but may include VM/VNF images, their potential instances, Edge, IoT and UE (user equipment) devices. Inclusion into a security slice may also involve other NFV MANO components such as the Orchestrator and VNFM in making these
10 decisions. Interaction with the OSS/BSS layer and other MANO components is also permissible and possible.

As shown in FIG. 6, in response to receiving the add element instruction 62, a get element information message 63a may be sent from the TSM 56 to the VIM 54 and a get message
15 information message 63b may be sent from the VIM 54 to the NFVI element 52. In response to the message 63b, the request information is provided by to the NFVI element to the VIM 54 (in return message 64a) and to the TSM 56 (in a return message 64b).

It should be noted that the messages 63 and 64 are provided by way of example only.
20 Alternative arrangements to enable the TSM 56 to obtain the requirement element information from the NFVI element 52 could be provided.

As shown in FIG. 6, a check instruction 65 is received at the trusted slice manager 56. In response to the check instruction 65, checks are performed on the relevant network
25 elements and subsequent decisions made about the level of assurance achieved. The TSM 56 sends a request 66 to the attestation server (AS) 58 for the security attributes corresponding to the relevant network element. The AS 58 is responsible for doing all the integrity checks required for the LoA of the trusted slice (gathering measurements and determining trust status). The AS 58 may also be only responsible for gathering all the
30 measurements and the TSM 56 may have the final decision on the trust status of the element after analysing the information it has about its integrity levels.

In response to the request 66, the AS 58 may send a message 67 to the relevant element 52 for the security attributes, which are received in the reply message 68 and then returned to
35 the TSM 56 in a message 69. Thus, the message 69 provides the security attributes requested in the message 66 to the TSM 56. In the example message sequence 60, the AS

- 15 -

58 checks the elements either directly or by proxy via the VIM/VNFM, and the subsequently makes a decision on the trustworthiness and trust status of that element.

The TSM 56 processes the received security attributes to determine a security status of the security slice (as indicated by the analysis step 70 and decision step 71). The TSM 56 may perform any amount of analysis and decision-making either at a point in time or also including historical information as required to determine whether an element or whole or partial slice has achieved the required level of assurance.

The TSM 56 may send a message 72 to the VIM 54 indicating the security status of the security slice. The decision to inform either the VIM 54 or other components is optional in some embodiments.

If an element within a security slice fails the LoA checks (e.g. is below a required level) then it may be marked accordingly. Various options are available for handling this situation, including but not limited to:

- Ignore the relevant element
- Inform the VIM
- Inform the Security Orchestrator
- Inform the SDN
- Perform repair

The exact choices made (there may be more than one, unless failure is ignored) may depend upon the system configuration. For example, LoA failure might trigger an alert to the/a security orchestrator, which in turn may cause the VIM to migrate workload away from that element.

FIG. 7A is a block diagram of a system, indicated generally by the reference numeral 80A, in accordance with an example embodiment. The system 80A comprises a network function virtualization infrastructure (NFVI) 81. The NFVI 81 comprises a plurality of network elements 82. As discussed earlier, the plurality of network elements 82 may take many forms. FIG. 7B is a block diagram of a system, indicated generally by the reference numeral 80B, in accordance with an example embodiment. The system 80B shows the different types of exemplary network elements 82. For example, a network element 82 may be one or more of a virtualized network function 83, core network element 84, edge device 85, mobile device 86, and/or Internet of Things (IoT) device 87 (such as a wireless sensor). Clearly more or fewer elements may be provided within NFVI 81.

FIG. 8 is a block diagram of a system, indicated generally by the reference numeral 100, in accordance with an example embodiment. The system 100 comprises an attestation server 102, a first element 106, a second element 107, and a third element 108. The system 100 additionally comprises an attestation database 103, attestation tools 104, and an attestation user interface 105 in communication with the attestation server 102. The first, second and third elements 106 to 108 are provided by way of example. Clearly more (or fewer) elements may be provided within a system. The elements 106 to 108 may be elements of a cloud computing system (e.g. a trusted cloud). The first to third elements 106 to 108 may be hardware modules and, as shown in FIG. 8, are in communication with the attestation server 102 via a network bus. The elements 106 to 108 may form part of a distributed or cloud computing system. The attestation server 102 may, for example, be tasked with monitoring a trust status of the system 100 and the elements within the system.

The elements 106 to 108 are examples on the network elements discussed above with respect to FIGS 1 to 7.

In order to monitor aspects of the system 100, the attestation server 102 may communicate with each of the elements 106 to 108 to obtain measurements. For example, a trusted platform module (TPM) may be provided at each element to generate a cryptographic hash that summarises the hardware and software configuration of the relevant module. A set of platform configuration registers (PCRs) may be provided which store cryptographic hashes of measurements of the relevant components. A hash may, for example, be obtained from a TPM by a mechanism known as quoting. A quote may be generated for a set of PCRs, with the TPM generating a hash from the contents of the set of PCRs and signing them with an attestation key (AK) unique to the respective TPM (e.g. with a private key of an attestation key pair).

The attestation server 102 may offer a query application programming interface (API) that can be used, for example, by command line tools. The attestation user interface 105 of the system 100 (e.g. a web application) may enable a user to interact with the attestation server 102 (e.g. to enable viewing of a trust status of a cloud or to request measurements of one or more elements of the system 100).

The first element 106 comprises a trust agent 110a, a trusted platform module (TPM) software stack 110b and a trusted platform module 110c. Similarly, the second element 107

- 17 -

comprises a trust agent 112a, a trusted platform module (TPM) software stack 112b and a trusted platform module 112c. The third element 108 comprises a trust agent 114a, a trusted platform module (TPM) software stack 114b and a trusted platform module 114c.

- 5 The trust agents 112a, 112a and 114a at each element of the system 100 provide an interface between the respective element and the attestation server 102.

As indicated above, each of the elements 106 to 108 of the system may have a trusted platform module associated therewith. The trusted platform module may form part of the
10 respective element. The trusted platform module may be implemented as a device of the respective element, but may alternatively be distributed. In essence, the trusted platform module is a specification of behaviour implemented by the relevant element.

The trusted platform modules (TPMs) 110c, 112c and 114c may store cryptographic keys,
15 certification and confidential data. For example, two unique key-pairs may be stored at (or be available to) each TPM: an endorsement key pair (EK) and an attestation key pair (AK). A set of platform configuration registers (PCRs) may be provided to store measurements, in the form of hashes, of hardware or software components of the relevant machine (e.g. the element within which the TPM is installed). A TPM may be asked to provide a “quote”
20 for a defined set of PCRs (e.g. a hash over the stored value of the defined PCRs). The TPM may then return the quote for the requested PCRs, a cryptographic signature of that quote (signed by the attestation key, e.g. the private key of the attestation key pair) and possibly other information, such as a timestamp and a reboot count.

25 An attestation policy is a set of expected values for different measurements that can be taken from a machine (such as one or more of the elements 106 to 109 described above). If a machine has a TPM, a policy can be mapped between a PCR number and an expected value. An administrator can define policies in order for a machine to be considered in a trusted state. When attestation is carried out, the measurements can be checked against
30 the expected values in the policy. The expected values are reference values that can be used to define a “correct” state of a system and/or can be used to detect changes in a system. When quoting, if a machine stops following a certain policy, this may indicate that what was measured by the policy has changed in the system.

35 The attestation server 102 may be responsible for obtaining quotes from the elements 106 to 108. For example, the attestation server 102 may be responsible for attesting the devices and checking the status of the relevant system (e.g. the system 100). During an attestation

- 18 -

process, the attestation server 102 may compare values obtained by quoting an element to a defined attestation policy for the relevant element(s). Then, if measurements from an element no longer satisfy the relevant policy/policies, an action may be initiated (e.g. generating system alerts for administrators).

5

FIG. 9 is a flow chart showing an algorithm, indicated generally by the reference numeral 120, in accordance with an example embodiment. The algorithm 120 may be implemented as any one of the elements 106 to 108 of the system 100.

The algorithm 120 starts at operation 122, where a request is received from a first module
10 (such as the attestation server 102) at a request receiving means of one of the elements of the system (such as one of the elements 106 to 108 of the system 100 described above). As described further below, the request may include a command, a nonce (to prevent replay attacks) and details of a cryptographic key for use in responding to the request. In addition, cryptographic structures may be provided by the relevant transport layer (e.g.
15 secure sockets layer (SSL) or transport layer security (TLS)).

At operation 124, a response to said request is generated at a response generating means of the respective element of the system 100. The response may be generated at a trust agent of the respective element (such as one of the trust agents 110a, 112a and 114a
20 described above). The response may include one or more of the following (depending on the received command): an identity of said element; a cryptographic hash of data representing configurations of said element; and capabilities relating to said element of the computing system.

25 At operation 126, the response is provided to the first module (such as the attestation server 102) in response to said request. As described further below, the response includes the nonce (as provided in the request) and is signed using the cryptographic key (as identified in the request).

30 Thus, in the system 100, the relevant trust agent (110a, 112a, 114a) may receive the request from the attestation server 102 and return the response to the attestation server. The relevant trust agent may, at least in part, generate the response to the request. Thus, the trust agent may be one or more of: the means for receiving the request (operation 122 discussed above); the means for generating the response (operation 124 discussed above);
35 and the means for providing the response (operation 126 discussed above).

- 19 -

FIG. 10 shows a message sequence, indicated generally by the reference numeral 130, in accordance with an example embodiment. The message sequence 130 is an example implementation of the algorithm 120 described above.

- 5 The message sequence 130 is implemented between a trust agent 132 (such as one of the trust agents 110a, 112a and 114a described above) and an attestation server 134 (such as the server 102 described above).

A request 136 is received at the trust agent 132 from the attestation server 134,
10 implementing the operation 122. The request consists of a command (such as the `get_identity`, `get_quote` and `get_capabilities` commands discussed further below) and possible additional data, such as a nonce and a cryptographic key. Of course, other commands could also be implemented in example embodiments.

The trust agent 132 processes the request 136 and runs any commands on the system
15 needed for gathering the requested information (as indicated by the reference numeral 137).

Finally, the trust agent sends a response 138 to the attestation server with the requested information, implementing the operation 126. Details of example responses 138 are
20 provided below.

As indicated above, the request 132 may include one or more of: `get_identity`; `get_quote`; and `get_capabilities` commands.

25 A `get_identity` command may request the identity of the element that the trust agent 132 is running on (e.g. the identity of the relevant element 106 to 108). In the case of a device with a trusted platform module (TPM), the identity may take the form of public keys of the trusted platform module (e.g. public keys of endorsement key (EK) and attestation key (AK) pairs). Alternatively, or additionally, the identity may include metadata that can be
30 used to identify the relevant machine, but may not be permanent identities (e.g. an IP address, MAC address, system information OpenStack ID etc.). Other implementations (e.g. non-TPM based implementations) are possible. For example, a single key may be provided. In some hardware security modules, for example, a single key, sometimes called an attestation key, may be provided.

35

A `get_quote` command may request the results of quoting or measuring an element on which the trust agent 132 is running, according to some policy indicated by the attestation

- 20 -

server 134. The quote may take the form of a cryptographic hash of data representing configurations of said element and may be generated by a trusted platform module. In one embodiment, a cryptographic hash of data representing configurations of an element is a cryptographic hash of data representing hardware, firmware and/or software configurations of said element (as stored, for example, in one or more platform confirmation registers).

A get_capabilities command may request information about the capabilities of the device, such as the trusted platform module, TPM. A response to a get_capabilities command may identify measurements (or other data) that can be provided to the attestation server. Thus, the capabilities may be used to decide what kind of measurements can be obtained by the attestation server 134 from the respective element. The capabilities information may also be used to identify properties of the TPM, such as the manufacturer or the installed firmware version.

The response 138 may include one or more of the following fields:

- Type. The type field may identify a type of the relevant device. The type field may, for example, be used by the attestation server 134 to determine what fields and/or data to expect in the response;
- Nonce (e.g. the nonce included in the request). Thus, the response can be matched with the appropriate request;
- Command (e.g. the name of the command, as provided in the request, such as 'get_identity', 'get_quote', or 'get_capabilities');
- timestamp_start (e.g. indicating when the request was received);
- timestamp_end (e.g. indicating when the trust agent completed processing the request); and
- signature (e.g. the signature of the response object, using the key indicated in the parameters of the request).

Of course, other fields may be provided instead of, or in addition to, some or all of the fields described above.

By way of example, in the case of a device with a TPM 2.0, a response to a get_identity request may take the following form:

```
{  
  'type': 'TPM2',  
  'nonce': 1234,
```

- 21 -

```

'command': 'get_identity',
'timestamp_start': '2018-06-07 10:08:20.789000+03:00',
'timestamp_end': '2018-06-07 10:08:22.856000+03:00',
'identity': {
5  'ek': '-----BEGIN PUBLIC KEY----- (...) -----END PUBLIC KEY-----',
  'ak': '-----BEGIN PUBLIC KEY----- (...) -----END PUBLIC KEY-----',
  },
  'extra_data': {
    'uname': 'Linux localhost.localdomain 4.16.7-200.local.fc27.x86_64 #1 SMP Fri May 4
10 00:20:26 EEST 2018 x86_64 x86_64 x86_64 GNU/Linux',
    ...
  },
  'signature': '...'
}

```

15

For completeness, FIG. 11 is a schematic diagram of components of one or more of the example embodiments described previously, which hereafter are referred to generically as processing systems 300. A processing system 300 may have a processor 302, a memory 304 closely coupled to the processor and comprised of a RAM 314 and ROM 312, and, optionally, user input 310 and a display 318. The processing system 300 may comprise one or more network/apparatus interfaces 308 for connection to a network/apparatus, e.g. a modem which may be wired or wireless. Interface 308 may also operate as a connection to other apparatus such as device/apparatus which is not network side apparatus. Thus, direct connection between devices/apparatus without network participation is possible.

25

The processor 302 is connected to each of the other components in order to control operation thereof.

30

The memory 304 may comprise a non-volatile memory, such as a hard disk drive (HDD) or a solid-state drive (SSD). The ROM 312 of the memory 314 stores, amongst other things, an operating system 315 and may store software applications 316. The RAM 314 of the memory 304 is used by the processor 302 for the temporary storage of data. The operating system 315 may contain code which, when executed by the processor implements aspects of the algorithms 30, 40 and 120 or the message sequence 60 described above. Note that in the case of small device/apparatus the memory can be most

35

- 22 -

suitable for small size usage i.e. not always hard disk drive (HDD) or solid-state drive (SSD) is used.

The processor 302 may take any suitable form. For instance, it may be a microcontroller, a plurality of microcontrollers, a processor, or a plurality of processors.

The processing system 300 may be a standalone computer, a server, a console, or a network thereof. The processing system 300 and needed structural parts may be all inside device/apparatus such as IoT device/apparatus i.e. embedded to very small size

In some example embodiments, the processing system 300 may also be associated with external software applications. These may be applications stored on a remote server device/apparatus and may run partly or exclusively on the remote server device/apparatus. These applications may be termed cloud-hosted applications. The processing system 300 may be in communication with the remote server device/apparatus in order to utilize the software application stored there.

FIGS. 12A and 12B show tangible media, respectively a removable memory unit 365 and a compact disc (CD) 368, storing computer-readable code which when run by a computer may perform methods according to example embodiments described above. The removable memory unit 365 may be a memory stick, e.g. a USB memory stick, having internal memory 366 storing the computer-readable code. The memory 366 may be accessed by a computer system via a connector 367. The CD 368 may be a CD-ROM or a DVD or similar. Other forms of tangible storage media may be used. Tangible media can be any device/apparatus capable of storing data/information which data/information can be exchanged between devices/apparatus/network.

Embodiments of the present invention may be implemented in software, hardware, application logic or a combination of software, hardware and application logic. The software, application logic and/or hardware may reside on memory, or any computer media. In an example embodiment, the application logic, software or an instruction set is maintained on any one of various conventional computer-readable media. In the context of this document, a “memory” or “computer-readable medium” may be any non-transitory media or means that can contain, store, communicate, propagate or transport the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer.

- 23 -

Reference to, where relevant, “computer-readable storage medium”, “computer program product”, “tangibly embodied computer program” etc., or a “processor” or “processing circuitry” etc. should be understood to encompass not only computers having differing architectures such as single/multi-processor architectures and sequencers/parallel architectures, but also specialised circuits such as field programmable gate arrays FPGA, application specific circuits ASIC, signal processing devices/apparatus and other devices/apparatus. References to computer program, instructions, code etc. should be understood to express software for a programmable processor firmware such as the programmable content of a hardware device/apparatus as instructions for a processor or configured or configuration settings for a fixed function device/apparatus, gate array, programmable logic device/apparatus, etc.

As used in this application, the term “circuitry” refers to all of the following: (a) hardware-only circuit implementations (such as implementations in only analogue and/or digital circuitry) and (b) to combinations of circuits and software (and/or firmware), such as (as applicable): (i) to a combination of processor(s) or (ii) to portions of processor(s)/software (including digital signal processor(s)), software, and memory(ies) that work together to cause an apparatus, such as a server, to perform various functions) and (c) to 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.

If desired, the different functions discussed herein may be performed in a different order and/or concurrently with each other. Furthermore, if desired, one or more of the above-described functions may be optional or may be combined. Similarly, it will also be appreciated that the flow charts and message sequences of Figures 4 to 6 and 9 are examples only and that various operations depicted therein may be omitted, reordered and/or combined.

It will be appreciated that the above described example embodiments are purely illustrative and are not limiting on the scope of the invention. Other variations and modifications will be apparent to persons skilled in the art upon reading the present specification.

Moreover, the disclosure of the present application should be understood to include any novel features or any novel combination of features either explicitly or implicitly disclosed herein or any generalization thereof and during the prosecution of the present application

- 24 -

or of any application derived therefrom, new claims may be formulated to cover any such features and/or combination of such features.

- 25 -

Claims

1. An apparatus comprising:
means for sending one or more requests to an attestation server, wherein each
5 request requests security attributes corresponding to one of one or more network
elements of a security slice of a system;
means for receiving the requested security attributes from the attestation server;
and
means for processing the received security attributes to determine a security status
10 of the security slice.
2. An apparatus as claimed in claim 1, further comprising means for outputting the
determined security status of the security slice.
- 15 3. An apparatus as claimed in claim 1 or claim 2, further comprising means for
comparing the determined security status of the security slice with a required security
status for the security slice.
4. An apparatus as claimed in any one of claims 1 to 3, further comprising means for
20 adding one or more additional network elements to the security slice.
5. An apparatus as claimed in claim 4, wherein the means for adding one or more
network elements to the security slice comprises means for obtaining information relating
to each of the one or more network elements to be added to the security slice.
25
6. An apparatus as claimed in any one of the preceding claims, further comprising:
means for receiving, from the attestation server, security attributes of a first
additional network element to be added to the security slice;
means for determining an integrity level of the first additional network element
30 based on the received security attributes of said first additional network element; and
means for preventing the first additional network element from being added to the
security slice in the event that said integrity level is below a required level.
7. An apparatus as claimed in any one of the preceding claims, wherein the means for
35 processing the received security attributes further comprises means for determining
whether any of the network elements of the security slice fail to satisfy a defined
requirement.

- 26 -

8. An apparatus as claimed in any one of the preceding claims, wherein the apparatus is one of a security attribute manager and a trusted slice manager.

5 9. An apparatus as claimed in any one of the preceding claims, wherein a network element comprises at least one of: a physical network element, and/or a virtualized network function, and/or a virtual machine image, and/or a virtual machine instance.

10 10. An apparatus as claimed in any of the preceding claims, wherein a network element comprises at least one of: a core network element, and/or an edge device, and/or a mobile device, and/or an Internet of Things device such as a wireless sensor.

11 11. An apparatus as claimed in any one of the preceding claims, wherein the requested security attributes comprise at least one of: a measured boot capability, and/or a secure boot capability, and/or a runtime integrity measurement, and/or a virtual machine integrity level.

12 12. An apparatus as claimed in any one of the preceding claims, wherein the security status of the security slice comprises a level of assurance.

13 13. An apparatus as claimed in any one of the preceding claims, wherein the means comprises at least one processor; and at least one memory including computer program code, the at least one memory and computer program code configured to, with the at least one processor, cause the performance of the apparatus.

14 14. A system comprising an apparatus as claimed in any one of the preceding claims and further comprising:

an attestation server for receiving requests for security attributes; and
one or more network elements.

15 15. A system as claimed in claim 12, wherein each network element further comprises a trust agent for providing an interface between the respective network element and said attestation server.

16 16. A method comprising:

- 27 -

sending one or more requests to an attestation server, wherein each request requests security attributes corresponding to one of one or more network elements of a security slice of a system;

receiving the requested security attributes from the attestation server; and

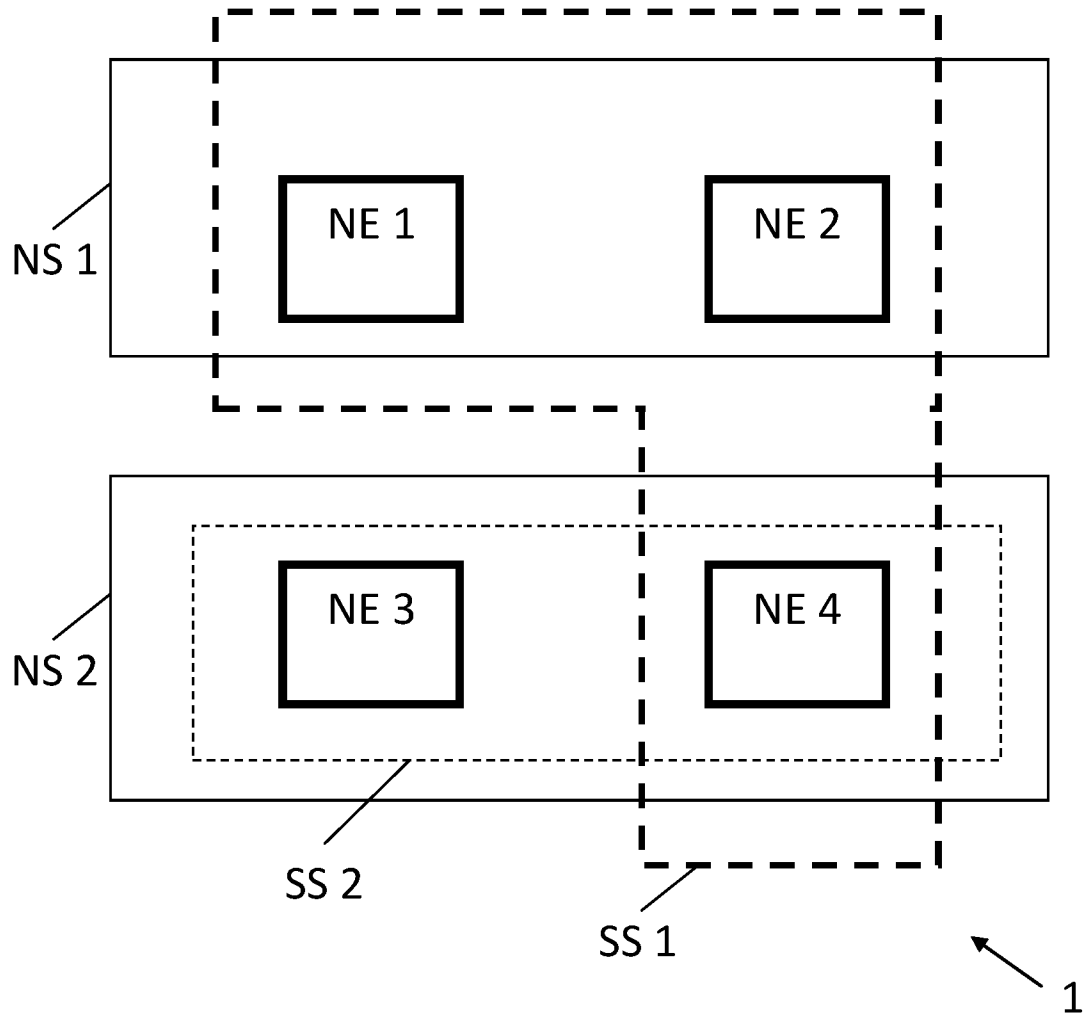
5 processing the received security attributes to determine a security status of the security slice.

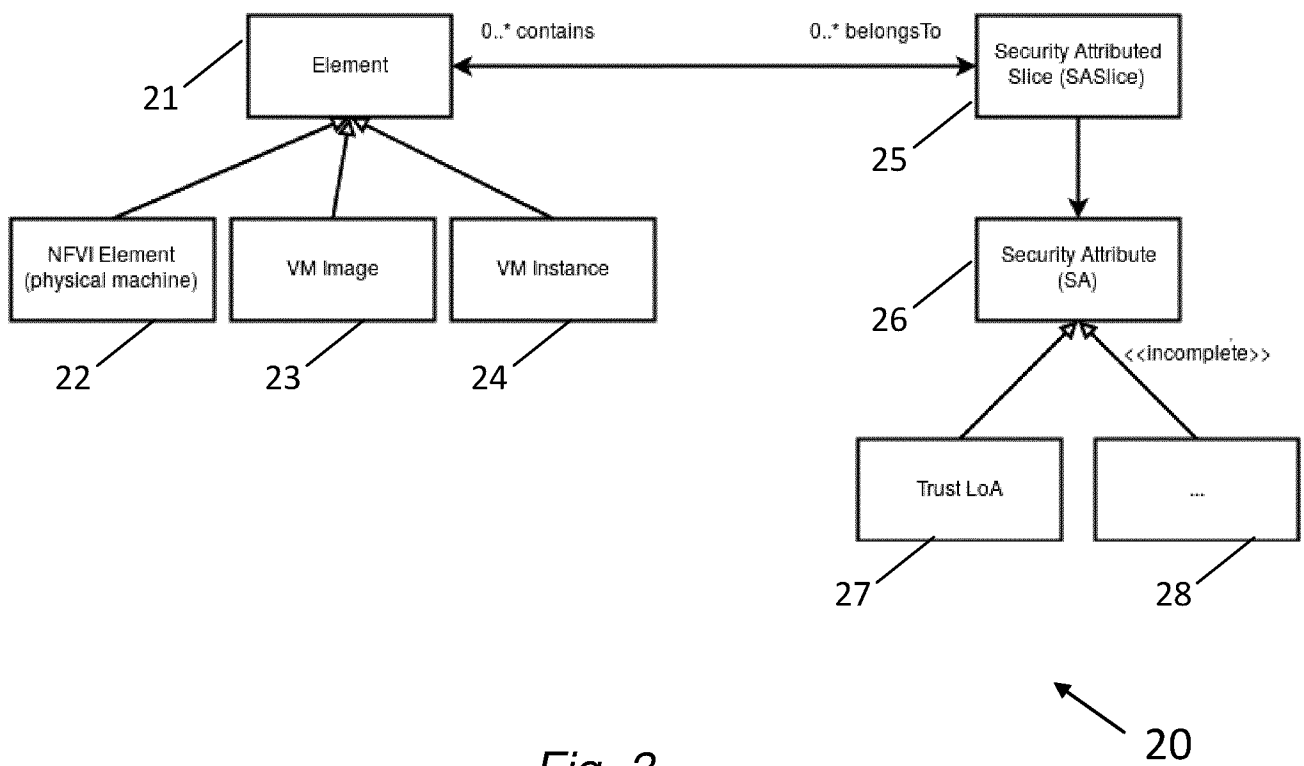
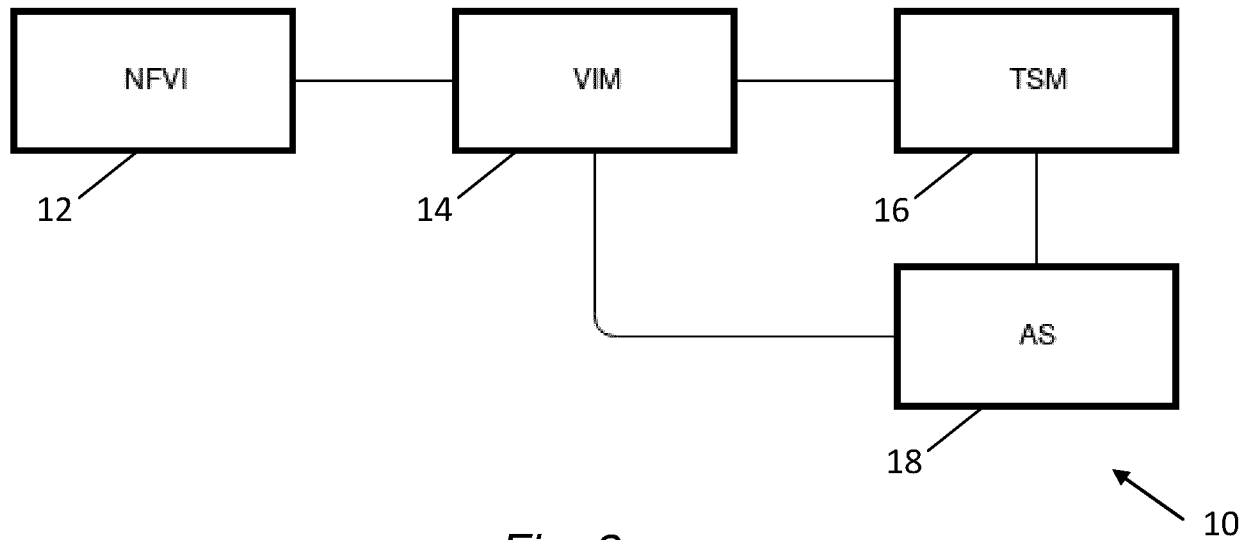
17. Computer-readable instructions which, when executed by computing apparatus, cause the computing apparatus to perform a method of:

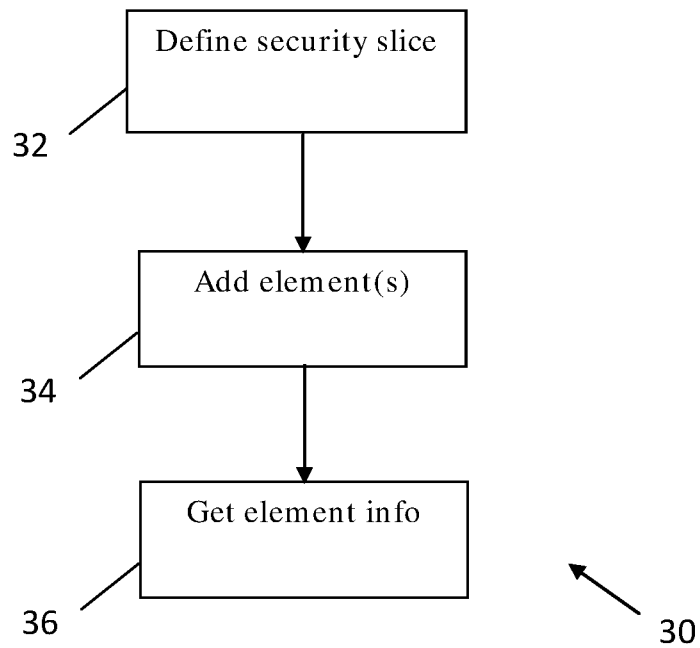
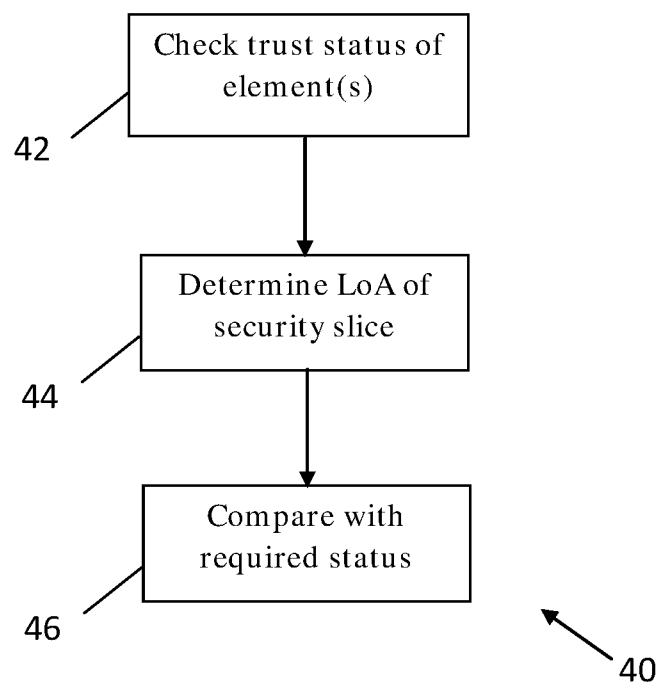
10 sending one or more requests to an attestation server, wherein each request requests security attributes corresponding to one of one or more network elements of a security slice of a system;

receiving the requested security attributes from the attestation server; and

15 processing the received security attributes to determine a security status of the security slice.

*Fig. 1*



*Fig. 4**Fig. 5*

4/9

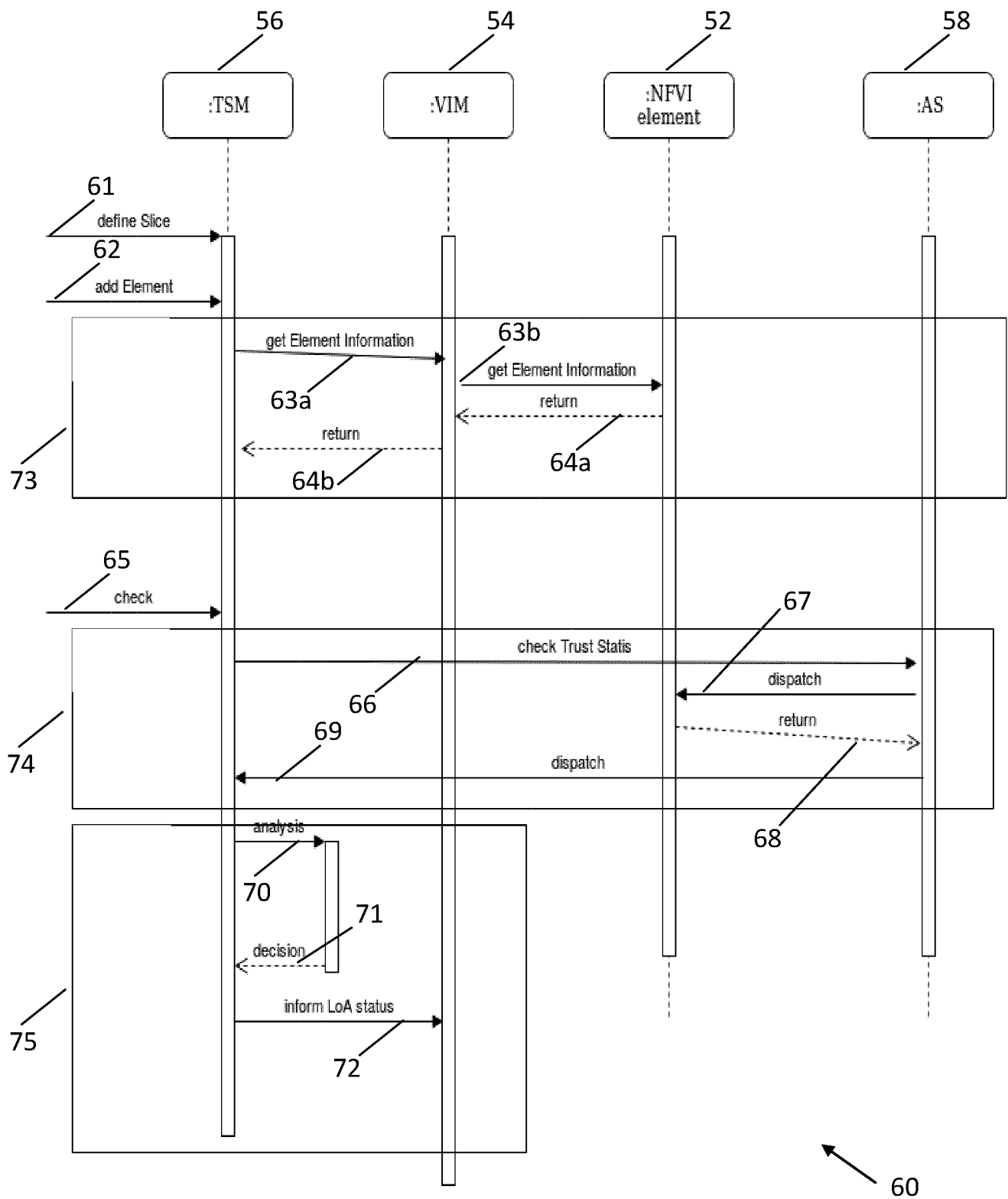


Fig. 6

5/9

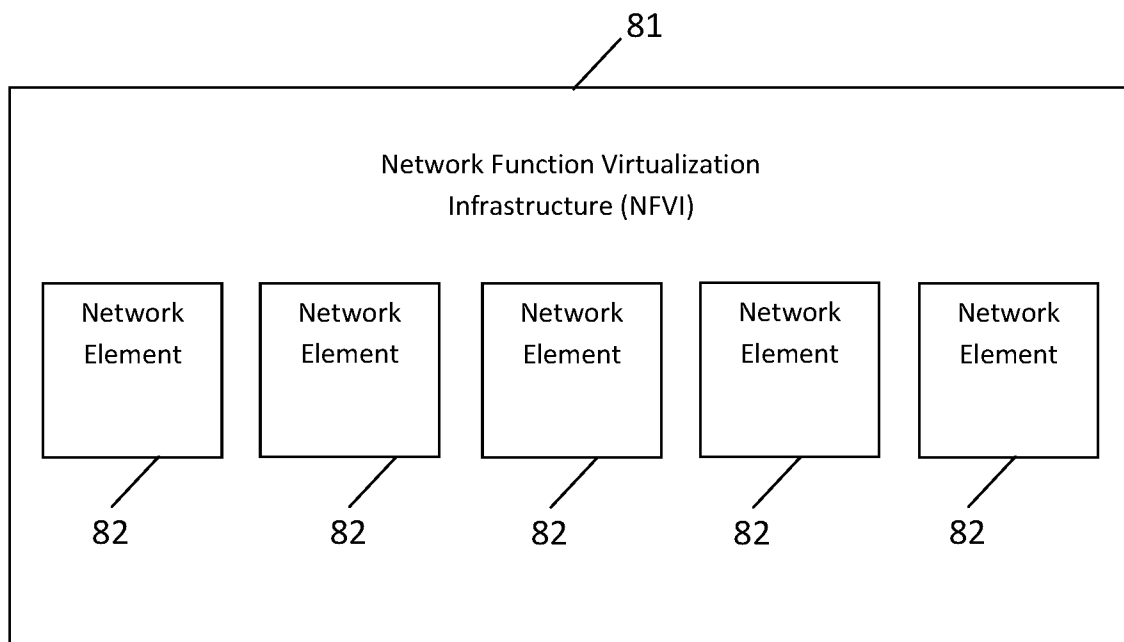


Fig. 7A

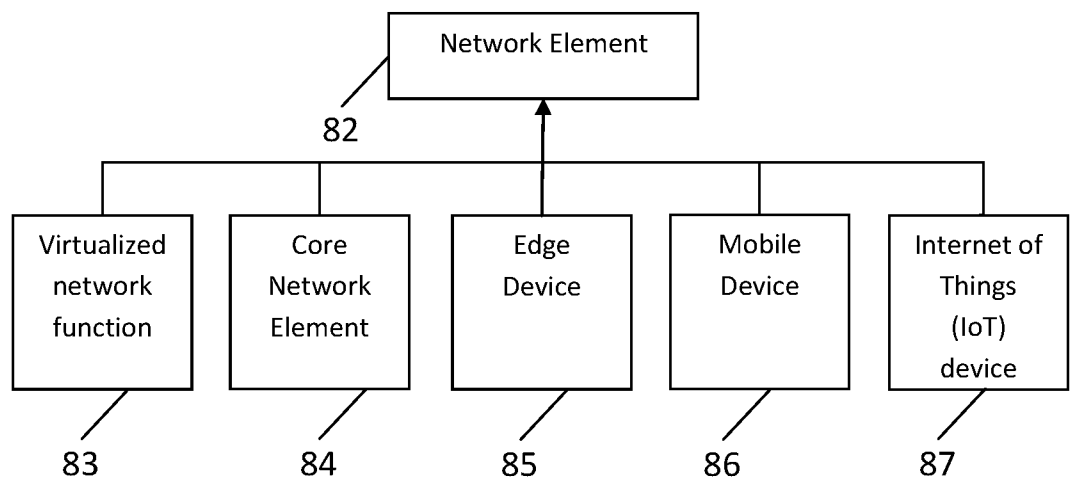


Fig. 7B

6/9

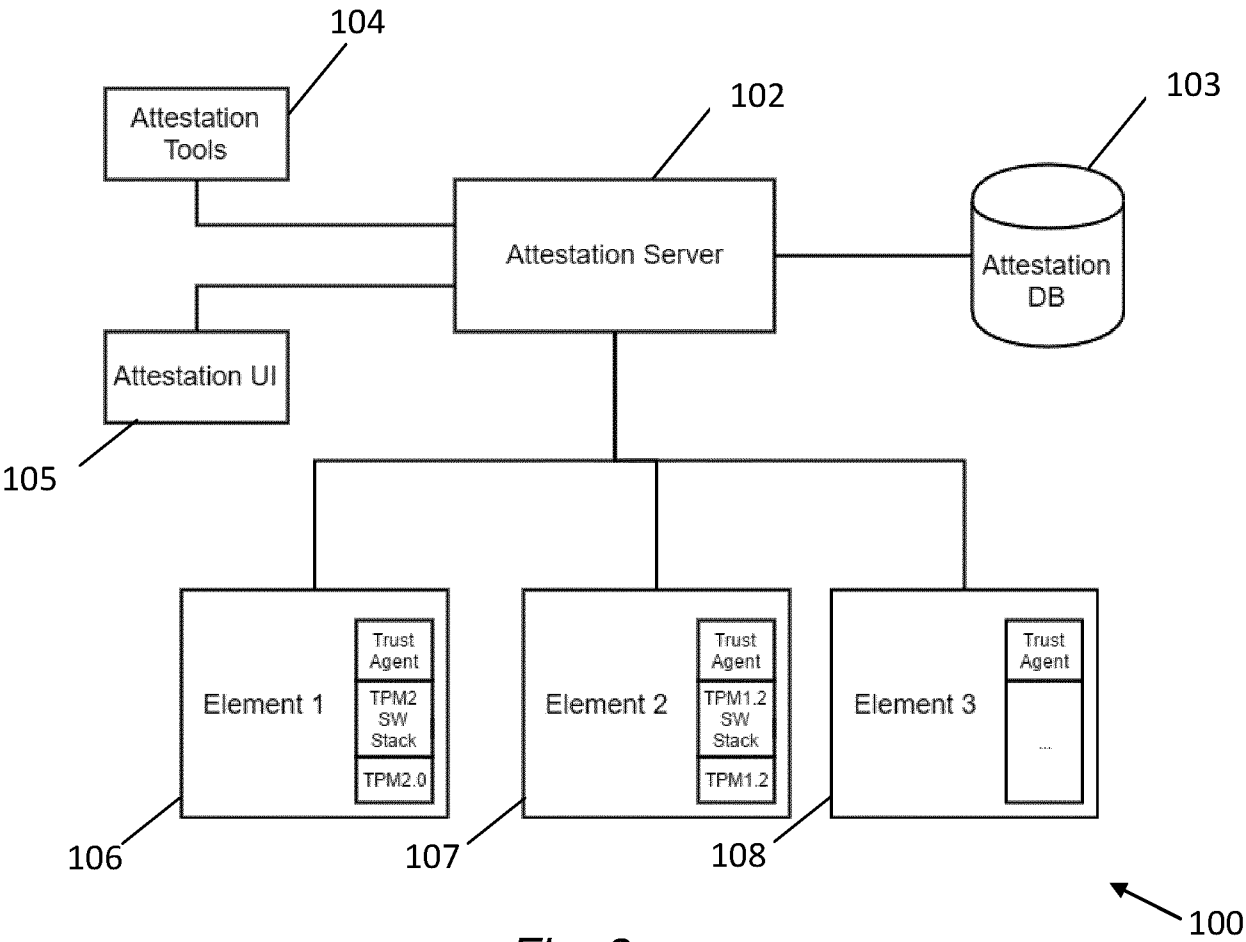
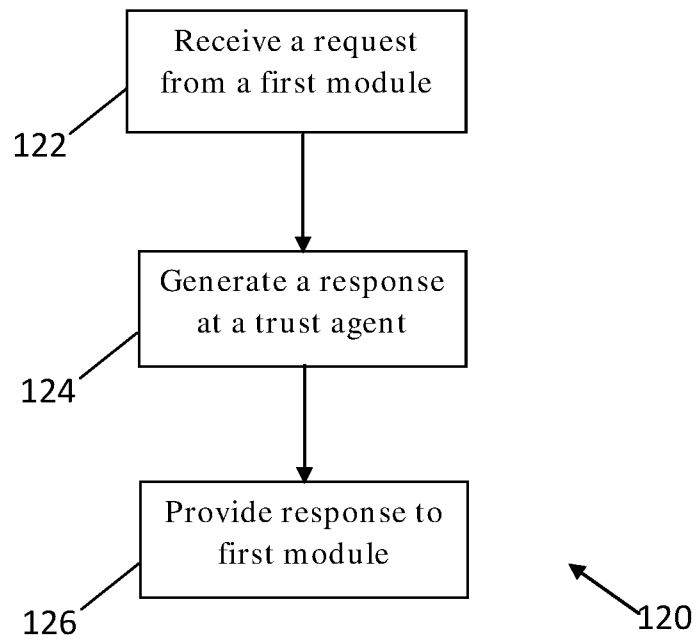
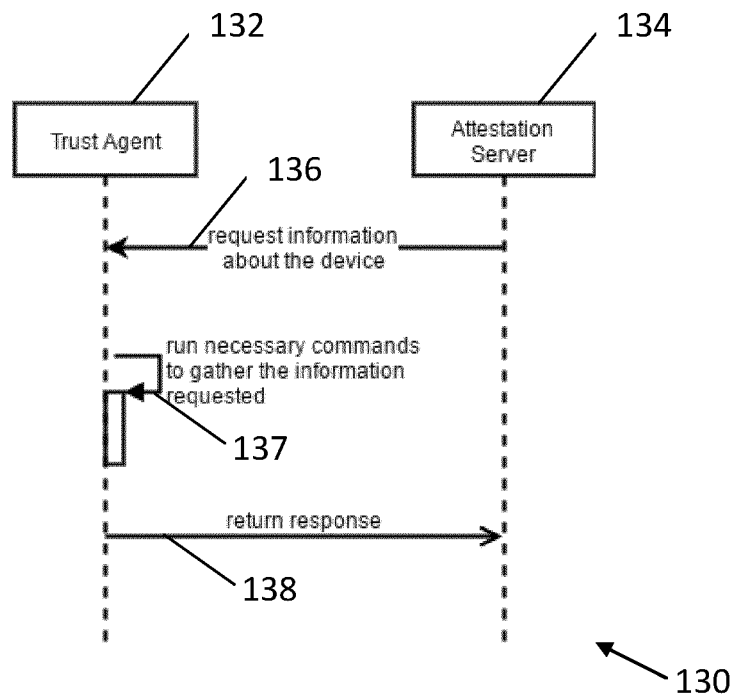


Fig. 8

*Fig. 9**Fig. 10*

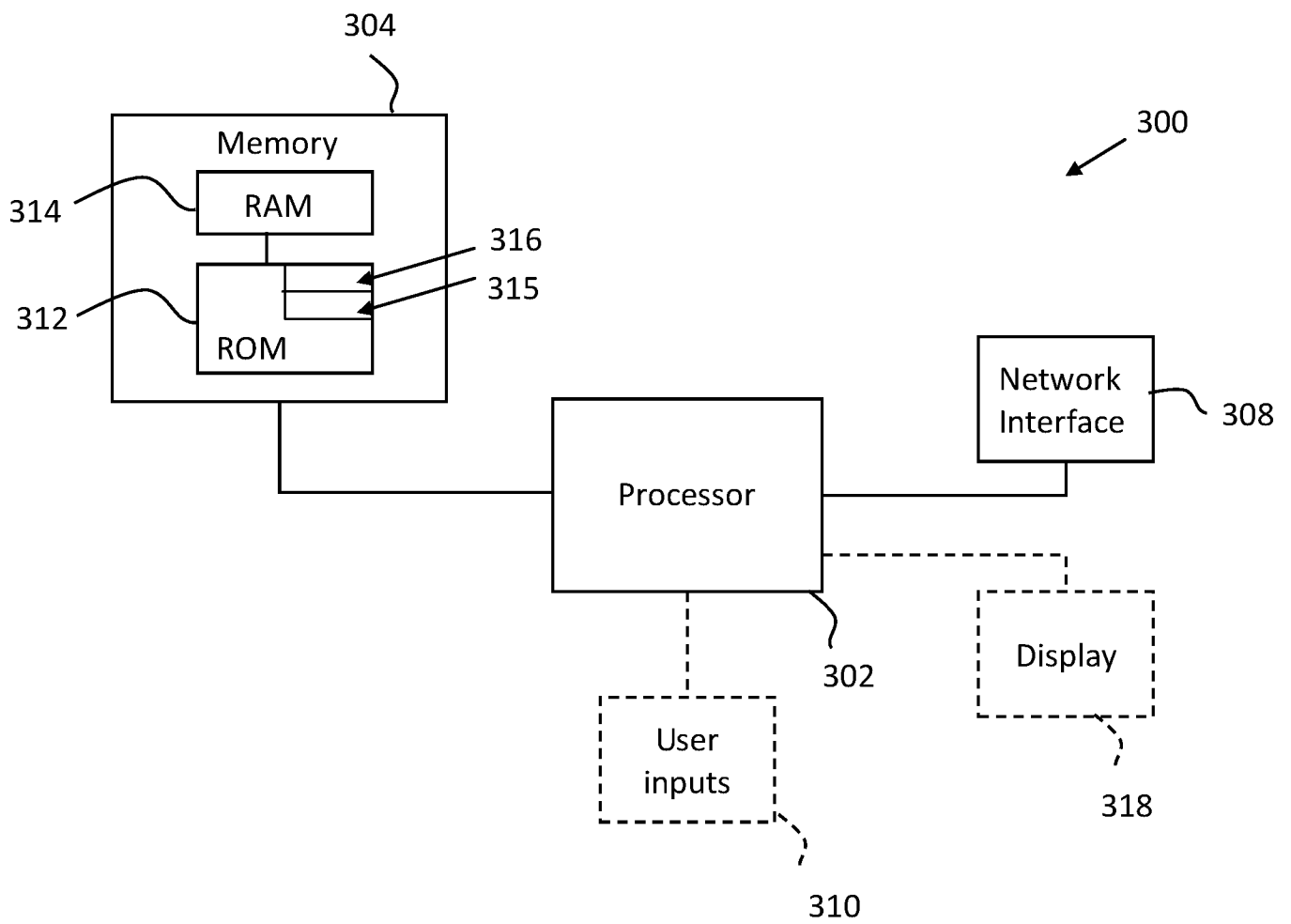
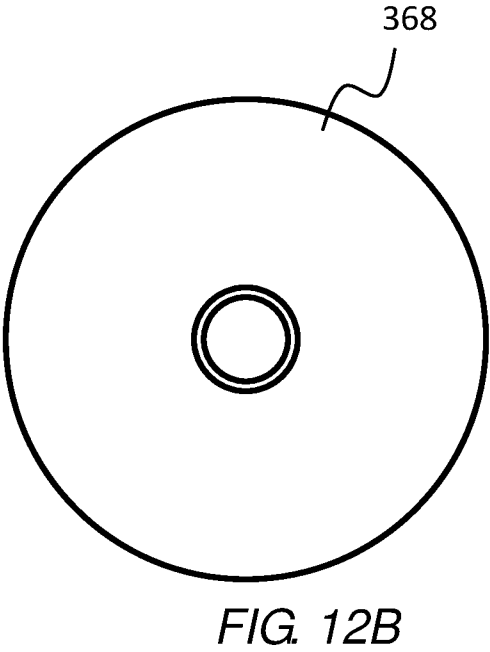
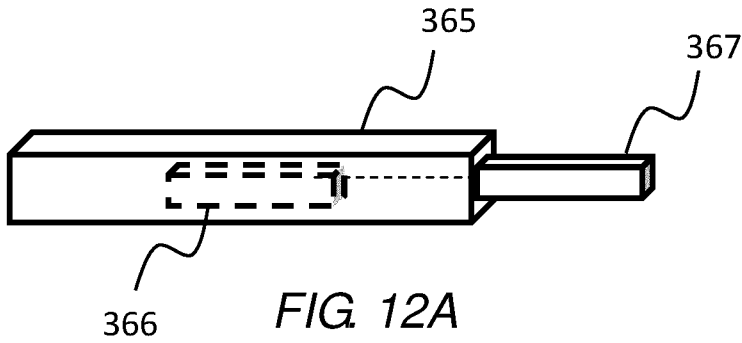


FIG. 11



INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2020/052076

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/57 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)
G06F 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 2018/324187 A1 (HEINTEL MARKUS [DE] ET AL) 8 November 2018 (2018-11-08) paragraphs [0002], [0012] - [0017], [0022] - [0023], [0036] - [0040], [0045] - [0050], [0058] claim 1 figures 1, 2A	1-17
A	US 2018/213003 A1 (FAYNBERG IGOR [US] ET AL) 26 July 2018 (2018-07-26) paragraphs [0002], [0005] - [0006], [0017], [0020], [0022], [0024], [0029] - [0031] claims 1, 2 figures 1, 2	1-17
	----- -/-	



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

20 March 2020

Date of mailing of the international search report

30/03/2020

Name and mailing address of the ISA/

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

Authorized officer

Volpato, Gian Luca

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2020/052076

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	"Network Functions Virtualisation (NFV); Trust; Report on Attestation Technologies and Practices for Secure Deployments", ETSI DRAFT SPECIFICATION; NFV-SEC 007, EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (ETSI), 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS ; FRANCE , vol. WG NFV SEC Security, no. V0.0.7 18 February 2017 (2017-02-18), pages 1-27, XP014303445, Retrieved from the Internet: URL:docbox.etsi.org/ISG/NFV/Open/Drafts/SEC007_NFV_Attestation_report/NFV-SEC007v007.docx [retrieved on 2017-02-18] cited in the application the whole document	1-17
A	----- WO 2017/188392 A1 (NEC CORP [JP]) 2 November 2017 (2017-11-02) paragraphs [0001], [0004], [0009], [0031] - [0038], [0080] figures 1, 3 -----	1-17

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2020/052076

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018324187 A1	08-11-2018	CN 108141433 A EP 3366018 A1 US 2018324187 A1 WO 2017067599 A1	08-06-2018 29-08-2018 08-11-2018 27-04-2017
US 2018213003 A1	26-07-2018	CA 3051411 A1 CN 110199283 A DE 112018000525 T5 GB 2573726 A US 2018213003 A1 WO 2018140628 A1	02-08-2018 03-09-2019 10-10-2019 13-11-2019 26-07-2018 02-08-2018
WO 2017188392 A1	02-11-2017	EP 3449610 A1 JP 2019516309 A US 2019141081 A1 WO 2017188392 A1	06-03-2019 13-06-2019 09-05-2019 02-11-2017