



(51) International Patent Classification:

H04L 9/08 (2006.01) *H04L 29/06* (2006.01)
H04L 9/32 (2006.01)

(21) International Application Number:

PCT/US2017/014423

(22) International Filing Date:

20 January 2017 (20.01.2017)

(25) Filing Language:

English

(26) Publication Language:

English

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

(71) Applicant (for LC only): **NOKIA USA INC.** [US/US]; 200
South Mathilda Avenue, Sunnyvale, CA 94086 (US).

(72) Inventors: **LAL, Shankar**; Lintukorventie 2 N 149, 02660
Espoo (FI). **OLIVER, Ian Justin**; Hitantie 153, 01150

Söderkulla (FI). **KALLIOLA, Aapo**; Otsolahdentie 16
A 44, 02110 Espoo (FI). **MICHE, Yoan Jean Claude**;
Joupinmäenkuja 3 B40, 02760 Espoo (FI).

(74) Agent: **ZHANG, Li (Lily)** et al.; Mintz Levin Cohn Fer-
ris Glovsky and Popeo, P.C., 3580 Camel Mountain Road,
Suite 300, San Diego, CA 92130-6768 (US).

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IR, IS, 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.

(54) Title: MULTIPLE REMOTE ATTESTATION SERVICE FOR CLOUD-BASED SYSTEMS

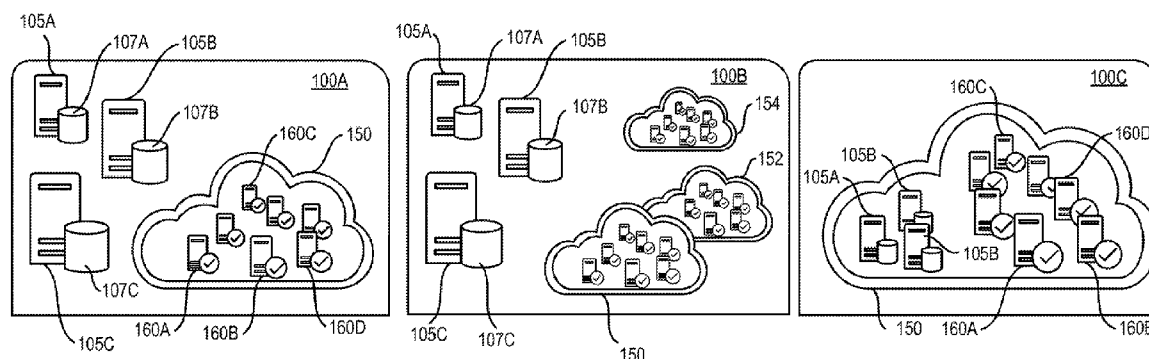


FIG. 1

(57) Abstract: Methods and apparatus, including computer program products, are provided for attesting trust. In some example embodiments, there may be provided a method that includes receiving, by a root attestation server, a request to attest a client, the request including an identity of the client and at least one measurement value for the client; sending, by the root attestation server, in response to receiving the request, a plurality of attestation requests to a plurality of attestation servers, the attestation requests including the identity and the at least one measurement; receiving, from the plurality of attestation servers, a plurality of attestation responses indicating whether the client is attested; determining, based on the received plurality of responses, whether to attest the client; and sending, by the root attestation server, the final attestation response to another node, the final attestation response based on the determining. Related systems, methods, and articles of manufacture are also described.

(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).

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

Multiple Remote Attestation Service for Cloud-Based Systems

Field

[001] The subject matter described herein relates security, and in particular, attestation.

Background

[002] When virtualization technology is used in a cloud-based system, the hardware or software in the cloud (for example, one or more servers coupled to the Internet) can be compromised. For example, the underlying cloud-based hardware or the virtualization software therein can be compromised by an attacker. However, users of cloud-based services expect the cloud-based system to be a trusted system. Generally, the expectation of trust includes the maintaining integrity of some, if not all, system critical components in a given virtualization operating environment (for example, a virtual machine, a hypervisor, and/or the like) being able to verify the integrity of these components via a remote access.

Summary

[003] Methods and apparatus, including computer program products, are provided for attesting trust.

[004] In some example embodiments, there may be provided a method that includes receiving, by a root attestation server, a request to attest a client, the request including an identity of the client and at least one measurement value for the client; sending, by the root attestation server, in response to receiving the request, a plurality of attestation requests to a plurality of attestation servers, the attestation requests including the identity and the at least one measurement; receiving, from the plurality of attestation servers, a plurality of attestation responses indicating whether the client is attested; determining, based on the received plurality

of responses, whether to attest the client; and sending, by the root attestation server, the final attestation response to another node, the final attestation response based on the determining.

[005] In some variations, one or more of the features disclosed herein including the following features can optionally be included in any feasible combination. The plurality of attestation servers may include the root attestation server. Each of the plurality of attestation servers may be attested, when the request is received. The attesting may include comparing stored, reference measurement values for the plurality of attestation servers to current, measurement values for the plurality of attestation servers. The at least one measurement value may include a hash value and/or a measurement value obtained from a non-volatile memory of a hardware security module. The plurality of attestation requests sent to the plurality of attestation servers may enable the plurality of attestation servers to attest the client based on at least a comparison of the at least one measurement to at least one stored, reference measurement for the client. The final attestation response may be determined based on voting logic. The client may include a host machine in a cloud-based system coupled to the internet. The root attestation server may include redundant hardware. The root attestation server may be selected from among the plurality of attestation servers to attest the client.

[006] In some example embodiments, there may be provided a method that includes sending an attestation request, the attestation request including an identity of the client and at least one measurement value for the client; receiving from the plurality of attestation servers a plurality of attestation responses indicating whether the client is attested; of whether the client is attested.

[007] In some variations, one or more of the features disclosed herein including the following features can optionally be included in any feasible combination. The plurality of attestation requests sent to the plurality of attestation servers may enable the plurality of attestation servers to attest the client based on at least a comparison of the at least one

measurement to at least one stored, reference measurement for the client. The at least one measurement value may include a hash value and/or a measurement value obtained from a non-volatile memory of a secure component. A verifier may send the attestation request, receive the plurality or attestation responses, determine whether to attest, and send the indication. The verifier may include redundant hardware.

[008] The above-noted aspects and features may be implemented in systems, apparatus, methods, and/or articles depending on the desired configuration. The details of one or more variations of the subject matter described herein are set forth in the accompanying drawings and the description below. Features and advantages of the subject matter described herein will be apparent from the description and drawings, and from the claims.

Description of Drawings

[009] In the drawings,

[010] FIG. 1 depicts an example of a system including a plurality of attestation servers, in accordance with some example embodiments;

[011] FIG. 2 depicts an example of a process for attestation based on a root server, in accordance with some example embodiments;

[012] FIG. 3 depicts an example of a process for attestation based on distributed attestation servers, in accordance with some example embodiments, in accordance with some example embodiments;

[013] FIG. 4A depicts an example of a process for handling attestations, in accordance with some example embodiments;

[014] FIG. 4B depicts an example of another process for handling attestations, in accordance with some example embodiments and

[015] FIG. 5 depicts an example of an apparatus, in accordance with some example embodiments.

[016] Like labels are used to refer to same or similar items in the drawings.

Detailed Description

[017] Attestation is a way to verify the authenticity of a given software-based system including cloud-based systems. In attestation, the software-based system may testify that it is what it claims to be. In trusted computing, attestation may be carried out by a remote, trusted third party, such as a remote attestation service at an Internet-coupled server. A reason for using the remote third party is that the software-based system may not be able monitor itself in secure manner, while another reason may be the desire to have an entity other than the software-based system make determinations regarding authenticity and trust. For example, when the software-based system is attacked and/or compromised, the software-based system's monitoring tools can be compromised as well and switched off. Attestation and its associated reporting to a third party server may be used to prove that the components of a software-based system can be trusted, when the components are integral, untampered, and/or are in a known state. These components may include software components such as the physical server's files, software, firmware, BIOS, bootloader, operating system, hypervisor, and/or the like.

[018] In trusted computing, the attestation may be based on verifying a given, measurement (or, for example, list of measurements). The measurements may represent information that is relevant to the determination of the integrity and/or trust of the component. These measurements may correspond to current measurements representative of runtime integrity, data protection, and/or boot integrity. The current measurement values may be compared to known, or reference, measurement values. For example, the current measurement value(s) may comprise one or more hash values of some, if not all, of the software-based system's software components. This may also include code, values, identifiers, and/or the like written into/on disk, memory, read only memory, ASIC, FPGA, and/or the like. When the

software components at the software-based system are changed, the measurement values, such as the hashes of the software components, change. In this example, when the current measurement values do not match the reference measurement values, a possible compromise or an unknown state may exist at the software-based system (or physical server(s) or client machine(s) hosting the software-based system), and this possible compromise or an unknown state may indicate an untrusted state.

[019] The physical servers hosting the software-based system being attested may include a trusted computing component, such as a trusted platform module (TPM), a secure (or trusted) module, and/or a secure (or trusted) hardware module. The trusted computing component may include hardware, such as a circuitry, configured to securely measure and/or record the state of the physical server including the software components such as the physical server's files, software, firmware, BIOS, bootloader, operating system, hypervisor, and/or the like. In the case of a TPM, the TPM may securely store, in platform configuration registers (PCR), the current measurement values as hash values, such as a one-way hash values, of the software components. These current measurement values may be compared to known, reference measurement values (which may also be in the form of hash values) to determine whether physical server including the software components are in a known state or whether a change has occurred (as indicated by a difference between the current measurement values and the known, reference measurement values). When a current measurement value does not match the a reference measurement value, the physical server including the software components cannot be attested, in which case the system cannot be trusted. In some example embodiments, the TPM may be configured in accordance with one or more standards including: ISO/IEC 11889-1:2009 Information technology—Trusted Platform Module —Part 1: Overview; ISO/IEC 11889-2:2009 Information technology—Trusted Platform Module—Part 2: Design principles; ISO/IEC 11889-3:2009 Information technology—Trusted Platform Module—Part 3: Structures; and/or ISO/IEC

11889-4:2009 Information technology—Trusted Platform Module—Part 4: Commands. Although some of the examples herein refer to TPM, other secure modules may be used as well. For example, a non-volatile memory portion of a secure component may be used as well. When this is the case, the memory may store the historical hash measurements to enable a trust determination. Moreover, secure or trusted components/platforms other than TPM may be used as well.

[020] As noted above, attestation may be carried out by a remote, trusted third party, such as a remote attestation service at an Internet-coupled server. The remote, third party server (which is remotely attesting a software-based system, such as a cloud-based system) may be the primary, if not, only entity that decides about the trust state of the software-based system, such as a cloud-based system. Supposing however a scenario in which the third party server (hereinafter the attestation server) is itself compromised by, for example, an attacker and/or the like. This compromise may take many forms. For example, the attestation server may be attacked so that the attestation server redefines what measurements are performed on the cloud-based system during attestation. Alternatively or additionally, the attestation server may be attacked so that the attestation server generates a false attestation report regarding the trust of the cloud-based system including the host platforms or software components at the hosts. Alternatively or additionally, the attestation server may be attacked so that the attestation server is unavailable for verification requests resulting in a trust failure for the hosts at the cloud-based system, for example. However, using only a single, remote attestation server, as noted above, may also pose a risk of having a single point of failure, in which case the attestation server can be the target of denial of service attack. Moreover, because the attestation server may not always run on trusted hardware, the attestation server can be considered an easier target for attack by adversaries seeking to penetrate or harm the core, cloud-based system infrastructure.

[021] In some example embodiments, a plurality of remote attestation servers may be provided to attest a system, such as a software-based system including for example a cloud-based system. In some example embodiments, the plurality of remote attestation servers may be configured into a hierarchy of remote attestation servers.

[022] In some example embodiments, some, if not all, of the remote attestation servers may operate and execute code or instructions on at least one trusted machine (such as a trusted machine/platform with secure component(s) for example, a TPM and/or the like) configured to at least provide verification of the hardware and/or software of the software-based system, such as the cloud-based system. Moreover, the trusted machine(s) may be configured to allow verification of the integrity among or between remote attestation servers (herein after attestation servers). Although some of the examples described herein refer to attesting a cloud-based system, the attestation disclosed herein may be used to attest other types of systems including software-based systems.

[023] In some example embodiments, a plurality of attestation servers may implement voting to attest the trust of a cloud-based system. For example, when voting to attest a given client system, each attestation server may vote on the client system's integrity based on one or more reference measurement values stored in the attestation server's database. To determine a trust state of the client system, the votes from each attestation server may be taken into account. For example, suppose there are 3 attestation servers voting to attest a client system virtualized at the cloud-based system. If 2 out of 3 remote servers vote that the client system is trusted, the client system would be attested as trusted. However, if 2 out of 3 remote servers vote that the client system is untrusted, the client system would be attested as untrusted.

[024] FIG. 1 depicts implementation examples 100A-C of a plurality of attestations servers, in accordance with some example embodiments.

[025] At system 100A, the system 100A may include a plurality of attestation servers 105A-C coupled via at least the Internet to a cloud-based system 150 including one or more clients (also referred to as hosts) 160A-D and/or the like. In the example of system 100A, the plurality of attestation servers 105A-C may be configured in a structured and/or predetermined way, such as in a hierarchy of attestation servers 105A-C. This hierarchy may be used to attest the trust of the cloud-based system 150 and/or any of the client's 160A-D of the cloud-based system. In the configuration depicted, the plurality of attestation servers 105A-C attest for a single cloud-based system 150.

[026] The attestation servers 105A-C may each include a corresponding database 107A-C containing one or more reference measurement values (also referred to as measurement lists). In some example embodiments, the attestation servers 105A-C may attest one another in addition to attesting clients, such as clients 160A-D. For example, in addition to attesting clients, a first attestation server, such as attestation server 105A, may attest one or more other attestation servers 105B, C, and/or the like. To attest attestation server 105B for example, remote attestation server 105A may compare a reference measurement value (for example, a stored hash of the file(s) at attestation server 105B) with a current measurement value for attestation server 105B. In some implementations, each of the attestation servers 105A-C may be implemented using fault tolerant hardware and/or redundancy to minimize outages and to avoid partial outages or compromise of an attestation server that can cause a Byzantine fault from affecting the resulting attestation of clients or other attestation servers.

[027] The one or more clients 160A-D may comprise at least one processor and at least one memory including instructions to cause one or more of the operations described herein. The one or more clients 160A-D may be coupled to at least the Internet to provide cloud-based services, such as software as a service (SaaS), cloud-based storage, and/or other services. For example, the one or more client machines 160A-D may each include virtualization software to

provide a virtual machine out of which a service can be provided. The cloud-based system 150 may also include at least one hypervisor as well. The client machines 160A-D may also include a trusted component, such as a TPM and/or the like, to process and/or perform measurements, as noted above, on the corresponding client machine's component(s). In some example embodiments, the one or more client machines 160A-D may each include an attestation agent to gather measurement values, request attestation, send attestation requests (including measurement values) to other devices, and receive responses to the attestation requests.

[028] The system 100B may be similar to system 100A in some respects but the hierarchy of attestation servers 105A-C may provide attestation for a plurality of cloud-based systems 150-154 including the clients therein.

[029] The system 100C is similar to systems 100A and 100B in some respects but the hierarchy of attestation servers 105A-C is within the cloud-based system 150, rather than outside the cloud-based system as in the case at system 100A and 100B.

[030] In some example embodiments, at least one of the attestation servers may be designated a "root" attestation server. When this is the case, the attestation servers may be considered to be in a hierarchical attestation server configuration. In the hierarchical attestation server configuration, the root attestation server may be configured to be trusted, and may be used to attest other attestation servers. However, in some example embodiments, a root attestation server may not be used, in which case the attestation servers may be configured in a distributed configuration, which may be considered non-hierarchical. When the distributed configuration is used, the attestation servers may be configured to configure voting logic to attest a client such as client 160A for example and/or to establish a voting logic to attest other remote attestation servers.

[031] The use of different voting logic may allow for different and/or configurable voting configurations. For example, in some instances having a voting logic that requires an odd

number of attestation servers avoids having a tie vote (which would cause delays in attestations). As noted in the example above where 3 attestation servers are implemented, there would never be a tie since the vote would always be 2 to 1 either attesting trust of a client or indicating trust is not present at a client. However, the odd server-voting scheme may not be suitable when the attestations servers attest each other. For example, in the case of 3 remote attestation servers, having the two other servers attest the third server may yield a tie. As such, the voting logic configuration to attest a remote attestation server may rely on just one other remote attestation server, rather than a majority vote. To illustrate another voting scheme, a large pool of attestation servers may be implemented, in which case a single server would be removed from the pool to arrive at an uneven number of voting attestation servers. Although some example voting schemes are provided, the attestation may use other voting schemes or avoid voting altogether.

[032] FIG. 2 depicts an example of a system 200 including a plurality of attestation servers 105A-D configured in a hierarchal manner, in accordance with some example embodiments.

[033] In some example embodiments, the system 200 may include a verifier 290, in accordance with some example embodiments. The verifier 290 may provide cloud management to manage at least cloud-based system 150 and a cloud attestation controller to enable verification and attestation of the clients in the cloud-based system 150.

[034] In some example embodiments, the verifier 290 may be configured to send to the root attestation server 105A a request (also referred to herein as an attestation request) to attest a given cloud client, such as client 160A, for example. In response to the attestation request, the verifier 290 may receive from the root attestation server 105A a response including final trust state determination for the client, such as client 160A. If the response indicates that

client 160A can be trusted, the verifier 290 may enable client 160A to be allocated a task or work at cloud 150.

[035] In the example of FIG. 2, at least one of the attestation servers, such as attestation server 105A may be configured to serve as a root attestation server. The root attestation server 105A may include reference measurement values (for example, a white list of known, hash values), of some, if not all, of the other three attestation servers 105B-D (which are lower in the hierarchy). For example, a first, reference measurement value may, as noted above, be stored in a database, and the first, reference measurement value may comprise one or more hashes of the file(s) at remote attestation server 105B. The second reference measurement value may be stored in a database, and may comprise reference measurement values for remote attestation server 105C; and so forth. In this way, the root attestation server 105A can receive a current measurement value from another remote attestation server, and then compare the current measurement value to the stored reference measurement value for a given remote attestation server to perform attestation.

[036] In some example embodiments, the quantity of attestation servers 105B-D below root attestation server 105A configured, or allowed, to vote on a given attestation request may be an odd quantity to avoid ties as noted above in the attestation. In some example embodiments, the root attestation server can be selected randomly from a pool of servers for a given session, and this selection can be a random selection (although the root attestation server may be selected in other ways as well).

[037] In some example embodiments, attestation servers 105A-D may include corresponding databases 107A-D. As noted above, the root attestation server 105A may store at database 107A reference measurement values for the other remote attestation servers 105B-D. While the databases 107B-D may store reference measurement values (for example, white list measurements) for some, if not all, of the clients in the cloud-based system 150, such as clients

160A, B, C, D, and so forth. In some example embodiments, the attestation server(s) and/or client machines 160A-D may each include a trusted computing component, such as a TPM and/or the like, to enable generation of current measurement values, which can be later be compared to the reference measurement values. Moreover, an attestation server may be configured to send, as part of attestation, its current measurement values to a trusted third party server, such as another attestation server and/or the like. In some example embodiments, communications between attestations servers may be encrypted. Moreover, communications with a trusted third party/attestation server(s) and/or the verifier may be encrypted and/or protected using protocols, such as SSL/TLS (secure sockets layer/transport layer security). An attestation server may also include key handlers for securely exchanging keys.

[038] In some example embodiments, each client 16-A-D may have a client unique identifier (ID). When this is the case, databases 107B-D may store the reference measurement values based on the client unique identifier, so a query for the client unique identifier retrieves from the corresponding database the reference measurement values for the client mapped to the client unique identifier.

[039] At 210, the verifier 290 may receive a client identifier for a given client, such as client 160A, being attested, and receive a measurement value for that client, in accordance with some example embodiments. For example, the measurement value may comprise hash values determined for client 160A. These hash values may be a hash of the files, such as the software components and other files, at host 160A. In some example embodiments, the measurement values may be obtained from non-volatile memory (for example, a PCR) at a trusted component (for example, TPM circuitry) at the client being attested.

[040] At 220, the verifier 290 may form an attestation request including the client's unique identifier and the measurement value and then send the formed attestation request to the root attestation server 105A, in accordance with some example embodiments. For example, an

attestation request may include the client's unique identifier mapped to the measurement value. Moreover, an attestation request may include a plurality of unique identifiers for a plurality of clients being attested. Each unique identifier may be mapped to the corresponding measurement value for that client.

[041] At 230, the root attestation server 105A may (if it has not already done so) attest the other attestation servers 105B-C, and may send the attestation request to one or more attestation servers which have been determined to be trusted, in accordance with some example embodiments. In some example embodiments, the receipt of the attestation request at the root attestation server 105A may trigger server 105A to attest the other attestation server 105B-C. For example, root attestation server 105A may, as noted above, receive a current measurement value, such as a hash of the files, for each of the other attestation servers 105B-D, and then compare the current measurement value to the stored, reference measurement values (which can be stored as a so-called "white list" of measurement values) to attest each server 105B-D. In some example embodiments, the root attestation server 105A may, however, verify the trust of the other attestation servers 105B-D at other times as well.

[042] At 240A-C, each of the attestation servers 105B-D may perform an attestation in response to the received attestation request, and may return the attestation result to root attestation server 105A, in accordance with some example embodiments. For example, each attestation server may compare the current measurement value (which is included in the attestation request) to the reference measurement value, which may be stored in a corresponding database. If the comparison yields a match, the client being attested can be attested as being trusted. But if the comparison does not yields a match (for example, the current hash value does not match the reference hash values), the client being attested cannot be trusted. Each of the attestation servers 105B-D may respond to the root attestation server 105A with the results of the comparison to indicate whether the client can be trusted.

[043] At 250, the root attestation server 105A may compare the results from each of the other attestation servers, and may send to the verifier 290 a final decision regarding the trust state of the client that is the subject of the attestation request, in accordance with some example embodiments. Although the final results may be determined in a variety of ways, in some example embodiments, the root attestation server 105A may use voting logic and select the final attestation result based on a majority of votes for attestation of the client. Returning to the 3 attestation servers example above, if 2 out of 3 remote servers vote that the client can be trusted, the client would be attested as trusted. But if 2 out of 3 remote servers vote that the client is untrusted, the client would be attested as untrusted. The attestation response message (which may be sent by root attestation server 105A to the verifier 290) may include the client's unique identifier and its trust state.

[044] At 260, the verifier 290 may determine whether to initiate a session, such as place a workload at the client or initiate another task, at the client based on the trust state indicated by the response provided at 250, in accordance with some example embodiments. For example, if at 250, the root attestation server indicates that client 160A is trusted, verifier 290 may initiate or allow to continue a session at client 160A. But if at 250, the root attestation server indicates that client 160A is not trusted, verifier 290 may not initiate or kill a session at client 160A.

[045] To recapitulate by way of an example, an attestation request from verifier 290 may include a client ID and set of current, measurement values indicative of the integrity of the client. When the attestation request is received, the root attestation server 105A may query for the trust status of some, if not all, of the attestation servers. Next, the root attestation server 105A may send attestation requests (including the measurement value(s) and client ID) to only those attestation servers that can be trusted. The attestation servers participating in voting may then send back the client's trust state (which may include the client's ID) to root attestation

server based on a comparison of the client's current measurement values with the reference measurement values. The attestation response from root attestation server to the verifier may comprise a client's unique identifier (ID) and its trust state. Based on the attestation result from the root attestation server, the verifier may determine whether to allow to continue a session at client 160A.

[046] Alternatively or additionally, the verifier 290 may send attestation request(s) to each of the attestation servers 105B-D, rather than to the root attestation server 105A. When this is the case, each of the attestation servers 105B-D may query root attestation server 105A to verify their trust state before sending the attestation response to the verifier 290. In the case of voting, the verifier may include voting logic to determine the final trust determination for the client.

[047] FIG. 3 depicts another example of an implementation of a system 300 in which the attestation servers are configured, in accordance with some example embodiments. System 300 is similar to system 200 in some respects but system 300 is configured in a non-hierarchical manner. In the example embodiment of FIG. 3, the attestation servers 105A-D are each configured to receive attestation requests from the verifier 290. Moreover, the attestation servers 105A-D may also be coupled to enable attestation among of each other.

[048] In some example embodiments, the quantity of attestation servers 105-A-D allowed to attest to a given attestation request may be odd to avoid the ties as noted above, although even quantities of attestation servers may be used as well. Moreover, although some of the examples refer or show specific quantities of attestation servers, this is merely an example as other quantities may be used as well.

[049] Like system 200, the attestation servers 105A-D may be coupled to, or include, a corresponding database 107A-D. The database(s) 107A-D may include the reference

measurement values (also referred to herein as whitelist measurements) of the clients, such as clients 160A-D and reference lists of the other attestation servers.

[050] At 310, the verifier 290 may receive a client identifier for a given client to be attested, and receive a measurement list for that client, in accordance with some example embodiments. In some example embodiments, 310 may be implemented in a similar manner as noted above with respect to 210.

[051] At 320, the verifier 290 may form an attestation request including the client's unique identifier and the measurement list and then send the formed attestation request to at least one, if not all, of the attestation servers 105A-B, in accordance with some example embodiments. The attestation request from the verifier may have the same or similar form as noted above with respect to FIG. 2.

[052] In response to receiving the attestation request, the attestation servers 105A-D may each respond, at 330, to the verifier with trust state for the client identified in the attestation request, in accordance with some example embodiments. For example, each attestation server may compare the current measurement value (which is included in the received attestation request) to the reference measurement value, which may be stored in the corresponding database 107A-B. If the comparison yields a match, then the files at the client being attested can be attested to as being in a trusted state. But if the comparison does not yields a match (for example, the two measurement list/hashes do not match), then the files at the client being attested cannot be trusted.

[053] At 340, the verifier 290 may receive the attestation responses from the attestation servers 105A-C and, based on the responses, determine a final trust state of the client, in accordance with some example embodiments. For example, the verifier 290 may include voting logic to, as noted above, to tally and determine whether the client is in a trusted state.

[054] At 350, the verifier 290 may determine whether to initiate a session, such as place a workload at the client or initiate another task, at the client based on the trust state indicated by the response provided at 340, in accordance with some example embodiments. If at 340 for example, a determination is made that a client, such as client 160A is trusted, verifier 290 may initiate or allow to continue a session at client 160A. But if at 340, a determination is made that a client, such as client 160A is not trusted, verifier 290 may not initiate or kill a session at client 160A.

[055] In some example embodiments, before processing an attestation request, the attestation servers 105A-D may attest each other and then proceed to verify a client's trust state, although the attestation of the attestation servers may be performed at other times as well.

[056] In some example embodiments, the attestation response may include a set of trust states provided by some, if not all, of the attestation servers 105A-D along with client trust state. For example, the set of trust states may include for example "true" for trusted and "false" for untrusted systems, although other types of indicators may be used for trusted state and untrusted state. In some example embodiments, the attestation response may be of the following form:

Response = [{attestation server₁ = True, attestation server₂ = True, attestation server₃ = False}, Client_{UUID} = True],

wherein "True" represents that the corresponding attestation server₁ is trusted.

[057] The verifier 290 may process attestation responses from the attestation servers 105A-D, and may determine a final trust state based on for example a majority of attestation responses (although other voting schemes including a plurality, unanimity, and/or the like may be used as well). Table 1 depicts an example of an attestation results from each attestation server and the final attestation results.

[058] Table 1

Use Case	Attestation server 105A	Attestation server 105B	Attestation server 105C	Final Attestation Result
1.	✓	✓	✓	True
2.	✓	✓	✗	True, notify
3.	✓/✗	✓/✗	-	Byzantine fault
4.	✗	✗	-	False

[059] In use case 1, all of the attestation servers 105A-D provide an attestation response indicating a true state (as indicated by the ✓), so the client being attested may have a trusted state (as indicated by the OK in the last column). In use case 2, attestation servers 105A-B provide an attestation response indicating true but attestation server 105C does not indicate a trusted state (as indicated by the ✗). In this second use case, the client being attested may, due to voting, have a trusted state (as indicated by the OK in the last column), but the root attestation server or verifier may send a message to notify of the false, untrusted indication (for possible further processing and assessment). The third use case shows inconsistent attestation responses (as indicated by the ✗ and ✓), which may be associated with a fault such as a Byzantine fault situation. The third use case corresponds to a use case in which a response from a given attestation server corresponds to an uncertain state, which may be caused by for example an attestation server or its link having a failure causing it to not send a response. When this is the case, if the remaining majority of attestation servers (which in this example is two) reply regarding trust state, a decision should not be taken regarding a final decision regarding the attestation as the final decision may not be made. The fourth use case shows a final result as untrusted, such as an attest failure, when at least a majority of the attestation results indicate false (in which case the client cannot or should not be trusted and placed outside of trusted computing pools).

[060] In some example embodiments, the verifier 290 may be included within the cloud-based system infrastructure (for example, included within a cloud management console), although the verifier may be implemented separately and/or remotely from the cloud-based system infrastructure. In some example embodiments, the verifier 290 may be included in at least one of the attestation servers. For example, the root attestation server may include the verifier 290. In some example embodiments, the verifier may be implemented to include a trusted system (for example, trust may be achieved by running verifier code in tamper-proof circuitry).

[061] FIG. 4A depicts an example of a process 400 for handling attestations in a hierarchal manner, in accordance with some example embodiments. The description of process 400 also refers to FIG. 2.

[062] At 402, a request for attesting a client may be received, in accordance with some example embodiments. For example, root attestation server 105A may receive a request to attest a client 160A in the cloud-based system 150. The request may include an identity of the client 160 and at least one measurement value for the client. For example, the identity may comprise a unique identifier for the client 160, although any other identifier may be used as well. The at least one measurement value may include at least one hash of the files at the client 160A.

[063] At 404, in response to receiving the request, a plurality of attestation requests may be sent to a plurality of attestation servers, in accordance with some example embodiments. For example, the root attestation server 105A may send a plurality of attestation requests to attestation servers 105B-D. The attestation requests may include the information received in the request, such as the identity and the at least one measurement.

[064] At 406, a plurality of attestation responses indicating whether the client is attested may be received from the plurality of attestation servers, in accordance with some

example embodiments. For example, the root attestation server 150A may receive from the plurality of attestation servers 2401-5B-D responses indicating whether the client 160A is attested.

[065] At 408, based on the received plurality of responses, a determination may be made regarding whether to attest the client, in accordance with some example embodiments. For example, root attestation server 105A may determine whether to attest the client 160A and indicate whether the client can be attested (for example, trusted), not attested (for example, not trusted), and/or uncertain (for example, cannot or should not attest to the client 160A). At 410, the root attestation server 105A may send the final attestation response regarding the client 160A.

[066] FIG. 4B depicts an example of a process 499 for handling attestations, in accordance with some example embodiments. The description of process 499 also refers to FIG. 3.

[067] At 482, an attestation request may be sent to a plurality of attestation servers, in accordance with some example embodiments. For example, verifier 390 may send an attestation request to attestation servers 105A-C. The attestation requests may each include an identity of the client 160 and at least one measurement value for the client. For example, the identity may comprise a unique identifier for the client 160, although any other identifier may be used as well. The at least one measurement value may include at least one hash of the files at the client 160A.

[068] At 484, the verifier may receive a plurality of responses indicative of whether the client is attested, in accordance with some example embodiments. At 486, based on the received plurality of responses received from the plurality of attestation servers, the verifier may determine whether to attest the client, in accordance with some example embodiments. For example, the verifier 290 may receive a plurality of attestation responses which may indicate

trusted, not trusted, or uncertain. The verifier may then use voting logic to determine whether to attest the client 160. For example, if a majority voting scheme is used, the verifier may use a majority of the attestation responses to determine whether to attest the client, although other voting schemes may be used as well. At 488, the verifier may provide the result (or indication thereof) of the attestation to another device, in accordance with some example embodiments. For example, the indication may be provided to a controller at the cloud-based system 150 to allow the client to perform work if the final attestation result for the client is attested as trusted or to inhibit the client if the final attestation result for the client is not trusted.

[069] FIG. 5 illustrates a block diagram of an apparatus 10, in accordance with some example embodiments. The apparatus 10 (or portions thereof) may be configured to provide a client device, such as client 160A, B, C, D, and/or the like. The apparatus may be implemented as any device including a server, a computer, a wireless device, a smart phone, a cell phone, a machine type communication device, a wireless sensor, an Internet of things (IoT) device, and/or any other processor and memory based device. Alternatively or additionally, the apparatus 10 may provide an attestation server including a root attestation server. Alternatively or additionally, the apparatus 10 may provide a verifier 290.

[070] The apparatus 10 may include at least one antenna 12 in communication with a transmitter 14 and a receiver 16. Alternatively transmit and receive antennas may be separate. The apparatus 10 may also include a processor 20 configured to provide signals to and receive signals from the transmitter and receiver, respectively, and to control the functioning of the apparatus. Processor 20 may be configured to control the functioning of the transmitter and receiver by effecting control signaling via electrical leads to the transmitter and receiver. Likewise, processor 20 may be configured to control other elements of apparatus 10 by effecting control signaling via electrical leads connecting processor 20 to the other elements, such as a display or a memory. The processor 20 may, for example, be embodied in a variety of ways

including circuitry, at least one processing core, one or more microprocessors with accompanying digital signal processor(s), one or more processor(s) without an accompanying digital signal processor, one or more coprocessors, one or more multi-core processors, one or more controllers, processing circuitry, one or more computers, various other processing elements including integrated circuits (for example, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), and/or the like), or some combination thereof. Accordingly, although illustrated in FIG. 5 as a single processor, in some example embodiments the processor 20 may comprise a plurality of processors or processing cores.

[071] Signals sent and received by the processor 20 may include signaling information in accordance with an air interface standard of an applicable cellular system, and/or any number of different wireline or wireless networking techniques, comprising but not limited to Wi-Fi, wireless local access network (WLAN) techniques, such as Institute of Electrical and Electronics Engineers (IEEE) 802.11, 802.16, and/or the like. In addition, these signals may include speech data, user generated data, user requested data, and/or the like.

[072] The apparatus 10 may be capable of operating with one or more air interface standards, communication protocols, modulation types, access types, and/or the like. For example, the apparatus 10 and/or a cellular modem therein may be capable of operating in accordance with various first generation (1G) communication protocols, second generation (2G or 2.5G) communication protocols, third-generation (3G) communication protocols, fourth-generation (4G) communication protocols, Internet Protocol Multimedia Subsystem (IMS) communication protocols (for example, session initiation protocol (SIP) and/or the like. For example, the apparatus 10 may be capable of operating in accordance with 2G wireless communication protocols IS-136, Time Division Multiple Access TDMA, Global System for Mobile communications, GSM, IS-95, Code Division Multiple Access, CDMA, and/or the like. In addition, for example, the apparatus 10 may be capable of operating in accordance with 2.5G

wireless communication protocols General Packet Radio Service (GPRS), Enhanced Data GSM Environment (EDGE), and/or the like. Further, for example, the apparatus 10 may be capable of operating in accordance with 3G wireless communication protocols, such as Universal Mobile Telecommunications System (UMTS), Code Division Multiple Access 2000 (CDMA2000), Wideband Code Division Multiple Access (WCDMA), Time Division-Synchronous Code Division Multiple Access (TD-SCDMA), and/or the like. The apparatus 10 may be additionally capable of operating in accordance with 3.9G wireless communication protocols, such as Long Term Evolution (LTE), Evolved Universal Terrestrial Radio Access Network (E-UTRAN), and/or the like. Additionally, for example, the apparatus 10 may be capable of operating in accordance with 4G wireless communication protocols, such as LTE Advanced, 5G, and/or the like as well as similar wireless communication protocols that may be subsequently developed.

[073] It is understood that the processor 20 may include circuitry for implementing audio/video and logic functions of apparatus 10. For example, the processor 20 may comprise a digital signal processor device, a microprocessor device, an analog-to-digital converter, a digital-to-analog converter, and/or the like. Control and signal processing functions of the apparatus 10 may be allocated between these devices according to their respective capabilities. The processor 20 may additionally comprise an internal voice coder (VC) 20a, an internal data modem (DM) 20b, and/or the like. Further, the processor 20 may include functionality to operate one or more software programs, which may be stored in memory. In general, processor 20 and stored software instructions may be configured to cause apparatus 10 to perform actions. For example, processor 20 may be capable of operating a connectivity program, such as a web browser. The connectivity program may allow the apparatus 10 to transmit and receive web content, such as location-based content, according to a protocol, such as wireless application protocol, WAP, hypertext transfer protocol, HTTP, and/or the like.

[074] Apparatus 10 may also comprise a user interface including, for example, an earphone or speaker 24, a ringer 22, a microphone 26, a display 28, a user input interface, and/or the like, which may be operationally coupled to the processor 20. The display 28 may, as noted above, include a touch sensitive display, where a user may touch and/or gesture to make selections, enter values, and/or the like. The processor 20 may also include user interface circuitry configured to control at least some functions of one or more elements of the user interface, such as the speaker 24, the ringer 22, the microphone 26, the display 28, and/or the like. The processor 20 and/or user interface circuitry comprising the processor 20 may be configured to control one or more functions of one or more elements of the user interface through computer program instructions, for example, software and/or firmware, stored on a memory accessible to the processor 20, for example, volatile memory 40, non-volatile memory 42, and/or the like. The apparatus 10 may include a battery for powering various circuits related to the mobile terminal, for example, a circuit to provide mechanical vibration as a detectable output. The user input interface may comprise devices allowing the apparatus 20 to receive data, such as a keypad 30 (which can be a virtual keyboard presented on display 28 or an externally coupled keyboard) and/or other input devices.

[075] As shown in FIG. 5, apparatus 10 may also include one or more mechanisms for sharing and/or obtaining data. For example, the apparatus 10 may include a short-range radio frequency (RF) transceiver and/or interrogator 64, so data may be shared with and/or obtained from electronic devices in accordance with RF techniques. The apparatus 10 may include other short-range transceivers, such as an infrared (IR) transceiver 66, a BluetoothTM (BT) transceiver 68 operating using BluetoothTM wireless technology, a wireless universal serial bus (USB) transceiver 70, a BluetoothTM Low Energy transceiver, a ZigBee transceiver, an ANT transceiver, a cellular device-to-device transceiver, a wireless local area link transceiver, and/or any other short-range radio technology. Apparatus 10 and, in particular, the short-range

transceiver may be capable of transmitting data to and/or receiving data from electronic devices within the proximity of the apparatus, such as within 10 meters, for example. The apparatus 10 including the Wi-Fi or wireless local area networking modem may also be capable of transmitting and/or receiving data from electronic devices according to various wireless networking techniques, including 6LoWpan, Wi-Fi, Wi-Fi low power, WLAN techniques such as IEEE 802.11 techniques, IEEE 802.15 techniques, IEEE 802.16 techniques, and/or the like.

[076] The apparatus 10 may comprise memory, such as a subscriber identity module (SIM) 38, a removable user identity module (R-UIM), an eUICC, an UICC, and/or the like, which may store information elements related to a mobile subscriber. In addition to the SIM, the apparatus 10 may include other removable and/or fixed memory. The apparatus 10 may include volatile memory 40 and/or non-volatile memory 42. For example, volatile memory 40 may include Random Access Memory (RAM) including dynamic and/or static RAM, on-chip or off-chip cache memory, and/or the like. Non-volatile memory 42, which may be embedded and/or removable, may include, for example, read-only memory, flash memory, magnetic storage devices, for example, hard disks, floppy disk drives, magnetic tape, optical disc drives and/or media, non-volatile random access memory (NVRAM), and/or the like. Like volatile memory 40, non-volatile memory 42 may include a cache area for temporary storage of data. At least part of the volatile and/or non-volatile memory may be embedded in processor 20. The memories may store one or more software programs, instructions, pieces of information, data, and/or the like which may be used by the apparatus for performing operations disclosed herein. The memories may comprise an identifier, such as an international mobile equipment identification (IMEI) code, capable of uniquely identifying apparatus 10. The memories may comprise an identifier, such as an international mobile equipment identification (IMEI) code, capable of uniquely identifying apparatus 10. In the example embodiment, the processor 20

may be configured using computer code stored at memory 40 and/or 42 to control and/or provide one or more aspects disclosed herein (see, for example, FIGs. 2 and 3).

[077] Some of the embodiments disclosed herein 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 40, the control apparatus 20, or electronic components, for example. In some 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 “computer-readable medium” may be any non-transitory media 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 or data processor circuitry, with examples depicted at FIG. 5, computer-readable medium may comprise a non-transitory computer-readable storage medium that may be any media that can contain or store the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer.

[078] Without in any way limiting the scope, interpretation, or application of the claims appearing below, a technical effect of one or more of the example embodiments disclosed herein is eliminating the dependency on single point of failure, allowing establishing distributed trust over multiple attestation servers, providing the means of detecting if some remote attestation server are compromised, and providing the technical effect of multiple remote attestation with options for hierarchical or inter-attesting attestation servers.

[079] The subject matter described herein may be embodied in systems, apparatus, methods, and/or articles depending on the desired configuration. For example, the base stations and user equipment (or one or more components therein) and/or the processes described herein can be implemented using one or more of the following: a processor executing program code, an application-specific integrated circuit (ASIC), a digital signal processor (DSP), an embedded

processor, a field programmable gate array (FPGA), and/or combinations thereof. These various implementations may include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device. These computer programs (also known as programs, software, software applications, applications, components, program code, or code) include machine instructions for a programmable processor, and may be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the term “computer-readable medium” refers to any computer program product, machine-readable medium, computer-readable storage medium, apparatus and/or device (for example, magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions. Similarly, systems are also described herein that may include a processor and a memory coupled to the processor. The memory may include one or more programs that cause the processor to perform one or more of the operations described herein.

[080] Although a few variations have been described in detail above, other modifications or additions are possible. In particular, further features and/or variations may be provided in addition to those set forth herein. Moreover, the implementations described above may be directed to various combinations and subcombinations of the disclosed features and/or combinations and subcombinations of several further features disclosed above. Other embodiments may be within the scope of the following claims.

[081] 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. Although various aspects of

some of the embodiments are set out in the independent claims, other aspects of some of the embodiments comprise other combinations of features from the described embodiments and/or the dependent claims with the features of the independent claims, and not solely the combinations explicitly set out in the claims. It is also noted herein that while the above describes example embodiments, these descriptions should not be viewed in a limiting sense. Rather, there are several variations and modifications that may be made without departing from the scope of some of the embodiments as defined in the appended claims. Other embodiments may be within the scope of the following claims. The term “based on” includes “based on at least.” The use of the phrase “such as” means “such as for example” unless otherwise indicated.

WHAT IS CLAIMED

1. An apparatus comprising:
 - at least one processor; and
 - at least one memory including program code which when executed configures the apparatus to at least:
 - receive a request to attest a client, the request including an identity of the client and at least one measurement value for the client;
 - send, in response to receiving the request, a plurality of attestation requests to a plurality of attestation servers, the attestation requests including the identity and the at least one measurement;
 - receive, from the plurality of attestation servers, a plurality of attestation responses indicating whether the client is attested;
 - determine, based on the received plurality of responses, whether to attest the client; and
 - send, based on the determination, the final attestation response to another node.
2. The apparatus of claim 1, wherein the apparatus represents a root attestation server.
3. The apparatus of any of claims 1-2, wherein the apparatus is further configured to at least attest each of the plurality of attestation servers, when the request is received.
4. The apparatus of any of claims 1-3, wherein the apparatus attests the plurality of attestation servers by at least comparing stored, reference measurement values for the

plurality of attestation servers to current, measurement values for the plurality of attestation servers.

5. The apparatus of any of claims 1-4, wherein the at least one measurement value comprises a hash value and/or a measurement value obtained from a non-volatile memory of a hardware security module.
6. The apparatus of any of claims 1-5, wherein the plurality of attestation requests sent to the plurality of attestation servers enable the plurality of attestation servers to attest the client based on at least a comparison of the at least one measurement to at least one stored, reference measurement for the client.
7. The apparatus of any of claims 1-6, wherein the apparatus determines the final attestation response based on voting logic.
8. The apparatus of any of claims 1-7, wherein the client comprises a host machine in a cloud-based system coupled to the internet.
9. The apparatus of any of claims 1-8, wherein the apparatus comprises redundant hardware.
10. The apparatus of any of claims 1-9, wherein the plurality of attestation servers include the apparatus.
11. The apparatus of any of claims 1-10, wherein the apparatus is selected from among the plurality of attestation servers to attest the client.

12. An apparatus comprising:

at least one processor; and

at least one memory including program code which when executed configures the apparatus to at least:

send, to a plurality of attestation servers, an attestation request, the attestation request including an identity of the client and at least one measurement value for the client;

receive from the plurality of attestation servers a plurality of attestation responses indicating whether the client is attested;

determine, based on the received plurality of responses, whether to attest the client, the determination based on voting logic; and

send, based on the determination, an indication to another device, the indication of whether the client is attested.

13. The apparatus of claim 12, wherein the plurality of attestation requests sent to the plurality of attestation servers enable the plurality of attestation servers to attest the client based on at least a comparison of the at least one measurement to at least one stored, reference measurement for the client.

14. The apparatus of any of claims 12-13, wherein the at least one measurement value comprises a hash value and/or a measurement value obtained from a non-volatile memory of a secure component.

15. The apparatus of any of claims 12-14, wherein the apparatus comprises redundant hardware.

16. A method comprising:

receiving, by a root attestation server, a request to attest a client, the request including an identity of the client and at least one measurement value for the client;

sending, by the root attestation server, in response to receiving the request, a plurality of attestation requests to a plurality of attestation servers, the attestation requests including the identity and the at least one measurement;

receiving, from the plurality of attestation servers, a plurality of attestation responses indicating whether the client is attested;

determining, based on the received plurality of responses, whether to attest the client; and

sending, by the root attestation server, the final attestation response to another node, the final attestation response based on the determining.

17. The method of claim 16, wherein the plurality of attestation servers includes the root attestation server.

18. The method of any of claims 16-17 further comprising:

attesting each of the plurality of attestation servers, when the request is received.

19. The method of any of claims 16-18, wherein the attesting comprises comparing stored, reference measurement values for the plurality of attestation servers to current, measurement values for the plurality of attestation servers.

20. The method of any of claims 16-19, wherein the at least one measurement value comprises a hash value and/or a measurement value obtained from a non-volatile memory of a secure component.
21. The method of any of claims 16-20, wherein the plurality of attestation requests sent to the plurality of attestation servers enable the plurality of attestation servers to attest the client based on at least a comparison of the at least one measurement to at least one stored, reference measurement for the client.
22. The method of any of claims 16-21, further comprising determining the final attestation response based on voting logic.
23. The method of any of claims 16-22, wherein the client comprises a host machine in a cloud-based system coupled to the internet.
24. The method of any of claims 16-23, wherein the root attestation server comprises redundant hardware.
25. The method of any of claims 16-24 further comprising:
selecting the root attestation server from among the plurality of attestation servers to attest the client.
26. A non-transitory computer-readable storage medium including program code which when executed by at least one processor causes operations comprising:
receiving, by a root attestation server, a request to attest a client, the request including an identity of the client and at least one measurement value for the client;

sending, by the root attestation server, in response to receiving the request, a plurality of attestation requests to a plurality of attestation servers, the attestation requests including the identity and the at least one measurement;

receiving, from the plurality of attestation servers, a plurality of attestation responses indicating whether the client is attested;

determining, based on the received plurality of responses, whether to attest the client; and

sending, by the root attestation server, the final attestation response to another node, the final attestation response based on the determining.

27. An apparatus comprising:

means for receiving a request to attest a client, the request including an identity of the client and at least one measurement value for the client;

means for sending, by the root attestation server, in response to receiving the request, a plurality of attestation requests to a plurality of attestation servers, the attestation requests including the identity and the at least one measurement;

means for receiving, from the plurality of attestation servers, a plurality of attestation responses indicating whether the client is attested;

means for determining, based on the received plurality of responses, whether to attest the client; and

sending, by the root attestation server, the final attestation response to another node, the final attestation response based on the determining.

28. The apparatus of claim 27 further comprising means for performing any of claims 17-25.

29. A method comprising:

sending an attestation request, the attestation request including an identity of the client and at least one measurement value for the client;

receiving from the plurality of attestation servers a plurality of attestation responses indicating whether the client is attested;

determining, based on the received plurality of responses, whether to attest the client, the determination based on voting logic; and

sending, based on the determination, an indication to another device, the indication of whether the client is attested.

30. The method of claim 29, wherein the plurality of attestation requests sent to the plurality of attestation servers enable the plurality of attestation servers to attest the client based on at least a comparison of the at least one measurement to at least one stored, reference measurement for the client.

31. The method of any of claims 29-30, wherein the at least one measurement value comprises a hash value and/or a measurement value obtained from a non-volatile memory of a secure module.

32. The method of any of claims 29-31, wherein a verifier sends the attestation request, receives the plurality of attestation responses, determines whether to attest, and sends the indication.

33. The method of claim 32, wherein the verifier comprises redundant hardware.

34. A non-transitory computer-readable storage medium including program code which when executed by at least one processor causes operations comprising:

 sending an attestation request, the attestation request including an identity of the client and at least one measurement value for the client;

 receiving from the plurality of attestation servers a plurality of attestation responses indicating whether the client is attested;

 determining, based on the received plurality of responses, whether to attest the client, the determination based on voting logic; and

 sending, based on the determination, an indication to another device, the indication of whether the client is attested.

35. An apparatus comprising:

 means for sending an attestation request, the attestation request including an identity of the client and at least one measurement value for the client;

 means for receiving from the plurality of attestation servers a plurality of attestation responses indicating whether the client is attested;

 means for determining, based on the received plurality of responses, whether to attest the client, the determination based on voting logic; and

 means for sending, based on the determination, an indication to another device, the indication of whether the client is attested.

36. The apparatus of claim 35 further comprising means for performing any of claims 31-33.

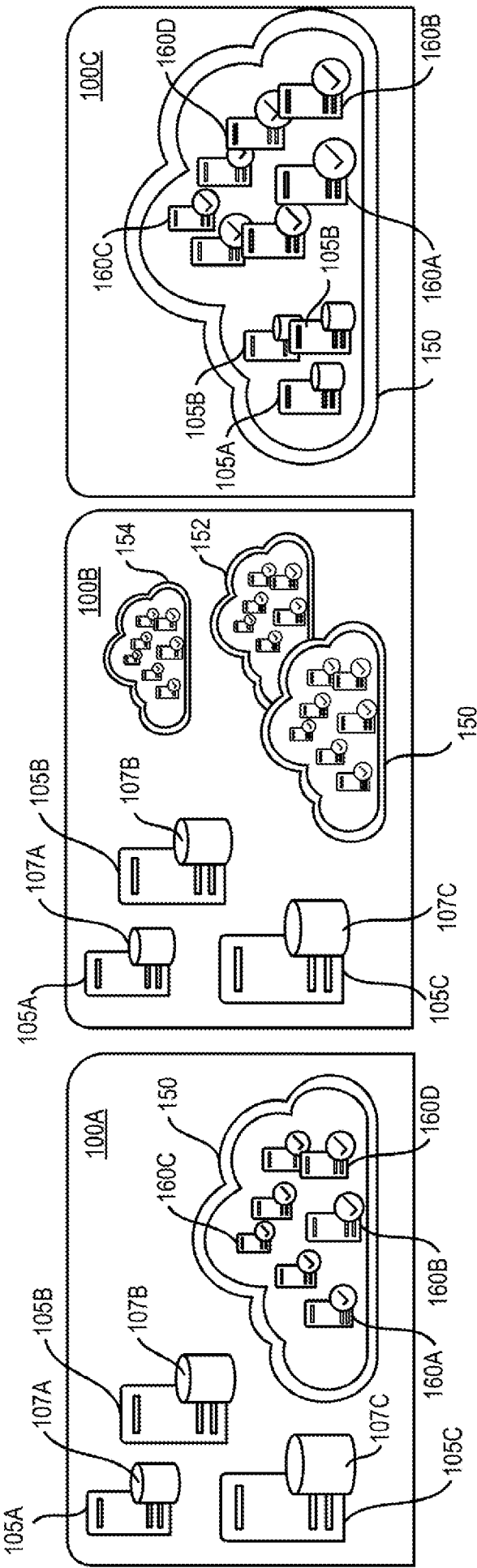


FIG. 1

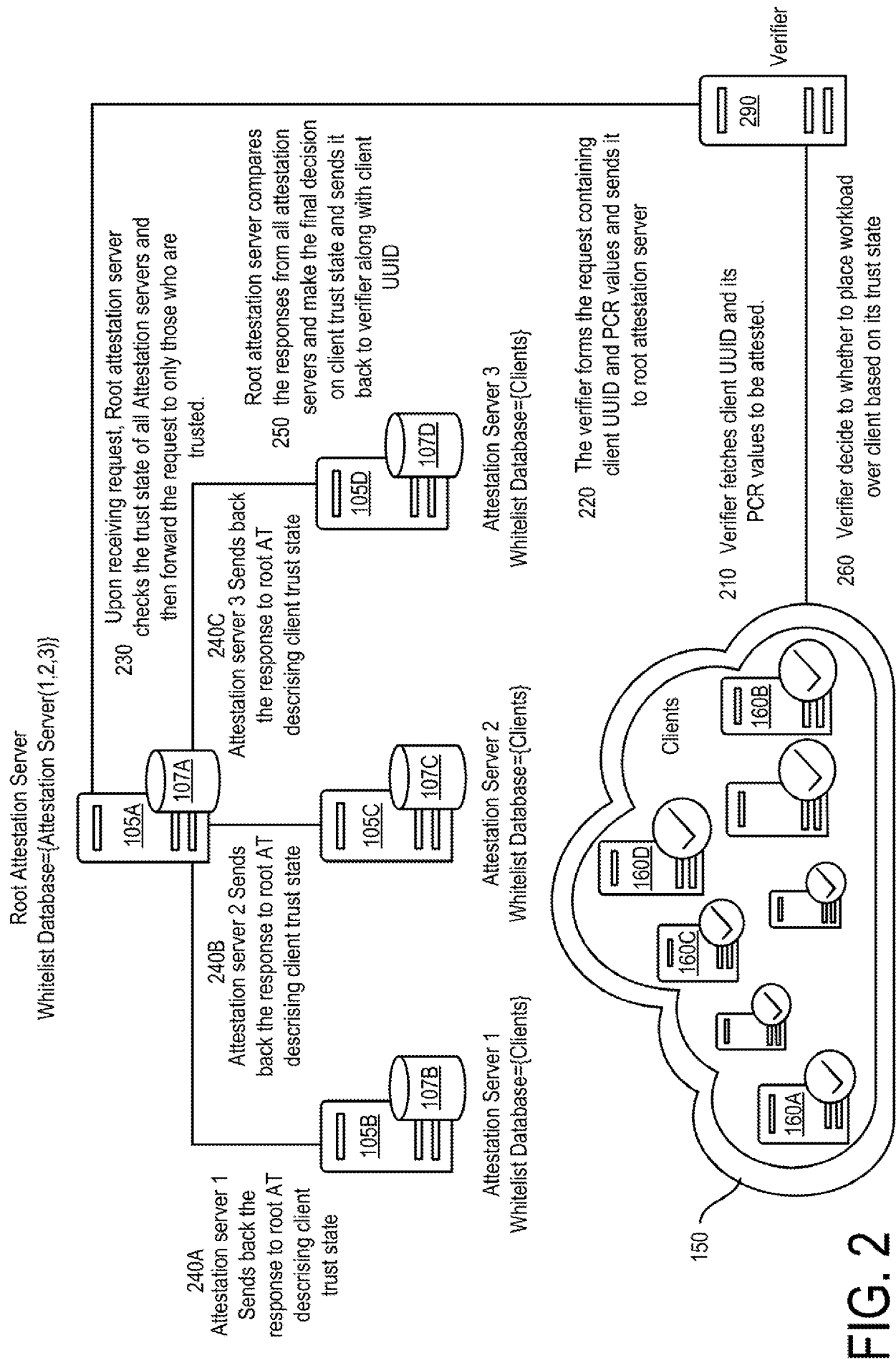


FIG. 2

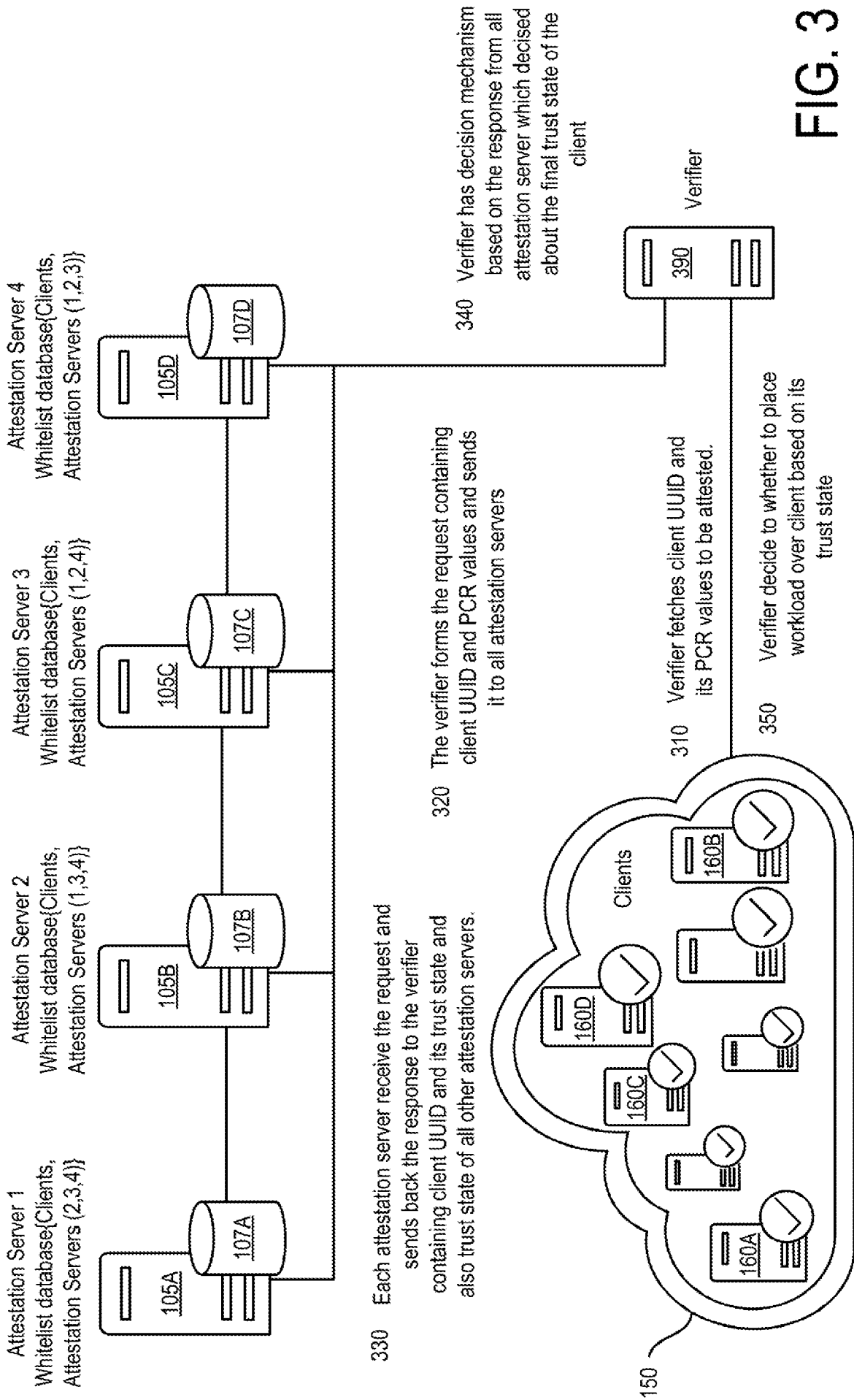
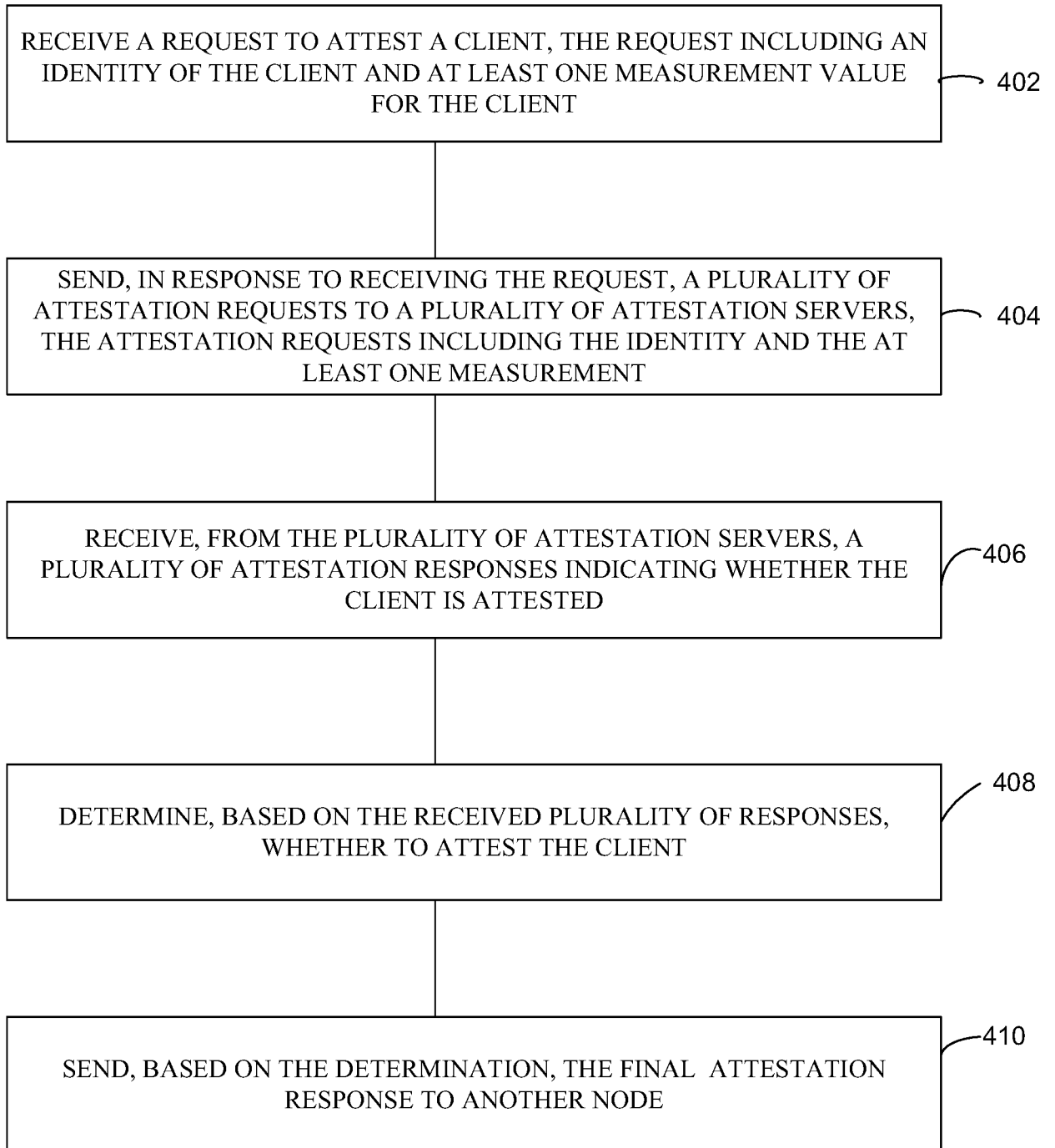
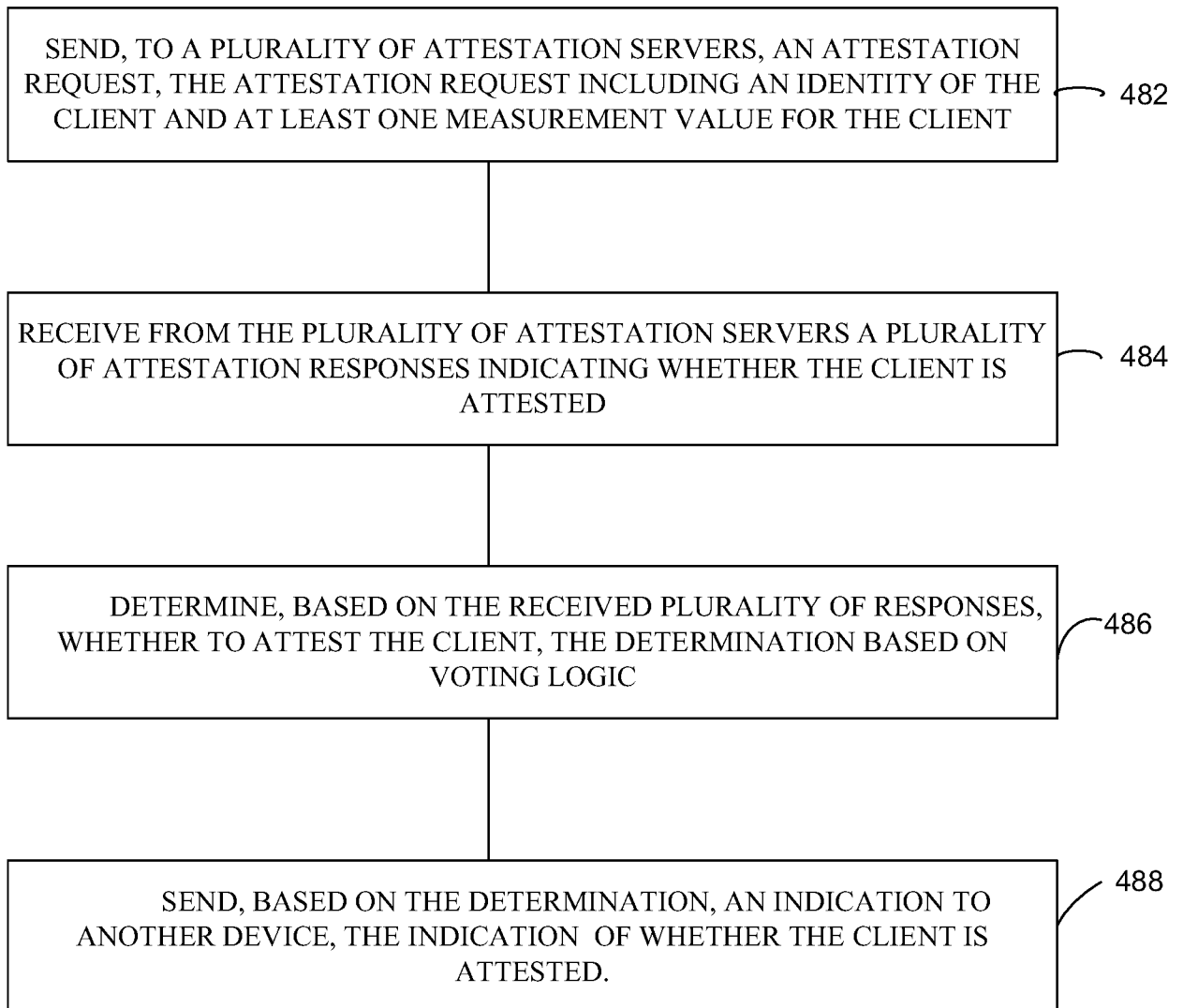


FIG. 3

400**FIG. 4A**

499**FIG. 4B**

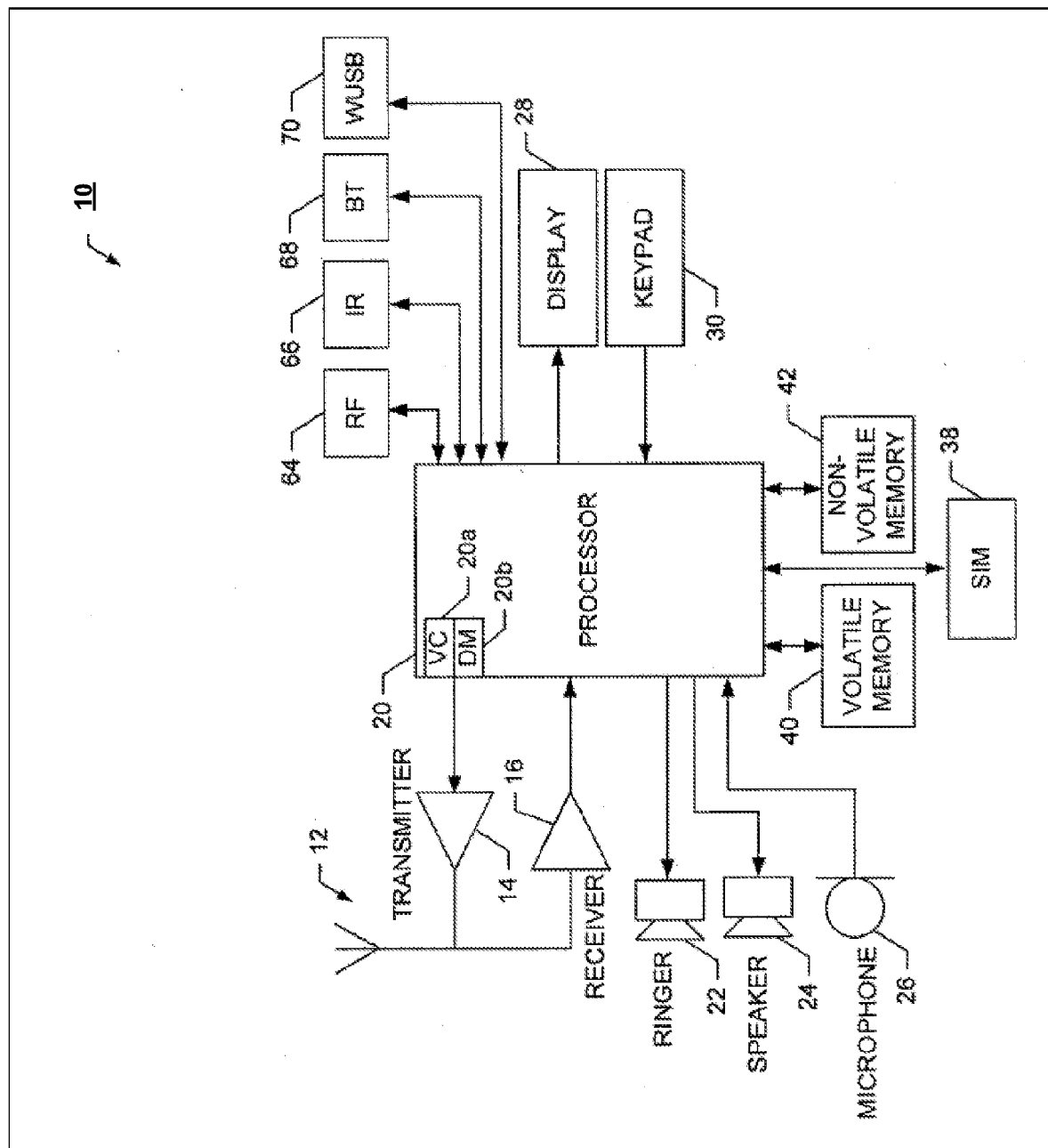


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2017/014423

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L9/08 H04L9/32 H04L29/06
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2016/366185 A1 (LEE RUBY B [US] ET AL) 15 December 2016 (2016-12-15) abstract paragraph [0012] - paragraph [0018] paragraph [0033] - paragraph [0057]; figures 1-3 paragraph [0063] - paragraph [0067] -----	1-35
A	LESLIE LAMPORT ET AL: "The Byzantine Generals Problem", ACM TRANSACTIONS ON PROGRAMMING LANGUAGE AND SYSTEMS, ACM, NEW YORK, NY, vol. 4, no. 3, 1 July 1982 (1982-07-01), pages 382-401, XP058212778, ISSN: 0164-0925, DOI: 10.1145/357172.357176 abstract page 383, line 5 - page 384, line 19 page 397, line 1 - page 400, line 33 -----	1-35



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

29 September 2017

Date of mailing of the international search report

11/10/2017

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

Spranger, Stephanie

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2017/014423

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2016366185	A1	15-12-2016	NONE
