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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
07 November 2024 (07.11.2024)

(10) International Publication Number
WO 2024/228045 A1

(51) International Patent Classification:
H04L 9/40 (2022.01)

(21) International Application Number:
PCT/IB2023/054674

(22) International Filing Date:
04 May 2023 (04.05.2023)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; SE-164 83 Stockholm (SE).

(72) Inventors: **OQAILY, Momen**; 1650, Apt.1505, De Maisonneuve Ouest, Montreal, Québec H3H 2P3 (CA). **WANG, Lingyu**; 1515 St. Catherine West, EV009.183, Montreal, Québec H3G 2W1 (CA). **NOUR, Boubakr**; 106-1122 Rue de Villeray, Montreal, Québec H2R 1J6 (CA). **JARRAYA, Yosr**; 6566, boulevard St-Michel, Montreal, Québec H1Y 2G1 (CA). **PUROHIT, Hinddeep**; 2125 rue Saint-Marc, Montreal, Québec H3H 2P1 (CA). **DEBBABI, Mourad**; 270, Rue Baffin, Dollard Des Ormeaux, Québec H9A 3E4 (CA).

(74) Agent: **WEISBERG, Alan M.**; Christopher & Weisberg, P.A., 1232 N. University Drive, Plantation, Florida 33322 (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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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.

(54) Title: RUNTIME BLACKBOX DETECTION OF NETWORK ATTACKS AGAINST SERVICE FUNCTION CHAINS USING VIRTUAL TRAILER

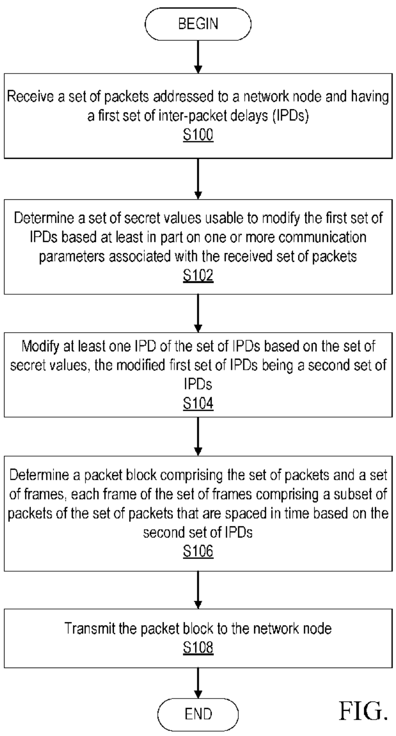

FIG. 3

(57) Abstract: A first agent node is described. The first agent node includes processing circuitry configured to receive a set of packets addressed to a network node and having a first set of inter-packet delays (IPDs), determine a set of secret values usable to modify the first set of IPDs based at least in part on one or more communication parameters associated with the received set of packets, and modify at least one IPD of the set of IPDs based on the set of secret values. A packet block including the set of packets and a set of frames is determined, where each frame of the set of frames includes a subset of packets of the set of packets that are spaced in time based on a second set of IPDs. Further, the packet block is transmitted to the network node.

WO 2024/228045 A1

[Continued on next page]

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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))*

— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

RUNTIME BLACKBOX DETECTION OF NETWORK ATTACKS AGAINST SERVICE FUNCTION CHAINS USING VIRTUAL TRAILER

TECHNICAL FIELD

The present disclosure relates to wireless communications, and in particular, to management of communication packets to detect network attacks.

BACKGROUND

NFV Background

Network Function Virtualization (NFV) has emerged as an innovative networking architecture that aims to overcome the limitations of traditional networks. NFV exploits sophisticated virtualization technologies to decouple Network Functions (NFs) from proprietary physical boxes and run them as virtualized software applications on commodity hardware. NFV may increase the availability and flexibility to design new services and provide on-demand service provisioning in a pay-as-you-go billing manner.

The Third Generation Partnership Project (3GPP) has developed and is developing standards for Fourth Generation (4G) (also referred to as Long Term Evolution (LTE)) and Fifth Generation (5G) (also referred to as New Radio (NR)) wireless communication systems. Such systems provide, among other features, broadband communication between network nodes, such as base stations, and mobile wireless devices (WD), as well as communication between network nodes and between WDs. The 3GPP is also developing standards for Sixth Generation (6G) wireless communication networks.

5G may use NFV to offer an enhanced user experience as well as cost savings for operators. In light of virtualization, softwarization, and deployment of 5G network functions within a third-party managed cloud (*i.e.,* hyper cloud providers), security attacks targeting the integrity of the traffic passing through the network functions of a given service (via a service function chaining (SFC)) can occur more easily (due to multi-tenancy and vulnerabilities leading to privilege escalation, etc.

Integrity Protection and Verification Background

Integrity protection has been recommended in one or more standards such as 3GPP standards, where it may be only recommended for network access server (NAS) signaling, radio resource connection (RRC) signaling, and user plane data. Some NFV challenges and best practices, attacks from within an NFV (e.g., related to security standard subversion attacks (ATT4)) have been discussed. An example may include an attack where the attacker can "leverage a known vulnerability (in dual elliptic curve deterministic random bit generator (dual deterministic random bit generator (DRBG)) that allows a potential back door may be used in an open secure session layer (OpenSSL) module. The attacker with knowledge of the trapdoor may decrypt, for example, transport layer security (TLS) encryption or VPN traffic, read all traffic, and modify the data as needed. That is, even though cryptographic-based integrity protection can help, it is not undeceivable.

Furthermore, integrity protection using cryptographic methods may add an overhead to the packet size and increase the load at the network functions (NFs). In addition, the VNF may be updated if a new cryptographic solution is deployed.

Other than integrity mechanisms defined in standards, several works including attempted solutions such as FlowTags, vSFC, FlowCloack, AuditBox, and SFC-Checker have been described. These solutions can be broadly categorized into cryptographic and non-crypto-based solutions: (i) crypto-based solutions that leverage cryptographic techniques to validate the correct traversal of packets in predefined forwarding paths through forwarding path verification protocols (FPVP), and (ii) non-crypto-based solutions that either modify the VNF implementation or change the packet structure by adding fields.

For instance, FlowTags, vSFC, FlowCloack, and cSFC may perform the verification by adding a tag to the end of the packet and modifying the VNF implementation to add a verification layer that checks the added tags. The verification layer is separated from VNF logic and deployed on trusted execution environments or enclaves to preserve data confidentiality and VNF software integrity. Further, a protocol that performs packet modification by adding different fields to the packet header and a message authentication code (MAC) to the packet trailer has been described. In addition, other solutions use statistical and formal verification. In SFC-Checker, a static analysis-based framework that checks the correct forwarding behavior of dynamic and stateful forwarding paths is described. A snapshot of the network state with the flow tables and the VNFs model to create a stateful forwarding graph using a Finite State Machine may be used. In addition, a formal model that allows network administrators to specify forwarding policies and a broad spectrum of anomalies may be used.

Watermarking Background

Watermarking techniques have been used to link network flows (*i.e.,* flow correlation) to detect stepping-stone attacks and to compromise anonymity systems. More specifically, a watermarker may modify communication patterns of an observed network flow, e.g., so that it modulates an L-bits watermark tag into that flow. The L-bits watermarks convey some information about the carrying flow, *e.g.,* its network origin, hence it might have different values across different flows. A watermarked flow passes through a noisy network, *e.g.,* the Internet, before it is intercepted by a fingerprint extractor that, then, tries to extract its L-bits watermark tag.

The existing designs for active traffic analysis are referred to as flow watermarks. A flow watermarking system is composed of a watermarker that tags network flows by perturbing their patterns, and intercepted flows may be used to identify those carrying the watermark perturbation. The information carried by the watermarks can answer different questions such as: i) which specific watermarker, out of all watermarker, tagged this flow? ii) in which region of the network has this flow been tagged? iii) which specific flows are related to the observed flow?

In addition, a first non-blind (i.e., information about the watermarks is shared with the extractor) approach for flow watermarking may be used and may comprise inserting delays smaller than previous blind watermarks. The flow watermarking problem may be formulated, and its application scenarios contrasted with flow watermarking. Furter, a class of flow fingerprints that uses a non-blind architecture may be used, and watermark-based correlation scheme may be used.

However, crypto-based solutions impose significant processing overhead burden resulting from encryption to generate message authentication codes, key management and sharing, packet modification, and acknowledgment packet sharing. Non-crypto solutions need: (i) access to the underlying cloud infrastructure, which is not always practical, (ii) change of packets by adding fields, which add extra overhead and impact the quality of service, or (iii) require implementation amendments in the VNF logic, which is not always feasible/practical.

SUMMARY

Some embodiments advantageously provide methods, systems, and apparatuses for detection of network attacks. Some other embodiments provide a runtime blackbox detection of network attacks against service function chains using virtual trailer.

Conventional technology does not provide (or take into account) the following:

- *Verification from tenant view*: The most common drawback among conventional technologies is that they have been designed as a whitebox-based solution and assume that the tenant (i) trusts the cloud provider and (ii) has access to the underlying cloud infrastructure for deployment. However, when tenants mistrust the cloud provider or at least seek some sort of transparency to audit their own deployments, they should not set their verification mechanisms by relying on network services states claimed by the cloud provider. Verifying network services from the tenants' side may be mandatory to trust.

- *Blackbox integrity verification*: Conventional technologies require (i) access to the underlying cloud infrastructure and (ii) perform deployment and implementation amendments to the VNF. However, in practical terms, the cloud provider does not give the tenants access to the underlying deployment as it will increase the attack surface.
- *Design complexity and computational cost:* conventional crypto-based technology require extra computation overhead (*e.g.,* increases min-sized packets by 3× for a 4-VNF chain, and incurs 1.69× overhead for a strong attack model) such as to perform encryption/decryption operations as well as key management. Conventional non-crypto-based technology requires extra time and overhead to either change content or add extra fields to the packets. Both technologies have added overhead and design complexity that detrimentally impacts overall network performance.
- *Offline and discrete auditing*: Conventional technologies include offline-based auditing, such that a snapshot of the deployment is taken either before or during deployment to perform auditing using forwarding graphs or formal methods. Such methodology fails to match up with the high dynamicity nature of NFV, where anomalies can exist and disappear before being detected. Further, discrete auditing performed as continuous auditing may result in detrimental computational costs.

Some embodiments provide an efficient integrity verification solution. In some other embodiments, the integrity verification may be deployed independently without the intervention of the service provider to satisfy the tenant needs and to build trust with the NFV technology. The integrity verification may not overwhelm the deployed network services.

In some embodiments, a watermarking process (e.g., that uses interpacket delay as a side channel) is used. In some other embodiments, whether the modification to the watermark is linked to an attack against the service chain or other factors (*e.g.,* routing issues) is determined. The type of attack (whether it is traffic injection, flow replay, or one/multiple VNF skipping), e.g., those related to the service chain, may be identified. In some embodiments, multi-hop aspects are considered and/or used such as to tackle end-to-end service attack detection.

In some other embodiments, a system (e.g., named ChainPatrol) and/or nodes of the system may be configured to verifies the integrity of network traffic between network nodes (e.g., virtual network functions (VNFs)) along a service function chain and detect/identify the types of attacks occurring between network nodes (e.g., VNF in an SFC) as well as occurring from end-to-end of a service chain. The system (e.g., ChainPatrol) may detect and/or identify several attacks including packet injection, dropping, and duplication; traffic flow injection, dropping, duplication; and one or several network functions skipping.

In some embodiments, one or more steps and/or features (e.g., a virtual trailer) are provided, which comprise determining a set of secret values. The secret values may be embedded in (e.g., be a characteristic of) blocks of packets. For example, inter-packet delay (IPD) of the packets of the packet blocks may be determined/modified. In some other embodiments, a packet block may comprise one or more (e.g., four) frames, and each frame has a specific usage in attack type detection and identification. At run-time, a first agent node (e.g., the ChainPatrol agent such as one agent located near a first network node such as a VNF) intercepts egress network traffic from the first network node (e.g., sender VNF (*i.e.,* source)). Secret values may be embedded or made a characteristic of the packets by modifying the IPDs (e.g., by using watermarking) before forwarding them towards a second network node (e.g., receiver VNF, destination VNF).

In some embodiments, at the second network node (e.g., the receiver VNF), a second agent node (e.g., the second ChainPatrol agent) may monitor (e.g., passively monitor) the received packets IPD and perform the first level of verification (e.g., detect/identify block-related attacks and traffic flow-based attacks) by reconstructing the virtual trailer and matching it with an expected one. A second-level verification (e.g., detect/identify packet-related attacks and NF skipping attacks) may be performed at an orchestrator node (e.g., the ChainPatrol orchestrator) which may also manage the agent nodes and/or perform service chain level integrity verification.

One or more embodiments provide one or more of the following benefits:

- A distributed system that enables the detection at runtime of network attacks against network nodes (e.g., service function chains in NFV) and enables the identification of their types without relying on cryptographic tools and without accessing data from the underlying infrastructure provider (e.g., Blackbox approach). The distributed system may provide:

  - Detection and identification that comprises embedding and extracting using one or more steps (e.g., virtual trailer);
  - Detection and identification that are executed at two levels: level one is close to the second network node (e.g., the receiver VNF), and level two is at the orchestrator level (i.e., orchestrator node).
  - Detection of network attacks between pairs of consecutive VNFs and from end-to-end across a service function chain.

In some embodiments, specific information that captures a set of (traffic-related) attributes may be provided (e.g., using virtual trailer). This virtual trailer may be embedded to enable the detection and unique identification of each occurring attack from several possible attack types without changing/editing packets/content.

Some embodiments further provide the following advantages:

- Lightweight and non-crypto-based solution to detect attacks related to chaining in NFV (*e.g.,* 5G service-based architecture (SBA));
- Continuous and runtime verification of NF chaining integrity;
- Identification of attack types and breaches (*e.g.,* packet/flow injection and duplication);
- Strengthening of the security of services provided by tenants without the need to involve the cloud provider in the auditing process or accessing the lower-level data or infrastructure; and
- Overcoming of challenges associated with partial visibility over network services;
- Software/hardware agnostic solution(s) that can be implemented as a plug-and-play.

According to one aspect, a first agent node is described. The first agent node includes processing circuitry configured to receive a set of packets addressed to a network node and having a first set of inter-packet delays (IPDs), determine a set of secret values usable to modify the first set of IPDs based at least in part on one or more communication parameters associated with the received set of packets, and modify at least one IPD of the set of IPDs based on the set of secret values. The modified first set of IPDs is a second set of IPDs. A packet block including the set of packets and a set of frames is determined, where each frame of the set of frames includes a subset of packets of the set of packets that are spaced in time based on a second set of IPDs. Further, the packet block is transmitted to the network node.

In some embodiments, the processing circuitry is further configured to one or more of measure each IPD of the first set of IPDs; perform a watermarking based on the measurement of each IPD, a watermarking amplitude, and the packet block; and cause transmission of a record to the orchestrator node, the record being based on the watermarking and to be stored in an IPD registrar of the orchestrator node.

In some other embodiments, performing the watermarking includes determining a first seed value based on an initial seed value and network connection information corresponding to a first network connection; determining a first secret value, a second secret value, a third secret value, and a fourth secret value of the set of secret values for the first network connection based on the first seed value; determining another packet block is associated with the first network connection; and incrementing the first secret value for the other packet block.

In some embodiments, performing the watermarking further comprises determining a second seed value based on the initial seed value and network connection information corresponding to a second network connection and determining the first secret value, the second secret value, the third secret value, and the fourth secret value of the set of secret values for the second network connection based on the second seed value. The first secret value for the first network connection and the first secret value for the second network connection are different. The second secret value of the second network connection is incremented based on the second secret value of the first network connection.

In some other embodiments, the first secret value is a sequence number of the packet block, the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier.

In some embodiments, the first secret value corresponds to a first frame, the second secret value corresponds to a second frame, the third secret value corresponds to a third frame, and the fourth secret value corresponds to a fourth frame of the set of frames.

In some other embodiments, each one of the first, second, third, and fourth frames include one or both of a first packet and a second packet; and a third packet and a fourth packet.

In some embodiments, performing the watermarking further includes determining a first IPD, a second IPD, and a third IPD of the second set of IPDs. The first IPD defines a first space in time between the first packet and the second packet, the second IPD defines a second space in time between the second packet and the third packet, and the third IPD defines a third space in time between the third packet and the fourth packet.

In some other embodiments, performing the watermarking further comprises converting each one of the first, second, third, and fourth secret values to a bit value having a plurality of bits. Each bit of the plurality of bits is transformed to a transformed value based on the value of the bit, where the transformed value is +1 or -1. The first, second, third, and fourth IPDs are based on a watermarking amplitude, corresponding IPDs of the first set of IPDs and the transformed value.

In some embodiments, one or more of the set of packets are received from a first virtual network function comprised in a first network node, where the set of packets is addressed to a second network node; the first agent node comprises a first chain patrol agent; the second network node comprises a second virtual network function configured to receive the set of packets; the set of packets are to be obtained by a second agent node that comprises a second chain patrol agent; and the first agent node is configured to communicate with an orchestrator node that comprises a chain patrol orchestrator.

According to another aspect, a method in a first agent node is described. The method includes receiving a set of packets addressed to a network node and having a first set of inter-packet delays (IPDs); determining a set of secret values usable to modify the first set of IPDs based at least in part on one or more communication parameters associated with the received set of packets; modifying at least one IPD of the set of IPDs based on the set of secret values, the modified first set of IPDs being a second set of IPDs; determining a packet block comprising the set of packets and a set of frames, where each frame of the set of frames includes a subset of packets of the set of packets that are spaced in time based on the second set of IPDs; and transmitting the packet block to the network node.

In some embodiments, the method further comprises one or more of measuring each IPD of the first set of IPDs; performing a watermarking based on the measurement of each IPD, a watermarking amplitude, and the packet block; and transmitting a record to the orchestrator node, where the record is based on the watermarking and to be stored in an IPD registrar of the orchestrator node.

In some other embodiments, performing the watermarking includes determining a first seed value based on an initial seed value and network connection information corresponding to a first network connection; determining a first secret value, a second secret value, a third secret value, and a fourth secret value of the set of secret values for the first network connection based on the first seed value; determining another packet block is associated with the first network connection; and incrementing the first secret value for the other packet block.

In some embodiments, performing the watermarking further includes determining a second seed value based on the initial seed value and network connection information corresponding to a second network connection and determining the first secret value, the second secret value, the third secret value, and the fourth secret value of the set of secret values for the second network connection based on the second seed value. The first secret value for the first network connection and the first secret value for the second network connection are different, and the second secret value of the second network connection is incremented based on the second secret value of the first network connection.

In some other embodiments, the first secret value is a sequence number of the packet block, the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier.

In some embodiments, the first secret value corresponds to a first frame, the second secret value corresponds to a second frame, the third secret value corresponds to a third frame, and the fourth secret value corresponds to a fourth frame of the set of frames.

In some other embodiments, each one of the first, second, third, and fourth frames comprise one or both of a first packet and a second packet; and a third packet and a fourth packet.

In some embodiments, performing the watermarking further includes determining a first IPD, a second IPD, and a third IPD of the second set of IPDs. The first IPD defines a first space in time between the first packet and the second packet, the second IPD defines a second space in time between the second packet and the third packet, the third IPD defines a third space in time between the third packet and the fourth packet.

In some other embodiments, performing the watermarking further includes converting each one of the first, second, third, and fourth secret values to a bit value having a plurality of bits, where each bit of the plurality of bits is transformed to a transformed value based on the value of the bit. The transformed value is +1 or -1, and the first, second, third, and fourth IPDs are based on a watermarking amplitude a corresponding IPD of the first set of IPDs and the transformed value.

In some embodiments, one or more of the set of packets are received from a first virtual network function comprised in a first network node, where the set of packets being addressed to a second network node; the first agent node comprises a first chain patrol agent; the second network node comprises a second virtual network function configured to receive the set of packets; the set of packets are to be obtained by a second agent node that comprises a second chain patrol agent; and the first agent node is configured to communicate with an orchestrator node that comprises a chain patrol orchestrator.

According to an aspect, a second agent node configured to communicate with an orchestrator node is described. The second agent node includes processing circuitry configured to obtain a packet block comprising the set of packets and a set of frames, where each frame of the set of frames includes a subset of packets of the set of packets, and the subset of packets is spaced in time based on a second set of inter-packet delays (IPDs). A set of secret values is extracted from the packet block based on the second set of IPDs, a verification of the packet block is performed based on the secret values and a record stored at an IPD registrar of the orchestrator node. The verification verifies whether the packet block is associated with a network attack.

In some embodiments, the processing circuitry is further configured to one or more of monitor the set of packets; construct the packet block; measure each IPD of the second set of IPDs; read the record stored in the IPD Registrar; store each IPD locally in a local audit table; perform the verification of the packet block at a block level and a flow level; and trigger the orchestrator node to store information from the local audit table in an orchestrator audit table and store each measured IPD of the second set of IPDs.

In some other embodiments, extracting the set of secret values includes determining a set of expected secrets values using an expected packet block corresponding to a watermarking process, where the set of expected secret values is generated for each record of a connection and each packet block record shared by a first agent node.

In some embodiments, extracting the set of secret values includes extracting each secret value of the set of secret values from the packet block to form a computed packet block and forming the computed packet block.

In some other embodiments, performing the verification of the packet block includes comparing the computed packet block with the expected packet block and determining that the packet block matches the expected packet block based on the comparison.

In some embodiments, performing the verification of the packet block includes one or more of determining that the packet bock is not associated with the network attack when the packet block matches the expected packet block; updating at least on audit table when one packet block is found and each frame of the one packet block is verified; determining that the packet bock is associated with the network attack when at least one secret value of the set of secret values from the packet block does not match a corresponding expected secret value, where the determination is based on at least a corresponding IPD; marking the packet block as an unverified packet block; and cause transmission of information associated with the unverified packet block to the orchestrator to perform another verification of the unverified packet block.

In some other embodiments, one or more of the second set of IPDs are based on a first seed value, where the first seed value is based on an initial seed value and network connection information corresponding to a first network connection; the set of secret values comprise a first secret value, a second secret value, a third secret value, and a fourth secret value for the first network connection based on the first seed value; the second set of IPDs is further based on a second seed value, where the second seed value is based on the initial seed value and network connection information corresponding to a second network connection. The first secret value for the first network connection and the first secret value for the second network connection are different, and the second secret value of the second network connection are incremented based on the second secret value of the first network connection.

In some embodiments, one or more of the first secret value is a sequence number of the packet block, where the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier; the first secret value corresponds to a first frame, where the second secret value corresponds to a second frame, the third secret value corresponds to a third frame, and the fourth secret value corresponds to a fourth frame of the set of frames; each one of the first, second, third, and fourth frames comprise one or both of a first packet, a second packet; and a third packet, and a fourth packet.

In some other embodiments, one or both of the second set of IPDs includes a first IPD, a second IPD, and a third IPD of the second set of IPDs. The first IPD defines a first space in time between the first packet and the second packet, the second IPD defines a second space in time between the second packet and the third packet, the third IPD defines a third space in time between the third packet and the fourth packet; and each one of the first, second, third, and fourth secret values is based on a bit value having a plurality of bits. Each bit of the plurality of bits being transformed to a transformed value based on the value of the bit, and the transformed value is +1 or -1. The first, second, third, and fourth IPDs are based on a watermarking amplitude a corresponding IPD of a first set of IPDs and the transformed value.

In some embodiments, one or more of the set of packets are transmitted by a first virtual network function comprised in a first network node, where the set of packets is addressed to a second network node; the second network node includes a second virtual network function configured to receive the set of packets; the second agent node comprises a second chain patrol agent; and the orchestrator node comprises a chain patrol orchestrator.

According to another aspect, a method in a second agent node configured to communicate with an orchestrator node is described. The method includes obtaining a packet block comprising the set of packets and a set of frames, where each frame of the set of frames includes a subset of packets of the set of packets, and the subset of packets is spaced in time based on a second set of inter-packet delays (IPDs); extracting a set of secret values from the packet block based on the second set of IPDs; and performing a verification of the packet block based on the secret values and a record stored at an IPD registrar of the orchestrator node, where the verification verifies whether the packet block is associated with a network attack.

In some embodiments, the method further includes monitoring the set of packets; constructing the packet block; measuring each IPD of the second set of IPDs; reading the record stored in the IPD Registrar; storing each IPD locally in a local audit table; performing the verification of the packet block at a block level and a flow level; and triggering the orchestrator node to store information from the local audit table in an orchestrator audit table and store each measured IPD of the second set of IPDs.

In some other embodiments, extracting the set of secret values includes determining an expected packet block based on watermarking process and generating a set of expected secrets values using the expected packet block, where the set of expected secret values is generated for each record of a connection and each packet block record shared by a first agent node.

In some embodiments, extracting the set of secret values includes extracting each secret value of the set of secret values from the packet block to form a computed packet block and forming the computed packet block.

In some other embodiments, performing the verification of the packet block includes comparing the computed packet block with the expected packet block and determining that the packet block matches the expected packet block based on the comparison.

In some embodiments, performing the verification of the packet block includes one or more of determining that the packet bock is not associated with the network attack when the packet block matches the expected packet block; updating at least on audit table when one packet block is found and each frame of the one packet block is verified; determining that the packet bock is associated with the network attack when at least one secret value of the set of secret values from the packet block does not match a corresponding expected secret value, where the determination is based on at least a corresponding IPD; marking the packet block as an unverified packet block; and transmitting information associated with the unverified packet block to the orchestrator to perform another verification of the unverified packet block.

In some other embodiments, one or more of the second set of IPDs are based on a first seed value, the first seed value being based on an initial seed value and network connection information corresponding to a first network connection; the set of secret values comprise a first secret value, a second secret value, a third secret value, and a fourth secret value for the first network connection based on the first seed value; the second set of IPDs is further based on a second seed value, where the second seed value is based on the initial seed value and network connection information corresponding to a second network connection; the first secret value for the first network connection and the first secret value for the second network connection are different, where the second secret value of the second network connection are incremented based on the second secret value of the first network connection.

In some embodiments, one or more of the first secret value is a sequence number of the packet block, where the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier; the first secret value corresponds to a first frame, where the second secret value corresponds to a second frame, the third secret value corresponds to a third frame, and the fourth secret value corresponds to a fourth frame of the set of frames; each one of the first, second, third, and fourth frames comprise a first packet, a second packet, a third packet, and a fourth packet.

In some other embodiments, one or both of the second set of IPDs includes a first IPD, a second IPD, and a third IPD of the second set of IPDs, where the first IPD defines a first space in time between the first packet and the second packet, the second IPD defines a second space in time between the second packet and the third packet, the third IPD defines a third space in time between the third packet and the fourth packet; and each one of the first, second, third, and fourth secret values is based on a bit value having a plurality of bits. Each bit of the plurality of bits is transformed to a transformed value based on the value of the bit, the transformed value is +1 or -1, and the first, second, third, and fourth IPDs are based on a watermarking amplitude a corresponding IPD of a first set of IPDs and the transformed value.

In some embodiments, one or more of the set of packets are transmitted by a first virtual network function comprised in a first network node, where the set of packets is addressed to a second network node; the second network node includes a second virtual network function configured to receive the set of packets; the second agent node includes a second chain patrol agent; and the orchestrator node includes a chain patrol orchestrator.

According to an aspect, an orchestrator node configured to communicate with a first agent node and a second agent node is described. The orchestrator node includes processing circuitry configured to perform a second verification of a packet block based on a first verification of the packet block performed by the second agent node and determine whether the packet block is associated with a network attack.

In some embodiments, the processing circuitry is further configured to receive, from the second agent node, information associated with an unverified packet block, where the second verification is performed based on the information.

In some other embodiments, the orchestrator node further includes an inter-packet delay (IPD) registrar, and the processing circuitry is further configured to one or more of receive a record from the first agent node based on a watermarking process performed by the first agent node; store the record in the IPD registrar of the orchestrator node; store information from a local audit table associated with the second agent node in an orchestrator audit table; and store one or more IPDs measured by the second agent node.

In some embodiments, the processing circuitry is further configured to instantiate the first and second agent nodes and cause transmission of an initial seed at least to the first agent node for modifying at least one IPD.

In some other embodiments, the processing circuitry is further configured to receive a first set of IPDs, one or more network connection numbers before a watermarking process is performed and cause transmission of the first set of IPDs to the second agent node for extraction of watermarking information.

In some embodiments, the processing circuitry is further configured to receive one or more IPDs from the second agent node, where the one or more IPDs are associated with an unverified packet block and usable to perform the second verification, and retrieve audit table records.

In some other embodiments, the processing circuitry is further configured to determine whether the network attack is a packet level attack based on the one or more IPDs and audit table records.

In some embodiments, determining whether the network attack is a packet level attack includes using a window of fixed size that moves across the one or more IPDs and recovering information about whether the one or more IPDs correspond to watermarked message that has not been found.

In some other embodiments, the processing circuitry is further configured to determine whether the network attack is associated with a skipping a network node.

In some embodiments, one or more of the first agent node includes a first chain patrol agent; the second agent node includes a second chain patrol agent; and the orchestrator node includes a chain patrol orchestrator.

According to another aspect, a method in an orchestrator node configured to communicate with a first agent node and a second agent node is described. The method includes performing a second verification of a packet block based on a first verification of the packet block performed by the second agent node and determining whether the packet block is associated with a network attack.

In some embodiments, the method further includes receiving, from the second agent node, information associated with an unverified packet block, where the second verification is performed based on the information.

In some other embodiments, the orchestrator node further includes an inter-packet delay (IPD) registrar, and the method further includes one or more of receiving a record from the first agent node based on a watermarking process performed by the first agent node; storing the record in the IPD registrar of the orchestrator node; storing information from a local audit table associated with the second agent node in an orchestrator audit table; and storing one or more IPDs measured by the second agent node.

In some embodiments, the method further includes instantiating the first and second agent nodes and transmitting an initial seed at least to the first agent node for modifying at least one IPD.

In some other embodiments, the method further includes receiving a first set of IPDs, one or more network connection numbers before a watermarking process is performed and transmitting the first set of IPDs to the second agent node for extraction of watermarking information.

In some embodiments, the method further includes receiving one or more IPDs from the second agent node, where the one or more IPDs are associated with an unverified packet block and usable to perform the second verification, and retrieving audit table records.

In some other embodiments, the method further includes determining whether the network attack is a packet level attack based on the one or more IPDs and audit table records.

In some embodiments, determining whether the network attack is a packet level attack includes using a window of fixed size that moves across the one or more IPDs; and recovering information about whether the one or more IPDs correspond to watermarked message that has not been found.

In some other embodiments, the method further includes determining whether the network attack is associated with a skipping of a network node.

In some embodiments, one or more of the first agent node includes a first chain patrol agent, the second agent node includes a second chain patrol agent, and the orchestrator node includes a chain patrol orchestrator.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the present embodiments, and the attendant advantages and features thereof, will be more readily understood by reference to the following detailed description when considered in conjunction with the accompanying drawings wherein:

FIG. 1 is a schematic diagram of an example network architecture illustrating a communication system according to the principles in the present disclosure;

FIG. 2 is a block diagram of an example orchestrator node, an example agent node, and example network node according to some embodiments of the present disclosure;

FIG. 3 is a flowchart of an example process in a first agent node according to some embodiments of the present;

FIG. 4 is a flowchart of an example process in a second agent node according to some embodiments of the present;

FIG. 5 is a flowchart of an example process in an orchestrator node according to some embodiments of the present;

FIG. 6 shows an overview of an example system comprising an orchestrator node, a first network node, a second network node, a first agent node, and a second agent node according to some embodiments of the present;

FIG. 7 shows an example plurality of blocks according to some embodiments of the present;

FIG. 8 shows an example process (e.g., in an agent node) according to some embodiments of the present disclosure;

FIG. 9 shows another example process (e.g., in an agent node) according to some embodiments of the present disclosure;

FIG. 10 shows example process (e.g., a watermarking process in an agent node) according to some embodiments of the present disclosure;

FIG. 11 shows another example process (e.g., in an agent node) according to some embodiments of the present disclosure;

FIG. 12 shows an example audit table according to some embodiments of the present disclosure;

FIG. 13 shows an example process (e.g., an extracting process in an agent node) according to some embodiments of the present disclosure;

FIG. 14 shows another example audit table according to some embodiments of the present disclosure;

FIG. 15 shows an example process (e.g., an orchestrator node) according to some embodiments of the present disclosure;

FIG. 16 shows an example process (e.g., where the network is not attacked) according to some embodiments of the present disclosure;

FIG. 17 shows another example process (e.g., where the network attacked) according to some embodiments of the present disclosure; and

FIG. 18 shows an example process (e.g., where the network attacked) according to some embodiments of the present disclosure.

DETAILED DESCRIPTION

Before describing in detail example embodiments, it is noted that the embodiments reside primarily in combinations of apparatus components and processing steps related to detection of network attacks. Accordingly, components have been represented where appropriate by conventional symbols in the drawings, showing only those specific details that are pertinent to understanding the embodiments so as not to obscure the disclosure with details that will be readily apparent to those of ordinary skill in the art having the benefit of the description herein. Like numbers refer to like elements throughout the description.

As used herein, relational terms, such as "first" and "second," "top" and "bottom," and the like, may be used solely to distinguish one entity or element from another entity or element without necessarily requiring or implying any physical or logical relationship or order between such entities or elements. The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the concepts described herein. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises," "comprising," "includes" and/or "including" when used herein, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

In embodiments described herein, the joining term, "in communication with" and the like, may be used to indicate electrical or data communication, which may be accomplished by physical contact, induction, electromagnetic radiation, radio signaling, infrared signaling or optical signaling, for example. One having ordinary skill in the art will appreciate that multiple components may interoperate, and modifications and variations are possible of achieving the electrical and data communication.

In some embodiments described herein, the term "coupled," "connected," and the like, may be used herein to indicate a connection, although not necessarily directly, and may include wired and/or wireless connections.

The term "network node" used herein can be any kind of network node which may further comprise any of a network function (e.g., VNF). In some embodiments, the network node may comprises one or more nodes associated with a 3GPP-based network such as base station (BS), radio base station, base transceiver station (BTS), base station controller (BSC), radio network controller (RNC), g Node B (gNB), evolved Node B (eNB or eNodeB), Node B, multi-standard radio (MSR) radio node such as MSR BS, multi-cell/multicast coordination entity (MCE), integrated access and backhaul (IAB) node, relay node, donor node controlling relay, radio access point (AP), transmission points, transmission nodes, Remote Radio Unit (RRU) Remote Radio Head (RRH), a core network node (e.g., mobile management entity (MME), self-organizing network (SON) node, a coordinating node, positioning node, MDT node, etc.), an external node (e.g., 3rd party node, a node external to the current network), nodes in distributed antenna system (DAS), a spectrum access system (SAS) node, an element management system (EMS), etc. The network node may also comprise test equipment.

In some embodiments, the network node may comprise a wireless device (WD) (e.g., a user equipment (UE)). The WD herein can be any type of wireless device capable of communicating with a network node or another WD over radio signals, such as wireless device (WD). The WD may also be a radio communication device, target device, device to device (D2D) WD, machine type WD or WD capable of machine to machine communication (M2M), low-cost and/or low-complexity WD, a sensor equipped with WD, Tablet, mobile terminals, smart phone, laptop embedded equipped (LEE), laptop mounted equipment (LME), USB dongles, Customer Premises Equipment (CPE), an Internet of Things (IoT) device, or a Narrowband IoT (NB-IOT) device, etc.

Also, in some embodiments the generic term “radio network node” is used. It can be any kind of a radio network node which may comprise any of base station, radio base station, base transceiver station, base station controller, network controller, RNC, evolved Node B (eNB), Node B, gNB, Multi-cell/multicast Coordination Entity (MCE), IAB node, relay node, access point, radio access point, Remote Radio Unit (RRU) Remote Radio Head (RRH).

Note that although terminology from one particular wireless system, such as, for example, 3GPP LTE and/or New Radio (NR), may be used in this disclosure, this should not be seen as limiting the scope of the disclosure to only the aforementioned system. Other wireless systems, including without limitation Wide Band Code Division Multiple Access (WCDMA), Worldwide Interoperability for Microwave Access (WiMax), Ultra Mobile Broadband (UMB) and Global System for Mobile Communications (GSM), may also benefit from exploiting the ideas covered within this disclosure.

In some embodiments, the term “block" is used and may refer to a set of elements. For example, a packet block may comprise a set of packets (e.g., communication packets).

A packet block may be used to embed secrets (e.g., secret values, inter-packet delays (IPDs), etc.). A packet block may comprise a plurality of frames such as four frames. In some other embodiments, each packet block may comprise a virtual trailer that comprise a set of secrets (such as four secrets) where each secret is embedded into a frame. The term "virtual trailer" may refer to information that is added, embedded, encoded, etc., such as without modification of contents of packets (e.g., without adding a physical trailer such as additional bits, MAC information, etc.)

In some embodiments, virtual trailer may enable detecting network attacks such as SFC attacks and classifying them such as at runtime without adding or changing any packet content (i.e., using a virtual trailer). Determining a virtual trailer and/or using a virtual trailer may comprise appending a verifiable physical trailer (e.g., IPDs) to each packet to ensure forwarding path correctness and SFC attacks detection. In some other embodiments, virtual trailer may refer to the IPDs (or any other communication parameters) that are modified before being received by a receiver VNF or network node. In some embodiments, a packet block (and/or any of the components of the packet block) having modified IPDs (and/or secret values) may be referred to as a virtual trailer. Performing one or more actions associated with a packet block (and/or any of the components of the packet block) may refer to performing one or more actions associated with a virtual trailer. In a nonlimiting example, verifying a packet block may refer to verifying a virtual trailer (e.g., secret values, IPDs, and any other characteristic) associated with the packet block.

In some other embodiments, a virtual trailer is a set of secret messages that carry specific information about a group of packets (e.g., in a block), which can be encoded into those packets such as without adding any bits of data to the packets (e.g., by using watermarking). In some other embodiments, the virtual trailer is a n-tuple (e.g., 4-tuple) comprising a SeqNum, FlowID, RecNF, and SendNF. SeqNum may be an integer value uniquely identifying a packet block related to a flow, FlowID may be an integer value uniquely identifying a flow between two network nodes (e.g., VNFs), RecNF and SenNF are each an integer value uniquely identifying the receiver VNF and the sender VNF, respectively, per each flow.

Note further, that functions described herein as being performed by an agent node, an orchestrator node, and a network node may be distributed over a plurality of agent nodes, orchestrator nodes, and network nodes. In other words, it is contemplated that the functions of the agent node, orchestrator node, and network node described herein are not limited to performance by a single physical (and/or virtual) device and, in fact, can be distributed among several physical (and/or virtual) devices.

Unless otherwise defined, all terms (including technical and scientific terms) used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this disclosure belongs. It will be further understood that terms used herein should be interpreted as having a meaning that is consistent with their meaning in the context of this specification and the relevant art and will not be interpreted in an idealized or overly formal sense unless expressly so defined herein.

Referring now to the drawing figures, in which like elements are referred to by like reference numerals, there is shown in FIG. 1 a schematic diagram of a system 10, according to an embodiment, such as a VNF network that may support one or more communication standards and/or protocols. System 10 may comprise network 20 (e.g., a trusted network) which may comprise one or more agent nodes (e.g., agent nodes 12a, 12b, etc.) and/or orchestrator nodes 14 and/or network nodes 16 (e.g., network nodes 16a, 16b, etc.). One or more of network 20, agent nodes 12, orchestrator nodes 14, and network nodes 16 may be configured to communicate with another network 30 (e.g., untrusted network). Any element of system 10 may communicate with another element of system 10 via direct/indirect and/or wired/wireless communication.

Example implementations, in accordance with an embodiment, of the agent node 12, orchestrator node 14, and network node 16 discussed in the preceding paragraphs will now be described with reference to FIG. 2. In system 10, a network node 16 comprises hardware (HW) 38 including a communication interface 40 configured to set up and maintain a wired or wireless connection with an interface of a different communication device of the system 10. The network node 16 further comprises processing circuitry 42, which may have storage and/or processing capabilities. The processing circuitry 42 may include a processor 44 and memory 46. In particular, in addition to or instead of a processor, such as a central processing unit, and memory, the processing circuitry 42 may comprise integrated circuitry for processing and/or control, e.g., one or more processors and/or processor cores and/or FPGAs (Field Programmable Gate Array) and/or ASICs (Application Specific Integrated Circuitry) adapted to execute instructions. The processor 44 may be configured to access (e.g., write to and/or read from) memory 46, which may comprise any kind of volatile and/or nonvolatile memory, e.g., cache and/or buffer memory and/or RAM (Random Access Memory) and/or ROM (Read-Only Memory) and/or optical memory and/or EPROM (Erasable Programmable Read-Only Memory).

Processing circuitry 42 may be configured to control any of the methods and/or processes described herein and/or to cause such methods, and/or processes to be performed, e.g., by network node 16. Processor 44 corresponds to one or more processors 44 for performing network node 16 functions described herein. The network node 16 includes memory 46 that is configured to store data, programmatic software code and/or other information described herein. In some embodiments, the software 48 and/or the host application 50 may include instructions that, when executed by the processor 44 and/or processing circuitry 42, causes the processor 44 and/or processing circuitry 42 to perform the processes described herein with respect to network node 16. The instructions may be software associated with the network node 16.

The software 48 may be executable by the processing circuitry 42. The software 48 includes a software application 50. The software application 50 may be operable to provide a VNF functions to a user. The processing circuitry 42 of the network node 16 may enable the network node 16 to perform one or more actions associated with a VNF. The processing circuitry 42 of the network node 16 may include a NN management unit 54 configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., VNF functions.

The system 10 further includes an orchestrator node 14 including hardware 58 enabling it to communicate with the network node 16 and agent node 12. The hardware 58 may include a communication interface 60 for setting up and maintaining a wired or wireless connection with an interface of a different communication device of the system 10 (e.g., agent node 12, network node 16). The communication interface 60 may be formed as or may include, for example, one or more RF transmitters, one or more RF receivers, and/or one or more RF transceivers.

In the embodiment shown, the hardware 58 of the orchestrator node 14 further includes processing circuitry 68. The processing circuitry 68 may include a processor 70 and a memory 72. In particular, in addition to or instead of a processor, such as a central processing unit, and memory, the processing circuitry 68 may comprise integrated circuitry for processing and/or control, e.g., one or more processors and/or processor cores and/or FPGAs (Field Programmable Gate Array) and/or ASICs (Application Specific Integrated Circuitry) adapted to execute instructions. The processor 70 may be configured to access (e.g., write to and/or read from) the memory 72, which may comprise any kind of volatile and/or nonvolatile memory, e.g., cache and/or buffer memory and/or RAM

(Random Access Memory) and/or ROM (Read-Only Memory) and/or optical memory and/or EPROM (Erasable Programmable Read-Only Memory).

Thus, the orchestrator node 14 further has software 74 stored internally in, for example, memory 72, or stored in external memory (e.g., database, storage array, network storage device, etc.) accessible by the orchestrator node 14 via an external connection. The software 74 may be executable by the processing circuitry 68. The processing circuitry 68 may be configured to control any of the methods and/or processes described herein and/or to cause such methods, and/or processes to be performed, e.g., by orchestrator node 14. Processor 70 corresponds to one or more processors 70 for performing orchestrator node 14 functions described herein. The memory 72 is configured to store data, programmatic software code and/or other information described herein. In some embodiments, the software 74 may include instructions that, when executed by the processor 70 and/or processing circuitry 68, causes the processor 70 and/or processing circuitry 68 to perform the processes described herein with respect to orchestrator node 14. For example, processing circuitry 68 of the orchestrator node 14 may include orchestrator node (ON) management unit configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., ON functions. Further, processing circuitry 68 of the orchestrator node 14 may include an agents manager unit 100, an audit unit 102, a services integrity unit 104, and an IPD registrar unit 106, any of which may be included in ON management unit 32. Agents manager unit 100 may be configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., agents manager functions.

Audit unit 102 may be configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., auditing steps, determining an audit table, etc. Services integrity unit 104 may be configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., verification of service integrity. IPD registrar unit 106 may be configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., storing and/or registering information associated with IPDs.

The system 10 further includes agent node 12 already referred to. The agent node 12 may have hardware 80 that may include a communication interface 82 configured to set up and maintain a network connection 64 and network connection 52. The communication interface 82 may be formed as or may include, for example, one or more transmitters, one or more receivers, and/or one or more transceivers.

The hardware 80 of the agent node 12 further includes processing circuitry 84. The processing circuitry 84 may include a processor 86 and memory 88. In particular, in addition to or instead of a processor, such as a central processing unit, and memory, the processing circuitry 84 may comprise integrated circuitry for processing and/or control, e.g., one or more processors and/or processor cores and/or FPGAs (Field Programmable Gate Array) and/or ASICs (Application Specific Integrated Circuitry) adapted to execute instructions. The processor 86 may be configured to access (e.g., write to and/or read from) memory 88, which may comprise any kind of volatile and/or nonvolatile memory, e.g., cache and/or buffer memory and/or RAM (Random Access Memory) and/or ROM (Read-Only Memory) and/or optical memory and/or EPROM (Erasable Programmable Read-Only Memory).

Thus, the agent node 12 may further comprise software 90, which is stored in, for example, memory 88 at the agent node 12, or stored in external memory (e.g., database, storage array, network storage device, etc.) accessible by the agent node 12. The software 90 may be executable by the processing circuitry 84. The software 90 may include a client application 92. The client application 92 may be operable to provide a service to a human or non-human user via the agent node 12.

The processing circuitry 84 may be configured to control any of the methods and/or processes described herein and/or to cause such methods, and/or processes to be performed, e.g., by agent node 12. The processor 86 corresponds to one or more processors 86 for performing agent node 12 functions described herein. The agent node 12 includes memory 88 that is configured to store data, programmatic software code and/or other information described herein. In some embodiments, the software 90 and/or the client application 92 may include instructions that, when executed by the processor 86 and/or processing circuitry 84, causes the processor 86 and/or processing circuitry 84 to perform the processes described herein with respect to agent node 12. For example, the processing circuitry 84 may include an agent node (AN) management unit 34 configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., agent node functions. Processing circuitry 84 may further include a controller unit 110, a watermarker unit 112, and extractor unit 114, any of which may be comprised in AN management unit 34. Controller unit 110 may be configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., controller functions. Watermarker unit 112 may be configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., watermarking functions. Further, extractor unit 114 may be configured to perform any step and/or task and/or process and/or method and/or feature described in the present disclosure, e.g., extractor functions.

In some embodiments, orchestrator node 14 may communicate (e.g., via communication interface 60) with agent node 12 (e.g., via communication interface 82) using network connection 64 and with network node 16 (via communication interface 40) using network connection 66. Further, agent node 12 may communicate (e.g., via communication interface 82) with network node 16 (e.g., via communication interface 40) using network connection 52. Any one of network connections 52, 64, 66 may be a physical and/or logical network connection. In some embodiments, network connections 52, 64, 66 may refer to one or more physical network connections associated to one or more logical network connections.

In some other embodiments, the inner workings of the agent node 12, orchestrator node 14, and network node 16 may be as shown in FIG. 2 and independently, the surrounding network topology may be that of FIG. 1.

Although FIG. 2 show various "units" such as ON management unit 32, AN management unit 34, NN management unit, etc., as being within a respective processor, it is contemplated that these units may be implemented such that a portion of the unit is stored in a corresponding memory within the processing circuitry. In other words, the units may be implemented in hardware or in a combination of hardware and software within the processing circuitry.

FIG. 3 is a flowchart of an example process in an agent node 12 (e.g., a first agent node 12a). One or more blocks described herein may be performed by one or more elements of wireless device 22 such as by one or more of processing circuitry 84 (including the AN management unit 34), processor 86, communication interface 82 and/or communication interface 60. Agent node 12 such as via processing circuitry 84 and/or processor 86 and/or communication interface 82 is configured to receive (Block S100) a set of packets 124 addressed to a network node 16 and having a first set of inter-packet delays (IPDs) 126, determine (Block S102) a set of secret values usable to modify the first set of IPDs 126 based at least in part on one or more communication parameters associated with the received set of packets 124, modify (Block S104) at least one IPD 126 of the set of IPDs 126 based on the set of secret values, where the modified first set of IPDs 126 being a second set of IPDs 126, determine (Block S106) a packet block comprising the set of packets 124 and a set of frames122, where each frame of the set of frames122 includes a subset of packets 124 of the set of packets 124 that are spaced in time based on the second set of IPDs 126, and transmit (Block S108) the packet block 120 to the network node 16.

In some embodiments, the method further comprises one or more of measuring each IPD 126 of the first set of IPDs 126; performing a watermarking based on the measurement of each IPD 126, a watermarking amplitude, and the packet block; and transmitting a record to the orchestrator node, where the record is based on the watermarking and to be stored in an IPD registrar of the orchestrator node.

In some other embodiments, performing the watermarking includes determining a first seed value based on an initial seed value and network connection information corresponding to a first network connection; determining a first secret value, a second secret value, a third secret value, and a fourth secret value of the set of secret values for the first network connection based on the first seed value; determining another packet block is associated with the first network connection; and incrementing the first secret value for the other packet block.

In some embodiments, performing the watermarking further includes determining a second seed value based on the initial seed value and network connection information corresponding to a second network connection and determining the first secret value, the second secret value, the third secret value, and the fourth secret value of the set of secret values for the second network connection based on the second seed value. The first secret value for the first network connection and the first secret value for the second network connection are different, and the second secret value of the second network connection is incremented based on the second secret value of the first network connection.

In some other embodiments, the first secret value is a sequence number of the packet block, the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier.

In some embodiments, the first secret value corresponds to a first frame, the second secret value corresponds to a second frame, the third secret value corresponds to a third frame, and the fourth secret value corresponds to a fourth frame of the set of frames 122.

In some other embodiments, each one of the first, second, third, and fourth frames 122a, 122b, 122c, 122d comprise one or both of a first packet 124a and a second packet 124b; and a third packet 124c and a fourth packet 124d.

In some embodiments, performing the watermarking further includes determining a first IPD 126a, a second IPD 126b, and a third IPD 126c of the second set of IPDs 126. The first IPD 126a defines a first space in time between the first packet and the second packet, the second IPD 126b defines a second space in time between the second packet 124b and the third packet 124c, the third IPD 126c defines a third space in time between the third packet 124c and the fourth packet 124d.

In some other embodiments, performing the watermarking further includes converting each one of the first, second, third, and fourth secret values to a bit value having a plurality of bits, where each bit of the plurality of bits is transformed to a transformed value based on the value of the bit. The transformed value is +1 or -1, and the first, second, third, and fourth IPDs 126 are based on a watermarking amplitude a corresponding IPD 126 of the first set of IPDs 126 and the transformed value.

In some embodiments, one or more of the set of packets 124 are received from a first virtual network function comprised in a first network node 16a, where the set of packets 124 being addressed to a second network node 16b; the first agent node 12a comprises a first chain patrol agent; the second network node 16b comprises a second virtual network function configured to receive the set of packets 124; the set of packets 124 are to be obtained by a second agent node 12b that comprises a second chain patrol agent; and the first agent node 12a is configured to communicate with an orchestrator node 14 that comprises a chain patrol orchestrator.

FIG. 4 is a flowchart of an example process in an agent node 12 (e.g., a second agent node 12b). One or more blocks described herein may be performed by one or more elements of wireless device 22 such as by one or more of processing circuitry 84 (including the AN management unit 34), processor 86, communication interface 82 and/or communication interface 60. Agent node 12 such as via processing circuitry 84 and/or processor 86 and/or communication interface 82 is configured to obtain (Block S200) a packet block comprising the set of packets 124 and a set of frames, where each frame of the set of frames comprising a subset of packets 124 of the set of packets 124, the subset of packets 124 being spaced in time based on a second set of inter-packet delays (IPDs) 126; extract (Block S202) a set of secret values from the packet block 120 based on the second set of IPDs 126; and perform (Block S204) a verification of the packet block 120 based on the secret values and a record stored at an IPD registrar of the orchestrator node, where the verification verifies whether the packet block 120 is associated with a network attack.

In some embodiments, the method further includes monitoring the set of packets 124; constructing the packet block 120; measuring each IPD 126 of the second set of IPDs 126; reading the record stored in the IPD Registrar; storing each IPD 126 locally in a local audit table; performing the verification of the packet block 120 at a block level and a flow level; and triggering the orchestrator node to store information from the local audit table in an orchestrator audit table and store each measured IPD 126 of the second set of IPDs 126.

In some other embodiments, extracting the set of secret values includes determining an expected packet block 120 based on watermarking process and generating a set of expected secrets values using the expected packet block 120, where the set of expected secret values is generated for each record of a connection and each packet block 120 record shared by a first agent node 12a.

In some embodiments, extracting the set of secret values includes extracting each secret value of the set of secret values from the packet block 120 to form a computed packet block 120 and forming the computed packet block 120.

In some other embodiments, performing the verification of the packet block 120 includes comparing the computed packet block 120 with the expected packet block 120 and determining that the packet block 120 matches the expected packet block 120 based on the comparison.

In some embodiments, performing the verification of the packet block 120 includes one or more of determining that the packet bock is not associated with the network attack when the packet block 120 matches the expected packet block 120; updating at least on audit table when one packet block 120 is found and each frame of the one packet block 120 is verified; determining that the packet bock 120 is associated with the network attack when at least one secret value of the set of secret values from the packet block 120 does not match a corresponding expected secret value, where the determination is based on at least a corresponding IPD 126; marking the packet block 120 as an unverified packet block 120; and transmitting information associated with the unverified packet block 120 to the orchestrator to perform another verification of the unverified packet block 120.

In some other embodiments, one or more of the second set of IPDs 126 are based on a first seed value, the first seed value being based on an initial seed value and network connection information corresponding to a first network connection; the set of secret values comprise a first secret value, a second secret value, a third secret value, and a fourth secret value for the first network connection based on the first seed value; the second set of IPDs 126 is further based on a second seed value, where the second seed value is based on the initial seed value and network connection information corresponding to a second network connection; the first secret value for the first network connection and the first secret value for the second network connection are different, where the second secret value of the second network connection are incremented based on the second secret value of the first network connection.

In some embodiments, one or more of the first secret value is a sequence number of the packet block 120, where the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier; the first secret value corresponds to a first frame, where the second secret value corresponds to a second frame, the third secret value corresponds to a third frame, and the fourth secret value corresponds to a fourth frame of the set of frames; each one of the first, second, third, and fourth frames 122a, 122b, 122c, 122d comprise a first packet 124a, a second packet 124b, a third packet 124c, and a fourth packet 124d.

In some other embodiments, one or both of the second set of IPDs 126 includes a first IPD 126a, a second IPD 126b, and a third IPD 126c of the second set of IPDs 126, where the first IPD 126a defines a first space in time between the first packet 124a and the second packet 124b, the second IPD 126b defines a second space in time between the second packet 124b and the third packet 124c, the third IPD 126c defines a third space in time between the third packet 124c and the fourth packet 124d; and each one of the first, second, third, and fourth secret values is based on a bit value having a plurality of bits. Each bit of the plurality of bits is transformed to a transformed value based on the value of the bit, the transformed value is +1 or -1, and the first, second, third, and fourth IPDs 126 are based on a watermarking amplitude a corresponding IPD 126 of a first set of IPDs 126 and the transformed value.

In some embodiments, one or more of the set of packets 124 are transmitted by a first virtual network function comprised in a first network node 16a, where the set of packets 124 is addressed to a second network node 16b; the second network node 16b includes a second virtual network function configured to receive the set of packets 124; the second agent node 12b includes a second chain patrol agent; and the orchestrator node 14 includes a chain patrol orchestrator.

FIG. 5 is a flowchart of an example process in an orchestrator node 14. One or more blocks described herein may be performed by one or more elements of network node 16 such as by one or more of processing circuitry 68 (including the ON management unit

32), processor 70, radio interface 62 and/or communication interface 60. orchestrator node 14 such as via processing circuitry 68 and/or processor 70 and/or communication interface 60 is configured to perform (Block S300) a second verification of a packet block 120 based on a first verification of the packet block 120 performed by the second agent node and determine (Block S302) whether the packet block 120 is associated with a network attack.

In some embodiments, the method further includes receiving, from the second agent node, information associated with an unverified packet block 120, where the second verification is performed based on the information.

In some other embodiments, the orchestrator node further includes an inter-packet delay (IPD) registrar, and the method further includes one or more of receiving a record from the first agent node based on a watermarking process performed by the first agent node; storing the record in the IPD registrar of the orchestrator node; storing information from a local audit table associated with the second agent node in an orchestrator audit table; and storing one or more IPDs 126 measured by the second agent node.

In some embodiments, the method further includes instantiating the first and second agent nodes 12a, 12b and transmitting an initial seed at least to the first agent node 12a for modifying at least one IPD 126.

In some other embodiments, the method further includes receiving a first set of IPDs 126, one or more network connection numbers before a watermarking process is performed and transmitting the first set of IPDs 126 to the second agent node 12b for extraction of watermarking information.

In some embodiments, the method further includes receiving one or more IPDs 126 from the second agent node 12b, where the one or more IPDs 126 are associated with an unverified packet block 120 and usable to perform the second verification, and retrieving audit table records.

In some other embodiments, the method further includes determining whether the network attack is a packet level attack based on the one or more IPDs 126 and audit table records.

In some embodiments, determining whether the network attack is a packet level attack includes using a window of fixed size that moves across the one or more IPDs 126; and recovering information about whether the one or more IPDs 126 correspond to watermarked message that has not been found.

In some other embodiments, the method further includes determining whether the network attack is associated with a skipping of a network node 16.

In some embodiments, one or more of the first agent node 12a includes a first chain patrol agent, the second agent node 12b includes a second chain patrol agent, and the orchestrator node 14 includes a chain patrol orchestrator.

Having described the general process flow of arrangements of the disclosure and having provided examples of hardware and software arrangements for implementing the processes and functions of the disclosure, the sections below provide details and examples of arrangements for detection of network attacks.

One or more embodiments describe detection and identification of network attacks/breaches (e.g., using ChainPatrol) such as related to a service chain of VNFs. In some embodiments, detection and identification is achieved using a black box method such as neither by accessing the service provider's infrastructure/information nor by amending the VNF implementation.

FIG. 6 shows an overview of an example system comprising an orchestrator node 14, a first network node 16a, a second network node 16b, a first agent node 12a, and a second agent node 12b according to some embodiments of the present. In some embodiments, first agent node 12a may be comprised (or part of) first network node 16a, and/or second agent node 12b may be comprised (or part of) second network node 16b. First agent node 12a (e.g., ChainPatrol Agent) may comprise control unit 110 (e.g., controller), watermarker unit 112 (e.g., watermarker), and extractor unit 114 (e.g., extractor) and be configured to communicate with network node 16a (e.g., sender VNF), agent node 12b (e.g., ChainPatrol Agent), orchestrator node 14 (e.g., ChainPatrol orchestrator), and network 30 (e.g., untrusted network). Second agent node 12b may be configured to communicate with network node 16b (e.g., receiver VNF), orchestrator node 14, and network 30. Agent node 12b may comprise control unit 110 (e.g., controller), watermarker unit 112 (e.g., watermarker), and extractor unit 114 (e.g., extractor). Further, orchestrator node 14 may comprise an agents manager unit 100 (e.g., agents manager), audit unit 102 (e.g., audit table), services integrity unit 104 (e.g., service integrity verifier), and IPD registrar unit 106 (e.g., IPD registrar).

More specifically, agent nodes 12a, 12b may be attached to VNFs (one per VNF) in the chain and one centralized orchestrator node 14, e.g., having a global view over all agents.

In some embodiments, integrity verification is performed first hop-by-hop along the service function chain (between every two network nodes 16 (VNFs)) and/or globally from end-to-end. Agent node 12a (e.g., ChainPatrol agent, at the sender VNF,) may construct a block of packets, build the virtual trailer by embedding the secret values in the block such as by modifying their timing patterns, and/or sends packets using the virtual trailer without changing or adding any packets or fields in the packets themselves. At the receiver side, agent node 12b (e.g., the ChainPatrol agent) may passively monitor the traffic, reconstruct the virtual trailer, and extract the embedded secret values by observing the packets new timing patterns. In some other embodiments, agent node 12b may compare those values with the expected ones in order to detect whether a network attack has modified the traffic or not. Agent nodes 12 may share specific information via the orchestrator using an audit table and the IPD Registrar unit 106.

In some embodiments, agent nodes 12 may be configured to perform one or more (e.g., two) levels of detection (*i.e.,* block-level and packet-level). In some other embodiments, the block-level verification may be prioritized at the extractor unit 114 and delay the packet-level verification for unverified blocks to be performed by the orchestrator node 14. One goal of this verification separation is to accelerate the extracting process (e.g., without having the agent node perform full verification), e.g., while providing the same accuracy as in packet-level integrity verification. In some other embodiments, other attacks can be detected at the orchestrator level such as attacks related to VNF skipping. More specifically:

- The first level of verification (block-level integrity verification) may be performed at the agent level and may check the integrity of the blocks. If the verification for one block fails, the verification may be moved to a next block, and the unverified block details may be sent to orchestrator node 14. One goal of this step may be to speed up the extraction process.
- The second level of verification (packet-level integrity verification) may be performed at the orchestrator level and may be used to verify the unverified blocks triggered by the agent node 12 and to perform in-depth analysis to understand modifications that happened inside the block. It may also delegate to the orchestrator node 14 VNF skipping detection, e.g., when it can only be verified using a global view of the chain.

<u>Virtual Trailer</u>

Virtual trailer may refer to a feature that allows virtually carrying secret messages embedded in a block of packets by using the interpacket delays IPDs, and without adding or changing any packet or the original content. For each connection (e.g., connection C), blocks of packets are built and then secrets values are embedded to the blocks (e.g., the virtual trailer).

FIG. 7 shows an example plurality of blocks according to some embodiments of the present. Blocks 120 (e.g., blocks 120a, 120) are shown. Further details of block 120a are discussed for ease of understanding, which may be equally applicable to other blocks 120. Block 120a comprises one or more frames 122 (e.g., Frame 1, Frame 2, Frame 3, Frame 4). Frame 122 may comprise one or more packets 124 (e.g., 124a, 124b, 124c, 124d) which may have IPDs 126 (e.g., 126a, 126b, 126c).

In some embodiments, each block 120 may comprise S packets 124 (i.e., block size). The block size may differ from one connection (i.e., flow) to another based on the application type. A block 120 may be divided into four frames, namely, SeqNum, FlowID, RecNF, SenNF. Each frame may comprise N=S/4 packets 124 to carry a secret message of length (W=N-1) bits.

In some other embodiments, the secrets related to the virtual trailer embedded in each frame of a given block may be:

- *SeqNum*: Unique value per block 120. It represents the ordering of blocks between two network nodes 16 (e.g., VNFs) for a given network connection. The *SeqNum* may be generated randomly for the first block using pseudorandom generator (PRG) and the seed value of each connection and then incremented by one for the next blocks within the same connection.
- *FlowID*: Unique value per network connection. It may be generated randomly using the seed value of the connection and the PRG (e.g., only for the first connection), then incremented by one for each new network connection.
- *RecNF* and *SenNF*: These two values refer to the network node 16a, 16b (e.g., receiver VNF identifier (ID) and the sender VNF ID), respectively. These values may be unique per network connection and may be generated randomly using the PRG and the seed value per network connection.

Table 1 lists example attack types that may be detected by each secret frame:

Secret	Attack type
SeqNum	Packet and block injection, dropping, duplication, and reordering
FlowID	Flow injection and dropping
RecNF, *SenNF*	VNF skipping

Table 1. - Type of attacks to be detected by each secret frame.

Initialization parameters

Agent nodes 12 and orchestrator node 14 may share specific parameters (in advance through pre-configuration or at runtime via the orchestrator node 14, such as:

- Size of the block per connection type: block size may depend on the traffic load generated per connection type. For example, secure socket shell (SSH), hypertext transfer protocol (HTTP), or file transfer protocol (FTP) traffic may have a different number of packets to be involved in a network connection, thus the block size may be adapted accordingly.
- Initial seed value per pair of communicating agents (depending on the service chain specification): this seed may be used to generate other seeds (to be used per network connection) and to generate the secret values of each virtual trailer. Using a shared seed may ensure that the two agent nodes 12 can generate the same (random) values using the Pseudo Random Generator (PRG) module. For example, an initial seed such as Seed 0 may be shared by the orchestrator node per pair of agents nodes. Seed 0 may be used to generate another seed such as Seed 1 when a new connection is detected, which may be used to generate yet another seed such as Seed 2 (e.g., if another new connection is detected).

Agent Node

The agent node 12 (e.g., ChainPatrol Agent) may be instantiated next to (and/or in proximity to, as part of, etc.) a network node 16 (e.g., a given VNF) and may maintain communication with the orchestrator node 14. Agent node 12 may comprise one or more components: controller unit 110, watermarker unit 112, and extractor unit 114.

Controller

The controller unit 110 (e.g., controller) may configured for: (i) monitoring the network performance metrics (*e.g.*, network jitter), at each VNF side (ii) intercepting the packets at the sender side and passively monitoring the packets (*i.e.*, from the untrusted network) from the receiver side, (iii) interacting with the IPD Registrar unit 106 (at the orchestrator node 14) to update it with new information (at the sender) or read new information (at the receivers) about connections, the blocks within each connection, and the corresponding original IPD values of the received packets at the sender per block, (iv) building and embedding the virtual trailer to send the corresponding packets (at the sender side), and (v) updating the audit unit 102 (e.g. Audit Table) at the orchestrator node 14 (from the receiver side) to trigger further attack identification, if needed.

FIG. 8 shows an example process (e.g., in an agent node) according to some embodiments of the present disclosure, e.g., workflow of the controller and the tasks performed by the controller at both the sender side and the receiver side. At both sides (sender and receiver), at step S400, the controller unit 110 monitors the network performance to measure the jitter, from which at step S402, it computes the watermark amplitude. Depending on its location, the controller unit 110 may trigger the watermarker unit 112 (if at the sender) or the extractor unit 114 (if at the receiver) to perform one or more actions. More specifically:

- At agent node 12a (e.g., the sender VNF side): the controller unit 110 may: (1) at step S404, intercept the packets and construct a block of packets, (2) at step S406, measure the IPDs between packets within the block (referred to as original IPDs), (3) at step S408, trigger the watermarker by sending these original IPDs, the watermark amplitude, and the block of packets to the watermarker unit 112, (4) at step S410, receive the watermarked packets from the watermarker unit 112, (5) at step S412, store a record in the IPD Registrar at the orchestrator side consisting of the connection, the block number, and a vector of original IPDs, and (6) at step S414, send the watermarked packets through the network.
- At agent node 12b (e.g., the receiver VNF side): the controller unit 110 may: (1) at step S416, monitor the packets received by the receiver VNF and constructs a block of N packets, (2) at step S418, measure the IPD between packets (3) at step S420, read new records in IPD Registrar and stores IPDs locally in Audit Table (4) at step S422, send measured IPD and watermarking amplitude to extractor unit 114, (5) at step S424, receive results from extractor unit 114 and perform block-level and flow-level verification (6) at step S426, store information from local Audit table to Audit table at orchestrator node 14 as well as locally measured IPDs.

The process at both sides may be iterated for each block of new packets. Furthermore, network performance monitoring and watermark amplitude calculation may be restarted periodically to tune the watermarking process.

In some embodiments, controller unit 110 passes details about the unverified block (*i.e., information about* all the packets between the last verified block and the next found block) to the orchestrator node 14 which may perform an in-depth analysis to identify packet-level breaches that cause the block verification failure. Additionally, if a flow has been detected to be injected, the controller unit 110 may pass the information to the orchestrator node 14. The orchestrator node 14 may have visibility over all connections among network nodes 16 (e.g., all VNFs) and identify possible network node (e.g., VNFs) skipping attacks.

Watermarker

The watermarker unit 112 may be used by the agent node 12 (e.g., agent node 12a at the sender VNF) and be configured for (1) generating the secret values of the virtual trailer for each block, and (2) embedding the secret values into the block of packets to form the virtual trailer.

FIG. 9 shows another example process (e.g., in an agent node) according to some embodiments of the present disclosure, e.g., the workflow of the watermarker where the flow inside the dashed box shows the generation of the virtual trailer secret values after receiving a block of unwatermarked packets. More specifically, at step S500, watermarker unit 112 receives IPDs, watermarking amplitude, and the block of packets from controller unit 110. At step S502, the watermarker unit determines whether there is a new network connection. At step S504, a seed value is determined such as randomly and/or based on a prior seed value. At step S506, secret values are determined such as randomly based on a corresponding seed value. At step S508, if the network connection is not new, secret values (e.g., SenNF, RecNF, FlowID) may be reused for the same connection, and the most recent SeqNum may be incremented. At step S510, watermarking is performed and watermarked packets sent to controller unit 110. In doing so, the watermarker unit 112 maintains a watermark table, which may be a data structure used to store run-time information used in the watermarking process.

Table 2 shows an example watermark table.

Connection	**Secret**	**SenNF**	**RecNF**	**FlowID**	**SeqNum**
C1	Seed 1	X1	Y1	Z1	N1

		X1	Y1	Z1	N1+1
		…	…	…	…
C2	Seed 2	X2	Y2	Z1+1	N2
		X2	Y2	Z1+1	N2+1
		…	…	…	…
Ci	Seed i	…	…	…	…

Table 2. - Structural example of the watermark table.

In some embodiments, the watermark table may show the information in the order it is created (e.g., based on the received traffic). For example, the table may have C1, then C2, then C1 again. The above table is an example provided for understanding the layout of the table, i.e.,, the table may have other arrangements.

For each new network connection (*i.e.,* the same tuples < source internet protocol (IP) address, destination IP address, source port number, destination port number, the protocol number>), a seed value is generated, which may be used to generate values of other fields (e.g., SenNF, RecNF, FlowID, SeqNum). For example, Seed 0 may be used for the generation of each new seed value for each new connection. The generation of these values may be performed for each block of each network connection as follows:

- The first block may carry the four values (secrets, i.e., SeqNum, FlowID, RecNF, and SenNF) shown in the four last columns, which may be generated using the seed value and the PRG. The four secrets in the virtual trailer are encoded in the packet block such as trailing each corresponding secret to a subset of the IPDs (i.e., frame) in a packet block.
- If another block is built after the first block, for the same connection, the same SenNF, RecNF, FlowID may be used, but the SeqNum may be incremented.

After generating the secret values, the watermark unit 112 may calculate the new IPD values to be used as a new timing pattern between packets (frame per frame) and then encode the secret values of the virtual trailer into the block of packets. The encoding may be added based on the IPD between each pair of packets within the packet block.

FIG. 10 shows an example process (e.g., a watermarking process in an agent node) according to some embodiments of the present disclosure. The detailed steps are explained as follow (for the sake of simplicity, we use a secret value to be embedded to a frame of five packets, which means it has 4 IPDs, as example):

- At step S600, the integer secret value is transformed into a binary value. For example, if a FlowID is of value 4, it is transformed to 0100.
- At step S602, the binary value is encoded into a watermarking message, where bits 1 are transformed to +1 and bits 0 are transformed to -1. Following the same example, the value 0100 is transformed to -1,+1,-1,-1,
- At step S604, each value in the watermarking message (± 1) (representing one delay between two packets) is first multiplied by the watermarking amplitude and then added to the original IPD. Generally, the amplitude a is calculated from the signal to noise formula and a << IPDs (original IPD measured at the sender side). Thus, this ensures that we never get a negative IPD value. More precisely, the new IPD value T_i^w , between packet p_k and P_{k+1} is computed using the following equation:

$$T_i^w = T_i^u + a * m_i^e \qquad (1)$$

Where:

- T_i^u is the original inter-packet delay before embedding the watermark,
- a is the watermarking amplitude and chosen to be small enough so that the artificial jitter caused by watermark embedding is invisible to ordinary users.
- m_i^e is one value of the watermarking message, which has value -1 or +1, depending on the binary value of the secret, at the considered position I (position of the pair of packets considered in the IPD).

Thus, for an amplitude of a = 2, and original IPD $T_i^u = 4$, the new IPD $T_i^w = 4 - 2 = 2$. Given this, packet P_{k+1} will be sent just 2 ms after P_k(and not after 4 ms as it was originally done by the sender VNF). At step S608, the IPDs are embedded in packets.

<u>Extractor</u>

The extractor unit 114 (e.g., extractor) is used by the agent node 12 (e.g., 12b at the receiver VNF). It is responsible for (1) generating the secrets values of the expected virtual trailer for each new record of a connection/block shared by the controller at the sender VNF, (2) extracting the secret values from the block of packets to form the computed virtual trailer, and (3) matching the computed virtual trailer with the expected virtual trailer (4) updating the audit table when a block is found, and all frames are verified.

FIG. 11 shows another example process (e.g., in an agent node) according to some embodiments of the present disclosure, e.g., extractor main workflow for each new block received. At step S700, a new record is read in the audit table with the new IPD. At step S702, expected secret values are generated. At step S704, watermark amplitude and measured IPDs are received from controller unit 110. At step S706, the extracted watermark is matched with (e.g., compared to) the expected virtual trailer (frame per frame). At step S740, whether the SeqNum match is determined. If true, at step S712, the local audit table (found) is updated. At step S714, the other frames are checked to determine whether there is a match. If true, at step S716, the local audit table (verified) is updated. If at least one frame does not match, at step S714, results are sent for further investigation. If SeqNum does not match at step S740, at step S718, the method includes determining whether other new IPDs are present, and if so, at least step S706 is performed.

In some embodiments, if the extractor unit 114 needs to monitor new records added by the controller unit 110 about new blocks and their original IPDs (in IPD registrar as inserted by the agent at the sender), the extractor unit 114 may follow the same process as the watermark to generate the expected virtual trailer secret values. If the extractor unit 114 receives measured IPDs (related to a newly received block by the VNF at the receiver) and the watermarking amplitude (as estimated by the controller), the extractor unit 114 may start the extraction of the watermarking. Based on the obtainer secret values, extractor unit 114 may match them with the values from the audit table (which were generated at the reception of the new IPDs). This process may be performed for each frame secret value starting by the SeqNum field.

In some other embodiments, matching the latter will confirm that the received block matches with one of the sent blocks. Thus, the extractor unit 114 may update local audit table record by setting the value found to 1. Then, each of the three other values is extracted and matched. In case all of them match, the extractor unit 114 may set the value verified to 1 (which will be a tag for more investigation by the orchestrator for packet-level verification). In case of no match, other possible IPDs of other blocks, which are received from IPD Registrar but have not been investigated yet, need to be investigated more (to identify re-ordered blocks for instance). In case there was a problem in matching the frame of a given block, this is reported to the controller unit 110 to start further investigation about that block. This will be sent by controller unit 110 to the orchestrator for packet-level verification.

To do so, the extractor unit 114 and the controller unit 110 may use a local audit table. FIG. 12 shows an example audit table according to some embodiments of the present disclosure. The connection column and IPDs column are used by the controller to store the newly received records from the IPD Registrar (which represents blocks already sent from sender side but need to be extracted at this side (receiver side). The extractor unit 114, based on these and the PRG, may generate a seed value as well as the expected virtual trailer secret values. Column labeled “verified” may be used by the extractor unit 114 to update whether all frames of the received matching block were recovered and the column labeled “found” is used by extractor to update whether the block in this record was found (using SeqNum secret value). It is also used by the controller unit 110 to update for each record what attack has been identified at block-level and flow level. For instance, the controller unit 110 can trigger that block was dropped if the IPD was never matched and the next block was already matched.

FIG. 13 shows an example process (e.g., an extracting process in an agent node) according to some embodiments of the present disclosure. At step S800, measured IPDs are received at the extractor unit 114. At step S802, delay value and watermarking amplitude are used to generate an extracted watermarked message. At step S804, the watermarked message is transformed to a binary value, and at step S806, the binary value is transformed to an integer secret value.

<u>ChainPatrol Orchestrator</u>

The orchestrator node 14 (e.g., ChainPatrol Orchestrator) comprises: (1) agents manager unit 100 (e.g., agents manager) configured to orchestrate the agent nodes 12 (e.g., ChainPatrol agents) instantiated along with network nodes 16 (e.g., VNFs), (2) audit unit 102 (e.g., comprising audit table), (3) IPD Registrar unit 106 (e.g., IPD Registrar) configured to store original IPD values used for watermarking and extracting, and (4) service integrity unit 104 (e.g., Service Integrity Verifier) configured for performing a packet-level verification, e.g., when necessary.

<u>Agents Manager</u>

Agent manager unit 100 (e.g., the agent manager) is configured for instantiating different agent nodes 12 (e.g., agents attached to VNFs, next to or in proximity to VNFs, comprised in the corresponding VNF, etc.) and sharing the initial seed values with pairs of VNFs via a secure channel. Whenever an agent node 12 (e.g., ChainPatrol agent) receives a new connection, it sends the connection information to the orchestrator node 14 (*i.e.,* source IP, destination IP, source port number, destination port number, the protocol number). The agent manager unit 100 stores this information in the audit table, generates a new seed value, determines the block size, and then sends the same seed value and the block size information to the agent node 12 (e.g., ChainPatrol agent at the receiver VNF).

IPD Registrar

IPD registrar unit 106 may be configured as a centralized database to assist the controller unit 110 of the agent node 12 at the receiver VNF to perform the block-level verification. IPD registrar unit 106 is populated by the sender VNF during the watermarking. In some embodiments, the agent node 12 at the sender VNF sends only the list of original IPDs along with the connection identifier before watermarking to the IPD Registrar. The latter shares original IPDs with the extractor unit 114 of the agent node 12 at the receiver VNF in order to extract the watermark message, as explained in equation (2). In some embodiments, the connection identifier may be used to identify characteristics of connection and/or comprise <sender IP, receiver IP, sender port number, receiver port number, protocol number>.

Connection	IPDs (S values)
c1	$[IPD_{c1,1,1}, IPD_{c1,1,2}, \ldots, IPD_{c1,1,s}]$
	$[IPD_{c1,2,1}, IPD_{c1,2,2}, \ldots, IPD_{c1,2,s}]$
	…
c2	$[IPD_{c2,1,1}, IPD_{c2,1,2}, \ldots, IPD_{c2,1,s}]$
	$[IPD_{c2,2,1}, IPD_{c2,2,2}, \ldots, IPD_{c2,2,s}]$
	…

Table 3. - A structural example of IPD Registrar.

In some embodiments, IPD registrar may show the information in the order it is created (based on the received traffic), i.e., c1, c2, then c1.

Service Integrity Verifier

In some embodiments, the service integrity unit 104 (e.g., service integrity verifier) may be a passive component triggered only when a breach is reported by the controller unit 110 of the agent node 12 at the receiver VNF. In such a scenario, the controller unit 110 of the agent node 12 at the receiver VNF shares the IPDs of the received and unverified packets by the block-level integrity verification phase with the orchestrator node 14. The blocks that were already successfully verified are marked *fully verified* and the rest are only partially verified (*i.e.,* some frames have not been able to be recovered), which needs further investigation. This information is stored on the audit table. FIG. 14 shows another example audit table according to some embodiments of the present disclosure.

The audit table helps not only in detecting attacks/breaches but also in identifying them. To achieve an accurate identification, the audit table maintains information about the fully or partially verified blocks that would help in detecting packet-level attacks that can be conducted along multiple blocks (*e.g.,* packet injection, dropping).

In some embodiments, the packet-level verification is a more complex form of verification such that the orchestrator node 14 is in charge to understand the reason for the verification failure of the received traffic by reconstructing all the possible combinations that those packets can form to reconstruct the originally inserted watermark. Since packet-level verification may consume more resources compared with block-level verification, it is executed by the service integrity unit 104 at the orchestrator node 14, e.g., in an offline manner.

FIG. 15 shows an example process (e.g., an orchestrator node) according to some embodiments of the present disclosure. The detailed workflow (e.g., for ChainPatrol) at the orchestrator node 14 may be as follows:

- Step S900: the orchestrator node 14 retrieves the IPD values of packets received by agent nodes 12 (e.g., all agent nodes associated with all VNFs) and the updated/added audit table records. If packet-level is needed, step 904 is activated. Otherwise, the orchestrator checks if VNF skipping needs to be verified, at step S906. If true, step S908 is activated, otherwise, the iteration ends until step 900 is reactivated.
- Step S904: The orchestrator node 14 checks the packet-level attacks, where the orchestrator node 14 uses a window of fixed size that moves (e.g., one step at the time) across the IPD of received packet by the receiver VNF and tries to recover if they correspond to one of the not found yet watermarked message. This allows identifying packets-level attacks such as injection or dropping.
- Step S908: The orchestrator node 14 checks the VNF skipping attacks. When the agent node 12 at receiver side receives injected flows, it may be possible that those flows are supposed to be destined to some previous VNF in the chain. As the orchestrator node 16 has a global view across all VNFs, the

orchestrator node 14 can identify VNF skipping attacks. In some embodiments, VNF attacks may be missed flows detected by other agent node 12 (e.g., ChainPatrol agent) at some other VNF.

Running example

Two nonlimiting examples are described to showcase how agent nodes 12 (e.g., ChainPatrol agents) work at the sender and receiver VNF. The first example illustrates the watermarking and extracting without any breach in the network, while the second scenario showcases a detection of a breach by agent node 12 (e.g., ChainPatrol) at the receiver side as well as at the orchestrator node 14.

Without an attack

FIG. 16 shows an example process (e.g., where the network is not attacked) according to some embodiments of the present disclosure. At step S1000, egress traffic is received by controller unit 110a. At step S1002, a block is constructed. At step S1004, a watermark table is created. At step S1006, new IPDs are calculated and secreted codes encoded into the block. At step S1008 the packet with the virtual trailer is sent. At step S1010, passive monitoring is performed. At step S1012, the virtual trailer is reconstructed. At step S1014, a local audit table is used, which may be fetched from the IPD registrar unit 106. AT step S1016, embedded secret codes are extracted. At step S1018, block-level integrity verification is performed.

In some embodiments, controller unit 110 of agent node 12 at the sender VNF constructs the block, and the watermarker unit 112 calculates the new IPDs in order to generate a virtual trailer by embedding the secret message in the block. For this example, the secret messages are SeqNum 001, FlowID 010, RecNF 010, and SenNF 001. The control unit 110 may end the packets with the virtual trailer (e.g., having a second set of IPDs) via the untrusted network 30.

At the receiver NFV side, the controller unit 110 of agent node 12 (e.g., agent node 12b) reconstructs the virtual trailer and then triggers the extractor unit 114 to extract the embedded secret messages. In this scenario, the extracted values are SeqNum 001, FlowID 010, RecNF 010, and SenNF 001. The controller unit 110 then performs block-level integrity verification by comparing the extracted virtual trailer with the expected virtual trailer. Since both virtual trailers match, no breach is detected/identified.

With attack detection

(1) Block injection attack: FIG. 17 shows another example process (e.g., where the network attacked) according to some embodiments of the present disclosure, e.g., example

of a block replay attack. At step S1100, virtual trailer (i.e., packets being separated per newly determined IPDs). At step S1102, a replayed block is sent (e.g., by a malicious user). At step S1104, passive monitoring is performed. At step S1106, virtual trailer is reconstructed. At step S1108, extractor unit 114 generates or updates the local audit table. At step S1110, embedded secret codes are extracted. At step S1112, block-level integrity verification is performed.

In some embodiments, at the receiver VNF, the extracted secret messages from the received virtual trailer are SeqNum 101, FlowID 110, RecNF 001, and SenNF 101. These values do not match with the initial virtual trailer created at the sender VNF. In this case, the SeqNum does not match which leads to identifying the attack as a block injection attack. The block is marked unverified, and its related information is sent to the orchestrator node 14 for further investigation.

(2) Packet injection attack: In light of the previous example, the orchestrator node 14 may be configured to investigate more to identify attacks related to the packet level. FIG. 18 shows an example process (e.g., where the network attacked) according to some embodiments of the present disclosure, e.g., example of packet injection. At step S1200, virtual trailer is sent by agent node 12 at sender VNF. A packet is injected at step S1202. Passive monitoring is performed at step S1204. The virtual trailer is reconstructed at step S1206. The local audit table is generated or updated at step S1208, and embedded secret codes are extracted at step S1210. At step S1212, block-level integrity verification is performed. At step S1214, in-depth investigation is performed by orchestrator node 14, and at step S1216, packet-level integrity verification is performed.

In some embodiments, controller unit 110 of agent node 12 at the receiver VNF constructs the virtual trailer and then extracts the embedded secret messages. Since a new packet has been injected (*e.g.*, the 4th packet) the block witnesses a packet shifting (starting from the 5th packet) which leads to a change in the IPDs and an extra added packet in the blook (S+1). This, in return, leads to a change in the secret message embedded by the watermarker unit 112 in the frames.

In some other embodiments, the controller unit 110 may mark the block unverified and asks the orchestrator node 14 for further verification. The service integrity unit 104 (e.g., Service Integrity Verifier) performs an in-depth analysis of the received IPDs from the receiver VNF and the expected IPDs that should be received. The analysis leads to identifying that a new packet has been injected.

In some embodiments, any one of the elements of system 10 (e.g., agent nodes 12, orchestrator node 14) may determine attacks categories (e.g., covered by Chain-Patrol). The determination may be based on a table such as Table 4.

Level	Attack	Classification
Packet	Injection	Unrecov. or recov. VT with extra packets more than the block size
	Reordering	Partially recovered VT with total number of packets equal to the block size
	Replay	Unrecov. VT with extra packets more than or less the block size
	Dropping	Partially recovered VT with total number of packets less than the block size
Block	Injection	Unrecov. VT with number of packets equal to the block size*
	Reordering	Out of order SeqNum
	Replay	Recov. VT with repeated SeqNum frame
	Dropping	Missing VT with specific SeqNum
Flow	Injection	Unregistered conn. in watermark table
	Reordering	Out of order FlowID
	Replay	Repeated VT for all blocks
	Dropping	Missing FlowID
SFC	VNF Injection	Unregistered conn. in watermark table
	Reordering	Unexpected SenNF/RecNF
	VNF-Loop	Unexpected SenNF/RecNF
	Skipping	Unexpected SenNF/RecNF

Table 4. - Excerpt of example attacks detected and classified agent nodes 12 and/or orchestrator node 14. VT refers to virtual trailer, and "Unrecov. VT" may refer to a VT that may be matched neither fully nor partially.

More specifically, Table 4 shows an excerpt of the attacks with categories of attacks (column 1, i.e., level), the attack (column 2), and how attack is classified (column 3). Therein, in addition to other packet-level attacks, block-level attacks involve the manipulation of a whole block of packets. Flow-level attacks involve all blocks (packets) related to a given flow between VNFs, and SFC-level attacks are SFC-specific attacks involving the partial or complete modification of the forwarding paths in the chain.

In some embodiments, the first agent node 12a may intercept communication between the first network node 16a (sender VNF) and the second network node 16b (receiver VNF). In some other embodiments, the second agent node 12b passively monitors the traffic that was sent by the first agent node 12a to the second network node 16b. In some embodiments, agent nodes 12a, 12b are configured to communicate with the orchestrator node 14. In some other embodiments, any agent node 12 may be internal or external to the corresponding network node 16 (VNF).

In some other embodiments, a set of packets is received which may have a size equal to a block. In some embodiments, the “secret values” are generated and used to convey “communication parameters” such as (e.g. SeqNum, FlowID, etc.). In some other embodiments, the seed value is a mechanism for modifying the IPDs. In some embodiments, a seed value may be pre-configured in the agent node 12 before being deployed. In some other embodiments, all IPDs are modified to protect all packets.

In some embodiments, the packet block is determined, which may comprise assembling or structuring the received packets in accordance with the modified IPDs to form blocks.

In some other embodiments, the first secret value is a sequence number of the packet block, the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier. In some embodiments, third and fourth secret values uniquely identify the receiver VNF and the sender VNF, respectively and are unique per connection and pair of VNFs. In some other embodiments, VNF1 may communicate with VNF2 using two different connection c1 and c2 different from c1.

In some embodiments, a frame may have at least packets, and the size of the frame (in terms of number of packets) may depend on the block size. The block size may depend on the connection type.

In some embodiments, the second agent node is configured to monitor the set of packets, construct the packet block, measure each IPD of the second set of IPDs, read the record stored in the IPD Registrar, and/or computes an expected virtual trailer used to perform the verification of the packet blocks.

In some other embodiments, the second set of IPDs comprises a first IPD 126a, a second IPD 126b, and a third IPD 126c of the second set of IPDs. In some embodiments, there are more than three IPDs in a frame. In some other embodiments, if the number of packets is N, the number of IPDs is N-1.

As will be appreciated by one of skill in the art, the concepts described herein may be embodied as a method, data processing system, computer program product and/or computer storage media storing an executable computer program. Accordingly, the concepts described herein may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects all generally referred to herein as a "circuit" or "module." Any process, step, action and/or functionality described herein may be performed by, and/or associated to, a corresponding module, which may be implemented in software and/or firmware and/or hardware. Furthermore, the disclosure may take the form of a computer program product on a tangible computer usable storage medium having computer program code embodied in the medium that can be executed by a computer. Any suitable tangible computer readable medium may be utilized including hard disks, CD-ROMs, electronic storage devices, optical storage devices, or magnetic storage devices.

Some embodiments are described herein with reference to flowchart illustrations and/or block diagrams of methods, systems and computer program products. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer (to thereby create a special purpose computer), special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

These computer program instructions may also be stored in a computer readable memory or storage medium that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer readable memory produce an article of manufacture including instruction means which implement the function/act specified in the flowchart and/or block diagram block or blocks.

The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

It is to be understood that the functions/acts noted in the blocks may occur out of the order noted in the operational illustrations. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality/acts involved. Although some of the diagrams include arrows on communication paths to show a primary direction of communication, it is to be understood that communication may occur in the opposite direction to the depicted arrows.

Computer program code for carrying out operations of the concepts described herein may be written in an object oriented programming language such as Python, Java® or C++. However, the computer program code for carrying out operations of the disclosure may also be written in conventional procedural programming languages, such as the "C" programming language. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer. In the latter scenario, the remote computer may be connected to the user's computer through a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

Many different embodiments have been disclosed herein, in connection with the above description and the drawings. It will be understood that it would be unduly repetitious and obfuscating to literally describe and illustrate every combination and subcombination of these embodiments. Accordingly, all embodiments can be combined in any way and/or combination, and the present specification, including the drawings, shall be construed to constitute a complete written description of all combinations and subcombinations of the embodiments described herein, and of the manner and process of making and using them, and shall support claims to any such combination or subcombination.

Abbreviations that may be used in the preceding description include:

MAC	Message Authentication Code
NFV	Network Function Virtualization

IPD	Inter-Packets Delay
PRG	Pseudo Random Generator
SFC	Service Function Chaining
VNF	Virtual Network Functions

It will be appreciated by persons skilled in the art that the embodiments described herein are not limited to what has been particularly shown and described herein above. In addition, unless mention was made above to the contrary, it should be noted that all of the accompanying drawings are not to scale. A variety of modifications and variations are possible in light of the above teachings without departing from the scope of the following claims.

What is claimed is:

1. A first agent node (12a) comprising processing circuitry (84) configured to:

receive a set of packets (124) addressed to a network node (16) and having a first set of inter-packet delays, IPDs;

determine a set of secret values usable to modify the first set of IPDs (126) based at least in part on one or more communication parameters associated with the received set of packets (124);

modify at least one IPD (126) of the set of IPDs (126) based on the set of secret values, the modified first set of IPDs (126) being a second set of IPDs (126);

determine a packet block (120) comprising the set of packets (124) and a set of frames, each frame (122) of the set of frames comprising a subset of packets (124) of the set of packets (124) that are spaced in time based on the second set of IPDs (126); and

cause transmission of the packet block (120) to the network node (16).

2. The first agent node (12a) of Claim 1, wherein the processing circuitry (84) is further configured to one or more of:

measure each IPD (126) of the first set of IPDs (126);

perform a watermarking based on the measurement of each IPD (126), a watermarking amplitude, and the packet block (120); and

cause transmission of a record to an orchestrator node (14), the record being based on the watermarking and to be stored in an IPD registrar (106) of the orchestrator node (14).

3. The first agent node (12a) of Claim 2, wherein performing the watermarking comprises:

determining a first seed value based on an initial seed value and network connection information corresponding to a first network connection;

determining a first secret value, a second secret value, a third secret value, and a fourth secret value of the set of secret values for the first network connection based on the first seed value;

determining another packet block (120) is associated with the first network connection; and

incrementing the first secret value for the other packet block (120) .

4. The first agent node (12a) of Claim 3, wherein performing the watermarking further comprises:

determining a second seed value based on the initial seed value and network connection information corresponding to a second network connection; and

determining the first secret value, the second secret value, the third secret value, and the fourth secret value of the set of secret values for the second network connection based on the second seed value, the first secret value for the first network connection and the first secret value for the second network connection being different, the second secret value of the second network connection being incremented based on the second secret value of the first network connection.

5. The first agent node (12a) of any one of Claims 3 and 4, wherein the first secret value is a sequence number of the packet block (120) , the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier.

6. The first agent node (12a) of any one of Claims 3-5, wherein the first secret value corresponds to a first frame (122a) , the second secret value corresponds to a second frame (122b) , the third secret value corresponds to a third frame (122c) , and the fourth secret value corresponds to a fourth frame (122d) of the set of frames.

7. The first agent node (12a) of Claim 6, wherein each one of the first, second, third, and fourth frames (122a, 122b, 122c, 122d) comprise one or both of:

a first packet (124a) and a second packet (124b); and

a third packet (124c) and a fourth packet (124d).

8. The first agent node (12a) of Claim 7, wherein performing the watermarking further comprises:

determining a first IPD, a second IPD, and a third IPD (126a, 126b, 126c) of the second set of IPDs (126), the first IPD (126a) defining a first space in time between the first packet (124a) and the second packet (124b), the second IPD (126b) defining a second space in time between the second packet (124b) and the third packet (124c), the third IPD (126c) defining a third space in time between the third packet (124c) and the fourth packet (124d).

9. The first agent node (12a) of Claim 8, wherein performing the watermarking further comprises:

converting each one of the first, second, third, and fourth secret values to a bit value having a plurality of bits, each bit of the plurality of bits being transformed to a transformed value based on the value of the bit, the transformed value being +1 or -1, the first, second, third, and fourth IPDs (126a, 126b, 126c, 126d) being based on a watermarking amplitude, corresponding IPDs (126) of the first set of IPDs (126) and the transformed value.

10. The first agent node (12a) of any one of Claims 1-9, wherein one or more of:

the set of packets (124) are received from a first virtual network function comprised in a first network node (16a), the set of packets (124) being addressed to a second network node (16b);

the first agent node (12a) comprises a first chain patrol agent;

the second network node (16b) comprises a second virtual network function configured to receive the set of packets (124);

the set of packets (124) are to be obtained by a second agent node (12b) that comprises a second chain patrol agent; and

the first agent node (12a) is configured to communicate with an orchestrator node (14) that comprises a chain patrol orchestrator.

11. A method in a first agent node (12a), the method comprising:

receiving (S100) a set of packets (124) addressed to a network node (16) and having a first set of inter-packet delays, IPDs (126);

determining (S102) a set of secret values usable to modify the first set of IPDs (126) based at least in part on one or more communication parameters associated with the received set of packets (124);

modifying (S104) at least one IPD (126) of the set of IPDs (126) based on the set of secret values, the modified first set of IPDs (126) being a second set of IPDs (126);

determining (S106) a packet block (120) comprising the set of packets (124) and a set of frames, each frame (122) of the set of frames comprising a subset of packets (124) of the set of packets (124) that are spaced in time based on the second set of IPDs (126); and

transmitting (S108) the packet block (120) to the network node (16).

12. The method of Claim 11, wherein the method further comprises one or more of:

measuring each IPD (126) of the first set of IPDs (126);

performing a watermarking based on the measurement of each IPD, a watermarking amplitude, and the packet block (120) ; and

transmitting a record to an orchestrator node (14), the record being based on the watermarking and to be stored in an IPD registrar (106) of the orchestrator node (14).

13. The method of Claim 12, wherein performing the watermarking comprises:

determining a first seed value based on an initial seed value and network connection information corresponding to a first network connection;

determining a first secret value, a second secret value, a third secret value, and a fourth secret value of the set of secret values for the first network connection based on the first seed value;

determining another packet block (120) is associated with the first network connection; and

incrementing the first secret value for the other packet block (120) .

14. The method of Claim 13, wherein performing the watermarking further comprises:

determining a second seed value based on the initial seed value and network connection information corresponding to a second network connection; and

determining the first secret value, the second secret value, the third secret value, and the fourth secret value of the set of secret values for the second network connection based on the second seed value, the first secret value for the first network connection and the first secret value for the second network connection being different, the second secret value of the second network connection being incremented based on the second secret value of the first network connection.

15. The method of any one of Claims 13 and 14, wherein the first secret value is a sequence number of the packet block (120) , the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier.

16. The method of any one of Claims 13-15, wherein the first secret value corresponds to a first frame (122a) , the second secret value corresponds to a second frame (122b) , the third secret value corresponds to a third frame (122c) , and the fourth secret value corresponds to a fourth frame (122d) of the set of frames.

17. The method of Claim 16, wherein each one of the first, second, third, and fourth frames (122a, 122b, 122c, 122d) comprise one or both of:

a first packet (124a) and a second packet (124b); and

a third packet (124c) and a fourth packet (124d).

18. The method de of Claim 17, wherein performing the watermarking further comprises:

determining a first IPD (126a), a second IPD (126b), and a third IPD (126c) of the second set of IPDs (126), the first IPD (126a) defining a first space in time between the first packet (124a) and the second packet (124b), the second IPD (126b) defining a second space in time between the second packet (124b) and the third packet (124c), the third IPD (126c) defining a third space in time between the third packet (124c) and the fourth packet (124d).

19. The method of Claim 18, wherein performing the watermarking further comprises:

converting each one of the first, second, third, and fourth secret values to a bit value having a plurality of bits, each bit of the plurality of bits being transformed to a transformed value based on the value of the bit, the transformed value being +1 or -1, the first, second, third, and fourth IPDs (126a, 126b, 126c, 126d) being based on a watermarking amplitude a corresponding IPD (126) of the first set of IPDs (126) and the transformed value.

20. The method of any one of Claims 11-19, wherein one or more of:

the set of packets (124) are received from a first virtual network function comprised in a first network node (16a), the set of packets (124) being addressed to a second network node (16b);

the first agent node (12a) comprises a first chain patrol agent;

the second network node (16b) comprises a second virtual network function configured to receive the set of packets (124);

the set of packets (124) are to be obtained by a second agent node (12b) that comprises a second chain patrol agent; and

the first agent node (12a) is configured to communicate with an orchestrator node (14) that comprises a chain patrol orchestrator.

21. A second agent node (12b) configured to communicate with an orchestrator node (14), the second agent node (12b) comprising processing circuitry (84) configured to:

obtain a packet block (120) comprising the set of packets (124) and a set of frames, each frame (122) of the set of frames comprising a subset of packets (124) of the set of packets (124), the subset of packets (124) being spaced in time based on a second set of inter-packet delays, IPDs (126);

extract a set of secret values from the packet block (120) based on the second set of IPDs (126); and

perform a verification of the packet block (120) based on the secret values and a record stored at an IPD registrar (106) of the orchestrator node (14), the verification verifying whether the packet block (120) is associated with a network attack.

22. The second agent node (12b) of Claim 21, wherein the processing circuitry (84) is further configured to one or more of:

monitor the set of packets (124);

construct the packet block (120) ;

measure each IPD (126) of the second set of IPDs (126);

read the record stored in the IPD registrar (106) ;

store each IPD (126) locally in a local audit table;

perform the verification of the packet block (120) at a block level and a flow level; and

trigger the orchestrator node (14) to:

store information from the local audit table in an orchestrator audit table; and

store each measured IPD (126) of the second set of IPDs (126).

23. The second agent node (12b) of any one of Claims 21 and 22, wherein extracting the set of secret values comprises:

determining a set of expected secrets values using an expected packet block (120) corresponding to a watermarking process, the set of expected secret values being generated for each record of a connection and each packet block (120) record shared by a first agent node (12a).

24. The second agent node (12b) of any one of Claims 21-23, wherein extracting the set of secret values comprises:

extracting each secret value of the set of secret values from the packet block (120) to form a computed packet block (120); and

forming the computed packet block (120).

25. The second agent node (12b) of any one of Claims 21-24, wherein performing the verification of the packet block (120) comprises:

comparing the computed packet block (120) with the expected packet block (120); and

determining that the packet block (120) matches the expected packet block (120) based on the comparison.

26. The second agent node (12b) of Claim 25, wherein performing the verification of the packet block (120) comprises one or more of:

determining that the packet bock is not associated with the network attack when the packet block (120) matches the expected packet block (120);

updating at least on audit table when one packet block (120) is found and each frame (122) of the one packet block (120) is verified;

determining that the packet bock is associated with the network attack when at least one secret value of the set of secret values from the packet block (120) does not match a corresponding expected secret value, the determination being based on at least a corresponding IPD;

marking the packet block (120) as an unverified packet block (120); and

cause transmission of information associated with the unverified packet block (120) to the orchestrator to perform another verification of the unverified packet block (120).

27. The second agent node (12b) of any one of Claims 21-26, wherein one or more of:

the second set of IPDs (126) are based on a first seed value, the first seed value being based on an initial seed value and network connection information corresponding to a first network connection;

the set of secret values comprise a first secret value, a second secret value, a third secret value, and a fourth secret value for the first network connection based on the first seed value;

the second set of IPDs (126) is further based on a second seed value, the second seed value being based on the initial seed value and network connection information corresponding to a second network connection;

the first secret value for the first network connection and the first secret value for the second network connection are different, the second secret value of the second network connection are incremented based on the second secret value of the first network connection.

28. The second agent node (12b) of Claim 27, wherein one or more of:

the first secret value is a sequence number of the packet block (120), the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier;

the first secret value corresponds to a first frame (122a) , the second secret value corresponds to a second frame (122b) , the third secret value corresponds to a third frame (122c) , and the fourth secret value corresponds to a fourth frame (122d) of the set of frames;

each one of the first, second, third, and fourth frames (122a, 122b, 122c, 122d) comprise one or both of:

a first packet (124a), a second packet (124b); and

a third packet (124c), and a fourth packet (124d);

29. The second agent node (12b) of any one of Claims 21-28, wherein one or both of:

the second set of IPDs (126) comprises a first IPD (126a), a second IPD (126b), and a third IPD (126c) of the second set of IPDs (126), the first IPD (126a) defining a first space in time between the first packet (124a) and the second packet (124b), the second IPD (126b) defining a second space in time between the second packet (124b) and the third packet (124c), the third IPD (126c) defining a third space in time between the third packet (124c) and the fourth packet (124d); and

each one of the first, second, third, and fourth secret values is based on a bit value having a plurality of bits, each bit of the plurality of bits being transformed to a transformed value based on the value of the bit, the transformed value being +1 or -1, the first, second, third, and fourth IPDs (126a, 126b, 126c, 126d) being based on a watermarking amplitude a corresponding IPD of a first set of IPDs (126) and the transformed value.

30. The second agent node (12b) of any one of Claims 21-29, wherein one or more of:

the set of packets (124) are transmitted by a first virtual network function comprised in a first network node (16a), the set of packets (124) being addressed to a second network node (16b);

the second network node (16b) comprises a second virtual network function configured to receive the set of packets (124);

the second agent node (12b) comprises a second chain patrol agent; and

the orchestrator node (14) comprises a chain patrol orchestrator.

31. A method in a second agent node (12b) configured to communicate with an orchestrator node (14), the method comprising:

obtaining (S200) a packet block (120) comprising the set of packets (124) and a set of frames, each frame (122) of the set of frames comprising a subset of packets (124) of the set of packets (124), the subset of packets (124) being spaced in time based on a second set of inter-packet delays, IPDs (126);

extracting (S202) a set of secret values from the packet block (120) based on the second set of IPDs (126); and performing (S204) a verification of the packet block (120) based on the secret values and a record stored at an IPD registrar (106) of the orchestrator node (14), the verification verifying whether the packet block (120) is associated with a network attack.

32. The method of Claim 31, wherein the method further comprises:

monitoring the set of packets (124);

construct the packet block (120);

measuring each IPD (126) of the second set of IPDs (126);

reading the record stored in the IPD registrar (106) ;

storing each IPD (126) locally in a local audit table;

performing the verification of the packet block (120) at a block (120) level and a flow level; and

triggering the orchestrator node (14) to:

store information from the local audit table in an orchestrator audit table; and

store each measured IPD (126) of the second set of IPDs (126).

33. The method of any one of Claims 31 and 32, wherein extracting the set of secret values comprises:

determining an expected packet block (120) based on watermarking process; and

generating a set of expected secrets values using the expected packet block (120), the set of expected secret values being generated for each record of a connection and each packet block (120) record shared by a first agent node (12a).

34. The method of any one of Claims 31-33, wherein extracting the set of secret values comprises:

extracting each secret value of the set of secret values from the packet block (120) to form a computed packet block (120); and

forming the computed packet block (120).

35. The method of any one of Claims 31-34, wherein performing the verification of the packet block (120) comprises:

comparing the computed packet block (120) with the expected packet block (120); and determining that the packet block (120) matches the expected packet block (120) based on the comparison.

36. The method of Claim 35, wherein performing the verification of the packet block (120) comprises one or more of:

determining that the packet bock is not associated with the network attack when the packet block (120) matches the expected packet block (120);

updating at least on audit table when one packet block (120) is found and each frame (122) of the one packet block (120) is verified;

determining that the packet bock is associated with the network attack when at least one secret value of the set of secret values from the packet block (120) does not match a corresponding expected secret value, the determination being based on at least a corresponding IPD;

marking the packet block (120) as an unverified packet block (120); and

transmitting information associated with the unverified packet block (120) to the orchestrator to perform another verification of the unverified packet block (120).

37. The method of any one of Claims 31-36, wherein one or more of:

the second set of IPDs (126) are based on a first seed value, the first seed value being based on an initial seed value and network connection information corresponding to a first network connection;

the set of secret values comprise a first secret value, a second secret value, a third secret value, and a fourth secret value for the first network connection based on the first seed value;

the second set of IPDs (126) is further based on a second seed value, the second seed value being based on the initial seed value and network connection information corresponding to a second network connection;

the first secret value for the first network connection and the first secret value for the second network connection are different, the second secret value of the second network connection are incremented based on the second secret value of the first network connection.

38. The method of Claim 37, wherein one or more of:

the first secret value is a sequence number of the packet block (120), the second secret value is a flow identifier for each network connection, the third secret value is a first network node identifier, and the fourth secret value is a second network node identifier;

the first secret value corresponds to a first frame (122a) , the second secret value corresponds to a second frame (122b) , the third secret value corresponds to a third frame (122c) , and the fourth secret value corresponds to a fourth frame (122d) of the set of frames; and

each one of the first, second, third, and fourth frames (122a, 122b, 122c, 122d) comprise a first packet (124a), a second packet (124b), a third packet (124c), and a fourth packet (124d).

39. The method of any one of Claims 31-38, wherein one or both of:

the second set of IPDs (126) comprises a first IPD (126a), a second IPD (126b), and a third IPD (126c) of the second set of IPDs (126), the first IPD (126a) defining a first space in time between the first packet (124a) and the second packet (124b), the second IPD (126b) defining a second space in time between the second packet (124b) and the third packet (124c), the third IPD (126c) defining a third space in time between the third packet (124c) and the fourth packet (124d); and

each one of the first, second, third, and fourth secret values is based on a bit value having a plurality of bits, each bit of the plurality of bits being transformed to a transformed value based on the value of the bit, the transformed value being +1 or -1, the first, second, third, and fourth IPDs (126a, 126b, 126c, 126d) being based on a watermarking amplitude a corresponding IPD (126) of a first set of IPDs (126) and the transformed value.

40. The method of any one of Claims 31-39, wherein one or more of:

the set of packets (124) are transmitted by a first virtual network function comprised in a first network node (16a), the set of packets (124) being addressed to a second network node (16b);

the second network node (16b) comprises a second virtual network function configured to receive the set of packets (124);

the second agent node (12b) comprises a second chain patrol agent; and

the orchestrator node (14) comprises a chain patrol orchestrator.

41. An orchestrator node (14) configured to communicate with a first agent node (12a) and a second agent node (12b), the orchestrator node (14) comprising processing circuitry (68) configured to:

perform a second verification of a packet block (120) based on a first verification of the packet block (120) performed by the second agent node (12b); and

determine whether the packet block (120) is associated with a network attack.

42. The orchestrator node (14) of Claim 41, wherein the processing circuitry (68) is further configured to:

receive, from the second agent node (12b), information associated with an unverified packet block (120), the second verification being performed based on the information.

43. The orchestrator node (14) of any one of Claims 41 and 42, wherein the orchestrator node (14) further comprises an inter-packet delay, IPD, registrar, and the processing circuitry (68) is further configured to one or more of:

receive a record from the first agent node (12a) based on a watermarking process performed by the first agent node (12a);

store the record in the IPD registrar (106) of the orchestrator node (14);

store information from a local audit table associated with the second agent node (12b) in an orchestrator audit table; and

store one or more IPDs (126) measured by the second agent node (12b).

44. The orchestrator node (14) of any one of Claims 41-43, wherein the processing circuitry (68) is further configured to:

instantiate the first and second agent nodes (12a, 12b); and

cause transmission of an initial seed at least to the first agent node (12a) for modifying at least one IPD (126).

45. The orchestrator node (14) of any one of Claims 41-44, wherein the processing circuitry (68) is further configured to:

receive a first set of IPDs (126), one or more network connection numbers before a watermarking process is performed; and cause transmission of the first set of IPDs (126) to the second agent node (12b) for extraction of watermarking information.

46. The orchestrator node (14) of any one of Claims 41-45, wherein the processing circuitry (68) is further configured to:

receive one or more IPDs (126) from the second agent node (12b), the one or more IPDs (126) being associated with an unverified packet block (120) and usable to perform the second verification; and

retrieve audit table records.

47. The orchestrator node (14) of Claim 46, wherein the processing circuitry (68) is further configured to:

determine whether the network attack is a packet level attack based on the one or more IPDs (126) and audit table records.

48. The orchestrator node (14) of Claim 47, wherein determining whether the network attack is a packet level attack comprises:

using a window of fixed size that moves across the one or more IPDs (126); and

recovering information about whether the one or more IPDs (126) correspond to watermarked message that has not been found.

49. The orchestrator node (14) of any one of Claims 41-48, wherein the processing circuitry (68) is further configured to:

determine whether the network attack is associated with a skipping a network node (16).

50. The second agent node (12b) of any one of Claims 41-49, wherein one or more of:

the first agent node (12a) comprises a first chain patrol agent;

the second agent node (12b) comprises a second chain patrol agent; and

the orchestrator node (14) comprises a chain patrol orchestrator.

51. A method in an orchestrator node (14) configured to communicate with a first agent node (12a) and a second agent node (12b), the method comprising:

performing (S300) a second verification of a packet block (120) based on a first verification of the packet block (120) performed by the second agent node (12b); and

determining (S302) whether the packet block (120) is associated with a network attack.

52. The method of Claim 51, wherein the method further comprises:

receiving, from the second agent node (12b), information associated with an unverified packet block (120), the second verification being performed based on the information.

53. The method of any one of Claims 51 and 52, wherein the orchestrator node (14) further comprises an inter-packet delay, IPD, registrar, and the method further comprises one or more of:

receiving a record from the first agent node (12a) based on a watermarking process performed by the first agent node (12a);

storing the record in the IPD registrar (106) of the orchestrator node (14);

storing information from a local audit table associated with the second agent node (12b) in an orchestrator audit table; and

storing one or more IPDs (126) measured by the second agent node (12b).

54. The method of any one of Claims 51-53, wherein the method further comprises:

instantiating the first and second agent nodes (12a, 12b); and

transmitting an initial seed at least to the first agent node (12a) for modifying at least one IPD (126).

55. The method of any one of Claims 51-54, the method further comprises:

receiving a first set of IPDs (126), one or more network connection numbers before a watermarking process is performed; and

transmitting the first set of IPDs (126) to the second agent node (12b) for extraction of watermarking information.

56. The method of any one of Claims 51-55, wherein the method further comprises:

receiving one or more IPDs (126) from the second agent node (12b), the one or more IPDs (126) being associated with an unverified packet block (120) and usable to perform the second verification; and

retrieving audit table records.

57. The method of Claim 56, wherein the method further comprises:

determining whether the network attack is a packet level attack based on the one or more IPDs (126) and audit table records.

58. The method of Claim 57, wherein determining whether the network attack is a packet level attack comprises:

using a window of fixed size that moves across the one or more IPDs (126); and

recovering information about whether the one or more IPDs (126) correspond to watermarked message that has not been found.

59. The method of any one of Claims 51-58, wherein the method further comprises:

determining whether the network attack is associated with a skipping of a network node (16).

60. The method of any one of Claims 51-59, wherein one or more of:

the first agent node (12a) comprises a first chain patrol agent;

the second agent node (12b) comprises a second chain patrol agent; and

the orchestrator node (14) comprises a chain patrol orchestrator.

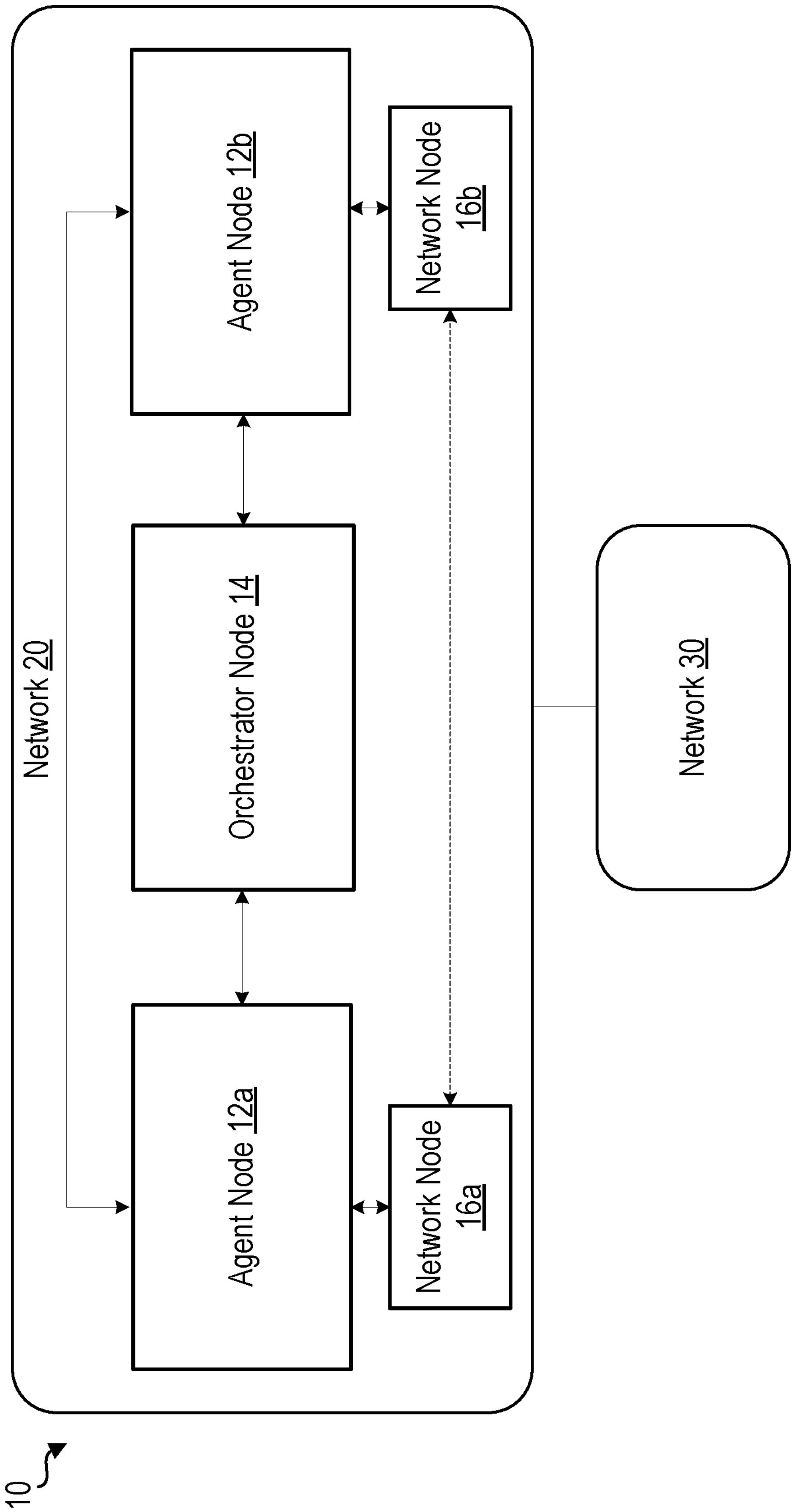
Network 20
Agent Node 12b
Network Node 16b
Orchestrator Node 14
Agent Node 12a
Network Node 16a
Network 30
10

FIG. 1

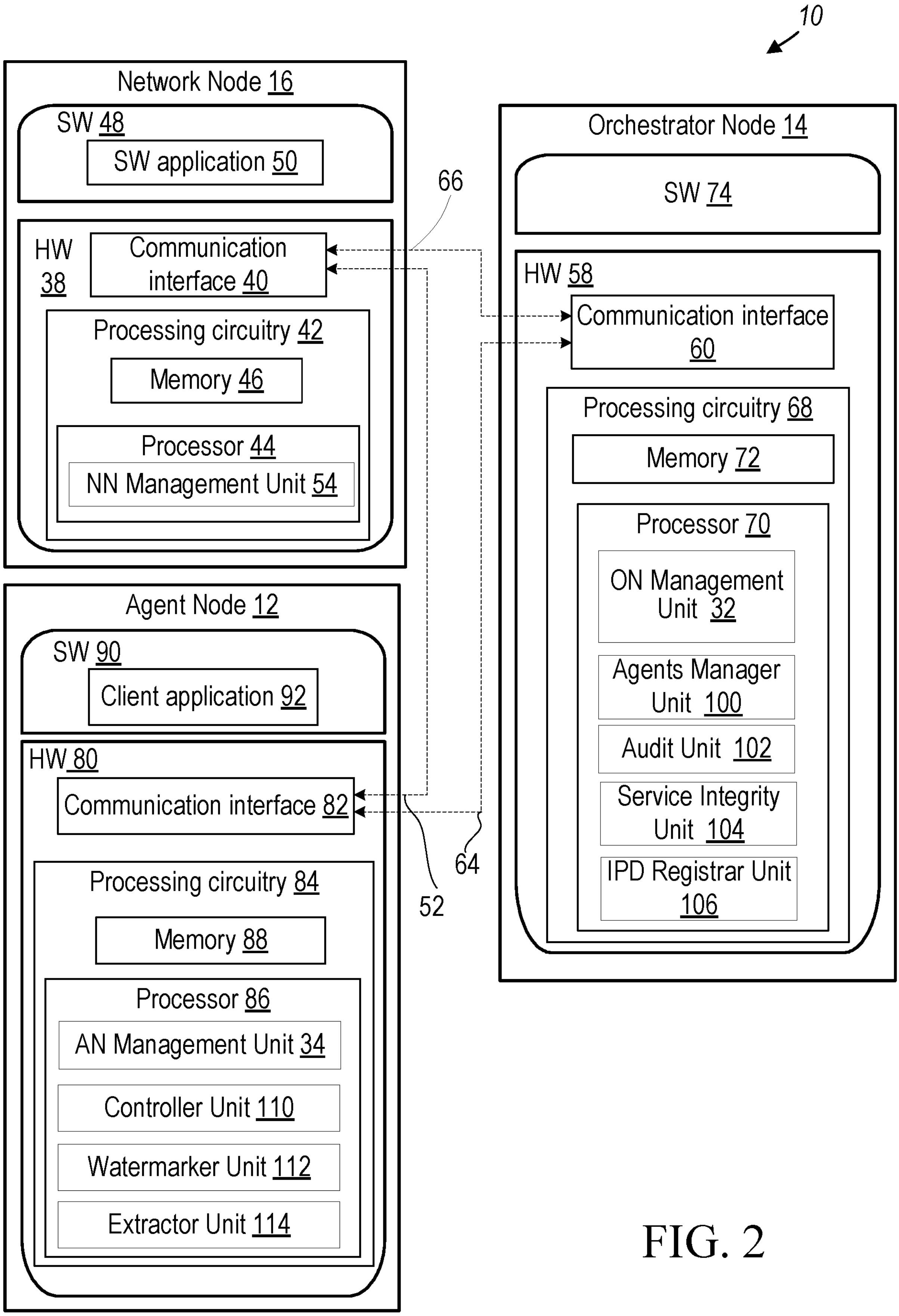
10
Network Node 16
SW 48
SW application 50
HW 38
Communication interface 40
Processing circuitry 42
Memory 46
Processor 44
NN Management Unit 54
66
Orchestrator Node 14
SW 74
HW 58
Communication interface 60
Processing circuitry 68
Memory 72
Processor 70
ON Management Unit 32
Agents Manager Unit 100
Audit Unit 102
Service Integrity Unit 104
IPD Registrar Unit 106
Agent Node 12
SW 90
Client application 92
HW 80
Communication interface 82
Processing circuitry 84
Memory 88
Processor 86
AN Management Unit 34
Controller Unit 110
Watermarker Unit 112
Extractor Unit 114
64
52

FIG. 2

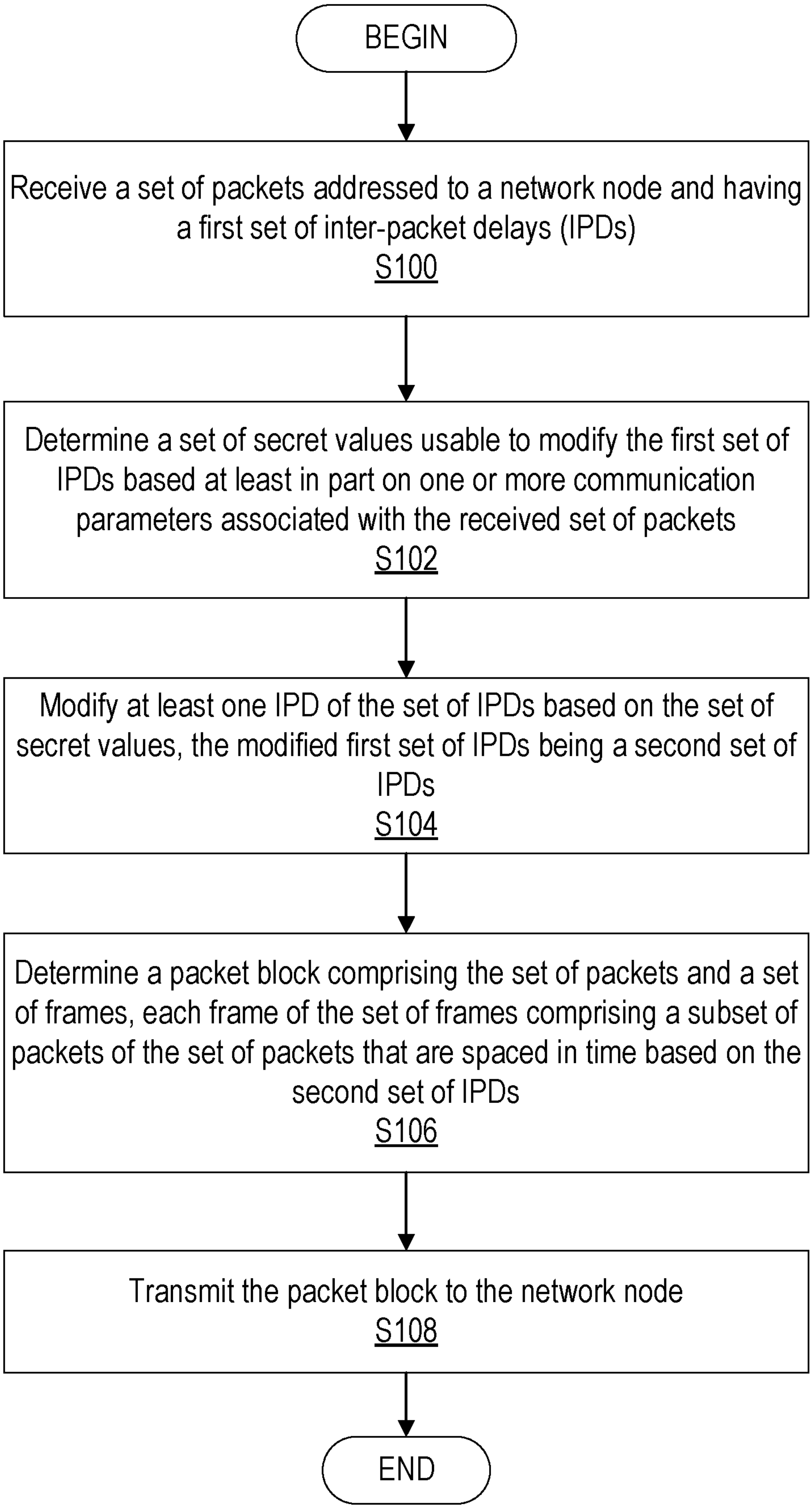
BEGIN
Receive a set of packets addressed to a network node and having a first set of inter-packet delays (IPDs)
S100
Determine a set of secret values usable to modify the first set of IPDs based at least in part on one or more communication parameters associated with the received set of packets
S102
Modify at least one IPD of the set of IPDs based on the set of secret values, the modified first set of IPDs being a second set of IPDs
S104
Determine a packet block comprising the set of packets and a set of frames, each frame of the set of frames comprising a subset of packets of the set of packets that are spaced in time based on the second set of IPDs
S106
Transmit the packet block to the network node
S108
END

FIG. 3

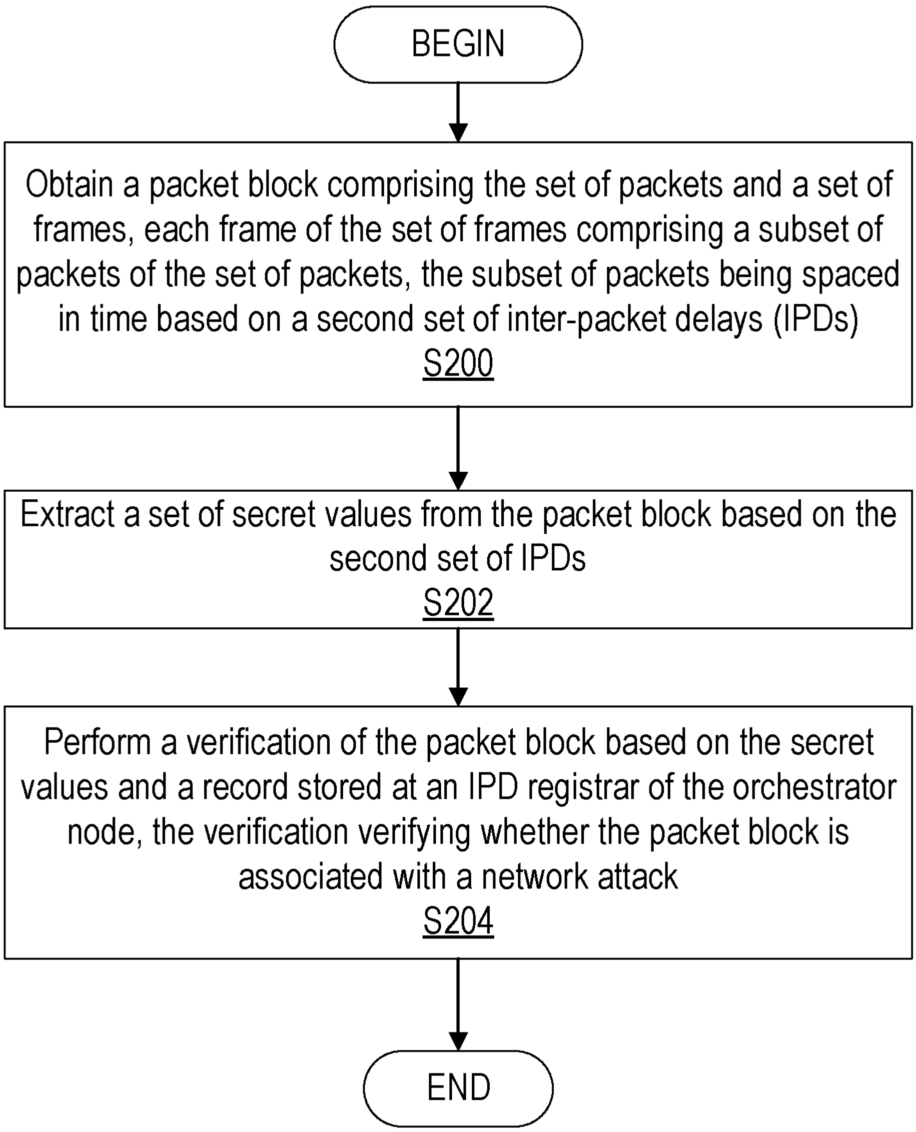
BEGIN
Obtain a packet block comprising the set of packets and a set of frames, each frame of the set of frames comprising a subset of packets of the set of packets, the subset of packets being spaced in time based on a second set of inter-packet delays (IPDs)
S200
Extract a set of secret values from the packet block based on the second set of IPDs
S202
Perform a verification of the packet block based on the secret values and a record stored at an IPD registrar of the orchestrator node, the verification verifying whether the packet block is associated with a network attack
S204
END

FIG. 4

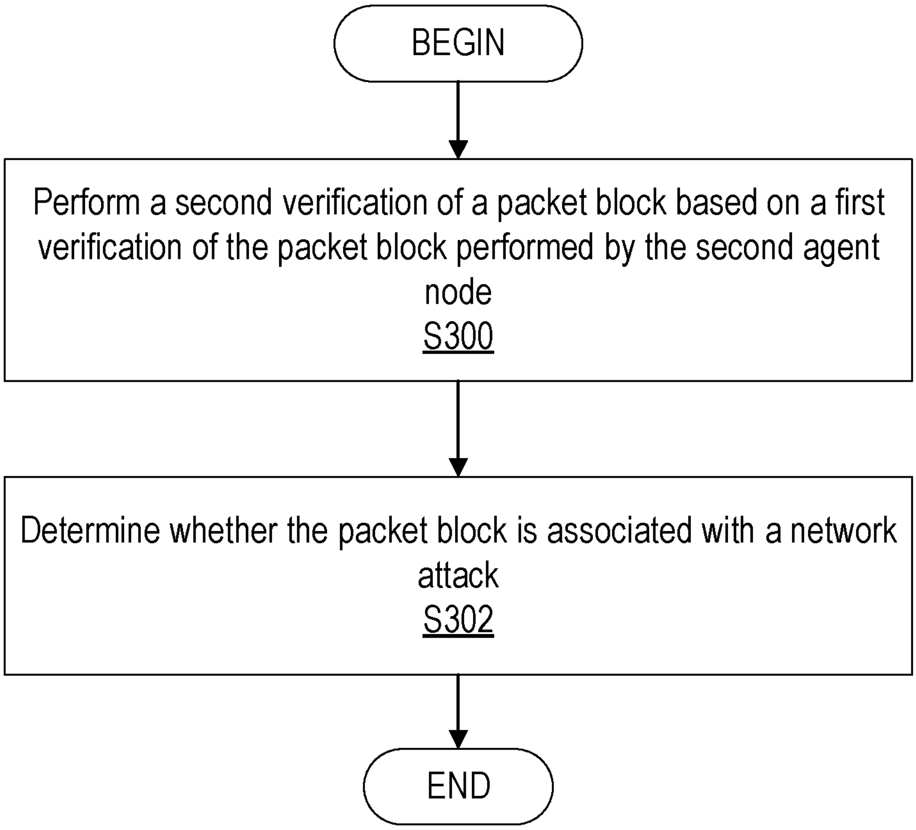
BEGIN
Perform a second verification of a packet block based on a first verification of the packet block performed by the second agent node
S300
Determine whether the packet block is associated with a network attack
S302
END

FIG. 5

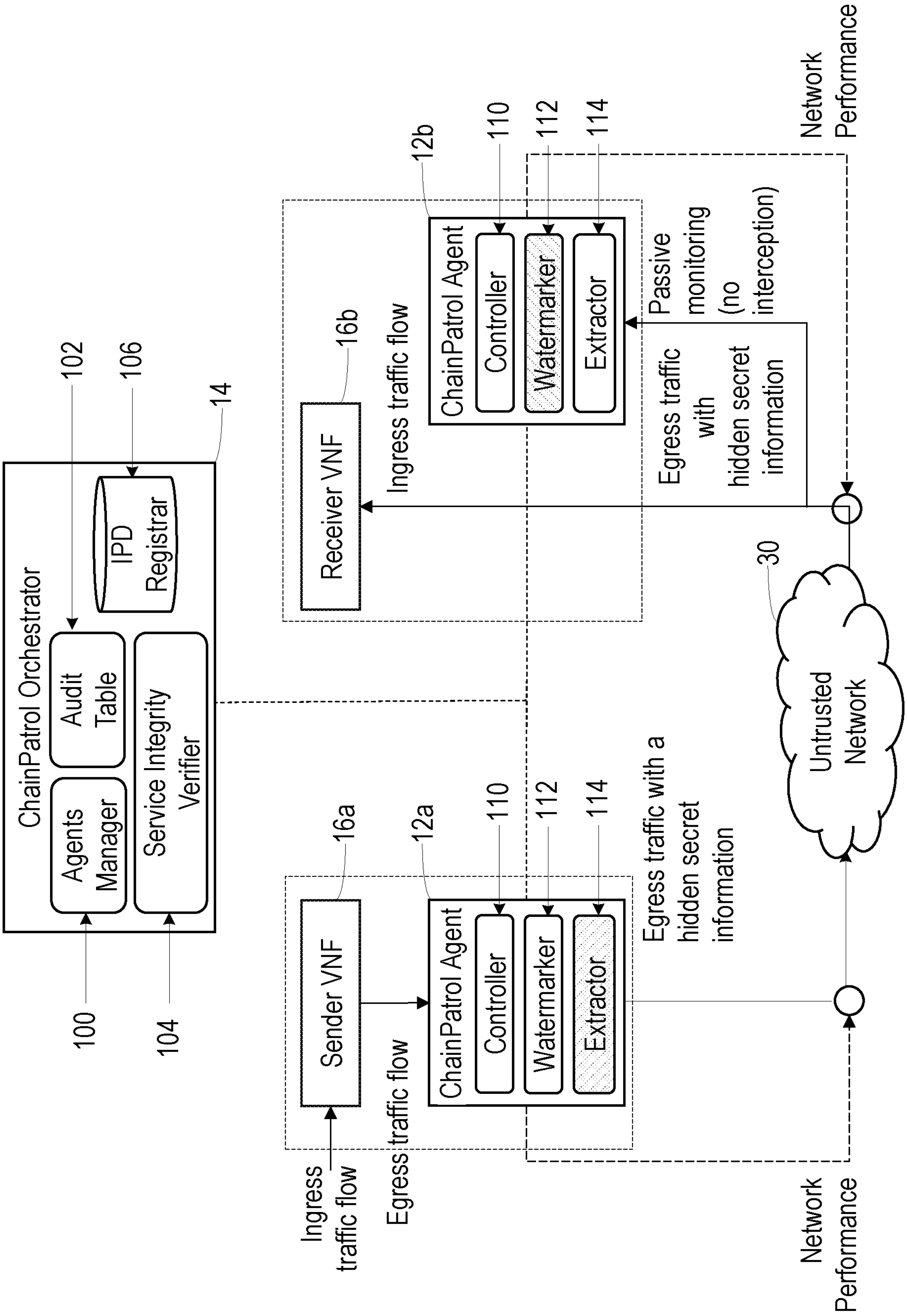
ChainPatrol Orchestrator
Agents Manager
Audit Table
Service Integrity Verifier
IPD Registrar
100
102
104
106
14
Sender VNF
16a
12a
ChainPatrol Agent
Controller
Watermarker
Extractor
110
112
114
Ingress traffic flow
Egress traffic flow
Egress traffic with a hidden secret information
Network Performance
Untrusted Network
30
Receiver VNF
16b
12b
Ingress traffic flow
Egress traffic with hidden secret information
Passive monitoring (no interception)

FIG. 6

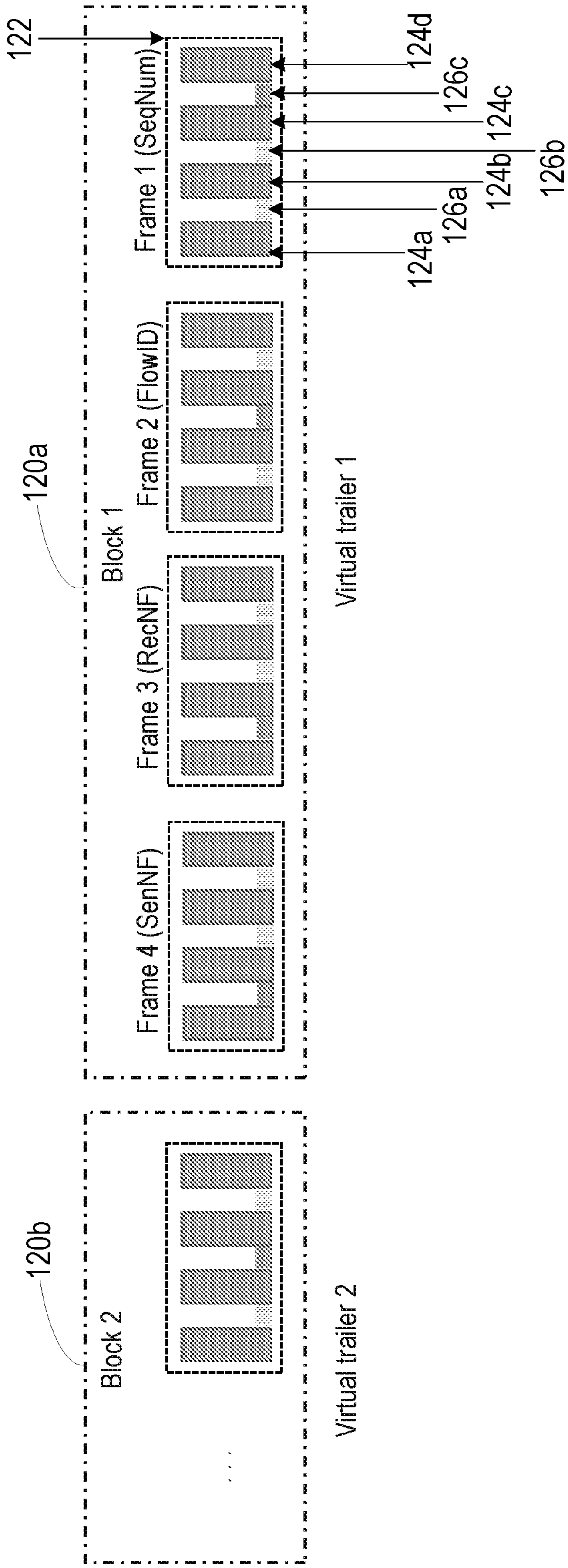
122
Frame 1 (SeqNum)
124d
126c
124c
126b
124b
126a
124a
Frame 2 (FlowID)
120a
Block 1
Frame 3 (RecNF)
Frame 4 (SenNF)
Virtual trailer 1
120b
Block 2
Virtual trailer 2

FIG. 7

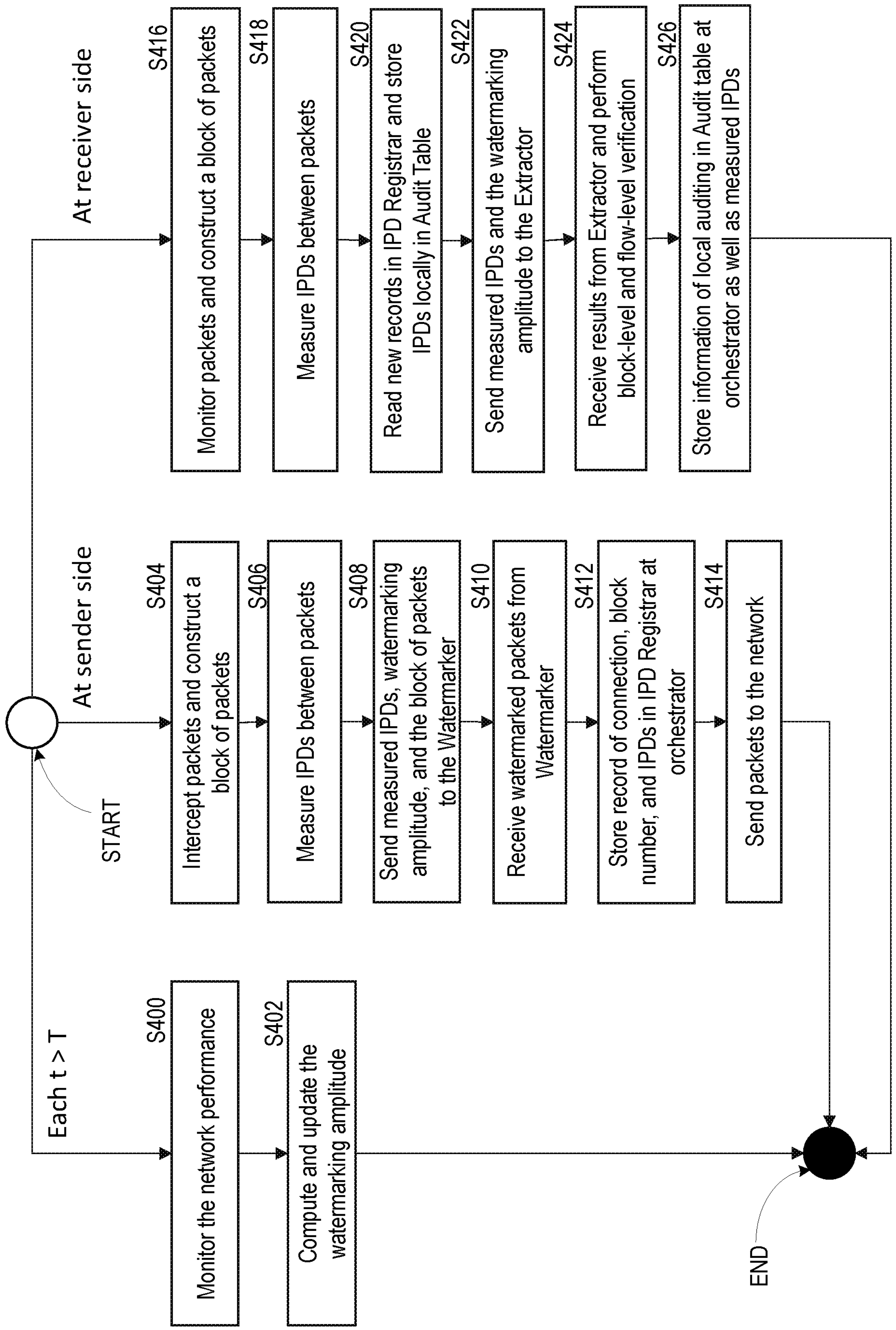
START
END
Each t > T
S400
Monitor the network performance
S402
Compute and update the watermarking amplitude
At sender side
S404
Intercept packets and construct a block of packets
S406
Measure IPDs between packets
S408
Send measured IPDs, watermarking amplitude, and the block of packets to the Watermarker
S410
Receive watermarked packets from Watermarker
S412
Store record of connection, block number, and IPDs in IPD Registrar at orchestrator
S414
Send packets to the network
At receiver side
S416
Monitor packets and construct a block of packets
S418
Measure IPDs between packets
S420
Read new records in IPD Registrar and store IPDs locally in Audit Table
S422
Send measured IPDs and the watermarking amplitude to the Extractor
S424
Receive results from Extractor and perform block-level and flow-level verification
S426
Store information of local auditing in Audit table at orchestrator as well as measured IPDs

FIG. 8

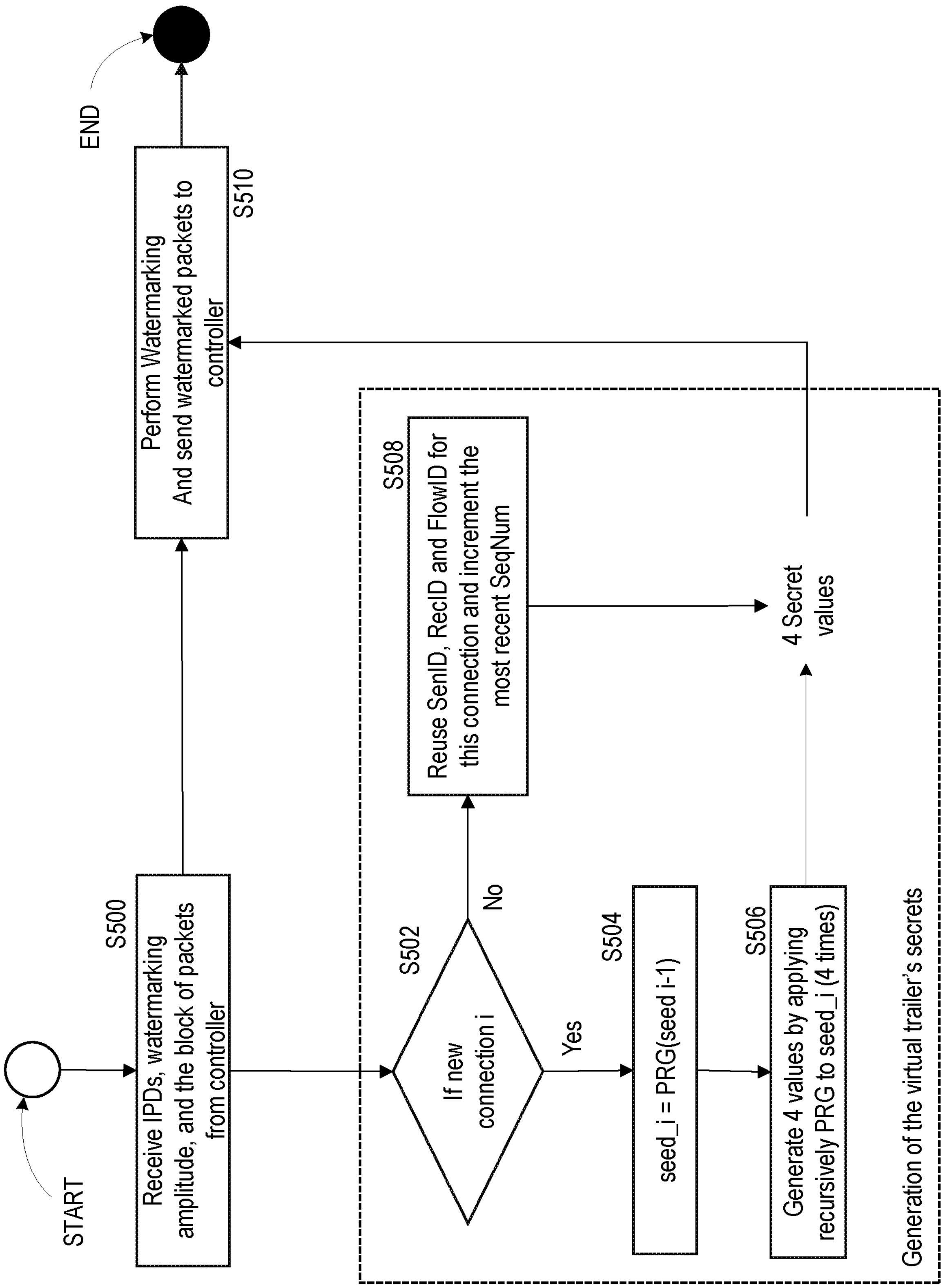
START
S500
Receive IPDs, watermarking amplitude, and the block of packets from controller
S502
If new connection i
Yes
No
S504
seed_i = PRG(seed i-1)
S506
Generate 4 values by applying recursively PRG to seed_i (4 times)
S508
Reuse SenID, RecID and FlowID for this connection and increment the most recent SeqNum
4 Secret values
Generation of the virtual trailer's secrets
S510
Perform Watermarking And send watermarked packets to controller
END

FIG. 9

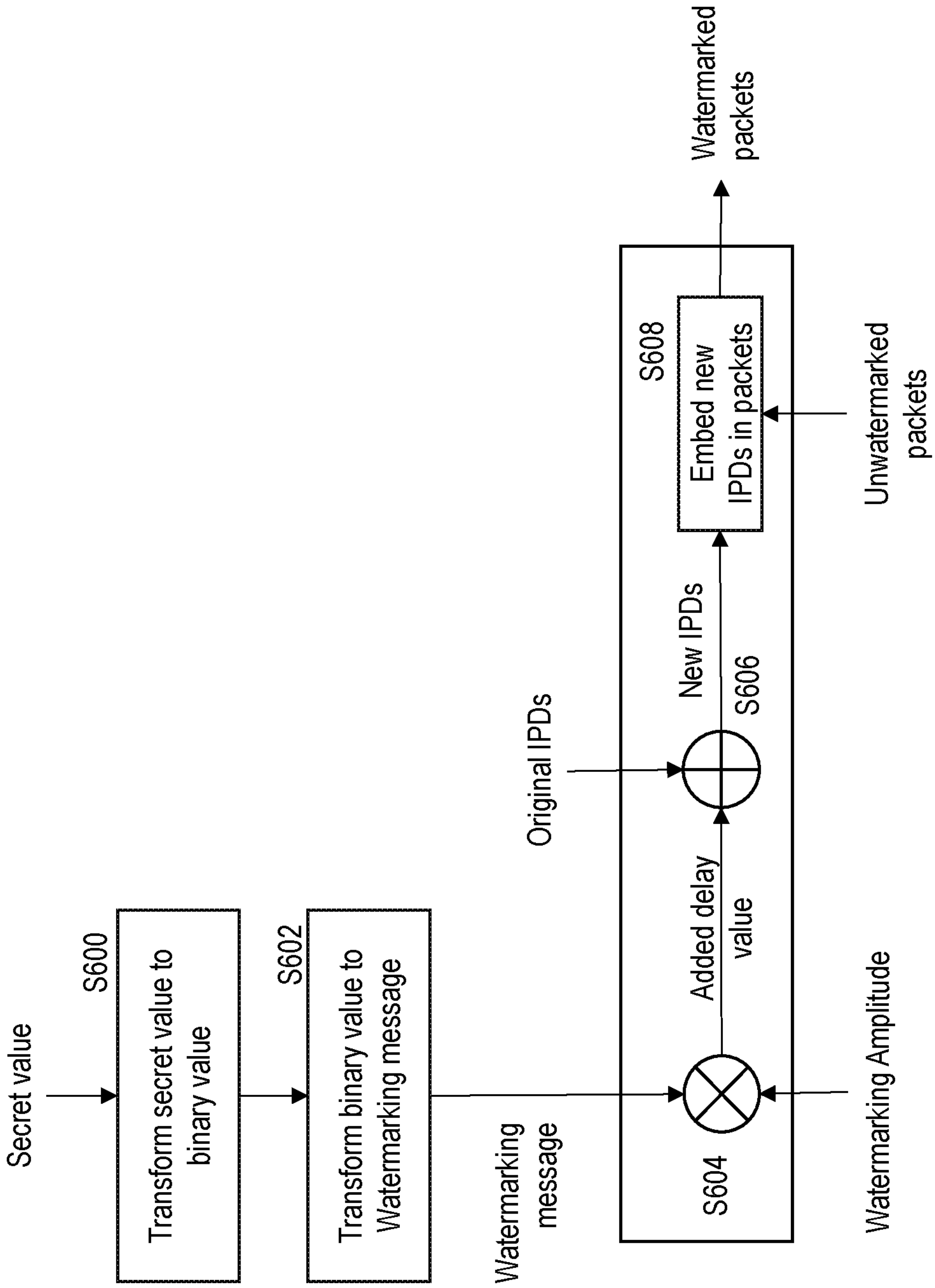
Secret value
S600
Transform secret value to binary value
S602
Transform binary value to Watermarking message
Watermarking message
S604
Watermarking Amplitude
Added delay value
Original IPDs
S606
New IPDs
S608
Embed new IPDs in packets
Unwatermarked packets
Watermarked packets

FIG. 10

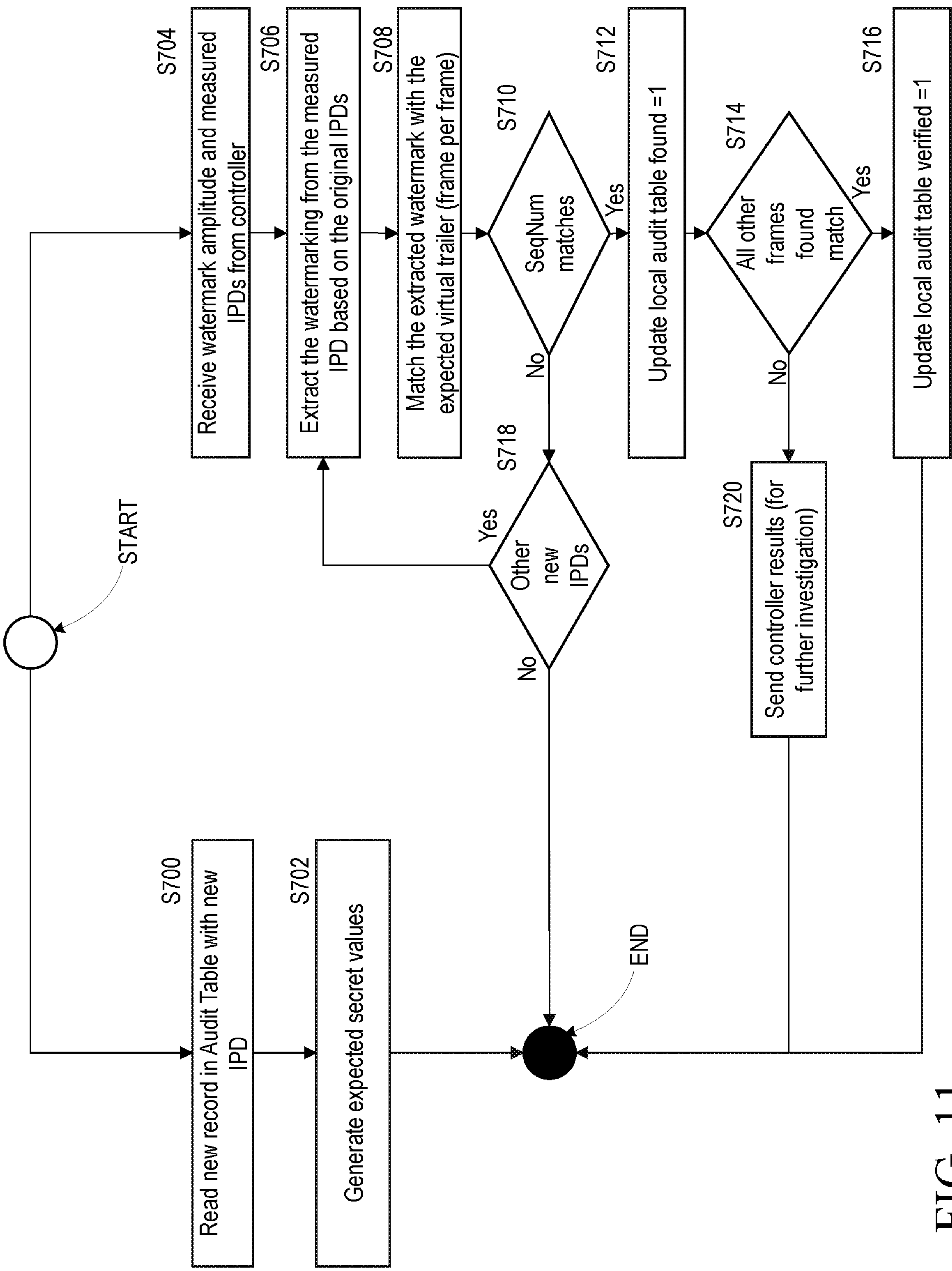
START
S700
Read new record in Audit Table with new IPD
S702
Generate expected secret values
S704
Receive watermark amplitude and measured IPDs from controller
S706
Extract the watermarking from the measured IPD based on the original IPDs
S708
Match the extracted watermark with the expected virtual trailer (frame per frame)
S710
SeqNum matches
Yes
No
S712
Update local audit table found =1
S714
All other frames found match
Yes
No
S716
Update local audit table verified =1
S718
Other new IPDs
Yes
No
S720
Send controller results (for further investigation)
END

FIG. 11

Conn.	IPDs	Seed	Expected Virtual Trailer				Verified	Found	Block-level attack Identification				Flow-level attack identification			
			SenID	RecID	FlowID	Seq Num			Inj.	Drop.	Reo.	Dup.	Inj.	Drop.	Reo.	Dup.
C1		Seed 1														
C2		Seed 2														
...	...	..														
C1		Seed 1														
C1		Seed 1														

FIG. 12

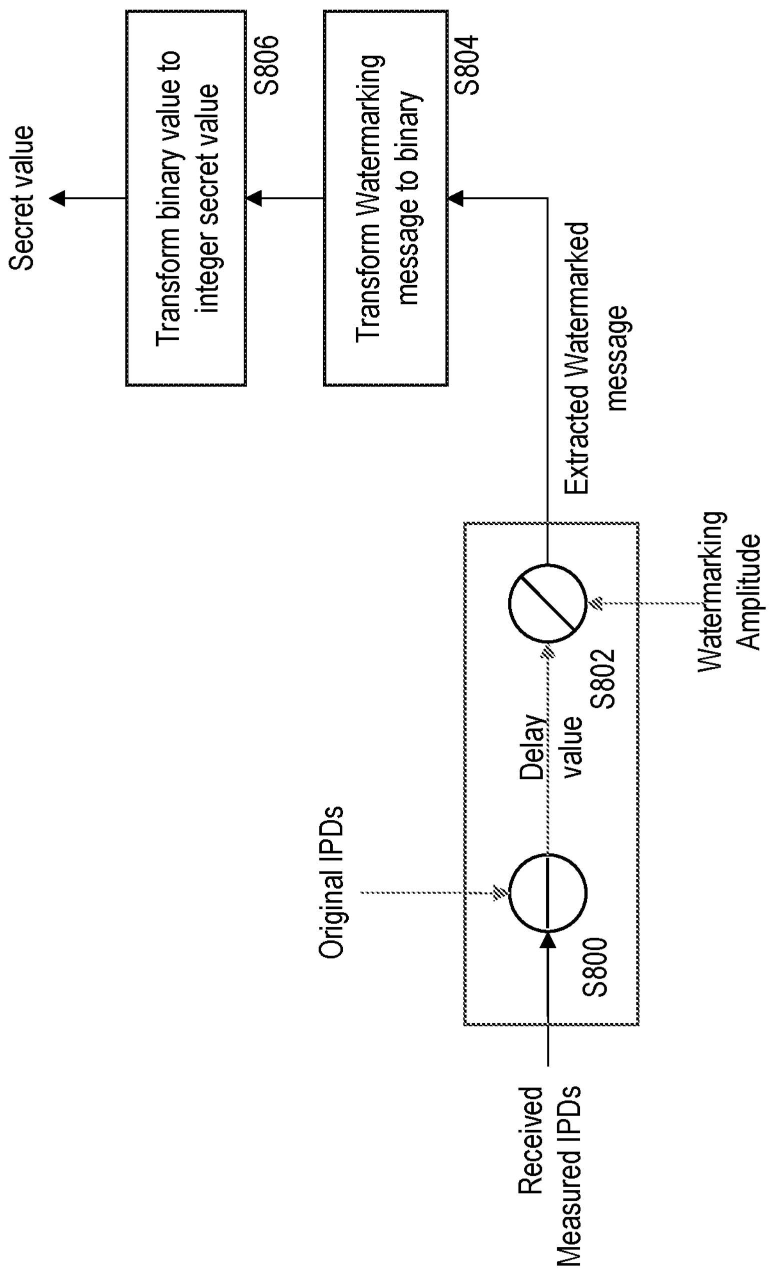
Received
Measured IPDs
Original IPDs
S800
Delay
value
S802
Watermarking
Amplitude
Extracted Watermarked
message
Transform Watermarking
message to binary
S804
Transform binary value to
integer secret value
S806
Secret value

FIG. 13

Connection	Secret Seed	Flow ID	Seq Num.	Fully Verified	Partially Verified	Detection	Identification					Found
							Inj.	Drop.	Reo.	Dup.	Skip.	
c1	Seed 1											
c2	Seed 2											
cn	Seed 3											

FIG. 14

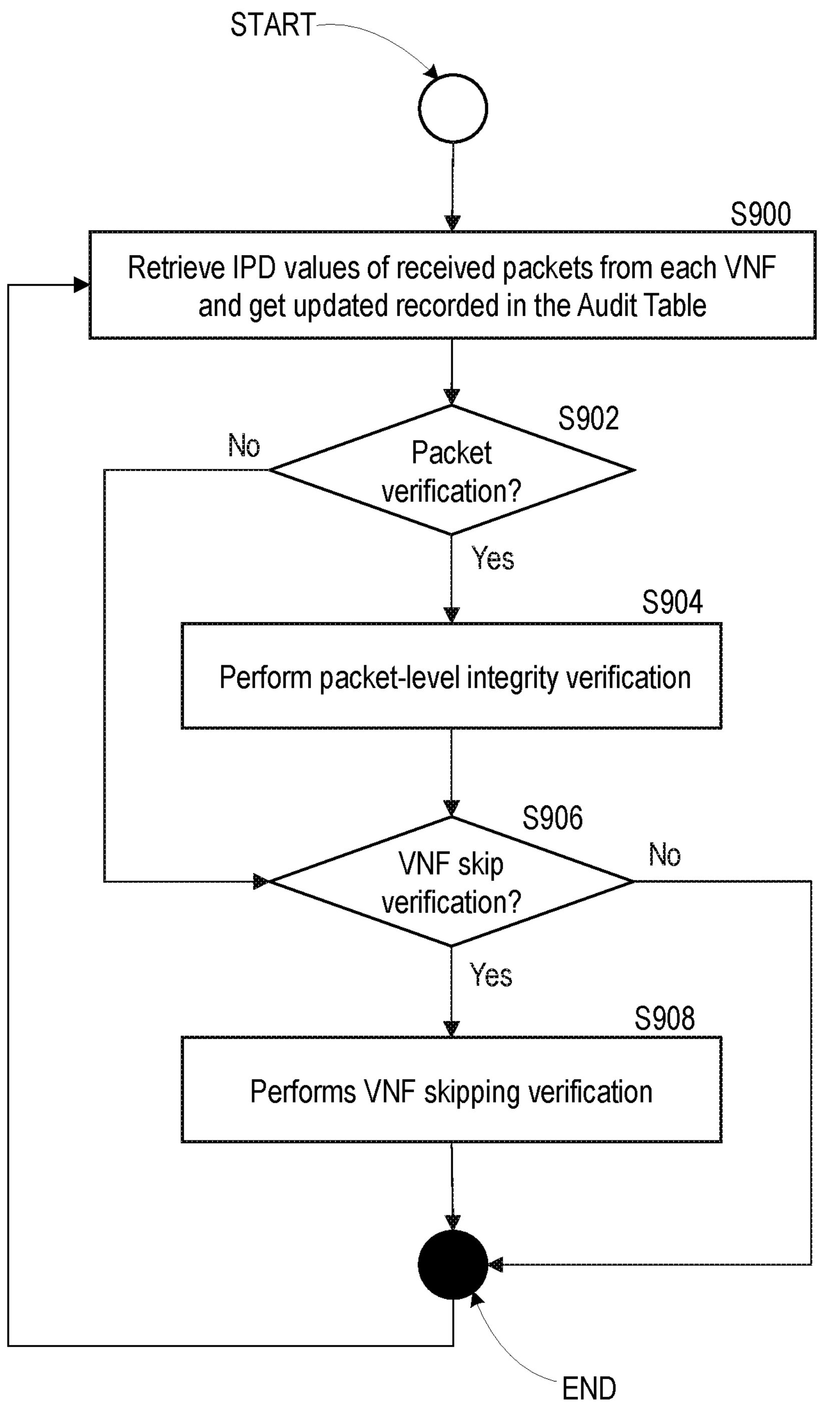
START
S900
Retrieve IPD values of received packets from each VNF and get updated recorded in the Audit Table
S902
Packet verification?
No
Yes
S904
Perform packet-level integrity verification
S906
VNF skip verification?
No
Yes
S908
Performs VNF skipping verification
END

FIG. 15

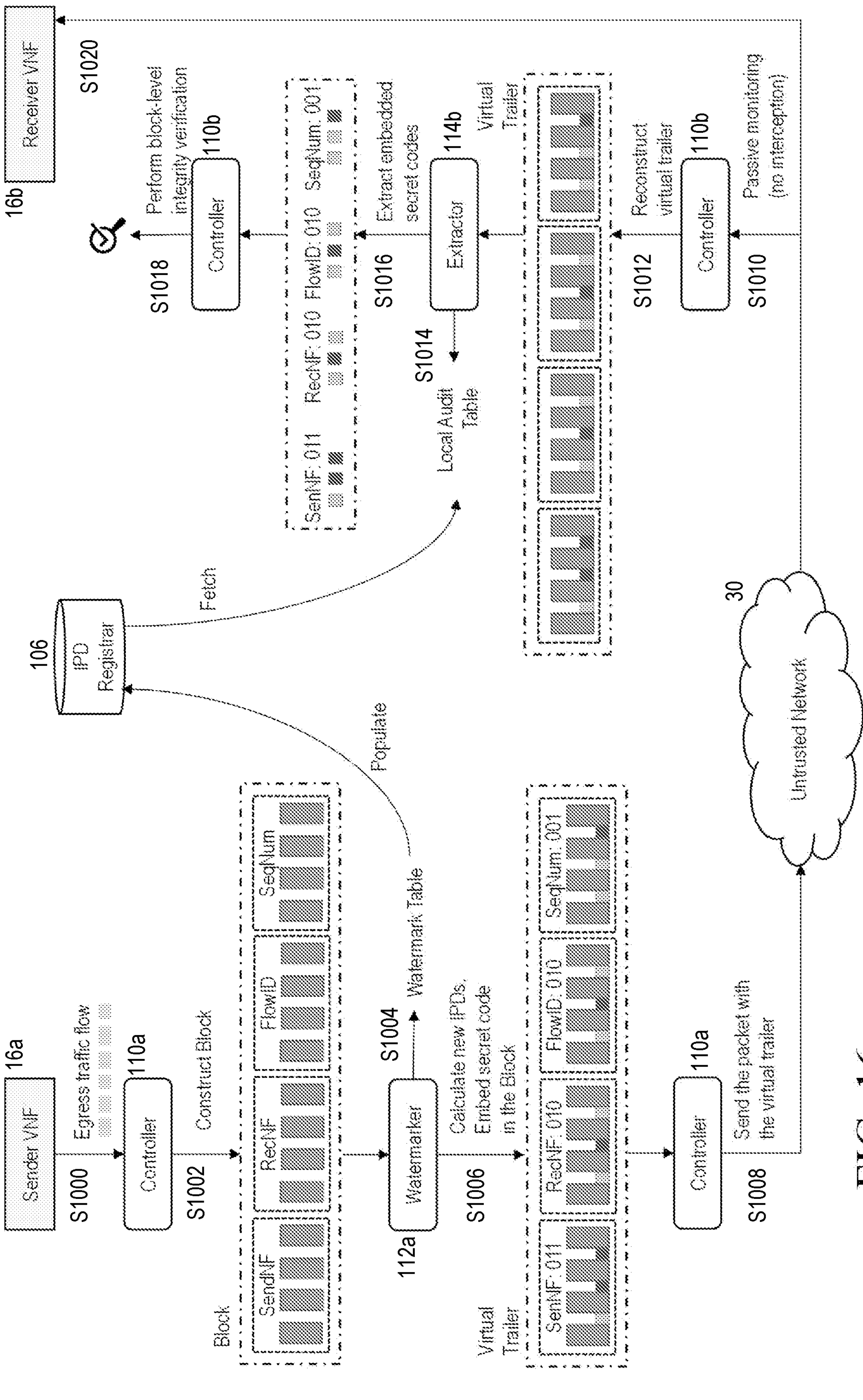
Sender VNF
16a
S1000
Egress traffic flow
Controller
110a
S1002
Construct Block
Block
SendNF
RecNF
FlowID
SeqNum
Watermarker
112a
S1004
Watermark Table
S1006
Calculate new IPDs, Embed secret code in the Block
Virtual Trailer
SenNF: 011
RecNF: 010
FlowID: 010
SeqNum: 001
Controller
110a
S1008
Send the packet with the virtual trailer
Populate
106
IPD Registrar
Fetch
30
Untrusted Network
S1010
Passive monitoring (no interception)
Controller
110b
S1012
Reconstruct virtual trailer
Virtual Trailer
Extractor
114b
S1014
Local Audit Table
S1016
Extract embedded secret codes
SenNF: 011
RecNF: 010
FlowID: 010
SeqNum: 001
Controller
110b
S1018
Perform block-level integrity verification
Receiver VNF
16b
S1020

FIG. 16

Connection	Secret	Calculated Virtual Trailer	Expected Virtual Trailer	Found	Identification				
					Inj.	Drop.	Reo.	Dup.	Skip.
C1	Seed 1	**[101, 011, 110, 101]**	[011, 010, 010, 001]	**True**	1				
...	...								

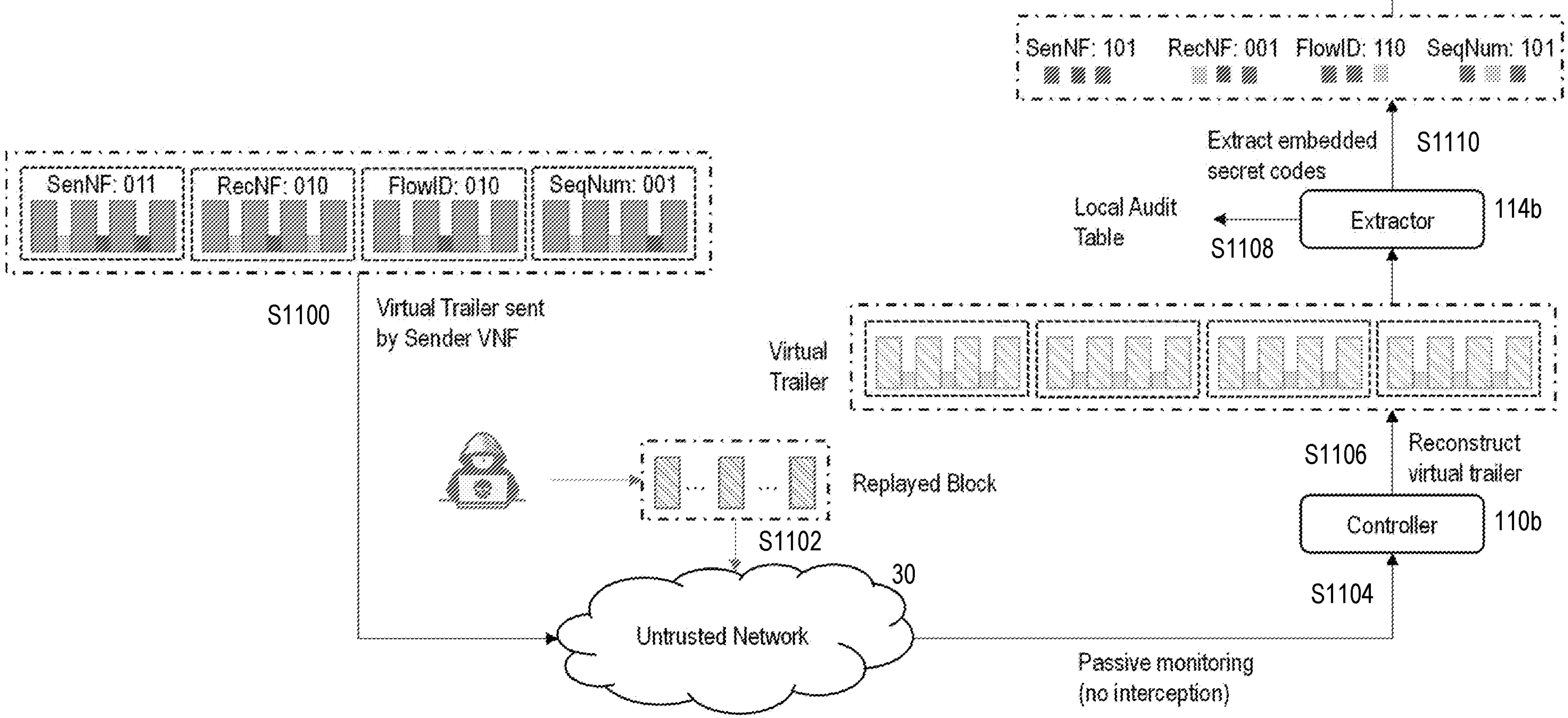

FIG. 17

Connection	Flow ID	Seq Num	Block Num.	Fully Verified	Partially Verified	Detection	Identification					Found
							Inj.	Drop.	Reo.	Dup.	Skip	
C1	010	001	1		True	True	1					
...												

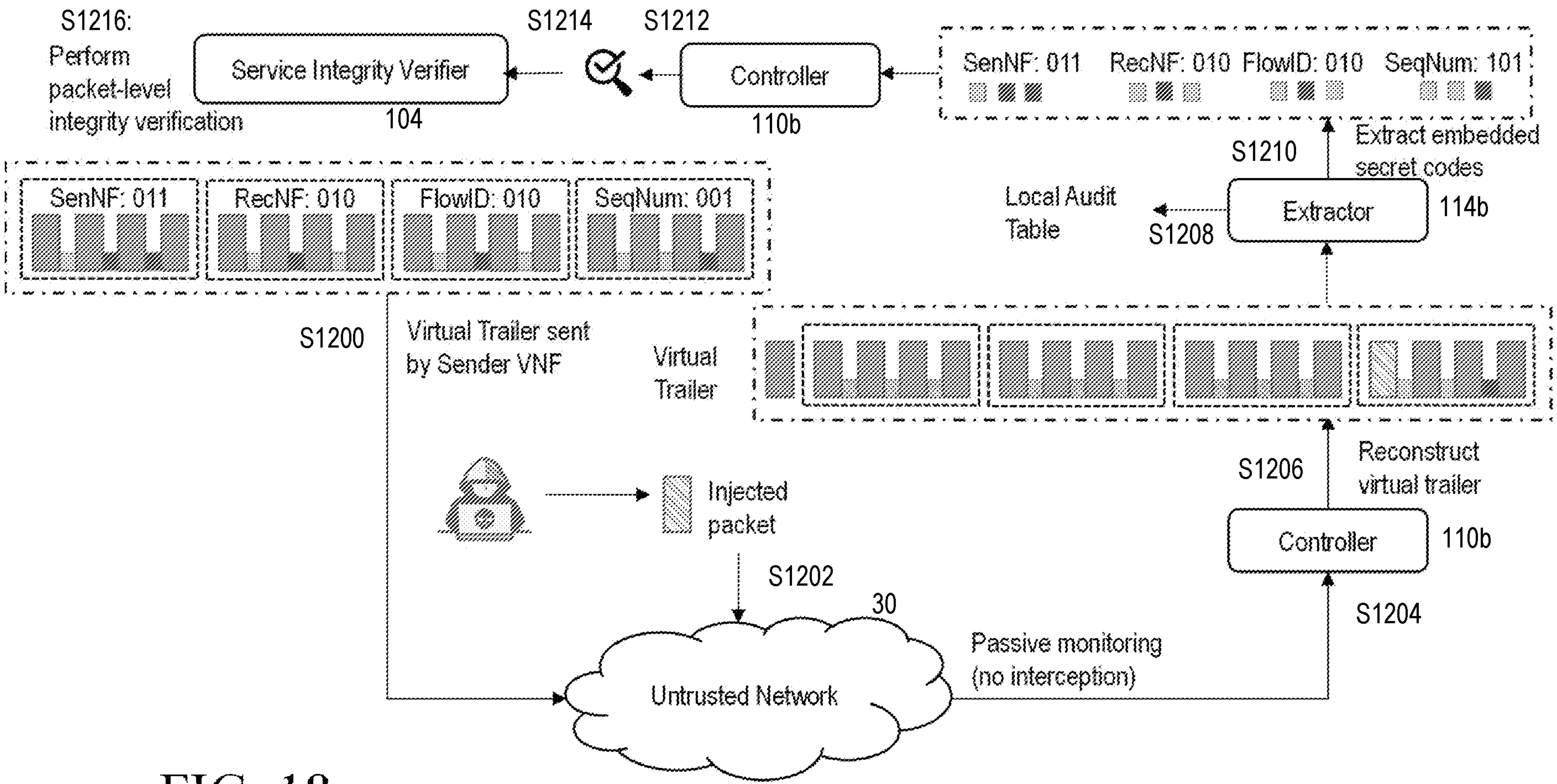

FIG. 18

INTERNATIONAL SEARCH REPORT

International application No

PCT/IB2023/054674

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/40
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, COMPENDEX, INSPEC, IBM-TDB, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	US 2008/080558 A1 (WANG XINYUAN [US] ET AL) 3 April 2008 (2008-04-03) abstract paragraph [0022] - paragraph [0031] figures 1, 2, 3 -----	1,11 2-10, 12-40
A	ALFONSO IACOVAZZI ET AL: "DROPWAT: an Invisible Network Flow Watermark for Data Exfiltration Traceback", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 26 May 2017 (2017-05-26), XP081276689, DOI: 10.1109/TIFS.2017.2779113 the whole document -----	1-40

-/--

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

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 January 2024	30/01/2024

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

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 11 503 069 B2 (AT & T IP I LP [US]) 15 November 2022 (2022-11-15) abstract figure 1a column 3, line 9 - column 5, line 64 -----	41-60

INTERNATIONAL SEARCH REPORT

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims;; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☒ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-40

Transmitting a secret value from the first agent node to the network node.

---

2. claims: 41-60

Detecting a network attack by an orchestrator

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/IB2023/054674

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008080558 A1	03-04-2008	US 2008080558 A1	03-04-2008
		US 2011069721 A1	24-03-2011
US 11503069 B2	15-11-2022	US 2021266344 A1	26-08-2021
		US 2023073668 A1	09-03-2023
		WO 2021168196 A1	26-08-2021