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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 February 2014 (27.02.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/031306 A2

(51) International Patent Classification:
H04L 9/32 (2006.01)

(21) International Application Number:
PCT/US2013/053274

(22) International Filing Date:
1 August 2013 (01.08.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/591,311 22 August 2012 (22.08.2012) US

(71) Applicant: **ALCATEL LUCENT** [FR/FR]; 3, avenue Octave Gréard, F-75007 Paris (FR).

(72) Inventor: **KOLESNIKOV, Vladimir, Y.**; 600-700 Mountain Avenue, Murray Hill, NJ 07974-0636 (US).

(74) Agent: **SANTEMA, Steven, R.**; Alcatel-lucent USA INC., Attention: Docket Administrator - Room 3B-212F, 600-700 Mountain Avenue, Murray Hill, NJ 07974-0636 (US).

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

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

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: MAC AGGREGATION WITH MESSAGE MULTIPLICITY FOR USE IN A MULTI-NODE DATA NETWORK

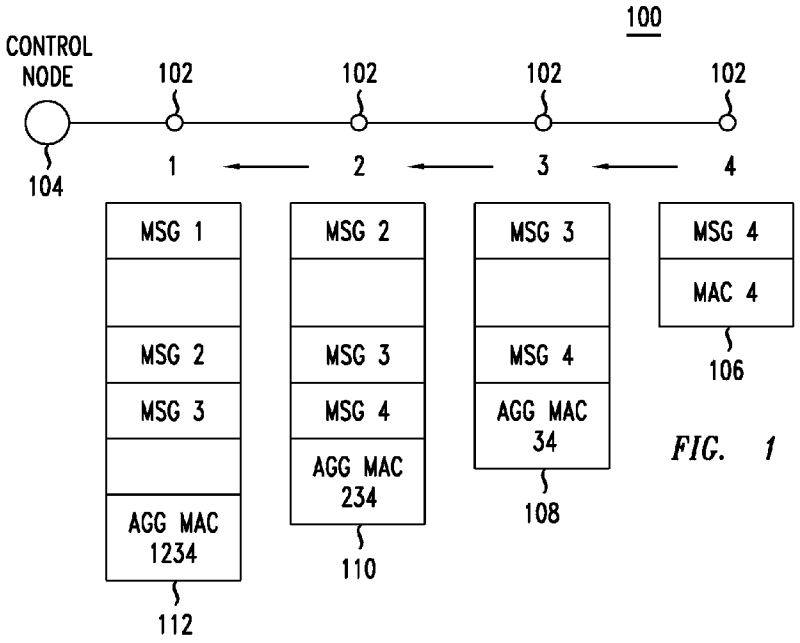


FIG. 1

(57) Abstract: A MAC aggregation technique utilizing a large field addition operation is disclosed. The large field addition operation defines the addition of two or more MACs mod p, where the two or MACs may comprise constituent MACs or aggregate MACs, and where p is a prime number that is large relative to the size of the MACs. The disclosed MAC aggregation technique yields an aggregate MAC much shorter than the concatenation of constituent MACs while achieving security even in the case where constituent MACs may be aggregated in duplicate.

WO 2014/031306 A2

MAC AGGREGATION WITH MESSAGE MULTIPLICITY FOR USE IN A MULTI-NODE DATA NETWORK

FIELD OF THE INVENTION

This invention relates generally to data security and, more particularly, to techniques for aggregation of message authentication codes for use in multi-node data networks.

BACKGROUND OF THE INVENTION

A message authentication code (MAC, or MAC "tag") is a digital signal sequence used for authenticating a message exchanged between a sender and receiver each having a shared secret "key." In a typical MAC authentication scenario, the sender of a message runs it through a cryptographic hash function having the message and the secret key as inputs, yielding a MAC tag as an output. The sender then sends the message and the tag to the receiver. The receiver runs the received message through the same MAC algorithm using the same key, yielding a second MAC tag. The receiver then compares the tag generated by itself to the tag received from the sender and if they are the same, the message is deemed authentic. If they are not the same, it is an indication that the message was altered or compromised in some manner during the exchange.

Although MAC tags are relatively short, they can represent a significant overhead in data networks that include multiple nodes generating multiple messages and hence, multiple different MAC tags. This overhead is most pronounced when the MAC tags are larger than the messages themselves. As one example, distributed sensor networks (such as "Smart Grid" energy metering networks) rely on a multitude of deployed cheap sensors to report measurements, such as temperature, electricity consumption, etc. This data is transmitted hop-by-hop over multiple nodes and needs to be authenticated and verified by a central node. The measurement data can be quite small (e.g., on the order of 10-15 bits), whereas the typical MAC tag is 128 bits which is roughly an order of magnitude larger than the data it authenticates.

To mitigate the overheads inherent in multi-node data networks, the concept of aggregate MACs (comprising an aggregation of multiple MAC tags into a shorter tag) has been proposed by the cryptographic community. Aggregate MACs are much

shorter than the concatenation of constituent MACs, thus greatly reducing the network overhead, yet can still be verified by a central node that shares a distinct key with each sending node. However, existing MAC aggregation techniques do not guarantee security of the aggregate MAC if more than two identical messages are aggregated together. This is a significant impediment in the practice of secure networking, since a plurality of message delivery and routing protocols, such as flooding, rely on (or at least allow) duplicate messages being authenticated. Accordingly, there is a need for improved MAC aggregation techniques that not only greatly reduce the overhead of a multi-node data network, but that guarantee security even in the case where constituent MACs may be aggregated in duplicate. Embodiments of the present invention are directed to addressing this need.

SUMMARY OF THE INVENTION

Generally, embodiments of the present invention describe an improved MAC aggregation technique that not only greatly reduces the network overhead, but also guarantees security even in the case where constituent MACs may be aggregated in duplicate.

In one embodiment, there is provided a MAC aggregation method carried out by a sending node in a communication network including a plurality of successive sending nodes operably linked to a control node. The method comprises receiving from a previous node a first message authentication code (MAC), comprising one of a constituent MAC or an aggregated MAC; generating or receiving a second message authentication code (MAC), comprising one of a constituent MAC or an aggregated MAC; executing a MAC aggregation function to combine the first and second MACs, yielding an output aggregate MAC shorter than the concatenation of the first and second MACs, wherein the MAC aggregation function comprises a large field addition operation performed on the first and second MACs; and sending the output aggregate MAC to a next consecutive node for use in deriving a next instance of aggregate MAC, or if the next consecutive node is the control node, for use in authenticating data originated from the sending nodes.

In one embodiment, there is provided an apparatus for performing MAC aggregation of constituent MACs, the apparatus at a sending node comprising an input interface, an output interface, a memory and a processor. The processor is operably coupled to the input interface, output interface and memory and configured to: (a) receive from a previous node a first message authentication code (MAC), the first MAC comprising one of a constituent MAC or an aggregated MAC; (b) generate a second message authentication code (MAC), the second MAC comprising one of a constituent MAC or an aggregated MAC; (c) execute a MAC aggregation function to combine the first and second MACs, yielding an output aggregate MAC shorter than the concatenation of the first and second MACs, wherein the MAC aggregation function comprises a large field addition operation performed on the first and second MACs; and (d) send the aggregate MAC to a next consecutive node for use in deriving a next instance of aggregate MAC, or if the next consecutive node is the control node, for use in authenticating data originated from the sending nodes.

In one embodiment, there is provided a MAC aggregation method carried out by a control node in a communication network including a plurality of successive sending nodes operably linked to a control node. The method comprises receiving from a previous node a message string including payload data from a plurality of sending nodes and at least one aggregate MAC computed by aggregating a plurality of constituent MACs generated by the respective sending nodes; executing a MAC computation function to recompute the constituent MACs from the payload data and one or more shared secret keys, yielding a plurality of recomputed constituent MACs; executing a MAC aggregation function on the recomputed constituent MACs, yielding a computed aggregate MAC shorter than the concatenation of the recomputed constituent MACs, wherein the MAC aggregation function comprises a large field addition operation performed on the recomputed constituent MACs; and comparing the computed aggregate MAC to the received aggregate MAC to determine an authentication status of the received aggregate MAC.

In one embodiment, there is provided an apparatus for performing MAC aggregation of constituent MACs, the apparatus at a control node comprising a communication interface, a memory and a processor. The processor is operably coupled to the communication interface and memory and configured to: (a) receive from a previous node a message string including payload data from a plurality of sending nodes and at least one aggregate MAC computed by aggregating a plurality of constituent MACs generated by the respective sending nodes; (b) execute a MAC computation function to recompute the constituent MACs from the payload data and one or more shared secret keys, yielding a plurality of recomputed constituent MACs; (c) execute a MAC aggregation function on the recomputed constituent MACs, yielding a computed aggregate MAC shorter than the concatenation of the recomputed constituent MACs, wherein the MAC aggregation function comprises a large field addition operation performed on the recomputed constituent MACs; and (d) compare the computed aggregate MAC to the received aggregate MAC to determine an authentication status of the received aggregate MAC.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other advantages of the invention will become apparent upon reading the following detailed description and upon reference to the drawings in which:

FIG. 1 illustrates an exemplary message sequence in a multi-node data network utilizing MAC aggregation;

FIG. 2 illustrates a MAC aggregation function according to the prior art;

FIG. 3 illustrates an exemplary MAC aggregation function according to embodiments of the present invention;

FIG. 4 is a block diagram illustrating a logical hardware configuration of a generic sending node in a multi-node data network;

FIG. 5 is a block diagram illustrating a logical hardware configuration of a central node in a multi-node data network;

FIG. 6 is a flowchart showing steps performed by a generic sending node according to embodiments of the present invention; and

FIG. 7 is a flowchart showing steps performed by a central node according to embodiments of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

FIG. 1 illustrates an exemplary message sequence in a multi-node data network 100 utilizing MAC aggregation. The multi-node data network 100 includes a plurality of sending nodes 102 (four shown) operably connected to a control node 104. For convenience, the sending nodes 102 are identified as "node 1," "node 2," "node 3" and "node 4" in relation to their proximity to the control node 102. The sending nodes may comprise, generally, any system, server, router, application or function adapted to communicate data with other nodes of the multi-node data network 100. The control node may comprise, generally, any system, server, router, application or function adapted to receive and authenticate data received from the sending nodes.

Most typically, the sending nodes and the control node reside in separate physical devices or structures distributed throughout the network 100. For example and without limitation, the sending nodes may comprise or be associated with multiple sensors deployed throughout a sensor network (such as a "Smart Grid"

energy metering network) and the control node may comprise an energy monitoring and/or billing system, server or application that receives and authenticates the sensor data. As will be appreciated, the sending nodes and the control node are functional elements that may embody generally any data networking modality, may be arranged in alternative network topologies and may be distributed and/or combined among generally any combination of physical devices or structures.

In the example network of FIG. 1, the sending nodes 4, 3, 2, 1 are linked together in sequence and communicate successively ("hop-by-hop") to the control node. For purposes of illustration, it is presumed that the outermost sending node (node 4) has initiated a message ("msg 4") to be transported through the multi-node network 100. Node 4 develops a message string 106 including the message ("msg 4") and a message authentication code ("MAC 4") and sends the message string 106 to the next successive node (node 3). In one embodiment, MAC 4 is computed in conventional fashion, by running msg 4 through a cryptographic hash function having msg 4 and a shared secret key (shared by node 4 and the central node) as inputs, yielding the tag "MAC 4" as an output.

Node 3 generates its own message ("msg 3") and computes its own associated message authentication code ("MAC 3") and, having received message string 106 including MAC 4, executes a MAC aggregation function to combine MAC 3 and MAC 4 yielding an aggregate MAC ("$AggMAC_{34}$"). The MAC aggregation function will be described in greater detail in relation to FIG. 3. Node 3 develops a message string 108 including its own message ("msg 3") and the message from node 4 ("msg 4") and the aggregate MAC ("$AggMAC_{34}$"), and sends the message string 108 to the next successive node (node 2).

Node 2 generates its own message ("msg 2") and associated message authentication code ("MAC 2") and, having received message string 108 including $AggMAC_{34}$, executes a MAC aggregation function to derive a next instance of aggregate MAC ("$AggMAC_{234}$") based on a combination of MAC 2 and $AggMAC_{34}$. Node 2 develops a message string 110 including its own message ("msg 2") and the messages from node 3 ("msg 3") and node 4 ("msg 4") and the aggregate MAC ("$AggMAC_{234}$"), and sends the message string 110 to the next successive node (node 1).

Node 1 generates its own message ("msg 1") and associated message authentication code ("MAC 1") and, having received message string 110 including $AggMAC_{234}$, executes a MAC aggregation function to derive a next instance of aggregate MAC ("$AggMAC_{1234}$") based on a combination of MAC 1 and $AggMAC_{234}$. Node 1 develops a message string 112 including its own message ("msg 1") and the messages from node 2 ("msg 2"), node 3 ("msg 3") and node 4 ("msg 4") and the aggregate MAC ("$AggMAC_{1234}$"), and sends the message string 110 to the control node 104.

The control node 104, having received the message string 110 including payload data from the previous nodes and possession of the respective shared secret keys, recomputes the constituent MACs ("MAC 1," "MAC 2," "MAC 3" and "MAC 4"). Further, having knowledge of the network topology, the control node executes a MAC aggregation function on the recomputed constituent MACs (i.e., in the same sequence executed by the constituent nodes) to produce a computed aggregate MAC ("$AggMAC_{1234}$"). The control node then compares the computed aggregate MAC to the received aggregate MAC to determine an authentication status of the received aggregate MAC. If the computed aggregate MAC is equal to the received aggregate MAC, the received aggregate MAC is deemed valid and the constituent MACs from which the aggregate MAC was derived are known to be valid. Conversely, if the computed aggregate MAC and received aggregate MAC are not the same, it is an indication that some form of data impairment occurred in the network, either accidentally or maliciously.

In MAC aggregation techiques of the prior art, if at least any two of the messages included in the aggregation are equal to each other, and are MACed with the same key, the resulting aggregated MAC scheme is not guaranteed to be secure. That is, it would be possible for an attacker to insert messages of his choice into the AggMAC, which will be later accepted by a control node as genuine. In contrast, according to embodiments of the present invention, the aggregate MAC is constructed in a manner where such an attack would not be possible.

Referring to FIG. 2, a MAC aggregation function according to the prior art comprises a bitwise exclusive OR (XOR) operation on successive MACs 202 (e.g., MAC 1, MAC 2...MAC n), yielding an aggregated MAC 204. A bitwise XOR

function operates on two bit patterns of equal length and performs the logical XOR operation on each pair of corresponding bits. The resulting bit pattern is of equal length and the bit in any position is 1 if the corresponding bits are different (i.e., if only one of the first bit or second bit is 1 and the other of the first and second bit is 0) or 0 if the corresponding bits are the same (i.e., both are 1 or both are 0). The MAC aggregation function shown in FIG. 2 will therefore result in an aggregate MAC 204 that is equal in length to the constituent MACs 202 with a bit pattern derived by the constituent MACs. It is apparent that should any of the constituent MACs be inserted into the aggregated MAC in duplicate, either accidentally or maliciously, the resulting aggregate MAC will cancel out the authenticator for the corresponding message, and hence the message can not be verified by the control node.

Now referring now to FIG. 3, a MAC aggregation function according to embodiments of the present invention comprises a “large field addition" operation on successive MACs 302 (e.g., MAC 1, MAC 2...MAC n), yielding an aggregated MAC 304. In one embodiment, the large field addition operation defines the addition of two or more successive MACs (e.g., comprising either constituent MACs or aggregated MACs) mod p, where p is a prime number that is large relative to the size of the MACs. For example, to perform a large field addition of two message sets M_1, M_2 with AggMACs τ_1, τ_2, we simply compute new AggMAC (M_1, M_2) = $\tau_1 + \tau_2$ (mod p). As will be understood to those skilled in the art, given two positive numbers, a and p, a modulo p (abbreviated as a mod p) is the remainder of the Euclidean division of a by p. For instance, the expression "5 mod 4" would evaluate to 1 because 5 divided by 4 leaves a remainder of 1, while "9 mod 3" would evaluate to 0 because the division of 9 by 3 leaves a remainder of 0.

In an exemplary large field addition operation according to embodiments of the present invention, we choose a modulo value p that is sufficiently large relative to the size of the component MACs so as to provide computational security. For example and without limitation, where n is the length of a MAC tag, a prime number $p > 2^n$ is sufficiently large to provide security. Generally, security may be achieved for any prime number p > n; and even for selected p less than n in instances where p is sufficiently large. For example, considering the typical MAC tag is 128 bits, p on the order of 100 bits or greater should provide sufficient security.

FIG. 4 and FIG. 5 illustrate a logical hardware configuration of a generic sending node 400 and a central node 500, respectively, in a multi-node data network according to embodiments of the present invention.

The generic node 400 comprises any system, server, application or function adapted to communicate data with other nodes of a multi-node data network 100. For example and without limitation, generic nodes may collectively comprise or be associated with multiple sensors deployed throughout a sensor network (such as a "Smart Grid" energy metering network), wherein each generic node nominally communicates payload data (e.g., sensor data or the like) and MAC tags to other nodes. The generic nodes may be arranged in virtually any type of network configuration, including linear, tree or star-based architectures or combinations thereof.

The generic node 400 includes a processor 402 and memory 404 for effecting communication transactions with other nodes and for generating MAC tags. The processor and memory are logically connected to an input interface 406 that receives inputs (e.g., payload data and MAC tags) from other node(s); and an output interface 408 that sends output (e.g., payload data and MAC tags) to other node(s). The logical hardware configuration further includes a MAC generation process module 410 for generating MAC tags. The MAC generation process module 410 may be implemented, for example and without limitation, by the processor 402 executing program code (e.g., including but not limited to operating system firmware/software and application software) stored in the memory 404 to generate MAC tags.

According to embodiments of the present invention, the MAC generation process module 410 is operable to generate two types of MAC tags: a "standard" MAC 412 (e.g., an AES-based tag, typically 128 bits) derived from the payload and a shared secret key; and an aggregate MAC ("AggMAC") derived by combining the MAC 412 with a MAC or AggMAC from a previous node using a MAC aggregation function. For example and without limitation, the AggMAC may be derived by a large field addition operation such as described in relation to FIG. 3. As will be appreciated, the MAC generation process module 410 is a logical hardware component that may embody firmware, microchips (e.g., ASICs), software executable on a hardware device, hardware, specialized hardware, and/or the like, may be

implemented in one or more physical devices and may implement one or more communication technologies including wired, wireless or packet-based links.

The control node 500 comprises any system, server, application or function adapted to receive, process and authenticate data from other nodes of a multi-node data network 100. For example and without limitation, the control node 500 may comprise an energy monitoring and/or billing system, server or application that receives, processes and authenticates payload data (e.g., sensor data or the like) and MAC tags received from one or more nodes.

The control node 500 includes a processor 502 and memory 504 for effecting communication with and for processing and authenticating data received from one or more generic nodes. The processor and memory are logically connected to a communication interface 506 that receives inputs (e.g., payload data and MAC tags) from other node(s); and may send outputs, where appropriate, to one or more node(s). The logical hardware configuration further includes a MAC generation process module 510 and a node authentication process module 512. The MAC generation process module 510 and node authentication process module 512 may be implemented, for example and without limitation, by the processor 502 executing program code (e.g., including but not limited to operating system firmware/software and application software) stored in the memory 504.

According to embodiments of the present invention, the MAC generation process module 510 operates (similarly as the MAC generation process module 410 of the generic nodes) to generate two types of MAC tags: "standard" MACs (comprising for example recomputed constituent MACs derived from the payload data of constituent nodes) and an aggregate MAC ("computed AggMAC") derived by executing a MAC aggregation function on the recomputed constituent MACs. For example and without limitation, the AggMAC may be derived by a large field addition operation such as described in relation to FIG. 3.

In one embodiment, the node authentication process module 512 operates to authenticate respective "standard" MACs and/or received AggMAC. This is accomplished by comparing the computed AggMAC to the received AggMAC and, if they are the same, the received AggMAC is deemed valid and the constituent MACs from which the aggregate MAC was derived are known to be valid.

As will be appreciated, the MAC generation process module 510 and the node authentication process module 512 are logical hardware components that may embody firmware, microchips (e.g., ASICs), software executable on a hardware device, hardware, specialized hardware, and/or the like, may be implemented in one or more physical devices and may implement one or more communication technologies including wired, wireless or packet-based links.

FIG. 6 is a flowchart showing steps performed by a generic node 400 (FIG. 4) according to embodiments of the present invention. For convenience, the steps of FIG. 6 will be described with reference to a generic node (e.g., node 1, 2, 3, 4) operating within a linear network topology such as described in relation to FIG. 1. As will be appreciated, however, the steps of FIG. 6 may be performed by generic nodes in virtually any type of network topology.

At step 610, the generic node may optionally receive input from a previous node depending on its location within the network topology. For example, with reference to FIG. 1, intermediate nodes 3, 2 and 1 receive input from their respective previous nodes 4, 3 and 2; whereas node 4 is the initiating node and does not receive input from any previous nodes. In one embodiment, the input received by the intermediate nodes includes payload data from one or more previous nodes and a previous MAC (which depending on location within the network topology, may comprise a "standard" MAC or an aggregate MAC). For example, with reference to FIG. 1, node 3 receives payload data (msg 4) and a standard MAC (MAC 4) from node 4; node 2 receives payload data (msg 3, msg 4) and an aggregate MAC ($AggMAC_{34}$) from node 3; and node 1 receives payload data (msg 2, msg 3 and msg 4) and an aggregate MAC ($AggMAC_{234}$) from node 2.

At step 620, the generic node generates its own payload data and standard MAC. For example, with reference to FIG. 1, node 3 generates msg 3, node 2 generates msg 2 and node 1 generates msg 1; and each respective node generates its own standard MAC (e.g., using standard AES-based hash algorithms) based on its payload data and a shared secret key, to yield MAC 3, MAC 2 and MAC 1, respectively.

At step 630, the generic node generates an aggregate MAC, based on the MAC (or aggregate MAC) received from the previous node and the standard MAC

generated by itself. For example, with reference to FIG. 1, node 3 generates $AggMAC_{34}$ based on an aggregation of MAC 3 with MAC 4; node 2 generates $AggMAC_{234}$ based on an aggregation of MAC 2 with $AggMAC_{34}$; and node 1 generates $AggMAC_{1234}$ based on an aggregation of MAC 1 with $AggMAC_{234}$. For example and without limitation, the respective AggMACs may be derived by a large field addition operation such as described in relation to FIG. 3.

At step 640, the generic node creates a message string including payload data from itself (and, depending on its location within the network topology, from one or more previous nodes) and the aggregate MAC generated by itself. For example, with reference to FIG. 1, node 3 generates message string 108, node 2 generates message string 110 and node 1 generates message string 112, each of the message strings 108, 110, 112 including an aggregate MAC generated by the corresponding node.

Then, at step 650, the generic node sends the message string to the next consecutive node. For example, with reference to FIG. 1, node 3 sends message string 108 to node 2, node 2 sends message string 110 to node 1 and node 1 sends message string 112 to the control node. Thereafter, at each consecutive node, the received aggregate MAC is used to derive a next instance of aggregate MAC, or if the next consecutive node is the control node, it is used to authenticate data originated from the respective nodes, for example, such as described in relation to FIG. 5.

FIG. 7 is a flowchart showing steps performed by a control node 500 (FIG. 5) according to embodiments of the present invention. For convenience, the steps of FIG. 7 will be described with reference to the control node 104 operating within a linear network topology such as described in relation to FIG. 1. As will be appreciated, however, the steps of FIG. 7 may be performed by control node(s) in virtually any type of network topology.

At step 710, the control node receives a message string from a previous node. In one embodiment, the received message string includes payload data from one or more previous nodes and a previous MAC (which depending on the previous node's location within the network topology, may comprise a "standard" MAC or an aggregate MAC). For example, with reference to FIG. 1, the control node receives a message string 112 from node 1. The message string 112 includes payload data from

4 previous nodes (msg 1, msg 2, msg 3 and msg 4) and an aggregate MAC ($AggMAC_{1234}$) from node 1.

At step 720, the control node recomputes constituent MACs of the previous nodes that contributed to the message string. For example, with reference to FIG. 1, the control node recomputes constituent MACs of nodes 1, 2, 3 and 4 based on the payload data (msg 1, msg 2, msg 3 and msg 4) and respective shared secret keys associated with the constituent nodes, yielding recomputed MACs (MAC 1, MAC 2, MAC 3 and MAC 4).

At step 730, the control node computes an aggregate MAC ("computed AggMAC") derived by executing a MAC aggregation function on the recomputed constituent MACs. For example, with reference to FIG. 1, the control node computes $AggMAC_{1234}$ by successively executing a MAC aggregation function on recomputed MACs (MAC 1, MAC 2, MAC 3 and MAC 4) of the constituent nodes. For example and without limitation, the computed AggMAC may be derived by a large field addition operation such as described in relation to FIG. 3.

At step 740, the control node compares the computed AggMAC to the received AggMAC to determine an authentication status of the received AggMAC. If the computed and received AggMACs are the same, the process proceeds to step 780 whereby the received AggMAC is authenticated as valid and the constituent MACs contributing to the received AggMAC are known to be valid. But if they differ, the control node knows that some form of data impairment occurred in the network and the message string is deemed to have been compromised at step 750.

FIGS. 1-7 and the foregoing description depict specific exemplary embodiments of the invention to teach those skilled in the art how to make and use the invention. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The present invention may be embodied in other specific forms without departing from the scope of the invention which is indicated by the appended claims. All changes that come within the meaning and range of equivalency of the claims are to be embraced within their scope.

For example, embodiments of the MAC aggregation function has been described herein with reference to a linear network topology and a MAC aggregation function based on a large field addition operation, whereby MAC aggregation is

accomplished by adding in a sufficiently large group the individual MACs of constituent nodes to the aggregate in sequential fashion. However, it will be appreciated that embodiments of the present invention may be implemented in any number of alternative network topologies and/or with any number of alternative MAC aggregation operations.

For example and without limitation, embodiments described herein may be implemented in a "tree" topology, whereby MAC aggregation is accomplished in at least one instance by a gateway node combining two aggregate MACs received from different branches; or a combination of tree topology with linear branches.

Moreover, MAC aggregation may be accomplished by a much larger class of functions, instead of just a group addition of the components.

WHAT IS CLAIMED IS:

1. In a communication network including a plurality of successive sending nodes operably linked to a control node, a method, carried out by at least one sending node, comprising:

receiving from a previous node a first message authentication code (MAC), the first MAC comprising one of a constituent MAC or an aggregated MAC;

generating or receiving a second message authentication code (MAC), the second MAC comprising one of a constituent MAC or an aggregated MAC;

executing a MAC aggregation function to combine the first and second MACs, yielding an output aggregate MAC shorter than the concatenation of the first and second MACs, wherein the MAC aggregation function comprises a large field addition operation performed on the first and second MACs; and

sending the output aggregate MAC to a next consecutive node for use in deriving a next instance of aggregate MAC, or if the next consecutive node is the control node, for use in authenticating data originated from the sending nodes.

2. The method of claim 1, wherein the large field addition operation defines the equation $\tau_1 + \tau_2 \pmod{p}$, where τ_1 is the first MAC, τ_2 is the second MAC, and p is a prime number that is large relative to the size of the first and second MACs.

3. The method of claim 1, wherein the first MAC comprises:

a constituent MAC, if the previous node is a first successive sending node; or

an aggregated MAC, if the previous node is an intermediate sending node.

4. The method of claim 1, wherein the second MAC comprises a duplicate of the first MAC.

5. Apparatus for performing MAC aggregation of constituent MACs, in accordance with a communication network including a plurality of successive sending nodes operably linked to a control node, the apparatus at at least one sending node comprising:

an input interface;

an output interface;

a memory; and

at least one processor operably coupled to the input interface, output interface and memory and configured to:

(a) receive from a previous node a first message authentication code (MAC), the first MAC comprising one of a constituent MAC or an aggregated MAC;

(b) generate or receive a second message authentication code (MAC); the second MAC comprising one of a constituent MAC or an aggregated MAC;

(c) execute a MAC aggregation function to combine the first and second MACs, yielding an output aggregate MAC shorter than the concatenation of the first and second MACs, wherein the MAC aggregation function comprises a large field addition operation performed on the first and second MACs; and

(d) send the aggregate MAC to a next consecutive node for use in deriving a next instance of aggregate MAC, or if the next consecutive node is the control node, for use in authenticating data originated from the sending nodes.

6. The apparatus of claim 5, wherein the large field addition operation defines the equation $\tau_1 + \tau_2 \pmod{p}$, where τ_1 is the first MAC, τ_2 is the second MAC, and p is a prime number that is large relative to the size of the first and second MACs.

7. In a communication network including a plurality of sending nodes operably linked to a control node, a method, carried out by the control node, comprising:

receiving from a previous node a message string including payload data from a plurality of sending nodes and at least one aggregate MAC computed by aggregating a plurality of constituent MACs generated by the respective sending nodes;

executing a MAC computation function to recompute the constituent MACs from the payload data and one or more shared secret keys, yielding a plurality of recomputed constituent MACs;

executing a MAC aggregation function on the recomputed constituent MACs, yielding a computed aggregate MAC shorter than the concatenation of the recomputed constituent MACs, wherein the MAC aggregation function comprises a large field addition operation performed on the recomputed constituent MACs; and

comparing the computed aggregate MAC to the received aggregate MAC to determine an authentication status of the received aggregate MAC.

8. The method of claim 7, wherein the large field addition operation defines the addition of the recomputed constituent MACs mod *p*, where p is a prime number that is large relative to the size of the recomputed constituent MACs.

9. Apparatus for performing MAC aggregation of constituent MACs, in accordance with a communication network including a plurality of successive sending nodes operably linked to a control node, the apparatus at the control node comprising:

a communication interface;

a memory; and

at least one processor operably coupled to the communication interface and memory and configured to:

(a) receive from a previous node a message string including payload data from a plurality of sending nodes and at least one aggregate MAC computed by aggregating a plurality of constituent MACs generated by the respective sending nodes;

(b) execute a MAC computation function to recompute the constituent MACs from the payload data and one or more shared secret keys, yielding a plurality of recomputed constituent MACs;

(c) execute a MAC aggregation function on the recomputed constituent MACs, yielding a computed aggregate MAC shorter than the concatenation of the recomputed constituent MACs, wherein the MAC aggregation function comprises a large field addition operation performed on the recomputed constituent MACs; and

(d) compare the computed aggregate MAC to the received aggregate MAC to determine an authentication status of the received aggregate MAC.

10. The apparatus of claim 9, wherein the large field addition operation defines the addition of the recomputed constituent MACs mod p, where p is a prime number that is large relative to the size of the recomputed constituent MACs.

FIG. 1

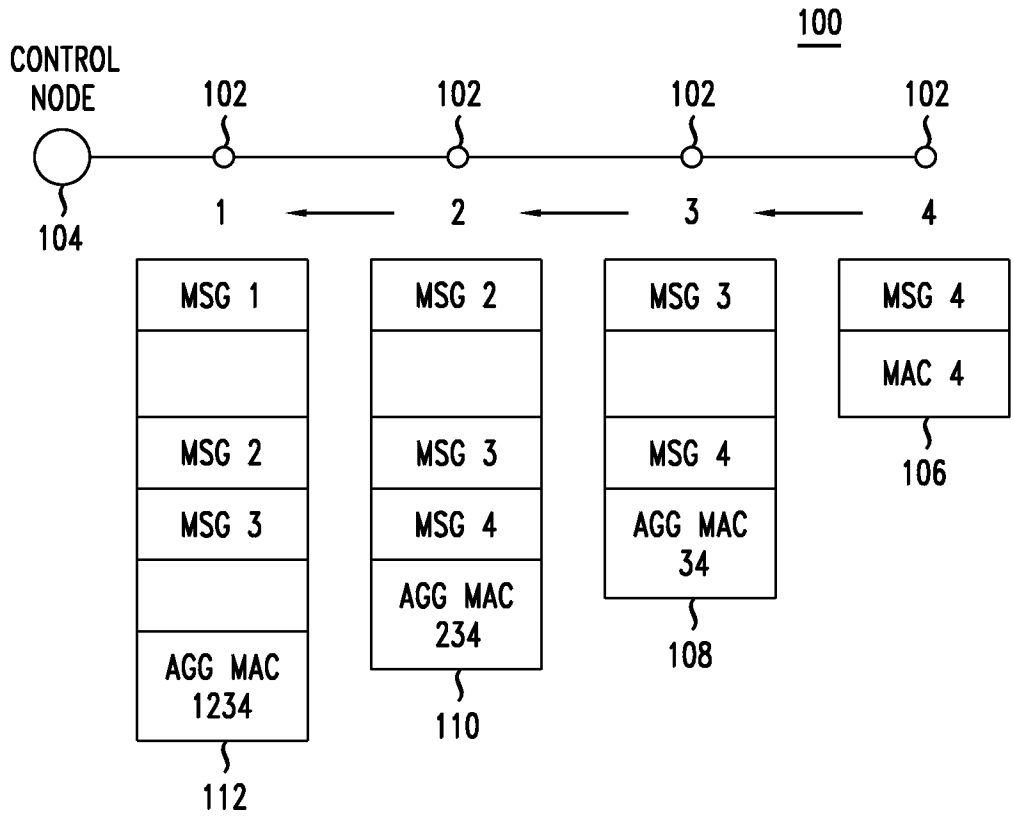


FIG. 2
(PRIOR ART)

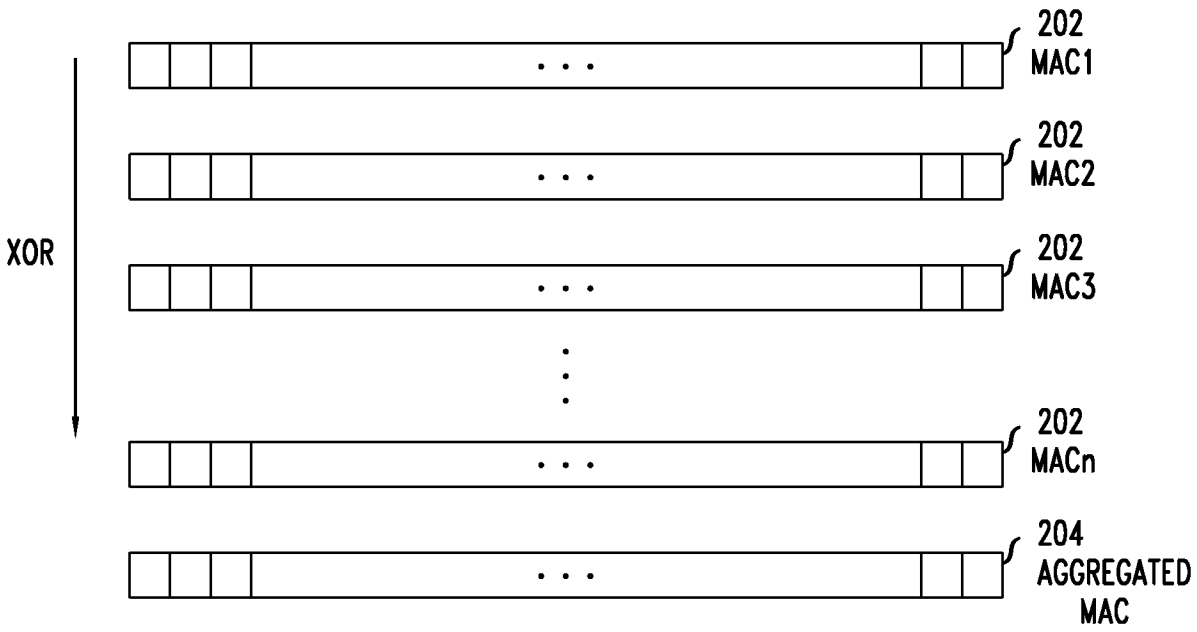


FIG. 3

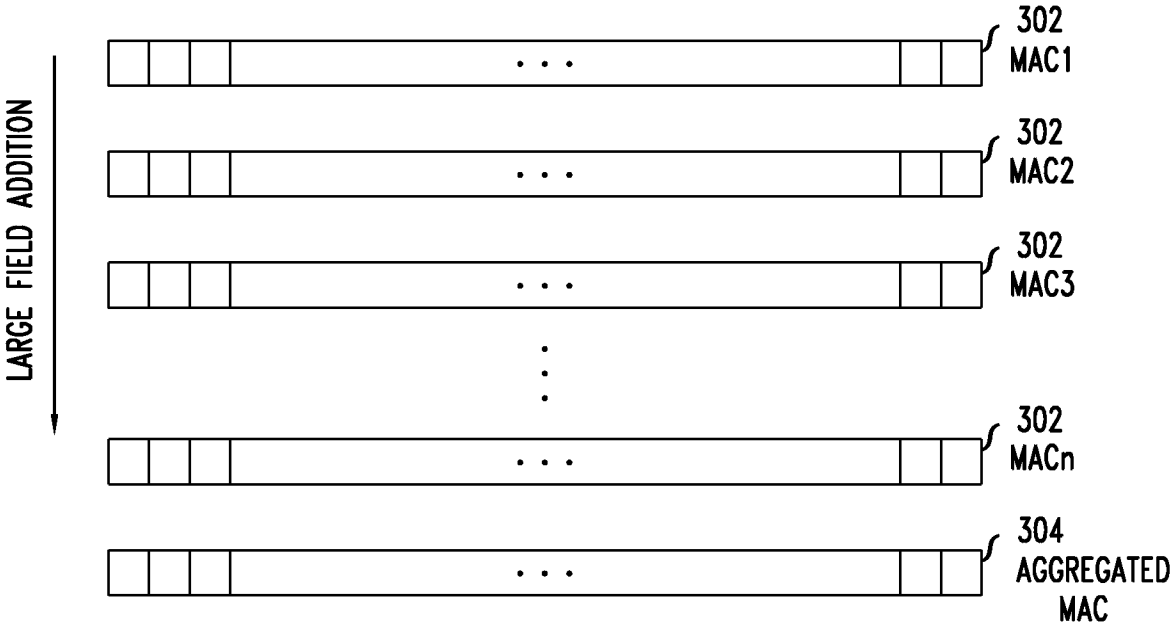

LARGE FIELD ADDITION
302
MAC1
302
MAC2
302
MAC3
302
MACn
304
AGGREGATED
MAC

FIG. 4

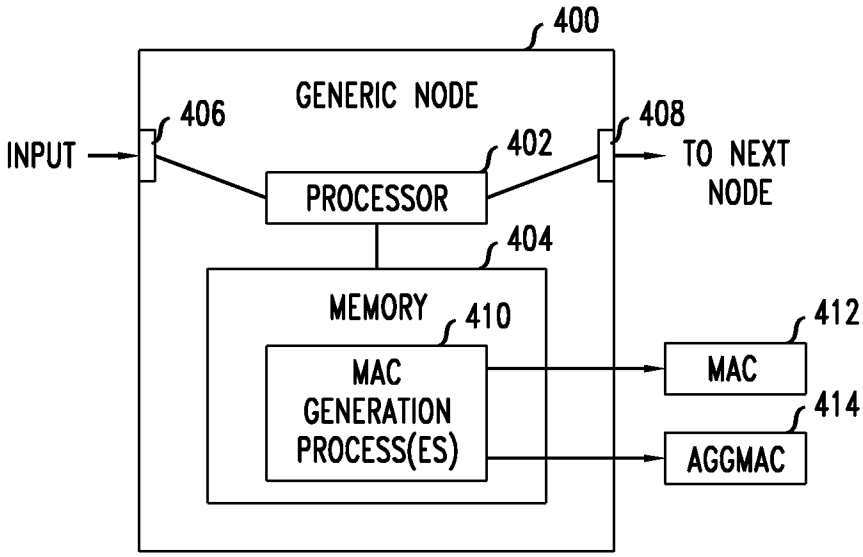

400
GENERIC NODE
406
408
INPUT
402
TO NEXT NODE
PROCESSOR
404
MEMORY
410
412
MAC GENERATION PROCESS(ES)
MAC
414
AGGMAC

FIG. 5

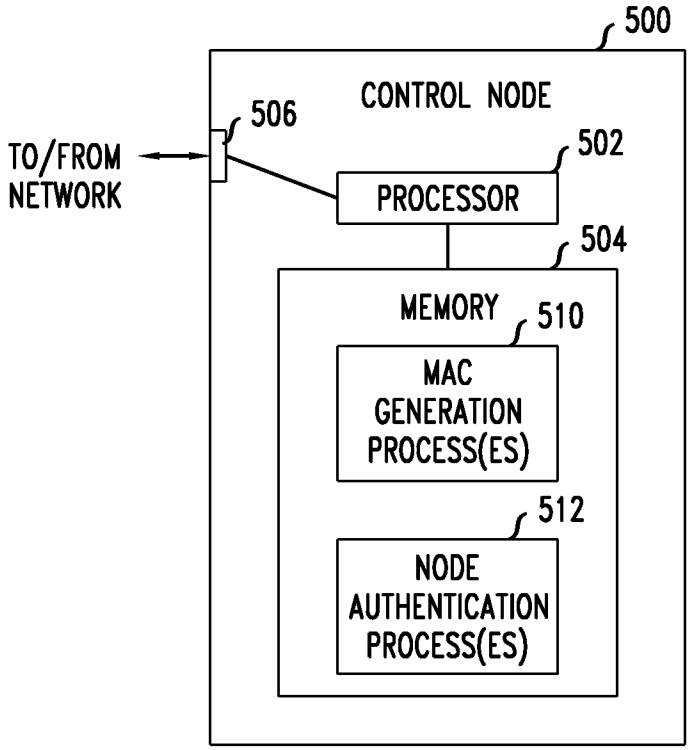

500
CONTROL NODE
506
TO/FROM NETWORK
502
PROCESSOR
504
MEMORY
510
MAC GENERATION PROCESS(ES)
512
NODE AUTHENTICATION PROCESS(ES)

FIG. 6

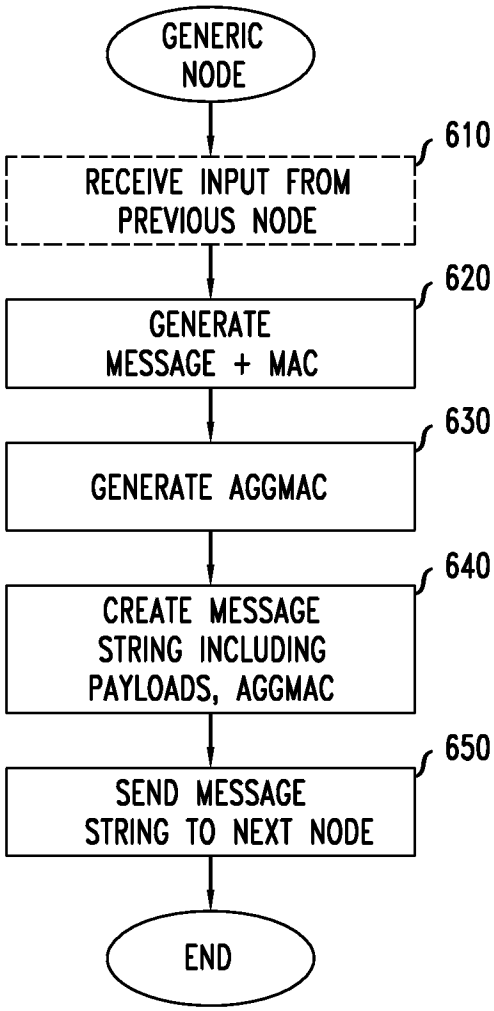
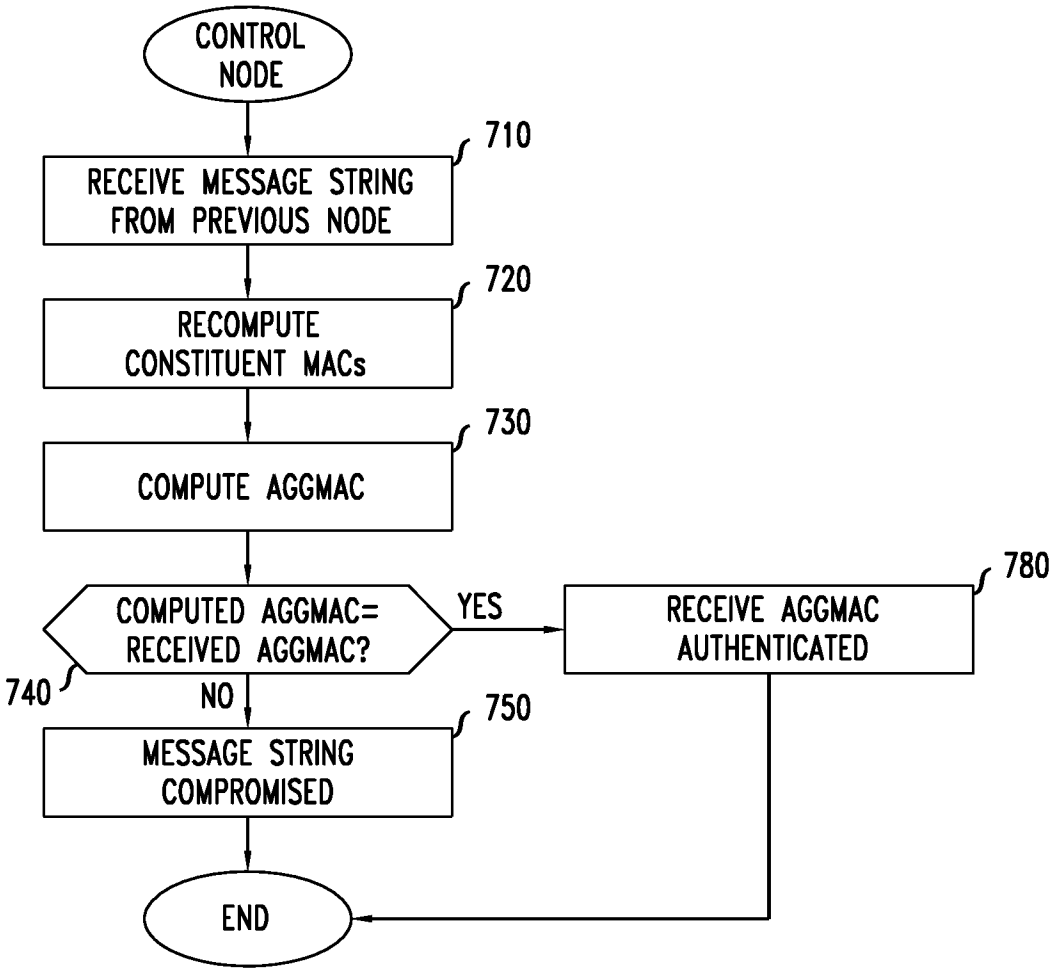

GENERIC NODE
610
RECEIVE INPUT FROM PREVIOUS NODE
620
GENERATE MESSAGE + MAC
630
GENERATE AGGMAC
640
CREATE MESSAGE STRING INCLUDING PAYLOADS, AGGMAC
650
SEND MESSAGE STRING TO NEXT NODE
END

FIG. 7


CONTROL NODE
710
RECEIVE MESSAGE STRING FROM PREVIOUS NODE
720
RECOMPUTE CONSTITUENT MACs
730
COMPUTE AGGMAC
COMPUTED AGGMAC= RECEIVED AGGMAC?
YES
780
RECEIVE AGGMAC AUTHENTICATED
740
NO
750
MESSAGE STRING COMPROMISED
END