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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 April 2015 (09.04.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/049138 A1

(51) International Patent Classification:
H04L 9/32 (2006.01) *H04J 3/06* (2006.01)
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/EP2014/070347

(22) International Filing Date:
24 September 2014 (24.09.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2939/DEL/2013 3 October 2013 (03.10.2013) IN

(71) Applicant: **ALCATEL LUCENT** [FR/FR]; 148/152 route de la Reine, F-92100 Boulogne-Billancourt (FR).

(72) Inventors: **PARIDA, Amaresh**; Alcatel-Lucent India Limited, Nagawara Village, Kasaba Taluk, Outer Ring Road, Manyata Embassy Business PK, 560045 Bangalore (IN). **DEBNATH, Pronoy**; Alcatel-Lucent India Limited, Nagawara Village, Kasaba Taluk, Outer Ring Road, Manyata Embassy Business PK, 560045 Bangalore (IN). **POTE, Parag Narayanrao**; 3c, North Block, Klassic Comforts, HSR Layout, 2nd Sector, Kernataka, 560102 Bangalore (IN).

(74) Agent: **SARUP, David Alexander**; Intellectual Property Business Group, Christchurch Way, Greenwich, London Greater London SE10 0AG (GB).

(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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

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

(54) Title: SECURE TRANSMISSION OF TIME SYNCHRONIZATION PACKETS

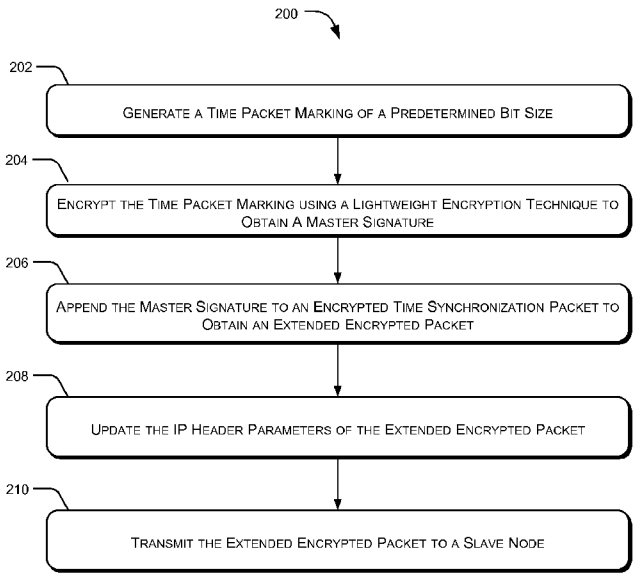

Figure 2

(57) Abstract: Methods and systems for secure transmission of time synchronization packets by a master node (102) in a network environment (100) are described. The method includes generating, by a processor (108), a time packet marking of a predetermined bit size. The method further comprises encrypting, by the processor (108), the time packet marking using a lightweight encryption technique to generate a master signature. Further, the method comprises appending, by the processor (108), the master signature to an encrypted time synchronization packet generated by the master node (102) to obtain an extended encrypted packet, where the encrypted time synchronization packet includes a transmittal time stamp for time synchronization. The method further comprises transmitting, by the master node (102), the extended encrypted packet to a slave node (104) in the network environment (100) for time synchronization.

WO 2015/049138 A1

SECURE TRANSMISSION OF TIME SYNCHRONIZATION PACKETS

FIELD OF INVENTION

[0001] The present subject matter generally relates to time synchronization in a computing environment and, particularly but not exclusively, to secure transmission of time synchronization packets.

BACKGROUND

[0002] Time synchronization protocols are typically implemented to ensure time synchronization between various nodes in a network. Generally, time synchronization protocols involve one system component, say, a time server node or a master node providing timing information to all other components, say, slave nodes in the network so that all the components in the network are synchronized and run in accordance with a common timing information. The master node typically sends the timing information in form of time synchronization packets, which have transmittal timestamps indicating the time at which the time synchronization packets were transmitted by the master node. The slave nodes, on receiving the time synchronization packets, timestamp it to mark the receiving timestamp indicating the time at which the time synchronization packets were received. The slave nodes then decode the time synchronization packets to obtain the transmittal timestamp. The slave nodes may then use the transmittal timestamp and the receiving timestamp to synchronize with the master node and the other components of the network. However, manipulation of the transmittal timestamp or distribution of false timestamp by any intermediate malicious node may affect one or more slave nodes leading to various issues, such as denial of service and accuracy degradation of the affected slave node.

SUMMARY

[0003] This summary is provided to introduce concepts related to systems and methods for secure transmission of time synchronization packets in a computing environment. This summary is neither intended to identify features of the claimed subject matter nor is it intended for use in determining or limiting the scope of the claimed subject matter.

[0004] In one implementation, a method for secure transmission of time synchronization packets by a master node is described. The method includes generating a

time packet marking of a predetermined bit size. The method further comprises encrypting the time packet marking using a lightweight encryption technique to generate a master signature. Further, the method comprises appending the master signature to an encrypted time synchronization packet generated by the master node to obtain an extended encrypted packet, where the encrypted time synchronization packet includes a transmittal timestamp for time synchronization. The method further comprises transmitting the extended encrypted packet to a slave node for time synchronization.

[0005] In another implementation, master node for secure transmission of time synchronization packets is described. The master node comprises a processor and a time packet marking module coupled to the processor. The time packet marking module generates a time packet marking of a predetermined bit size. The time packet marking module further encrypts the time packet marking using a lightweight encryption technique to obtain a master signature. The time packet marking module further appends the master signature to an encrypted time synchronization packet generated by the master node to obtain an extended encrypted packet, where the encrypted time synchronization packet includes a transmittal timestamp for time synchronization. Further, the master node includes a communication module coupled to the processor to transmit the extended encrypted packet to a slave node for time synchronization.

[0006] In another implementation, a method for secure reception of time synchronization packets by a slave node is described. The method includes receiving an encrypted packet from a master node. The method further comprises obtaining a predetermined number of bits from the encrypted packet as a string based on a predetermined bit size shared between the salve node and the master mode. Further, the method comprises determining the encrypted packet to be an extended encrypted packet based on a comparison of the string with a predetermined master signature of the predetermined bit size. The method further comprises timestamping the encrypted packet by marking a receiver timestamp for time synchronization with the master node.

[0007] In yet another implementation, a slave node for secure reception of time synchronization packets is described. The slave node comprises a processor and a communication module coupled to the processor to receive an encrypted packet from a master node. The slave node further comprises a time packet marking module coupled to the

processor to obtain a predetermined number of bits, as a string, from an end of the encrypted packet based on a predetermined bit size shared between the salve node and the master mode. The time packet marking module further determines the encrypted packet to be an extended encrypted packet based on a comparison of the master signature with a predetermined master signature of the predetermined bit size, based on the comparison result. The time packet marking module further timestamps the encrypted packet by marking a receiver timestamp for time synchronization with the master node.

[0008] In yet another implementation, a computer-readable medium having embodied thereon a computer program for executing a method for secure communication of time synchronization packets. The method comprises generating a time packet marking of a predetermined bit size. The method further comprises encrypting the time packet marking using a lightweight encryption technique to generate a master signature. Further, the method comprises appending the master signature to an encrypted time synchronization packet generated by the master node to obtain an extended encrypted packet, where the encrypted time synchronization packet includes a transmittal timestamp for time synchronization. The method further comprises transmitting the extended encrypted packet to a slave node for time synchronization.

BRIEF DESCRIPTION OF THE FIGURES

[0009] The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Some embodiments of system or methods in accordance with embodiments of the present subject matter are now described, by way of example, and with reference to the accompanying figures, in which:

[0010] Figure 1 illustrates a network environment for secure transmission of time synchronization packets, according to an embodiment of the present subject matter.

[0011] Figure 2 illustrates a method of secure transmission of time synchronization packets for time synchronization, according to an embodiment of the present subject matter.

[0012] Figure 3 illustrates a method of secure reception of time synchronization packets for time synchronization, according to an embodiment of the present subject matter.

[0013] It should be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative systems embodying the principles of the present subject matter. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudo code, and the like, represent various processes which may be substantially represented in computer readable medium and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

DESCRIPTION OF EMBODIMENTS

[0014] Systems and methods for secure transmission of time synchronization packets in a network environment are described. Time synchronization between various nodes of a network, such as Ethernet and wireless network is vital for successful implementation and working of the network components, implemented in various systems, for instance process control systems; batch control systems; and heating, ventilation, and air conditioning (HVAC) control systems. The time synchronization may be useful for various reasons, such as for smooth and synchronized functioning of a system implementing the network as an error in time synchronization may affect applications running on the various nodes, thus causing malfunctioning of the system.

[0015] Typically, time synchronization protocols, such as precision time protocol (PTP) and Network Time Protocol (NTP), are deployed in networks to ensure synchronization between the various nodes employed in the network. Usually, a particular type of node which are utilized for time synchronization provide timing information to all other components or nodes by the way of time synchronization packets. Examples of such nodes include, but are not limited to, a time server node, a master node or a grandmaster node slave node in the network. The master node is locally attached to a clock device to ensure accurate timestamping of the time synchronization packets having transmittal timestamps indicating the time at which the time synchronization packets, hereinafter interchangeably referred to as the packets, are transmitted by the master node. On receiving the packets, the slave node may decode the time synchronization packets to obtain the transmittal timestamp. The slave node may then use the transmittal timestamps and a receiving timestamp, generated by the slave node upon receipt of the packet, to synchronize with the master node and other nodes of the network. The time synchronization packets are, however, subject to various security threats, such as malicious nodes that may want to disturb the synchronization

between the various nodes for various reasons. For instance, the packets may be subjected to spoofing, interception and manipulation, replay attack, rogue master attack, interception and removal, packet delay manipulation, cryptographic performance attacks, and denial of service (DoS) attacks.

[0016] The time synchronization packets are thus secured before being transmitted to the slave node using various techniques, such as Internet protocol security (IPSec). IPSec is a protocol typically used for securing internet protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. For instance, IPSec is used for securing time synchronization packets used in PTP. IPSec typically uses heavyweight encryption techniques that, although protect the packets from security attacks, affect the clock accuracy. On receiving the encrypted packets, the slave node may first need to decrypt the encrypted packet to identify whether the packet is a time synchronization packet. On determining that the encrypted packet is a time synchronization packet, the slave node timestamps the encrypted packet by marking a receiving timestamp and processes the transmittal timestamps and the receiving timestamp for synchronization. However, such a delay in timestamping caused due to decryption of the encrypted time packet by the slave node may result in an inaccurate synchronization. Further, implementation of such protocols involves dedicated algorithms and hardware capable of fast processing, thus increasing the implementation and working cost of the time synchronization protocols.

[0017] One of the techniques for secure transmission of time synchronization packets involves IEEE 1588 experimental security extension. The IEEE 1588 experimental security extension is a PTP security extension and protocol that involves adding a security authentication Type-length-value (TLV) message extension to the packet. The security authentication TLV message includes a message authentication code called as integrity check value (ICV) to indicate whether the packet was transmitted by an authenticated source or not. The ICV further indicates whether the packet has been modified by an intermediate node or not. Thus, an inspection of the ICV at the slave node may alert the slave node about whether the time information provided in the packet is reliable or not based on which the slave node may or may not use the packet for time synchronization with the other nodes. The slave node may thus drop the packet and wait for another packet having correct timing information for time synchronization. The above technique may thus save the slave node from incorrect synchronization by indicating malicious timing information. However, such a technique may

still result in malfunctioning of the slave node since repeated dropping of packets due to modification of the timing information may result in the slave node not getting synchronized with the master node. Further, addition or deletion of such security authentication TLV message extension requires complex replay counter handling, thus increasing the load on the master and slave node.

[0018] Another technique for secure transmission of time synchronization packets involves using extended Wrapped Encapsulating Security Payload (WESP) header. Such technique involves adding a time packet identifier in the WESP header of the packet to indicate whether the packet is a time synchronization packet or not. Thus, an inspection of the time packet identifier at the slave node may help the slave node distinguish the time synchronization packets from the other packets, thus saving the time required by the slave node for decrypting the packet before timestamping. However, providing the time packet identifier in an unencrypted form makes the packets vulnerable to security attacks as any malicious node too may identify the time synchronization packet and modify the time synchronization packet. This may thus result in the slave node not getting synchronized with master node. Further, using the above technique requires the nodes to be request for comments (RFC) 5840 and extended WESP compliant, thus making the network operators update the nodes in order to use the above technique. Thus, the techniques typically used for time synchronization are either vulnerable to security attacks or may not provide accurate time synchronization information, which in turn may lead to malfunctioning of the nodes employing such techniques.

[0019] According to an implementation of the present subject matter, systems and methods for secure transmission of time synchronization packets are described. Although the systems and methods are described herein for the IPSec framework using PTP for time synchronization, it will be understood that the systems and method may be implemented for other time synchronization protocols, such as NTP, albeit with few modifications.

[0020] In accordance with an embodiment of the present subject matter, an extended time synchronization packet, hereinafter referred to as extended encrypted packet, is transmitted by a master node to a slave node for time synchronization to ensure secure transmission of time synchronization packet. The extended encrypted packet may include a time packet marking in an encrypted form to indicate that the extended encrypted packet includes a time synchronization packet. The master node, such as a grandmaster linked to a

master clock may send the extended encrypted packet to the slave node, which on receiving the extended encrypted packet may identify the time packet marking and timestamp the extended encrypted packet without any delay, thus improvising the accuracy of time synchronization.

[0021] In one implementation, the master node may initially generate the time packet marking using a dynamic string and a pre-shared key, such that the time packet marking indicates start of flow of time synchronization packets. The pre-shared key may be any alphanumeric key shared between the master node and the slave node. Further, the dynamic string may be a string, such as 'master-slave' and may be different for each time synchronization packet.

[0022] The master node may subsequently encrypt the time packet marking using a lightweight encryption technique to generate an encrypted time packet marking, hereinafter referred to as a master signature of a predetermined bit size, for example, 64 bit. The master node may then generate a time synchronization packet having a transmittal timestamp indicating the time at which the time synchronization packet is transmitted by the master node. Further, the time synchronization packet may be encrypted using an encryption technique, such as ESP to obtain an encrypted time synchronization packet. The master signature is then appended to the encrypted time synchronization packet to obtain the extended encrypted packet. Further IP header parameters may be updated accordingly to ensure successful transmission of the extended encrypted packet. The extended encrypted packet may then be transmitted, as an encrypted packet, to the slave node for time synchronization. As will be understood, since both the time synchronization packet and the time packet marking are encrypted, any intermediate node which may obtain the packet during transmission may not be able to differentiate the extended encrypted from any other encrypted packet.

[0023] Upon receiving the encrypted packet, the slave node obtains a string of predetermined number of bits from the end of the encrypted packet based on the predetermined bit size. Since the slave node knows that the master signature is of the predetermined bit size, it obtains the predetermined number of bits, equal to the predetermined bit size, from the end of the encrypted packet as the string. The string is then compared with a predetermined master signature of the predetermined bit size. In one implementation, the predetermined master signature is same as the master signature and is generated by the slave node using the dynamic string and the pre-shared key. Based on the

comparison, it is determined whether the encrypted packet is an extended encrypted packet or not. In case the string is same as the predetermined master signature, the encrypted packet may be identified as the extended encryption packet. The encrypted packet may then be timestamped by the receiver node by marking a receiver timestamp, thus avoiding the delay typically caused due to heavy decryption procedure used for decryption of the encrypted packet before timestamping.

[0024] The encrypted packet is further processed, by removing the master signature from the extended encrypted packet, updating IP header parameters, and decrypting the encrypted time synchronization packet to obtain the transmittal timestamp. The transmittal timestamp and the received timestamp may then be used for time synchronization between the slave node and the master node. Further, the receiver node and the master node may initiate the time synchronization process and exchange other messages, such as a follow-up message and a delay request message using a similar encryption technique as described above.

[0025] The present subject matter thus facilitates secure transmission of time synchronization packets by the master node to slave node. Appending the encrypted time packet marking to the encrypted time synchronization packet helps the slave node in easy and quick identification of the encrypted time synchronization packet from among the various encrypted packets sent by the master node. Such a fast and easy identification of the encrypted time synchronization packet helps in avoiding a delay in timestamping of the extended encrypted packet at the slave node. Further, encrypting the time packet marking helps in securing a safe transmission of the time synchronization packet as the intermediate nodes may not be able to distinguish the extended encrypted packet from other encrypted packets, as described. Furthermore, using lightweight encryption technique for encrypting the time packet marking facilitates in avoiding any delay in encryption of the time packet marking and in turn the transmission of the extended encryption packet.

[0026] It should be noted that the description and figures merely illustrate the principles of the present subject matter. It will thus be appreciated that those skilled in the art will be able to devise various arrangements that, although not explicitly described or shown herein, embody the principles of the present subject matter and are included within its spirit and scope. Furthermore, all examples recited herein are principally intended expressly to be for pedagogical purposes to aid the reader in understanding the principles of the present subject matter and the concepts contributed by the inventor(s) to furthering the art, and are to

be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the present subject matter, as well as specific examples thereof, are intended to encompass equivalents thereof.

[0027] It will also be appreciated by those skilled in the art that the words during, while, and when as used herein are not exact terms that mean an action takes place instantly upon an initiating action but that there may be some small but reasonable delay, such as a propagation delay, between the initial action and the reaction that is initiated by the initial action. Additionally, the words "connected" and "coupled" are used throughout for clarity of the description and can include either a direct connection or an indirect connection.

[0028] The manner in which the systems and the methods of secure transmission of time synchronization packets may be implemented has been explained in details with respect to the Figures 1 to 3. While aspects of described systems and methods for secure transmission of time synchronization packets can be implemented in any number of different computing systems and transmission environments, the embodiments are described in the context of the following system(s).

[0029] Figure 1 illustrates a network environment 100 for secure transmission of time synchronization packets according to an embodiment of the present subject matter. The network environment 100 includes a master node 102 communicating with one or more slave nodes 104-1, 104-2, …., 104-n, hereinafter collectively referred to as slave nodes 104 and individually referred to as slave node 104, over a network 106. Communication links between the slave nodes 104 and master node 102 are enabled through a desired form of communication, for example, via dial-up modem connections, cable links, digital subscriber lines (DSL), wireless or satellite links, or any other suitable form of communication.

[0030] In one implementation, the master node 102 may be implemented as one or more systems or computing devices, such as a desktop computer, a hand-held device, a cloud server, a mainframe computer, a workstation, a multiprocessor system, a personal digital assistant (PDA), a smart phone, a laptop computer, a network computer, a minicomputer, and a gateway server. Further, the slave nodes 104 may be implemented as one or more computing systems, such as personal computers, multiprocessor systems, laptops, wireless devices, wireless sensors, M2M devices, and cellular communicating devices, such as a personal digital assistant, a smart phone, and a mobile phone, and the like.

[0031] The network 106 may be a wireless network, a wired network, or a combination thereof. The network 106 can also be an individual network or a collection of many such individual networks, interconnected with each other and functioning as a single large network, e.g., the Internet or an intranet. The network 106 can be implemented as one of the different types of networks, such as intranet, local area network (LAN), wide area network (WAN), the internet, and such. Further, the network 106 may include network devices that may interact with the master node 102 and the computing devices 104 through communication links.

[0032] According to an embodiment of the present subject matter, the master node 102 may communicate with the slave node 104 over the network 106 by sending data packets for various purposes, such as time synchronization. For the purpose, the master node 102 transmits time synchronization packets to the slave node 104 to enable the slave node 104 to synchronize their local clocks (not shown in the figure) with a master clock (not shown in the figure) associated with the master node 102. In said embodiment, the master node 102 sends the time synchronization packets in an encrypted form to ensure secure transmission of the time synchronization packets. Further, the master node 102 may append an encrypted time packet marking to the time synchronization packet to generate an extended encrypted packet. The encrypted time packet marking may be understood as a signature added to indicate that the extended encrypted packet includes a time synchronization packet. The master node 102 may then send the extended encrypted packet to the slave node 104, which on receiving the extended encrypted packet may identify the time packet marking and timestamp the extended encrypted packet without any delay, thus improvising the accuracy of time synchronization.

[0033] For the purpose, the master node 102 and the slave node 104 include processors 108-1, 108-2, respectively. The processors 108-1, 108-2, collectively referred to as processor(s) 108 hereinafter, may be implemented as one or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, logic circuitries, and/or any devices that manipulate signals based on operational instructions. Among other capabilities, the processor(s) 108 fetches and executes computer-readable instructions stored in the memory.

[0034] The functions of the various elements shown in the figure, including any functional blocks labeled as "processor(s)", may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate

software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Moreover, explicit use of the term "processor" should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital signal processor (DSP) hardware, network processor, application specific integrated circuit (ASIC), field programmable gate array (FPGA), read only memory (ROM) for storing software, random access memory (RAM), non-volatile storage. Other hardware, conventional and/or custom, may also be included.

[0035] The master node 102 and the slave node 104 include I/O interface(s) 110-1 and 110-2, respectively. The I/O interface(s) 110-1 and 110-2, collectively referred to as I/O interfaces 110, may include a variety of software and hardware interfaces that allow the master node 102 and the slave node 104 to interact with the network 106 and with each other. Further, the I/O interfaces 110 may enable the master node 102 and the slave node 104 to communicate with other communication and computing devices, such as web servers and external repositories.

[0036] The master node 102 and the slave node 104 may include memory 112-1 and 112-2, respectively, collectively referred to as memory 112. The memory 112-1 and 112-2, collectively referred to as memory 112 hereinafter, may be coupled to the processor 108-1, and the processor 108-2, respectively. The memory 112 may include any computer-readable medium known in the art including, for example, volatile memory (e.g., RAM), and/or non-volatile memory (e.g., EPROM, flash memory, etc.).

[0037] The master node 102 and the slave node 104 further include modules 114-1, 114-2, and data 116-1, 116-2, respectively, collectively referred to as modules 114 and data 116, respectively. The modules 114 include routines, programs, objects, components, data structures, and the like, which perform particular tasks or implement particular abstract data types. The modules 114 further include modules that supplement applications on the master node 102 and the slave node 104, for example, modules of an operating system.

[0038] Further, the modules 114 can be implemented in hardware, instructions executed by a processing unit, or by a combination thereof. The processing unit can comprise a computer, a processor, such as the processor 108, a state machine, a logic array or any other suitable devices capable of processing instructions. The processing unit can be a general-

purpose processor which executes instructions to cause the general-purpose processor to perform the tasks or, the processing unit can be dedicated to perform the functions.

[0039] In another aspect of the present subject matter, the modules 114 may be machine-readable instructions (software) which, when executed by a processor/processing unit, perform any of the described functionalities. The machine-readable instructions may be stored on an electronic memory device, hard disk, optical disk or other machine-readable storage medium or non-transitory medium. In one implementation, the machine-readable instructions can be also be downloaded to the storage medium via a network connection. The data 116 serves, amongst other things, as a repository for storing data that may be fetched, processed, received, or generated by one or more of the modules 114.

[0040] In an implementation, the modules 114-1 of the master node 102 include a packet generation module 118, a time synchronization module 120, an encryption-decryption module 122, a time packet marking module 124, a communication module 126, and other module(s) 128. In said implementation, the data 116-1 of the master node 102 includes packet data 130, time marking data 132, time stamping data 134, and other data 136. The other module(s) 128 may include programs or coded instructions that supplement applications and functions, for example, programs in the operating system of the master node 102, and the other data 136 comprise data corresponding to one or more other module(s) 128.

[0041] Similarly, in an implementation, the modules 114-2 of the slave node 104 include a packet generation module 138, a time synchronization module 140, an encryption-decryption module 142, a time packet marking module 144, a communication module 146, and other module(s) 148. In said implementation, the data 116-2 of the slave node 104 includes time marking data 150, time stamping data 152, packet data 154, and other data 156. The other module(s) 148 may include programs or coded instructions that supplement applications and functions, for example, programs in the operating system of the slave node 104, and the other data 156 comprise data corresponding to one or more other module(s) 148.

[0042] In operation, while initiating a time synchronization process, the time packet marking module 124 of the master node 102 may initially generate the time packet marking. The time packet marking may be a dynamically generated signature and may thus be different for each cycle of time synchronization process. For instance, the time packet marking module 124 may use a string “master-slave” indicating the time synchronization packet is being

transmitted by the master node to the slave node, thus indicating the start of flow of time synchronization packets. In one implementation, the time packet marking may be a combination of a dynamic string and a pre-shared key indicating that the encrypted packet to which the time packet marking is appended is sent by the master node 102 and is a time synchronization packet. The pre-shared key may be an alphanumeric key of a predetermined length, shared between the master node and the slave node for generating time packet markings. The dynamic string may be a string, such as 'master-slave' and may be different for each time synchronization packet. The time packet marking thus generated may be further saved in the time marking data 132 by the time packet marking module 124.

[0043] The time packet marking module 124 may subsequently encrypt the time packet marking to obtain an encrypted time packet marking, hereinafter referred to as a master signature. The time packet marking module 124 may encrypt the time packet marking using a lightweight encryption technique. The lightweight encryption technique may be any conventionally known encryption technique, such as hash technique, that may be easy to compute, fast to process, and less resource intensive. Further, the master signature may be of a predetermined bit size, say, 64 bits or 128 bits shared between the slave node 104 and the master node 102. In one implementation, the bit size of the master signature may be determined based on various parameters, such as encryption time, encryption complexity, channel bandwidth, and size of the time synchronization packet. Further, sharing the bit size with the slave node 104 facilitates easy identification of the master signature by the slave node 104.

[0044] In one implementation, the time packet marking module 144 of the slave node 104 may use the same dynamic string, as used by the master node 102, to generate a predetermined master signature, such that the predetermined master signature is same as the master signature and can thus be used to identify the master signature in the encrypted packets received by the slave node 104. The time packet marking module 144 may use the same dynamic string and the pre-shared key used by the time packet marking module 124 for generating the master signature. The time packet marking module 124 and the time packet marking module 144 may thus generate the same signatures using the above described process of encrypting the time packet markings and save them as the master signature and the predetermined master signature, respectively.

[0045] Further, the time packet marking module 124 may select a signature from a set of signatures shared between the slave node 104 and the master node 102. Selecting the

master signature from among the shared set of signatures facilitates in easy and faster identification of the master signature by the slave node 104. Further, the time packet marking module 124 may select the master signature based on a predetermined sequence such that the slave node 104 would know the master signature being generated and appended to the time synchronization packet by the master node 102.

[0046] Further, the master signature generated for a time synchronization packet may be used as the dynamic string for generating a master signature and a predetermined master signature for a subsequent time synchronization packet. Using a previous master signature as the dynamic string for generating a subsequent master signature facilitates the slave node 104 to easily generate the predetermined master signature without requiring any synchronization with the master node 102.

[0047] The time packet marking module 144 and the time packet marking module 124 may further generate a slave signature and a predetermined slave signature, respectively, using the above described process of encrypting the time packet marking. In one implementation, the slave signature may be appended by the slave node 104 to data packets, such as a delay request message sent by the slave node 104 to the master node 102. The predetermined slave signature may be used by the master node 102 to identify such data packets sent by the slave node 104 and thus is similar to the slave signature. The time packet marking module 124 may subsequently save the master signature and the predetermined slave signature in the time marking data 132. The time packet marking module 144 may save the slave signature and the predetermined master signature in the time marking data 150

[0048] Further, the packet generation module 118 may generate the time synchronization packet for being transmitted to the slave node 104 for the time synchronization. As will be understood, the time synchronization packet may be generated and transmitted using any of the known time synchronization protocols, such as PTP and NTP. The time synchronization packet may thus include various fields, such as media access control (MAC) address, Internet protocol (IP) address, header, and data depending on the time synchronization protocol used by the master node 102. Further, the time synchronization module 120 may timestamp the time synchronization packet by marking a transmittal timestamp to the time synchronization packet. The transmittal timestamp may indicate the time at which the time synchronization packet is transmitted to the slave node 104 by the master node 102. In one implementation, the time synchronization module 120 may obtain the time of transmittal from the master clock associated with the master node 102.

[0049] The encryption-decryption module 122 may then encrypt the time synchronization packet to obtain an encrypted time synchronization packet using a conventionally known encryption technique. In one implementation, the encryption-decryption module 122 may use an encapsulating security payload (ESP) technique to encrypt the time synchronization packet. The encrypted encapsulating security payload may then be saved by the encryption-decryption module 122 in the packet data 130. Further, the time packet marking module 124 may append the master signature to the encrypted time synchronization packet to obtain an extended encrypted time synchronization packet, hereinafter referred to as extended encrypted packet. As previously described, appending the master signature to the encrypted time synchronization packet to obtain the extended encrypted packet facilitates in securing a safe transmission of the transmittal timestamp since a malicious intermediate node may never be able to differentiate between a conventional encrypted packet and the extended encrypted packet as both the packets would appear to be certain bits in an encrypted form.

[0050] The time packet marking module 124 may further update IP header parameters of the extended encrypted packet and in turn the time synchronization packet to ensure safe transmission of the extended encrypted packet to the slave node 104. The communication module 126 may then transmit the extended encrypted packet to the slave node 104 for time synchronization. In one implementation, the communication module 126 may transmit the extended encrypted packet as any other encrypted packet using a conventional technique.

[0051] The extended encrypted packet may then be received by the communication module 146 of the slave node 104 as a conventional encrypted packet and processed. Initially the time packet marking module 144 of the slave node 104 may obtain a string of a predetermined number of bits from the end of the encrypted packet in order to determine whether the encrypted packet is a time synchronization packet or a general data packet. In one implementation, the time packet marking module 144 may obtain the string such that the string is of the predetermined bit size, i.e., the string is of the same size as the master signature. For instance, in case the master signature is of 64 bits, then the time packet marking module 144 may obtain the last 64 bits of the encrypted packet as the string.

[0052] The time packet marking module 144 may subsequently compare the string with the predetermined master signature. As described previously, the predetermined master signature is same as the master signature and is generated by encrypting a time packet marking based on the pre-shared key and the dynamic string corresponding to the master

node 102. Based on the comparison, the time packet marking module 144 may determine whether the encrypted packet is a time synchronization packet or not. For instance, in case, the string is different from the predetermined master signature, the time packet marking module 144 may determine the encrypted packet to be a general data packet, i.e., a non-time synchronization packet. The time packet marking module 144 may save the encrypted packet in the other data 156 for being processed by the slave node 104.

[0053] In case the string is same as the predetermined master signature, the time packet marking module 144 may determine the encrypted packet to be the extended encrypted packet having the master signature appended to the time synchronization packet. On determining the encrypted packet to be the extended encrypted packet, the time packet marking module 144 timestamps the encrypted packet. The time packet marking module 144 may mark a receiving timestamp indicating the time at which the encrypted packet is received by the slave node 104. The above described time stamping takes place within a very short time, ensuring there is no delay in the time stamping, thus maintaining high accuracy in the time synchronization process. The time packet marking module 144 may further remove the master signature from the extended encrypted packet to obtain the encrypted time synchronization packet. The time packet marking module 144 may further update the IP header parameters of the encrypted time synchronization packet and save the receiver timestamp in the time stamping data 152. The encryption-decryption module 142 may then decrypt the encrypted time synchronization packet to obtain the time synchronization packet for being processed by the time synchronization module 140. The time synchronization module 140 may process the time synchronization packet to obtain the transmittal timestamp and the receiver timestamp for time synchronization. The transmittal timestamp and the receiver timestamp may be subsequently saved in the time stamping data 152 and used by the time synchronization module 140 for time synchronization using the conventional time synchronization process. For instance, the master node 102 and the slave node 104 may exchange few other messages, such as a follow-up message and a delay request message for obtaining few additional timestamps for time synchronization.

[0054] Further, in one implementation, the communication module 126 of the master node 102 may transmit a follow up packet having a corrected transmittal timestamp using the above described process. In one implementation, the time packet marking module 124 may initially generate a new time packet marking using a new dynamic string and the pre-shared key. As previously described, the master signature may be used as the new dynamic string for

generating the new time packet marking. The time packet marking module 124 may encrypt the new time packet marking to generate a new master signature and append the new master signature to an encrypted follow-up packet to obtain a new extended encrypted packet, interchangeably referred to as extended follow-up message. The extended follow-up message may then be transmitted by the communication module 126 to the slave node 104.

[0055] On receiving the new extended encrypted packet, the slave node 104 may process the new extended encrypted packet using the above described process to obtain the transmittal timestamp. For instance, the time packet marking module 144 may obtain a new string and compare it with a new predetermined master signature to determine if the encrypted packet is the follow-up message. The time packet marking module 144 may then timestamp the encrypted packet and the encryption-decryption module 142 may decrypt the encrypted packet to obtain the corrected transmittal timestamp. Further, in a way similar to the new master signature, the new predetermined master signature may be generated using the previous predetermined master signature. Using the previous predetermined master signature facilitates in fast and easy generation of the subsequent predetermined master signature without utilizing any synchronization between the slave node 104 and the master node 102.

[0056] Furthermore, the slave node 104 may generate a time synchronization packet, such as a delay request message for being sent to the master node 102 to determine any possible delay in message exchange between the master node 102 and the slave node 104. In one implementation, the time packet marking module 144 may generate the delay request message in a way similar to the generation and transmission of the time synchronization packet by the master node 102. For instance, the packet generation module 138 may generate the delay request message and save it in the packet data 154. The time synchronization module 140 may then mark a transmittal timestamp on the delay request message. The encryption-decryption module 142 of the slave node 104 may then encrypt the delay request message. The time packet marking module 144 may subsequently append the slave signature to the encrypted delay request message to obtain extended delay request message, interchangeably referred to as another extended encrypted packet. The communication module 146 of the slave node 104 may then transmit the other extended encrypted packet to the master node 102 as a conventional encrypted packet. Although, the above description is described in context of a delay request message, however, the same process may be followed for generating any time synchronization packet by the slave node 104.

[0057] On receiving the encrypted packet, the master node 102 may process the encrypted packet using the above described process to obtain the delay request timestamp. For instance, the time packet marking module 124 may obtain a string of the predetermined bit size, equal to the bit size of the slave signature, from the end of the encrypted packet. The time packet marking module 124 may then compare the string with the predetermined slave signature to determine if the encrypted packet is the delay request message. The time packet marking module 124 and the encryption-decryption module 122 of the master node 102 may then timestamp and decrypt the extended delay request message, respectively, to obtain the delay request message. The delay request message may then be processed by the time synchronization module 120 to obtain the delay request timestamp.

[0058] Further, in response to the delay request message, the master node 102 may generate and encrypt a delay response message, append a new master signature and transmit an extended delay response message to the slave node 104 using the above described method. On receiving the extended delay response message, the slave node 104 may obtain a new string, compare the string with a new predetermined master signature, timestamp the extended delay response message based on the comparison result, and obtain the timestamps shared in the delay response message for time synchronization.

[0059] The present subject matter thus facilitates secure transmission of time synchronization packets without affecting time stamping accuracy of the slave nodes 104 and the master node 102.

[0060] Figure 2 and 3 illustrate a method 200 and a method 300, respectively, for secure transmission and reception of time synchronization packets, according to an embodiment of the present subject matter. The order in which the method is described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the methods 200 and 300 or any alternative methods. Additionally, individual blocks may be deleted from the methods without departing from the spirit and scope of the subject matter described herein. Furthermore, the method(s) can be implemented in any suitable hardware, software, firmware, or combination thereof.

[0061] The method(s) may be described in the general context of computer executable instructions. Generally, computer executable instructions can include routines, programs, objects, components, data structures, procedures, modules, functions, etc., that perform particular functions or implement particular abstract data types. The methods may

also be practiced in a distributed computing environment where functions are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, computer executable instructions may be located in both local and remote computer storage media, including memory storage devices.

[0062] A person skilled in the art will readily recognize that steps of the method(s) 200 and 300 can be performed by programmed computers. Herein, some embodiments are also intended to cover program storage devices or computer readable medium, for example, digital data storage media, which are machine or computer readable and encode machine-executable or computer-executable programs of instructions, where said instructions perform some or all of the steps of the described method. The program storage devices may be, for example, digital memories, magnetic storage media, such as a magnetic disks and magnetic tapes, hard drives, or optically readable digital data storage media. The embodiments are also intended to cover both communication network and communication devices to perform said steps of the method(s).

[0063] Figure 2 illustrates the method 200 of secure transmission of time synchronization packets for time synchronization, according to an embodiment of the present subject matter.

[0064] At block 202, a time packet marking of a predetermined bit size is generated. In one implementation, a master node, such as the master node 102 may generate the time packet marking for identification of time synchronization packets by a slave node, such as the slave node 104 thus ensuring a secure transmission of the time synchronization packets. In one implementation, the time packet marking may be generated using a dynamic string, such as 'master-slave' and a pre-shared key shared between the master node and the slave node.

[0065] At block 204, the time packet marking is encrypted to obtain a master signature. In one implementation, the time packet marking may be encrypted using a lightweight encryption technique to ensure that encryption and decryption of the time packet marking doesn't consume heavy resources and time. Further, the master signature may be saved in time packet marking data of the master node 102.

[0066] At block 206, the master signature is appended to an encrypted time synchronization packet to obtain an extended time packet. In one implementation, the master signature may be appended to the encrypted time synchronization packet to ensure safe

transmission of the extended encrypted packet, as an intermediate node not having the knowledge of the master signature may assume it to be a part of a general encryption packet and thus may not be able to distinguish an extended encrypted packet from other encrypted packets.

[0067] At block 208, IP header parameters of the extended encrypted packet are updated. In one implementation, upon generating the extended encrypted packet, the IP header parameters may be updated to ensure successful transmission of the extended encrypted packet.

[0068] At block 210, the extended encrypted packet may be transmitted by the master node to a slave node. In one implementation, the extended encrypted packet may be transmitted as a conventional encrypted packet using conventional techniques of transmitting data packets. Transmitting the extended encrypted packet as the encrypted packet facilitates in ensuring that an intermediate node is not able to identify the extended encrypted packet.

[0069] Figure 3 illustrates a method of secure reception of time synchronization packets for time synchronization, according to an embodiment of the present subject matter.

[0070] At block 302, an encrypted packet is received from a master node. In one implementation a slave node may receive the encrypted packet from the master node over a transmission channel.

[0071] At block 304, a string of a predetermined bit size is obtained from the end of the encrypted packet. In one implementation, the string may be of a predetermined bit size shared between the slave node and the master node. For instance, in case the predetermined bit size is 128 bits, the last 128 bits of the encrypted packet may be obtained as the string.

[0072] At block 306, a determination is made to ascertain whether the encrypted packet is an extended encrypted packet and in turn a time synchronization packet or not. In one implementation, the string is compared with a predetermined master signature. In one implementation, the predetermined master signature may be of the predetermined bit size, i.e., same as the size of the master signature. Further, the predetermined master signature may be similar to a master signature appended by the master node to a time synchronization packet for secure transmission. If the string is different from the predetermined master signature, the slave node may determine the encrypted packet to be a normal data packet,

which is the 'No' path from the block 306, the method moves to the block 308 where the encrypted packet is processed.

[0073] In case at block 306 it is determined that the string is same as the predetermined master signature, i.e., the encrypted packet is an extended encrypted packet, which is the 'Yes' path from the block 306, the extended encrypted packet is timestamped at block 310, without any delay, thus ensuring high accuracy in time synchronization.

[0074] At block 312 the encrypted packet is decrypted to obtain a transmittal timestamp for time synchronization. In one implementation, the encrypted packet may be initially processed to remove the master signature to obtain the encrypted time synchronization packet. The encrypted time synchronization packet is then decrypted to obtain the time synchronization packet having the transmittal timestamp. The transmittal timestamp and the receiver timestamp are then used for time synchronization with the master node.

[0075] Although embodiments for secure transmission of time synchronization packets have been described in a language specific to structural features or method(s), it is to be understood that the invention is not necessarily limited to the specific features or method(s) described. Rather, the specific features and methods are disclosed as embodiments for secure transmission of time synchronization packets.

I/We claim:

1. A method for secure transmission of time synchronization packets by a master node (102) in a network environment (100), the method comprising:

generating, by a processor (108), a time packet marking of a predetermined bit size;

encrypting, by the processor (108), the time packet marking using a lightweight encryption technique to generate a master signature;

appending, by the processor (108), the master signature to an encrypted time synchronization packet, generated by the master node (102), to obtain an extended encrypted packet, wherein the encrypted time synchronization packet includes a transmittal timestamp for time synchronization; and

transmitting, by the master node (102), the extended encrypted packet to a slave node (104) in the network environment (100) for time synchronization.

2. The method as claimed in claim 1, wherein the method further comprises:

generating, by the processor (108), a new time packet marking based on the master signature, used for a previous extended encrypted packet, and a pre-shared key, wherein the pre-shared key is shared between the master node (102) and the slave node (104);

generating a new extended encrypted packet subsequent to the extended encrypted packet using at least the new time packet marking.

3. The method as claimed in claim 1, wherein the method further comprises generating, by the processor (108), a predetermined slave signature for identifying time synchronization packets received by the master node (102) from the slave node (104), wherein the time synchronization packets are transmitted by the slave node (104) upon receiving the extended encrypted packet from the master node (102).

4. The method as claimed in claim 1, wherein the method further comprises updating, by the processor (108), internet protocol (IP) header parameters of the extended encrypted packet.

5. A master node (102) for secure transmission of time synchronization packets, the master node (102) comprising:

a processor (108);

a time packet marking module (124) coupled to the processor (108) to:

generate a time packet marking of a predetermined bit size;

encrypt the time packet marking using a lightweight encryption technique to generate a master signature; and

append the master signature to an encrypted time synchronization packet, generated by the master node (102), to obtain an extended encrypted packet, wherein the encrypted time synchronization packet includes a transmittal timestamp for time synchronization; and

a communication module (126) coupled to the processor (108) to transmit the extended encrypted packet to a slave node (104) for the time synchronization.

6. The master node (102) as claimed in claim 5, wherein the time packet marking module (124) further generates a predetermined slave signature for identifying time synchronization packets received by the master node (102) from the slave node (104).

7. The master node (102) as claimed in claim 5, wherein the time packet marking module (124) further,

obtains a string of predetermined bit size from an end of an encrypted packet received from the slave node (104);

compares the string with a predetermined slave signature; and

determines the encrypted packet to be a time synchronization packet based on the comparison.

8. The master node (102) as claimed in claim 5, wherein the time packet marking module (124) further generates a new time packet marking based on the time packet marking and a pre-shared key, wherein the new time packet marking is used for generating a new extended encrypted packet subsequent to the extended encrypted packet, and wherein the pre-shared key is shared between the master node (102) and the slave node (104).

9. A method for secure reception of time synchronization packets by a slave node (104) in a network environment (100), the method comprising:

receiving, by the slave node (104), an encrypted packet from a master node (102) in the network environment (100);

obtaining, by the slave node (104), a predetermined number of bits from the encrypted packet as a string based on a predetermined bit size shared between the salve node (104) and the master mode (102);

determining, by a processor (108), the encrypted packet to be an extended encrypted packet based on a comparison of the string with a predetermined master signature of the predetermined bit size; and

timestamping, by the processor (108), the encrypted packet by marking a receiver timestamp for time synchronization with the master node (102).

10. The method as claimed in claim 9, wherein the method further comprises:

removing, by the processor (108), the master signature from the encrypted packet to obtain an encrypted time synchronization packet;

updating, by the processor (108), internet protocol (IP) header parameters of encrypted time synchronization packet;

decrypting, by the processor (108), the encrypted time synchronization packet; and

obtaining, by the processor (108), a transmittal timestamp from the encrypted time synchronization packet, for time synchronization.

11. The method as claimed in claim 9, wherein the method further comprises generating the predetermined master signature and a slave signature based on a set of IDs shared between the master node (102) and the slave node (104).

12. A slave node (104) for secure reception of time synchronization packets, the slave node (104) comprising:

a processor (108);

a communication module (146) coupled to the processor (108) to receive an encrypted packet from a master node (102); and

a time packet marking module (144) coupled to the processor (108) to:

obtain a predetermined number of bits from an end of the encrypted packet as a string based on a predetermined bit size shared between the salve node (104) and the master mode (102);

determine the encrypted packet to be an extended encrypted packet based on a comparison of the string with a predetermined master signature of the predetermined bit size; and

timestamp the encrypted packet by marking a receiver timestamp for time synchronization with the master node (102).

13. The slave node (104) as claimed in claim 12 further comprising:

an encryption-decryption module (142) coupled to the processor (108) to decrypt the encrypted time synchronization packet; and

a time synchronization module (140) coupled to the processor (108) to obtain a transmittal timestamp from the encrypted time synchronization packet, for time synchronization.

14. The slave node (104) as claimed in claim 12, wherein the time packet marking module (144) further generates at least one of a slave signature and the predetermined master signature based on a set of IDs shared between the master node (102) and the slave node (104).

15. A non-transitory computer-readable medium having embodied thereon a computer program for executing a method for secure communication of time synchronization packets, the method comprising:

generating a time packet marking of a predetermined bit size;

encrypting the time packet marking using a lightweight encryption technique to generate a master signature;

appending the master signature to an encrypted time synchronization packet generated by the master node to obtain an extended encrypted packet, wherein the encrypted time synchronization packet includes a transmittal timestamp for time synchronization; and

transmitting the extended encrypted packet to a slave node for time synchronization.

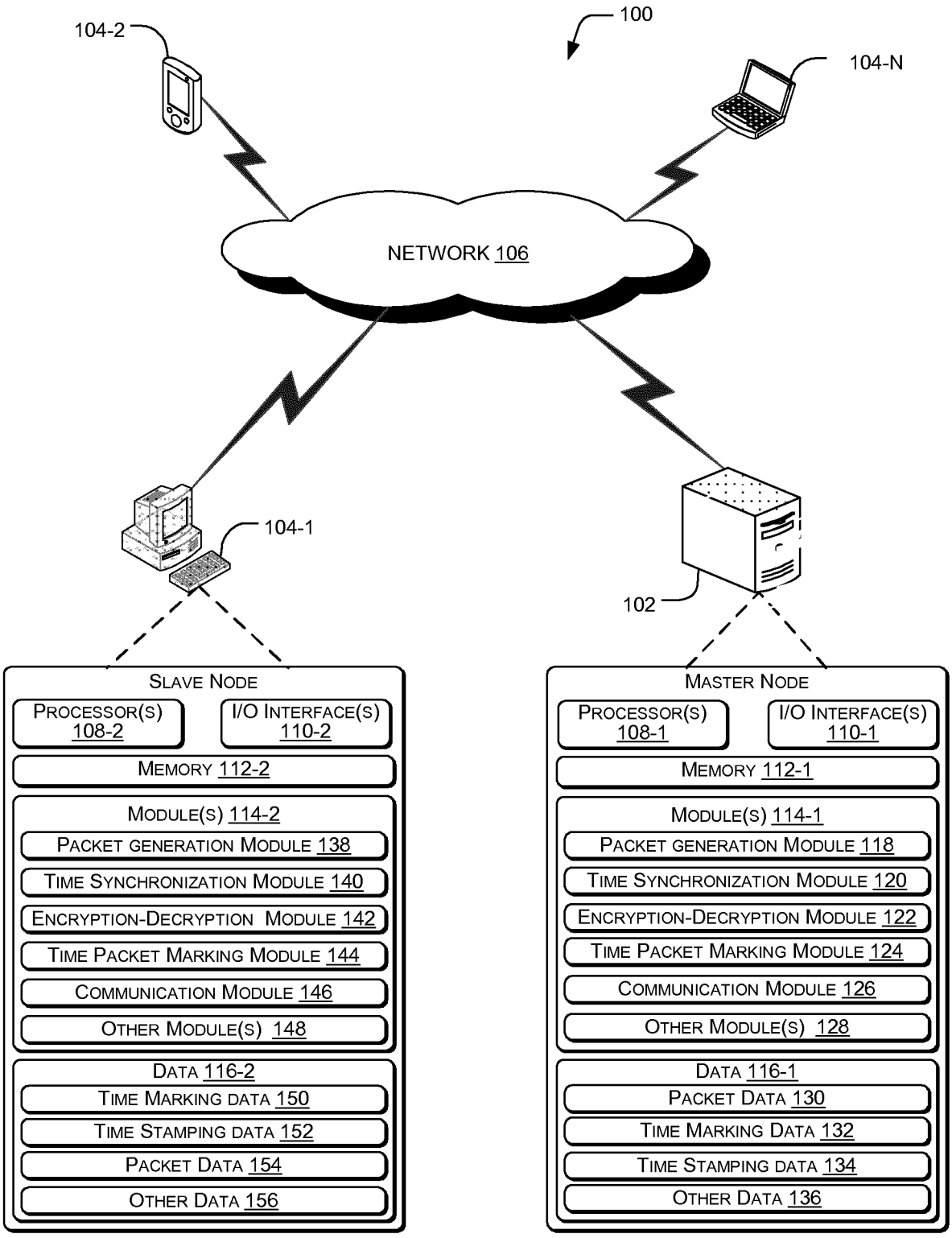
100
104-2
104-N
NETWORK 106
104-1
102
SLAVE NODE
PROCESSOR(S) 108-2
I/O INTERFACE(S) 110-2
MEMORY 112-2
MODULE(S) 114-2
PACKET GENERATION MODULE 138
TIME SYNCHRONIZATION MODULE 140
ENCRYPTION-DECRYPTION MODULE 142
TIME PACKET MARKING MODULE 144
COMMUNICATION MODULE 146
OTHER MODULE(S) 148
DATA 116-2
TIME MARKING DATA 150
TIME STAMPING DATA 152
PACKET DATA 154
OTHER DATA 156
MASTER NODE
PROCESSOR(S) 108-1
I/O INTERFACE(S) 110-1
MEMORY 112-1
MODULE(S) 114-1
PACKET GENERATION MODULE 118
TIME SYNCHRONIZATION MODULE 120
ENCRYPTION-DECRYPTION MODULE 122
TIME PACKET MARKING MODULE 124
COMMUNICATION MODULE 126
OTHER MODULE(S) 128
DATA 116-1
PACKET DATA 130
TIME MARKING DATA 132
TIME STAMPING DATA 134
OTHER DATA 136

Figure 1

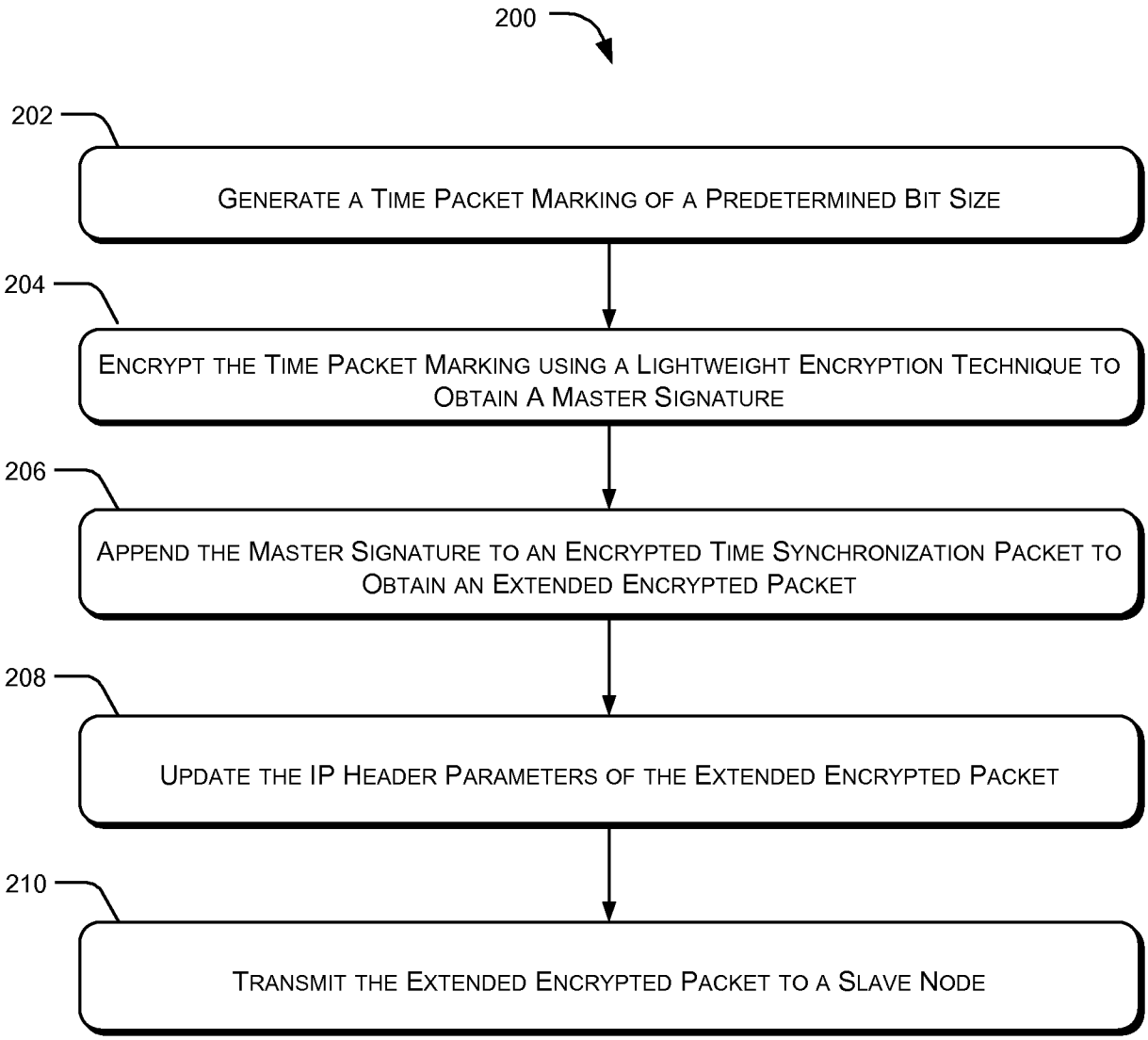
200
202
GENERATE A TIME PACKET MARKING OF A PREDETERMINED BIT SIZE
204
ENCRYPT THE TIME PACKET MARKING USING A LIGHTWEIGHT ENCRYPTION TECHNIQUE TO OBTAIN A MASTER SIGNATURE
206
APPEND THE MASTER SIGNATURE TO AN ENCRYPTED TIME SYNCHRONIZATION PACKET TO OBTAIN AN EXTENDED ENCRYPTED PACKET
208
UPDATE THE IP HEADER PARAMETERS OF THE EXTENDED ENCRYPTED PACKET
210
TRANSMIT THE EXTENDED ENCRYPTED PACKET TO A SLAVE NODE

Figure 2

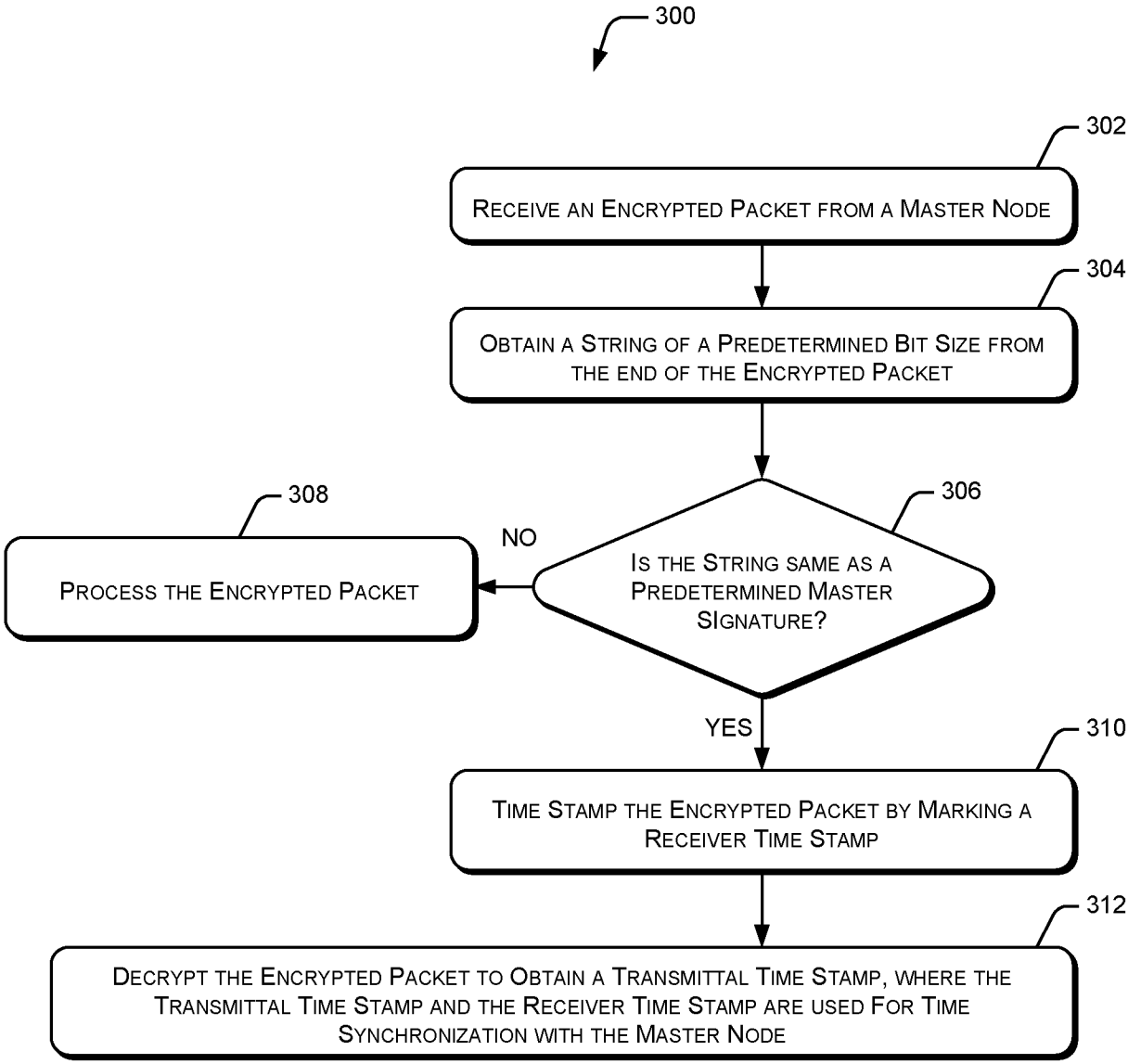
300
302
RECEIVE AN ENCRYPTED PACKET FROM A MASTER NODE
304
OBTAIN A STRING OF A PREDETERMINED BIT SIZE FROM THE END OF THE ENCRYPTED PACKET
306
IS THE STRING SAME AS A PREDETERMINED MASTER SIGNATURE?
NO
308
PROCESS THE ENCRYPTED PACKET
YES
310
TIME STAMP THE ENCRYPTED PACKET BY MARKING A RECEIVER TIME STAMP
312
DECRYPT THE ENCRYPTED PACKET TO OBTAIN A TRANSMITTAL TIME STAMP, WHERE THE TRANSMITTAL TIME STAMP AND THE RECEIVER TIME STAMP ARE USED FOR TIME SYNCHRONIZATION WITH THE MASTER NODE

Figure 3

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2014/070347

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/32 H04L29/06 H04J3/06
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L H04J

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	DE 10 2005 025325 A1 (SIEMENS AG [DE]) 7 December 2006 (2006-12-07) abstract; claims 1,11 paragraphs [0004], [0007], [0008], [0011], [0012], [0014] paragraphs [0020] - [0025], [0027], [0028], [0030]	1-15
X	US 2010/223399 A1 (KIM SEUNG-HWAN [KR] ET AL) 2 September 2010 (2010-09-02) abstract; claims 1,2,14,15; figures 1,2 paragraphs [0008] - [0010], [0024] - [0031], [0036], [0038], [0039] paragraphs [0042], [0043], [0045] - [0051], [0052] - [0057]	1-15

-/--

[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
27 November 2014	04/12/2014

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	Wolters, Robert

1

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2014/070347

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	"IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems;IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002) ED - Anonymous", IEEE STANDARD; [IEEE STANDARD], IEEE, PISCATAWAY, NJ, USA, 24 July 2008 (2008-07-24), pages c1-269, XP017604130, ISBN: 978-0-7381-5400-8 Annex K: mainly sections K.2, K.6, K.14.2	1-15
A	US 2010/153742 A1 (KUO LUNG-CHIH [TW] ET AL) 17 June 2010 (2010-06-17) abstract; claims 1,9 paragraphs [0009], [0013] - [0015], [0021] - [0024]	1-15
A	CAGRI ONAL ET AL: "Security improvements for IEEE 1588 Annex K: Implementation and comparison of authentication codes", PRECISION CLOCK SYNCHRONIZATION FOR MEASUREMENT CONTROL AND COMMUNICATION (ISPCS), 2012 INTERNATIONAL IEEE SYMPOSIUM ON, IEEE, 24 September 2012 (2012-09-24), pages 1-6, XP032257248, DOI: 10.1109/ISPCS.2012.6336632 ISBN: 978-1-4577-1714-7 the whole document	1-15

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2014/070347

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
DE 102005025325 A1	07-12-2006	DE 102005025325 A1	07-12-2006
		WO 2006128747 A1	07-12-2006
US 2010223399 A1	02-09-2010	JP 2010206777 A	16-09-2010
		KR 20100098025 A	06-09-2010
		US 2010223399 A1	02-09-2010
US 2010153742 A1	17-06-2010	TW 201023579 A	16-06-2010
		US 2010153742 A1	17-06-2010