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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 May 2007 (10.05.2007)

PCT

(10) International Publication Number
WO 2007/053708 A2

(51) International Patent Classification:
G06F 21/20 (2006.01)

(21) International Application Number:
PCT/US2006/042713

(22) International Filing Date: 31 October 2006 (31.10.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/732,019 31 October 2005 (31.10.2005) US
60/808,313 24 May 2006 (24.05.2006) US

(71) Applicant *(for all designated States except US)*: **THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK** [US/US]; 412 Low Memorial Library, 535 West 116th Street, New York, NY 10027 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **STOLFO, Salvatore, J.** [US/US]; 80 Kenilworth Road, Ridgewood, NJ 07450 (US). **CRETU, Gabriela** [RO/US]; 414 W. 120th Street, #211, New York, NY 10027 (US). **FRIAS-MARTINEZ, Vanessa** [ES/US]; 434 W. 120th, 4A, New York, NY 10027 (US). **PAREKH, Janak** [US/US]; 110 Bayview Road, Manhasset, NY 11030 (US).

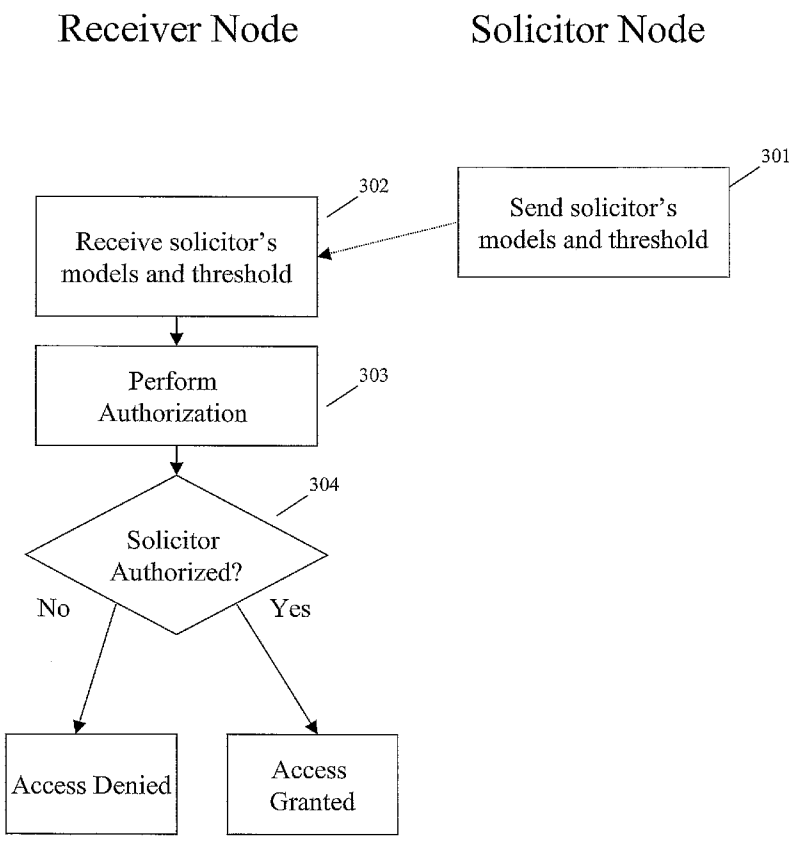
(74) Agents: **BYRNE, Matthew, T.** et al.; Wilmer Cutler Pickering Hale and Dorr LLP, 399 Park Avenue, New York, NY 10022 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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,

[Continued on next page]

(54) Title: METHODS, MEDIA, AND SYSTEMS FOR SECURING COMMUNICATIONS BETWEEN A FIRST NODE AND A SECOND NODE



(57) Abstract: Methods, media, and systems for securing communications between a first node and a second node are provided. In some embodiments, methods for securing communication between a first node and a second node are provided. The methods comprising: receiving at least one model of behavior of the second node at the first node; and authorizing the first node to receive traffic from the second node based on the difference between the at least one model of behavior of the second node and at least one model of behavior of the first node.

WO 2007/053708 A2

GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHODS, MEDIA, AND SYSTEMS FOR SECURING COMMUNICATIONS BETWEEN A FIRST NODE AND A SECOND NODE

Cross-Reference to Related Applications

[0001] This application claims the benefit of United States Provisional Patent Application No. 60/732,019, filed October 31, 2005, and of United States Provisional Patent Application No. 60/808,313, filed May 24, 2006, each of which is hereby incorporated by reference herein in its entirety.

Technical Field

[0002] The disclosed subject matter relates to methods, media, and systems for securing communications between a first node and a second node.

Background

[0003] Digital processing devices can be attacked by other digital processing devices with which they are communicating. Accordingly, some digital processing devices only accept communication from devices that they trust. A device can establish trust of a requesting device by, for example, examining information identifying the requesting device or information describing the type or configuration of the requesting device. However, such information may not always be a useful predictor of whether a device is, for example, already infected by malicious software, likely to launch attacks, and/or likely to become infected by malicious software.

[0004] Furthermore, in some types of networks, for example, in mobile ad-hoc networks (MANETS) which can have, for example, no central control, no wireless switches, no base stations, and can have devices entering and leaving the network dynamically, decisions of whether to trust communications can be especially challenging due to the quickly changing topology and membership of the network. For example, an intrusion detection system can perceive packet dropping or communication from unknown outside digital processing devices as an indication of attack. However, such occurrences can be commonplace and benign in MANETS.

Summary

[0005] Methods, media, and systems for securing communications between a first node and a second node are provided. In some embodiments, methods for securing communications between a first node and a second node are provided. The methods comprising: receiving at least one model of behavior of the second node at the first node; and authorizing the first node to receive traffic from the second node based on the difference between the at least one model of behavior of the second node and at least one model of behavior of the first node.

[0006] In some embodiments, a device that secures communication between a first node and a second node is provided. The device comprising: an interface in communication with a network; a memory; and a processor in communication with the memory and the interface; wherein the processor: receives at least one model of behavior of the second node; and authorizes the first node to receive traffic from the second node based on the difference between the at least one model of behavior of the second node and at least one model of behavior of the first node.

[0007] In some embodiments, a computer-readable medium containing computer-executable instructions, that when executed by a processor, cause the processor to perform methods for securing communications between a first node and a second node are provided. The methods comprising: receiving at least one model of behavior of the second node; and authorizing the first node to receive traffic from the second node based on the difference between the at least one model of behavior of the second node and at least one model of behavior of the first node.

[0008] In some embodiments, methods for securing communications between a first node and a second node are provided. The methods comprising: receiving at least one model of behavior of the second node at the first node; and rejecting traffic sent from the second node to the first node based on the difference between the at least one model of behavior of the second node and at least one model of behavior of the first node.

Brief Description of the Drawings

[0009] FIG. 1 is a schematic diagram of an illustrative system suitable for implementation of an application that secures communication between digital processing devices in accordance with some embodiments of the disclosed subject matter.

[0010] FIG. 2 is an example of the server and one of the clients of FIG. 1 that can be used in accordance with some embodiments of the disclosed subject matter.

[0011] FIG. 3 is a simplified illustration of a method for securing communications between digital processing devices in accordance with some embodiments of the disclosed subject matter.

[0012] FIG. 4 is an illustrative example of an arrangement of a network in accordance with some embodiments of the disclosed subject matter.

[0013] FIG. 5 is another illustrative example of an arrangement of a network in accordance with some embodiments of the disclosed subject matter.

[0014] FIG. 6 is another illustrative example of an arrangement of a network in accordance with some embodiments of the disclosed subject matter.

[0015] FIG. 7 is an illustrative example of a table used to store information in accordance with some embodiments of the disclosed subject matter.

Detailed Description

[0016] Methods, media, and systems for securing communications between a first node and a second node are disclosed. Some embodiments of the disclosed subject matter can authorize and/or authenticate, for example, a node so that it can be granted or denied entry and/or access to, for example, a network, another node, and/or a service. In some embodiments of the disclosed subject matter, a decision of whether to grant or deny access can be based on models of behavior. Some embodiments of the disclosed subject matter can compare, for example, a model to another model or a model to observed behavior and calculate a distance or difference value between the two models or the model and the behavior. Some embodiments can compute a threshold value, based on, for example, models or behaviors of nodes already in a network. Such a threshold can be used, in some embodiments, to grant or deny access to, for example, a node, network, or service based on a comparison of the threshold and a distance between, for example, models and/or observed behavior. In some embodiments, for example, if a model of a node requesting entry to a network is determined to be too different from the behavior of other nodes in the network, the node can be denied entry. Additionally, in some embodiments, for example, after being granted access to a network, a node can be blocked or otherwise removed from a network if,

for example, its behavior model and its actual behavior are not within a certain distance. The disclosed subject matter can be used, in some embodiments, for example, to either augment or replace existing techniques of authenticating and authorizing nodes in a network.

[0017] Some embodiments of the disclosed subject matter can use any appropriate model type. For example, some embodiments can use a content anomaly detector that can calculate a model, for one or more ports, packet length, and direction of traffic flow (i.e., ingress and egress). This model can represent normal traffic for a set of features. The set (e.g., {port, packet length, direction}) can be modeled as one or more centroids that define the payload profile of normal traffic. Accordingly, such an anomaly detector can may be referred to as a payload anomaly detector. Each centroid can represent, for example, a l-gram frequency analysis of the payload content and centroids can be clustered to reduce the size of the model. A model training phase, in such an anomaly detector, can be, for example, either online or offline. During the training phase, l-gram frequency analysis can be computed for received packets. Training can generate normal input and output models and these models can be used during a testing phase to detect whether a payload going through a node follows a provided model or is too different from the normal profile and/or or model. In order to measure that difference, thresholds can be defined. After a training phase and before a testing phase, a calibration phase can be carried out over the same training data in order to obtain thresholds. When the traffic going through the testing phase has a payload whose distance to the normal model is greater than the calculated threshold, an alarm can be generated. More information on some examples of such anomaly detectors can be found in United States Patent Application No. 10/986,447 filed on November 22, 2004, which is hereby incorporated by reference herein in its entirety.

[0018] Some embodiments of the disclosed subject matter can use, for example, content anomaly detectors that can build behavior profiles or models as Bloom filters. For example, a mixture of n-grams from the content of packets can be hashed into a Bloom filter to define the normal model from clean traffic seen by a node. Some embodiments of such detectors can also provide, for example, a second Bloom filter that can contain malicious n-grams, extracted from a set of online Snort rules associated with specific worms. Packets can be tested during detection by, for example, counting the number of malicious n-grams that were not seen during a training phase. N-grams appearing in a packet and in a malicious Bloom filter can be scored higher than other packets. Threshold logic can determine, for

example, which packets to consider anomalous. More information on some examples of such detectors can be found in United States Provisional Patent Application No. 60/778,008, filed on February 28, 2006, and Ke Wang, Janak J. Parekh, Salvatore J. Stolfo, "Anagram: A Content Anomaly Detector Resistant To Mimicry Attack" *In Proceedings of the Nineth International Symposium on Recent Advances in Intrusion Detection(RAID 2006), Hamburg, Germany, September 2006,* each of which is hereby incorporated by reference herein in its entirety.

[0019] FIG. 1 is a schematic diagram of an illustrative system 100 that may be used for model creation, model exchange, intrusion and anomaly detection, securing communications between digital processing devices, and/or for protecting applications from attack in accordance with some embodiments of the disclosed subject matter. As illustrated, system 100 can include one or more clients 102. Clients 102 can be local to each other or remote from each other, and can be connected by one or more communications links 104 to a communications network 106. Communications network 106 can also be linked through a communications link 108 to a server 110. Various embodiments of the disclosed subject matter can be implemented on at least one of the server and the clients. It is also possible that a client and a server can be connected through communication links 108 or 104 directly and not through a communication network 106.

[0020] In system 100, server 110 can be any suitable server or digital processing device for executing an application, such as, for example, a processor, a computer, a data processing device, or a combination of such devices. Communications network 106 can be any suitable computer network including the Internet, an intranet, a wide-area network (WAN), a local-area network (LAN), a wireless network, a digital subscriber line (DSL) network, a frame relay network, an asynchronous transfer mode (ATM) network, a virtual private network (VPN), a mobile ad-hoc network (MANET), or any combination of any of the same. Communications links 104 and 108 can be any communications links suitable for communicating data between clients 102 and server 110, such as network links, dial-up links, wireless links, hard-wired links, etc. Clients 102 can be any suitable digital processing devices, such as, for example, personal computers, laptop computers, mainframe computers, data displays, Internet browsers, personal digital assistants (PDAs), two-way pagers, wireless terminals, portable telephones, etc., or any combination of the same. Clients 102 and server 110 can be located at any suitable location. In one embodiment, clients 102 and server 110

can be located within an organization. Alternatively, clients 102 and server 110 can be distributed between multiple organizations.

[0021] The server and one of the clients, which are depicted in FIG. 1, are illustrated in more detail in FIG. 2. Referring to FIG. 2, client 102 and server 110 can include respectively, among other things, processors 202 and 220, displays 204 and 222, input devices 206 and 224, and memory 208 and 226, which can be interconnected. In one embodiment, memory 208 and 226 contain a storage device for storing a program for controlling processors 202 and 220. Memory 208 and 226 can also contain applications for model creation, model exchange, intrusion and anomaly detection, and for protecting applications from attack. In some embodiments, various applications can be resident in the memory of client 102 or server 110. It should be noted that variations and combinations of system 100 might be suitable for different embodiments of the disclosed subject matter. In addition, although some embodiments are described herein as being implemented on a client and/or a server, this is only illustrative. Various components of some embodiments of the disclosed subject matter can be implemented on any suitable platform.

[0022] It should be noted that system 100 can be, for example, a mobile ad-hoc network (MANET). Such a system can be self-configuring, where nodes, such as, for example, routers, clients, and/or servers can be connected by wireless links and form arbitrary topologies. Various components of the system 100 can be free to move randomly and organize arbitrarily. Accordingly, in some embodiments of the disclosed subject matter, the topology and membership of system 100 can change rapidly and unpredictably. A system 100 that is embodied as, for example, a MANET, can be connected to or in communication with other systems 100 that can be, for example, MANETs or any other type of appropriate network. Additionally, a system 100 can be a combination of various network types, for example, a system 100 can include some mobile nodes that can move in and out of the network and other nodes that can be stationary.

[0023] Some embodiments of the disclosed subject matter can be separated into two phases, such as, for example, an authorization phase and a communication phase. During an authorization phase, a node can request access to communicate with another node and can be either granted or denied that access. Each node can contain, for example, an anomaly detector that can build a model of, for example, its input and output traffic to define its normal behavior. Such models can be exchanged and compared among nodes and it can be

determined how similar the models are to each other, by using, for example, threshold logic. The results of the comparison can be used to grant or deny access to a node.

[0024] During a communication phase, each node can compare, for example, actual behavior to various behavior models to validate that other nodes are acting according to their respective behavior models. This can detect, for example, if a node misrepresented its normal behavior during authorization or if its behavior changes thereafter. It should be noted, however, that some embodiments of the disclosed subject matter can be separated into more than two phases while other embodiments of the disclosed subject matter are not separated into phases at all. For example, in some embodiments, an authorization phase can be integrated with a communication phase.

[0025] When a node approaches the communication range of a network, authorization can begin. Authorization can result in a node being either accepted into, or rejected from, a network. A node already in the network, which is receiving a request to grant entry, can be referred to as a receiver and a new incoming node can be referred to as a solicitor. A node can request access to a network because, for example, it is a node with wireless connectivity that has reached the range of a wireless network such as a MANET. A node can also request access simply because it now needs or wants access to a network with which it was already able to send requests.

[0026] FIG. 3 illustrates authorization being performed between a receiver node and a solicitor node. A solicitor can send, at 301, its models and a traffic threshold to a receiver. The receiver can receive these, at 302, and perform various tests to determine if the solicitor should be granted access to, for example, the network of the receiver, at 303. These tests can include, for example, calculations of the differences between the models of the receiver and the models of the solicitor. A threshold value, for example, can be used to determine if these differences are too large for the solicitor to be admitted to the network. If the receiver node determines, at 304, that the models of the solicitor node are sufficiently similar to, for example, the models of the receiver, the solicitor can be admitted to the network. The roles of the node that was the solicitor and the node that was the receiver can be switched. That is the former solicitor node can also receive the models and a traffic threshold from the former receiver node and attempt to authorize the former receiver node to access the former solicitor node's network. These two authorizations can take place in parallel, in sequence, or partly overlapping in time. In some embodiments, once a node has been granted access to

communicate with another node, it can be authorized to send traffic to that node despite it not yet allowing traffic to be received from that node. In some embodiments, until both a receiver node and a solicitor node have exchanged and authorized each other's models, traffic cannot be sent between them in either direction. In some embodiments, it is possible that only one of the nodes is required to go through an authorization process. In such embodiments, once a receiver authorizes a solicitor, both nodes can accept traffic from, and send traffic to, the other.

[0027] As illustrated in FIG. 4, node 404 is in range of network 410, which contains nodes 401, 402, and 403. In this example, node 402 is in the communication range of node 404 and has received a request for access from node 404. Accordingly, node 402 can be referred to as the receiver, and node 404 can be referred to as the solicitor. An authorization phase can take place between node 402 and node 404 that can enable nodes 402 and 404 to communicate.

[0028] In some embodiments, if a receiver authorizes a solicitor, the solicitor is only authorized to send traffic to that specific receiver. If the solicitor wishes to communicate with additional nodes in the network of the receiver, the solicitor can perform authorization with those nodes as well. Such embodiments can guarantee that each node has considered the models of each node it communicates with. For example, in some embodiments, as illustrated in FIG. 5, if node 404 is authorized by node 402, node 404 is authorized to communicate only with node 402. In such embodiments, if node 404 also needs to communicate, for example, with node 401, then nodes 401 and 404 can go through authorization. The configuration created by node 404 being authorized by both nodes 402 and 401, in such an embodiment, is illustrated in FIG. 6. In some embodiments, however, performing authorization with one node of a network can be sufficient to be granted access to all nodes in that network.

[0029] If a solicitor, such as node 404 is accepted into the network, its input, output, and worm models, as well as its threshold information can be saved in a receiver's model table together with the distances to the corresponding models. For example, FIG. 7 illustrates a model table of node 402. The models and the thresholds can be saved together with the distances between them and the input, output, and model of attack behavior. For example, such a table can contain information for each of the nodes 701 that a node has authorized.

The information can include, for example, the input models 702, output models 703, and anomaly models (e.g., worm models) 704 for each of the nodes.

[0030] When, for example, referring again to FIG. 4, node 404 wants to communicate with network 410 it can provide information to node 402 to begin authorization. This information can include, for example, a certificate identifying node 404, node 404's input and output models, a traffic threshold of node 404's output model, and a model containing known attacks, such as a worm model. Node 404's input and output models can define, for example, the type of input and output traffic that node 404 usually sees and generates. Node 404's traffic threshold can define, for example, the logic and/or numerical threshold that it uses to decide whether traffic is anomalous. Such thresholds can be stored in column 705 of the table of FIG. 7. It should also be noted that node 404's model of known attacks can be, for example, a model of malicious n-grams, can be stored in a signature database, and can specify a set of worms and/or other malicious attackers known by node 404.

[0031] Node 402 can receive the information, sent from node 404, and use it to decide whether to grant access to node 404. In making a decision, node 402 can analyze the information in various ways. For example, node 402 can perform a serious of tests on this information and if each of the tests passes, node 404 can be admitted to network 410. A first test can be, for example, a test of the distance between node 402's model of known attacks and node 404's model of known attacks. This can be accomplished by, for example, calculating the distance between these models and if this distance is less than a certain threshold, granting access to node 404 or continuing on to another test. However, if the distance is, for example, greater than a certain threshold, node 404 can be denied access to the network 410 because, for example, its model of known attacks is not up to date. One reason for rejecting a solicitor on this basis is that, because the solicitor's state of protection can require updating, the solicitor can already be infected and thus could become a source of entry for worms or other attackers.

[0032] A second test, for example, can be a test of the distance between node 402's model of known attacks and the input and output models of node 404. This can, for example, be used to determine whether node 404's models contain malicious n-grams known by node 402. Like the first test, if this test passes, node 404 can be admitted to the network or another test can be performed. If the test fails, node 404 can be denied access to the network. One reason for rejecting a solicitor on this basis is that the presence of malicious traffic in a

solicitor's output model can indicate that the receiver has sent malicious traffic and can, therefore, be likely to continuing sending malicious traffic if admitted to a network.

[0033] A third test, for example, can be a test of the distance between node 404's input model and node 402's input model and a test of the distance between node 402's output model and node 404's output model. These distances can be calculated to determine the similarity between node 402's behavior and node 404's behavior. If these distances are below a certain threshold, the models can be considered similar to normal models accepted by, for example, the network 410, and node 404 can therefore be granted access. If the distances are too great, however, node 404 can be denied access to, for example, the network.

[0034] The type of distance calculations performed, for example, in the above discussed tests, can depend on the type of models being used. If, for example, a payload based model is used, the distance can be, for example, a Manhattan distance to calculate how different the two models are in statistical distribution. If, for example, anomalous n-gram and/or Bloom filter based detectors are used, a count of the number of n-grams that the two models have in common can be used. It should be noted, however, that any appropriate type of model, combination of types of models, and/or distance calculations can be used.

[0035] In some embodiments of the disclosed subject matter, once nodes have completed authorization, the nodes can communicate with each other. This communication can take place, for example, by a sender sending traffic (e.g., packets) and a receiver receiving that traffic. It is possible, for example, that a malicious sender harboring a worm unknown to the receiver, has falsified one or more of its models and used those models to be authorized by the receiver. It is also possible that, for example, a sender was authorized with the proper models, but that the sender changes its behavior and now may attack the receiver with malicious traffic. For at least these reasons, some embodiments of the disclosed subject matter require tests during communication to continue to protect the security of a node.

[0036] The receiver can check incoming traffic for malicious content using the models that were exchanged during authorization. For example, referring again to FIG. 5, node 404 entered the network through an authorization phase communicating with node 402. Hence, node 402 can have the output model of node 404 saved in its model table (FIG. 7). When node 404 sends traffic to node 402, node 402 can check for the presence of anomalies in the traffic. For example, node 402 can check if the distance between the input from node

404 and its own input model is smaller than its own traffic threshold. This check can determine whether the input traffic is anomalous compared to the normal input traffic that node 402 uses as its ingress model. Another test can be made of the distance between the input traffic from node 404 and the output model of node 404 to see if it is below the sender's output traffic threshold. This test, can determine whether node 404 lied about its output model or its model of attack behavior when it entered the network. It is also possible, for example, that node 404 lied about its own output traffic threshold so that, given a value high enough, all traffic would be accepted as normal when compared to the output model of node 404. To avoid this, node 402 can redefine the traffic threshold of node 404 as the minimum of its own output threshold and the output threshold of node 404. In some embodiments, additional tests can be used in combination with, or in addition to, the example tests discussed above. In some embodiments, the test can be performed in various orders and that, upon failure of one tests, it may not be necessary to perform further tests. Some embodiments, however, can continue to perform test after a failing test and grant or deny entry based on, for example, the percentage of tests passed by a solicitor.

[0037] Similar to the distance functions used during authorization, the distance functions selected to compare models and traffic during communication, in some embodiments, can vary depending upon, for example, the model type, the traffic sensor being used to examine the traffic, and/or the type of traffic. For example, if the sensor is payload based, the distance can be represented, for example, with the Mahalanobis distance that checks how different are the input traffic packets from the definition of normal content learned by each node. If the sensor is n-gram and/or Bloom filter based, for example, the distance can be measured as the normalized number of unknown n-grams seen at each packet. However, any appropriate model, traffic sensor, and/or distance function can be used.

[0038] Some embodiments of the disclosed subject matter calculate and utilize thresholds to determine if a distance between, for example, models or behaviors is too great. Thresholds related to the models can be used, for example, during authorization and thresholds related to traffic can be used, for example, during communication. However, in some embodiments, various thresholds, including model thresholds and traffic thresholds, can be used during both authorization and communication.

[0039] As discussed, in some embodiments, model thresholds can be used during authorization to determine whether various distances between models and/or behaviors are

within certain model thresholds. These tests can consider, for example, whether the distance between two models is less than or equal to the threshold. The threshold can be calculated as the maximum distance between a model of the receiver and each of the corresponding models in the other nodes of the network. For example, referring to FIG. 4, when authorizing node 404, node 402 can calculate the distances between its input model and the input models of nodes 401 and 403. Node 402 can then use the larger of those two distances as the threshold when comparing its model to the input model of node 404. In some embodiments, a node does not have to calculate theses distances each time it needs a threshold value because the distances can have been calculated during authorizing and been stored in a node's model table (e.g., column 706 of FIG. 7).

[0040] In some embodiments, models in each of the node's model table can be changed, deleted, updated, or refreshed. These changes can occur at various intervals or be instigated by, for example, the network. Some embodiments, for example, can, after the passing of a fixed or configurable amount of time, require all nodes to reauthorize their access to a network. When models are renewed, the model threshold can be, for example, zeroed and recalculated to adapt the new values. Furthermore, during reauthorization, some nodes can be expelled from the network

[0041] In some embodiments, the order in which nodes enter a network can have an effect on the threshold. For example, if the first node to enter has a large distance from the only other node in the network, the threshold can be correspondingly large. Accordingly, some embodiments can include, for example, additional restrictions or considerations in threshold calculation. For example, some embodiments can include a default threshold to be used if, for example, only one node is in network. Additionally, some embodiments can, for example, select the smallest distance between member nodes as a threshold or average the distance among various member nodes to calculate a threshold.

[0042] A traffic threshold can be calculated during the training of a model and can be used, in some embodiments, in the communication phase. A receiver can compare the distance between the incoming traffic and its own input model against an input threshold of the receiver (Th_in_r). Additionally, a receiver can compare this distance between the incoming traffic and the sender's output model against an output threshold of the sender (Th_out_s). The two thresholds, Th_in_r and Th_out_s, can be calculated, for example, during the training of the model built by an anomaly detection sensor installed in each node.

A node can use its own threshold to compare the input traffic against its own input model. When comparing the input traffic against the sender's output model, the threshold used can be the minimum between its own output traffic threshold and the sender's output traffic threshold. The choice of the minimum between the two thresholds can decrease what is considered an acceptable difference between models, when detecting anomalous content, and can enhance performance in situations where a sender may have sent a correct model but lied about its traffic threshold.

[0043] In some embodiments of the disclosed subject matter, a node may not compute its own model. This can be, for example, because the node lacks sufficient computational ability, memory, batter power, traffic, or simply for reasons of design choice. In such embodiments, models can be computed by, for example, a server or another node.

[0044] In some embodiments, a server can generate models for a node. In such embodiments, model training can be performed on a server and the model can be distributed to a node. If, for example, a node has wide-area network (WAN) connectivity, the node can request and download a model, such as the most recent model, from the server. Alternatively, the server could push the model to the node on, for example, a periodic basis. Other embodiments can use a hierarchical distribution where, for example, a single node can download the model over a WAN link and can use a wireless-local-area network (WLAN) link to distribute the updated model to other nodes. This may be useful, for example, if use of the WAN link is expensive. In other embodiments, for example, if no connectivity is available, nodes can be initialized with a model before deployment and synchronized when connectivity is available. This can be useful, for example, for handheld devices, such as personal digital assistants (PDAs), which often have bases stations where synchronization can be performed. However, it should be noted that, in some embodiments, handheld devices can compute their own models.

[0045] In some embodiments, where, for example, a server is not available, a node can be selected as a model-generating node to listen to traffic on the network, generate models, and distribute them to other nodes. The model-generating node can be selected based on any appropriate criteria, for example, processing power, battery power, and/or network connectivity. Alternatively, a node from a first network, which is able to compute a model, can provide models to nodes in a second network that cannot or choose not to generate their own models. In some embodiments, nodes can compute some of their own

models, for example, input and output models, and have a model of attack behavior (e.g., a worm model) calculated for them by another source using, for example, one or more of the above described systems and methods.

[0046] As discussed above, each node in a network 410 can have a model table (e.g., FIG. 7) that can contain, for example, input models, output models, known attack models, traffic thresholds, and distances between the various models. A node can also have a copy of, for example, its own input and output models. Also, as discussed above, in some embodiments, it can be of benefit to update the models as time passes and/or as nodes seem more traffic. Updates to the models can be performed using any appropriate system and method. Some embodiments, for example, can replace an old model with a newly received or generated model. Other embodiments can process the new model in relation to the node's old model, for example, a node can aggregate or merge an old model with a new model.

[0047] Some embodiments, for example, can redefine a node's input and output models as a merge between an old model and a new model of a node whose model has been accepted as having a similar set of known worms and a similar profile during an authorization. Other embodiments can train a model in epochs. This is accomplished by shifting the model update decision to a sensor. At each epoch, a sensor can decide whether a model is good enough or if it should continue to train the model. This can result in an automated update process. It should also be noted that, for example, worm models can be updated periodically with new worm models when, for example, new worms are detected.

[0048] Some embodiments can aggregate models to produce a model that represents a unified view of the current network. Such aggregation can, for example, reduce false suspicions of anomalous behavior and can enable the incremental, decentralized evolution of the network as the nature of the tasks running and the distribution of the nodes changes.

[0049] Some embodiments of the disclosed subject matter can decide how to update a model, for example, based on the similarity of a new model and an old model. For example, if a number of nodes exhibit similar behavior and/or performing similar tasks, it can be of benefit to aggregate those nodes' models to produce a unified view of the network. This can be useful, for example, in reducing false positive detections of suspicious behavior due to small differences in the models.

[0050] Some embodiments of the disclosed subject matter can be implemented, tested, and/or demonstrated in a local area network environment. Such an environment can contain various machines offering various services. Traffic information can be gathered, for example, for http services from port 80. This traffic can be clean traffic with no worms. A set of packets gathered can be partitioned into logical disjoint sets, for example, four sets named A, B, C and D. Furthermore, part of this data (e.g., 80%) can be used to train models and another part (e.g., 20%) can be used as test data. Each set can correspond to a logical node of, for example, a simulated MANET. Hence, data can be randomly drawn to generate training for each ingress and egress model of each node (i.e., set). Training data for an ingress model of one of the four nodes can be randomly drawn from the packets in data corresponding to the three other nodes. For example, the ingress model for node A can be drawn from the packets of nodes B, C, and D. Any appropriate amount of training data can be used. For example, some embodiments can use approximately 40K packets to train each node. Training data for each egress model can be randomly drawn from its own set of packets. For example, the egress model of node A can be trained using the packets in set A.

[0051] An n-gram and/or Bloom filter based sensor can be used to produce A set of input and output models, such as, for example, m_in_A, m_out_B, m_out_C, m_out_D, where m_in_i stands for i's input model and m_out_i stands for i's output model. For purposes of explanation, it can be assumed that A is the receiver node in a MANET, and that B, C, and D will try to enter the MANET, and if successful, communicate with A.

[0052] Some embodiments can calculate thresholds for the distances between worm models and normal models (i.e., Dist(mw_i, m_j) and Dist(m_i, m_j)). It may be of benefit to find thresholds that differentiate normal models from worm models and that divide normal models from suspicious models (e.g., models built for nodes that have seen a small amount of malicious traffic). A model can have a corresponding threshold value used to decide whether a test value (e.g. input traffic and/or output traffic) is considered close to that model. Because, in some embodiments, a node can test incoming traffic from a sending node, which was previously granted access, two models can be tested. A receiving node can have its own model of input that it considers to be normal. This model can have a threshold, Th_a_i(2). A sending node can send its own egress model to a receiving node. A receiving node can test an egress model to determine whether the sending node is sending traffic considered to be too different from the behavior described in the egress model. A threshold, Th_a_i(3), can be

used for this purpose. Models with distances to the receiver's worms model greater than a threshold Th_a_i(2) or with distances to the receiver's normal model greater than an estimated threshold, Th_a_i(3) can be rejected.

[0053] Some embodiments can compute normal and worm models for each of the nodes using, for example, detectors based on Bloom filters as described above. Any appropriate Bloom filter size can be chosen. For example, a Bloom filter size of 2^{20} bits, which can be an appropriate size in, for example, some MANETS, to be exchanged among nodes.

[0054] As discussed, a distance threshold can be calculated to separate normal models from models that contain worms. Various output models can be obtained from a network to accomplish this. A size of n-gram (e.g., a 5-gram) can also be selected as the building size for the models. The distance between A's input model and B, C, and D's output models can be calculated. The distance can be defined as a squared Euclidean distance of

th1 = distance(A, Y) = cardinality(A AND Y) / cardinality(A), and its asymmetric,

th2 = distance(Y, A) = cardinality(Y AND A) / cardinality(Y), where "AND" is logical AND, A is A's normal model, and Y is the set of B's, C's, and D's normal models.

[0055] During a testing phase, the models used to obtain thresholds can be infected with known worms. Worms can be chosen to attack the port 80 http service. Any appropriate worms can be selected, for example, iiswebdav, iismedia, crii, and php. Using the distance threshold, th1, obtained during training, many of the worms that populate the models can be captured. Some worms, which may not be detected during training, can be detected during communication. This can be because, in some embodiments, detection during the communication phase can be done at the packet level. This can be true, for example, of the php worm.

[0056] Threshold distances between a bad Bloom filter of A (e.g., a collection of known worms and viruses) and a set of normal output models of B, C and D can also be calculated. The threshold distances can define the maximum distance between normal models and the bad Bloom filter. A distance above the threshold can mean that the model is too close to the bad Bloom filter (i.e., it may indicate a worm or other abnormal behavior).

Threshold distances are calculated as squared Euclidean distances, as describe above. However, in this case, A is A's bad Bloom filter

[0057] During the testing phase, the normal models can be infected with known worms (e.g., iiswebdav, iismedia, crii, and php) and the distances between these models and A's bad Bloom filter can be calculated to see whether the distances are, for example, above the threshold th2 (i.e., the model is too close to the bad Bloom filter and it may be infected with a worm) or below the threshold th2 (i.e., the model is within the limits of normalcy learned during the training phase).

[0058] Although the invention has been described and illustrated in the foregoing illustrative embodiments, it is understood that the present disclosure has been made only by way of example, and that numerous changes in the details of implementation of the invention can be made without departing from the spirit and scope of the invention, which is limited only by the claims that follow. The disclosed subject matter can be used to prevent attacks in addition to the illustrative example attacks described above. It should be noted that features of the disclosed embodiments can be combined and rearranged in various ways.

What is claimed is:

1. A method for securing communications between a first node and a second node, comprising:

receiving at least one model of behavior of the second node at the first node; and

authorizing the first node to receive traffic from the second node based on the difference between the at least one model of behavior of the second node and at least one model of behavior of the first node.

2. The method of claim 1, further comprising determining a distance between the at least one model of behavior of the second node and the at least one model of behavior of the first node.

3. The method of claim 2, further comprising comparing the distance to a threshold.

4. The method of claim 3, further comprising calculating the threshold based on the distance between the at least one model of behavior of the first node and at least one model of behavior of at least one other node.

5. The method of claim 1, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a model of output behavior.

6. The method of claim 1, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a model of input behavior.

7. The method of claim 1, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a model of anomalous behavior.

8. The method of claim 1, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a Bloom filter.

9. The method of claim 1, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is the output of n-gram frequency analysis.

10. The method of claim 1, further comprising rejecting traffic sent to the first node from the second node based on the difference between the at least one model of behavior of the second node and actual behavior of the second node.

11. The method of claim 10, wherein the actual behavior of the second node is its output behavior.

12. The method of claim 1, further comprising:

receiving at least one model of behavior of the first node at the second node; and

authorizing the second node to receive traffic from the first node based on the difference between the at least one model of behavior of the first node and the at least one model of behavior of the second node.

13. A device that secures communications between a first node and a second node, comprising:

an interface in communication with a network;

a memory; and

a processor in communication with the memory and the interface; wherein the processor:

receives at least one model of behavior of the second node; and

authorizes the first node to receive traffic from the second node based on the difference between the at least one model of behavior of the second node and at least one model of behavior of the first node.

14. The device of claim 13, wherein the processor further determines a distance between the at least one model of behavior of the second node and the at least one model of behavior of the first node.

15. The device of claim 14, wherein the processor further compares the distance to a threshold.

16. The device of claim 15, where the processor further calculates the threshold based on the distance between the at least one model of behavior of the first node and at least one model of behavior of at least one other node.

17. The device of claim 13, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a model of output behavior.

18. The device of claim 13, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a model of input behavior.

19. The device of claim 13, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a model of anomalous behavior.

20. The device of claim 13, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a Bloom filter.

21. The device of claim 13, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is the output of n-gram frequency analysis.

22. The device of claim 13, where the processor further rejects traffic sent from the second node based on the difference between the at least one model of behavior of the second node and actual behavior of the second node.

23. The device of claim 22, wherein the actual behavior of the second node is its output behavior.

24. A computer-readable medium containing computer-executable instructions that, when executed by a processor, cause the processor to perform a method for securing communications between a first node and a second node, comprising:

receiving at least one model of behavior of the second node; and

authorizing the first node to receive traffic from the second node based on the difference between the at least one model of behavior of the second node and at least one model of behavior of the first node.

25. The computer-readable medium of claim 24, the method further comprising determining a distance between the at least one model of behavior of the second node and the at least one model of behavior of the first node.

26. The computer-readable medium of claim 25, the method further comprising comparing the distance to a threshold.

27. The computer-readable medium of claim 26, the method further comprising calculating the threshold based on the distance between the at least one model of behavior of the first node and at least one model of behavior of at least one other node.

28. The computer-readable medium of claim 24, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a model of output behavior.

29. The computer-readable medium of claim 24, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a model of input behavior.

30. The computer-readable medium of claim 24, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a model of anomalous behavior.

31. The computer-readable medium of claim 24, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is a Bloom filter.

32. The computer-readable medium of claim 24, wherein at least one of the at least one model of behavior of the second node and the at least one model of behavior of the first node is the output of n-gram frequency analysis.

33. The computer-readable medium of claim 24, the method further comprising rejecting traffic from the second node based on the difference between the at least one model of behavior of the second node and actual behavior of the second node.

34. The computer-readable medium of claim 33, wherein the actual behavior of the second node is its output behavior.

35. A method for securing communications between a first node and a second node, comprising:

receiving at least one model of behavior of the second node at the first node; and

rejecting traffic sent from the second node to the first node based on the difference between the at least one model of behavior of the second node and at least one model of behavior of the first node.

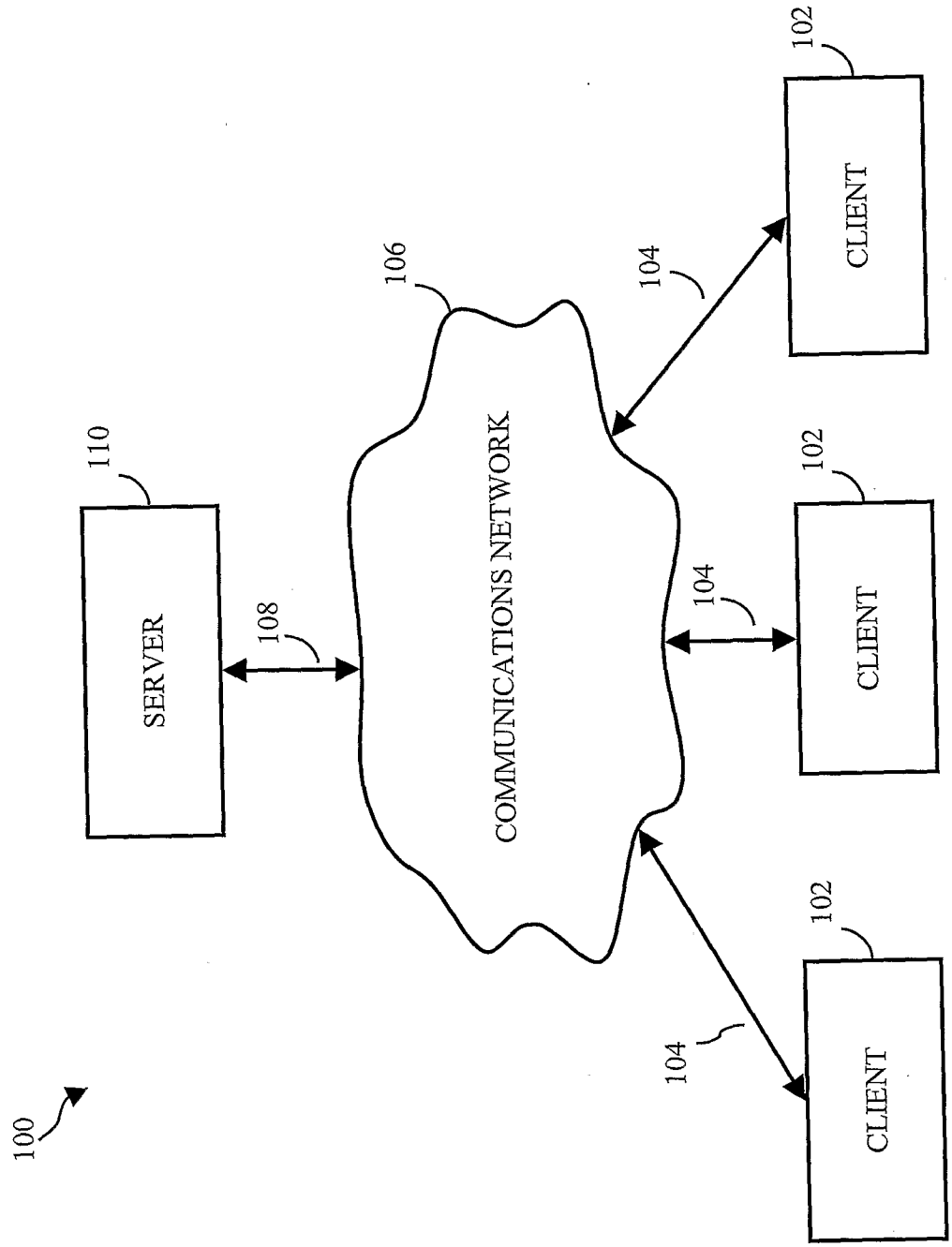

100
SERVER
110
108
COMMUNICATIONS NETWORK
106
104
CLIENT
102
104
CLIENT
102
104
CLIENT
102

FIG. 1

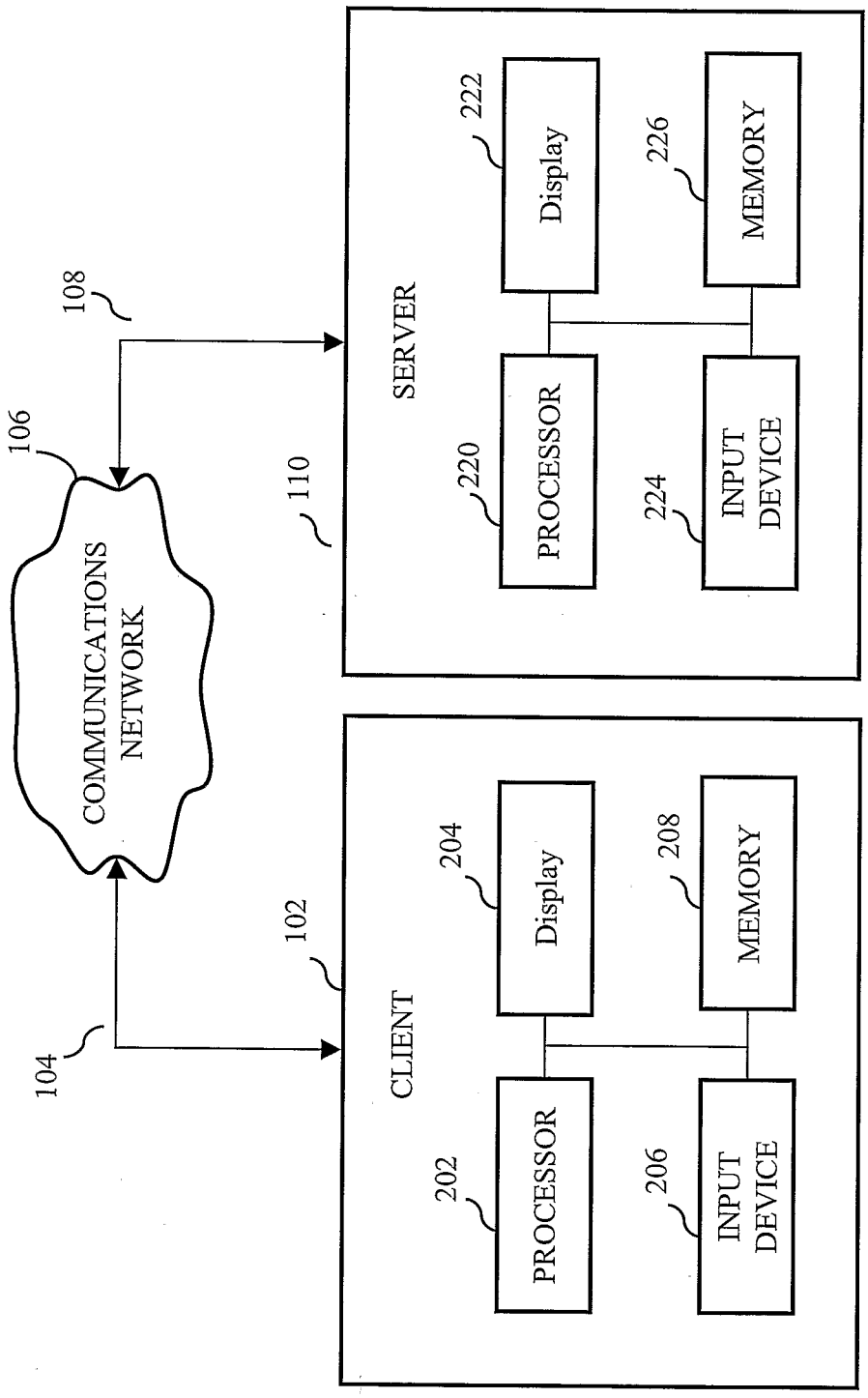

108
106
COMMUNICATIONS NETWORK
104
102
CLIENT
202
PROCESSOR
204
Display
206
INPUT DEVICE
208
MEMORY
110
SERVER
220
PROCESSOR
222
Display
224
INPUT DEVICE
226
MEMORY

FIG. 2

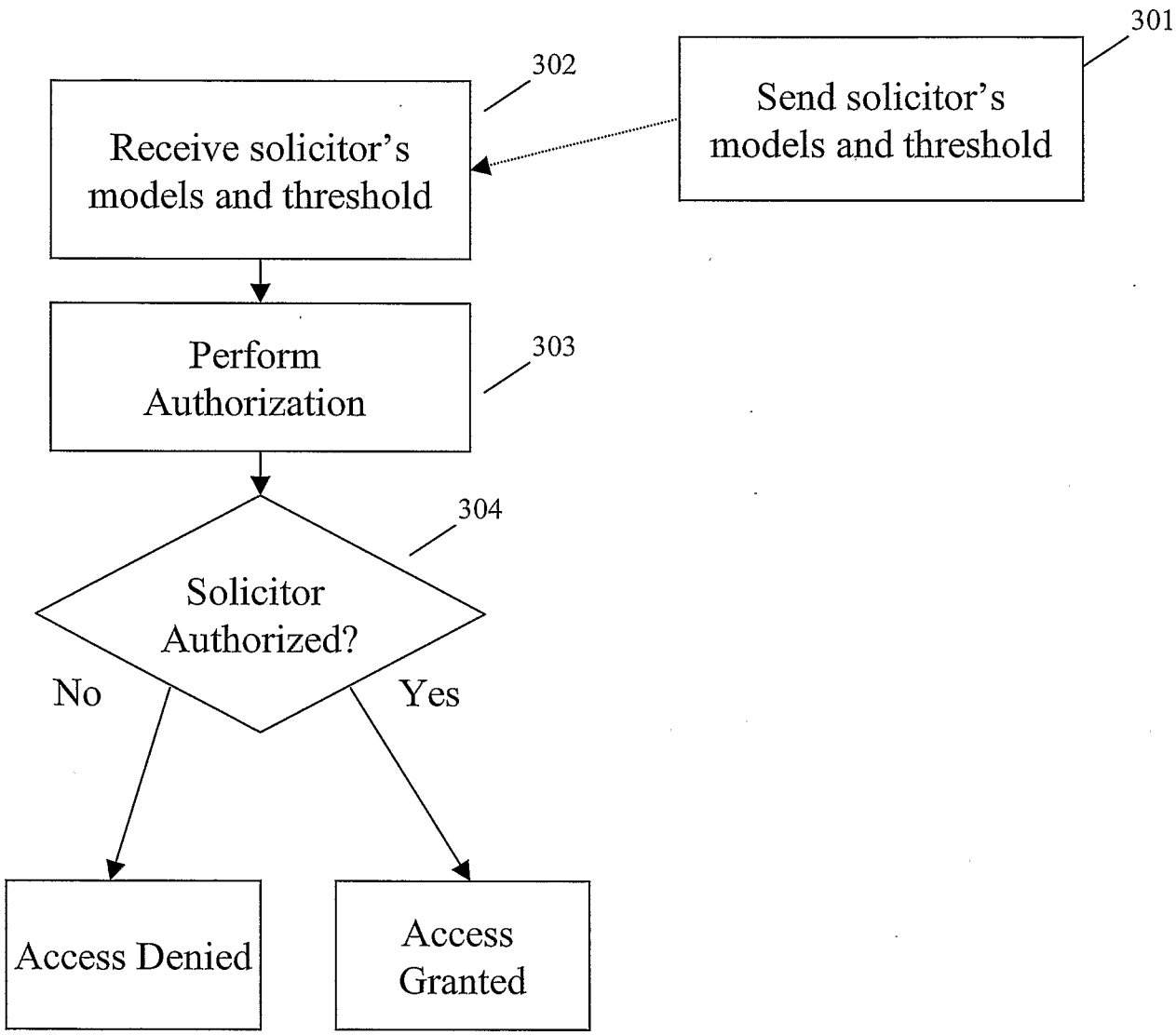

Receiver Node
Solicitor Node
301
Send solicitor's models and threshold
302
Receive solicitor's models and threshold
Perform Authorization
303
304
Solicitor Authorized?
No
Yes
Access Denied
Access Granted

FIG. 3

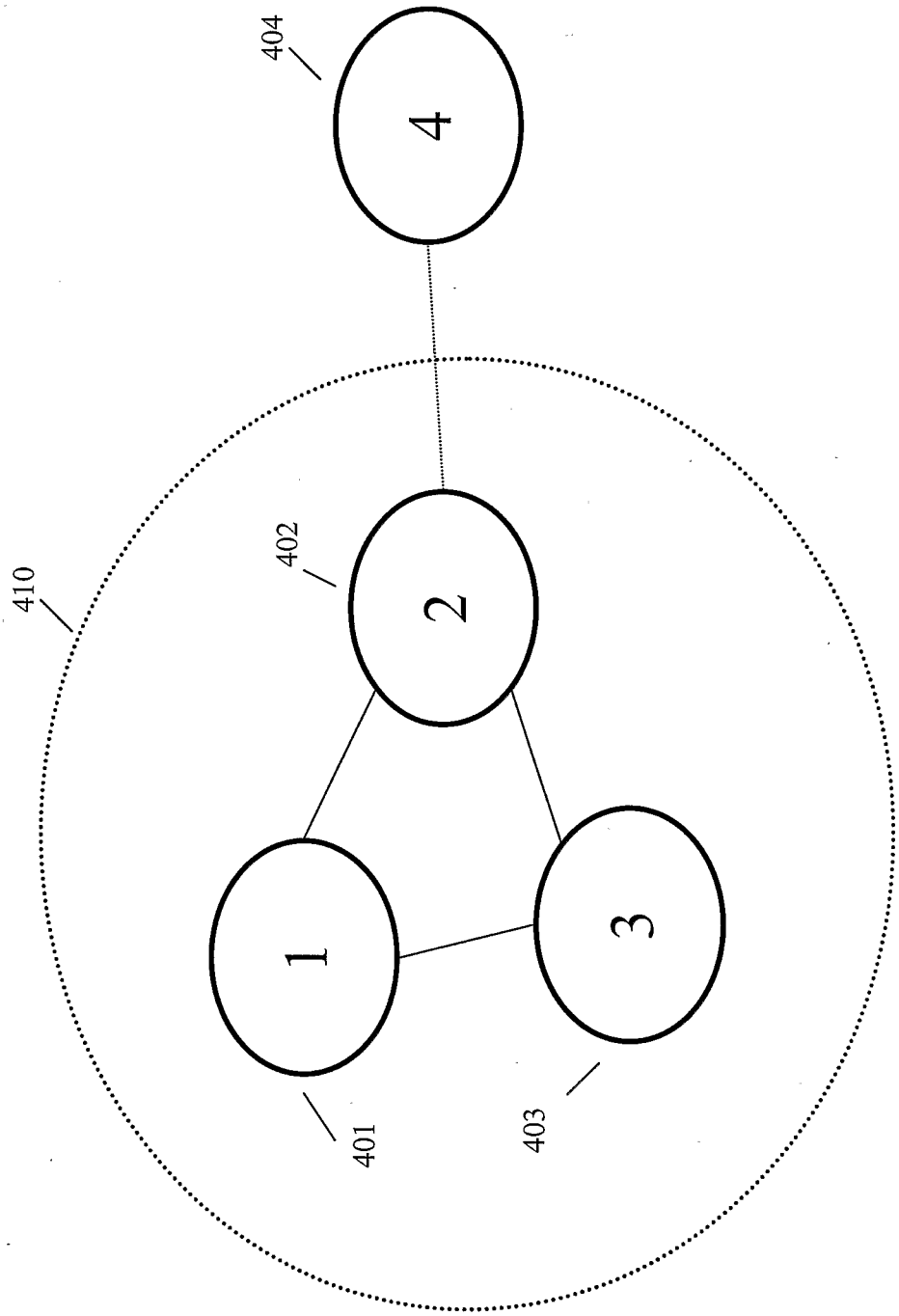


FIG. 4

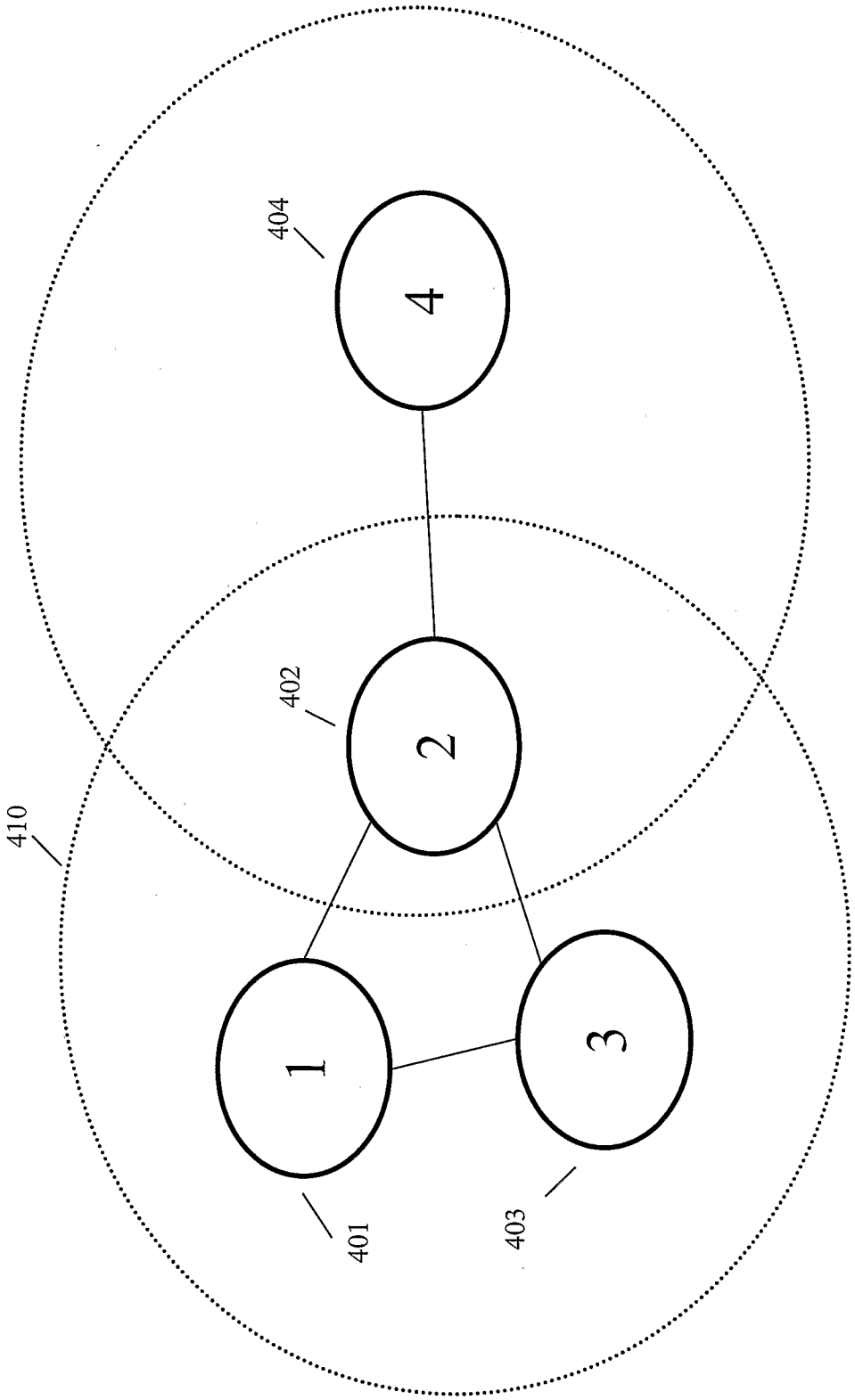


FIG. 5

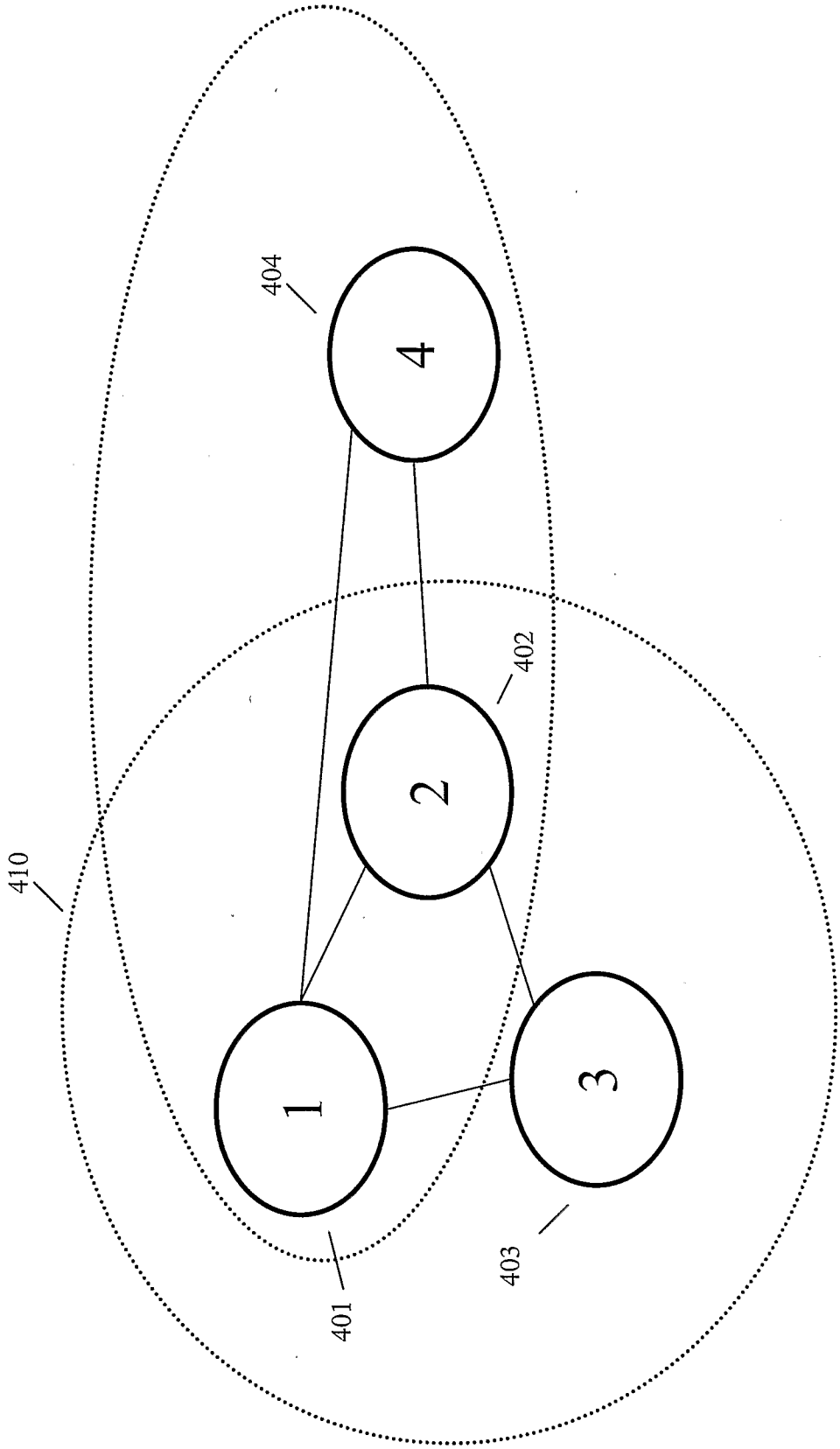


FIG. 6

701	702	703	704	705	706
Node 1	m_in_1	m_out_1	mw_1	Thc_out_1	dist{{in,out,w}2,{in,out,w}1}
Node 3	m_in_3	m_out_3	mw_3	Thc_out_3	dist{{in,out,w}2,{in,out,w}3}
Node 4	m_in_4	m_out_4	mw_4	Thc_out_4	dist{{in,out,w}2,{in,out,w}3}

Model Table for Node 2

FIG. 7