



(19) **United States**  
(12) **Patent Application Publication**  
**Hofer**

(10) **Pub. No.: US 2015/0365413 A1**  
(43) **Pub. Date: Dec. 17, 2015**

(54) **SECURE CONFIGURATION OF AUTHENTICATION SERVERS**

**Publication Classification**

- (71) Applicant: **Emulex Corporation**, Costa Mesa, CA (US)
- (72) Inventor: **Larry Dean Hofer**, Costa Mesa, CA (US)
- (21) Appl. No.: **14/835,895**
- (22) Filed: **Aug. 26, 2015**

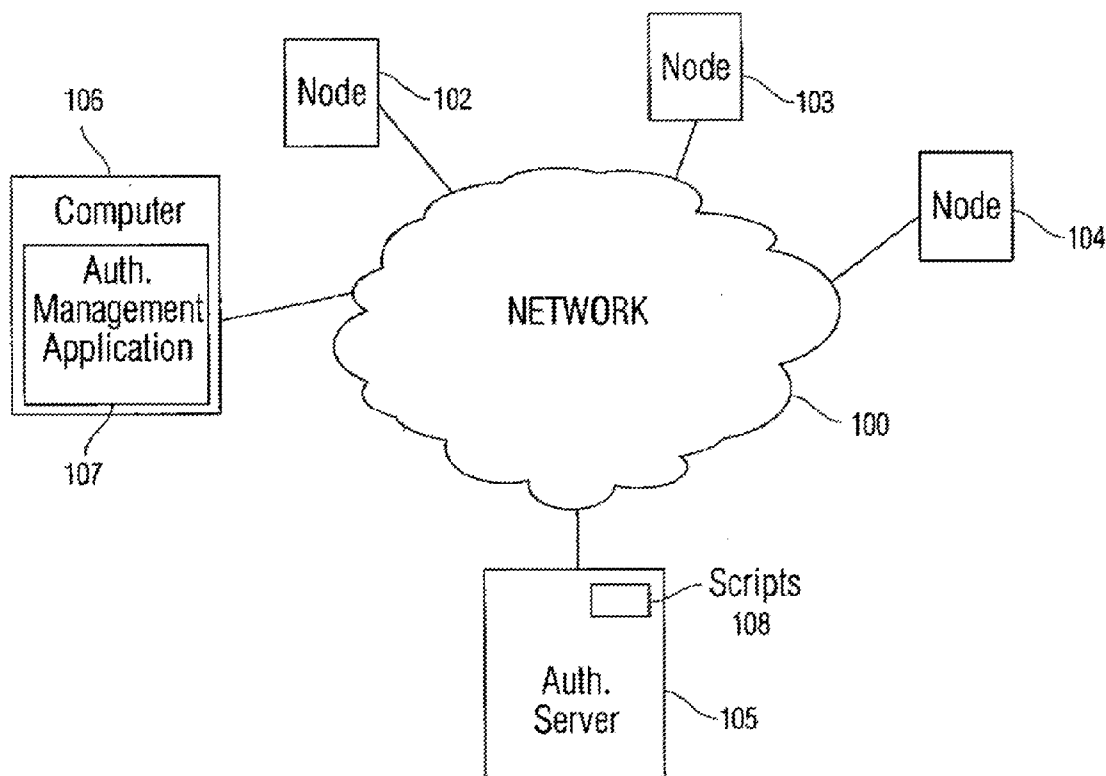
- (51) **Int. Cl.**  
*H04L 29/06* (2006.01)  
*G06F 17/30* (2006.01)  
*G06F 21/60* (2006.01)
- (52) **U.S. Cl.**  
CPC ..... *H04L 63/0884* (2013.01); *H04L 63/083* (2013.01); *H04L 63/20* (2013.01); *G06F 21/602* (2013.01); *G06F 17/30876* (2013.01); *G06F 17/30312* (2013.01)

**Related U.S. Application Data**

- (63) Continuation of application No. 14/516,205, filed on Oct. 16, 2014, now Pat. No. 9,148,412, which is a continuation of application No. 13/970,500, filed on Aug. 19, 2013, now Pat. No. 8,892,602, which is a continuation of application No. 12/123,401, filed on May 19, 2008, now Pat. No. 8,515,996.

(57) **ABSTRACT**

Embodiments of the invention are directed to automatically populating a database of names and secrets in an authentication server by sending one or more lists of one or more names and secrets by a network management software to an authentication server. Furthermore, some embodiments provide that the lists being sent are encrypted and/or embedded in otherwise inconspicuous files.



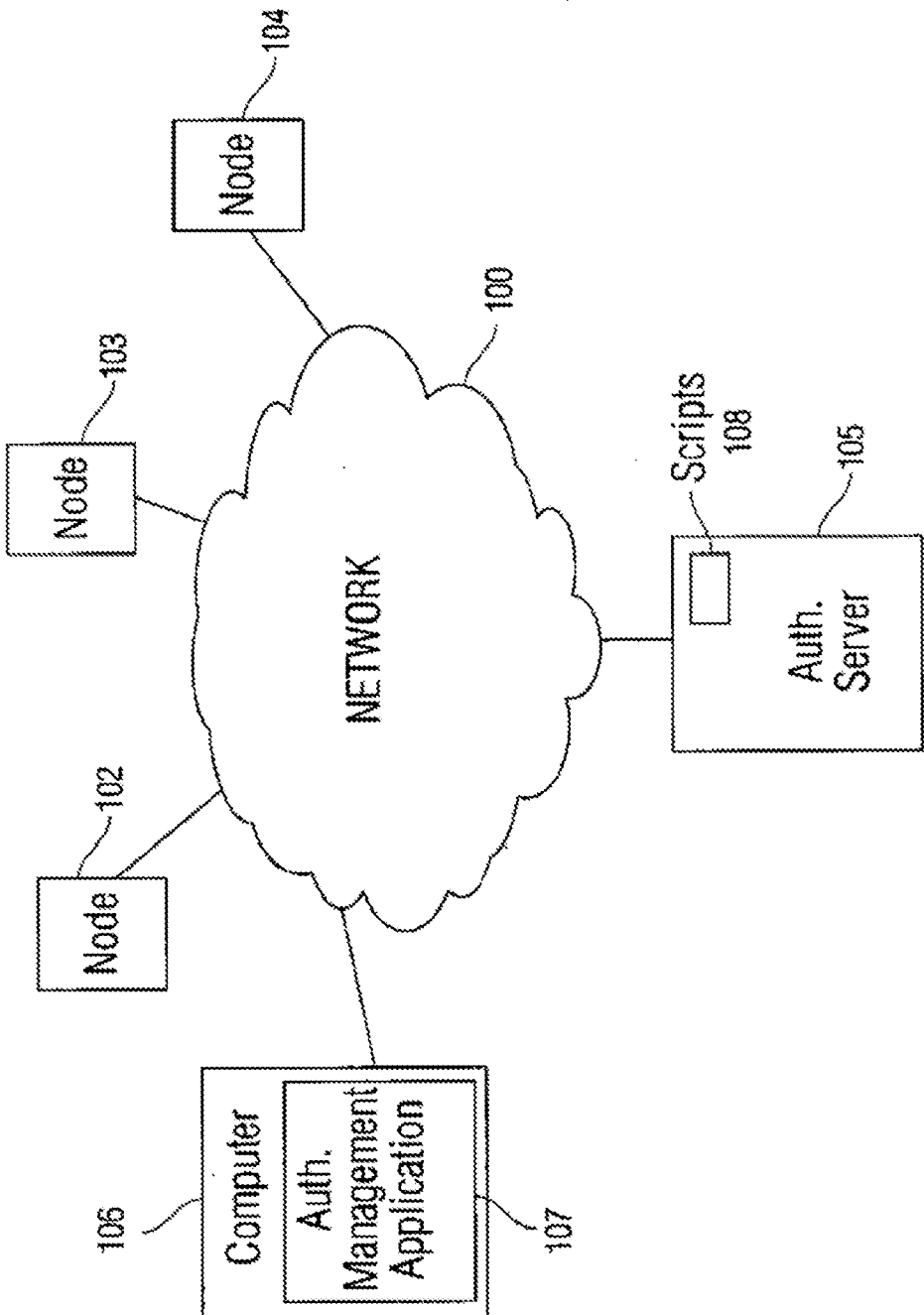


Fig. 1

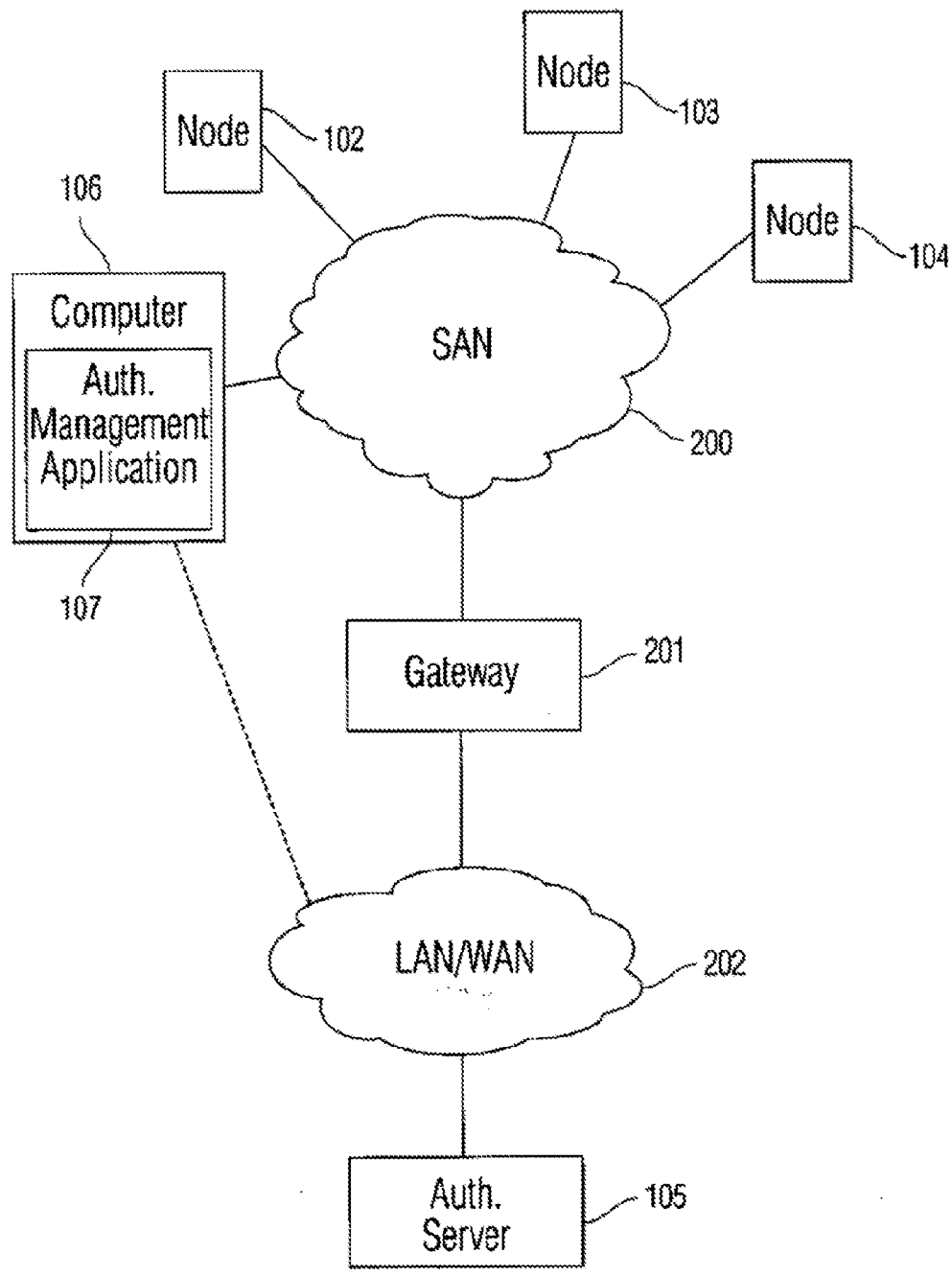


Fig. 2

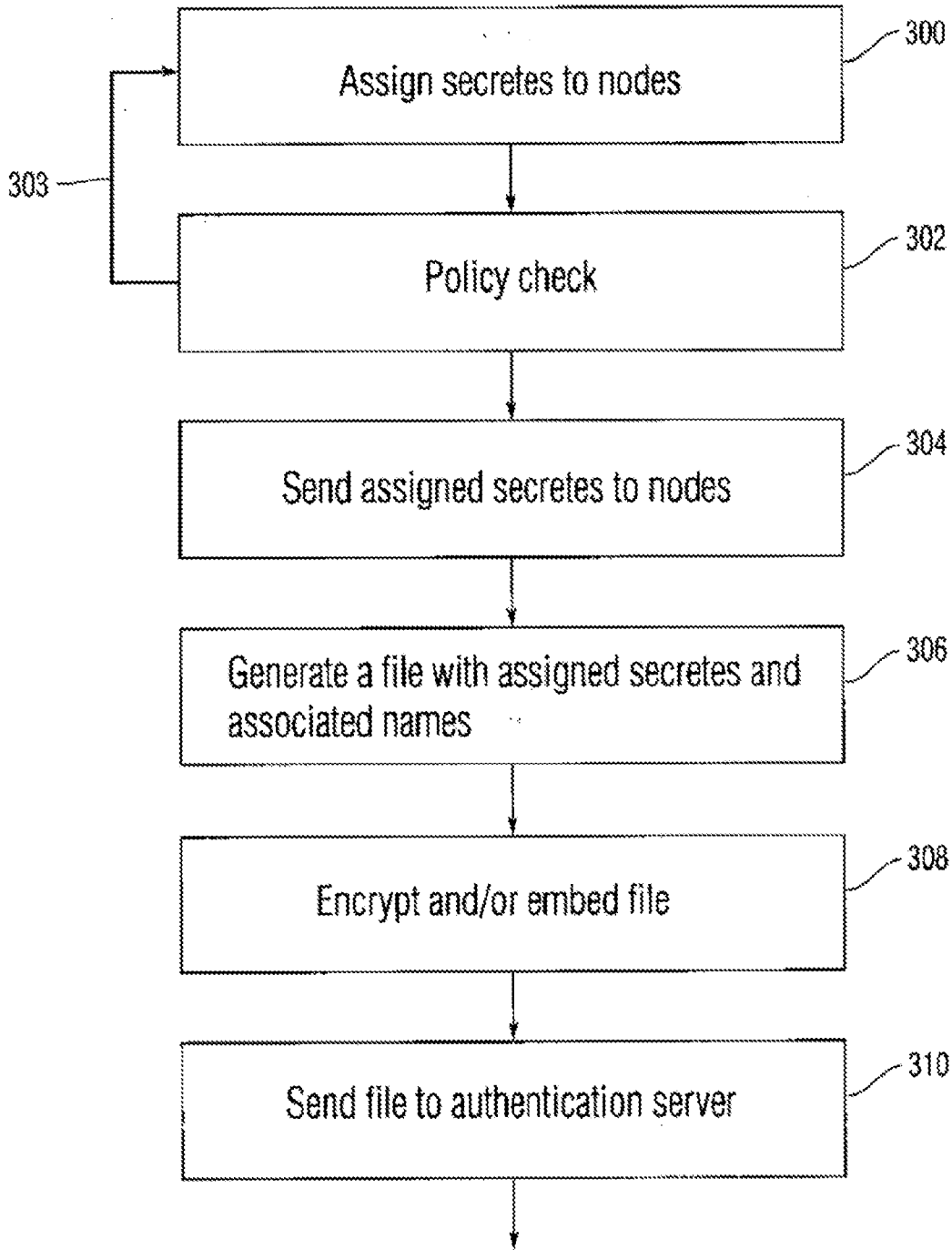


Fig. 3A

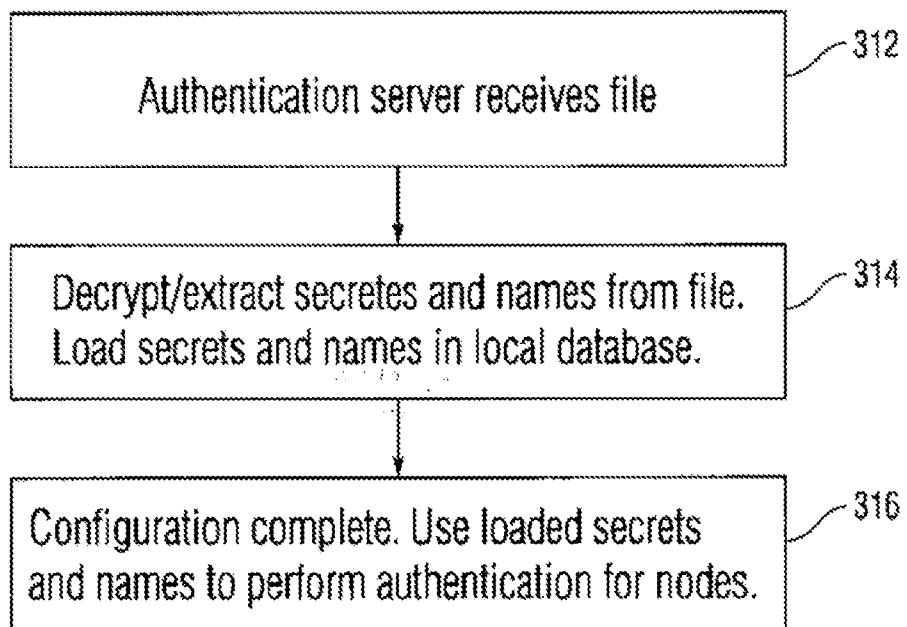


Fig. 3B

**SECURE CONFIGURATION OF AUTHENTICATION SERVERS**

**FIELD OF THE INVENTION**

[0001] This relates generally to authentication, and more particularly, to secure and automatic configuration of an authentication server with device names and secrets.

**BACKGROUND OF THE INVENTION**

[0002] An authentication server is a server that provides authentication for various devices. For example, an authentication server may provide authentication for devices on a network. Each device on the network may be assigned a unique name that identifies that device on the network. For authentication purposes, each device may also be assigned a secret (such as a password). Each device may provide its name and secret in order to authenticate itself (i.e., prove that it is in fact the device it purports to be). An authentication server may store the assigned names and secrets of the various devices and use them to determine whether authentications taking place on the network are successful. Authentication servers are generally known in the art. For example, two known types of authentication servers are Remote Authentication Dial In User Service (RADIUS) and DIAMETER servers. RADIUS and DIAMETER are also names of the protocols these servers follow. While RADIUS refers to "Dial In" it is no longer exclusively used for dial in networks and may apply to other types of networks.

[0003] As noted above, an authentication server must have the correct name and secret of a device it is to authenticate before it attempts to authenticate it. In many types of networks (such as for example, storage area networks or SANs) the authentication server is loaded with these names and secrets manually (e.g., by a user typing the names and secrets in an authentication server interface). This process can be time consuming, expensive, insecure and susceptible to user errors.

**SUMMARY OF THE INVENTION**

[0004] Embodiments of the invention are directed to automatically populating a database of names and secrets in an authentication server by sending one or more lists of one or more names and secrets by a network management software to an authentication server. Furthermore, some embodiments provide that the lists being sent are encrypted and/or embedded in otherwise inconspicuous files.

**BRIEF DESCRIPTION OF THE DRAWINGS**

[0005] FIG. 1 is a diagram of an exemplary network according to embodiments of the present invention.

[0006] FIG. 2 is another diagram of an exemplary network according to embodiments of the present invention.

[0007] FIGS. 3A and 3B are a flowchart showing an exemplary method of configuration of an authentication server according to embodiments of the present invention.

**DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

[0008] In the following description of preferred embodiments, reference is made to the accompanying drawings which form a part hereof, and in which it is shown by way of illustration specific embodiments in which the invention can

be practiced. It is to be understood that other embodiments can be used and structural changes can be made without departing from the scope of the embodiments of this invention.

[0009] Embodiments of the invention are directed to automatically populating a database of names and secrets in an authentication server by sending one or more lists of one or more names and secrets by a network management software to an authentication server. Furthermore, some embodiments provide that the lists being sent are encrypted and/or embedded in otherwise inconspicuous files.

[0010] While the following description centers on Fibre Channel (FC) storage area networks, embodiments to the present invention are not limited to these networks. Accordingly, embodiments of the invention can be used in conjunction with other types of storage network such as, for example, iSCSI networks, Fibre Channel over Ethernet (FCoE), serial attached SCSI (SAS) networks, etc. Furthermore, embodiments of the present invention can be used with various non-SAN networks that may use similar authentication mechanisms as the networks discussed above.

[0011] FIG. 1 is a diagram of an exemplary network according to some embodiments of the invention. Network 100 can be a SAN, such as an FC, iSCSI, or FCoE SAN, or another type of network. Network 100 can be a combination of two or more networks.

[0012] A plurality of nodes may be connected to network 100. These can include various types of computers or network enabled electronic devices such as, for example, storage servers and/or redundant array of inexpensive drives (RAID) controllers connected to a storage area network. The nodes can be configured to communicate with each other according to a protocol that may require or allow authentication. For example, in the case of a Fibre Channel network, the nodes can communicate through the Fibre Channel protocol and use the DH CHAP protocol for authentication. In an iSCSI network, the CHAP protocol can be used for authentication. Each node of nodes 102-104 can store a name and secret associated with itself and use it when authenticating itself to other nodes.

[0013] Authentication server 105 can be server used to assist nodes 102-104 in authentication. Server 105 can be, for example, a RADIUS or a DIAMETER server. It should be noted that RADIUS and DIAMETER are designations of protocols according to which server 105 can operate, and not designations of specific implementations. For example, there are several different types of RADIUS server software available today from different vendors, such as the Internet Authentication Service provided by Microsoft, Alepo Radius Server, FreeRADIUS, etc. The authentication server 105 can be a generally programmable computer executing an authentication server software (such as one of the software products listed above) or an application specific hardware appliance (such as, for example, the Infoblox RADIUSone). The authentication server may also be implemented at one of the network elements of network 100, such as, for example, as software running at a switch in network 100.

[0014] The authentication server may store the names and secrets of the nodes and verify authentications of the nodes. For example, if node 102 needs to authenticate node 103, it can send a challenge to node 103. Node 103 can respond to the challenge with a message including its name and a response that is the results of cryptographic hash of the secret and other values including the challenge value back to node 102. (In

some cases, the challenge can include a random number, and node **103** can use the random number in building the response that is a cryptographic hash value (i.e. one way encryption) of the secret before sending it across the network). Node **102** can then send node **103**'s name and secret to the authentication server **105**. The authentication server **105** can check the received name and secret of node **103** against respective values stored in its local database, and determine if there is a match. If there is a match the authentication is successful. If there is no match, the authentication has failed. The authentication server can send a message back to node **102** indicating whether the authentication was successful. Based on this message, node **102** can determine whether node **103** is really the node it purports to be.

**[0015]** Thus, nodes **102-104** can authenticate each other without necessitating that each node store all the authentication data of all other nodes (which would make authentication very insecure). Therefore, nodes **102-104** can be referred to as clients of authentication server **105** and can include hardware or software that performs authentication client functionality according to a relevant protocol (such as, for example, RADIUS or DIAMETER).

**[0016]** Computer **106** may also be connected to network **100**. Computer **106** may run an authentication management application **107**. Application **107** can be an application that manages the authentication of a set of network connected devices, such as nodes **102-104**. The authentication management application may also provide additional network management functionalities and may be referred to as a network management application. Alternatively, computer **106** can be an application specific appliance that is configured to perform the authentication management function.

**[0017]** In some embodiments, the authentication management application **107** may be implemented at one of the network elements of network **100**, such as, for example, as software running at a switch in network **100**. In some embodiments, the authentication management application may be implemented at the authentication server **105**, or may be executed at the same computer as the authentication server.

**[0018]** The authentication management application may obtain the names of the node it manages. It may perform this by using standard discovery protocols of network **100**, or it may allow a user to enter the names of the nodes to be managed. Having obtained the names, the authentication management application can assign or otherwise establish secrets to the nodes it manages (i.e., nodes **102-104**), and associate each secret with the name of the node it is assigned to. After assigning or establishing the secrets, the authentication management application may send to each node its assigned secret. The secrets may be encrypted during transmission. The authentication management application can assign secrets automatically, by a predefined algorithm by the use of random number generation or other key agreement protocols. Alternatively it may allow a user of the application to enter the secrets. In some embodiments, nodes **102-104** can assign their own secrets and send their secrets to the authentication management application **107** through network **100**. The authentication management application may store the secrets assigned to devices **102-104** locally.

**[0019]** In some embodiments, the authentication management application may also perform a policy check of the assigned secrets. In some embodiments, there may exist rules for what types of secrets may be used. These rules may be used to prevent the assignment of "weak" secrets (i.e., secrets

or passwords that are easy to compromise). Thus, the rules may specify that the secrets must be of a predefined length, a predefined complexity, that they must exclude dictionary words, etc. If a given secret entered by a user or provided by a node fails the policy check, the authentication management application may refuse to assign that secret and may request the user or the node to provide another secret.

**[0020]** Once the authentication management application has assigned all secrets, it may assemble the secrets and the respective names of their associated nodes in a single data structure (such as, for example, a file) and send them to the authentication server. The authentication server can load the secrets and associated names in its local database and use them for authentication of nodes **102-104** as discussed above. In order to send the file of names and secrets to the authentication server, the authentication management application may also include authentication client functionality (such as, for example, RADIUS or DIAMETER client functionality).

**[0021]** The authentication management application may encrypt the file before it sends it to the authentication server. Alternatively, or in addition, the authentication management application may also use steganography to embed the file with names and secrets in another file. Steganography is a known technique of embedding a first set of data in a second set of data. Thus, the file with names and secrets can be embedded in a more inauspicious looking file, such as a video or an audio file. Accordingly, steganography may be used to provide additional security by preventing any rogue device that may be monitoring network communications from flagging the transmitted file as important. The authentication server can extract the file with names and secrets from the other file and/or decrypt it.

**[0022]** The clients of authentication server (such as nodes **102-104** and authentication management software **107**) may be required to provide an authentication server password to communicate with the authentication server. In some embodiments, that same password or derivation thereof may be used to encrypt the file of secrets sent by the authentication management application **107** to the authentication server.

**[0023]** In some embodiments, the file including the list of names and secrets can be arranged in a predefined interoperable format. The format can be referred to as interoperable because it may be readable and processable by different types of authentication servers offered by different vendors.

**[0024]** In some embodiments, additional software **108** (such as, for example, one or more scripts) may reside at the authentication server **105** or possibly on the computer **106**. The additional software **108** may be used to send or receive or translate the file including the names and secrets of the nodes, perform any necessary decryptions and/or extraction and load the file into a database for authentication server **105**. Thus, additional software **108** may allow embodiments of the present invention to be used with various existing authentication server software products, without having to further modify them for the purposes of the present invention.

**[0025]** Furthermore, different versions of additional software **108** may be developed for different versions of existing authentication servers. The different versions may be configured to operate with a single format of the file comprising the names and secrets of the nodes which is to be generated and sent by the authentication management application. Thus, embodiments of the present invention may provide for that a single authentication management application may be interoperable with different types of authentication servers.

In some embodiments, once added to the authentication server, the additional software **108** may be considered to be part of the authentication server or the authentication management application.

[0026] FIG. 2 is a diagram of another exemplary network according to embodiments of the present invention. More specifically, FIG. 2 is a diagram of a special case of the network of FIG. 1 involving a storage area network. In FIG. 2, the generic network **100** of FIG. 1 actually comprises two connected networks—**200** and **202**. Network **200** can be a storage area network, such as, for example, an FC, iSCSI, FCoE and/or SAS networks. As noted above, nodes **102-104** can be various nodes that may connect to a SAN such as, for example, file servers, web servers, RAID arrays, etc.

[0027] Network **202** can be a local area network (LAN) or a wide area network (WAN). LAN/WAN **202** can be, for example, an Ethernet network. Gateway **201** can be a device for interconnecting to different types of networks such as SAN **200** and LAN/WAN **202**.

[0028] As shown, nodes **102-104** as well as the computer **106** hosting the authentication management application **107** can be connected to SAN **200**. Thus, they may communicate with authentication server **105** through gateway **201**. In some embodiments, computer **106** may be alternatively connected to LAN/WAN **202**. Alternatively, computer **106** may be simultaneously be connected to both networks **200** and **202**.

[0029] As shown, authentication server **105** may be connected to network **202**. The embodiment of FIG. 2 may be beneficial because most existing authentication servers are not configured for direct connection with most existing SANs and instead require connection to an ordinary Ethernet network. Furthermore, some embodiments may provide that authentication server **105** serve multiple SANs connected to LAN/WAN **202**. In some embodiments, multiple authentication servers can be present. In some embodiments, a single authentication management application can serve multiple authentication servers.

[0030] As noted above, the combination of network **200**, gateway **201** and network **202** may be considered to be a single heterogeneous network.

[0031] FIGS. 3A and 3B are a flowchart showing an exemplary method of configuration of an authentication server according to embodiments of the present invention. Referring to FIG. 3A, at step **300**, the authentication management application assigns secrets to various nodes. At step **302**, the authentication management application performs a policy check on the secrets. If one or more of the secrets fail the policy check they may need to be reassigned (arrow **303**). At step **304**, the authentication management application sends the assigned secrets to their respective nodes. The secrets may be encrypted before being sent to the nodes. As noted above, other embodiments may provide that the nodes generate their own secrets and forward them to the authentication management application instead. Or in other embodiments, the authentication management application may use a protocol to establish the secret for a remote node.

[0032] At step **306**, the authentication management application generates a data structure including a list of the assigned secrets and the names of the devices the respective secrets are associated with. As noted above, the data structure may, but need not necessarily be a file. At step **308**, the file is encrypted and/or embedded in another file using steganography. At step **310**, the file is sent to the authentication server.

[0033] Continuing to FIG. 3B, at step **312**, the authentication server receives the file. At step **314**, the authentication server decrypts and/or extracts the secrets and names from the received file and loads them in its local database. At step **316**, configuration is complete. The nodes being served by the authentication server (e.g., nodes **102-104** of FIGS. 1 and 2) may resume normal operation and the authentication server may use the stored names and secrets to perform authentication for these nodes.

[0034] Thus, authentication can be performed without having to manually enter nodes' names and secrets into an authentication server. This may greatly improve the ease of administration of a network. This may also improve the security and reliability of a network, as manual entry of names and secrets into an authentication server may be considered to be a security vulnerability as well as a source of inadvertent entry errors.

[0035] Various devices discussed herein, such as authentication server **105**, computer **106**, and nodes **102-104** may run on programmable computers, or other types of programmable electronic devices. A programmable device may include a processor, such as a central processing unit (CPU) and a memory. The processor may operate by executing instructions stored in the memory. These instructions may comprise, for example, the authentication management application **107**, one of various authentication server software products executing at authentication server **105**, and/or scripts **108**. The memory can also store data, such as the authentication server's local database, the file of node names and secrets, etc.

[0036] The various devices can also include networking hardware. The networking hardware may include, for example, a network interface card (NIC—often used for connecting to Ethernet networks) and/or a host bus adapter (HBA—often used for connecting to FC or iSCSI networks). Some devices may include both a NIC and an HBA. For example, in the embodiment in which the computer **106** is connected to both SAN **200** and LAN/WAN **202**, computer **106** may include a NIC for connecting to LAN/WAN **202** and an HBA and its associated driver software for connecting to SAN **200**.

[0037] Although embodiments of this invention have been fully described with reference to the accompanying drawings, it is to be noted that various changes and modifications will become apparent to those skilled in the art. Such changes and modifications are to be understood as being included within the scope of embodiments of this invention as defined by the appended claims.

1. A method for configuring a management database comprising:

- assigning a plurality of secrets to a plurality of nodes of a network;
- generating a data structure in each node of the plurality of nodes, the data structure comprising the assigned secrets;
- securing the data structure in each node of the plurality of nodes using a password generated according a key agreement protocol; and
- storing each generated password in the management database.

2. The method of claim 1, further comprising integrating the data structure into an authentication server database.

3. The method of claim 1, wherein securing the data structure comprises encrypting the data structure.



4. The method of claim 1, wherein encrypting the data structure comprises encrypting the data structure with a password used for communications with the authentication server or a derivation thereof.

5. The method of claim 1, wherein securing the data structure comprises embedding the data structure within a second data structure through the use of steganography.

6. The method of claim 1, further comprising generating the plurality of secrets.

7. The method of claim 1, wherein the assigning, generating, securing and storing are performed by an authentication management application executed at a computer that is distinct from an authentication server.

8. The method of claim 1, further comprising associating each secret with a unique name of a node the secret is assigned to and saving the associated names in the data structure.

9. The method of claim 1, wherein the network comprises a storage area network.

10. The method of claim 9, wherein the storage area network is a network selected from the group consisting of a Fibre Channel network, an iSCSI network and an FCoE network.

11. The method of claim 1, further comprising:  
obtaining the assigned secrets from the data structure by an authentication server; and  
using the assigned secrets by the authentication server to perform authentication for the plurality of nodes.

12-15. (canceled)

16. A device comprising a processor and a memory, the memory comprising a plurality of instructions executable at the processor and configured to cause the processor to:

assign a plurality of secrets to a plurality of nodes of a network;

generate a data structure in each node of the plurality of nodes, the data structure comprising the assigned secrets;

secure the data structure in each node of the plurality of nodes using a password generated according to a key agreement protocol; and

store each generated password in a management database.

17. The device of claim 16, wherein the instructions are further configured to cause the processor to generate or otherwise establish the plurality of secrets.

18. The device of claim 16, wherein the instructions are part of an authentication management application, and the device is distinct from an authentication server.

19. The device of claim 16, wherein the instructions are further configured to cause the processor to associate each secret with a unique name of a node the secret is assigned to and save the associated names in the data structure.

20. The device of claim 16, wherein the network comprises a storage area network.

21. The device of claim 20, wherein the storage area network is a network selected from the group consisting of a Fibre Channel network, an iSCSI network and an FCoE network.

22-25. (canceled)

26. A network device comprising:  
a memory comprising a management database; and  
a hardware processor operable to:  
generate a password according to a key agreement protocol;  
store the password in the management database; and  
assign a plurality of secrets to a network node, the plurality of secrets comprising the password, one or more of the plurality of secrets being stored in a data structure in each node, the data structure being secured by the password.

27. The network device of claim 26, wherein the data structure is encrypted by with the password.

28. The network device of claim 26, wherein the data structure is embedded through the use of steganography.

\* \* \* \* \*