



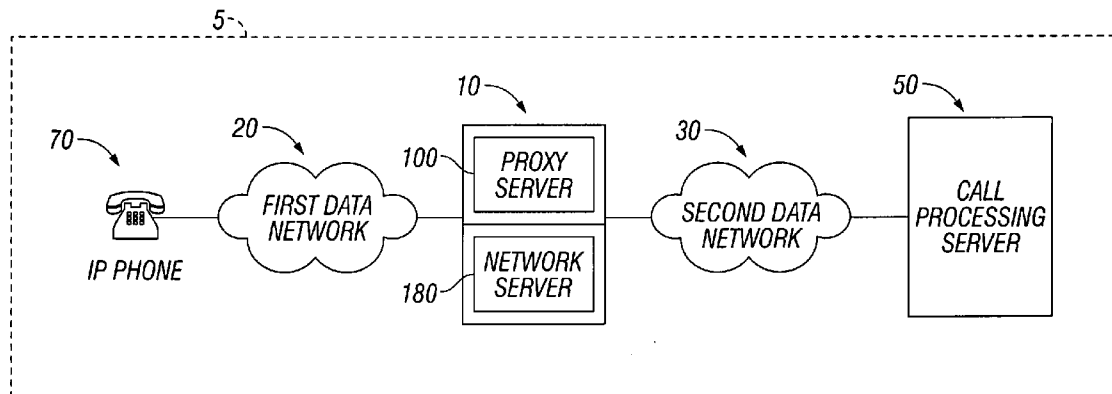
US 20060262916A1

(19) **United States**(12) **Patent Application Publication**
Marascio et al.(10) **Pub. No.: US 2006/0262916 A1**(43) **Pub. Date: Nov. 23, 2006**(54) **PROXY SERVER FOR INTERNET
TELEPHONY**(52) **U.S. Cl. 379/201.01; 370/352**(76) Inventors: **Louis R. Marascio**, Austin, TX (US);
Adam Chaney, Austin, TX (US); **Paul
Long**, Spicewood, TX (US); **James
Dixon**, Austin, TX (US); **Robert Scott
Comer**, Austin, TX (US)(57) **ABSTRACT**

Correspondence Address:

**J. SCOTT DENKO
ANDREWS & KURTH LLP
111 CONGRESS AVE., SUITE 1700
AUSTIN, TX 78701 (US)**(21) Appl. No.: **11/368,171**(22) Filed: **Mar. 3, 2006****Related U.S. Application Data**(63) Continuation-in-part of application No. 11/133,407,
filed on May 19, 2005.**Publication Classification**(51) **Int. Cl.**
H04M 3/42 (2006.01)
H04L 12/66 (2006.01)

An apparatus and a method are provided for proxying communications between devices on an IP telephony network. In a preferred embodiment, an appliance comprising a network server and proxy server software includes filtering rules which can be modified to enable proxied communication between an IP phone on a first data network, such as, for example, the internet, and a call processing server located on a second data network. In another embodiment, an appliance comprises a proxy server that proxies call-processing and signaling communications between an authenticated IP phone on a first data network and a call processing server located on a second data network. In another embodiment, the appliance relays voice traffic between IP phones. In another embodiment, the proxy server software operates in a secure mode to permit encryption and decryption of voice and control communications. In another embodiment, the proxy server enables proxied communications between a call processing server and an IP phone by modifying the provisioning data exchanged between the call processing server and the IP phone.



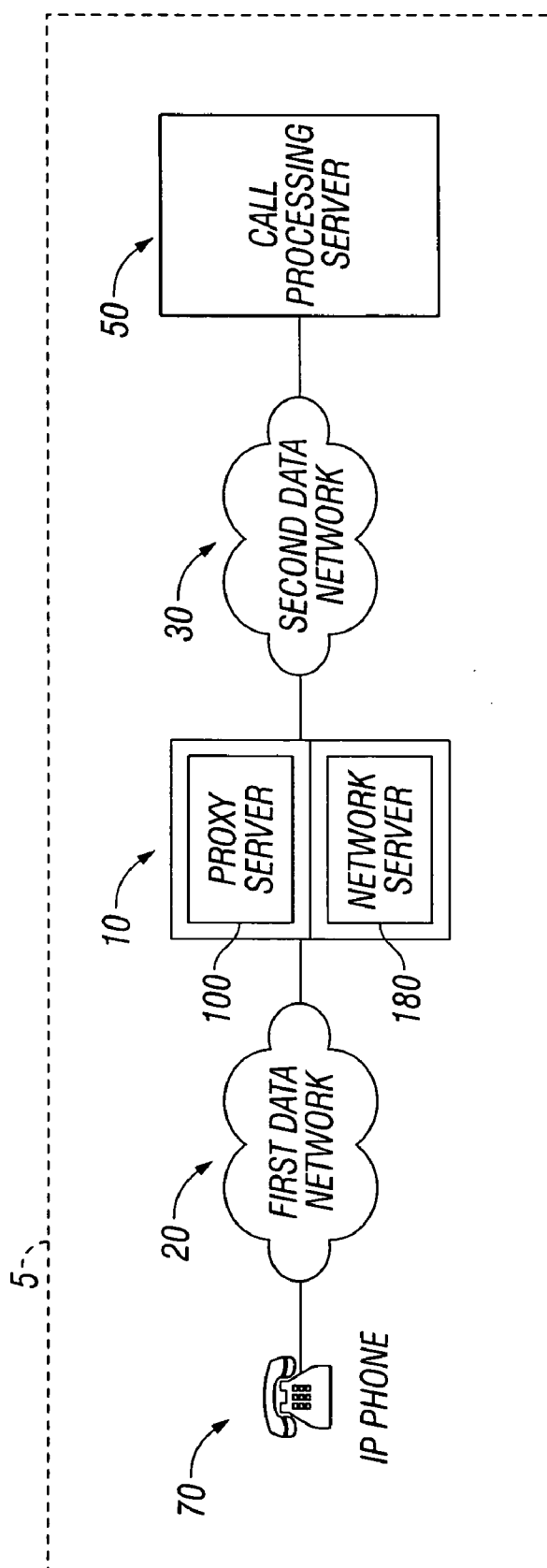


FIG. 1

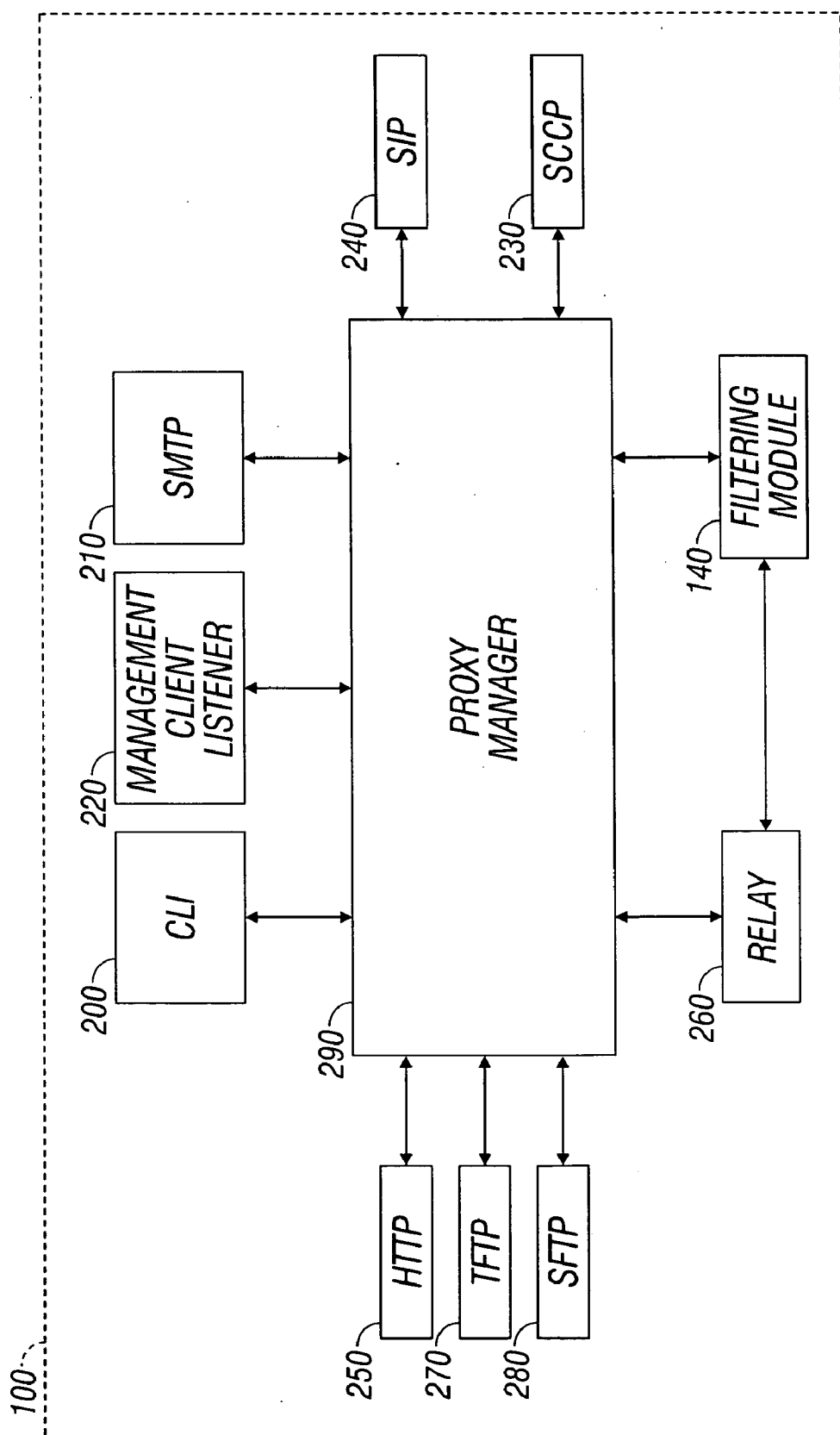


FIG. 2

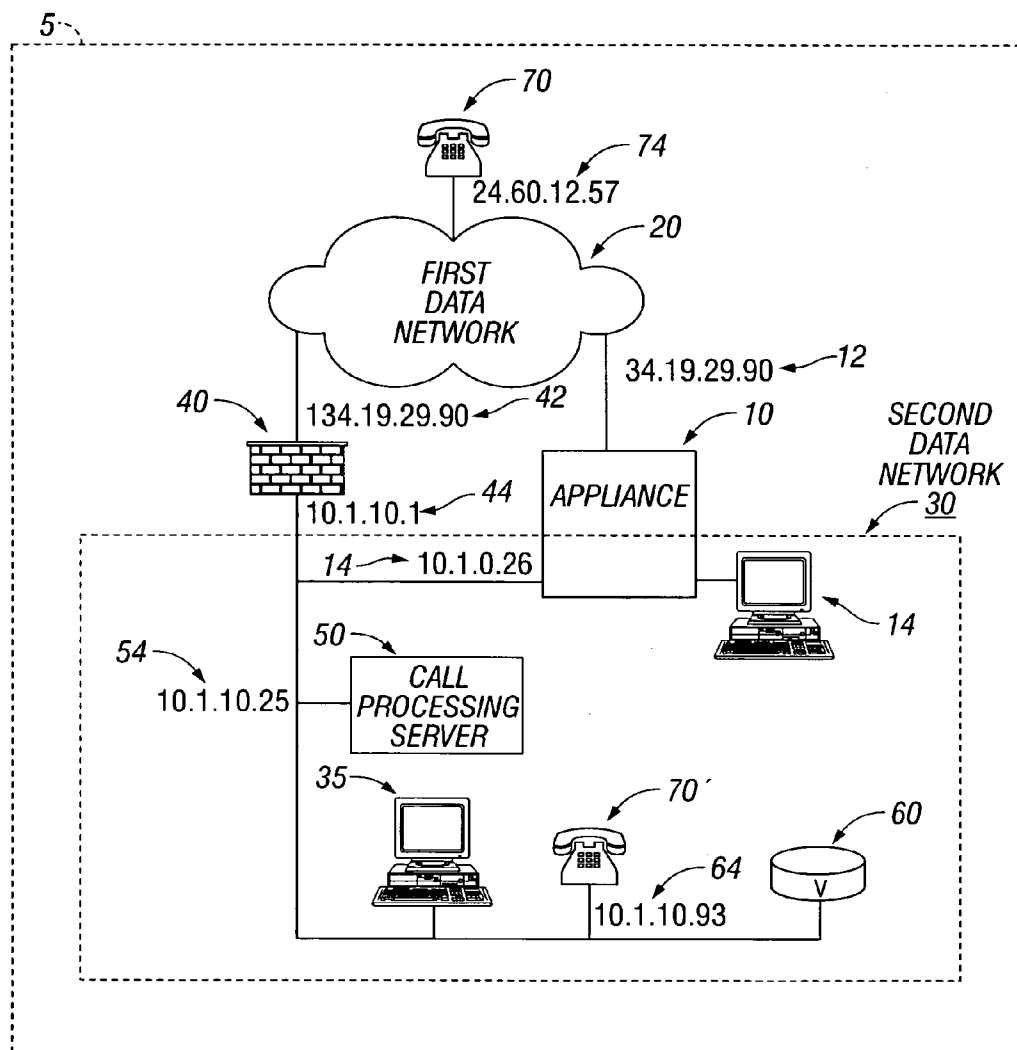


FIG. 3

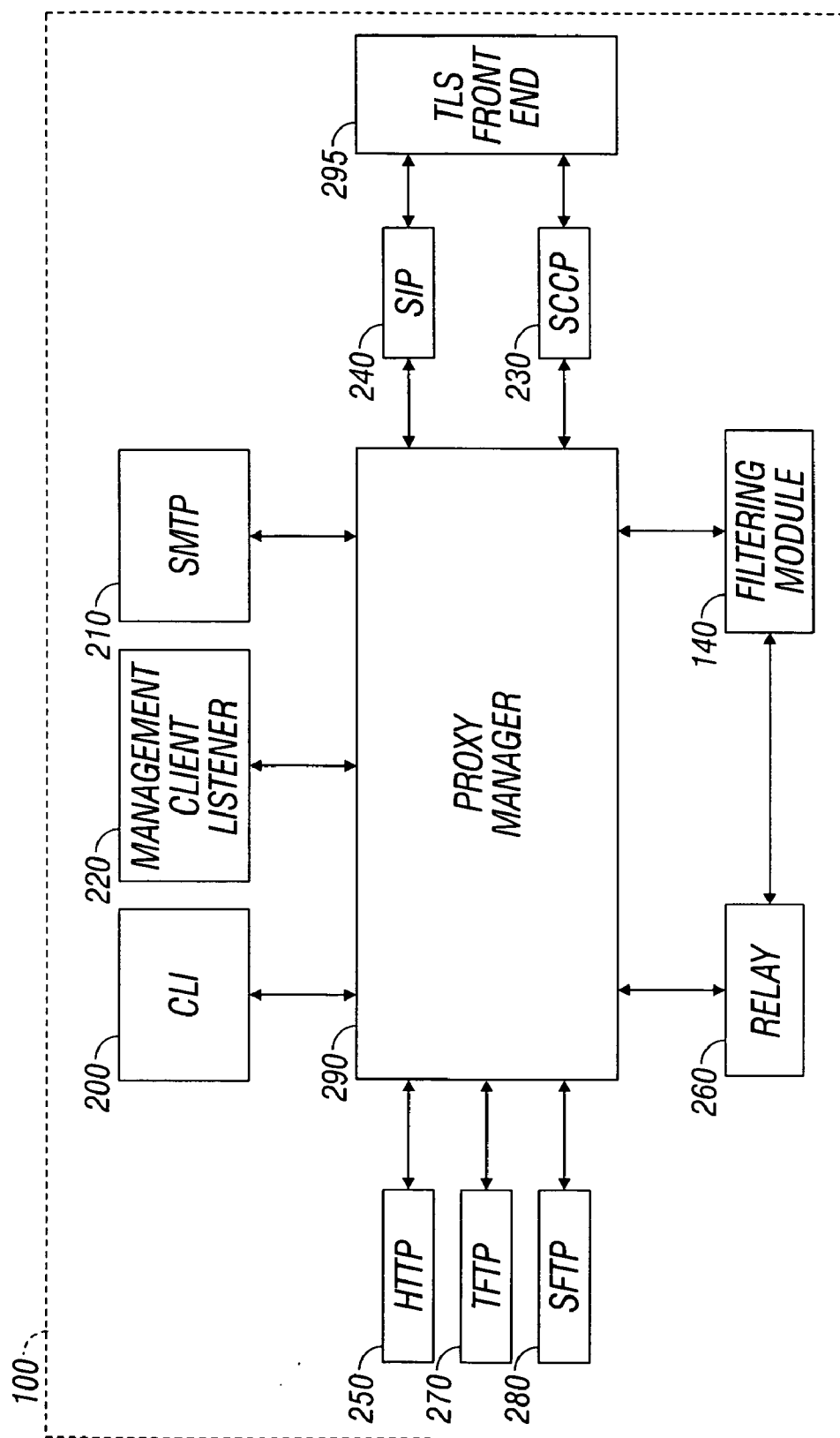


FIG. 4

PROXY SERVER FOR INTERNET TELEPHONY

RELATED APPLICATIONS

[0001] The present application is a continuation-in-part of U.S. patent application Ser. No. 11/133,407, filed May 19, 2005, pending. U.S. patent application Ser. No. 11/133,407 is incorporated by reference herein.

FIELD

[0002] The present invention relates to an improved apparatus and method for internet protocol (IP) telephony.

BACKGROUND

[0003] The gradual adoption of Voice over Internet Protocol (VoIP) technologies in recent years has helped reduce networking costs while bridging voice and data networks. A VoIP-enabled telephone (IP phone) can communicate with other IP phones over a data network also used for transmission of data by computers. In a common embodiment, administrative tasks associated with IP telephony such as provisioning the IP phones on the network and setting up and tearing down calls are handled by a call processing server (sometimes referred to as an IP PBX) that communicates with the IP phones over the same data network.

[0004] Although often the IP phones and associated call processing server are co-located on a secure private data network, it is sometimes the case that one or more IP phones are located on an untrusted data network such as the internet. This configuration requires connecting the call processing server directly or indirectly to the untrusted data network.

[0005] It is well-known that directly connecting a server to an untrusted data network such as the internet exposes the server to unauthorized intrusion and attacks from the untrusted data network, such as, for example, denial of service attacks, man-in-the-middle attacks, theft of service, and/or Trojan horses or malware. These security concerns are magnified for call processing servers in an IP telephony environment. Because the call processing server handles all call processing requests, any form of attack that can disable or disrupt a call processing server can jeopardize the entire IP telephony network. An unauthorized intruder, moreover, could use unauthorized access to the call processing server to eavesdrop on the phone calls of authorized users or to make unauthorized phone calls.

[0006] A firewall can shield servers and other network devices from intruders and attacks. A firewall is a system or combination of systems, implemented in hardware and/or software, that enforce a boundary between a trusted data network and an untrusted data network. The untrusted network could be a first data network such as the internet. It could also be part of a private data network (such as a LAN) where it is desired to protect sensitive components against internal attacks such as those from disgruntled employees. A network-level firewall, often called a packet filter, can filter traffic at the network protocol packet level, allowing or denying access to incoming packets based on the source address, destination address, and ports of the packet. An application-level firewall can route or filter incoming traffic at the application level, sometimes intelligently routing payload based on protocols such as HTTP. Many firewalls also rely on network address translation (NAT), wherein the

firewall converts internal network addresses on the trusted network, otherwise invisible externally, to public addresses directly accessible on the untrusted network, and vice versa, and routes communications accordingly.

[0007] A conventional firewall, however, does not optimally protect the call processing servers in a typical IP telephony environment. If the firewall is to act as a packet filter, the amount of access the firewall will be required to provide to the untrusted data network to permit proper functioning of the call processing server leaves the call processing server too exposed to malicious attacks; and filtering at the application level introduces a substantial, potentially excessive, amount of latency (delay) into the processing of packets containing call control information and/or voice traffic. What is needed, then, is an appliance that can shield call processing servers without the deficiencies of a conventional firewall.

[0008] A virtual private network (VPN) is a known solution to the security problems created by using untrusted data networks such as the internet for transmission of IP telephony communications. A VPN allows secure communication from one trusted network to another trusted network where part of the path traverses an untrusted network, and it can do so through a combination of authentication of user identification, encryption of call processing and/or voice payload, and tunneling, i.e. embedding the encrypted packets in an IP packet for transmission with TCP/IP. A VPN typically carries other types of data in addition to IP Telephony traffic, however, and the additional latency added by the VPN and the cost and complexity of licensing, configuration, support, and maintenance make a VPN less than optimal for IP telephony applications.

[0009] What is needed, then, is an appliance that will provide some or all of the security advantages of a VPN without the additional cost, complexity and overhead associated with a VPN.

SUMMARY

[0010] An apparatus and a method are provided for proxying communications between devices on an IP telephony network. In a preferred embodiment, an appliance comprising a network server and proxy server includes filtering rules which can be modified to enable proxied communication between an IP phone on a first data network, such as, for example, the internet, and a call processing server located on a second data network. In another embodiment, an appliance comprises proxy server that proxies call-processing and signaling communications between an authenticated IP phone on a first data network and a call processing server located on a second data network. In another embodiment, the appliance relays voice traffic between IP phones. In another embodiment, the proxy server software operates in a secure mode to permit encryption and decryption of voice and control communications. In another embodiment, the proxy server enables proxied communications between a call processing server and an IP phone by modifying the provisioning data exchanged between the call processing server and the IP phone.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 depicts a high level view of the components of an IP telephony environment according to a preferred embodiment of the present invention.

[0012] **FIG. 2** depicts a block diagram showing a module design of a proxy server component of an appliance according to a preferred embodiment of the present invention.

[0013] **FIG. 3** depicts a more detailed view of the components of an IP telephony environment according to a preferred embodiment of the present invention.

[0014] **FIG. 4** depicts a block diagram showing an alternative embodiment of a module design of a proxy server component of an appliance according to a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0015] **FIG. 1** depicts a high level view of the components of an IP telephony environment **5** according to a preferred embodiment of the present invention.

[0016] IP telephony environment **5**, in a preferred embodiment of the present invention, includes one or more devices such as, for example, IP phones **70** which communicate via a first data network **20** with an appliance **10** which is connected via a second data network **30** to one or more call processing servers **50**.

[0017] An appliance preferably could be a computer which is optimized for specific tasks such as, for example, a firewall to protect computers on the network from malicious attacks from the internet, or traffic management devices, such as routers. The task-related optimization may be accomplished through selection of hardware, special-purpose software, or a combination of hardware and software. An appliance may be fully capable of uses and applications other than the specific function for which it has been optimized, however.

[0018] In a preferred embodiment of the present invention, appliance **10** comprises a network server and a proxy server. A proxy server preferably is an intermediate system placed topologically between two classes of network entities through which entities of one class communicate with entities of the other using the same protocol. The classes may be different, as in a client class and a server class, or they may be the same, as in a class of peer-to-peer entities. The typical proxy server is transparent—it appears to members of a class as if it is a member of the other class. A proxy server can provide various services by, for example, selectively modifying and (less often) augmenting or removing information in data streams passing through it between entities of the two classes. A proxy server can provide additional services by simply relaying information without alteration. The proxy server preferably retains the essential semantics of the data streams.

[0019] Appliance **10** comprises a network server **180** which, in a preferred embodiment, will be familiar to those of skill in the art, after appreciation of the specification and drawings included herein. In a preferred embodiment, appliance **10** further comprises a proxy server **100** and software which, in a preferred embodiment, may be, for example, the Linux operating system, an Apache web page server, and database software such as postgresQL. A configuration of proxy server **100** is described in more detail below, in connection with **FIG. 2**.

[0020] An IP phone **70**, in a preferred embodiment, is a communications device that is TCP/IP aware, i.e., it is a

valid device on an internet protocol (IP) data network, and is capable of transmitting and receiving voice payload, i.e. digitized packets of voice or fax data, among other potential features.

[0021] IP telephony requires protocols for call control and signaling (control protocols). Control protocols are used to set up and tear down the connection between two or more devices (including IP phones) in a VoIP network. Control protocols in use in VoIP networks may include, for example, Session Initiation Protocol (SIP), a peer-to-peer protocol such as H.323, and client/server protocols such as Media Gateway Control Protocol (MGCP) and Skinny Client Control Protocol (SCCP, or “Skinny”).

[0022] Control protocols are to be distinguished from the voice protocols that are used to carry the voice payload (i.e., telephone and fax) in IP telephony. In a preferred embodiment of the invention, voice packets are transported across the IP network by using the Real-Time Transport Protocol (RTP) and RTP Control Protocol (RTCP) stack and by using the User Datagram Protocol (UDP) as a transport layer protocol.

[0023] In a preferred embodiment of the present invention, IP phone **70** is a Cisco IP phone adapted to use the SCCP control protocol, such as, for example, Cisco IP phone models 7905, 7912, 7940, 7960, 7936, 7970. In an alternative embodiment, IP phone **70** is a software-based application that runs on a personal computer. An example is the Cisco IP Communicator. Other instances of an IP phone **70** may include a traditional analog phone that is connected either to an IP telephony network via an analog telephone adapter (ATA) or to a voice gateway.

[0024] With continuing reference to **FIG. 1**, IP phone **70** is connected to a first data network **20**. First data network **20**, in a preferred embodiment of the present invention, is an untrusted public data network such as, for example, the internet. Appliance **10**, in a preferred embodiment, is interposed between first data network **20** and second data network **30** by, for example, being connected via an ethernet cable or other cable or physical connection or via a wireless connection. Appliance **10** preferably proxies communications between devices on first data network **20**, such as, for example, IP phone **70**, and devices on second data network **30**, such as, for example, call processing server **50**. Second data network **30**, in a preferred embodiment, preferably is a Gigabit Ethernet (GigE) local area network (LAN).

[0025] Still referring to **FIG. 1**, call processing server **50** preferably handles call-processing requests involving IP phone **70** in IP telephony environment **5**. Call processing server **50** is adapted to communicate over one or more data networks with an IP phone **70** according to one or more control protocols such as H.323, SIP, MGCP, and SCCP. Call processing server **50** preferably is adapted to store or obtain configuration or provisioning data for IP phone **70** and to supply IP phone **70** with configuration or provisioning data within IP telephony environment **5**.

[0026] IP telephony environment **5** can include one or more call processing servers **50**, and a preferred embodiment of the present invention comprises a cluster of up to **8** call processing servers **50** depending on the number of projected devices in the IP telephony network. Additional call processing servers **50** allow for upward scalability to handle additional devices, such as, for example, IP phones.

[0027] In a preferred embodiment of the present invention, call processing server **50** may be implemented as a Cisco Media Convergence Server with Cisco CallManager software, version **3.3** or above installed on it. Versions 3.3 and above of the Cisco CallManager software support the SCCP control protocol for processing call requests by an IP phone.

[0028] **FIG. 2** depicts a block diagram showing module design of proxy server module **100** of appliance **10** according to a preferred embodiment of the present invention.

[0029] Filtering module **140** includes filtering rules, i.e. rules that specify how packets of information received by the appliance from first data network **20** are to be processed, such as, for example, which packets of data are simply dropped, which packets are relayed or forwarded, and the destination address and port. The filtering rules, in a preferred embodiment, can be modified under software control. In a preferred embodiment of the present invention, filtering module **140** may be, for example, the IP Tables/Netfilter modules in Linux, which execute in kernel mode but can be modified by an application operating in user space or kernel space.

[0030] Proxy server module **100** in a preferred embodiment includes Proxy manager module **290**. In a preferred embodiment, Proxy manager module **290** controls and manages phone signaling proxy modules **230** (SCCP) and **240** (SIP), and using method calls and interprocess communication protocols relays messages and data with other modules, including relay module **260**, HTTP module **250**, SFTP module **280**, TFTP module **270**, command line interpreter (CLI) module **200**, SMTP module **210**, and Management Client listener module **220**. Proxy manager module **290** performs additional functions, as described below.

[0031] In a preferred embodiment of the present invention, phone signaling proxy modules **230** and **240** receive and process call control and signaling requests from authenticated IP phone **70**, such as, for example, requests for activation, registration, and call initiation and tear down. Phone signaling proxy module **230** handles control communications according to the SCCP control protocol; and phone signaling proxy module **240** handles control communications according to the SIP control protocol. In an alternative embodiment of the present invention, proxy server module might include other signaling proxy modules such as, for example, modules for processing MGCP and H.323 control protocols. The Proxy manager model also handles inter-appliance communication through a messaging format.

[0032] Relay module **260**, in a preferred embodiment, manages RTP relays of voice traffic between IP phones **70** by going into Linux kernel mode and modifying the Linux iptables routing rules so that the Linux kernel relays RTP packets between one IP phone **70** and another IP phone **70**. This packet-level approach eliminates RTP packet handling by the appliance at the application level, resulting in more efficient use of system resources and less delay of through traffic, but alternative embodiments are also within the scope of the invention.

[0033] IP phone **70** preferably is or can be adapted for communicating via "Trivial File Transfer Protocol" (TFTP). In a preferred embodiment, TFTP module **270** receives and processes requests from IP phone **70** for configuration files. The TFTP module makes a TFTP request for the specified

files from the call processing server **50** associated with IP phone **70**, adjusting the content as required to perform proxy function. The TFTP server preferably substitutes the appliance's address **12** where it appears in the configuration files returned to IP phone **70**, and also associates the call processing server's address with the device name of IP phone **70** to use later to proxy phone requests. The existing information about the phone is augmented with the assigned call processing server **50** and associated proxy address. In a cluster configuration, there are multiple assignments of proxy for a given device, and not all devices have the same assignment. There is preferably only one TFTP server assigned to any one IP phone **70**. Hence the TFTP server must notify the other appliance **10** in the cluster of the assigned call manager addresses for a given phone.

[0034] HTTP server module **250** receives authentication requests from a user wishing to activate IP phone **70**. In a preferred embodiment of the present invention, HTTP server module **250** receives authentication requests via a secure transmission. HTTP server module **250** preferably passes the authentication request to Proxy manager module **290**, which request includes the user name and password and the IP address of IP phone **70**. In a preferred embodiment, Proxy manager module **290** maintains an internal user list containing user specific data (such as, for example, a user name and a hash of the user's password), against which it compares the user name and password supplied by the user. The IP address is stored with the user-specific data upon successful authentication. In an alternative embodiment, the user contacts another authentication server, which then contacts proxy HTTP server, authenticates itself, and then authorizes the user by name and IP address. If successful, HTTP server module **250** is notified and it confirms authentication to the user. Proxy manager module **290** modifies the Linux iptables access rules so that filtering module **140** opens selected ports for access to other services (proxy, tftp, http) to requests from the address of IP phone **70**. HTTP Server module **250** also handles HTTP requests from IP phone **70** for service (initiated by the user pushing the service button on IP phone **70**), which it then proxies to the appropriate call processing server **50**, possibly updating the content in either direction.

[0035] In an alternative embodiment of the present invention, one or more appliances **10** can be clustered together for load management and fail-safe operation (not depicted). In a preferred embodiment with a cluster of appliances **10**, each appliance has a unique IP address, which is responsive to requests for call control. Additionally, each appliance is operative to receive HTTP requests and TFTP requests at a common IP address and port number, but only one appliance **10**, the active appliance, actually responds. When an appliance in the cluster learns a fact, such as, for example, user authentication data or a new modification to the filtering rules, it broadcasts it to all other appliances in the cluster. In a preferred cluster configuration, the active appliance preferably holds a token indicating that it is the active appliance. The other appliances preferably poll the active appliance regularly to confirm that the "active" appliance has not failed. If the appliance **10** with the "active" token has failed, the other appliances negotiate who will get the token and become the next "active" appliance **290**, and thus respond to HTTP and TFTP requests on the common IP address and port.

[0036] SMTP module 210 implements a Simple Mail Transfer Protocol. SMTP module preferably sends e-mail to system personnel regarding system status and/or emergency messages and, when enabled allows the user to update the local password database or recover from a forgotten password.

[0037] SFTP module 280 preferably is used to retrieve log files, and to deliver configuration files and updates.

[0038] Management Client Listener module 220 listens for connections from Management Client 35 and responds to requests for status, reports, and configuration changes. Management Client 35, depicted in FIG. 3, is a Java-enabled client with a full-featured GUI (graphic user interface) for status, reports, and configuration of the system.

[0039] CLI module 200 receives and performs simpler requests for status, reports, and configuration changes using a text command line from, preferably, a console 14 connected to appliance 10 via an SSH or direct serial connection. It is primarily used during initial installation, and also perhaps by remote personnel during installation debugging.

[0040] FIG. 3 depicts a more detailed view of the components of an IP telephony environment 5 according to a preferred embodiment of the present invention.

[0041] FIG. 3 depicts a preferred embodiment comprising one IP phone 70 connected to first data network 20 with an IP address 74 (24.60.12.57) publicly registered on first data network 20. One of ordinary skill in the art will readily appreciate, in light of the specification and drawings herein, that IP phone 70 could be connected directly to the first data network 20 or, in an alternative embodiment, could be connected indirectly via, for example, a user local area network (user LAN) in which IP phone 70 is shielded from first data network 20 by a NAT/firewall (not depicted). In such an alternative embodiment, the IP phone 70 might not necessarily have an IP address publicly registered on first data network 20 but the NAT/Firewall on the user LAN would have such a public address on first data network 20, and the NAT Firewall preferably would be responsible for NAT and routing VoIP traffic to IP phone 70 in a manner transparent to appliance 10 and call manager 50.

[0042] All IP addresses described herein, it will be understood, are solely for purposes of demonstrating the relationship between the various components of a preferred embodiment of the present invention, and may take different values in other embodiments.

[0043] FIG. 3 depicts one call processing server 50 connected to second data network 30 with unique IP address 54 (10.1.10.25). In a preferred embodiment of the present invention, up to eight (8) call processing servers can be clustered together, each with its own unique IP address (not depicted), on second data network 30.

[0044] In a preferred embodiment of the present invention, second data network 30 also includes a conventional NAT/Firewall 40 to shield the call processing servers 50 and other devices on the second data network 30 from intruders or other potentially harmful traffic coming from first data network 20. NAT/Firewall 40 has an external IP address 42 (134.19.29.89) registered on the first data network 20, and an internal IP address 44 (10.1.10.1) registered on the second data network 30.

[0045] A preferred embodiment of the present invention may include one or more instances of IP phone 70, depicted on FIG. 3 as 70', connected to second data network 30 at address 64 (10.1.10.93). In a preferred embodiment of the present invention, second data network 30 may also include other devices commonly found on LANs or IP telephony networks, such as, for example, routers or servers (not depicted), voice gateways to the packet-switched telephone network 60, and/or Java-enabled Management Client 35. Additional IP phones 70 may comprise analog phones connected to the voice gateway.

[0046] In another preferred embodiment of the present invention, NAT/firewall 40 is intended to shield call processing server from potentially harmful internal traffic such as, for example, from disgruntled employees on a corporate LAN. In this alternative embodiment, the first data network is not a public data network (such as the internet) but a private network.

[0047] Appliance 10 preferably displays an external IP address 12 to first data network 20, and appliance 10 displays an internal IP address 14 to second data network 30. For purposes of illustration, the external IP address 12 depicted in FIG. 3 for appliance 10 is 134.19.29.90, and the internal IP address 14 depicted in FIG. 3 for appliance 10 is 10.1.0.26. A preferred embodiment of the present invention comprise one or more appliances 10 interposed between second data network 30 and first data network 20, each of said appliance having a unique external IP address on first data network 20 (not depicted) and a unique internal IP address on second data network 30 (not depicted).

[0048] In a preferred embodiment, filtering module 140 may be configured to filter IP telephony traffic between the first data network to and from call processing server 50, as depicted in FIG. 3, thereby shielding call processing server 50 on second data network 30; and NAT/firewall 40 (as depicted in FIG. 3) is configured to filter other types of traffic for other applications on second data network 30.

[0049] Following are descriptions of exemplary processes involved in authenticating, activating, provisioning, and registering a phone according to a preferred embodiment of the present invention.

[0050] Initial state: In the preferred embodiment of the present invention the filtering rules in Filtering Module 140 are initially configured to not accept any request for access to Appliance 10 or call processing server 50 except for an HYMP request in secure mode (HTTPS) to TCP port 443 at the appliance's external IP address 12. Requests that are not accepted are preferably ignored.

[0051] Step 1 (Authentication): A user desiring access to the system to activate an IP phone connected to first data network 20, having been duly authorized by the system administrator, preferably may connect with a standard web browser to HTTPS using TCP port 443 and authenticate the device using an agreed-upon name and password. The authentication process will provide IP address 74 of IP phone 70 to Appliance 10.

[0052] Step 2 (Activation): If the authentication process is successful, HTTP module 250 preferably modifies the filtering rules in filtering module 140 to accept communications, data transmissions, messages, and/or service requests from IP address 74 of IP phone 70 to specific ports in

Appliance 10, such as, for example, (SCCP port, TCP 2000; HTTP port, TCP 80; TFTP port, UDP 69, plus additional ports, dynamically allocated. By setting a timer and then sending a subsequent message to close the ports, HTTP module 250 in a preferred embodiment of the present invention can hold these ports open for a period of time, typically 8-12 hrs, or perhaps until a fixed time such as midnight.

[0053] Step 3 (Provisioning): With respect to an embodiment that includes a provisioning function, the following steps are associated with a preferred embodiment of the present invention:

[0054] a. IP phone 70 preferably initiates a TFTP request from IP address 74 by sending a message to UDP port 69 and issuing requests for phone firmware and configuration files. IP phone 70 has a unique Station ID (SID) preferably associated with the unique name of the user who authenticated the IP address of the registration request. One request is for an XML configuration file named <SID>.cnf.xml.

[0055] b. The TFTP module 270 preferably makes a WIP request, using the internal IP address of Appliance 10, IP address 14 as the return address in the request packet of IP phone 70 and passes the request and SID number to call processing server 50 at internal IP address 54. Call processing server now believes that the IP address of IP phone 70 is the internal address of Appliance 10, IP address 14.

[0056] c. Call processing server 50 preferably returns phone and firmware provisioning data in the form of XML files, including one entitled <SID>.cnf.xml, to TFTP module 270 at IP address 14.

[0057] d. The XML file <SID>.cnf.xml preferably includes a variable named processNodeName value which call processing server 50 had initialized with its IP address 54. In an embodiment with more than one call processing server, call processing server 50 does not necessarily return its IP address 54, but might return the IP address of another call processing server 50' as the call processing server to be associated with IP phone 70 for purposes such as load allocation, redundancy, or geographic efficiency. For purposes of this explanation, call processing server 50 is the call processing server associated with IP phone 70.

[0058] e. TFTP module 270 preferably stores the SID of IP phone 70, IP address 74, and the initial values of processNodeName as the IP address of the call processing server associated with IP phone 70.

[0059] f. TFTP module 270 preferably changes processNodeName value in <SID>.cnf.xml to appliance 10's external IP address 12 for SCCP port TCP 2000, thereby making IP phone 70 believe that appliance 10 is the call processing server for IP phone 70.

[0060] g. The XML file <SID>.cnf.xml includes a variable named proxyserverURL which may be blank when it is first received from call processing server 50. TFTP module 270 preferably changes proxyserverURL in <SID>.cnf.xml to reference port 80 (i.e. the HTTP port) at IP address 12, thereby designating appliance 10 as the default HTTP server for IP phone 70.

[0061] h. TFTP module 270 preferably then returns the modified XML file via TFTP to IP phone 70 at IP address 74.

[0062] Step 4: (Registration): Now that IP phone 70's firmware is updated and the phone is configured, it preferably connects to Appliance 10, port TCP 2000 and sends an SCCP registration request using its SID and IP address. Proxy manager module 290 determines that call processing server 50 at IP address 54 is the call processing server associated with the SID and the return address of the phone, IP address 74, in the SCCP request. Proxy manager module 290 replaces the return address of the SCCP request with internal address 14 and forwards the SCCP request to the SCCP port at call processing server 50 at address 54. Whatever response the call processing server returns is forwarded to IP phone 70 at IP address 74.

[0063] Step 5: (Registered phone): Assuming successful registration, IP phone 70 preferably holds the connection open as long as it is able, sending and receiving various messages as it likes. Appliance 10 holds the corresponding connection to call processing server 50 open, too. Appliance 10 generally forwards all messages from IP phone 70 to call processing server 50, and all messages from call processing server 50 to IP phone 70. When Appliance 10 receives notification that the connection to IP phone 70 or call processing server 50 is broken, it breaks the other connection.

[0064] Following is a description of a preferred embodiment of a process of setting up an RTP relay to transmit voice traffic between IP phone 70 and, for explanation purposes, IP phone 70' at IP address 64. Preferably, IP phone 70' could be any instance of IP phone 70 with a SID and IP address distinct from IP phone 70:

[0065] a. If IP phone 70 makes or receives a call, certain messages exchanged between call processing server 50 and IP phone 70 via Appliance 10 must be monitored and/or updated. In the context of a preferred embodiment using SCCP, the messages may include OpenReceiveChannel, OpenReceiveChannelAck, CloseReceiveChannel, StartMediaTransmission, and StopMediaTransmission.

[0066] b. Call processing server 50 preferably sends OpenReceiveChannel to IP phone 70 at the address of appliance 10, IP address 14, indicating that IP phone 70' at IP address 64 is attempting to initiate a call.

[0067] c. Relay module 260 preferably allocates two RTP relay ports, one, facing first data network 20 at external IP address 12 and designated port 24970 for purposes of illustration (IP address 12: Port 24970); and the other, facing second data network 30 at internal IP address 14 and designated port 24972 for purposes of illustration (IP address 14: Port 24972).

[0068] d. IP phone 70 preferably returns OpenReceiveChannelAck to IP address 12, reporting that it will receive RTP packets at port 20000 at some IP address (for illustration only). Relay Module 260 intercepts the message, extracts the port information (i.e., port 20000) of IP phone 70's RTP receive channel from the message, substitutes IP address 14:Port 24972 in its place, and forwards the message on to Call processing server 50.

- [0069] e. Relay Module 260 preferably adjusts the filtering rules so that RTP packets that arrive at internal IP port 14:Port 24972 are relayed to Port 20000 at the public IP address of IP phone 70 where they appear to come from external port 12:port 24970.
- [0070] f. Call processing server 50 might send StartMediaTransmission to the IP phone 70 at internal address 14, reporting that IP phone 70 should send RTP packets to internal IP address 64: port 28298. Relay Module 260 preferably intercepts the message, extracts the address and port of the other phone's RTP receive channel (internal IP address 64: port 28298) from the message, substitutes external IP port 12: Port 24970 in its place, and forwards the message on to the phone at IP address 74.
- [0071] g. Relay Module 260 preferably modifies the filtering rules in filtering module 140 to forward RTP traffic arriving on external IP address 12:port 24970 to internal IP address 64: port 28298, where it will appear to come from internal IP address 12: 24972.
- [0072] h. Call processing server 50 sends StopMediaTransmission when the call is hung up or put on hold, which undoes the effect of step h. The message is forwarded to IP phone 70 unmodified.
- [0073] i. When the phone call is hung up or put on hold, Call processing server 50 preferably will send other messages to IP phone 70:
- [0074] ii. Call processing server 50 sends CloseReceiveChannel when the call is hung up or put on hold, which undoes the effect of steps b through h. The message is forwarded to the phone unmodified.
- [0075] In an alternative embodiment, proxy server 100 is adapted to operate in secure mode. FIG. 4 depicts a block diagram showing how the appliance would operate in secure mode according to a preferred embodiment of the present invention. TLS Front End 295 handles encryption and decryption of the call control and signaling traffic in secure mode.
- [0076] In a preferred embodiment, the proxy manager's secure mode is triggered by a TFTP request from IP phone 70 for a Certificate Trust List file. IP phone, in this embodiment, is adapted to operate in a secure environment by including, for example, software to implement encryption of payload data. Proxy Manager 100 generates a Certificate Trust list file according to a preordained pattern, and inserts a "signature" in the appropriate field with a private key, which it obtains in one embodiment from an e-token, and returns the file to IP phone 70. The private key may be, preferably, adapted for use in a private key encryption algorithm with, preferably, at least 128 bit encryption, such as AES 128.
- [0077] When IP phone 70 sends a provisioning request to Appliance 10, Appliance 10 proxies the request to the appropriate Call Processing Server 50. the provisioning data, in this embodiment, may not contain a signature. The Proxy Manager 100 substitutes into a field in <SID>cnf.srml a "signature" based on a private key and returns it to IP phone 70. IP phone 70 uses the public key to verify the signature. When IP phone 70 registers, it registers to the TLS port 2443 instead of the TCP port 2000.
- [0078] Regarding SCCP signaling and authentication, SCCP uses transport layer security (TLS) (RFC 2246) for signaling security. Therefore, in secure mode, proxy manager 100 preferably uses the TLS protocol to establish an authenticated and private connection between itself and IP phone 70.
- [0079] To set up a secure call in a preferred embodiment of the present invention, call processing server 50 sends an OpenReceiveChannel (ORC) message to Appliance 10. Proxy Manager 100 inserts a first AES-128 key for IP phone 70 into the ORC Message to use for decryption and passes the ORC request to IP phone 70. When call processing server 50 sends a StartMediaTransmission message (SMT), Proxy manager 100 inserts a second AES 128 key, (which could be identical to the first AES 128 key), which IP phone 70 will use for encryption into the SMT message and proxies it to IP phone 70. The relay settings are modified so that RTP packets from IP phone 70 are decrypted using the second AES 128 key before being relayed on to the other phone; and the RTP packets from the other phone towards IP phone 70 are encrypted using the first AES 128 key prior to relaying them to IP phone 70.
- [0080] Regarding media authentication and encryption, SCCP uses secure real time transport protocol (SRTP)(RFC 3711) for media security. Therefore, in secure mode, Relay Module 260 preferably uses the SRTP protocol in conjunction with a private key encryption algorithm such as, for example, AES 128, to provide packet authentication and privacy of the stream.
- [0081] Those of skill will recognize that the techniques of the present invention may be implemented to advantage in a variety of sequential orders and that the present invention may be generally implemented in magnetic media such as disks, tapes and CD-ROMs or other storage media for introduction into an IP telephony network. In such cases, instructions for executing the steps of the invention in an IP telephony network will be embedded in the media.
- [0082] Although many embodiments of the present invention have been described in detail, it will be apparent to those skilled in the art that many embodiments taking a variety of specific forms and reflecting changes, substitutions and alterations can be made without departing from the spirit and scope of the invention. The described embodiments illustrate the scope of the claims but do not restrict the scope of the claims.
1. An appliance for shielding a call processing server in an IP telephony network, comprising:
 - a. a network server interposed between a first data network and a second data network, said second data network comprising a call processing server;
 - b. a proxy server comprising filtering rules operative to restrict communication from said first data network to said call processing server; and
 - c. said proxy server operative to modify said filtering rules to permit proxied communications to said call processing server from a first device on said first data network, said first device being authenticated.
 2. The appliance of claim 1, wherein said first device is an IP phone.

3. The appliance of claim 2, wherein said first device is authenticated by said proxy server.

4. The appliance of claims 1 or 2 wherein said proxy server is operative to relay communications between said first device and a second authenticated device.

5. An appliance for enabling secure IP telephony, comprising:

- a. a network server interposed between a first data network and a second data network, said second data network comprising a call processing server;
- b. a proxy server comprising filtering rules operative to restrict communication from said first data network to said call processing server;
- c. said proxy server operative to modify said filtering rules to permit proxied communications between said call processing server and a first device on said first data network, said first device being authenticated; and
- d. said proxy server operative to exchange communications with said first device in a secure mode.

6. The appliance of claim 5, wherein said secure mode uses secure real time transport protocol and a private key encryption algorithm.

7. The appliance of claim 5, wherein said secure mode uses transport layer security.

8. An appliance for shielding a call processing server in an IP telephony network, comprising:

- a. a network server interposed between a first data network, said first data network comprising an IP phone, and a second data network, said second data network comprising a call processing server; and
- b. a proxy server operative to proxy the provisioning of said IP phone by said call processing server by modifying the provisioning data exchanged between said IP phone and said call processing server.

9. A method for shielding a call processing server in an IP telephony network comprising a network server interposed between a first data network and a second data network, said first data network comprising a first IP phone, said first IP phone being unauthenticated, said second data network comprising a call processing server, and said network server comprising filtering rules restricting communications between said first data network and said second data network, comprising the steps of:

- a. authenticating said first IP phone; and
- b. modifying said filtering rules to enable proxied communications between said authenticated first IP phone and said call processing server.

10. The method of claim 9 further comprising the step of relaying voice payload data between said authenticated first IP phone and a second IP phone.

11. A method for shielding a call processing server in an IP telephony network comprising a network server interposed between a first data network and a second data network, said first data network comprising a first IP phone, said first IP phone being unprovisioned, said second data network comprising a call processing server, comprising the steps of:

- a. proxying a request to said call processing server to provision said first IP phone;

- b. modifying the provisioning data exchanged between said IP phone and said call processing server; and

- c. proxying communications between said IP phone and said call processing server.

12. A method for enabling secure communications over an IP telephony network comprising a network server interposed between a first data network and a second data network, said first data network comprising a first IP phone, said second data network comprising a call processing server, comprising the steps of:

- a. proxying provisioning of said IP phone by said call processing server;
- b. supplying a private key to said IP phone with provisioning data from said call processing server; and
- c. communicating with said IP phone using a private key encryption algorithm.

13. An IP telephony system, comprising:

- a. a network server interposed between a first data network and a second data network,
- b. an IP phone connected to said first data network, said IP phone being unauthenticated,
- c. a call processing server connected to said second data network;
- d. a proxy server comprising filtering rules restricting transmission of data from said first data network to said second data network, said proxy server operative to authenticate said first IP phone, and said proxy server operative to modify said filtering rules to enable proxied communication between said authenticated first IP phone and said call processing server.

14. A storage medium embedded with instructions that operate in an IP telephony environment, said environment comprising a network server interposed between a first data network and a second data network, said first data network comprising a first IP phone, said IP phone being unauthenticated, said second data network comprising a call processing server, and said network server comprising filtering rules restricting communications between said first data network and said second data network, to:

- a. authenticate said first IP phone; and
- b. modify said filtering rules to enable proxied transmission of data from said authenticated first IP phone to said call processing server.

15. An appliance for shielding a call processing server in an IP telephony network, said IP telephony network comprising a network server interposed between a first data network and a second data network, said first data network comprising a first IP phone, said first IP phone being unauthenticated, said second data network comprising a call processing server, comprising:

- a. means for restricting communications between said first data network and said second data network according to filtering rules;
- b. means for authenticating said first IP phone; and

c. means for modifying said filtering rules to enable proxied communications between said authenticated first IP phone and said call processing server.

16. The appliance of claim 15, further comprising means for relaying communications between said authenticated first IP phone and a second IP phone.

17. The appliance of claim 15, further comprising means for communicating with said authenticated first IP phone in a secure mode.

* * * * *