SYSTEM AND METHOD FOR ENABLING SECURE ACCESS TO SERVICES IN A COMPUTER NETWORK

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks, and more particularly to a system and method for enabling secure access to services in a computer network.2.

Description of the Background Art

In its infancy, the Internet provided a research-oriented environment where users and hosts were interested in a free and open exchange of information, and where users and hosts mutually trusted one another. However, the Internet has grown dramatically, currently interconnecting about 100,000 computer networks and several million users. Because of its size and openness, the Internet has become a target of data theft, data alteration and other mischief.

Virtually everyone on the Internet is vulnerable. Before connecting, companies balance the rewards of an Internet connection against risks of a security breach. Current security techniques help provide client and server authentication, data confidentiality, system integrity and system access control.

The most popular of the current security techniques is a firewall, which includes an intermediate system positioned between a trusted network and the Internet. The firewall represents an outer perimeter of security for preventing unauthorized communication between the trusted network and the Internet. A firewall may include screening routers, proxy servers and application-layer gateways.

For users on the internet to gain access to protected services on the trusted network, they may be required to provide their identity to the firewall by some means such as entering a password or by computing a response to a challenge using a hardware token. With proper authentication, the user is allowed to pass through the firewall into the local network, but is typically limited to a predetermined set of services such as e-mail, FTP, etc.

Some local network managers place just outside the firewall a server, often referred to as a "sacrificial lamb" for storing non-confidential data which is easily accessible by the remote user but providing little security.

A De-Militarized Zone, or DMZ, sits between two firewalls protecting a trusted network. The external firewall protects servers in the DMZ from external threats while allowing HyperText Transfer Protocol (HTTP) requests. The internal firewall protects the trusted network in the event that one of the servers in the DMZ is compromised. Many companies use DMZs to maintain their web servers.

Another security technique for protecting computer networks is the issuance and use of a public key certificates. Public key certificates are issued to a party by a certificate authority, which via some method validates the party's identity and issues a certificate stating the party's name and public key. As evidence of authenticity, the certificate authority digitally signs the party's certificate using the certificate authority's private key.

Thus, when a user via a client computer connects to a server, the client computer and server exchange public key certificates. Each party verifies the authenticity of the received certificates by using the certificate authority's public key to verify the signature of the certificate. Then, by encrypting messages with the server's public key the user can send secure communications to the server, and by encrypting messages with the user's public key the server can send secure communications to the user. Although any party might present a public key certificate, only the real user and the real host have the corresponding private key needed to decrypt the message. Examples of authentication and key distribution computer security systems include the Kerberos™ security system developed by the Massachusetts Institute of Technology and the NetSP™ security system developed by the IBM Corporation.

These security techniques do not solve problems associated with the roaming (traveling) user. For the roaming user, maintaining identification and authentication information such as passwords, certificates, keys, etc. is a cumbersome process. Further, accessing multiple systems requires multiple keys, which often are too complex to track and use. Also, direct access to systems behind firewalls compromises security. Therefore, a

system and method are needed to enable remote access to computer services easily and securely.

SUMMARY OF THE INVENTION

The present invention provides a system and method for enabling secure access to services in a computer network. The network system includes a global server coupled via a computer network to computer services. The global server includes a communications engine for establishing a communications link with a client; security means coupled to the communications engine for determining client privileges; a servlet host engine coupled to the security means for providing to the client, based on the client privileges, an applet which enables I/O with a secured service; and a keysafe for storing keys which enable access to the secured services. The global server may be coupled to multiple sites, wherein each site provides multiple services. Each site may be protected by a firewall. Accordingly, the global server stores the keys for enabling communication via the firewalls with the services.

The method includes the steps of establishing a communications link with a client; identifying and authenticating the client; determining client privileges; providing to the client, based on the client privileges, an applet which enables I/O with a secured service; and retrieving a key which enables access to the secured service.

The system and method of the present invention advantageously provide a globally-accessible trusted third party, i.e., the global server. This trusted third party securely stores keys, and acts as a single identification and authentication service. Other systems may be accessed through the global server. The global server uses the stored keys to authenticate the user under an identity that is understood by the other system's existing security services, and establishes a secure communications channel to the desired service. Because of a global firewall, the global server is substantially protected from external threats. Accordingly, the global server provides authorized clients with secure communication through firewalls with services. The global server may enable multiple levels of identification and authentication services. Accordingly, the global server may enable multiple levels of resource access based

on the user's status, the strengths of the identification and the authentication and on the privacy of the communications channel.

Because of the global firewall and the identification and authentication services performed by the global server, corporations can store relatively secret information on the global server for use by authorized clients. Yet, the present invention also enables corporations to maintain only a portion of their secret information on the global server, so that there would be only this limited loss should the trusted third party system be compromised. Further, the global server advantageously may act as a client proxy for controlling access to services, logging use of keys and logging access of resources.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a roaming-user network access system, in accordance with the present invention;

FIG. 2 is a block diagram illustrating details of an example client of FIG. 1;

FIG. 3 is a block diagram illustrating details of the global server of FIG. 1;

FIG. 4 is a block diagram illustrating details of an example service server of FIG. 1;

FIG. 5 is a flowchart illustrating a method for remotely accessing a secure service;

FIG. 6 is a flowchart illustrating details of the FIG. 5 step of creating a link between a client and the global server of;

FIG. 7 illustrates an example web page;

FIG. 8A is a flowchart illustrating details of the FIG. 5 step of accessing a service in a first embodiment;

FIG. 8b is a flowchart illustrating details of the FIG. 5 step of accessing a service in a second embodiment; and

FIG. 8C is a flowchart illustrating details of the FIG. 5 step of accessing a service in a third embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram illustrating an exemplary roaming-user network access system 100 in accordance with the present invention. System 100 includes an interconnected network of computers referred to herein as an "Internet" 102. System 100 further includes a first company network 112, a second company network 118, a kiosk network 138 and an Internet Service Provider (ISP) network 143, each network being coupled to the Internet 102.

Company network 112 includes a firewall 116 coupled between the Internet 102 and a client computer 114a. Company network 118 includes a firewall 120 coupled between the Internet 102 and an internal network signal bus 126. Company network 118 further includes a first server 108a for providing a first service 110a, a second server 108b for providing a second service 110b, a first client computer 114b storing a program for providing a third service 110c and a second client computer 114c, each being coupled to signal bus 126. Example services 110a-110d include an e-mail service program, an address book service program, a calendar service program, a paging service program, and a company database service program.

The kiosk network 138 includes a first client computer 114d and a second client computer 114e, each being coupled to the Internet 102. The ISP network 143 includes an ISP 148 coupled via a wireless channel 146 to a first client computer 114f and coupled via modems 152 and 156 and via transmission line 154 to a second client computer 114g.

The Internet 102 includes a global server 106 which is protected by a global firewall 104 and includes a server 108c for providing a service 110d. Intercommunication between client computers 114a-114g and services 110a-110d is accomplished via the global server 106. If, for example, a user of any one of the client computers 114a-114g wants to access a service 110a-110d (which is provided at a location within system 100 that is unknown to the user), then the user applies a known Uniform Resource Locator (URL) to access a web page operated by global server 106. An example web page 700 is shown in and described with reference to FIG. 7. The global firewall 104 protects the global server 106 from external threats.

Before obtaining access privileges to the functionality provided by the global server 106, the user must first obtain authorization from the global server 106. Obtaining authorization typically requires user identification and authentication, for example, using public-key certificates. Once authenticated, the global server 106 provides the user with access to the services 110a-110d. It will be appreciated that varying levels of access to services 110a-110d will be granted based on varying strengths of identification and authentication and on the privacy of the communications channel.

To enable user access to and control of the services 110a-110d, the global server 106 may use conventional applets, servlets or agents in a distributed network environment, such as the Java™ distributed environment produced by the Netscape Corporation. The global server 106 provides the user's client with access to and control of the service 110a-110d. The global server 106 may redirect the user's client to access the service 110a-110d itself, the global server 106 may access the service 110a-110d itself and provide I/O to the client by proxy, or the global server 106 may provide the service 110a-110d itself. These three different modes of access to the services 110a-110d are described with reference to FIGs. 8A-8C.

The global server 106 maintains the network addresses of all the services 110a-110d, the user's public and private keys, the user's account numbers, firewall authentication information, etc. Firewall authentication information includes the necessary identification, passwords and certificates needed to pass firewalls 116 and 120. Accordingly, the user need only maintain the URL of the global server 106, and identification and authentication information such as a password or hardware token for obtaining access to the functionality of the global server 106. Thus, the roaming user can access computer services 110a-110d using any computer terminal which is connected to the Internet 102.

FIG. 2 is a block diagram illustrating details of a client computer 114, such that each of clients 114a-114d is an instance of the client 114. The client 114 includes a Central Processing Unit (CPU) 210 such as a Motorola Power PC® microprocessor or an Intel

Pentium® microprocessor. An input device 220 such as a keyboard and mouse, and an output device 230 such as a Cathode Ray Tube (CRT) display are coupled via a signal bus 240 to CPU 210. A communications interface 250, a data storage device 260 such as Read Only Memory (ROM) or a magnetic disk, and a Random-Access Memory (RAM) 270 are further coupled via signal bus 240 to CPU 210. The communications interface 250 of client computer 114 is coupled to the Internet 102 as shown in and described with reference to FIG. 1.

An operating system 280 includes a program for controlling processing by CPU 210, and is typically stored in data storage device 260 and loaded into RAM 270 for execution. Operating system 280 includes a communication engine 282 for generating and transferring message packets to and from the internet 106 via the communications interface 250.

Operating system 280 further includes an internet engine such as a web browser 284, e.g., the Netscape™ web browser produced by the Netscape Corporation or the Internet Explorer™ web browser produced by the Microsoft Corporation. The web browser 284 includes an encryption engine 285 for encrypting messages using public and private keys, and an applet engine 286 for executing applets 288 downloaded from the global server 106 to enable the access to computer services 110a-110d. Downloaded applets 288 may include security applets 290 for performing services such as user identification and authentication, message integrity services, and certificate verification. The browser 284 further receives web page data (391, FIG. 3), configuration data 390 and information identifying a set of selectable services 110a-110d, and uses the information to display the web page (700, FIG. 7). The web browser 284 enables a user via the client 114a-114g to select one of the services 110a-110d for execution.

It will be appreciated that a client 114a-114g such as client 114b may include a service engine 490 (see FIG. 4) for providing a service 110a-110d such as service 110c. Thus, it is possible for a client 114b user to request access to service 110c via the global server 106, without knowing that the service 110c is provided by client 114b. Accordingly,

the global server 106 will provide client 114 with an applet 288 for providing user interface I/O of service 110c back to client 114b.

FIG. 3 is a block diagram illustrating details of the global server 106, which includes a CPU 310 such as a Motorola Power PC® microprocessor or an Intel Pentium® microprocessor. An input device 320 such as a keyboard and mouse, and an output device 330 such as a CRT display are coupled via a signal bus 340 to CPU 310. A communications interface 350, a data storage device 360 such as ROM or a magnetic disk, and a RAM 370 are further coupled via signal bus 340 to CPU 310. The communications interface 350 is conventionally coupled as part of the Internet 102 to the clients 114. Although the global server 106 is described as a single computer, it will be appreciated that the global server 106 may include multiple computers networked together.

Operating system 380 includes a program for controlling processing by CPU 310, and is typically stored in data storage device 260 and loaded into RAM 370 for execution. Operating system 380 includes a communication engine 382 for generating and transferring message packets to and from client computers 114 via the communications interface 350.

Operating system 380 further includes, as part of global firewall 104, security services 384 for opening a communications channel with users. For example, when a client attempts to access the global server 106, the security services 384 first determines whether the global server 106 accepts in-bound communications from a particular port (not shown) and whether the servlet host engine 386, described below, is authorized to connect to that particular port. If so, the security services 384 allows the communications engine 382 to open a communications channel via the particular port to the client 114a-114g. Otherwise, no channel will be opened.

The operating system 380 further includes a web engine 387 which, based on user's identification, the strength of the user's authentication and the privacy of the communications channel, forwards web page data 391 and information identifying a set of available services 110a-110d to the client 114a-114g. An example web page 700 is shown and described with

reference to FIG. 7. The web engine 387 enables a user to select a service 110a-110d from the web page 700.

The web engine 387 includes a servlet host engine 286, which downloads security applets 290 including an authentication applet (not shown) to the client computer 114 and accordingly executes an authentication servlet 397 of servlets 398 for performing identification and authentication services. The authentication applet 290 prompts the user for identification and authentication information, and then communicates the information to the authentication servlet 397. The authentication servlet 397 verifies that the information is correct. It will be noted that the user's authentication information is not necessarily sent to the authentication servlet 397, but rather its existence and correctness is proven via a secure means such as a secure hash. The servlet host engine 386 further includes a secure communications engine 396 which may use public key certificates to negotiate a secure communications channel with the client computer 114.

Upon selection of a service 110a-110d, the servlet host engine 386 downloads a corresponding applet 388, corresponding configuration data 390 and corresponding user data 392 and may download corresponding service address information 394 to the client computer 114. Configuration data 390 includes information for configuring the user's web browser 284, for configuring the downloaded applets 288, and for configuring the selected service 110a-110d. User data 392 may include user-and-service-specific information such as stored bookmarks, calendar data, pager numbers, etc. which was specifically stored on the global server 106 for easy access. Service address information 394 identifies the location of the services 110a-110d provided in system 100 by the global server 106. The client computer 114 executes the corresponding downloaded applet 288, which via the servlet host engine 386 (possibly using a corresponding servlet 398) enables the user to access and to control the corresponding services 110a-110d. The downloadable applets 388, configuration data 390, user data 392 and service address information 394 may be stored on the data storage device 360.

A keysafe 395 is a data file for storing each user's identification information, each user's public and private keys, each firewall's password information, etc. The keysafe 395 is organized in a linked list format so that, based on the selected service 110a-110d, the global server 106 can retrieve the appropriate firewall's password information, the appropriate user's identification information and keys, etc. The keysafe 395 may be stored on the data storage device 360.

FIG. 4 is a block diagram illustrating details of a service server 108, such that servers 108a-108c and client 114b are instances of server 108. Server 108 includes a CPU 410 such as a Motorola Power PC® microprocessor or an Intel Pentium® microprocessor. An input device 420 such as a keyboard and mouse, and an output device 430 such as a CRT display are coupled via a signal bus 440 to CPU 410. A communications interface 450, a data storage device 460 such as ROM or a magnetic disk, and a RAM 470 are further coupled via signal bus 440 to CPU 410. The communications interface 450 is coupled to the clients 114 as shown in and described with reference to FIG. 1.

The operating system 480 includes a program for controlling processing by CPU 410, and is typically stored in data storage device 460 and loaded into RAM 470 for execution. Operating system 480 also includes a communications engine 482 for generating and transferring message packets via the communications interface 450 to and from clients 114 or to and from global server 106. Operating system 480 further includes security services 484 for negotiating a secure channel with users, a secure communications engine 486 for opening the secure channel with the users, and a service engine 490 for providing a service 110a-110d to the users.

The service engine 490 includes a service interface 492 for receiving and translating messages to and from downloaded applets 288 currently executing on the client 114, and includes a service processor 494 and service data 496 for processing the service requests from the user. The service data 496 may include previously-generated documents, database information, etc. It will be appreciated that the service data 496 is similar to the user data

392, such that it includes the same type of information but is maintained on the service server 108 instead of on the global server 108.

FIG. 5 is a flowchart illustrating a method 500 enabling a user to access services 110a-110d in computer network system 100. Method 500 begins by the client 114 in step 505 creating a communications link with the global server 106. Step 505 is described in greater detail with reference to FIG. 6. The global server 106 in step 510 confirms that the user has privileges to access the functionality of the global server 106. Confirming user access privileges may include examining a user certificate, obtaining a secret password, using digital signature technology, etc. It will be appreciated that the security services 384 may cause the servlet host engine 386 to forward a security applet 389 via the communications channel to the client 114 for performing user authentication.

After user access privileges are confirmed, the web page engine 387 of the global server 106 in step 515 downloads web page data 391 and configuration data 390 to the client 114. The browser 284 of the client 114 in step 520 uses the web page data 391 and the configuration data 390 to display a web page 700 (FIG. 7) on the output device 230 of the client 114 and to enable access to the services 110a-110d which are offered by the global server 106. An example web page 700 is shown and described with reference to FIG. 7.

From the options listed on the web page 700, the user in step 525 via input device 220 selects a service 110a-110d. In response, the servlet host engine 386 of the global server 106 in step 530 downloads the corresponding applet(s) 388, applet configuration data 390, user data 392 and possibly service address information 394 to the client 114. Applet configuration data 390 preferably includes user-specific preferences, such as user-preferred fonts, for configuring the selected service 110a-110d. User data 392 may include user-specific and service-specific information such as stored bookmarks, calendar data, pager numbers, etc. Service address information 394 identifies the location of the selected service 110a-110d. Alternatively, the corresponding applet(s) 388, applet configuration data 390, user data 392

and service address information 394 could have been downloaded in step 515 with the web page data 391 and the configuration data 390.

The applet engine 286 of the client 114 in step 535 executes the corresponding downloaded applet 288. The service server 108 in step 537 initiates the service engine 490. The global server 106 in step 538 selects one of the three modes of access described in FIGs. 8A-8C for enabling the client computer 114 to communicate with the corresponding service engine 490. For example, if the user selects the service 110d on server 108c, which is not protected by a separate firewall, then the global server 106 may provide the user with direct access. If the user selects service 110a provided by server 108a within company network 118, then the global server 106 may access the service 110a as a proxy for the user. It will be appreciated that each firewall 106 and 120 may store policies establishing the proper mode of access the global server 106 should select. Other factors for selecting mode of access may include user preference, availability and feasibility. The global server 106 in step 540 provides the client 114 user with access to the selected service 110a-110d. Step 540 is described in greater detail with reference to FIGs. 8A, 8B and 8C.

FIG. 6 is a flowchart illustrating details of step 505, which begins by the client 114 user in step 605 using a known Uniform Resource Locator (URL) to call the global server 106. The global server 106 and the client 114 in step 607 create a secure communications channel therebetween, possibly by applying Secure Sockets Layer (SSL) technology. That is, the security services 384 of the global server 106 in step 610 determine if in-bound secure communications are permitted and, if so, creates a communications channel with the client 114. The browser 284 of the client 114 and the security services 384 of the global server 106 in step 615 negotiate secure communications channel parameters, possibly using public key certificates. An example secure communications channel is RSA with RC4 encryption. It will be appreciated that the global server 106 may be configured to use one of ten encryption protocols and the client 114 may be enabled to use one of five encryption protocols. Step 615 thus may include selecting one of the encryption protocols which is common to both the

global server 106 and the client 114. The encryption engine 285 of the client 114 and secure communications engine 396 of the global server 114 in step 620 use the secure channel parameters to create the secure communications channel. Method 505 then ends.

FIG. 7 illustrates an example URL-addressable HyperText Markup Language (HTML)-based web page 700, as maintained by the servlet host engine 386. The web page 700 includes a title 710 "Web Page," a listing of the provided services 715 and a pointer 770 for selecting one of the provided services 715. As illustrated, the provided services 715 may include an e-mail service 720, a calendaring service 730, an internet access service 740, a paging service 750 and a fax sending service 760. Although not shown, other services such as bookmarking, QuickCard™, etc. may be included in the web page 700.

FIG. 8A is a flowchart illustrating details of step 540 in a first embodiment, referred to as step 540a, wherein the global server 106 provides the client 114 with a direct connection to the service 110a-110d. Step 540a begins by the downloaded applet 288 in step 805 retrieving the service address 394 of the selected service 110a-110d from data storage device 360 and the authentication information for the service 110a-110d from the keysafe 395. The communications engine 282 in step 810 creates a direct and secure connection with the communications engine 482 of the service server 108 at the retrieved service address, and uses the authentication information to authenticate itself. The applet 288 in step 815 acts as the I/O interface with the service engine. Step 540a then ends.

FIG. 8B is a flowchart illustrating details of step 540 in a second embodiment, referred to as step 540b, wherein the global server 106 acts for the client 114 as a proxy to the service 110a-110d . Step 540b begins with the applet 288 in step 840 retrieving the "service" address, which results in directing it to the global server 106. Thus, the applet 288 in step 845 creates a connection with the global server 106. The servlet host engine 386 of the global server 106 in step 850 retrieves the service address of the selected service 110a-110d and the

authentication information for the selected service 110a-110d from the keysafe 395. The secure communications engine 396 of the global server 106 in step 855 negotiate secure channel parameters for creating a secure channel with the secure communications engine 486 of the service server 108.

Thereafter, the applet 288 in step 860 acts as the I/O interface (enables the user to make requests of the service engine 490) with the secure communications engine 396 of the global server 106. If the servlet host engine 386 in step 865 determines that it is unauthorized to perform a client 114 user's request, then the servlet host engine 386 in step 870 determines whether the method 540b ends, e.g., whether the user has quit. If so, then method 820b ends. Otherwise, method 540b returns to step 860 to obtain another request. If the servlet host engine 386 in step 865 determines that it is authorized to perform the client 114 user's request, then the servlet host engine 386, possibly using servlets 398, acts as the proxy for the client 114 to the service engine 490. As proxy, the servlet host engine 386 forwards the service request to the service 110a-110d for the applet 288 and forwards responses to the requesting applet 288 currently executing on the client 114. Method 540b then returns to step 870.

FIG. 8C is a flowchart illustrating details of step 540 in a third embodiment, referred to as step 540c, wherein the service 110a-110d being requested is located on the global server 106. Step 540c begins with the applet 288 in step 880 retrieving the service address for the service 110a-110d, which results in providing the applet 288 with the service address of the service 110a-110d on the global server 106. Thus, the applet 288 in step 882 creates a secure connection with the global server 106. No additional step of identification and authentication is needed since the client 114 has already identified and authenticated itself to the global server 106 in step 510 of FIG. 5.

In step 884, a determination is made whether the service 110a-110d is currently running. If so, then in step 886 a determination is made whether the service 110a-110d can handle multiple users. If not, then the global server 106 in step 890 creates an instance for the

user, and the applet 288 in step 892 acts as the I/O interface with the service 110a-110d on the global server 106. Otherwise, if the service 110a-110d in step 886 determines that it cannot handle multiple users, then method 540a proceeds to step 892. Further, if in step 884 the global server 106 determines that the service 110a-110d is not currently running, then the global server 106 in step 888 initializes the service 110a-110d and proceeds to step 886.

The foregoing description of the preferred embodiments of the invention is by way of example only, and other variations of the above-described embodiments and methods are provided by the present invention. Components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. The embodiments described herein have been presented for purposes of illustration and are not intended to be exhaustive or limiting. Many variations and modifications are possible in light of the foregoing teaching. The invention is limited only by the following claims.



WHAT IS CLAIMED IS:

1. A system comprising:

a communications engine for establishing a communications link with a client;

security means coupled to the communications engine for determining client privileges;

a servlet host engine coupled to the security means for providing to the client, based on the client privileges, an applet which enables I/O with a secured service; and

a keysafe for storing a key which enables access to the secured service.

2. The system of claim 1, wherein the communications engine uses SSL technology to create a secure communications link with the client.

3. The system of claim 1, wherein communications engine negotiates an encryption protocol for transferring messages to and from the client.

4. The system of claim 1, wherein the communications engine uses public key certificates for transferring messages to and from the client.

5. The system of claim 1, wherein the security means uses public key certificates to authenticate the client.

6. The system of claim 1, wherein the security means examines client identity and the level of authentication to determine client privileges.

7. The system of claim 1, wherein the security means examines a global certificate to authenticate the client.

8. The system of claim 1, wherein the security means uses digital signature technology to authenticate the client.

9. The system of claim 1, wherein the servlet host engine forwards to the client a security applet for enabling the client to perform a security protocol recognized by the security means.

10. The system of claim 1, wherein the service is secured by a corporate firewall and the key is configured to enable communication through the firewall.

11. The system of claim 1, further comprising a global firewall for protecting the system.

12. The system of claim 1, further comprising a service address for identifying the location of the secured service.

13. The system of claim 1, wherein the applet provides to the client a direct connection with the secured service.

14. The system of claim 1, further comprising a proxy in communication with the secured service, and wherein the applet enables I/O with the proxy.

* C L M S *

15. A method comprising the steps of:

establishing a communications link with a client;

determining client privileges;

providing to the client, based on the client privileges, an applet which enables I/O with a secured service; and

retrieving a key which enables access to the secured service.

16. The method of claim 15, wherein establishing a communications link includes the step of using SSL technology to create a secure communications link with the client.

17. The method of claim 15, wherein establishing a communications link includes the step of negotiating an encryption protocol for transferring messages to and from the client.

18. The method of claim 15, wherein establishing a communications link includes the step of using public key certificates for transferring messages to and from the client.

19. The method of claim 15, wherein determining client privileges includes the step of using public key certificates to authenticate the client.

20. The method of claim 15, wherein determining client privileges includes the step of examining client identity and the level of authentication to determine client privileges.

21. The method of claim 15, wherein determining client privileges includes the step of examining a global certificate to authenticate the client.

22. The method of claim 15, wherein determining client privileges includes the step of using digital signature technology to authenticate the client.

23. The method of claim 15, wherein establishing a communications link includes forwarding to the client a security applet for enabling the client to perform a recognized security protocol.

24. The method of claim 15, further comprising the step of using the key to communicate through a firewall to the secured service.

25. The method of claim 15, wherein the method is performed by a global server and further comprising using a global firewall to protect the global server.

26. The method of claim 15, further comprising using a service address to identify the location of the secured service.

27. The method of claim 15, wherein providing includes the step of providing to the client a direct connection with the secured service.

28. The method of claim 15, further comprising using a proxy in communication with the secured service, and wherein providing includes enabling I/O with the proxy.

29. A system comprising:

means for establishing a communications link with a client;

means for determining client privileges;

means for providing to the client, based on the client privileges, an applet which enables I/O with a secured service; and

means for retrieving a key which enables access to the secured service.

30. A computer-based storage medium storing a program for causing a computer to perform the steps of:

establishing a communications link with a client;

determining client privileges;

providing to the client, based on the client privileges, an applet which enables I/O with a secured service; and

retrieving a key which enables access to the secured service.