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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 April 2004 (29.04.2004)

PCT

(10) International Publication Number
WO 2004/036360 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number: PCT/US2003/032570

(22) International Filing Date: 15 October 2003 (15.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/418,759 15 October 2002 (15.10.2002) US

(71) Applicant *(for all designated States except US)*: **INGRIAN NETWORKS, INC.** [US/US]; 475 Broadway Street, Redwood City, CA 94063 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BLYTHE, Matthew** [US/US]; 375 Eunice Avenue, Mountain View, CA 94040 (US). **FRINDELL, Alan** [US/US]; 575 Fairmont Avenue, Mountain View, CA 94041 (US).

(74) Agent: **TAN, Carina**; Perkins Coie LLP, 101 Jefferson Drive, Menlo Park, CA 94025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

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

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

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

WO 2004/036360 A2

(54) Title: CLIENT-SIDE SSL CONNECTION COMPLETION THROUGH SECURE PROXY SERVER

(57) Abstract: When a client establishes an SSL connection in a network that uses a proxy server as an SSL terminating device, the proxy server retrieves the client-side SSL connection information that is associated with the client. The proxy server converts the client-side SSL connection information into a format that can be pre-pended to the data stream sent by the client and that is destined for the back-end application server, in a manner that is independent of the underlying application protocol of the back-end application server.

CLIENT-SIDE SSL CONNECTION COMPLETION THROUGH SECURE PROXY SERVER

FIELD OF THE INVENTION

The present invention relates to secure connection techniques and, more specifically, to providing secure SSL connection through a third party server.

BACKGROUND OF THE INVENTION

FIG. 1 is a block diagram of a network communications system 100 using Secure Socket Layer (SSL). SSL is a protocol used for transmitting private documents. SSL works by using a public key to encrypt data for transfer over the SSL connection. The SSL protocol can be used to safely obtain confidential user information, such as credit card numbers.

Included in system 100 are client computers, of which only one client 102 is shown, communicating through a network 104, such as the Internet, to an application server 108 via an SSL connection 105 and an intermediate server 106. Assume that application server 108 employs a proprietary protocol. When SSL is used, however, performance degradation of the application server is encountered due to the nature of SSL. SSL acceleration techniques are commonly used to address the performance degradation problem. The design of the SSL protocol can involve the exchange of certificates to prove identity. The proprietary protocol, running on application server 108, may rely on the information in the client's certificate to authenticate the client to the server. Traditional SSL acceleration techniques, such as acting as a proxy between the client and the application server, prevent the client's certificate information from reaching the application server. Application servers are herein referred to as back-end application servers.

Restated, when an SSL connection is established, there are certain properties of the SSL connection which are lost to the back-end application servers if the SSL connection is handled by a proxy server which sits in front of the back-end application servers. The lost properties include SSL version information, symmetric cipher choice and strength, and any client certificate information that was presented by the client

when establishing the SSL connection. Information on such properties is herein referred to as client-side SSL connection information.

Often times, primarily for authentication and security purposes, applications running on the back-end application server need access to the client-side SSL connection information. If the underlying application protocol is HTTP, HTTP headers may be used for passing the client-side SSL connection information to backend application servers. However, if the protocol is not HTTP and is arbitrary, there is no defined way to send the client-side SSL connection information back to the back-end application servers.

Accordingly, what are needed are methods and techniques for accelerating traditional SSL connections through third party proprietary network protocols that still allow for client-side SSL connection information to reach a back-end application server.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

FIG. 1 is a block diagram of an network communications system 100 using Secure Sockets Layer (SSL).

FIG. 2A is a high-level block diagram that illustrates aspects of a computerized environment 200 in which client-side SSL connection information can be sent to the relevant back-end application server, according to certain embodiments.

FIG. 2B is a flowchart that illustrates some of the steps that the facility performs for allowing the back-end application server to access client-side SSL connection information, according to certain embodiments.

FIG. 3 is a block diagram that illustrates some of the components typically incorporated in at least some of the computer systems and other devices on which the facility executes.

FIG. 4 is a block diagram that illustrates one sample format into which the client-side SSL connection information can be converted.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A facility for sending client-side SSL connection information to a back-end application server that is using an arbitrary network protocol over SSL is described. For purposes of explanation, a software implementation of the facility is described. However, the facility may be a software implementation, or a hardware implementation, or a combination thereof and may vary from implementation to implementation. The current embodiments are not restricted to any particular implementation.

In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention. U.S. Patent Application No. 10/205,575 (Atty. Docket No. 36321-8010.US01), filed July 24, 2002, entitled "Method and System for Caching Secure Web Content", by Chawla et al., is herein incorporated by reference

FIG. 2A is a high-level block diagram that illustrates aspects of a computerized environment 200 in which client-side SSL connection information can be sent to the relevant back-end application server, according to certain embodiments. The environment 200 includes a client 202, a network 204, a proxy server 206 and a back-end server 208. There may be more than one client and more than one back-end server.

In certain embodiments, retrieval and subsequent conversion of the client-side SSL connection information associated with the client is performed with the aid of one or more other computer systems, such as proxy server 206. Components of the facility may reside on and/or execute on any combination of these computer systems, and intermediate results from the conversion may similarly reside on any combination of these computer systems. The facility may be embodied in a single device or distributed among various devices. In certain embodiments, the proxy server, such as proxy server 206, serves as an SSL termination device with respect to client 202 that is

attempting to send a data stream over an SSL connection to back-end application server 208.

In such embodiments, the client establishes an SSL connection with the proxy server, instead, because the proxy server is the SSL termination device. When the client establishes the SSL connection with the proxy server, the proxy server retrieves the client-side SSL connection information that is associated with that particular client. The proxy server then converts the client-side SSL connection information into a format that can be pre-pended to the data stream sent by the client and that is destined for the back-end application server.

The proxy server converts the client-side SSL connection information in a manner that is independent of the underlying application protocol of the back-end application server. Such a technique of sending client-side SSL connection information to the back-end application server ensures that the back-end application server can access the client-side SSL connection information irrespective of the underlying application protocol employed by the back-end application server.

The computer systems 200 shown in FIG. 2A are connected via network 204, which may use a variety of different networking technologies, including wired, guided or line-of-sight optical, and radio frequency networking. In some embodiments, the network includes the public switched telephone network. Network connections established via the network may be fully-persistent, session-based, or intermittent, such as packet-based. While the facility typically operates in an environment such as is shown in FIG. 2A and described above, those skilled in the art will appreciate the facility may also operate in a wide variety of other environments.

FIG. 2B is a flowchart that illustrates some of the steps that the facility performs for allowing the back-end application server to access client-side SSL connection information, according to certain embodiments. At block 220, the proxy server listens for the client to request an SSL connection with the back-end application server, and the proxy server intercepts the request. At block 222, the proxy server, acting as proxy for the back-end application server, establishes an SSL connection, such as connection A shown in FIG. 2A, with the client. At block 224, when the client has established an

SSL connection with the proxy server, the proxy server retrieves the client-side SSL connection information associated with the client. Client-side SSL connection information includes any information that can be used to identify and/or authenticate the client. Examples of client-side SSL connection information comprise SSL protocol version number, Cipher choice and strength, any and all information in the client certificate.

At block 226 of FIG. 2B, the proxy server, acting as proxy for the client, establishes a connection to the back-end application server. The connection to the back-end application server may be a clear connection or optionally, an SSL connection, such as connection B shown in FIG. 2A. At block 228, the proxy server converts the client-side SSL connection information into a format that is suitable for sending to the back-end application server.

At block 230, the proxy server sends the converted client-side SSL connection information to the back-end application server by pre-pending the converted client-side SSL connection information to the original data stream sent by the client and intended for the back-end application server. Once the back-end application server receives the client-side SSL connection information, the back-end applications server can use the client-side SSL connection information to identify and/or authenticate the client. When identification and/or authentication is complete, a secure tunnel is opened between the client and the back-end application server. At block 232 of FIG. 2B, the proxy server begins forwarding application-protocol-specific data to and from the client and the back-end application server using the secure tunnel between the client and the back-end application server.

FIG. 3 is a block diagram showing some of the components typically incorporated in at least some of the computer systems and other devices on which the facility executes, including some or all of the server and client computer systems shown in FIG. 2A. These computer systems and devices 300 may include one or more central processing units ("CPUs") 301 for executing computer programs; a computer memory 302 for storing programs and data while they are being used; a persistent storage device 303, such as a hard drive, for persistently storing programs and data; a

computer-readable media drive 304, such as a CD-ROM drive, for reading programs and data stored on a computer-readable medium; and a network connection 305 for connecting the computer system to other computer systems, such as via the Internet, to exchange programs and/or data. While computer systems configured as described above are typically used to support the operation of the facility, those skilled in the art will appreciate that the facility may be implemented using devices of various types and configurations, and having various components.

FIG. 4 is a block diagram that illustrates one example format 400 into which the client-side SSL connection information can be converted. In FIG. 4, the converted client-side SSL connection information 402 comprises a Version number of the SSL protocol, a Length information, a certificate subject, and the Carriage Return and Line Feed (CRLF) characters. The length information is for specifying the length of the certification subject plus the CRLF characters. The certificate subject is information from the client certificate that provides information on the identity of the client.

At the beginning of a connection to the back-end application server, 2 bytes are added (in network byte order) to specify the version of the SSL protocol being used. Next, another 2 bytes are added (in network byte order) to specify the length of the certificate subject AND the CRLF characters. The length of the certificate subject AND the CRLF characters can be calculated using equation 1:

EQUATION 1

$$\text{length(certificate subject)} + 2$$

The certificate subject can be sent in raw ASCII characters. The CRLF characters are used as a sentinel at the end of the client certificate information and the beginning of the original data stream.

When the feature for sending client-side SSL information to the back-end application server is enabled and either the connection did not require client certificates or if the connection was an SSL resume, then an empty header of the version could be sent followed by the length of the CRLF, 2, and the CRLF characters, as shown in equation 2:

EQUATION 2

0102\r\n[connection data] (where 1 is the version number)

In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is the invention, and is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any express definitions set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is

1. A method for sending SSL connection information to a back-end application server, the method comprising the computer-implemented acts of:

retrieving said SSL connection information when a client establishes an SSL session that is associated with said SSL connection information;

establishing a connection with said back-end application server;

converting said SSL connection information into a form suitable for attaching to a data stream that is sent by said client during said SSL session and destined for said back-end application server;

attaching said converted SSL connection information to said data stream to form a modified data stream; and

forwarding said modified data stream to said back-end application server.

2. The method as recited in Claim 1, wherein converting said SSL connection information involves using a format that is independent of an underlying protocol associated with said back-end application server.

3. The method as recited in Claim 1, wherein attaching said converted SSL connection information involves pre-pending said converted SSL connection information to said data stream such that said application back-end server reads said converted SSL connection information before reading said data stream.

4. The method as recited in Claim 1, wherein said back-end application server uses said converted SSL connection information to identify said client.

5. The method as recited in Claim 1, wherein said back-end application server uses said converted SSL connection information to authenticate said client.

6. The method as recited in Claim 5, further comprising establishing a secure communication tunnel between said client and said back-end application server after said back-end server has authenticated said client.

7. The method as recited in Claim 6, wherein said secure communication tunnel is used for passing secure data between said client and said back-end application server, using an application-specific protocol associated with said back-end application server, during a duration of said SSL session.

8. The method as recited in Claim 1, wherein said converted SSL connection information includes a certificate subject information from a client certificate that is associated with said client.

9. The method as recited in Claim 1, wherein said converted SSL connection information includes a cipher-information that is associated with a cipher that is associated with said data stream.

10. The method as recited in Claim 1, wherein said converted SSL connection information includes contents of an entire client certificate information that is associated with said client.

11. The method as recited in Claim 1, wherein said converted SSL connection information includes an SSL protocol version number that is associated with said SSL connection.

12. The method as recited in Claim 1, wherein said converted SSL connection information includes a length information that is associated with a total number of bytes of said converted SSL connection information.

13. The method as recited in Claim 10, wherein said converted SSL connection information includes a sentinel indication to denote an end of said client certificate information.

14. The method as recited in Claim 13, wherein said sentinel indication includes a carriage-return character.

15. The method as recited in Claim 13, wherein said sentinel indication includes a line-feed character.

16. A computer-readable medium carrying one or more sequences of instructions for sending SSL connection information to a back-end application server, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

retrieving said SSL connection information when a client establishes an SSL session that is associated with said SSL connection information;

establishing a connection with said back-end application server;

converting said SSL connection information into a form suitable for attaching to a data stream that is sent by said client during said SSL session and destined for said back-end application server;

attaching said converted SSL connection information to said data stream to form a modified data stream; and

forwarding said modified data stream to said back-end application server.

17. The computer-readable medium as recited in Claim 16, wherein converting said SSL connection information involves using a format that is independent of an underlying protocol associated with said back-end application server.

18. The computer-readable medium as recited in Claim 16, wherein attaching said converted SSL connection information involves pre-pending said converted SSL connection information to said data stream such that said application back-end server reads said converted SSL connection information before reading said data stream.

19. The computer-readable medium as recited in Claim 16, wherein said back-end application server uses said converted SSL connection information to identify said client.

20. The computer-readable medium as recited in Claim 16, wherein said back-end application server uses said converted SSL connection information to authenticate said client.

21. The computer-readable medium as recited in Claim 20, further comprising establishing a secure communication tunnel between said client and said back-end application server after said back-end server has authenticated said client.

22. The computer-readable medium as recited in Claim 21, wherein further said secure communication tunnel is used for passing secure data between said client and said back-end application server during a duration of said SSL session.

23. The computer-readable medium as recited in Claim 16, wherein said converted SSL connection information includes a certificate subject information from a client certificate that is associated with said client.

24. The computer-readable medium as recited in Claim 16, wherein said converted SSL connection information includes a cipher-information that is associated with a cipher that is associated with said data stream.

25. The computer-readable medium as recited in Claim 16, wherein said converted SSL connection information includes contents of an entire client certificate information that is associated with said client.

26. The computer-readable medium as recited in Claim 16, wherein said converted SSL connection information includes an SSL protocol version number that is associated with said SSL connection.

27. The computer-readable medium as recited in Claim 16, wherein said converted SSL connection information includes a length information that is associated with a total number of bytes of said converted SSL connection information.

28. The computer-readable medium as recited in Claim 25, wherein said converted SSL connection information includes a sentinel indication to denote an end of said client certificate information.

29. The computer-readable medium as recited in Claim 28, wherein said sentinel indication includes a carriage-return character.

30. The computer-readable medium as recited in Claim 28, wherein said sentinel indication includes a line-feed character.

31. A facility, for sending SSL connection information to a back-end application server, said facility comprising:

at least one processing device operable as a proxy server;

wherein said SSL connection information is associated with a client and said at least one processing device is adapted for packaging said SSL connection information into a format that is :

suitable for pre-pending to a data stream sent by said client and destined for said back-end application server;

independent of any underlying application protocol associated with said back-end application server.

32. A computer-implemented method suitable for use by a proxy server securing a back-end application server from an unsecure network, said method useful for establishing a secure communications channel from a client to said back-end application server, said method comprising the acts performed by said proxy server of:

intercepting a request initiated by said client to establish a secure client-to-application connection with said application server;

establishing a secure client-to-proxy connection with said client, wherein said proxy server acts as a proxy for said back-end application server such that said client-to-proxy connection appears as said client-to-application connection to said client;

retrieving secure connection information that is associated with said client;

establishing a proxy-to-application connection with said back-end application server, wherein said proxy server acts as a proxy for said client such that said proxy-to-application connection appears as said client-to-application connection to said application server; and

wherein said secure client-to-application connection is established by using at least both said secure client-to-proxy connection and said proxy-to-application;

converting said secure connection information into a form suitable for attaching to a data stream that is sent by said client and destined for said back-end application server;

attaching said converted secure connection information to said data stream to form a modified data stream; and

forwarding said modified data stream to said back-end application server.

33. A proxy server suitable for securing a back-end application server from an unsecure network, said back-end application server intended to provide services to clients via said unsecure network, said proxy server comprising:

persistent memory storing computer executable instructions for:

intercepting a request initiated by said client to establish a secure client-to-application connection with said application server;

establishing a secure client-to-proxy connection with said client, wherein said proxy server acts as a proxy for said back-end application server such that said client-to-proxy connection appears as said client-to-application connection to said client;

retrieving secure connection information that is associated with said client;

establishing a proxy-to-application connection with said back-end application server, wherein said proxy server acts as a proxy for said client such that said proxy-to-application connection appears as said client-to-application connection to said application server; and

wherein said secure client-to-application connection is established by using at least both said secure client-to-proxy connection and said proxy-to-application;

converting said secure connection information into a form suitable for attaching to a data stream that is sent by said client and destined for said back-end application server;

attaching said converted secure connection information to said data stream to form a modified data stream; and

forwarding said modified data stream to said back-end application server.

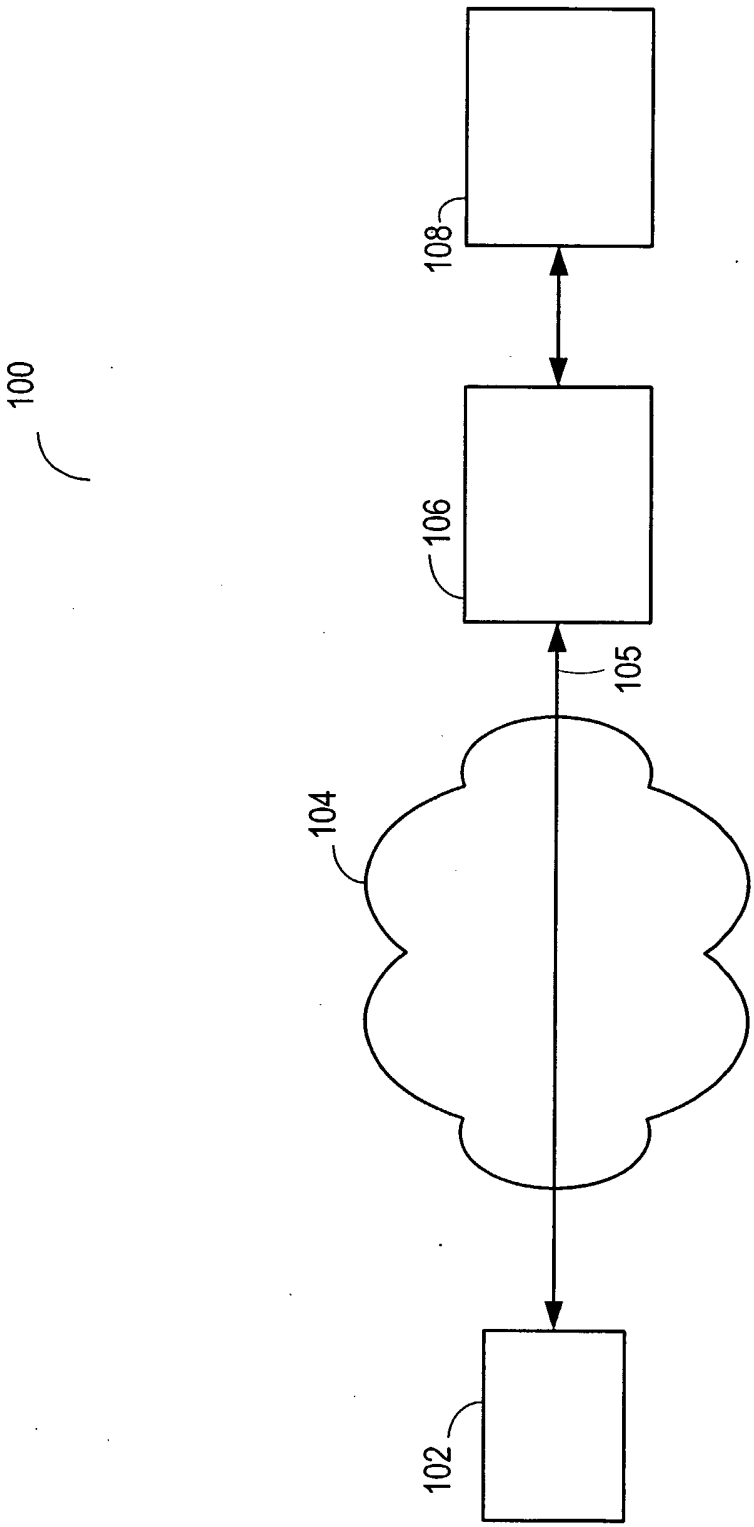

FIG. 1

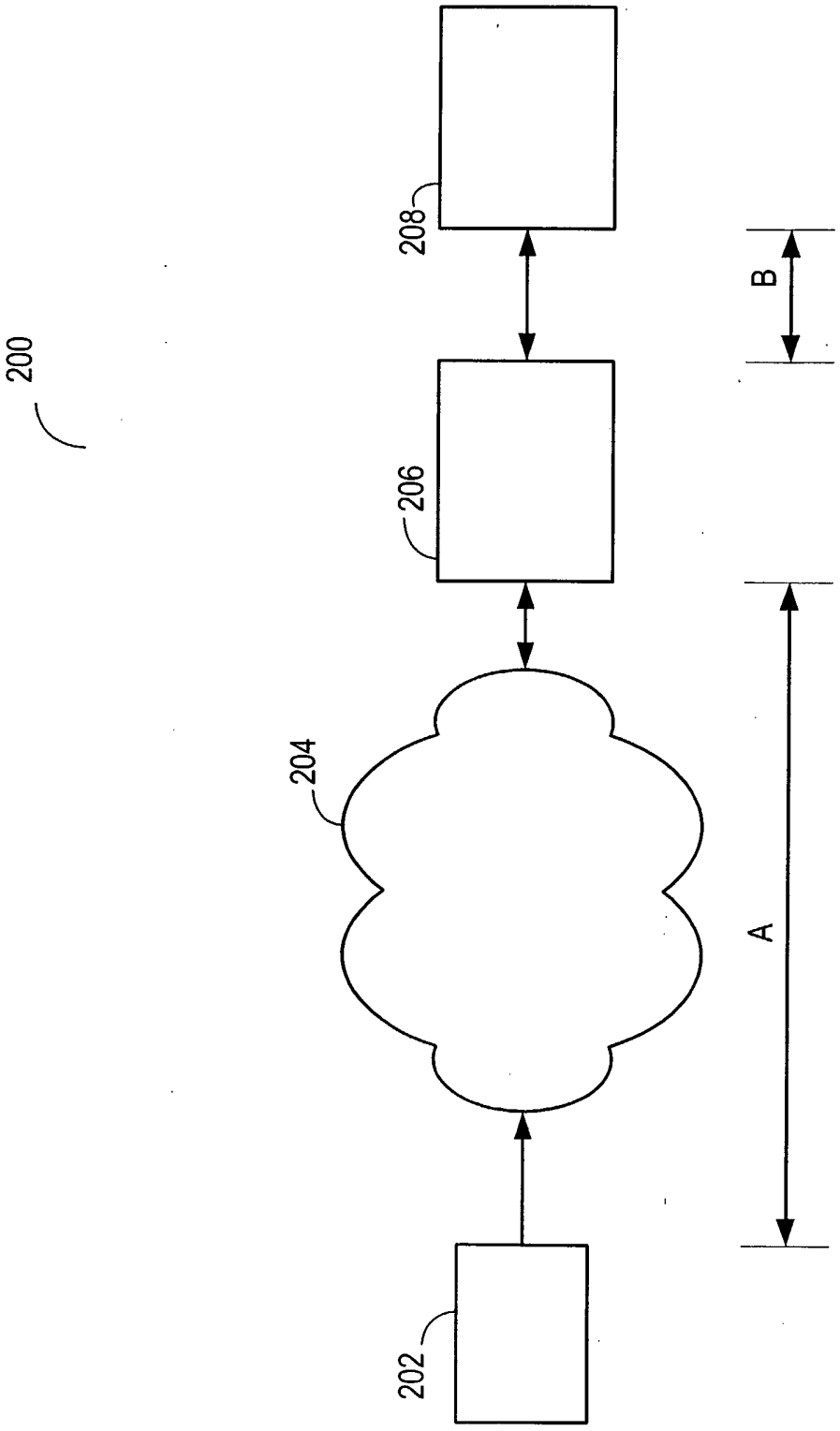

FIG. 2A

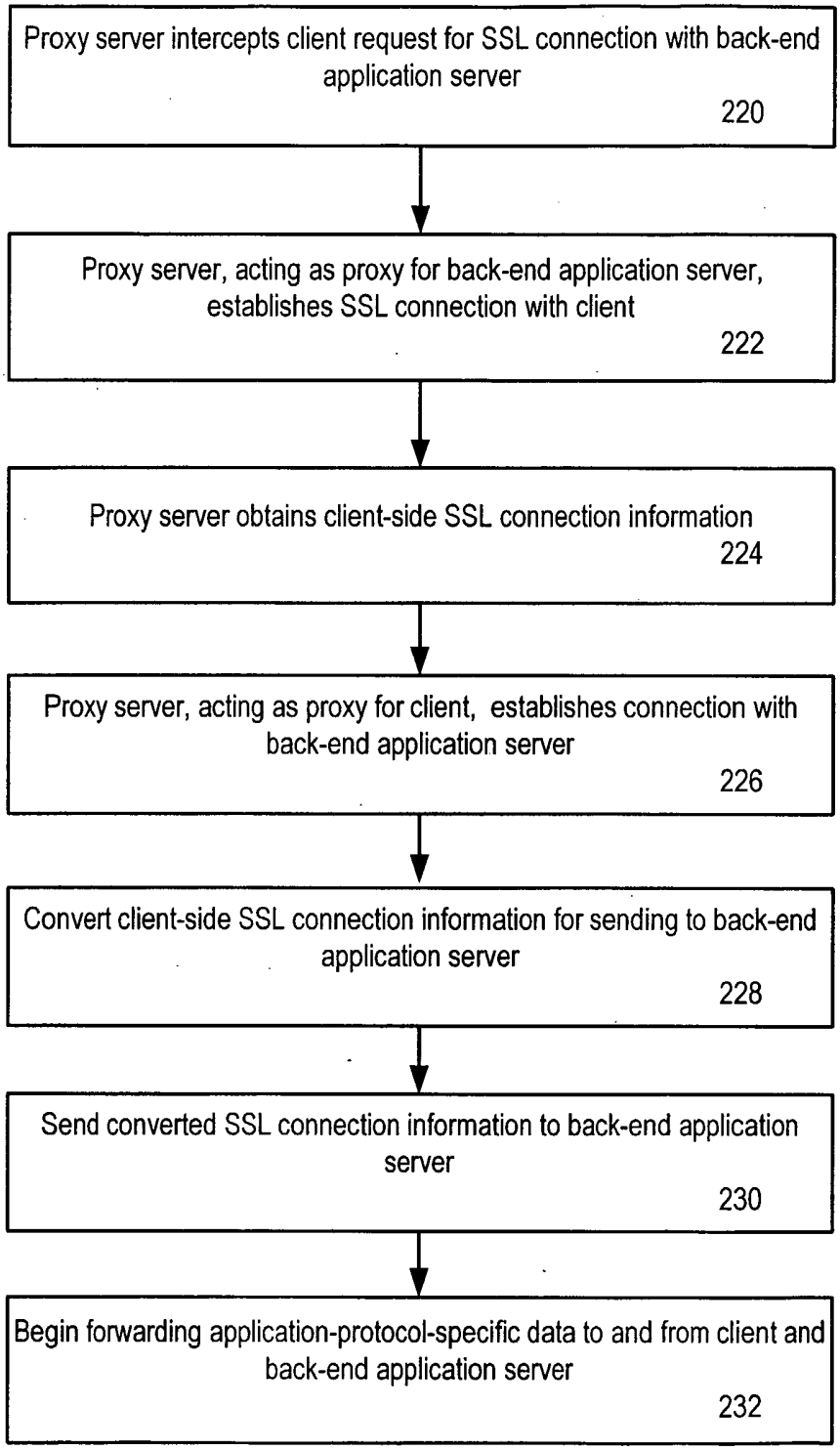
Proxy server intercepts client request for SSL connection with back-end application server
220
Proxy server, acting as proxy for back-end application server, establishes SSL connection with client
222
Proxy server obtains client-side SSL connection information
224
Proxy server, acting as proxy for client, establishes connection with back-end application server
226
Convert client-side SSL connection information for sending to back-end application server
228
Send converted SSL connection information to back-end application server
230
Begin forwarding application-protocol-specific data to and from client and back-end application server
232

FIG. 2B

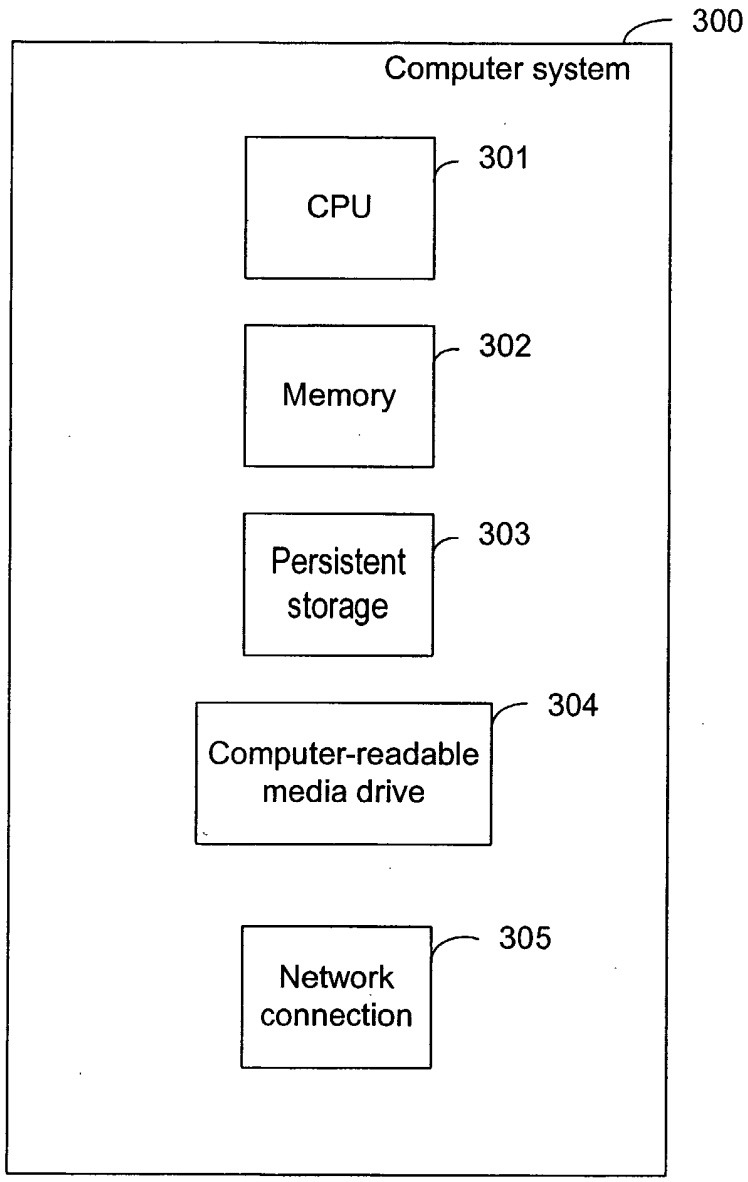
300
Computer system
CPU
301
Memory
302
Persistent storage
303
Computer-readable media drive
304
Network connection
305

FIG. 3

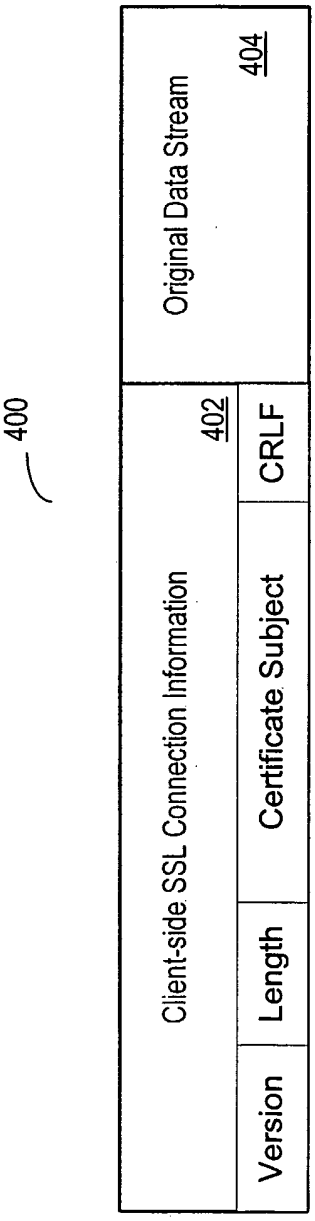
400
Client-side SSL Connection Information
402
Version
Length
Certificate Subject
CRLF
Original Data Stream
404

FIG. 4