

authentication IDs contains a plurality of authentication IDs. As described above, the authentication ID provider 940 may comprise code which assigns one or more authentication IDs to a client application 15 when the client application 15 logs into a web service 25. These authentication IDs are passed as parameters in the method calls 701, 751 as described above. The authentication ID validator 945 may comprise code to validate the authentication ID. This authentication code be done in a number of ways. In an example of an embodiment of the present invention, a working table mapping is setup when the client application 15 is authenticated (i.e., authentication ID returned). The authentication ID is checked every time a method is called, then deleted if the client application 15 logs off or the authentication ID expires. An alternative of using a hashing system would require care to remain as secure.

[0098] The user of a client application 15 logs onto a web service 25 by sending client application credentials, typically a user name and password, over a secured channel as described above. In return, the client application 15 will receive a group or pool of authentication IDs. The pool of authentication IDs returned is secure since the pool is sent back over the secured channel. The exact number of authentication IDs returned can vary depending on the system administration requirements for the web service 25. Once the client application 15 has this pool of authentication IDs, the client application 15 may use a different authentication ID from this pool with each successive call to the web service 25. The authentication ID that is used will then expire upon use so that it can not be reused. This means that even if an eavesdropper is able to compromise an authentication ID, the eavesdropper will not be able to use it since it can only be used once.

[0099] FIG. 11 shows the sequence of logging onto a web service 25 and using the pool of authentication IDs. In FIG. 11, the sequences are listed as A, B, C1, R1, . . . , Cn, Rn, where n is an integer greater than one. The step "A" represents a client application 15 sending client application credentials over a secured channel, such as https. The step "B" represents the server authenticating the user and returning a pool of n authentication IDs over the secured channel. The steps "C1" to "Cn" represent the client application 15 making up to n web service calls over an unsecured channel using a different authentication ID from the pool of n IDs returned. Each authentication ID will expire upon use. The steps "R1" to "Rn" represent the server validating the authentication ID used and returning the result of the web service call to the client application 15.

[0100] After the client application 15 has used up all the authentication IDs in the pool that was given, the client application 15 may log on again to receive another pool of authentication IDs. No one other than the client application 15 will be able to use the authentication IDs since the authentication IDs are always given to the client application 15 over a secured channel and the authentication IDs expire upon use. Each authentication ID is not compromised during or after its use over an unsecured channel because an unauthorized person who manages to capture an authentication ID only receives an expired authentication ID.

[0101] Further security features may be added to the pool of authentication IDs. For example, unused authentication IDs in a pool of authentication IDs can be set to expire after a preset event such as the expiry of a period of time.

[0102] FIG. 12 shows a method for providing a pool of authentication IDs (1200) for use in web services communication. The method begins with the client application interface unit 310 receiving a request for a pool of authentication IDs (1201) over a secured channel. Typically, the request will come from a user using a client application 15. The request is passed to the authentication ID provider 940 of the login services module 980. The authentication ID provider 940 creates and assigns a pool of authentication IDs (1202). The authentication IDs may be passed as parameters by the client application 15 during web service communication, such as SOAP communication. The authentication IDs may be created and assigned by code in the authentication ID provider 940. The pool of authentication IDs is passed to the client application interface unit 310 to be sent to the client application 15 (1203) over a secured channel and the method is done (1204). The client application 15 may now use the authentication IDs.

[0103] FIG. 13 shows a method for using a pool of authentication IDs. During subsequent client calls 701 over an unsecured channel such as http, an authentication ID from the pool of authentication IDs is sent as a parameter in the client calls 701. The client application interface unit 310 receive a client call 701 containing the authentication ID (1301). The authentication ID is parsed from the client call 701 by the communication processor 311, as described above, and passed to the authentication ID validator 945. If the authentication ID is not valid (1302), then the client call 701 is rejected and the method is done (1305). If the authentication ID is valid (1302), then the next step is to check whether the client application 15 is authorized to access the web service method relating to the client call 701 (1303). If the client application 15 is not authorized (1303), then the client call 701 is rejected and the method is done (1305). If the client application 15 is authorized (1303), then the WS call 702 is sent (1304), as described above, and the method is done (1305).

[0104] FIG. 14 shows another example of a login services module 1400 in accordance with an embodiment of the present invention. The login services module 1400 may be used by the gateway module 900. The login services module 1400 comprises an authentication ID provider 940, an authentication ID validator 945, an authentication module 520, an authorization module 525, and an information repository 1401. The authentication module 520, authorization module 525, authentication ID provider 940 and authentication ID validator 945 are similar to those described above. The information repository 1401 contains information used to authenticate and authorize client applications 15, as well as storing authentication ID allocations. The information repository 1401 may be a database. The authentication ID provider 940, authentication ID validator 945, authentication module 520, and authorization module 525 are connected to the information repository 1401 and may be accessed by the communication processor 311.

[0105] Alternatively, the repository 1401 may be accessed by components of the gateway module 900, including the metering module 950 and the billing module 970. Client applications 15 may be charged for the pool of authentication IDs based upon the size of the pool of authentication IDs. Packages of authentication IDs may be available for a client application 15 to order. For example, a client application 15 may order a basic package of 100 authentication