Title

An Authentication System

Technical Field

The invention concerns a method for authenticating a user for accessing secure data.

Background of the Invention

The Internet is used by companies and individuals to access data, for banking, and to perform certain transactions such as money transfer. Some data is private and confidential, such as personal email or financial accounts. Access to such private and confidential information is usually restricted to only authorised users. Currently, most servers that house these information require the authorised users to "log on" to the server and authenticate themselves. Basic authentication requires a username and a corresponding password. Usually, the username and password are encrypted and transmitted to the server via secure sockets layer (SSL).

Although encryption may be used, passwords that are transmitted over the Internet may be intercepted and decrypted. Passwords may also be captured by other means, for example by trojan horse applications that are secretly installed on computers to record the key strokes of people using the computer.

Some methods have been proposed to enhance online user authentication using two sets of passwords. Although the introduction of a second password enhances the authentication/authorisation process, it does not address the risks posed by stolen passwords especially when both sets of passwords are stolen.

Summary of the Invention

In a first preferred aspect, there is provided a method for authenticating a user for accessing secure data, the method comprising:

transmitting user identification information for identifying the user, from the user to an authentication server;

generating a session key for associating the user with the server in a session, the session key being transmitted from the server to the user; and

transmitting an authorisation key for authenticating the user to the server in the session, the authorisation key being transmitted from the user to the server;

wherein the authorisation key is formed by performing a predetermined operation on the session key, the predetermined operation being defined by a private key of the user; and

whereby if the authorisation key transmitted by the user matches a corresponding authorisation key generated by the server, access to the secure data is granted.

The predetermined operation may be a mathematical or logical operation. The mathematical operation may be one or more selected from the group consisting of: addition, subtraction, multiplication, division, and modulus.

The session key may consist of alphanumeric characters. If the session key comprises alphanumeric characters, one exemplary logical operation on the session key is that the alphanumeric characters are shifted by a predetermined number. For example, a session key of "EPH", after performing a shift of one alphabet provides an authorisation key of "DOG".

The private key may also include use of a password. The password may be user generated or server generated. For example, the private key may define an operation of shifting one alphabet and concatenating the password. In this example, a session key of "EPH" and password of "CAT", after performing a shift of one alphabet and concatenating the password provides an authorisation key of "DOGCAT".

The logical operation may be one or more selected from the group consisting of: appending the password before or after the session key, inserting the password within the session key, re-ordering the session key based on the password and using a portion of the session key based on the password.

The server may retrieve a copy of the private key of the user from a secure database of private keys.

The session key may be received by the user via a mobile communications device or a third party software application. The mobile communications device may be one or more selected from the group consisting of: a mobile phone, a pager, a

personal digital assistant (PDA) and a notebook computer with wireless communication functionality. The third party software application may be one or more selected from the group consisting of: an email client application and an instant messenger service.

The session key may be transmitted via one or more selected from the group consisting of: Short Messaging Service (SMS), Multimedia Message Service (MMS) message, pager message, instant message, and email.

The user identification information may be one or more selected from the group consisting of: the name of the user, a login name, phone number or pager number of the user, and an email address.

The user identification information and authorisation key may be transmitted by the user to the server via a client. The client may be one or more selected from the group consisting of: an Internet browser application, a Point of Sale (POS) device, an SMS or MMS application, an email client application, and an instant messaging application.

The session between the client and the server may be assigned a session identifier by the server. The session identifier may be stored in a cookie in the client. The session identifier may have a predetermined expiry time period. The session identifier may be defined by Uniform Resource Locator (URL) rewriting or Hidden Form Fields.

Each authorisation key may be unique to a pair of session and private keys.

The session key may have a predetermined expiry time period. The server may check whether the session key has expired, and if expired, another session key is generated for transmission to the user on request.

The user may request a session key from the server. The session key may be requested by one or more selected from the group consisting of: submitting a HTTP request from an Internet browser, transmitting an SMS message or MMS message, transmitting a paging message, transmitting an email to the server, and interacting with an Interactive Voice Response (IVR) unit connected to the server.

A retry limit may be imposed to prevent the user from re-entering the user identifier or authorisation key above the retry limit. The maximum retry limit may be imposed for a predetermined period of time.

The secure data may include personal financial data, subscription-based publications, or an email account. The secure data may include a protected resource such as access to web service or server-based software module.

In a second aspect, there is provided a system for authenticating a user for accessing secure data, the system comprising:

a client to transmit user identification information for identifying the user;

a server to generate a session key for associating the user with the server in a session;

a session key transmission module to transmit the generated session key to the user, the session key transmission module being in communication with the server; and

a session key receiver module to receive the generated session key from the server;

wherein an authorisation key is transmitted from the client to the server by the user, and the authorisation key is formed by performing a predetermined operation on the session key, the predetermined operation being defined by a private key of the user; and

whereby if the authorisation key from the user matches a corresponding authorisation key generated by the server, access to the secure data is granted.

The session key transmission module may be a Short Messaging Service Centre (SMSC), GSM modem or GSM phone.

The session key receiver module may be a mobile communications device of the user. The session key receiver module may be a software module installed on a mobile communications device of the user. For example, the session key receiver module may be a Java-based program for a mobile phone such as the Sony Ericsson P900.

In a third aspect, there is provided an authentication server for authenticating a user for accessing secure data, the server comprising:

a session key generation module to generate a session key for associating the user with the server in a session;

a communications module to receive user identification information for identifying the user, to transmit the generated session key via a session key transmission module to the user, and to receive an authorisation key from the user; and

a secure database for storing a copy of a private key of the user;

wherein the server generates an authorisation key by performing a predetermined operation defined by the stored private key of the user on the generated session key; and

whereby if the authorisation key from the user matches the authorisation key generated by the server, access to the secure data is granted.

The authorisation key generated by the server may be generated when the server receives an authorisation key from the user. Alternatively, authorisation key generated by the server may be generated when the generated session key is transmitted to the user.

A session key table may be provided to store generated session keys.

Brief Description of the Drawings

An example of the invention will now be described with reference to the accompanying drawings, in which:

Figure 1A is a first screen design diagram of the authentication process;

Figure 1B is a second screen design diagram of the authentication process;

Figure 2 is a block diagram of the authorisation system;

Figure 3 is a process flow diagram illustrating a session key request at the server; and

Figure 4 is a process flow diagram illustrating an authorisation key verification request at the server.

Detailed Description of the Drawings

Figure 1 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the present invention may be implemented. Although not required, the invention will be described in the

general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routines, programs, characters, components, data structures, that perform particular tasks or implement particular abstract data types. As those skilled in the art will appreciate, the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Referring to Figures 1A, 1B and 2, a system 10 for authenticating a user for accessing secure data is provided. The system 10 is a client/server system and comprises a server 210 to generate a session key for associating the user with the server in a session, and a client 230 to receive user identification information for identifying the user, and to receive an authorisation key from the user for authenticating the user to the server 210 in the session. The authorisation key is formed by performing a predetermined operation on the session key. The predetermined operation being defined by a private key of the user. Firstly, the user identification information is transmitted from the client 230 to the server 210. Secondly, the session key generated by the server 210 is transmitted from the session key transmission module 220 to the session key receiving module 240. Thirdly, the authorisation key is transmitted from the client 230 to the server 210. If the authorisation key from the user matches a corresponding authorisation key generated by the server 210, access to the secure data is granted.

The client 230 has an authorisation user interface for authenticating users. In one embodiment, the interface is via a web browser application. Other clients 230 include a Point of Sale (POS) device, an SMS or MMS application, an email client application, or an instant messaging application.

A first web page for a user to request for a session key is illustrated in Figure 1A. The first web page is sent from the server 210 to the client 230 when the user makes an initial request to access secure data or a protected resource. The secure data is only available to an authorised user.

The first web page generally comprises a section describing the authorisation process 101, a section describing a session key 102, a user identifier input field 103 and a submit button 104. The authorisation process description section 101 informs the user on the authorisation process and subsequent screens. The session key description section 102 informs the user on how the session key will be transmitted to them, the expiry time period of the session key, and any other useful information about the session key. The user identifier input field 103 allows the user to enter user identification information or a user identifier to uniquely identify themselves to the server 210. The submit button 104 allows the client 230 to submit a user identifier to the server 210 for verification purpose. When a user clicks the submit button 104, the client 230 transmits the user identifier in the input field 103 to the server 210. This acts as a request for the generation of a session key by the server 210 and provided to the user via a session key receiving module 240.

If the user identifier is invalid, the server 210 re-directs the user back to the first web page and informs the user to re-enter a valid user identifier. If the user identifier is valid, the server 210 dynamically generates a random session key and transmits this to the user to their nominated session key receiving module 240. For example, a mobile communications device or a third party software application. The mobile communications device includes a mobile phone, a pager, a personal digital assistant (PDA) or a notebook computer with wireless communication functionality. Third party software applications include an email client application or an instant messenger service. At the same time, a second web page illustrated in Figure 1B is presented to the client 230. The second web page confirms the validity of the user identifier and prompts the user to enter an authorisation key to complete the authorisation process. Users are prevented from directly accessing the second web page without successfully proceeding past the first web page. The second web page generally comprises a section describing the authorisation key 105, an authorisation key input field 106, and a submit button 107.

The authorisation key description section 105 on the second web page explains to the user how an authorisation key is formed using the session key and the private key of the user. The authorisation key is formed by performing certain operations determined by a private key on a session key. The private key is known to the user and stored in the server 210. The private key may include use of a personal

password of the user which is easily remembered by the user. In one embodiment, the private key defines concatenating the session key with the personal password of the user to generate an authorisation key. That is, the personal password and its concatenation operation constitute the private key of the user, and the operation of the private key with the session key generates the authorisation key. Other operations include the user performing a mathematical or logical operation on the session key using their private key. The mathematical operation may include addition, subtraction, multiplication, division, or modulus of both the session key and private key. The private key may define a logical operation of appending the user's password before the session key, or inserting the password within the session key. A private key without the use of a password may define re-ordering the session key based on the private key or using a portion of the session key based on the private key.

Once the user determines how to form the authorisation key, the user inputs the authorisation key into the authorisation key input field 106. The submit button 107 submits the authorisation key entered in the authorisation key input field 106 to the server 210 for verification.

After the user clicks the submit button 107, the client 230 transmits the authorisation key to the server 210 requesting that access be granted to the secure data. The server 210 ensures that the session key associated with the current session is not expired. If the session key has expired, the server 210 informs the user that a request for a new session key is required. The user is re-directed to the first web page to make this request. The server 210 performs a validity test by comparing the received authorisation key with another authorisation key generated by using the session key associated with the current session and the private key associated with the user of the same session.

The authorisation process is completed when there is a match between the server generated authorisation key and the authorisation key received from the user. If there is no match, the server 210 re-directs the user to the second web page again and asks the user to re-enter the correct authorisation key.

In an example where the private key defines concatenation of the session key with the use of a password, security is increased by ensuring that the session key and private key pair maintain a one-to-one correspondence to the authorisation key. In

other words, different pairs of the session key and private key yield different authorisation keys to prevent a coincidental match of the received authorisation key with a generated authorisation key. The server 210 may recommend several private keys which adhere to this principle to users if they cannot generate such private key themselves.

Referring to Figure 2, authorisation over the Internet via the World Wide Web is shown. The server 210 comprises a server engine 211, various Web resources 212, a user database 213, a session identifier/user identifier table 214, and a session identifier/session key table 215. The server engine 211 receives HTTP requests from clients to access Web resources identified by URLs and provides the Web resources to the various clients. Some of these Web resources are protected resources which may be linked to secure data, for example, online banking facilities to access the user's account. Access to these protected resources require authentication of the user. The user database 213 contains user information such as the name of the user, mobile number, and private key of the user. The session identifier/user identifier table 214 contains a mapping from each session identifier 232, which is a globally unique identifier that uniquely identifies a session to the user of the same session. The session identifier/session key table 215 contains a mapping from the same session identifier 232 to the session key associated with the same session.

The session key transmission module 220 is responsible for sending the session key generated by the server 210 to the session key receiving module 240 accessible by the user. In one embodiment, a Short Message Service Centre (SMSC) functions as a session key transmission module 220 by sending the session key to the mobile phone 240 of the user via SMS or MMS. Other methods to send the session key to the user are possible, such as email, instant message (ICQ or MSN messenger service), phone call to an IVR unit, or a paging message. Other session key transmission modules 220 include a GSM modem or a GSM phone.

The client 230 is provided with a web browser application 231. The session between the web browser 231 and the server 210 is assigned a session identifier 232. In one embodiment, the session identifier 232 is stored in a file, referred to as a "cookie" in the client 230. The server 210 assigns and initially sends the session identifier 232 to the client 230 when the client 230 makes the initial connection with

the server 210 for each session. Afterwards, the client 230 includes its session identifier 232 with all requests sent to the server 210 during each session so that the server 210 is able to identify the specific client 230 making the request.

In other embodiments, the session identifier 232 is defined by Uniform Resource Locator (URL) rewriting or Hidden Form Fields. For URL rewriting, the client 230 appends additional data to the end of each URL in HTTP requests to the server 210. This additional data identifies the session, and the server 210 associates this session identifier with stored user-specific data. For example, a URL of http://host/path/file.html;sessionid=a1234, has a session identifier attached in the form of sessionid=a1234. For Hidden Form Fields, when a form is submitted, the specified name and value of the form field are automatically included in the HTTP request. For example, the HTML tag of <INPUT TYPE="HIDDEN" NAME="session" VALUE="a1234">, has a session identifier of "a1234".

Referring to Figure 3, the process of making a session key request to the server 210 to obtain access to the secure data. The client 230 makes a request for a session key from the server 210. The server 210 retrieves 301 the user identifier from the request of the client 230. If the user identifier is valid 302, the server 210 continues 303 processing the request. Otherwise the server 210 requests that a valid user identifier be entered 304. The server 210 generates 303 a globally unique session identifier 232 for each session. The server 210 updates 305 the session identifier/user identifier table 214 to indicate that the generated session identifier 232 has been associated with the user. The server 210 generates 306 a session key with a predetermined expiry time period. The server 210 updates 307 the session identifier/session key table 215 to indicate that the generated session identifier 232 has been associated with the session key. The server 210 instructs the session key transmission module 220 to send 308 the session key to the session key receiving module 240. The session key receiving module 240 could be a mobile phone accessible by the user. In other embodiments, the session key receiver module includes a pager, a Personal Digital Assistant (PDA), an email client, an instant message client or similar devices. Next, the server 210 fulfills the session key request by supplying 309 the session identifier 232 and a confirmation message to the client 230.

Referring to Figure 4, the process for verifying an authorisation key request at the server 210 is shown. To complete the authorisation process, the user supplies a

valid authorisation key to the server 210. The server 210 retrieves 401 the session ID embedded in the request. If the session ID provided is valid 402, the server 210 continues verification. Otherwise the server 210 requests 404 the user to obtain a new session key by submitting another session key request. Next, the server 210 retrieves 403 the session key associated with the session ID. If the session key associated with the session has not expired 405, the server 210 continues verification. Otherwise, if the session key has expired, the server 210 requests 404 that the user to obtain a new session key by submitting another session key request. The server 210 retrieves 406 the private key associated with the user of the current session. In one embodiment, the server retrieves a copy of the private key of the user from a secure database of private keys (not shown). The server 210 generates 407 an authorisation key by using the session key associated with the current session and the private key associated with the user of the same session. The authorisation key is generated by the server 210 when the server 210 receives an authorisation key from the user. In another example, the authorisation key is generated by the server 210 when the session key generated by the server 210 is transmitted to the user. If the received authorisation key matches 408 the server generated authorisation key, the server 210 continues 410 verification. Otherwise the server 210 requests 409 the user to re-enter a valid authorisation key. To enhance security, the server 210 may place a limit on the number of attempts to re-enter the user identifier and/or authorisation key. The server 210 returns 410 a confirmation message to the user to inform that the authorisation key is accepted and access to the secure data is granted.

The session key may be made globally unique so that it can be treated as a session identifier 232 as well. As a result, the server 210 does not need to generate session identifiers 232 or maintain a session identifier/session key table 214.

Although requesting a session key via a web page (Figure 1A) has been described, other ways are possible. For example, the user may request a session key by sending an SMS or MMS in a specific format from their mobile phone to the Server 210. The user may request a session key by making a phone call and interacting with an Interactive Voice Response (IVR) unit. Alternatively, the user may send an email to the server 210 to request a session key.

To save computer resources at the server 210, an expiry condition may be assigned to a session identifier 232 similar to one assigned to a session key. The server 210 is able to re-use expired session identifiers and avoid depleting usable session identifiers 232.

In a further example, an authorisation key generating device (not shown) is provided. The device is connected to the mobile communications device of the user. The device may be embodied in software and communicates with the mobile communications device of the user. Generation of the authorisation key is only performed when password clearance has been effected. This password is part of the original private key.

Although a client/server system has been described, it is envisaged that other systems are possible such as a peer to peer system, or a multi-tier system. Although an Internet or web browser 231 has been described, other applications may be used for interacting with the server 210 from the client 230.

Various communication channels may be used with the present invention including the Internet, mobile communications network, local area networks, wide area networks, or point-to-point dial up connections. In one embodiment, the server 210 communicates with the session key transmission module 220 via a link such as the Internet, a wireless network, a leased line, or a data cable. The session key transmission module 220 and session key receiving module 240 may communicate with each other similarly.

The server 210 may comprise a combination of hardware or software programs that can process an authorisation request. The client 230 may comprise a combination of hardware or software programs to enable interaction with the server 210. The system 10 may include point-of-sales device or various other consumer products through which requests may be made.

It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the invention as shown in the specific embodiments without departing from the scope or spirit of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects illustrative and not restrictive.

WE CLAIM:

1. A method for authenticating a user for accessing secure data, the method comprising:

transmitting user identification information for identifying the user, from the user to an authentication server;

generating a session key for associating the user with the server in a session, the session key being transmitted from the server to the user; and

transmitting an authorisation key for authenticating the user to the server in the session, the authorisation key being transmitted from the user to the server;

wherein the authorisation key is formed by performing a predetermined operation on the session key, the predetermined operation being defined by a private key of the user; and

whereby if the authorisation key transmitted by the user matches a corresponding authorisation key generated by the server, access to the secure data is granted.

2. The method according to claim 1, wherein the predetermined operation is a mathematical or logical operation.

3. The method according to claim 2, wherein the mathematical operation is one or more selected from the group consisting of: addition, subtraction, multiplication, division, and modulus.

4. The method according to claim 1, wherein the session key consists of alphanumeric characters.

5. The method according to claim 1, wherein the private key includes use of a password.

6. The method according to claim 5, wherein the password is user generated or server generated.

7. The method according to claim 2, wherein the logical operation is one or more selected from the group consisting of: appending a password before or after the session key, inserting a password within the session key, re-ordering the

session key based on a password and using a portion of the session key based on a password.

8. The method according to claim 1, wherein the server retrieves a copy of the private key of the user from a secure database of private keys.

9. The method according to claim 1, wherein the session key is received by the user via a mobile communications device or a third party software application.

10. The method according to claim 9, wherein the mobile communications device is one or more selected from the group consisting of: a mobile phone, a pager, a personal digital assistant (PDA) and a notebook computer with wireless communication functionality.

11. The method according to claim 9, wherein the third party software application is one or more selected from the group consisting of: an email client application and an instant messenger service.

12. The method according to claim 1, wherein the session key is transmitted via one or more selected from the group consisting of: Short Messaging Service (SMS), Multimedia Message Service (MMS) message, pager message, instant message, and email.

13. The method according to claim 1, wherein the user identification information is one or more selected from the group consisting of: the name of the user, a login name, phone number or pager number of the user, and an email address.

14. The method according to claim 1, wherein the user identification information and authorisation key are transmitted by the user to the server via a client.

15. The method according to claim 14, wherein the client is one or more selected from the group consisting of: an Internet browser application, a Point of Sale (POS) device, an SMS or MMS application, an email client application, and an instant messaging application.

16. The method according to claim 14, the session between the client and the server are assigned a session identifier by the server.

17. The method according to claim 16, wherein the session identifier has a predetermined expiry time period

18. The method according to claim 16, wherein the session identifier is stored in a cookie in the client.

19. The method according to claim 16, wherein the session identifier is defined by Uniform Resource Locator (URL) rewriting or Hidden Form Fields.

20. The method according to claim 1, wherein each authorisation key is unique to a pair of session and private keys.

21. The method according to claim 1, wherein the session key has a predetermined expiry time period.

22. The method according to claim 21, wherein the server checks whether the session key has expired, and if expired, another session key is generated for transmission to the user on request.

23. The method according to claim 1, wherein the user requests a session key from the server.

24. The method according to claim 23, wherein the session key is requested by one or more selected from the group consisting of: submitting a HTTP request from an Internet browser, transmitting an SMS message or MMS message, transmitting a paging message, transmitting an email to the server, and interacting with an Interactive Voice Response (IVR) unit connected to the server.

25. The method according to claim 1, wherein a retry limit is imposed to prevent the user from re-entering the user identifier or authorisation key above the retry limit.

26. The method according to claim 25, wherein the maximum retry limit is imposed for a predetermined period of time.

27. The method according to claim 1, wherein the secure data includes personal financial data, subscription-based publications, or an email account.

28. The method according to claim 1, wherein the secure data includes a protected resource such as access to web service or server-based software module.

29. A system for authenticating a user for accessing secure data, the system comprising:

a client to transmit user identification information for identifying the user;

a server to generate a session key for associating the user with the server in a session;

a session key transmission module to transmit the generated session key to the user, the session key transmission module being in communication with the server; and

a session key receiver module to receive the generated session key from the server;

wherein an authorisation key is transmitted from the client to the server by the user, and the authorisation key is formed by performing a predetermined operation on the session key, the predetermined operation being defined by a private key of the user; and

whereby if the authorisation key from the user matches a corresponding authorisation key generated by the server, access to the secure data is granted.

30. The system according to claim 29, wherein the session key transmission module is a Short Messaging Service Centre (SMSC), GSM modem or GSM phone.

31. The system according to claim 29, wherein the session key receiver module is a mobile communications device of the user.

32. The system according to claim 29, wherein the session key receiver module is a software module installed on a mobile communications device of the user.

33. An authentication server for authenticating a user for accessing secure data, the server comprising:

a session key generation module to generate a session key for associating the user with the server in a session;

a communications module to receive user identification information for identifying the user, to transmit the generated session key via a session key transmission module to the user, and to receive an authorisation key from the user; and

a secure database for storing a copy of a private key of the user;

wherein the server generates an authorisation key by performing a predetermined operation defined by the stored private key of the user on the generated session key; and

whereby if the authorisation key from the user matches the authorisation key generated by the server, access to the secure data is granted.

34. The server according to claim 33, wherein the authorisation key generated by the server is generated when the server receives an authorisation key from the user.

35. The server according to claim 33, wherein the authorisation key generated by the server is generated when the generated session key is transmitted to the user.

36. The server according to claim 33, wherein a session key table is provided to store generated session keys.