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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 April 2007 (19.04.2007)

PCT

(10) International Publication Number
WO 2007/042608 A1

(51) International Patent Classification:
H04L 29/06 (2006.01) *H04L 9/32* (2006.01)

(21) International Application Number: PCT/FI2006/000329

(22) International Filing Date: 10 October 2006 (10.10.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20051023 11 October 2005 (11.10.2005) FI

(71) Applicant *(for all designated States except US)*: **MERIDEA FINANCIAL SOFTWARE OY** [FI/FI]; P.O. Box 63, FI-00381 Helsinki (FI).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **SUOMINEN, Markku** [FI/FI]; 4. Linja 3 B 43, FI-00530 Helsinki (FI). **IKONEN, Kimmo** [FI/FI]; Mäntytie 9 A 5, FI-00270 Helsinki (FI).

(74) Agent: **BERGGREN OY AB**; P.o. Box 16 (annankatu 42 C), FI-00101 Helsinki (FI).

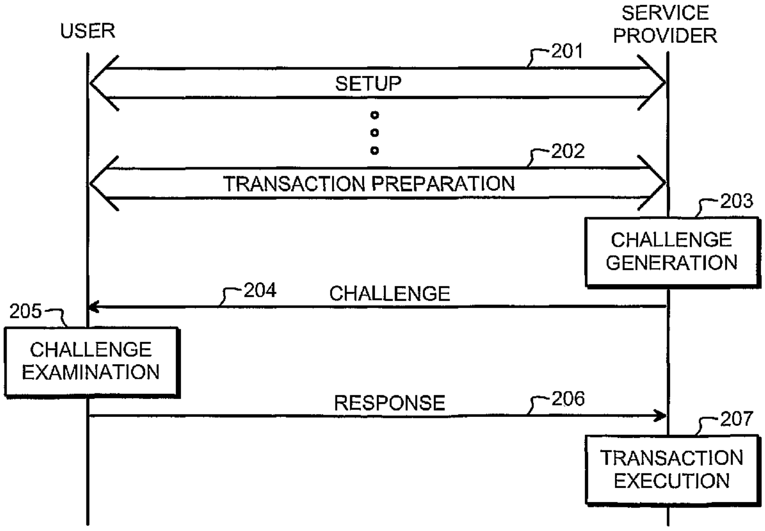
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ,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

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

Published:
— *with international search 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.

(54) Title: METHOD, DEVICES AND ARRANGEMENT FOR AUTHENTICATING A CONNECTION USING A PORTABLE DEVICE



(57) Abstract: A portable electronic device (311) is used for authenticating a connection between a remote computer and a server system (301). The device (311) receives a challenge from the server system (301), reads (511) from the challenge a piece of challenge data that describes a transaction to be executed at the server system (301) and outputs (513) said challenge data to a user. In generating a response to the challenge there is included response data that describes said transaction to be executed at the server system in the response. The response is digitally signed (601) using said response data and a shared secret that is known to the server system (301) and to the portable electronic device (311).

Method, devices and arrangement for authenticating a connection using a portable device

The invention concerns generally the technical field of enhancing the security of a network connection between two communicating electronic devices. Especially the invention concerns the task of executing a transaction over such a connection so that both communicating parties can be sure that they have the same information about the content and outcome of the transaction.

Executing a transaction over a connection between two electronic devices is everyday life in the world that is becoming increasingly dependent on the Internet and other widespread communication networks. A transaction means in this context very generally a series of events in which a piece of initial information is operated upon according to certain input information, resulting in output information. In a simple transaction involving a connection, the user of one device sends commands over the connection to the other device, telling it to make changes to some stored data.

Authentication becomes important if the transaction involves commodities having a value. A dishonest party may perform a so-called man-in-the-middle attack in order to change the course of the transaction to his own benefit. As an example we consider the remote banking transaction illustrated in fig. 1. The user of a remote computer 101 makes a connection to the server 102 of a bank in order to pay an invoice. Somewhere in the middle a dishonest party is operating an intermediate station 103 that can intercept the communications between the remote computer 101 and the server 102. We may assume, for instance, that the dishonest party has previously managed to infiltrate into the remote computer 101 a malicious program that has redirected a bookmark of a browser program so that it points to a web page of the dishonest party instead of the web service of the bank. Said web page of the dishonest party is an exact imitation of the bank's original web service page, so that what the user of the remote computer 101 sees on his screen makes him believe that he is communicating with the bank as usual.

At step 111 the user gives his user ID and password. The intermediate station 103 forwards these at step 112 to the server, which has no reason to believe that the contact would come from someone else than the legitimate user. At step 113 the server responds by opening a confidential service, typically by sending a payment form, which is an HTML (HyperText Markup Language) page that contains input fields. The intermediate station 103 forwards this transmission to the remote computer 101 intact at step 114.

At step 115 the user inserts the payee's account number, the sum payable and other required information to the appropriate fields. At step 116 the remote computer 101 sends the payment information to what the user thinks is the server 102. However, just like the previous transmissions, it goes to the intermediate station 103. Now it does not forward the transmission directly to the server, but makes changes at step 117. Typically the intermediate station changes the payee's account number and possibly also the sum so that instead of the intended payment, a much larger sum is transferred to a completely different account. At step 118 the intermediate station 103 forwards the changed payment information to the server 102, which uses the changed data in effecting the payment at step 119.

It is important to note that the attack illustrated in fig. 1 is possible despite of the use of SSL (Secure Sockets Layer) to secure the connections through the network, and despite of the one-off password system that is otherwise considered to make network payments relatively safe. The success of the man-in-the-middle depends on his ability to redirect the original connection setup appropriately and to thereafter faithfully duplicate all other transmissions than the crucial one containing the payment details. The user will face great difficulty trying to show later that the payment was effected in error, because the bank had all reason to believe that the transaction was proceeding as intended.

Methods and arrangements are known that should make the man-in-the-middle attack impossible. These are usually based on some shared secret that the user has obtained from the service provider through some different channel that is not prone to a similar fraud that allowed the man-in-the-middle to compromise the security of the actual communications channel. For example, suggestions exist that the bank should send a confirmation request to a mobile telephone of the user, and only effect the payment if the user is able to respond appropriately to the confirmation request.

Hardware authentication tokens are also known. These are small electronic devices that comprise a protected memory circuit and some local processing capacity that is capable of performing cryptographic algorithms. A hardware token can be directly connected to a computer e.g. through an USB (Universal Serial Bus) port, and is frequently equipped with a so-called PIN lock (Personal Identification Number), which requires the user to enter a secret PIN each time before the hardware token allows access to its protected memory. If a dishonest party gets hold of the hardware token, he can only try guessing the correct PIN a limited number of times before the hardware device locks itself and prevents all further attempts. The drawback of the hardware token approach is that it requires the user to acquire and maintain a separate device

An objective of the present invention is to present a method, devices and an arrangement for executing a transaction securely through a connection between two communicating electronic devices. A further objective of the invention is to enable secure authentication without requiring the use of protected memory. A yet further objective of the invention is to make authentication widely applicable to different kinds of needs without burdening users with requirements of maintaining specific hardware or performing complicated actions.

The objectives of the invention are achieved with a challenge-response strategy, in which a challenge created by the service provider includes coded information describing the transaction and the user sends a response only after having accepted the information contained in the challenge.

A method for a server system for authenticating a connection between a remote computer and the server system according to the invention is characterized by the features recited in the characterizing part of claim 1.

A method for a portable electronic device for authenticating a connection between a remote computer and a server system according to the invention is characterized by the features recited in the characterizing part of claim 11.

A server system for executing transactions upon instructions received from remote computers over communication connections according to the invention is characterized by the features recited in the characterizing part of claim 19.

A portable electronic device for authenticating a connection between a remote computer and a server system according to the invention is characterized by the features recited in the characterizing part of claim 20.

A computer program product for a portable electronic device for authenticating a connection between a remote computer and a server system according to the invention is characterized by the features recited in the characterizing part of claim 27.

We assume that the user and the service provider have previously ensured that they have a shared secret, for example in the form of a symmetric cryptographic key that they both know, or in the form of an asymmetric key pair, one half of which is in each one's possession. When the service provider is about to effect a transaction, it composes a so-called challenge that includes a brief description of the transaction and most advantageously also a digital signature. The service provider sends the challenge to the user. An electronic device that the user has at his disposal receives the challenge and makes the user aware of the information that describes the transaction. If the user accepts, he expresses his approval to the electronic device, which composes a response and encrypts it by using the shared secret before returning the encrypted response to the service provider. Only if the service provider finds that it can correctly decrypt the response, it allows the transaction to proceed.

For reasons explained in more detail later, the user's electronic device that is responsible for decrypting the challenge, announcing the transaction description to the user and encrypting the response is most advantageously a portable communications device. The way in which the user expresses his approval most advantageously involves giving a passphrase or otherwise proving that the correct user is present.

A man-in-the-middle could have intercepted the previous communications between the user and the service provider and forged some information, for example a payee's account number and a sum. When the service provider composes the challenge, it uses the information it has received, which in that case is the changed data coming from the man-in-the-middle. Without knowing how the service provider's digital signature is produced, the man-in-the-middle cannot forge a digitally signed challenge so that it would again reflect the original, correct information given by the user, even taken the fact that the man-in-the-middle knows what that information originally was. Thus the man-in-the-middle can only let the challenge go to the user unchanged. It is then on the user's responsibility to notice that the information he reads from the challenge does not reflect the transaction in the form he originally meant. The man-in-the-middle is also not able to make up a fake response to the challenge, because he knows neither the shared secret nor the user's passphrase or other personal way of expressing approval.

The exemplary embodiments of the invention presented in this patent application are not to be interpreted to pose limitations to the applicability of the appended claims. The verb "to comprise" is used in this patent application as an open limitation that does not exclude the existence of also unrecited features. The features recited in depending claims are mutually freely combinable unless otherwise explicitly stated.

The novel features which are considered as characteristic of the invention are set forth in particular in the appended claims. The invention itself, however, both as to its construction and its method of operation, together with additional objects and advantages thereof, will be best understood from the following description of specific embodiments when read in connection with the accompanying drawings.

Fig. 1	illustrates the known man-in-the-middle attack,
fig. 2	illustrates the principle of challenge and response,
fig. 3	illustrates a setup phase,

fig. 4 illustrates an alternative key generator,

fig. 5 illustrates challenge generation, transmitting and verification,

fig. 6 illustrates response generation, transmitting and verification,

fig. 7 illustrates features of an exemplary service provider's system,

fig. 8 illustrates features of an exemplary user's device,

fig. 9 illustrates a manual embodiment of challenge handling and response generation,

fig. 10 illustrates an automatic embodiment of challenge handling and response generation,

fig. 11 illustrates one way of making user aware of what he is acknowledging,

fig. 12 illustrates an alternative way of making user aware of what he is acknowledging, and

fig. 13 illustrates yet another way of making user aware of what he is acknowledging.

Fig. 2 illustrates on a general level the sequence of events in a challenge-response strategy according to an embodiment of the invention. At some previous stage, a setup phase 201 has taken place, resulting in a state in which an electronic device of the user and a service provider's system have a shared secret, and the electronic device of the user possesses the cryptographic algorithms and other functionalities that are needed at the later stages of the procedure. The shared secret may consist of a shared symmetric cryptographic key, a shared pair of asymmetric cryptographic keys, a shared secret algorithm or any combination of any numbers of these. At a moment that may take place immediately or even a very long time thereafter (like several years), the user decides to make a transaction in the service provider's system.

The transaction preparation phase 202 may include many kinds of messaging between the user and the service provider. For the purposes of the present invention we only assume that during phase 202 the user sends to the service provider some information about what the desired transaction should involve, and the service provider's system receives that information (or what it believes to be that information). A very simple example of a transaction is logging in to a service, in which case the invention can be applied to make sure that the user has logged in to the service he wanted at the service provider he wanted.

At step 203 the service provider's system generates a so-called challenge. It is a message that contains some description of the transaction the service provider's system is about to perform based on the information it has received during stage 202. Most advantageously generating the challenge at step 203 also involves adding a digital signature to it. At step 204 the service provider's system sends the challenge to the user.

At step 205 an electronic device of the user receives the challenge, checks the signature if any, and announces said description of the transaction to the user for example by displaying it on a display. The user examines the description of the transaction in order to check that it conforms to what he has ordered during stage 202. If it does, the user expresses his acceptance and – still during step 205 – the user's electronic device generates a response, which is cryptographically protected (e.g. digitally signed) utilizing the shared secret. At step 206 the user transmits the response to the service provider's system, which checks on reception thereof that it can correctly open the cryptographic protection of the response utilizing the shared secret. If it can, the service provider's system allows the previously prepared transaction to execute at step 207.

We will next analyze the different actions of fig. 2 in order to explain the most advantageous way of putting the invention into practice. Fig. 3 illustrates schematically certain parts and functionalities of a service provider's system 301 and a user's portable communications device 311, referred to below as the user's device. The parts and functionalities illustrated in fig. 3 are those that have specific importance to performing the setup phase 201 of fig. 2 in an advantageous manner. Fig. 3 can be considered both as illustrating apparatus-type features and as illustrating certain method steps.

The purpose is to establish a shared secret and to equip the user's device 311 with both its copy of the shared secret and the necessary client program it will need in further operation. In fig. 3 we have selected an approach in which the service provider's system generates the shared secret, which imposes a natural additional limitation that it must be delivered to the user's portable communications device 311 as safely as possible. Specifically, we assume that the shared secret is a symmetric encryption key, referred to below as the key.

A key generator 302 in the service provider's system 301 generates the key. The key is stored as such to the service provider's keystore 303, which we assume to be heavily protected against any unauthorized access. The user's copy of the key is not sent to the user's device 311 in one piece but divided into two halves. One of these key halves is "baked into" a client program that will be delivered to the user's terminal. We call it personalizing the client program; in fig. 3 there is illustrated schematically a client program personalizer 304, which receives the appropriate key half from the key generator 302 and uses it to personalize a generic client program read from a client program storage 305. The remaining key half that was not used for client program personalizing can be called the activation code.

The service provider's system 301 comprises first transmitting means 306 and second transmitting means 307 that transmit the activation code and the personalized client program respectively to the user's device 311. The implementation of said transmitter means is not important to the invention but depends merely on the selection of a first channel 321 and a second channel 322 that will be used for transmission. Since the second channel 322 must convey a personalized client program, it must be a channel that is applicable for easily transferring a whole digital file. Typically the second channel 322 involves a wireless data connection, a short-distance data downloading connection (cable, Bluetooth, infrared or the like), a portable memory means, or any combination of these. Since performing the setup phase requires securely authenticating both parties to each other, it is not unreasonable (but also not mandatory, if the required level of security is reached otherwise) to require it take place physically at the premises of the service provider or his authorized representative.

The first channel 321 must only convey a relatively short activation code, typically a character string, which gives more freedom in selecting the type of the channel. The activation code could be shown to the user on a piece of paper or on a screen, or it could be transmitted over the Internet or any other long-distance communications network, or the like. For maintaining the security of the setup phase it is advisable that the first and second channels 321 and 322 differ enough from each other to make it improbable that a dishonest party could have infiltrated both of them simultaneously.

The user's device 311 receives the activation code through first receiving means 312 and the personalized client program through second receiving means 313. Again, the actual implementation of receiving means is not important to the invention and only depends on the selection of the channels. The second receiving means 313 could be for instance a long-distance or short-distance data communications transceiver, and the first receiving means 312 could be for instance a short messaging receiver or even as simple as a keypad through which a user will input an activation code he has seen on paper or on a web page. The user's device stores the received client program into a client program storage 314.

There are basically two options: either the second key half only gets stored along with the personalized client program in the form it was received, or – as we have assumed in fig. 3 – alternatively or additionally the user's device extracts the second key half from the personalized client program and combines it with the first key half (i.e. the activation code) in a key combiner 315 to get the original symmetric encryption key. In the first option mentioned above the key combiner 315 is a triviality and what ends up at the key encrypter 316 is only the activation code. In any case, the key encrypter encrypts the key it received, using a confidential passphrase that the user gives through passphrase input means 317. We should note that block 317 in fig. 3 may physically be the same as e.g. block 312, especially if it is a keypad.

The user's device stores the encrypted key in a key storage 318 and erases all plaintext forms of the key from its memory. The last-mentioned action, as well as the fact that the key storage 318 does not need to be a secure, protected piece of memory, will be discussed in more detail later.

Fig. 4 illustrates an alternative key generator 302' for use in a service provider's system, if asymmetric cryptography is used. The key generator 302' is adapted to generate at least one key pair of an asymmetric cryptographic system. One key of the key pair remains in the keystore of the service provider's system, and the other key of the key pair is further divided into two parts, one of which is transmitted directly to the user while the other is used for personalizing a client program. Neither key of the key pair is made public. It should be noted that asymmetric cryptography arrangements require more memory and processing capacity than symmetric ones, which makes the symmetric cryptography approach of fig. 3 more advantageous.

Fig. 5 illustrates some parts and actions of a service provider's system 301 and a user's device 311 that have significance in the challenge generation step 203, challenge transmission step 204 and first parts of the challenge examination step 205 of fig. 2. We assume that there is a transaction in preparation, and a transaction information source 501 in the service provider's system 301 contains some information that can be used to describe said transaction. For example, if the transaction is a login, the transaction information source 501 may contain the last login time of the same user; if the transaction is a payment that the user is preparing to make, the transaction information source 501 may contain the sum and payee's account number that the service provider's system 301 has received; if the transaction is a stock trading action, the transaction information source 501 may contain the stock id and the number of stocks to be traded. The invention does not limit the types of transaction information that is available in the transaction information source 501.

Conceptually the challenge that is to be generated and transmitted to the user contains a kind of payload field, which can be used for transmitting any kind of short data sequences that describe the transaction. Since at least in some embodiments processing the challenge at the user's end will involve inputting it manually to a portable communications apparatus, it is still advisable to only convey commonly occurring alphanumeric characters in said payload field. In the following we use the designation "challenge data" for the part of transaction information that ends up in the eventual challenge.

Another type of input information that the challenge generator 502 receives is a counter value from a user-specific sequence counter 503. The purpose of using sequence counters at both ends of the connection is to provide additional security against retransmission attacks, in which a dishonest party could have recorded a sequence of previous transmissions and tried to retransmit previously recorded communications from the user to the service provider. Additionally using counter values in the process of digital signing improves the resistance against tampering of the transmitted messages. The counter values are preferably large integer values of 128 bits or more, and preferably proceed in pseudorandom order, so that by knowing one counter value a dishonest party would still have difficulties trying to guess the next counter value. There may be different sequence counters for different kinds of transactions, or a single counter for all operations concerning a particular user. The value of the sequence counter 503 is increased after each use.

The task of initializing the sequence counter 503 and the corresponding counter in the user's device deserves some consideration. The sequence counter program will be delivered to the user's device as a part of the personalized client program. If the counter program itself contained complete initialization routines, even in personalized form so that the initialization value(s) would have been set in the client program personalization stage, if a dishonest party can intercept the transmission of the personalized client program to the user's device he can find out the initialization value and replicate the resulting counter value sequence. It is better that an initialization value for the sequence counter 503 (and indeed for all sequence counters, if more than one are involved) is derived from the activation code that is delivered to the user's device separately from the personalized client program.

According to an advantageous embodiment of the invention the generation of the challenge in the challenge generator 502 involves collecting the challenge data and calculating a digital signature with some suitable one-way hashing algorithm, like the known HMAC (Hash Message Authenticating Code). The input values to the hashing algorithm are at least the challenge data obtained or derived from the contents of the transaction information source 501, and the counter value or a derivative of the counter value obtained from the sequence counter 503. It is possible to use also some secret information obtained from the service provider's keystore 303 as an input to the calculation of the digital signature. For reasons explained in more detail below, it is advantageous in that case not to use the shared secret that in the user's device is behind passphrase-dependent encryption. An exemplary format of the completed challenge could be DDDD DDDD SSSS, in which the D's signify challenge data and the S's convey the signature. The length of a challenge does not need to be fixed – longer challenges may be used to pass more information to the user about the transaction that is taking place.

Separately encrypting the challenge in the challenge generator 502 is generally not required, because we assume that since the transaction is likely to involve confidential matters, all transmissions between the service provider and the user will go through protected channels anyway. For example transmitting over a computer network connection typically involves using SSL, and transmitting over a cellular communications system, if applicable, will utilize the inherent confidentiality features of that system.

The service provider's system 301 transmits the completed challenge to the user's device 311. Several alternatives exist for implementing the transmission. For example, if the user is using a network-connected computer to communicate with the server provider's system, the challenge may be simply displayed to the user through the graphical user interface of his browser program, from which the user must the read the challenge and manually input it into the portable communications device in which the challenge processing will take place. The user's computer may also forward the challenge to the portable communications device through some local short-distance link. In a yet more automatic alternative the service provider's system may transmit the challenge through a cellular communications system to the user's portable communications device. The last-mentioned alternative is specifically advantageous, because it does not require the user to interact at all in the process of communicating the challenge and also because it delivers the challenge through a different communications network than the one through which the other transaction-related communications are taking place.

A challenge verifier 511 receives the challenge in the user's device and recreates the signature using the challenge data and the counter value of the local sequence counter 512 of the user's device. If the signature calculation algorithm involves using some secret information, the challenge verifier 511 receives the appropriate copy or counterpart of that information in the user's device. Like we pointed out earlier, it is most advantageous that no such part of the shared secret is needed that would require the user to input his passphrase for decrypting the shared secret. One reason for this is the relatively low level of security that is required at the mere reception of a challenge: even if the user's device had been stolen in the meantime and the thief could correctly receive a challenge without having to input a passphrase, this would not give him any significant advantage. A second reason is that contradictory as it may seem, requiring a passphrase at this stage would seriously undermine security. Namely, if the user's device really was stolen and the thief was now using it to receive and process an authentic challenge, he could try a brute force attack, i.e. experiment with all possible user's passphrases until he found one with which the challenge signature would match. Having cracked the passphrase this way, the thief could continue using it for other purposes.

One possibility is to use asynchronous cryptography for the challenge signing and challenge signature verifying steps. In such an alternative, the client program that the user obtained from the service provider could contain a public key of the service provider, and the service provider could use the corresponding private key for signing outgoing challenges.

In order to account for possible reasons of slight unsynchronism, it is advisable that if the challenge verifier 511 can not recreate the correct signature with a single try, it experiments with a small number of other counter values that fit within a predetermined window of allowable counter values close to the value tried first. If one of these gives a match, the challenge verifier 511 tells the sequence counter 512 to store that value as the current value. If none of the allowable counter values gives a match, the user's device 311 alerts the user and tells him to contact the service provider in order to find out the reason of unsynchronism.

If the challenge verifier 511 succeeds in recreating the correct signature, it extracts the challenge data and outputs it and/or some more expanded transaction description that it has derived from the challenge data. For example, if the output means 513 is a display and the attempted transaction was a payment, the user's device could display a text "You are about to make a payment to an account number ending with ...6789. The sum payable is in the order of magnitude between 5 and 50 euros. OK to continue?" It is then on the user's responsibility to check that the displayed approximate description of the transaction is in accordance with the input information he has previously given. Here we assume that the challenge data that came in the payload field of the challenge contained at least a transaction type indicator ("payment"), limited account number information ("ending with ...6789") and an indicator of the order of magnitude of the sum payable ("between 5 and 50").

Fig. 6 illustrates some parts and actions of a service provider's system 301 and a user's device 311 that have significance in the later parts of the challenge examination step 205 as well as the response transmission step 206 and transaction execution step 207 of fig. 2. We assume that the user found a challenge to appropriately reflect what he intended, which the user typically first announces by pressing an "OK" button. The user's device 311 begins creating a response in a response generator 601. A response must fulfill certain basic requirements: it must contain some kind of an indication of the challenge it is responding to, and it must be cryptographically processed with a key that only exists in encrypted form in the memory of the user's device. In other words, the service provider's system, which eventually receives the response, must know, who is responding to what. Concerning "manual" embodiments of the invention, in which the user must read the generated response from one device and input it to another, it is also advisable that the response is relatively short, in the order of about a dozen characters at the most.

According to an advantageous embodiment of the invention, the response resembles the challenge in that it contains a payload field and a signature. The payload field may simply contain a replica of the challenge data, although for reasons of semantic consistency it is better to call is response data. The response generator 601 calculates the signature with a one-way hashing algorithm, like HMAC for example. The input values to the hashing algorithm are the contents of the payload field (i.e. the response data), the counter value or a derivative of the counter value obtained from the sequence counter 512, and a decrypted form of the key that was encrypted and stored to the key storage 318 during the setup phase.

Since the key was encrypted using a passphrase given by the user, the user is prompted to give the key decrypter 602 the same passphrase again through passphrase input means 317. An important feature of the invention is that even if the user gives an incorrect passphrase, decrypting the key appears to succeed and the response generator 601 is able to use the result in generating a response that looks as if it was validly composed. The user does not have any means available for checking, whether the passphrase was correct or not, and consequently, whether the response will be accepted in the service provider's system or not.

After the response generator 601 has completed the response, it must be transmitted to the service provider's system 301. In a way similar to the transmission of the challenge, several alternatives are possible. In a manual embodiment of the invention the user's device 311 only displays the completed response on a screen (not shown in fig. 6), and the user reads it and inputs it to a field on a web page he is using to communicate with the service provider's system. If a short-distance local communications link exists between the user's device 311 and the computer displaying said web page, the user's device 311 may utilize it to save the user the effort of manually inputting the response. It is also possible to use the long-distance communications capability (if any) of the user's device 311 to transmit the response to the service provider's system over a cellular communications system or the like. Selecting the way of transmitting the response does not depend on the way in which the challenge came to the user's device 311, but it is natural to think that if automated challenge transmission exists in one way or another, similar automated response transmission is readily available.

Similarly to the case of the challenge, it is not generally required to separately encrypt the response in the response generator 601. If desired, it could be encrypted with a public key of the service provider, obtained from the client program storage 314.

When the response has arrived at the service provider's system 301, it goes to a response verifier 611 that recreates the signature in the response by using the response data, the user's key read from the service provider's keystore 303 and the current counter value from the sequence counter 503. Similarly to the case of the challenge handling, if the signature does not match on the first try, the response verifier 611 may try again with a number of close counter values in case the counters in the user's device 311 and the service provider's system 301 have gone out of synchronism. If one counter value is found to produce a match, it is stored as the current counter value.

If the response verifier 611 is unable to recreate the same signature, there is a reason to suspect that either the user gave an incorrect passphrase by chance or the response did not come from the correct user at all but from a dishonest party that tried to guess the correct passphrase. This is the first step of the process where the correctness of the passphrase is examined. Selecting just the service provider's system and not any part of the user's equipment as the place where the correctness of the passphrase is examined is an important countermeasure against brute force attacking. It also means that protected memory, such as some PIN-locked hardware token, was not needed to store the encrypted key in the user's device, as long as all plaintext copies of the key are always quickly erased from the memory of the user's device. If the user's device 311 had arrived in wrong hands, the dishonest party can at his best try guessing the correct passphrase. Even if the dishonest party knew the key encrypting and decrypting algorithms and could read the encrypted key from the unprotected memory of a stolen user's device, his only chance to verify whether he has given a correct passphrase (and consequently whether he has managed to produce a valid response) is to send the response to the service provider's system. If the last-mentioned does not accept a response, the user or the dishonest party has only a very limited number of times to retry inputting the passphrase. After a small number of consecutive failures (such as three or five) the service provider will disable further attempts, freeze the user's account and notify the user about the need to visit an authorized representative or to otherwise establish a secure state in which the situation can be sorted out.

We may consider a case in which a man-in-the-middle had found out the hash calculation algorithm used for signing, and somehow also the current counter value used in a connection between a particular user and the service provider. We also assume that the man-in-the-middle is trying the conventional attack, so that he had changed some characteristic data of the transaction. At the point of leaving the service provider's system the challenge thus contains challenge data that describes the changed transaction. The man-in-the-middle is now able to change the challenge so that he re-inserts data describing the transaction in its original form and calculates a new signature. The user receives the tampered challenge, checks the challenge data, and accepts, at which point the user's device starts generating the response. This far everything has proceeded as the man-in-the-middle intended. Without knowing the shared secret, however, the man-in-the-middle cannot tamper with the response in the same way he forged the challenge, so he must let the response go through unchanged. It is then on the responsibility of the service provider's system to check that the response data included in the response matches the challenge data, i.e. the response expresses the user's acceptance to exactly the same transaction from which the challenge was generated. If it doesn't, the service provider's system must reject the response even if the signatures matched.

If the response verifier 611 managed to correctly recreate the signature and also found the response data to match, it deduces that the user has accepted the transaction and gives a "green light" signal to a transaction handler 612, meaning that the transaction is allowed to proceed.

The passphrase-dependent encrypting and decrypting used to handle the confidential key or other shared secret in the user's device may be as simple as calculating a one-way hash from the passphrase and performing a logical exclusive-or function bitwise between the passphrase hash and the shared secret. The invention is naturally not limited to that but allows using essentially any kind of reliable cryptographic methods for the same purpose.

Even if the concept of a payload field in the challenge and the response has been introduced above, as a special case the invention allows also using a so-called zero challenge in which the payload field is omitted or filled with a dummy value that does not carry any specific information. This only requires that the challenge signing and response signing algorithms have been so designed that given a dummy payload value or no payload value at all they will operate in some deterministic way, i.e. by taking a default "zero challenge" value as an input.

Fig. 7 illustrates an exemplary arrangement of a service provider's system. As an example we consider that the service provider is a bank. A web banking server 701 acts as an interface between the Internet and the bank's mainframe computers 702. The web banking server 701 contains the server programs and that respond to users' connection requests coming through the Internet, as well as the appropriate web page files. The web banking server 701 is responsible for implementing all operational logic that is behind the web pages and makes the web banking functionality operate as desired. It also makes the necessary protocol conversions and implements SSL security. It may also contain a gateway functionality towards cellular systems, if connections to and/or from users go through a cellular communications system. An example of the last-mentioned is the automated communication of challenges and responses with the users' portable communications devices.

A security server 703 provides the web banking server 701 with the security functions that go beyond standard practices like SSL. In the framework of the present invention, the responsibility of the security server 703 covers key handling, client program personalization, challenge generation upon receiving a triggering command from the web banking server 701, response verification and maintaining of counters. Since especially automated embodiments of the invention may involve direct communication of challenges and responses with the users' portable communications devices, it is possible that such connections originate directly at the security server 703 without going through the web banking server 701. Also the setup phase, in which a personalized client program is delivered to the user, may involve direct communications with the security server 703. The service provider's secure keystore 303 is shown separately in fig. 7.

Fig. 8 illustrates an exemplary user's device, which here is assumed to resemble a so-called smart portable phone. It comprises a cellular transceiver 801 and optionally a short-distance transceiver 802. A central component of the user's device is a processing engine 803. Input and output means consist of a loudspeaker 804, a microphone 805, a display 806 and a keypad 807. In order to operate as a portable phone the device has a number of built-in programs 808 and a protected memory 809. For customization of the functions of the device it comprises space for application programs 810 and some ordinary, unprotected memory 811.

Considering the level of technology at the time of writing this description, an advantageous way of making the user's device operate in a desired way is to load one or more Java midlets into the space reserved for application programs 810. The word midlet (or MIDlet) as such is a general designation for Mobile Information Device Profile application and is understood to mean a piece of program code that can reside in the memory of a mobile station, enter an active state in which it performs certain operations utilizing the resources of the mobile station, and exit into a passive state after it has executed. A more Java-oriented description of a MIDlet is " a set of classes designed to be run and controlled by the application management software via an interface". The present invention is not limited to using midlets in their strictly standardised form. The word "midlet" is used here merely to reflect the fact that at the time of writing this description midlets were the most readily apparent widely known form of reducing the invention into practice.

In the following we assume that a midlet loaded into the user's device can be thought to consist of two functionalities, or alternatively there are two midlets, one of which controls the general operations of receiving and verifying a challenge, displaying its contents to the user, reading the user's reaction and generating and transmitting a response. The other midlet is a specialized cryptographic routine that performs the actual cryptographic operations needed in hash calculations and the like. Together these two midlets (or the dual functionality of a single midlet) constitute the personalized client program that has been loaded to a user's device.

Fig. 9 illustrates a sequence of events in a so-called manual embodiment of the invention. The illustrated parties are the security server, the user, the general operations midlet and the cryptographic midlet. The activity period 901 of the security server begins when it receives from a web banking server (not shown in fig. 9) a request 910 for generating a challenge and some payload information that the challenge should contain. In the following we designate the payload information as TxData. The activity period of the security server lasts until the moment it transmits a positive or negative acknowledgement 950 to the web banking server. The activity periods of the other parties are:

- the user: an activity period 902 from receiving a challenge from the service provider's system to transmitting a response to the service provider's system,

- the general operations midlet: an activity period 903 from receiving a challenge from the user to outputting a response to the user,

- the cryptographic midlet: a first activity period 904 from receiving from the general operations midlet a request for verifying a challenge to outputting the payload information to the general operations midlet, and a second activity period 905 from receiving from the general operations midlet a request for generating a response to outputting the response to the general operations midlet.

The drawing illustrates function calls with solid arrows, responses with dashed arrows and activity periods with hatched rectangles. From the viewpoint of the security server there are only three events: generating a challenge, increasing a counter and getting a response to the challenge. The last-mentioned contains numerous sub-events in the other parties' domain. A pseudocode representation of the sequence of events in fig. 9 is as follows:

```
1: Challenge:=GenerateChallenge(TxData, Counter)
- call to challenge-generating function at 911, said function active during 912
2: increaseCounter()
- call to counter-increasing function at 913, said function active during 914
3: Response:=VerifyChallenge(Challenge)
- challenge transmitted to user at 915
    3.1: Challenge()
    - challenge input to portable device at 916
        3.1.1: TxData:=decodeChallenge(Challenge)
        - call to cryptographic midlet at 917
            3.1.1.1: verifyChallenge(Challenge)
            - call to signature verifying function at 918, said function active
            during 919
            3.1.1.2: TxData:=decode(Challenge)
            - call to payload extracting function at 920, said function active
            during 921
            3.1.1.3: synchronizeChallenge()
            - call to counter synchronization function at 922, said function
            active during 923
        - TxData returned to general operations midlet at 924
        3.1.2: acceptTx(TxData)
        - TxData displayed to user at 925, user checks during 926, user gives
        response at 927
        3.1.3: Passphrase:=getPassphrase()
        - user prompted for passphrase at 928, user thinks during 929, user
        gives passphrase at 930
        3.1.4: Response:=getResponse(Passphrase)
        - call to cryptographic midlet at 931
            3.1.4.1: Key:=decryptKey(Passphrase)
```

- call to key decrypting function at 932, said function active during 933

3.1.4.2: Response:=generateResponse(Challenge,Counter)

- call to response generating function at 934, said function active during 935

3.1.4.3: increaseCounter()

- call to counter-increasing function at 936, said function active during 937

3.1.4.4: eraseDecryptedKey()

- call to key-erasing function at 938, said function active during 939

- completed response returned to general operations midlet at 940

- response displayed to user at 941

- response transmitted to service provider's system at 942.

Fig. 10 illustrates a so-called automated embodiment of the invention, which differs from the manual embodiment of fig. 9 in that the challenge transmission step 1015 involves transmitting the challenge automatically from the service provider's system to the user's device without any user interaction, and that the response transmission step 1041 involves transmitting the response automatically from the user's device to the service provider's system without any user interaction. From the user's viewpoint there is no similar long activity period as in the embodiment of fig. 9, but only the short activity periods of checking the displayed challenge data during 1026 and keying in the passphrase during 1029.

The exemplary embodiments explained above should not be construed to place limitations to only specific embodiments named. For example, even if speaking about a network-connected computer as the apparatus that the user utilizes for connecting to the service provider's system is typically construed to mean a desktop workstation at home or at the office, the same principle can be equally applied for example to automated teller machines, vending machines and other electronic devices that by nature are computers and have a remote connection to some service provider's system. The user's portable device does not need to be a cellular telephone, although at the time of writing this description a cellular telephone is by far the most common portable communications apparatus that people carry around all the time, which makes it a good choice for a user's device because the user would not need to acquire any additional hardware. Another point of the invention where terminology should not be understood to cause unnecessary limitations is the step in which the user's device outputs the challenge data to the user; although we have mainly referred to displaying the challenge data on a display, other forms of making the challenge data available for the human sensory system are possible as well. Giving a passphrase does not necessarily mean pressing some keys in appropriate order. It could refer to other forms of inputting information, including letting the user's device read a biometric identifier of the user.

An important class of further developments of the invention is related to forcing the user to actually notice, what kind of transaction he is acknowledging. Although the purpose of showing the user some information derived from the challenge is to make the user aware of what he is actually acknowledging, it may happen that a very seasoned user who routinely acknowledges transactions this way might hastily press the “OK” button or otherwise express his agreement without actually even taking a look at the displayed information. In such a case, even if the challenge clearly displayed that a man-in-the-middle had tampered with the contents of the transaction, he might succeed in his trick to the detriment of the careless user.

In order to make such an incident less probable there are several possibilities. Fig. 11 illustrates one possible way. At step 1101 the user’s device receives the challenge in one way or another. At step 1102 the user’s device generates at least one piece of fake challenge data, meaning some challenge data that to a user appears to describe a transaction with different contents than the intended transaction. At step 1103 the user’s device presents both the challenge data of the correct challenge and the fake challenge data to the user, preferably in some random order so that the user does not immediately know, which presented challenge data corresponds to the correct transaction. The user is now forced to familiarize himself with the presented challenge data at least in sufficient detail to notice, which challenge data represents the correct transaction that he actually intended to make. At step 1104 the user expresses acknowledgement with the selected challenge data. At step 1105 the user's device checks, whether the user acknowledged the challenge data of the genuine challenge or that of a fake. If the user made a wrong selection, he is given a chance to take corrective action at step 1106. If he fails to do so, the procedure is halted at step 1108 and the user's device will not transmit the correct challenge response. A correct selection by the user either directly at step 1105 or as corrective action at step 1106 makes the user's device respond normally to the challenge according to step 1107.

Fig. 12 illustrates an alternative way. At step 1201 the server embeds in the challenge a certain checksum (or more generally: a piece of verification information) that it derived from some information that describes the transaction. At step 1202 the user's device receives the challenge and decodes the checksum as a part of the other challenge data. At step 1203 the user's device prompts the user to enter some data that the user knows about the transaction and that corresponds with the information from which the server derived the checksum mentioned above. For example the whole receiving account number or the exact amount of money transferred may be required to be entered in the user's device. At step 1204 the user enters the requested data, and at step 1205 the user's device checks the entered data by calculating from it a checksum in the same way as the server did and comparing the checksums. If the checksums do not match, the user is given a chance for corrective action at step 1206. Failing to correct leads to halting the challenge-handling procedure at step 1208, while entering correctly matching information at either step 1205 or step 1206 makes the user's device respond normally to the challenge according to step 1207.

Fig. 13 illustrates yet another alternative way. This time the data, which the user is required to enter, is something that the user's device knows already from the ordinary contents of the challenge. Having received and decoded the challenge at steps 1301 and 1302 the user's device prompts the user to enter data at step 1303. The required data may be available for the user for example on the very display screen of the user's data on which also the prompt is given, like "You are about to pay a sum of 52 euros to an account ending with ...6789. Confirm by re-entering the sum." Again, when the user has entered the prompted matching piece of challenge data at step 1304, there is made a check at step 1305, from which the process may continue to step 1307 directly or through corrective action at step 1306; and failure leads to not responding to the challenge according to step 1308.

In all cases above where we have referred to not sending the (correct) challenge response to the server (because the user did not properly express his acceptance), a natural alternative is to send a challenge response to the server but to compose the response so that it informs the server about the user not having properly expressed his approval. We may use the designation "positive response" to describe a challenge response that allows the server to proceed executing the transaction, and the designation "negative response" to describe a challenge response that tells the server not to proceed.

Advantages of the invention include (but are not limited to) the following:

- there is no need for tamper resistant / protected key repository on the user's device

- both ends of the connection can be sure about the "legitimacy" of the other party

- payload information can be carried both ways in the challenge and the response and can not be forged.

Claims

1. A method for a server system for authenticating a connection between a remote computer and the server system, comprising:

- exchanging information between the remote computer and the server system concerning a transaction to be executed at the server system

- generating a challenge at the server system and transmitting the challenge, and

- receiving a response to the challenge at the server system,

characterized in that:

- generating the challenge comprises including information that describes the transaction to be executed as challenge data to the challenge,

- receiving the response at the server system comprises reading response data and a digital signature from the response, recreating the digital signature using said response data and a shared secret that is known to the server system and to an electronic device of the user of the remote computer and checking whether the digital signature read from the response is the same as the recreated digital signature, and

- the method comprises requiring the digital signature read from the response to be the same as the recreated digital signature for the response to be acceptable.

2. A method according to claim 1, **characterized** in that generating the challenge comprises digitally signing the challenge using said challenge data and secret information for challenges that is known to the server system and to said electronic device of the user of the remote computer.

3. A method according to claim 2, **characterized** in that digitally signing the challenge comprises using a counter value of a sequence counter as an input for said digital signing.

4. A method according to claim 1, **characterized** in that recreating the digital signature of a received response comprises using a counter value of a sequence counter as an input for said recreating of the digital signature.

5. A method according to claim 4, **characterized** in that as a response to a recreated digital signature not being the same as the digital signature read from the response in a first attempt, the method comprises repeatedly selecting other counter values that are not further away from the counter value used for the first attempt than a predetermined width of a counter window, and repeatedly recreating the digital signature using said response data, said shared secret and each selected counter value in turn until either a digital signature recreated with one selected counter value is the same as the digital signature read from the response or other counter values have been tried that are not further away from the counter value used for the first attempt than said width of said counter window.

6. A method according to claim 1, **characterized** in that before exchanging information between the remote computer and the server system concerning the transaction to be executed at the server system it comprises:

- generating and storing a shared secret within the server system,

- delivering a first part of the generated shared secret to said electronic device of the user of the remote computer and

- personalizing a client program by including a second part of the generated shared secret, different than said first part, into said client program, and delivering the personalized client program to said electronic device of the user of the remote computer through a different delivery channel than what was used for delivering the first part of the generated shared secret.

7. A method according to claim 6, **characterized** in that it additionally comprises generating and storing a piece of secret information for challenges and delivering said secret information for challenges to said electronic device of the user of the remote computer together with the personalized client program.

8. A method according to claim 1, **characterized** in that transmitting the challenge comprises transmitting the challenge to the remote computer.

9. A method according to claim 1, **characterized** in that transmitting the challenge comprises transmitting the challenge to a portable communications device of the user of the remote computer through a cellular communications network.

10. A method according to claim 1, **characterized** in that the method comprises only accepting the response if the digital signature read from the response is the same as the recreated digital signature and the response data matches the challenge data.

11. A method for a portable electronic device for authenticating a connection between a remote computer and a server system, comprising:

- receiving a challenge from the server system, and

- generating a response to the challenge for transmission to the server system,

characterized in that:

- receiving the challenge comprises reading from the challenge a piece of challenge data that describes a transaction to be executed at the server system and outputting said challenge data to a user, and

- generating the response comprises including response data that describes said transaction to be executed at the server system in the response and digitally signing the response using said response data and a shared secret that is known to the server system and to the portable electronic device.

12. A method according to claim 11, **characterized** in that digitally signing the response comprises additionally using a counter value of a sequence counter as an input for said digital signing.

13. A method according to claim 11, **characterized** in that receiving the challenge comprises reading a digital signature from the challenge, recreating said

digital signature using said challenge data and secret information for challenges that is known to the server system and to the portable electronic device and checking whether the digital signature read from the challenge is the same as the recreated digital signature, and

- the method comprises requiring the digital signature read from the challenge to be the same as the recreated digital signature for the challenge to be acceptable.

14. A method according to claim 13, **characterized** in that recreating the digital signature comprises using a counter value of a sequence counter as an input for said recreating of the digital signature.

15. A method according to claim 14, **characterized** in that as a response to a recreated digital signature not being the same as the digital signature read from the challenge in a first attempt, the method comprises repeatedly selecting other counter values that are not further away from the counter value used for the first attempt than a predetermined width of a counter window, and repeatedly recreating the digital signature using said challenge data and each selected counter value in turn until either a digital signature recreated with one selected counter value is the same as the digital signature read from the challenge or other counter values have been tried that are not further away from the counter value used for the first attempt than said width of said counter window.

16. A method according to claim 11, **characterized** in that digitally signing the response comprises receiving a passphrase from a user, using the received passphrase for decrypting the shared secret from an encrypted form that is the only form in which the shared secret is known to the portable electronic device, and erasing all decrypted forms of the shared secret after digitally signing the response.

17. A method according to claim 11, **characterized** in that before receiving a challenge from the server system it comprises:

- receiving a first part of a shared secret originating from the server system,

- receiving a personalized client program from the server system through a different delivery channel than what was used for receiving the first part of the shared secret, said personalized client program containing a second part of the shared secret, different than said first part,

- receiving a passphrase from a user,

- using the received passphrase for encrypting at least said first part of the shared secret and

- erasing from the portable electronic device all decrypted forms of those parts of the shared secret that were encrypted using the received passphrase.

18. A method according to claim 17, **characterized** in that it comprises combining the first part of the shared secret and the second part of the shared secret before encrypting, so that using the received passphrase for encrypting comprises encrypting a combination of the first and second parts of the shared secret.

19. A method according to claim 11, **characterized** in that it comprises:

- presenting to the user a piece of correct challenge data read from the challenge and a piece of fake challenge data, and

- proceeding to generating a positive response that includes response data that describes said transaction to be executed at the server system only in response to receiving from the user an input indicating selection of said correct challenge data.

20. A method according to claim 11, **characterized** in that it comprises:

- decoding from a received challenge a first piece of verification information associated with the transaction to be executed at the server system,

- receiving from the user an input related to the transaction to be executed at the server system, and deriving a second piece of verification information from said input,

- comparing said second piece of verification information to said first piece of verification information, and

- proceeding to generating a positive response that includes response data that describes said transaction to be executed at the server system only in response to said second piece of verification information matching said first piece of verification information.

21. A method according to claim 11, **characterized** in that it comprises:

- presenting the user with a first piece of challenge data read from the challenge, and prompting the user to enter a matching piece of challenge data, and

- proceeding to generating a positive response that includes response data that describes said transaction to be executed at the server system only in response to the user entering the prompted matching piece of challenge data.

22. A server system for executing transactions upon instructions received from remote computers over communication connections, the server system comprising:

- a transceiver adapted to exchange information between a remote computer and the server system concerning a transaction to be executed at the server system

- a challenge generator adapted to generate a challenge at the server system and to transmit the challenge, and

- a response verifier adapted to receive a response to the challenge at the server system,

characterized in that:

- the challenge generator comprises an input coupled to a transaction information source for including information that describes the transaction to be executed as challenge data to the challenge,

- the response verifier is adapted to read response data and a digital signature from the response, to recreate the digital signature using said response data and a shared secret that is known to the server system and to an electronic device of the user of the remote computer, and to check whether the digital signature read from the response is the same as the recreated digital signature, and

- the server system is adapted to require the digital signature read from the response to be the same as the recreated digital signature for the response to be acceptable.

23. A portable electronic device for authenticating a connection between a remote computer and a server system, comprising:

- a challenge verifier adapted to receive a challenge from the server system, and

- a response generator adapted to generate a response to the challenge for transmission to the server system,

characterized in that:

- the challenge verifier is adapted to read from the challenge a piece of challenge data that describes a transaction to be executed at the server system,

- the portable electronic device comprises output means coupled to said challenge verifier for outputting said challenge data to a user, and

- the response generator is adapted to include response data that describes said transaction to be executed at the server system in the response and to digitally sign the response using said response data and a shared secret that is known to the server system and to the portable electronic device.

24. A portable electronic device according to claim 23, **characterized** in that it comprises a sequence counter coupled to the response generator, and the response generator is adapted to additionally use a counter value obtained from said sequence counter as an input for said digital signing.

25. A portable electronic device according to claim 23, **characterized** in that the challenge verifier is adapted to read a digital signature from the challenge, recreate said digital signature using said challenge data and secret information for

challenges that is known to the server system and to the portable electronic device, and check whether the digital signature read from the challenge is the same as the recreated digital signature, and

- the portable electronic device is adapted to require the digital signature read from the challenge to be the same as the recreated digital signature for the challenge to be acceptable.

26. A portable electronic device according to claim 25, **characterized** in that it comprises a sequence counter coupled to said challenge verifier, and the challenge verifier is adapted to use a counter value obtained from said sequence counter as an input for said recreating of the digital signature.

27. A portable electronic device according to claim 23, **characterized** in that the electronic device is adapted to receive a passphrase from a user and to use the received passphrase for decrypting the shared secret from an encrypted form that is the only form in which the shared secret is known to the portable electronic device, and the portable device is adapted to erase all decrypted forms of the shared secret after digitally signing the response.

28. A portable electronic device according to claim 23, **characterized** in that in order to receive shared secrets from a server system it comprises:

- a first receiving means for receiving a first part of a shared secret originating from the server system,

- a second receiving means for receiving a personalized client program from the server system through a different delivery channel than what was used for receiving the first part of the shared secret,

- passphrase input means for receiving a passphrase from a user,

- an encrypter adapted to use the received passphrase for encrypting at least said first part of the shared secret and

- erasing means adapted to erase from the portable electronic device all decrypted forms of those parts of the shared secret that were encrypted using the received passphrase.

29. A portable electronic device according to claim 28, **characterized** in that it comprises a combiner adapted to combine a first part of the shared secret and a second part of the shared secret before encrypting, and the encrypter is adapted to receive a combination of the first and second parts of the shared secret from said combiner.

30. A computer program product for a portable electronic device for authenticating a connection between a remote computer and a server system, comprising:

- computer program means that, when loaded into a computer, cause the computer to receive a challenge from the server system, and

- computer program means that, when loaded into a computer, cause the computer to generate a response to the challenge for transmission to the server system,

characterized in that it comprises:

- computer program means that, when loaded into a computer, cause the computer to read from the challenge a piece of challenge data that describes a transaction to be executed at the server system and output said challenge data to a user, and

- computer program means that, when loaded into a computer, cause the computer to include response data that describes said transaction to be executed at the server system in the response and digitally sign the response using said response data and a shared secret that is known to the server system and to the portable electronic device.

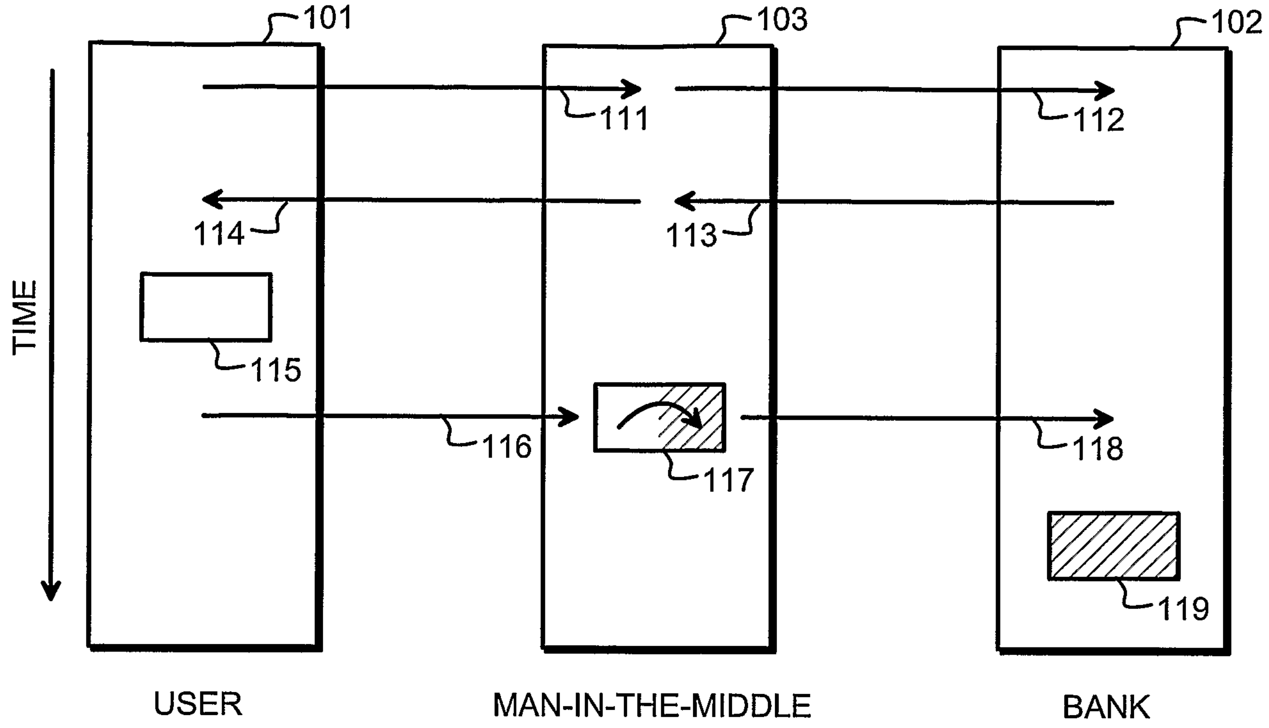


Fig. 1
PRIOR ART

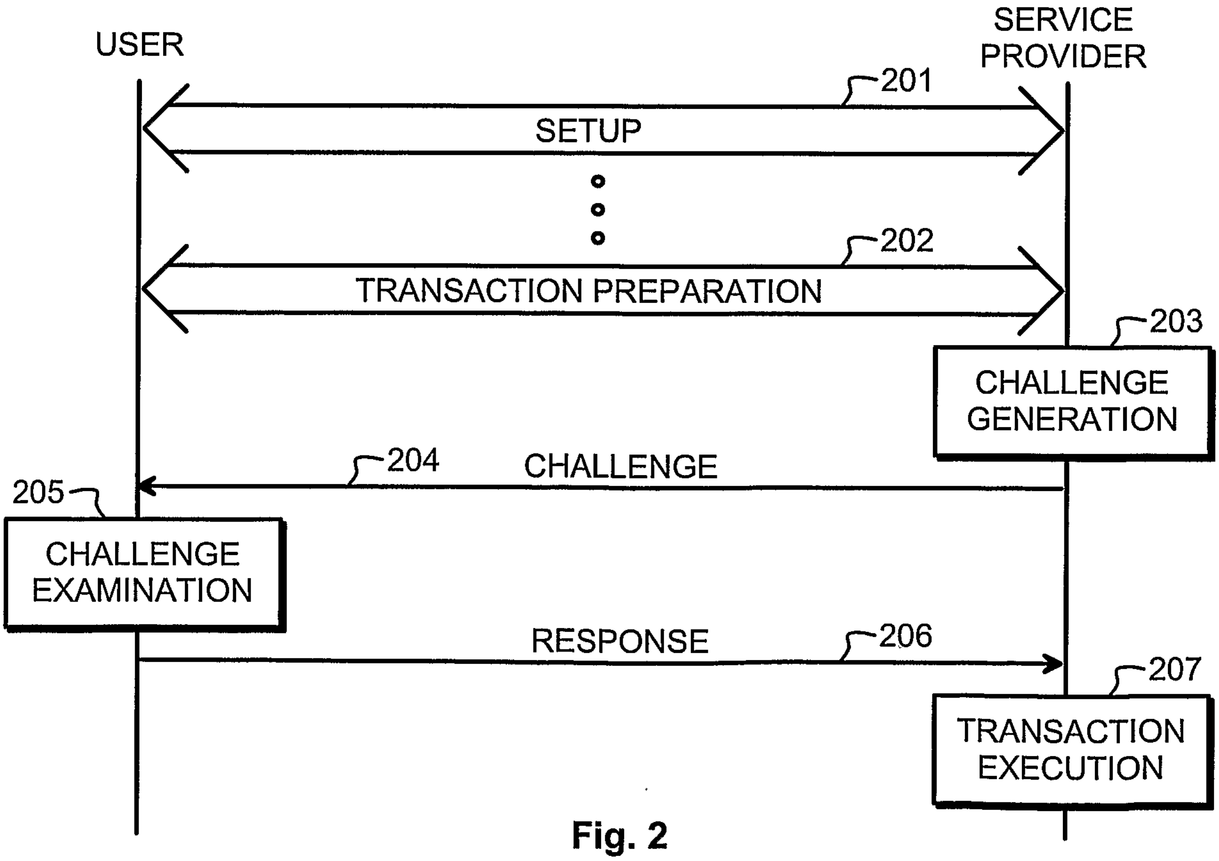


Fig. 2

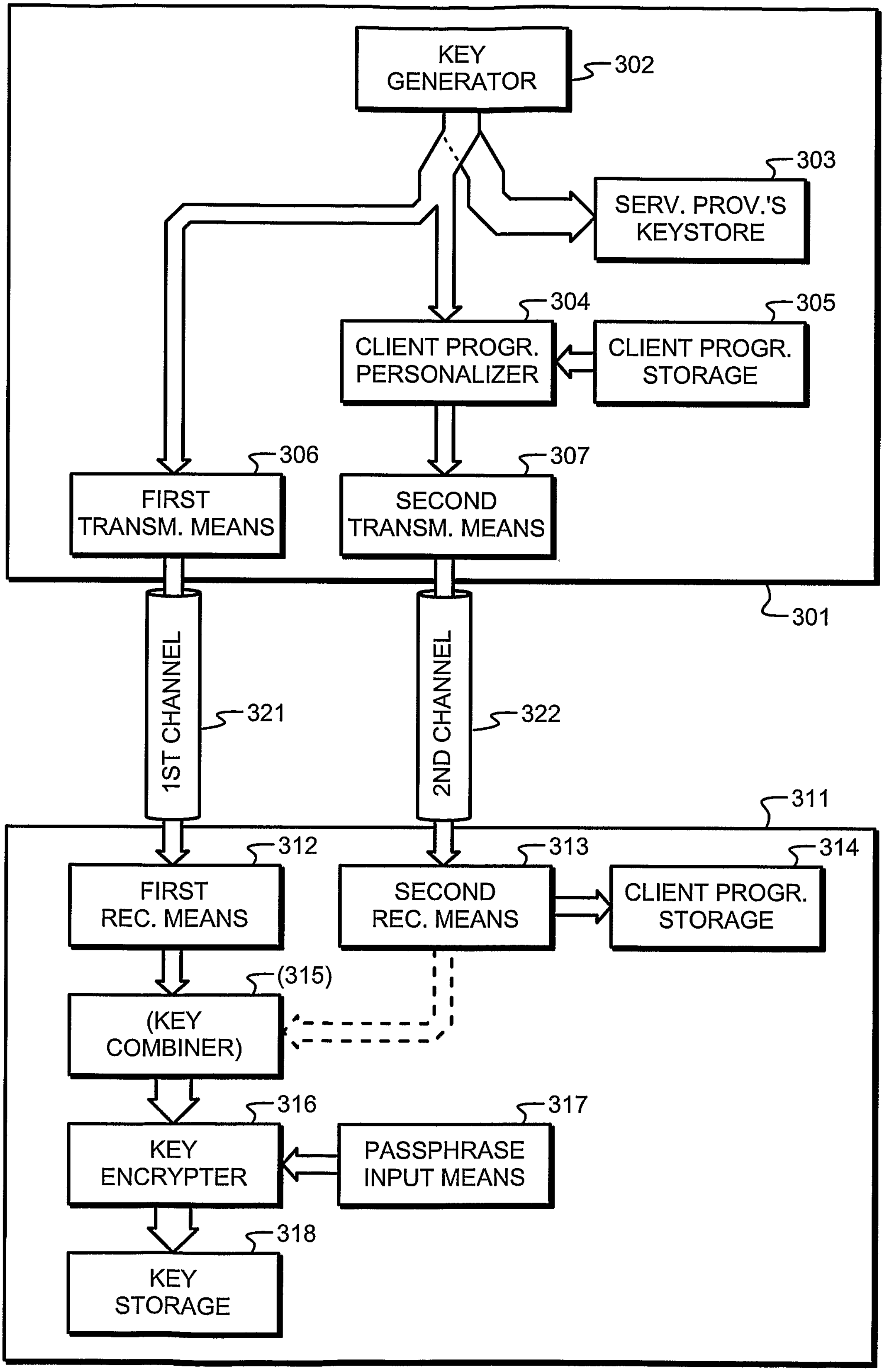

KEY GENERATOR
302
SERV. PROV.'S KEYSTORE
303
CLIENT PROGR. PERSONALIZER
304
CLIENT PROGR. STORAGE
305
FIRST TRANSM. MEANS
306
SECOND TRANSM. MEANS
307
301
1ST CHANNEL
321
2ND CHANNEL
322
311
FIRST REC. MEANS
312
SECOND REC. MEANS
313
CLIENT PROGR. STORAGE
314
(KEY COMBINER)
(315)
KEY ENCRYPTER
316
PASSPHRASE INPUT MEANS
317
KEY STORAGE
318

Fig. 3

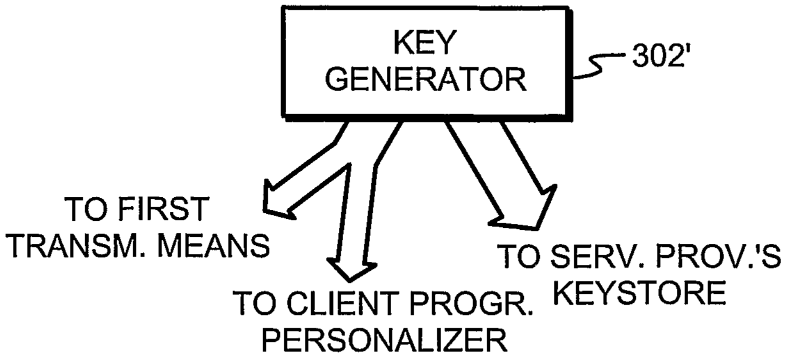

KEY GENERATOR
302'
TO FIRST TRANSM. MEANS
TO CLIENT PROGR. PERSONALIZER
TO SERV. PROV.'S KEYSTORE

Fig. 4

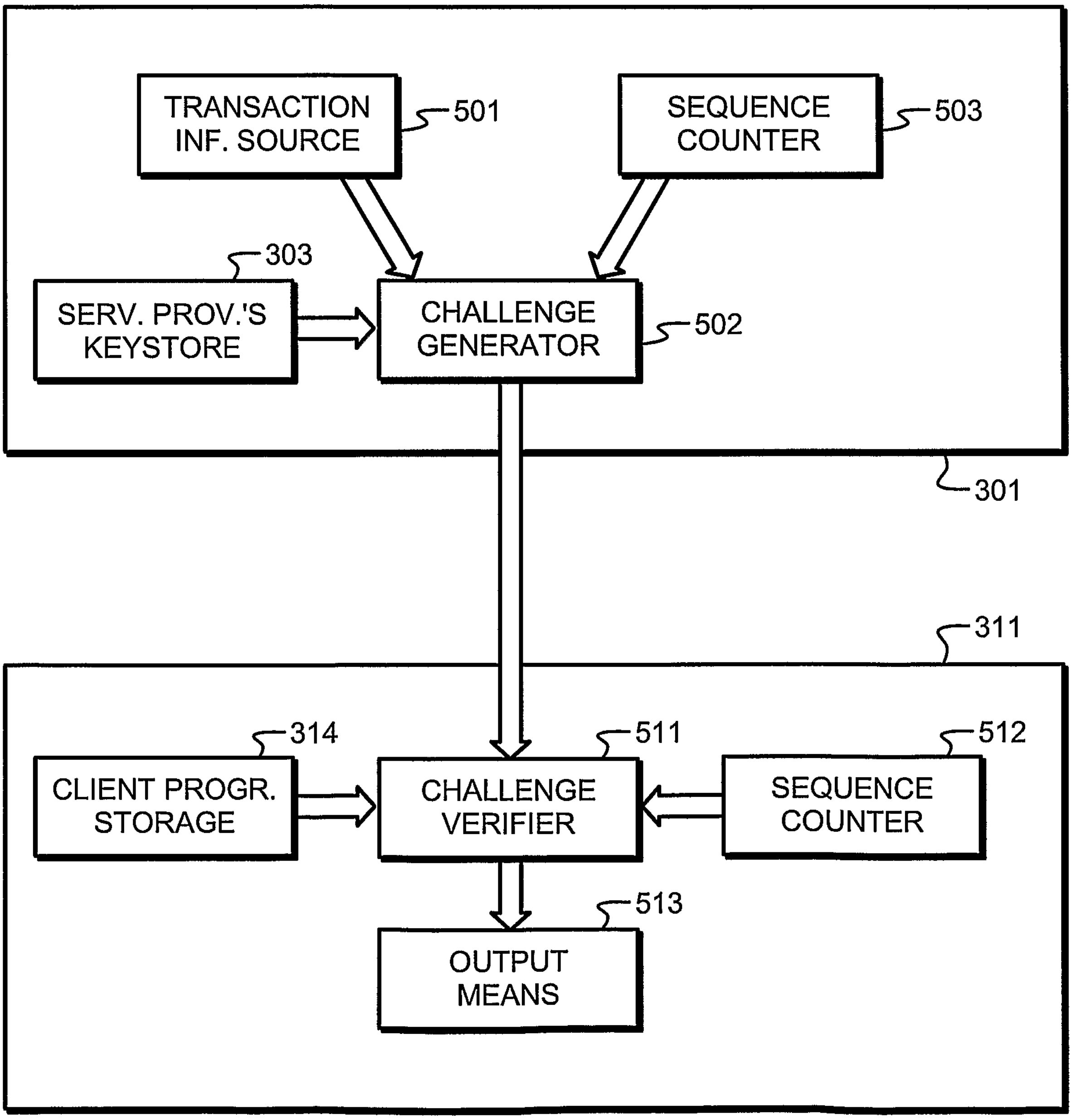

TRANSACTION INF. SOURCE
501
SEQUENCE COUNTER
503
303
SERV. PROV.'S KEYSTORE
CHALLENGE GENERATOR
502
301
311
314
CLIENT PROGR. STORAGE
511
CHALLENGE VERIFIER
512
SEQUENCE COUNTER
513
OUTPUT MEANS

Fig. 5

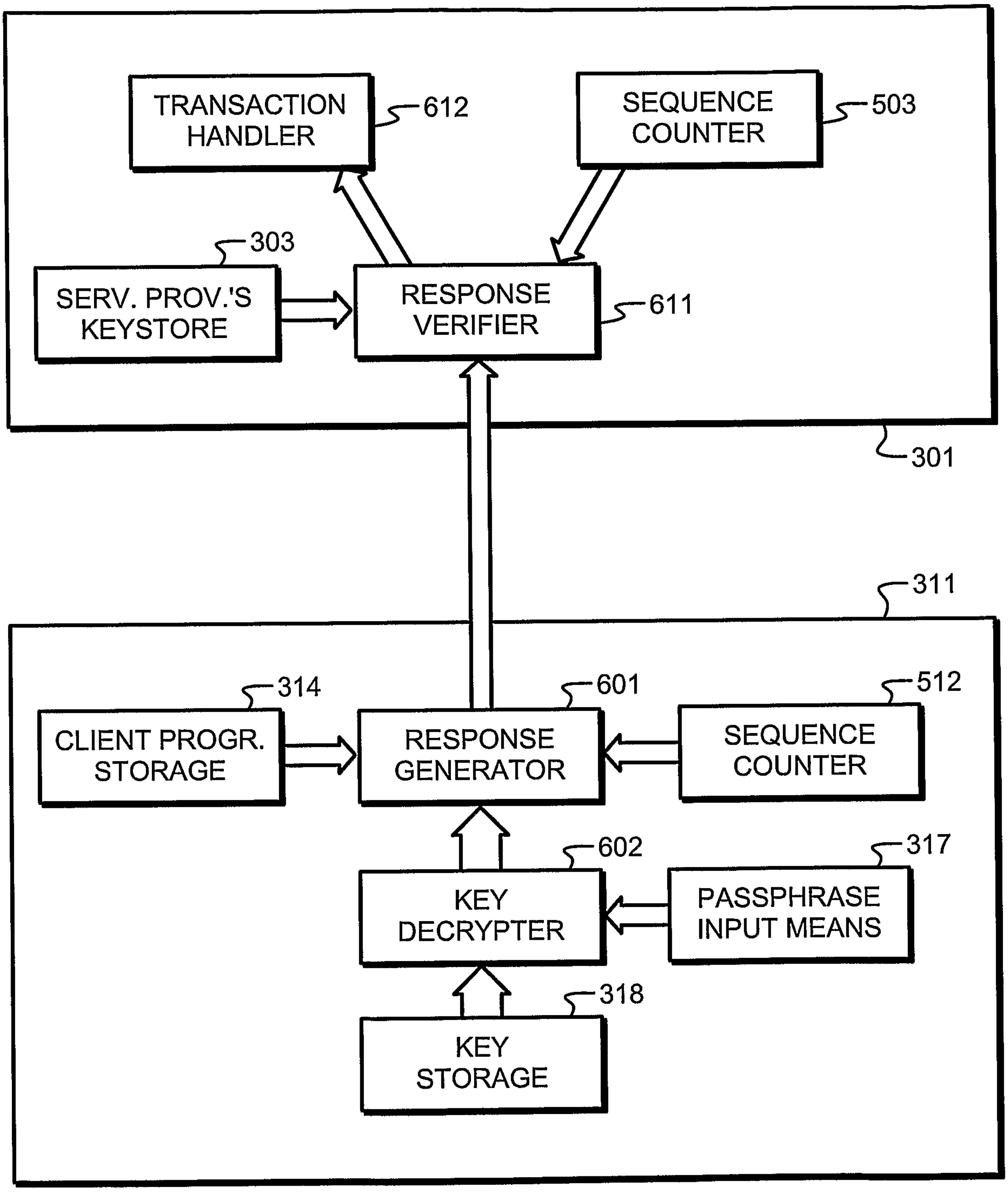

TRANSACTION HANDLER
612
SEQUENCE COUNTER
503
303
SERV. PROV.'S KEYSTORE
RESPONSE VERIFIER
611
301
311
314
CLIENT PROGR. STORAGE
601
RESPONSE GENERATOR
512
SEQUENCE COUNTER
602
KEY DECRYPTER
317
PASSPHRASE INPUT MEANS
318
KEY STORAGE

Fig. 6

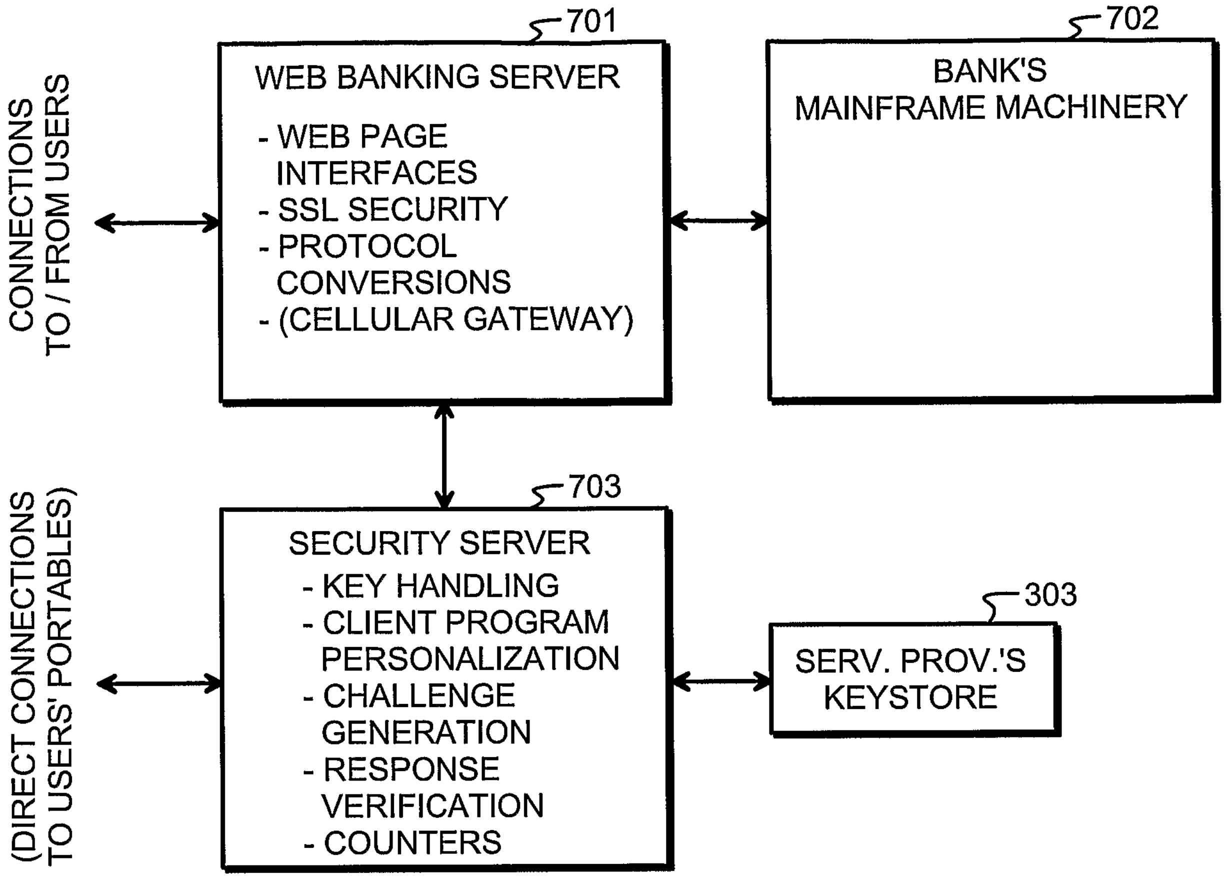

701
WEB BANKING SERVER
- WEB PAGE INTERFACES
- SSL SECURITY
- PROTOCOL CONVERSIONS
- (CELLULAR GATEWAY)
702
BANK'S MAINFRAME MACHINERY
CONNECTIONS TO / FROM USERS
703
SECURITY SERVER
- KEY HANDLING
- CLIENT PROGRAM PERSONALIZATION
- CHALLENGE GENERATION
- RESPONSE VERIFICATION
- COUNTERS
(DIRECT CONNECTIONS TO USERS' PORTABLES)
303
SERV. PROV.'S KEYSTORE

Fig. 7

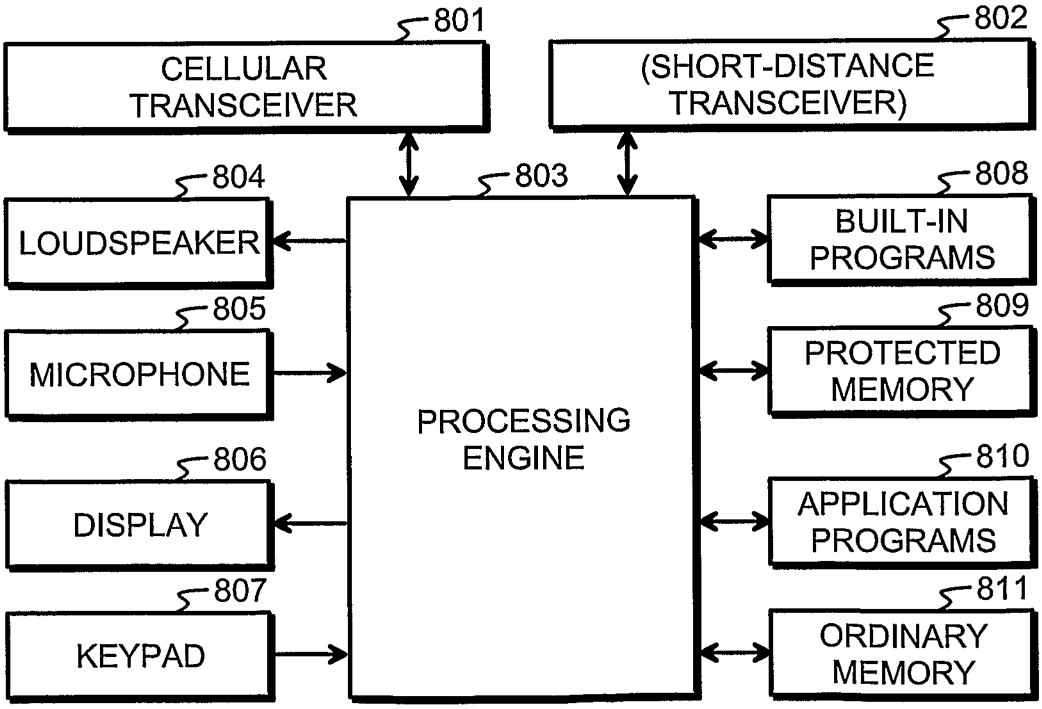

801
CELLULAR TRANSCEIVER
802
(SHORT-DISTANCE TRANSCEIVER)
803
PROCESSING ENGINE
804
LOUDSPEAKER
805
MICROPHONE
806
DISPLAY
807
KEYPAD
808
BUILT-IN PROGRAMS
809
PROTECTED MEMORY
810
APPLICATION PROGRAMS
811
ORDINARY MEMORY

Fig. 8

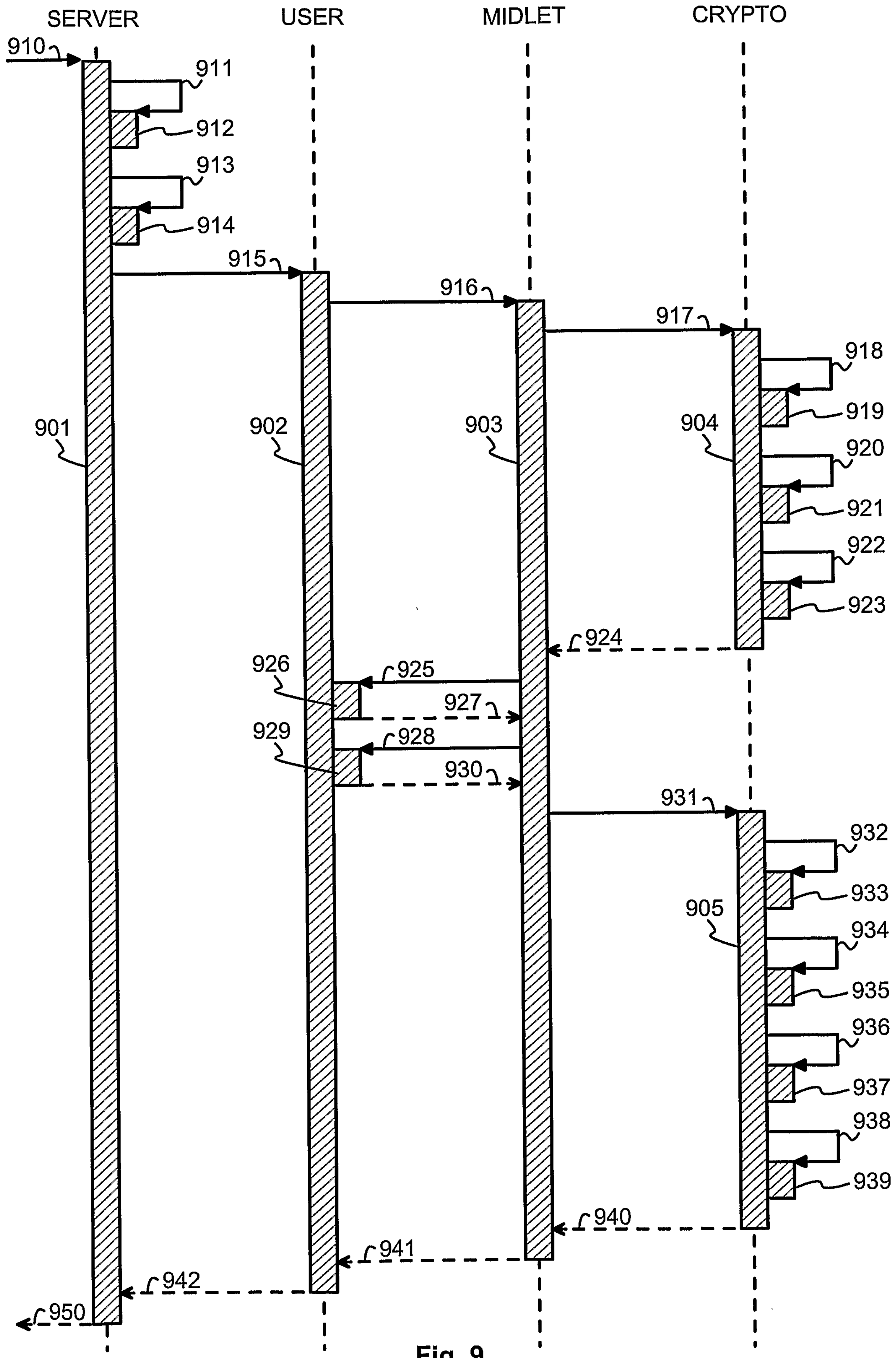

SERVER
USER
MIDLET
CRYPTO
910
911
912
913
914
915
916
917
918
919
920
921
922
923
901
902
903
904
924
925
926
927
928
929
930
931
932
933
905
934
935
936
937
938
939
940
941
942
950

Fig. 9

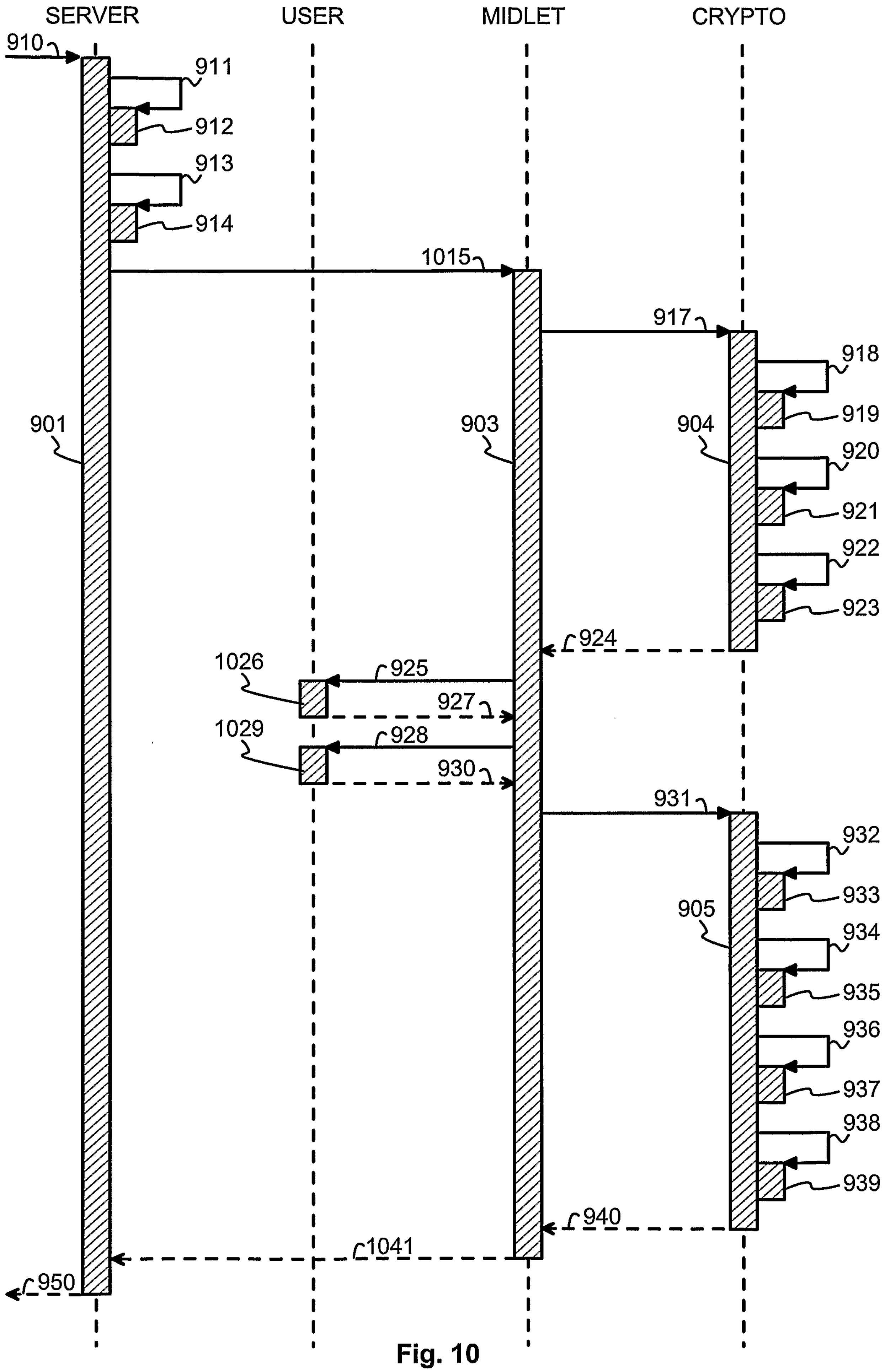

SERVER
USER
MIDLET
CRYPTO
910
911
912
913
914
1015
917
918
919
920
921
922
923
901
903
904
924
1026
925
927
1029
928
930
931
932
933
934
935
905
936
937
938
939
940
1041
950

Fig. 10

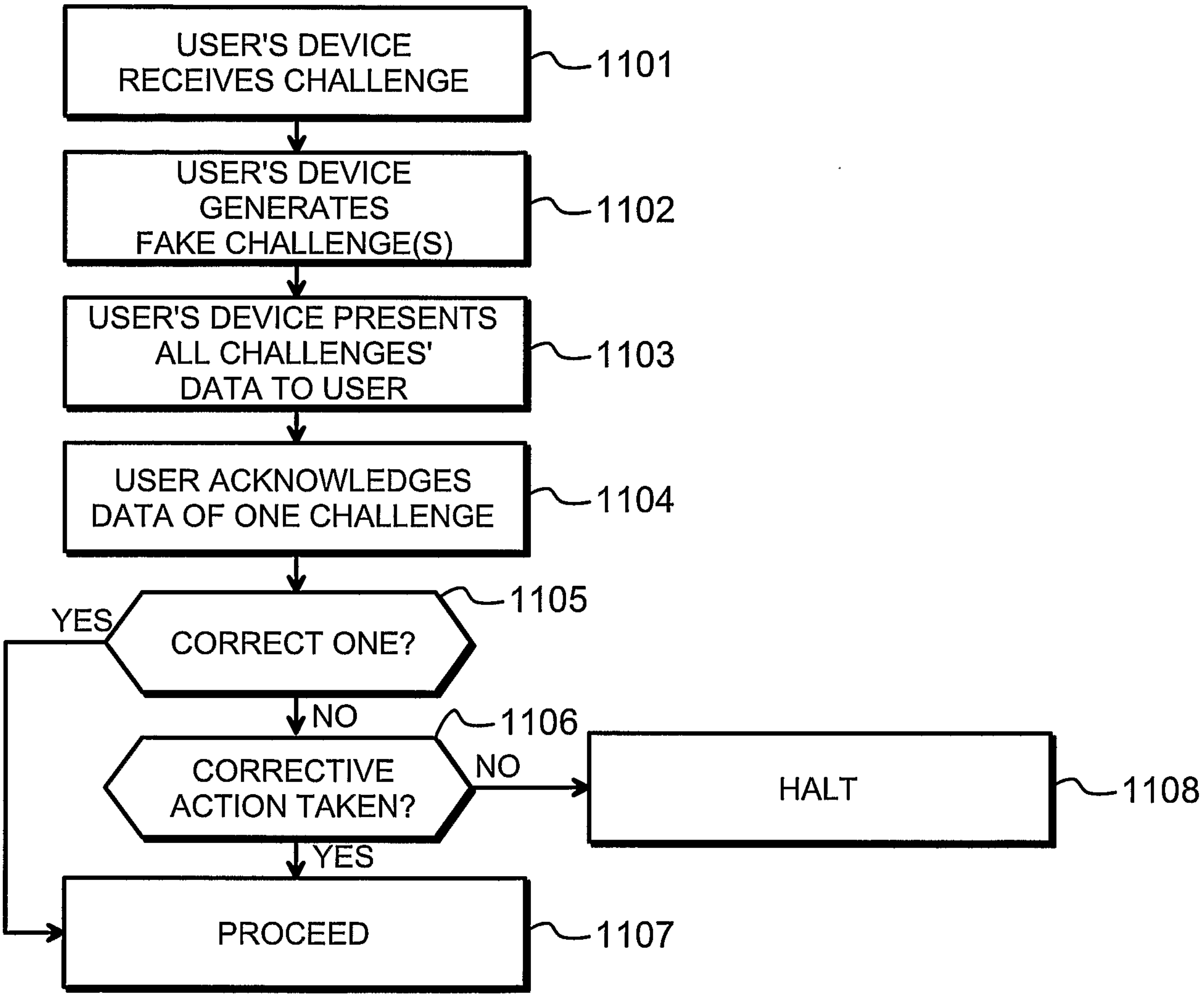

USER'S DEVICE RECEIVES CHALLENGE
1101
USER'S DEVICE GENERATES FAKE CHALLENGE(S)
1102
USER'S DEVICE PRESENTS ALL CHALLENGES' DATA TO USER
1103
USER ACKNOWLEDGES DATA OF ONE CHALLENGE
1104
CORRECT ONE?
1105
YES
NO
CORRECTIVE ACTION TAKEN?
1106
NO
HALT
1108
YES
PROCEED
1107

Fig. 11

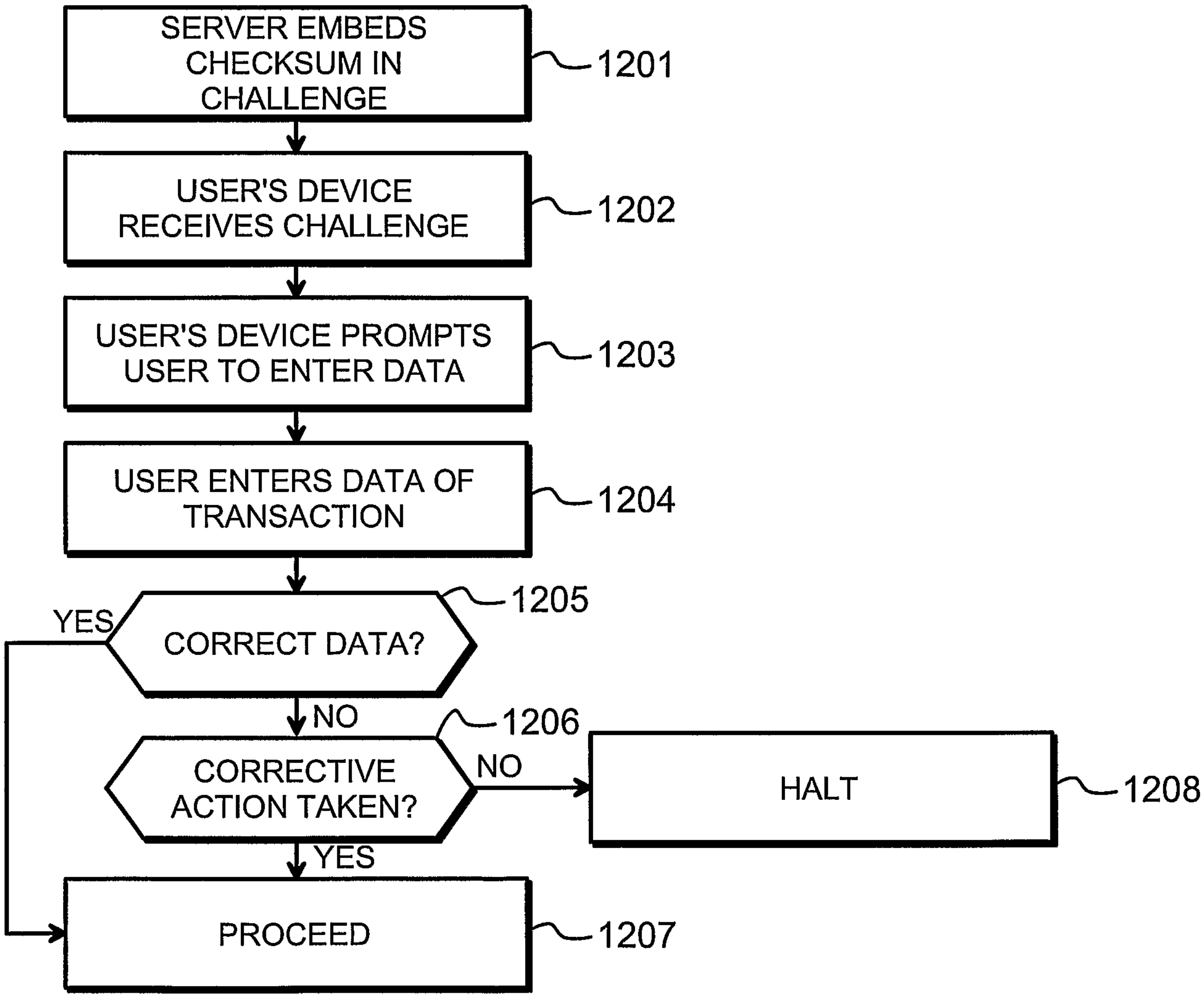

SERVER EMBEDS CHECKSUM IN CHALLENGE
1201
USER'S DEVICE RECEIVES CHALLENGE
1202
USER'S DEVICE PROMPTS USER TO ENTER DATA
1203
USER ENTERS DATA OF TRANSACTION
1204
CORRECT DATA?
1205
YES
NO
CORRECTIVE ACTION TAKEN?
1206
NO
HALT
1208
YES
PROCEED
1207

Fig. 12

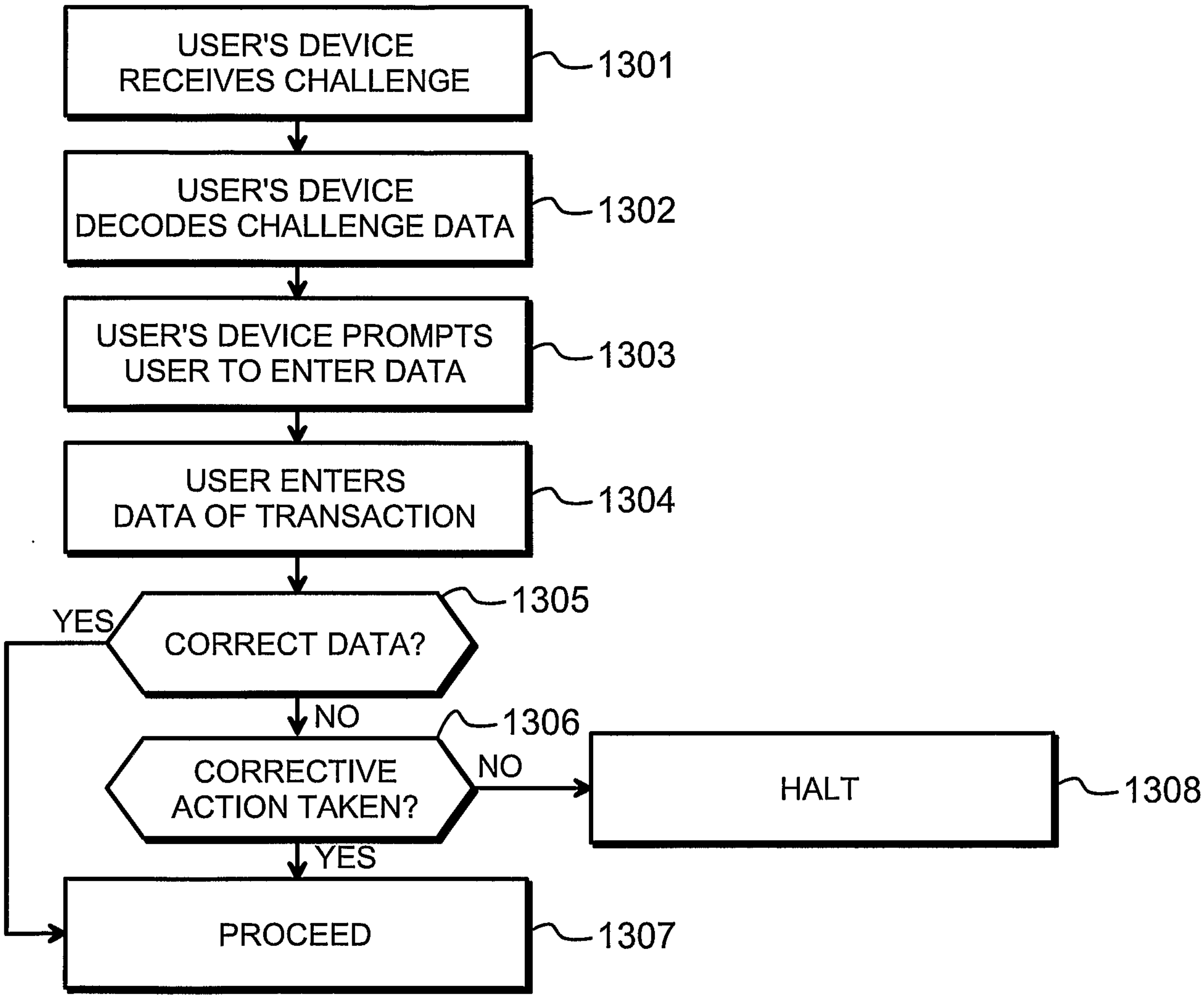

USER'S DEVICE RECEIVES CHALLENGE
1301
USER'S DEVICE DECODES CHALLENGE DATA
1302
USER'S DEVICE PROMPTS USER TO ENTER DATA
1303
USER ENTERS DATA OF TRANSACTION
1304
CORRECT DATA?
1305
YES
NO
CORRECTIVE ACTION TAKEN?
1306
NO
HALT
1308
YES
PROCEED
1307

Fig. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2006/000329

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 8: H04L, G06F, G06Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

FI, SE, NO, DK

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI, Inspec

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	US 2004/0139013 A1 (BARBIER ERIC et al.) 15 July 2004 (15.07.2004)	
A		3-7, 12, 14-21, 24-29
X	whole document; in particular: figures 1-3,11;[0017];[0025];[0103]; claims 34 and 37	1, 2, 8-11, 13, 22, 23, 30
	WO 03/049364 A1 (CONCEPTM COMPANY LTD et al.) 12 June 2003 (12.06.2003)	
A		3-7, 12, 14-21, 24-29
X	claim 1 (item c-iii, in particular); page 26, line 31-page 37, line 17	1, 2, 8-11, 13, 22, 23, 30
A	TAN, T. K., 'Phishing Redefined? Preventing Man-in-the-Middle Attacks for Web-based Transactions', www.dsssasia.com (public 15.5.2005, http://web.archive.org/web/*/http://www.dsssasia.com/htmdocs/company/news_events/Phishing_redefined_-_Preventing_Man-in-the-Middle_Attacks.pdf), section 5	1-30

☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 10 January 2007 (10.01.2007)	Date of mailing of the international search report 23 January 2007 (23.01.2007)
Name and mailing address of the ISA/FI National Board of Patents and Registration of Finland P.O. Box 1160, FI-00101 HELSINKI, Finland Facsimile No. +358 9 6939 5328	Authorized officer Olli-Pekka Piirilä Telephone No. +358 9 6939 500

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2006/000329

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ERIKSSON, M., 'An example of a man-in-the-middle attack against server authenticated SSL sessions', Internet publication, 10 July 2004, http://web.archive.org/web/*/http://www.cs.umu.se/education/examina/Rapporter/MattiasEriksson.pdf, whole document	1-30
A	WO 2004/051964 A2 (FUNK SOFTWARE INC) 17 June 2004 (17.06.2004), whole document	1-30
A	US 2004/0015725 A1 (BONEH DAN et al.) 22 January 2004 (22.01.2004), whole document	1-30
A	WO 03/094491 A1 (PAYCOOL INTERNAT LTD et al.) 13 November 2003 (13.11.2003), whole document	1-30

International application No.
PCT/FI2006/000329

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
US 2004/0139013 A1	15/07/2004	WO 02067534 A1	29/08/2002
		EP 1362466 A1	19/11/2003
		FR 2821225 A1	23/08/2002
WO 03/049364 A1	12/06/2003	US 2005177517 A1	11/08/2005
		CN 1618201 A	18/05/2005
		AU 2002349173 A1	17/06/2003
WO 2004/051964 A2	17/06/2004	CA 2508526 A1	17/06/2004
		AU 2003293381 A1	23/06/2004
US 2004/0015725 A1	22/01/2004	JP 2004054935 A	19/02/2004
WO 03/094491 A1	13/11/2003	MX PA04010737 A	18/10/2005
		US 2005246253 A1	03/11/2005
		JP 2005524184T T	11/08/2005
		EP 1504588 A1	09/02/2005
		CN 1625888 A	08/06/2005
		CA 2481872 A1	13/11/2003
		BR 0215729 A	22/02/2005
		AU 2002256595 A1	17/11/2003

CLASSIFICATION OF SUBJECT MATTER

Int.Cl.
H04L 29/06 (2006.01)
H04L 9/32 (2006.01)
G06Q 20/00 (2006.01)