(19) SUOMI - FINLAND
(FI)

PATENTTI- JA REKISTERIHALLITUS
PATENT- OCH REGISTERSTYRELSEN
FINNISH PATENT AND REGISTRATION OFFICE

(10) **FI 128035 B**
(12) **PATENTTIJULKAISU**
PATENTSKRIFT
PATENT SPECIFICATION

(45) Patentti myönnetty - Patent beviljats - Patent granted **15.08.2019**

(51) Kansainvälinen patenttiluokitus - Internationell patentklassifikation - International patent classification
G06Q 20/22 (2012.01)
G06Q 20/32 (2012.01)
G06Q 20/38 (2012.01)
G06Q 20/02 (2012.01)

(21) Patenttihakemus - Patentansökning - Patent application 20175372

(22) Tekemispäivä - Ingivningsdag - Filing date **25.04.2017**

(23) Saapumispäivä - Ankomstdag - Reception date **25.04.2017**

(43) Tullut julkiseksi - Blivit offentlig - Available to the public **26.10.2017**

(32) (33) (31) Etuoikeus - Prioritet - Priority
25.04.2016 FI 20165360 P

(73) Haltija - Innehavare - Proprietor
1 •Gurulogic Microsystems Oy, Linnankatu 34, 20100 TURKU, SUOMI - FINLAND, (FI)

(72) Keksijä - Uppfinnare - Inventor
1 •Kärkkäinen, Tuomas, TURKU, SUOMI - FINLAND, (FI)
2 •Kalevo, Ossi, TURKU, SUOMI - FINLAND, (FI)

(74) Asiamies - Ombud - Agent
Kolster Oy Ab, Salmisaarenaukio 1, 00180 Helsinki

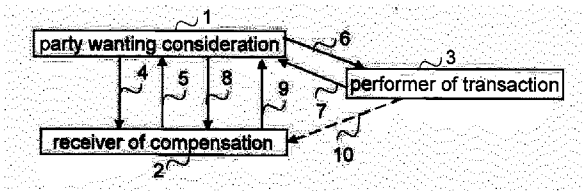
(54) Keksinnön nimitys - Uppfinningens benämning - Title of the invention
Transaktiojärjestely
Transaktionsarrangemang
Transaction arrangement

(56) Viitejulkaisut - Anförda publikationer - References cited
US 2013097081 A1, US 2007005613 A1, US 2008270246 A1

(57) Tiivistelmä - Sammandrag - Abstract

Reaaliaikainen vastikkeellinen transaktio toteutetaan siten, että kun hyödykkeen haluaja saa (1) tiedon vastineesta (5) hyödykkeen haltijalta eli vastineen saajalta (2), välittää hyödykkeen haluaja (1) tiedot (6) vastineesta ja saajasta suojatulla yhteydellä transaktion suorittajalle (3), joka palauttaa tiedon transaktion onnistumisesta (7) hyödykkeen haluajalle (1).

A real time transaction is so implemented that when a party wanting an item gets (1) information on a compensation (5) from the holder of the item, that is, the re-ceiver (2) of the compensation, the party (1) wanting the item conveys the infor-mation (6) on the compensation and the receiver on a secure connection to a per-former (3) of the transaction, which returns information on the success (7) of the transaction to the party (1) wanting the item.



Transaction arrangement

Field of the invention

The invention relates to implementing a transaction and, in particular, to conveying data needed in a transaction.

Background of the invention

Along with the development in communications technology, a large part of various kind of transactions to be conducted have become electronic. Examples of applications used for transactions include applications related to mobile payments, such as paying with a text message, a mobile wallet on smart phones, or a mobile credit card. In text message payments, the payment is typically forwarded to a seller through a mobile phone operator, which in turn adds the invoice to the phone charge. The mobile wallet uses money transferred in advance from an online bank to the electronic wallet, and the payment is performed from the electronic wallet by sending a keyword as a text message to a service number or by contactless reading by utilising the NFC (Near Field Communication) technology or another solution based on the RFID (Radio Frequency Identification) technology. The mobile credit card application is an example of cloud-based paying, and differs from the mobile wallet in that it both sends money directly to a recipient's bank account and an invoice to the person registered as the mobile credit card user on the smart phone. In the known solutions, the payment receiver must support the technology used and the payer must either in advance transfer money onto his terminal device or the payer must make the payment through a service with overdraft facility with which the payer has previously registered. When money is transferred in advance onto a terminal device, or when paying through a service with overdraft facility, the solution is not real time from the payer's viewpoint because the actual transaction debiting the account takes place at a different time.

Further examples relating to electronic payment are disclosed in US 2013/097081, US 2007/0005613 and US 2011/184867.

Brief description of the invention

The object of the invention is to develop a solution carrying out a real time transaction. The object of the invention is achieved by a method, apparatuses and system which are characterized by what is disclosed in the independent claims. Preferred embodiments of the invention are disclosed in the dependent claims.

Brief description of the drawings

The invention will now be described in more detail in connection with preferred embodiments and with reference to the accompanying drawings, in which:

Figures 1A and 1B show simplified general system architectures;

Figures 2A and 2B are flowcharts illustrating the exemplary functionality;

Figures 3A and 3B show simplified data transfer and functionality in an example;

Figure 3C shows simplified data transfer and functionality in another example;

Figure 4 shows simplified data transfer and functionality in a different example;

Figures 5A and 5B show the functionality of different parts of the system of the receiver of the compensation in the transaction in yet another different example;

Figure 6 shows an example of the chain of a real-time transaction;

Figures 7A-7C illustrate user interface views of different kinds; and

Figure 8 is a simplified block diagram of a user terminal device; and

Figure 9 is a simplified block diagram of a transfer server; and

Figure 10 is a simplified block diagram of the equipment of the receiver of the compensation in the transaction.

Detailed description of the invention

The embodiments presented are meant to be examples. Even though the description may refer to "an", "one", or "some" embodiment and/or example (embodiments/examples) in several locations, this does not necessarily mean that each such reference refers to the same embodiment (embodiments) and/or example (examples) or that the feature only applies to one embodiment and/or example. Single features of different embodiments and examples may also be combined to provide other embodiments and examples. It should furthermore be understood that some features, structures and elements used to set up context for the embodiments and/or examples being described may as such be irrelevant to the actual invention. Thus, the structures, words, and expressions in the description below are meant to illustrate, not to restrict the invention and its embodiments.

The invention will in the following be described by using the direct payment application as an example of the real time transaction arrangement without restricting the examples thereto.

Figures 1A and 1B illustrate general architecture according to an embodiment, only showing some elements and functional entities in a communications system 100. The connections and entities shown in Figures 1A and 1B are logical, functional connections and entities; the actual physical connections and entities may be different. It is obvious for a person skilled in the art that other elements, entities, structures and/or subsystems may be included in the system, which need not be described here in any greater detail.

In the example of Figure 1A, the system 100 comprises, as the device generating the initial data for a transaction, a point of sale terminal 110, a user terminal device 120 of the party, that is, the buyer/payer/transaction requesting party, a system carrying out the transaction in real time (real time transaction performer), that is, a banking system 130, a transfer server 140 transferring the transaction data, and a communication network 102 through which the user terminal, transfer server, and banking system may communicate.

The point of sale terminal 110 may be a conventional point of sale terminal or a part of a point of sale terminal system. It is enough that the point of sale terminal 110 comprises an invoice generating unit (l-y) 111 by means of which it forms an invoice of a user's purchases. An invoice advantageously comprises the compensation (the amount of money required), that is, the sum to be paid, and details of the receiver, that is, the compensation target, that is, at least one or more account numbers to which the payment may be paid, or similar details such as the receiver identification data, such as the business ID, phone number, e-mail address or name, by which account details may be found out in the banking system. The data may also be data on the payment receiver's (seller) card, such as payment cards, credit card, combination card, loyal customer card etc., by means of which the account details may be determined. It should be noted that the account details conveyed are account details of the seller and they are conveyed from the seller to the buyer. In other words, in the inventive solution pursuant to the disclosure, details of the user's (buyer) card are not conveyed to the point of sale terminal system (receiver/seller). Hence the details of the user's account, that can be determined using details of the card, are not revealed to the point of sale terminal system. That provides the further advantage that the details of the user's account, or details of the card, cannot be misused. If the seller has an account in a plurality of different

banks, the invoice may contain account details in each bank, or the seller may choose the bank or banks to be entered on the invoice. The seller may, for example, ask from which bank's account the buyer intends to pay and choose the account details on the invoice accordingly. An invoice may also include other data such as a reference number. An invoice may further contain or it may be displayed as one or more bar codes and/or QR codes. A point of sale terminal may also comprise one or more interfaces (not shown in Figure 1) for transferring the invoice details and/or performing the payment. One of the interfaces may be an NFC module or a similar short-range data transfer module by means of which an invoice may be transferred in electronic format to the user terminal 120 if that, too, is equipped with an NFC module or a similar module. Alternatively or additionally a point of sale terminal may comprise a display device as the interface, by means on which invoice details may be displayed, and/or a printer to print the invoice with its details. It is also possible to use an application programming interface (API), such as OPOS (Object linking and embedding for retail Point Of Sale), which makes it possible to integrate the functionality/functionalities straight into the point of sale terminal system and therefore in the point of sale terminals. In other words, any conventional or future point of sale terminal may be used, the invention requires no changes in them. In some embodiments, the point of sale terminal and/or point of sale terminal system may also (as part of a financial management system) be configured to carry out functions relating to receiving payments, as is described in greater detail in connection with Figures 4, 5A, and 5B. In these embodiments, the invoice generating unit may be a part of the invoice management unit.

The transaction trigger (the device initiating the transaction), that is, the user terminal device 120 is configured to be in contact, through the communications network 102, with the transfer server 140 transferring transaction data, and through the transfer server 140, by means of the communications network 102, to the actual system performing the transactions, that is, the banking system 130 in the example of Figure 1. The user terminal may be a mobile communication device, such as a smart phone, tablet, laptop computer, palmtop computer, smart glasses, or any wearable computer, smart clothing, or similar smart device (device that is mobile and includes computing power regardless of where the device has been installed). The user terminal may be wired, connectable, or wireless. In addition, the terminal may be active or passive. (A passive device, such as an ID tag, may be invoked and it may get its supply current from another device). The operating system (OS) of the terminal device may be any operating system, such as Android,

iOS, Firefox OS, Windows Phone, BlackBerry 10, Tizen, Sailfish OS and Ubuntu Touch. In other words, in the example of Figure 1A, the consumer terminal device 120 may be any terminal device enabling establishing a connection to the banking system 130 over one or more communications networks 102 through the transfer server 140 and wherein the terminal device 120 is configurable to initiate a transaction and to take part in it. For this purpose, the terminal device comprises one of more interfaces (IFs) 123 and a secure subarea 122, which comprises a transaction implementation unit (v-t-t-y) 122-1, which in the direct payment applications may be an online banking operations implementation unit, as well as a storing unit (t-y) 122-2 for storing in a secure memory 122-3 the identification information needed in a transaction, as well as a secure memory 122-3. The functionalities of the transaction implementation unit 122-1 will be described in closer detail below in connections with Figures 2 - 4. The storing unit 122-2 is configured to store the identification numbers needed for online banking operations (or mobile banking operations or any operation meant for real time electronic payments), such as log-in bank credentials and possible key codes of an online bank, or part of the credential numbers in secure form, as will be described below. In secure form, any encryption method may be used, including symmetric or asymmetric encryption technology, or technologies making use of these, the keychain developed by Apple, or a similar solution, or completely new solutions such as encryption key wallet and a secure way to use its content or that of a similar storage. The encryption key wallet, or actually an encryption key wallet pair, is an encryption and decryption mechanism between two parties, in which the wallet is opened using an identifier. The encryption key wallet is described in the patent application GB 1507154.1 and the patent application GB 1620553.6 then describes a solution where it is secured that the keys are never outside the key storage in an unencrypted form, and they are not even brought out of the storage but the keys are only referred to with their respective reference numbers.

The secure sub-area 122 is a secure environment isolated from the other terminal device applications, meant for an application carrying out a real time transaction so that the application is only executable when the integrity of the application has been ensured and the user identified, as will be described below. The secure sub-area may be implemented by using a solution developed by the equipment manufacturer, for example. An example of this is the Samsung KNOX, which isolates certain applications into its own area and encrypts the data in the

isolated application both in the idle state as in use. A part of the secure area may also be implemented by the use of an encryption technology.

The secure sub-area 122 comprises the secure memory 122-3. Its integrity may also be verified before launching the application or in connection with it, as will be described below. The secure memory 122-3 has, for the direct payment application, different kind of identification details 122-31 and possibly credential details 122-32 of online banks, at least some of which details, advantageously all, as encrypted and during use possibly partly decrypted. Such details include, for example, the private key of the Public Key Infrastructure (PKI) management system and the identification detail or details required for use of the application and advantageously also the network address of the transfer server. For example, the Samsung KNOX encrypts the data in the secure memory by using the Advanced Encryption Standard (AES) encryption algorithm. Other encryption algorithms, such as the Salsa20 and its variants, may also be used. The use of the data will be explained in greater detail below in connection with Figures 2-6. It is obvious for a person skilled in the art that the storage arrangement described here may be implemented in various ways, and some of the data may, for example, be stored in a common memory in encrypted form so that data decryption may only be carried out with the key/keys in the secure area.

In the example of Figure 1A, the performer of a transaction 130 is the system of a payment service provider, usually a bank. Therefore the term banking system will be used below. It enables real time payment over a communications network 102 at the electronic payment interface (payment interface unit, m-y) 131 it offers, for example a browser-based online bank application, mobile bank application, or online bank implemented by programming interface. The programming interface may be a strong customer authentication (SCA) interface, over which money may be transferred onto an account, from an account, and account details requested. In the inventive solutions pursuant to the disclosure, any current or future bank application may be used, that is, a user may freely choose the source of a compensation (from where the transaction is performed), that is, the bank being used and the account there and in some cases even the party transferring the payment (that is, the transfer server), and the invention does not require changes to be made in the banks' solutions. It should also be noted that various kind of loyal customer accounts that may be used to pay for one's purchases, possibly even using a payment method other than money, are in this invention equal sources of compensation, so directly comparable with bank accounts; for example, they appear as

accounts and the provider of a regular customer system may similarly be interpreted as one of the online banks.

In the system of Figure 1A, there is one or more (one, only, shown in Figure 1A) transfer servers 140 transferring transaction data, such as the server of a Payment Initiation Server Provider (PISP), configured to transfer over the communications network 102 data needed in the transaction between the system 130 that carries out the transaction and the trigger 120 of the transaction. The transfer server may be an application server, for example, which in its simplest form is a computer executing the application. In other words, in the example of Figure 1, the transfer server 140 may be any computer that enables the establishing a secure connection to the banking system 130 and the terminal device 120, and which may be configured to take part in a transaction. For this purpose, the transfer server comprises one or more interfaces (IFs) 143, a transaction data transfer unit (v-t-t-s-y) 141, and a memory 142 for at least storing identification details 142-1 and the parsers 142-2 of different online banks. The parsers represent herein performer-specific settings of a real time performer, which make possible the transfer of the required compensation data, and the authentication details possibly needed, by parsing the data to suit the corresponding interface. In other words, a parser adapts the events directly to the interface of the payment service provider. A parser may be web scraping or screen scraping, which on behalf of the user performs one or more machine language functions considered for the user to perform. Despite the name, data is not only scraped but also pushed, that is, given to the (payment) interface. Bank-specific parsers may take into account the slightly different payment interface implementations of different banks, namely online banking implementations, browser and SCA interfaces, differing from each other, which one bank may maintain in parallel for the online bank, and even country-specific differences in the payment interfaces. These interfaces have in common the fact that they enable the carrying out of the desired payment and require strong authentication in the system of the bank in question. It is the parser's task to find the fields or similar provided over the interface, into which data is fed, ask for/retrieve corresponding data and feed them in the form in which they transfer through the interfaces to the banking system so that the operation of the parser is transparent for the banking system. In other words, the bank's system does not know whether the data were input with the aid of a parser or directly by a user, for example. This functionality of the parser differs from the parser disclosed in the patent application US 2007/0005613, in which a scheme-based parser converts the format of received

data into the format used by the application. Likewise, the reason for a plurality of parsers is different: in the solution pursuant to the disclosure, they are needed, because data needs to be conveyed through a plurality of different interfaces, whereas the solution of the patent application US 2007/0005613 needs a plurality of parsers because the data in a plurality of formats must be converted to the one format that the application is using. The parsers may be implemented by any known or future parser technology. For example, the parser package called "Beautiful Soup", made with the Python programming language may be used. Compared to a solution in which the mobile payment solutions of banks would be stored in their entirety, the storing of the parsers, only, saves memory resources. In addition, the operating system of the server may be chosen freely; the applications are always made for an operating system, and if mobile payment applications were to be stored, it would limit the choice of the operating system. The usability is better, too. The functionalities of the transaction data transfer unit 141 will be described in closer detail below in connection with Figures 3A - 3C, in particular.

The communications network 102 of Figure 1A may comprise one or more networks, wherein a network may be a wireless network or a wired network or their combination. How the communications network is implemented bears no significance for the invention. The communications network or a part of it may be based on, for example, the third, fourth, or fifth generation wireless communications technology, a wireless local area network, such as Wi-fi or Li-fi, or other wireless close range transmission, such as infrared, bluetooth, or internet technology, broadband network technology and/or wired phone network.

Depending on the embodiment or manner of implementation, what, in the use of the application, is stored on the terminal device 120, and what in the transfer server 140, varies. At one extreme end, the transfer server 140 has the parsers, only, and other data is stored in an encrypted format on the terminal device 120 or the terminal device 120 is adapted to request data from the user. At the other extreme end the user terminal device 120 contains the data, only, that is used to ensure the user's right to use the application, and all other online credentials needed to use the bank services are stored encrypted in the transfer server 140. In intermediate embodiments, credentials remaining unchanged may, for example be stored encrypted in the transfer server and single-use credentials encrypted in the memory of the terminal device. Other ways to allocate what data is stored on the

terminal device and what in the transfer server may also be used. It is further possible that the same data is stored encrypted both in the transfer server and the terminal device.

A solution where practically all the data resides in the transfer server 140 reduces the network load as there is no need to request and transfer credential details. On the other hand, storing credential details on the user terminal device 120 increases security: it is less interesting to hack a single terminal device containing the information on one person than to hack a server including information on a plurality of people, because the benefits from hacking the online credentials of one individual will be far lesser.

The storing unit 122-2 is adapted to store the credential details 122-32 of online banks advantageously online bank specifically and, depending on the implementation, as encrypted either in parts or as a whole. The storing unit may store the data not only on the terminal device 120 but also in the transfer server 140. The storing unit 122-2 advantageously encrypts the data to be stored. Any encryption method may be used for data encryption. The standard data to be stored is advantageously encrypted by using a different encryption key or keys than used for encrypting variable data. This way, the data that is possibly known does not jeopardise the entire protection. The storing unit 122-2 may be configured to encrypt the data stored on the terminal device in such a way that the data may only be decrypted with the decryption key of the transfer server. The terminal device and the transfer server may also have encryption key wallets (an encryption key wallet pair) by means of which the data may be encrypted.

For example, when the encryption key wallet is used, the credentials may be stored in the wallet so that when the wallet is open they may all be read based on, for example, the serial number of the key. The credentials may also be stored as individually encrypted so that the serial number of that key is not stored, for example. By means of the address space thus obtained, the encrypted credentials will be found, and if they are stored on the terminal device, they may be transferred, depending on the encryption method, as encrypted from the terminal device to the transfer server or as individually decrypted. When the data is separately encrypted, the data security of the transfer server will be improved, because in there, too, the data then needs to be individually hacked and in addtion the discovered details need to be combined with the correct accounts.

The storing unit 122-2 of the terminal device may receive the details, such as a key code list, so that the user inputs the data, uses the QR scanner of the

terminal device or another similar scanner, takes a photo of the credentials, gets them in an electronic format from the bank, etc.

Besides getting the data through the terminal device 120, the transfer server 140 may get the data directly from the bank 130.

It should also be noted that in the future the bank connection may work with the encryption key wallet and then its (user-specific) counterpart either in the transfer server or on the terminal device may replace user-specific bank details. The solution using a transfer server may therefore have in the transfer server two counterparts of the key wallet for the same user - one for use between the transfer server and the terminal device, and the other for use between the transfer server and the bank. It is also possible that some time in the future one encryption key wallet pair (encryption key wallet pair common to the users) is used between the bank and the transfer server for all the users who have given their consent. The encryption key wallets may, in addition to the identifier, also have a serial number based on which it is identified which encryption key wallet is used at any given time. The serial number of the key wallet may be kept separate from the serial number of the key, but if so desired the serial numbers may also be combined. It is further possible that the same encryption key wallet is used for a group of different parties (so the group has a common key wallet). Therefore the encryption key wallet makes it possible to achieve the same as when applying the Diffie-Hellman key exchange protocol among a plurality of parties, except that when using the encryption key wallet, the process of encryption does not slow down which happens if the Diffie-Hellman key exchange protocol is used solely.

The exemplary system 100 of Figure 1B differs from the exemplary system of Figure 1A in that a separate transfer server does not exist, but the user terminal device 120a is configured to carry out the data transfer, too, directly with the performer of the transaction, that is, the banking system 130. For this purpose, the user terminal device 120a comprises an enhanced transaction implementation unit (l-v-t-t-y) 122-1a, which is a combination of the transaction implementation unit on the user terminal device in the example of Figure 1A and the transaction data transfer unit of the transfer server, and in the memory, either in the secure memory 122, as in Figure 1B, or in an unsecure memory, the parsers 123-33 of the different online banks. Depending on the implementation method, the memory may have the parsers of all the online banks, including the regular customer systems, or only the parsers of those online banks from whose accounts the user may perform a transaction, that is, to pay from. The latter solution uses up less memory resources.

It should be understood that instead of one enhanced transaction implementation unit the corresponding functionality may be implemented with several units (sub-units). For example, a transaction implementation unit and the transaction data transfer unit in the transfer server may be implemented as separate units also on the terminal device, that is, as components. In such an embodiment, the functionalities between these units may be shared in a number of ways, for example so that the transfer server (transaction data transfer unit) also carries out the functionalities of the transaction implementation unit, in which functionalities the data in the secure memory is used.

The use of the transfer server has the benefit that when the bank (or regular customer system) makes a change in the online bank functionality, the parser is correspondingly easy to update: the parser only needs to be updated in the transfer server (or transfer servers, if a plurality of transfer servers provide the online bank functionality of the bank). In the solutions that do not have a dedicated transfer server (dedicated transfer servers) operating in the network, updating the parsers causes a heavier load on the resources, because the parser needs to be updated in every terminal device that has one. For this purpose, information on the terminal devices must be maintained somewhere, and if the number of parsers stored on the terminal devices varies per terminal device, information needs to be maintained on what parsers there are on which terminal devices or alternatively ask, in connection with updates, what parsers a terminal device has. In any case, in these solutions, too, that do not have a dedicated transfer server operational in the network, when one payment service interface changes, one parser, only, or one application (one service) needs to be updated on the terminal devices.

Although the examples above describe the parsers as separate, it should be understood that a parser may be integrated into the application used either in the transfer service or terminal device either so that the parser, only, is updated as the payment interface changes, or so that the entire application is updated.

Figure 2A shows an example, together with Figures 7A-7C, of the operation of the transaction implementation unit, which in this example is a real time payment application using a transfer server. It should be understood that the transaction implementation unit may be divided into sub-units, which perform one or more of the functions. In the example of Figure 2A, it is assumed that the verification of the application, verification of the memory, and the user authentication were successful. If one of these fails, the process is naturally interrupted. It is fur-

ther assumed in the example of Figure 2A that the choosing of an account also selects the transfer server, but in another example both the account and the transfer service to be used may be chosen separately.

Referring to Figure 2A, when the activation of a payment application is detected at step 200, the payment application is verified (authenticated) at step 200. This step may also include the identification of the payment application by means of, for example, the public key management system. The activation may be triggered by the reception of an electronic invoice from a point of sale terminal or a user's input (selection) that activates the application. The user may, for example, choose, as shown in Figure 7A, the application (PAYER) 70 on the display 700 of his terminal device. The verification of the payment application may be performed by, for example, by calculating from the payment application its check sum or check key created, for example, with CRC (Cyclic Redundancy Check) or with help of a cryptographic hash created, for example, with a SHA function (Secure Hash Algorithm) such as SHA-1, SHA-2, or advantageously SHA-3, and comparing the calculation result to the check sum, check key, or hash of the application in the identification details. If they match, the application is unchanged, that is, verified. In other words, it is ensured in the practise that the code of the application has not been changed and that the application has been downloaded from an identified and reliable source. In this example, the memory is also verified at step 201. The memory may be verified in the same manner as the application. Typically, the device manufacturer has defined the way to ensure the memory. After this, user authentication is carried out at step 201. The user may be identified by means of, for example, a biometric identifier integrated into the terminal device, such as PositiveID, fingerprint recogniser, or iris recogniser, possibly as a two-phase identification in which the biometric authentication is linked with the user's password or the terminal device PIN code / security code and/or a security code opening the terminal device. Instead of the biometric identifier, or together with it, any digital credentials may be used. A secure option is to combine, in an inauguration of the user terminal device, the finger print recognition integrated into the terminal device or another strong authentication, with information received through another external route, such as information transferred through an NFC reader or RFID reader on a physical ring, watch, implant, or keyring. By storing the application in the secure memory and strong user authentication help prevent possible data leak problems caused by malware or spyware.

Once the user has been identified, that is, his right to use the accounts has been verified, at step 202 an invoice is read from the point of sale terminal, that is, the sum to be paid and the information on the account or accounts where the payment may be performed. Other information, such as a reference number or archiving identifier, may also be conveyed. The reading may be carried out by using any known technology. The reading may be performed by using, for example, an NFC reader or another short-range reader, such as infrared or wifi, or light, by reading the QR code(s) and bar code(s) with the aid of the terminal device application and camera. The invoice, or payment, may also be read by using vibration recognition by means of vibration, or by using voice recognition by means of voice. In this example, the reading of an invoice is also considered to include the manual feed of invoice details on the user interface of the terminal device. The information may also be transferred along a fixed bus, such as by a USB connection. The invoice details may be conveyed unencrypted, because they are not enough for paying alone, and the details based on which the actual payment may be charged from the payer's account are sent encrypted, as will be explained below.

Once the invoice has been read, it is converted, if need be, into a payment at step 203. Naturally, if the invoice to be read is directly in the payment format, no formatting is needed.

At the same time, or afterwards, it is determined at step 204 the accounts available to the user and the banks in the invoice. Naturally the accounts the user may use may have been defined at an earlier stage, by reading their details from the secure memory in connection with activation, for example. If the user has more than one account in use (step 205: yes), at step 206 the user is prompted to choose the account he wants to use. In another embodiment, if there are more than one bank on the invoice, the user is also prompted to choose the bank account to which he wishes to send the payment, no matter how many accounts the user has. The user is prompted to choose an account by the display of the request and the account options of the terminal device screen, or by the display of the accounts, only. Depending on the embodiment, the accounts may always be shown in the same order, or in the order in which they have most frequently been used, or in an order defined by a bank (if there are several accounts at the same bank) and/or the order may take into account the bank/banks in the payment so that at the top there are the accounts that are at the same bank as in the invoice, for example in the order of the payment, and/or the accounts may be displayed in the order in which they have balance on the account and/or just the accounts that have adequate balance

may be shown. An example is shown in Figure 7B, in which account selections 71 are shown on the application display 701 on the user interface of the user terminal device. In the example of Figure 7B, the user has two accounts at the bank P3, one account at the bank P1, and one account at the bank P2. It should be understood that in the above just a few examples on the bases of what and how to show accounts to a user have been described. When the user's account choice is received at step 207, the payment event is then shown at step 208 to the user together with the approval request of the payment, that is, a confirmation request. An example is shown in Figure 7C where the payment event 72 is shown on the application display 701 on the user interface of the user terminal device, that is, the sum and the account from which the sum is debited as well as selection buttons 72a, 72b by means of which the user may approve or reject the payment. In other words, at this step it is once more verified that the user wishes to pay the invoice. If there were no options (step 205: no), this step is reached directly. If the user does not approve the payment (step 209: no), paying and the application are interrupted at step 216.

If the user approves the payment (step 209: yes), the online payment is carried out. The performing of the online payment is started by establishing, at step 210, a secure connection to the transfer server, such as an HTTPS connection (HTTPS, Hypertext Transfer Protocol Secure), in which the HTTP data transfer protocol is used either with the TLS (Transport Layer Security) or SSL (Secure Sockets Layer) encryption protocol. The use of the latter requires that the parties trust each other's SSL certificate. The secure connection may also be established in other ways, such as by using the encryption key wallet referred to in the above. In addition to the secure connection, it is also possible to convey the data as secured/encrypted, for example with PGP (Pretty Good Privacy) according to the OpenPGP standard. If the data is encrypted so that only the transfer server may decrypt it, the data may also be sent on a normal communications link. In other words, it is possible to secure the connection and use an unsecure (unencrypted) data, or encrypt the data and use an unsecure connection. Naturally, encrypted information may also be sent on a secure connection.

When the secure connection has been established, at step 211 information relating to the paying is sent, that is, the account from where the payment is made, the account to which the payment is made, and the sum to be paid to the transfer server. In other words, the payer's account details are not at any stage transferred to the point of sale terminal, so the risk of sensitive information falling

in the wrong hands is smaller and the point of sale terminal system is therefore not required to be as secure as, for example, in bank card and credit card payments.

When a message is received from the transfer server (step 212), it is checked at step 213 whether it is a request for information relating to paying or not. Figures 3A and 3B show examples of different information requests. If it was a request (step 213: yes), at step 214 the required details are retrieved from online bank credentials, which have been decrypted as background processing, and they are sent at step 214, after which a return is made back to step 212 to receive a message from the transfer server.

If the message was not a request (step 213: no), information is received on a successfulness of the payment event and it is shown at step 215. If the payment was successful, the information is advantageously a receipt of the payment or information on a funds provision on the payment. When this is shown on the display, it may be shown to the seller and the purchased item or items are received. Naturally, when the above is used in online shopping, especially if the purchased item or items are delivered to an address given by the buyer, there is no need to show the receipt or information on the funds provision to receive the purchased item or items. Further, it should be appreciated that a purchased item may be received either immediately or after some time after showing the receipt, or after the successful payment. After this, the application may close automatically or it may wait for the "close" input by the user and/or offer the user the option to make a new payment. The application may also be arranged to recognise a failed payment, possibly also the reason for the failure, such as a wrong credential/identifier, or insufficient funds on the account. If the user has a plurality of accounts, a return may be made to step 206, and to display this time the accounts from which a payment has not yet been attempted to be done.

Figure 2B shows an example which differs from the example of Figure 2A in that a dedicated transfer server is not in use and the authentication is performed at a different step. The similar steps of Figures 2A and 2B are marked with the same reference numbers and their detailed description is unnecessary to be repeated here.

Referring to Figure 2B, when the activation of a payment application is detected at step 200, the payment application is verified (authenticated) at step 200 and the memory is verified at step 201, as described in the above. Then, at step 202 an invoice is read from the point of sale terminal, that is, the sum to be paid

and the information on the account or accounts whereto the payment may be performed. Other information, such as a reference number or archiving identifier, may also be conveyed. Once the invoice has been read, it is converted, if need be, into a payment at step 203. Naturally, if the invoice to be read is directly in the payment format, no forming is needed.

At the same time, or afterwards, it is determined at step 204 the accounts available to the user and the banks in the invoice. Naturally the accounts the user may use may have been defined at an earlier stage, by reading their details from the secure memory in connection with activation, for example. If the user has more than one account in use (step 205: yes), at step 206 the user is prompted to choose the account he wants to use. When the user's account choice is received at step 207, the payment event is then shown at step 208' to the user together with the authentication request and the approval request of the payment, that is, the confirmation request. In other words, at this step it is once more ensured that the user wishes to pay the invoice, and the user is authenticated only if he wishes to pay the invoice. If there were no options (step 205: no), this step is reached directly. If the user does not approve the payment or does not authenticate himself or authentication fails (step 209: no), paying and the application are interrupted at step 216.

If the user approves the payment and authenticates himself (step 209: yes), the online payment is carried out. The performing of the online payment is begun by selecting, at step 217, a parser and by establishing, at step 210, a secure connection to the interface of the online bank, such as a HTTPS connection referred to in the above. The secure connection may also be established in other ways, such as by using the encryption key wallet referred to in the above. In addition to the secure connection, it is also possible to convey the data as secured/encrypted, for example with PGP (Pretty Good Privacy) according to the OpenPGP standard.

When the secure connection has been formed, at step 211' information relating to the paying is sent, that is, the account from where the payment is made, the account to which the payment is made, and the sum paid to the interface of the online bank.

When a message is received from the online bank interface (step: 212'), the parser extracts its content (step 212') and then the application checks at step 213' whether it is a request for information concerning paying or not. Figures 3A and 3B show examples of different information requests. If it was a request (step

213': yes), at step 214' the required details are retrieved from online bank credentials, which have been decrypted as background processing, the parser modifies them into the form the online bank interface wants at step 214', and the information is sent at step 214', after which a return is made back to step 212' to receive a message from the online bank interface.

If it was not a request (step 213': no), information is received on a successfulness of the payment event from the online bank interface, and it is shown at step 215. If the payment was successful, the information is advantageously a receipt of the payment or information on a funds provision on the payment, and when this is shown on the display, it may be shown to the seller and purchased item or items are received. After this, the application may close automatically or it may wait for the "close" input by the user and/or offer the user the option to make a new payment. The application may also be arranged to recognise a failed payment, possibly also the reason for the failure, such as a wrong credentials/identifier, or insufficient funds on the account. If the user has a plurality of accounts, a return may be made to step 206, and to display this time the accounts from which a payment has not yet been attempted to be done.

Figures 3A and 3B describe in closer detail the data transfer related to actual online paying between the user terminal device (transaction implementation unit), the transfer server (transaction data transfer unit), online bank (online banking system, that is, real time performer of a transaction), and the receiver (the receiver's account system). It is assumed in Figures 3A and 3B that paying succeeds. In addition, the information sent on the secure connection in the figures are encrypted before sending and decrypted before using, but for reasons of clarity the encryption and similarly decryption are not repeated in connection with information exchange. Naturally, if an identifier or challenge-response pair is wrong, paying will fail. It should be understood that in some embodiments an identifier or challenge-response pair may be asked again, for example once, twice, or three times before paying fails. To increase clarity, the events are described by using a particular type of a payment interface operating on a web page. It should be understood that a payment interface of another type operating on a web page or API may similarly be used. In other words, the implementation of the same principle on a very interactive interface, where information is requested in a number of stages, or on an interface asking very little information, in which all the required details (or detail) is provided in one or two stages, is straightforward for a person skilled in the art based on what is described in this application.

Referring to Figure 3A, when the user terminal device has performed (point 3-1) the verification of the application, that is, the transaction implementation unit, user authentication, and the steps relating to paying all the way up to the approval of the payment, we are in a situation following step 209: yes in Figure 2. After this, a secure connection is established (messages 3-2) between the terminal device and the transfer server. As explained in connection with Figure 2, the connection may be an HTTPS connection in which the information is encrypted during the time of the connection (arrive unencrypted, are encrypted just before sending, decrypted at reception) by the TLS protocol, for example. When the connection has been established, the information related to paying is sent (message 3-3), such as the payer's account, seller's account, and sum. Instead either or both account details, there may be other information by means of which the corresponding account detail may be found out. Examples of such other details are in the above description. Yet further information, such as a reference number, may be related to paying.

After receiving the information on paying, the transfer server determines, based on the payer's account, the payer's bank and selects at point 3-4 the parser to be used and establishes a secure connection (messages 3-5, 3-6) such as an HTTPS connection, to the online bank of the bank to exchange information. Naturally, the transfer server and online bank may identify each other before the establishing of the secure connection. In the identification, a certificate issued by a reliable party may be used, such as a valid e-IDAS certificate.

After this, the actual establishing of the online bank connection is carried out, with logging in, by using secure connections.

First, the transfer server requests (message 3-7) from the online bank the hypertext mark-up language content of its log-in page (www page or URL, uniform resource locator), and having received in message 3-8 the content returned by the online bank, initiates the selected bank-specific parser and searches for at point 3-9, with the aid of the parser, the fields of the input elements required for the log-in. The fields are typically bank-specifically in an HTML form, defined for logging in. After finding them, the transfer server requests the required information (message 3-10) from the user terminal device.

The user terminal device retrieves (point 3-11) the requested information from the credential details of the online bank of the bank of the selected account and returns them to the transfer server in message 3-12. If the details have been stored in parts, the user terminal device may decrypt the encryption of the part containing these details either at this phase, or if the details have been secured

as a bank-specific whole, all the details. Alternatively, the secure information may be decrypted already earlier as background processing in response to the approval of the payment, for example. Depending on the implementation, these logging-in details may also be requested from the user, or at least one of them, if more than one logging-in detail is needed.

Having received the logging-in information, the transfer server compiles a response at point 3-13. In other words, in this example it compiles the input element fields into an HTTP POST request and sends them (message 3-14) into the URL address obtained in message 3-8. (In the example of the figure, the URL address is in the action attribute of the form element in the logging-in form). It is assumed in the example of the figure that the logging-in details were correct, so when the parser in the transfer server reads the response (message 3-15) , that is, the HTML content, returned by the online bank, it detects at point 3-16 being logged in to the online bank.

After the logging-in, the online bank session is activated.

First, the transfer server requests (message 3-17) from the online bank the HTML content of the activation form of the session, and having received the content returned by the online bank in message 3-18, the parser at point 3-19 searches for the challenge code of the challenge-response method, for example, the form element or input element of the activation form, and requests (message 3-20) the challenge code from the user terminal device.

The user terminal device retrieves (at point 3-21) the challenge code from the credential details of the online bank of the bank of the selected account and returns it to the transfer server in message 3-22. If the details have been stored in parts, the user terminal device may decrypt the encryption of the part containing this information at this stage. Depending on the implementation, the challenge code may also alternatively be requested from the user.

Having received the challenge code, the transfer server compiles a response at point 3-23. In other words, in this example it compiles the input element fields into an HTTP POST request and sends the filled-in fields of the input element of the challenge response code (message 3-24) to the URL address obtained in message 3-18. (In the example of the figure, the URL address is in the action attribute of the form element in the challenge response read form). It is assumed in the example of the figure that the challenge response code was correct, so when the par-

ser in the transfer server reads the response (message 3-25), that is, the HTML content, returned by the online bank, it discovers at point 3-26 that the online bank session has been activated.

After that, the payment is registered.

First, the transfer server requests (message 3-27) from the online bank the HTML content of its invoice payment form, and having received the content returned by the online bank in message 3-28, the parser searches and fills in, using the information received in message 3-3, at point 3-29 the input elements required for paying, such as the account number from which the payment is charged, name of the invoice receiver, invoice reference number, invoice sum, and account number of the receiver. Some of the input elements may be required elements, some non-required. Depending on the implementation, the transfer server may request possibly missing information from the user terminal device, or from the user by the user terminal device prompting the user to input the missing information to the user terminal device. When the details of the input elements needed for paying have been determined, the transfer server compiles the payment at point 3-29. In other words, in this example it compiles the input element fields into an HTTP POST request and sends (message 3-30) the filled-in invoice payment form to the URL address obtained in message 3-28. (In the example of the figure, the URL address is in the action attribute of the form element in the challenge response read form).

It is assumed in the example of Figure 3A that the account had money, so the online bank carries out a preliminary payment (registering of a payment provision) at point 3-31 and returns a response about it (message 3-32).

When the parser of the transfer server read the response (message 3-32), that is, the HTML content, returned by the online bank, it detects that the payment has been registered.

After the payment registration, the transfer server, or more correctly, the parser of the transfer server, therefore initiates payment confirmation and approval.

First, the transfer serve requests (message 3-33) from the online bank the HTML content of its payment confirmation form and receives, in message 3-34, the content returned by the online bank.

After that, the process of Figure 3A continues in Figure 3B.

Having received the content (message 3-34), the parser searches at point 3-35 the challenge code of the challenge response number in the content,

such as the form element or input element of the challenge response form, and requests (message 3-36) the challenge code from the user terminal device.

The user terminal device retrieves (at point 3-37) the challenge code from the credential details of the online bank of the bank of the selected account and returns it to the transfer server in message 3-38. If the details have been stored in parts, the user terminal device may decrypt the encryption of the part containing this information at this stage. Depending on the implementation, the challenge code may also alternatively be requested from the user.

Having received the challenge code, the transfer server compiles a response at point 3-39. In other words, in this example it compiles the input element fields into an HTTP POST request and sends the filled-in fields of the input element of the challenge response code (message 3-40) to the URL address obtained in message 3-34. (In the example of the figure, the URL address is in the action attribute of the form element in the challenge response read form). It is assumed in the example of the Figure that the challenge response code was correct, so the online bank carries out the payment at point 3-41 and returns information (message 3-42) on it to the transfer server. At the same time, information on the payment is conveyed (message 3-43), like a normal payment on an online bank, also to the banking system of the receiver in real time. This way, information on the payment may also be conveyed to the fiscal system required by the law and consequently to the local tax authority. This means that if the accounts are at the same bank, information on the payment (at point 3-52) is received immediately, in practise, but for the time being there may be delays between different banks.

When the parser of the transfer server reads the response (message 3-42), that is, the HTML content, returned by the online bank, it detects at point 3-44 that the payment has been paid.

After that, the payment is verified.

First, the transfer server requests (message 3-45) from the online bank the HTML content of the receipt of the paid invoice, and having received in message 3-46 the content that the online bank returned, the parser finds the archiving identifier in it and compiles by using it a receipt at point 3-47 and sends (message 3-48) it to the user terminal device. In this example, the user terminal device is adapted to show at point 3-49 the receipt on the terminal display. In another implementation, the receipt may instead, or additionally, be sent to the point of sale terminal which, if configured, may confirm to the point of sale terminal user that the payment has been made.

In the example of Figure 3B, after the payment authentication, the online bank session between the transfer server and online bank (messages 3-50) and the secure connection between the transfer server and the user terminal device (messages 3-51) are closed.

In embodiments in which credentials of the online banks have been stored in an encrypted format in the transfer server, message 3-3 may be used to convey information by means of which the credentials of the online banks may be decrypted. For example, a common shared secret or a secret key as the counterpart of a public key may be delivered. A most data secure method is also to deliver an identifier that opens the encryption key wallet. The conveying of the information needed for decryption may also involve sending a plurality of messages between the transfer server and the terminal device. In embodiments in which the details are stored in the transfer server, the messages 3-10, 3-12, 3-20, 3-22, 3-36, and 3-38 are internal data transfer within the transfer server, and points 3-11, 3-21 and 3-37 are carried out in the transfer server. In embodiments in which a part of the details are in the transfer server and a part in the terminal device, some of the messages described in connection with Figure 3 may be data transfer within the transfer server (and the corresponding points are carried out in the transfer server).

It is also possible that the receiver of the payment, such as a shopkeeper, requires an electronically signed receipt of the performed payment. Any known or future technology may be used for electronic signatures, such as the use of personal bank credentials for the electronic signature or the Signom signature service.

Figure 3C describes another example of the data transfer related to the actual online paying between the user terminal device (transaction implementation unit), the transfer server (transaction data transfer unit), online bank (online banking system, that is, real time performer of a transaction), and the receiver (the receiver's account system). The example of Figure 3C differs from the example of Figures 3A and 3B in that in the example of Figure 3C the transfer server has the required identifier details and the payment is performed without the payment provision. Similar reference numbers as in Figures 3A and 3B are used in Figure 3C at similar points and messages.

Referring to Figure 3C, when the user terminal has performed (at point 3-1) the verification of the application, that is, the transaction implementation unit, user authentication, and the phases relating to paying all the way up to the user

having approved the payment, after that a secure connection is established (messages 3-2) between the terminal device and the transfer server and the details relating to paying are sent (message 3-3) as described in connection with Figure 3A.

After receiving the information on paying, the transfer server determines, based on the payer's account, the interface of the payer's bank and selects at point 3-4' the parser to be used and the required identifier details of the payer and establishes a secure connection (messages 3-5') such as an HTTPS connection to the online bank of the bank to exchange information. Naturally, the transfer server and online bank may identify each other before the establishing of the secure connection. In the identification, a certificate issued by a reliable party may be used, such as a valid e-IDAS certificate.

In this example, after the connection establishment, the online bank sends in message 3-8' the details needed for logging in in the format of the interface. After receiving it, the transfer server initiates, unless it has already initiated previously, the selected bank-specific parser and searches for at point 3-13', with the aid of the parser, the fields of the input elements required for logging in. After finding them, the transfer server retrieves (point 3-13') the requested information from the credential details of the online bank of the bank of the selected account. If the details have been stored in parts, the transfer server may decrypt the encryption of the part containing these details either at this point, or if the details have been secured as a bank-specific whole, all the details. Alternatively, the encrypted information may already be decrypted earlier as background processing in response to the payment details (message 3-3) received from the user terminal device, for example. After that, the transfer server compiles at point 3-13' the response (message 3-14) according to the settings of the online bank interface. It is assumed in the example of the figure that the logging-in details were correct, so when the parser in the transfer server reads the response (message 3-15), that is, the HTML content, returned by the online bank, it discovers at point 3-16 being logged in to the online bank.

Then the transfer server requests (message 3-27) from the online bank the information needed to pay the invoice, for example, the content of its invoice payment form or similar information, and having received the content returned by the online bank in message 3-28, the parser searches and fills in, using the information received in message 3-3, the input elements required for paying at point 3-29, such as the account number from which the payment is charged, name of the invoice receiver, invoice reference number, invoice sum, and account number of

the receiver. Some of the input elements may be required elements, some non-required. Depending on the implementation, the transfer server may request possibly missing information from the user terminal device, or from the user through it. When the details of the input elements needed for paying have been determined, the transfer server compiles the payment at point 3-29. In other words, it compiles in this example the fields of the input elements into the format required by the interface of the online bank and sends (message 3-30) the details to the interface.

It is assumed in the example of Figure 3C that there was money on the account, so the online bank carries out the payment at point 3-41 and returns information (message 3-42) on it to the transfer server. At the same time, information on the payment is conveyed (message 3-43), like a normal payment on an online bank, also to the banking system of the receiver in real time. This way, information on the payment may also be conveyed to the fiscal system required by the law and consequently to the local tax authority. This means that if the accounts are at the same bank, information on the payment (at point 3-52) is received immediately, in practise, but for the time being there may be delays between different banks.

When the parser of the transfer server reads the response (message 3-42) returned by the online bank, it detects at point 3-44 that the payment has been done.

After that, the payment is authenticated.

First, the transfer server requests (message 3-45) from the online bank the receipt of the paid invoice, and having received in message 3-46 the content that the online bank returned, the parser searches for the archiving identifier in it and compiles by using it a receipt at point 3-47 and sends (message 3-48) it to the user terminal device. In this example, the user terminal device is adapted to show at point 3-49 the receipt on the terminal display. In another implementation, the receipt may instead, or additionally, be sent to the point of sale terminal which, if configured, may confirm to the point of sale terminal user that the payment has been made.

In the example of Figure 3C, after the payment authentication, the online bank session between the transfer server and online bank (messages 3-50) and the secure connection between the transfer server and the user terminal device (messages 3-51) are closed.

It should be understood that the examples described in the above in Figures 3A- 3C only illustrate the use of a parser and that the information and confirmation that an online bank requests may differ from what is described in the above according to the settings of the electronic payment interface of each online bank.

In embodiments that do not use a transfer server (transfer servers) separate from the terminal devices, the data transfer shown in Figures 3A and 3B or Figure 3C between the transfer server and the terminal device is internal data transfer carried out by the terminal device, or more precisely enhanced transaction implementation unit, or data transfer between and/or within two or more sub-units/components in the terminal device, which together form the enhanced transaction implementation unit, depending on how the functionalities have been shared among the sub-units. Internal data transfer need not be transferred on a secure connection, but care should be taken that other applications or processes do not get the transferred information. In addition, the data transfer through the communications network is lesser because over the network only the information is transferred that in the example 3A-3C is transferred between the transfer server and online bank.

As the above examples show, the application operates as an improved, for example, as a more secure, digital bank card - payments are made directly from one's own account without sending information on the account, and the user's right to use the application is verified more efficiently than in the case of a conventional bank card in which verification is based on a short PIN code.

In the examples above, it is assumed that a prior art point of sale terminal is used and no changes have been made in it. The benefit of such embodiments is that because the payer's and the payment receiver's devices need not support the same application, the uses are not restricted and the equipments need not support a plurality of applications to achieve a wide coverage. They additionally avoid the problems of applications operating on terminal devices, which enable the use of a mobile payment application of a bank for paying the invoice, if the invoice has been delivered with the same application that initiates the mobile payment application of the bank: the payer either needs to transfer money on his terminal device in advance, or register in the same service that transfers payments to the receiver as the receiver of the payment.

The examples above may also be applied when a change is performed in the point of sale terminal system, used to ensure in real time that a payment has been carried out also when the receiver's bank is different from the payer's bank.

In these example, too, shown by Figures 4, 5A and 5B, the problems described in the above are avoided, because with one application all the banks are covered, that is, an extensive coverage is achieved, money need not be transferred in advance, and there is no need for a separate registration. It should be noted that the application on the customer's terminal device does not necessarily have to be the same as the application (or the counterpart of an application pair) on the point of sale terminal system, but it is enough that the applications have enough common functions to ensure the functions described below.

The examples in Figures 4, 5A and 5B illustrate the functionality of the invoice management unit (or units). It is assumed in the example of Figure 4 that the verification is always requested, and in the example of Figures 5A and 5B it is assumed that verification is only requested if no real time information is available from one's own bank. Figure 4 also describes how a parser, for example, or another sub-unit of a transaction implementation unit in a user terminal device or transfer server is adapted to deliver the requested verification details. The same, of course, takes place in the example of Figures 5A and 5B although this is not separately shown. It is assumed in the examples that the application on the point of sale terminal (invoice management system) and the application on the terminal device/transfer server (such as a sub-unit of the transaction implementation unit) are configured to calculate a hash from the sum of the amount of money and the secret serial number of the application, and this way the application on the point of sale terminal may ensure that the application of the terminal device is unchanged (correct and genuine) and has just been used. In addition, in this example it is assumed that a single-use number, generated after the invoice formation by an application on the point of sale terminal, is added to the reference number, and from this sum another hash is calculated. This way, the freshness of the event is verified. It should be appreciated that these are just examples of how the confirmation may be performed. The essential thing is that the information on how the confirmation is performed is only at these trusted applications (or parsers).

It is assumed in the example of Figure 4 that no transfer server is used. Applying the example in a solution including the transfer server is obvious for a person skilled in the art. In addition, the information sent on the secure connection in the figure are encrypted before sending and decrypted before using, but for reasons of clarity the encryption and respectively decryption are not repeated in connection with information exchange.

Referring to Figure 4, the point of sale terminal offers a plurality of payment options for the customer, such as contactless payment. However, the customer chooses the direct payment application disclosed in this application. The point of sale terminal detects at 4-1 the choice of the direct payment application, forms an invoice and because the payment method is a direct payment method, pre-calculates at 4-1 the hash of the invoice sum (amount of money) and the secret serial number of the application. The point of sale terminal also conveys the invoice (message 4-2) electronically to the user terminal device by using, for example, the NFC technology as described in the above. After receiving the invoice, the terminal device carries out, together with the online bank, the measures relating to paying (at 4-3, messages 4-4). It is assumed in this example that the paying was successful and the message 4-5 and point 4-6 correspond with the message 3-42 and point 3-52 of Figure 3B. In this example, the terminal device, after compiling a receipt of the payment, sends the receipt in message 4-7 to the point of sale terminal and temporarily maintains the receipt in its memory. After receiving the receipt, the point of sale terminal is adapted to ensure that the correct payment application was used on the terminal device, and that is why the point of sale terminal sends a request in message 4-8 to the terminal device to calculate the hash. After calculating the hash (at 4-9) from the sum of the secret serial number of the direct payment application and the sum of money on the receipt the terminal device sends the hash to the point of sale terminal in message 4-10.

At 4-11, the point of sale terminal compares the hash received in message 4-10 with the hash it calculated at 4-1. If they are the same, the application is genuine. It is assumed in this example that they are the same. (If the application on the terminal device were not genuine, the point of sale terminal could inform the user of the payment failure, for example). Because the application was verified to be genuine, the point of sale terminal generates at 4-11 a single-use number x, calculates at 4-11 a hash of the single-use number x and of the reference number on the invoice formed at 4-1, and sends the single-use number x to the terminal device in message 4-12.

At 4-13, the terminal device calculates a hash from the sum of the single-use number x and from the reference number on the receipt and sends it to the point of sale terminal in message 4-14.

At 4-15, the point of sale terminal compares the hash received in message 4-14 with the hash it calculated at 4-11. If they are the same, the point of sale

terminal trusts that the invoice has been paid (transaction performed, compensation received) and shows at 4-15 the payment as having been carried out. If it is detected at 4-15 that the hashes are not the same, the payment is shown as having failed.

If a data transfer connection is not established between the point of sale terminal and the user terminal device, the applications may be arranged to show the corresponding details on the user interface, whereby the user of the point of sale terminal may compare the hash shown on the point of sale terminal with the hash the user terminal device shows, and both users may enter a single-use number chosen by, for example, the point of sale terminal user, in their respective devices (or the point of sale terminal user may tell/show the single-use number displayed on the screen of the point of sale terminal).

Figures 5A and 5B show an embodiment where the process described in the above is only performed if there is no real time confirmation from the bank that the payment was completed. In addition, it is assumed in the example of Figure 5A and 5B that the point of sale terminal is a part of a larger system, so Figure 5 describes the functionality of the point of sale terminal and Figure 5B the functionality in that part of the point of sale terminal system which gets real time information on account events. In connection with the description of Figures 5A and 5B, it is referred to as the point of sale system. In addition, the information sent on the secure connection in the figures is encrypted before sending and decrypted before using, but for reasons of clarity the encryption and respectively decryption are not repeated in connection with information exchange.

Referring to Figure 5A, the point of sale terminal generates at step 501 an invoice, which has a reference number, and sends it to the user terminal device at step 502 as explained above. In addition, the point of sale terminal sends at step 503 at least the reference number, possibly other information, too, to the point of sale system. After that, the point of sale terminal waits for a predetermined time t1 whether an acknowledgement is received from the point of sale system within the time t1 (step 504). The time t1 is a relatively short time, but sufficient to perform the transaction and to convey information on the performing of the transaction (paying of the payment).

If there is no acknowledgement received from the point of sale system within the time t1 on that the payment was made (step 504: no), the point of sale terminal calculates at step 505 a hash from the invoice sum (amount of money) and from the secret serial number of the application, and verifies at 506 the application

of the user terminal device as explained in Figure 4. In this example, too, it is assumed that the application was verified to be genuine. After that, the point of sale terminal generates at step 507 a single-use number and sends it to the terminal device. At the same time, the point of sale terminal calculates at step 508 a hash from the sum of the single-use number and the reference number, receives a hash from the terminal device, and compares them. The deduction made as a result of the comparison (same hashes - payment done, different hashes - payment failed) is shown at step 509 to the point of sale terminal user.

If an acknowledgement was received from the point of sale system within the time t1 (step 504: yes), at step 510 the point of sale terminal user is notified of a successful payment.

Referring to Figure 5B, the point of sale system receives at step 511 from the point of sale terminal at least the reference number and it monitors (step 512) whether it is acknowledged within the time t2 by a banking system that a payment has been done with the reference number. The time t2 is advantageously the same as that in Figure 5A, or slightly shorter. If the acknowledgement is received within the time t2 (step 512: yes), the point of sale system deduces at step 513, on the basis of the reference number in the acknowledgement of the banking system, the point of sale terminal to which to send at step 514 the information on the carrying out of the payment. If no acknowledgement is received within the time t2 (step 512: no), the point of sale system stops monitoring the reference number at step 515.

Other ways to ensure that the direct method payment has just been used may naturally be used instead of or in addition to the method described in the above. An example is that the calculation of the hash varies as the function of time, requiring that the clocks of the point of sale terminal and the user terminal device are accurately enough synchronized, such as at the interval of a few seconds, advantageously not more than a minute. Another example is that the application server is requested to select a hash calculation routine, which then either returns a hash or, for example, the seed of the routine or similar, by means of which the required hash is produced.

Although the examples in the above refer to a real time transaction relating to paying, it is obvious for a person skilled in the art that the above may be applied to any real time transaction. Figure 6 is a very simplified schematic view of a chart on the transfer of information or item in a transaction arrangement between the parties of the transaction.

Referring to Figure 6, a party 1 wanting an item shows or otherwise indicates to the item holder, that is, the receiver 2 of the compensation, the item 4 the party wants. The receiver 2 of the compensation on his part conveys to the party 1 wanting the item at least information on the compensation 5 and on where the compensation should be carried out (or allocated), that is, other information (of the receiver). Information on the compensation may also indicate the type of the transaction or the information on the type may be provided otherwise. The party 1 wanting the item conveys at least compensation and information on the receiver 6 to the performer 3 of the transaction, which after completing the transaction returns a confirmation 7 on the completed transaction to the party 1 wanting the item. This, on his part, shows or otherwise indicates the confirmation 8 to the receiver 2 of the compensation who gives or otherwise conveys 9 the item to the party 1 wanting the item. The performer 3 of the transaction sends information 10 on the transaction also to the receiver 2 of the compensation, i.e. to beneficiary of the compensation, who depending on data transfer delays and/or embodiment may receive information on it immediately or after a while. If the system is such that the information/confirmation is conveyed in real time to the receiver of the compensation, the party desiring the item need not show or indicate the confirmation.

When the examples described in the above are used, an item means a purchase or purchases, the acts of the party 1 wanting the item are performed by the buyer's terminal device or a combination of the buyer's terminal device and a transfer server, a transaction is the transfer of a particular sum of money from a buyer to a seller, the receiver 2 of the sum of money, is a shop, the compensation 5 is presented on an invoice, the information 6 includes, among other things, the details of the shop and amount of money to the banking system, which is the performer 3 of the transaction. As a confirmation 7, it returns a verification of the payment and as information 10 information on the amount of money received on the account.

In another example, the item means a loan, the acts of the party 1 wanting the item are performed by a borrower's terminal device, the receiver 2 of the transaction is a bank or another lender, the compensation 5 is an information on a collateral, in the information 6 the information on the bank and the collateral are carried to the loan system, which is the performer 3 of the transaction. As a confirmation 7, it returns a verification of the collateral and as information 10 information on the collateral lodged.

In another example, the item means a material to be borrowed from a library, the acts of the party 1 wanting the item is the borrower's terminal device, the receiver 2 of the transaction is the library, the compensation 5 is the identification information of the material associated with the borrower's library identifier, carried in the information 6. The performer 3 of the transaction is the library, which returns as the confirmation 7 a receipt on the borrowing and as information 10 the borrowing information.

In the above, the steps/points, messages and related operations described in connection with Figures 2-6 are not in an absolute chronological order and some steps/points may be carried out simultaneously or deviating from a given order and/or omitted. For example, the user authentication may be performed later than in the examples above, as long as it is done latest in connection with approving the compensation, as part of approving the compensation, for example. Other functions may also be performed between the steps/points or within the steps/points and other messages may be sent between the illustrated messages. For example, to ensure a payment and/or performing a payment, a plurality of challenge responses may be needed. Another example is that the user terminal device beeps when a payment has been successfully performed. An example is that the user may be displayed the selection of just Figure 7C out of the Figures 7A-7C, or not even that. An attempt may be made to solve a failed transaction or another error situation also automatically, by re-requesting information, for example. Yet another example is that the transfer server may retrieve balance information on accounts in advance, for example as background processing at specific intervals, or when a user authenticates himself on the transfer server, and may show the user the account together with their balances, or only the accounts having sufficient funds for the payment. The functions described illustrate procedures on the devices in question, and they may be implemented in one or more physical or logical entity. The messages are exemplary, only, and may even comprise a plurality of individual messages to send the same information. The messages may also include other information than the information described in the above. Other messages, too, may be sent. For example, if the point of sale terminal has a connection to the transfer server, the point of sale terminal may send an invoice which it either shows on its screen or conveys it to the user terminal device, also to the transfer server. The transfer server may be arranged to associate this invoice received from the point of sale terminal and the transaction request obtained from the terminal device on the basis of a reference number, for example, and once the invoice has

been paid, to send information on successful payment to the point of sale terminal, too, whereby the point of sale terminal receives the information directly from a reliable source and other confirmations are not needed. For this purpose, the transfer server may comprise a third counterpart of an encryption key wallet (an encryption key wallet between the point of sale system and transfer server). Other types of server solutions may be used, too.

Based on the examples described above, it is clear that a real time transaction succeeds in a secure manner.

The terminal device or transfer server or point-of-sale system or a similar device (devices) or their combination(s), implementing one or more embodiments described above in connection with Figures 2-6, or a functionality described in connection with a separate example, comprises in addition to prior art means also means for implementing one or more functionalities of a terminal device, transfer server described in connection with an embodiment/example, or a unit described in connection with them, and it may comprise separate means for each of the different functions or the means may be configured to implement two or more functions. A terminal device and/or transfer server may be configured as a computer or microprocessor, such as an element integrated on one microcircuit, which comprises at least memory to provide a storage area used by an arithmetic operation, and an operating processor to perform the arithmetic operation.

Figure 8 is a simplified block diagram illustrating some units, which a device 800 which is configured to be a user terminal device, or at least comprise a storing unit and/or a transaction implementation unit or its sub-unit or similar unit which is adapted to carry out similar functionalities or part of the functionalities described in connection with Figures 2-4. In the example of Figure 8, the device 800 comprises one or more interfaces (IFs) 801 in order to be in data transfer communication with other devices, such as the devices of the performing system of the transaction and/or the target devices of the transaction, one or more processors (PROC.) 802 configured to carry out the functionality of the transaction implementation unit and/or the storing unit and/or data transfer unit or that of at least one of its sub-units with the corresponding algorithm(s) (ALG.) 803 and memory (MEM) 804 to be used for storing at least the algorithm(s) or a similar group of operational instructions, and to provide a secure memory for the information stored therein, and further one or more user interfaces (U-IF) 801' for interaction with the user.

Figure 9 is a simplified block diagram illustrating some units, which a device 900 which is configured to be a transfer server, or at least comprise a transaction data transfer unit or a sub-unit of it or similar unit which is adapted to carry out similar functionalities or part of the functionalities described in connection with Figures 2-4. In the example of Figure 9, the device 900 comprises one or more interfaces (IFs) 901 in order to be in data transfer communication with other devices, such as the devices of the performing system of the transaction and/or the target devices of the transaction, one or more processors (PROC.) 902 configured to carry out the functionality or the transaction data transfer unit or that of at least one of its sub-units with the corresponding algorithm(s) 903, and memory (MEM) 904 to be used for storing at least the algorithm(s) or a similar group of operating instructions. It should be understood that the transfer server may be implemented as a decentralised server or cloud server.

Figure 10 is a simplified block diagram illustrating some units, which equipment 1000 of the receiver (beneficiary) of the compensation, which is configured to comprise an invoice management unit or a sub-unit of it or a similar unit which is adapted to carry out either together or as a combination or a plurality of devices/invoice management units the functionalities or part of the functionalities described in connection with Figures 4, 5A, and 5B. In the example of Figure 10, the device 1000 comprises one or more interfaces (IFs) 1001 in order to be in data transfer communication with other devices, such as the user terminal device and the other devices of the equipment, one or more processors (PROC.) 1002 configured to carry out the functionality of the invoice management unit or that of a sub-unit thereof with the corresponding algorithm(s) (ALG.) 1003, and memory (MEM) 1004 to be used for storing at least the algorithm(s) or a similar group of operating instructions. Part of the memory may be secure memory. It should be understood that the equipment may also be implemented by utilising a decentralised server or cloud server.

The processor 802, 902, 1002 may be a central processing unit, operational processor, logic circuit, ASIC (application-specific integrated circuit), and a re-programmable digital logic circuit FPGA (field-programmable gate array).

The memory 804, 904, 1004 may be centralised or decentralised memory and it may be implemented with any memory technology or a combination thereof, such as semiconductor memory, flash memory, magnetic memory and

optical memory. The memory may also be a fixed part of the device 800, 900, equipment 1000 or a combination of devices, or a detachable memory or a memory external to the device.

In an embodiment, a computer program is provided which is included on any processor to computer-readable data storage means and which comprises program instructions which, once downloaded to a device including a processor, are part of the transaction implementation unit and/or storing unit, and/or transaction data transfer unit and or enhanced transaction implementation unit and/or invoice management unit.

It should be understood that other units, too, may be implemented by means of various circuits/processors, micro software/pieces of micro software, and/or software/pieces of software.

A person skilled in the art will find it obvious that, as technology advances, the basic idea of the invention may be implemented in many different ways. The invention and its embodiments are thus not restricted to the above-described examples but may vary within the scope of the claims.

Claims

1. A method for implementing a real time transaction, the method comprising

maintaining in a memory, for one or more real time transaction performing systems, correspondingly one or more parsers for creating real time transaction-related requests according to the settings of the real time transaction performing system and for identifying the content of the responses according to the settings of the real time transaction performing system;

maintaining in a secure memory, in an encrypted format, transaction requester –specific information required for carrying out the real time transaction by a real time transaction performing system to carry out the compensation from a transaction requester's source, the transaction requester –specific information comprising details that are different from information identifying the source;

receiving (202, 3-3, 5), for carrying out a real time transaction, a compensation required of the consideration and receiver information on a receiver of the compensation, the receiver information comprising at least information whereto carry out the compensation;

verifying (201, 3-1, 3-2) the right of the transaction requester to use the encrypted transaction requester –specific information needed for carrying out the real time transaction from the source;

if the transaction requester has the right to use the encrypted transaction requester –specific information needed for carrying out the real time transaction from the source, the method further comprises:

decrypting in the secure memory at least part of the encrypted transaction requester -specific information needed for carrying out the real time transaction;

establishing (3-5, 3-6) an unsecure or a secure connection to the real time transaction performing system;

forming (3-29), by using the parser for the real time transaction performing system, a transaction request according to the settings of the real time transaction performing system, the transaction request including the compensation, the information on the receiver of the compensation, and at least one detail from the decrypted part of the transaction requester -specific encrypted information required for carrying out the transaction and maintained in the secure memory;

sending (3-30, 6) the transaction request on a secure connection or as encrypted on an unsecure connection to the real time transaction performing system; and

receiving (3-43, 3-42, 7, 10), from the real time transaction performing system, a transaction response indicating whether the transaction request was successfully carried out.

2. A method as claimed in claim 1, which further comprises, if the transaction requester has the right to use the encrypted transaction requester –specific information needed for carrying out the real time transaction from the source:

receiving (3-34) from the real time transaction performing system a request for further information comprising a request for one of more specific details according to the settings of the real time transaction performing system;

identifying (3-35), by using the parser, the requested one or more specific details;

retrieving from the decrypted part of encrypted transaction requester –specific information required for carrying out the real time transaction the requested one or more specific details;

forming (3-39), by using the parser, a response according to the settings of the real time transaction performing system, the response including the requested one or more specific details;

sending (3-40) the response on the secure connection or as encrypted on the unsecure connection to the real time transaction performing system.

3. A method as claimed in claim 2, which further comprises the decryption of the requested one or more specific details if they are still encrypted.

4. A method as claimed in any one of the preceding claims, further comprising:

maintaining, for two or more real time transaction performing systems, a parser for each performing system for creating transaction-related requests according to the settings of the real time transaction performing system and for identifying the content of the responses according to the settings of the real time transaction performing system;

selecting (3-4) a real time transaction performing system from a group of two or more real time performing systems; and

using the parser of the selected real time transaction performing system.

5. A method as claimed in any one of the preceding claims, further comprising:

requesting (209), through the user interface, a confirmation for the real time transaction from the transaction requester; and

sending the real time transaction request only if a confirmation is obtained through the user interface.

6. A method as claimed in any one of the preceding claims, further comprising, if a real time transaction may be performed from a plurality of sources:

requesting (206), through the user interface, the transaction requester to select the source; and

establishing an unsecure or a secure connection to the real time transaction performing system where the selected source is.

7. A method as claimed in any one of the preceding claims, further comprising:

if the transaction request was carried out successfully, sending (3-48, 10) information on the succeeding of the real time transaction also to the device from which the compensation and receiver information of the compensation were received.

8. A method as claimed in any one of the preceding claims, in which a transaction request sent on a secure connection includes at least one detail kept encrypted in the secure memory in an unencrypted format.

9. A method as claimed in any one of the preceding claims, in which the compensation and receiver information of the compensation are received (202, 3-3, 5) in an unencrypted format.

10. A method as claimed in any one of the preceding claims, further comprising:

if the transaction request was carried out successfully, sending (3-42) information on the succeeding of the real time transaction also to the place the receiver information of the compensation indicates.

11. Computer program product, which when run on a computer causes the computer to perform a method as claimed in any one of the preceding claims 1-9.

12. A device (120, 120a, 140) comprising at least

data transfer means (123, 143) for communication with a real time transaction performing system ;

characterized in that the device further comprises:

memory (122, 142) in which at least one parser has been stored for creating requests according to the settings of the real time transaction performing system and identifying the content of responses according to the settings of the real time transaction performing system, and of which memory at least a part is secure memory (122-3) on which transaction requester –specific details needed in the real time transaction are stored; and

transaction means (122-1, 122-1a, 141) for performing a method as claimed in any one of claims 1-9.

13. A device as claimed in claim 12 (120, 120a, 140), configured to verify the transaction means before performing a method as claimed in any one of claims 1-9.

14. A device (120, 120a, 140) as claimed in claim 12 or 13, configured to verify that a method as claimed in any one of claims 1-8 has been performed with a valid application and within a particular time period.

15. A device (120, 120a, 140) as claimed in claim 12, 13 or 14, which is a user terminal device, which further comprises a user interface for conveying information to the user and for receiving inputs from the user.

16. A device (140) as claimed in claim 12, 13 or 14, which is a transfer server in which the data transfer means are adapted to communicate with user terminal devices over a secure connection or over an unsecure connection by using encrypted information.

17. A device (140) which comprises at least

data transfer means (143) for communication with a user terminal device (120) and a real time transaction performing system ;

characterized in that

the device further comprises for each real time transaction performing system a parser (142-2) for creating transaction-related requests according to settings of the real time transaction performing system and for identifying content in responses according to the settings of the real time transaction performing system;

the device (140) being adapted, upon receiving from the user terminal device a compensation, information on the receiver of the compensation, the information comprising at least information whereto carry out the compensation, and information on a chosen real time transaction performing system, which the user of the terminal device wishes to use to perform the compensation to the receiver, to select a parser corresponding to an electronic interface of the transaction performing system, establish a secure or unsecure connection to the real time transaction performing system (130), to retrieve either from the user terminal device or the device memory at least one other user-specific detail needed to carry out the real time transaction from the user's source, the at least one other user-specific detail being different from information identifying the source, wherefrom the transaction is to be carried out, by means of the at least one other user-specific detail the transaction performing system verifies the user's right to carry out the compensation, to generate a transaction request according to settings of the real time transaction performing system, which transaction request includes the compensation, information on the receiver of the compensation, and at least one detail needed to carry out the transaction from the user's source, the at least one detail being different from the information identifying the source, to send the transaction request to the real time transaction performing system (130) on a secure connection or as encrypted on an unsecure connection, and to receive a transaction response which indicates whether the transaction request was performed successfully.

20175372 PRH 28-02-2019

18. A device (140) as claimed in claim 17, which is adapted, when a secure connection is used, to generate the transaction request so that at least one user-specific detail needed to perform the transaction is unencrypted.

19. A system (100) comprising:

at least one user terminal device (120);

at least one transfer server (140); and

at least one real time transaction performing system (130);

means (122-2) for maintaining transaction requester -specific information, needed for carrying out a transaction, in an encrypted format either on the user terminal device or transfer server and temporarily decrypt at least part of the information needed for carrying out the transaction;

in which system (100)

the user terminal device (120) includes in a secure memory (122) a real time transaction application (122-1) and at least transaction requester -specific information (122-3) that may be used to verify the user's right to use the application, the user terminal device being adapted to convey, after a successful user verification, at least information on the compensation required for the consideration and on the receiver of the compensation to the transfer server, the information on the receiver of the compensation comprising at least information whereto carry out the compensation;

characterized in that

the transfer server (140) is adapted to be in communication with the user terminal device (120) and the real time transaction performing system (130) and comprises a parser (142-2) for the real time transaction performing system for generating transaction-related requests according to the settings of the real time transaction performing system and for identifying the content in the responses according to the settings of the real time transaction performing system, the transfer server (140) being adapted, upon receiving from the user terminal device the compensation required for the consideration and information on the receiver of the compensation, to establish a secure or unsecure connection to the real time transaction performing system (130), to retrieve at least one other transaction requester -specific detail needed for carrying out the transaction in addition to the required compensation and the receiver information of the compensation, the transaction requester –specific detail being different from information identifying a source wherefrom the transaction is to be carried out, to generate a transaction

request according to settings of the real time transaction performing system, which transaction request includes the compensation, information on the receiver, and the retrieved at least one transaction requester –specific detail needed for carrying out the transaction, to send the transaction request to the real time transaction performing system (130) on a secure connection or as encrypted on an unsecure connection, and to receive the transaction response which indicates whether the transaction request was performed successfully.

20. A system (100) as claimed in claim 19, in which

the transfer server (141) is adapted to receive from the real time transaction performing system (130) one or more credential requests, retrieve the requested credential or credentials and send them to the real time transaction performing system (130) on a secure connection or in an encrypted format on an unsecure connection.

21. A system (100) as claimed in claim 19 or 20, comprising at least two different real time transaction performing systems (130); and

the transfer server (140) comprises at least for each real time transaction performing system a parser (142-2) for generating requests and responses according to the settings of the real time transaction performing system, the transfer server being adapted to select the parser on the basis of the chosen real time transaction performing system.

22. A system (100) as claimed in claim 19, 20 or 21, in which the real time transaction performing systems (130) is an online banking system.

23. A system (100) as claimed in claim 19, 20, 21, or 22, in which the transfer server (140) is adapted to generate a transaction request so that at least one of the details is unencrypted.

Patenttivaatimukset

1. Menetelmä reaaliaikaisen transaktion toteuttamiseksi, joka menetelmä käsittää:

ylläpidetään muistissa yhtä tai useampaa reaaliaikaista transaktiosuoritusjärjestelmää varten vastaavasti yhtä tai useampaa parseria reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten transaktioon liittyvien pyyntöjen luomiseksi ja reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten vastausten sisällön tunnistamiseksi;

ylläpidetään suojatussa muistissa, salatussa formaatissa, transaktion pyytäjäkohtaisia tietoja, joita tarvitaan reaaliaikaisen transaktion suorittamiseksi reaaliaikaisella transaktiosuoritusjärjestelmällä vastike transaktion pyytäjän lähteestä, transaktion pyytäjäkohtaisten tietojen käsittäessä tietoja, muita kuin lähteen identifioiva tieto;

vastaanotetaan (202, 3-3, 5) reaaliaikaisen transaktion suorittamiseksi maksusta vaadittava vastike ja vastikkeen saajan vastaanottajatietoja, vastaanottajatietojen käsittäessä ainakin tiedon, minne vastike suoritetaan;

varmennetaan (201, 3-1, 3-2) transaktion pyytäjän oikeus käyttää transaktion suorittamiseen lähteestä tarvittavia salattuja pyytäjäkohtaisia tietoja;

jos transaktion pyytäjällä on oikeus käyttää reaaliaikaisen transaktion suorittamiseen lähteestä tarvittavia salattuja pyytäjäkohtaisia tietoja, menetelmä käsittää lisäksi:

puretaan suojatussa muistissa salaus ainakin osasta salatuista pyytäjäkohtaisista transaktion suorittamiseen tarvittavista tiedoista;

muodostetaan (3-5, 3-6) suojaamaton tai suojattu yhteys reaaliaikaiseen transaktiosuoritusjärjestelmään;

muodostetaan (3-29) reaaliaikaisen transaktiosuoritusjärjestelmän parseria käyttäen reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukainen transaktiopyyntö, joka sisältää vastikkeen, vastikkeen saajan tietoja ja ainakin yhden transaktion suorittamiseen tarvittavan tiedon pyytäjäkohtaisten salattujen transaktion suorittamiseen tarvittavien ja suojatussa muistissa ylläpidettyjen tietojen salauksesta puretusta osasta;

lähetetään (3-30, 6) transaktiopyyntö suojatulla yhteydellä tai salattuna suojaamattomalla yhteydellä reaaliaikaiseen transaktiosuoritusjärjestelmään; ja

vastaanotetaan (3-43, 3-42, 7, 10) reaaliaikaiselta transaktiosuoritusjärjestelmältä transaktiovastaus, joka osoittaa, suoritettiinko transaktiopyyntö onnistuneesti.

2. Patenttivaatimuksen 1 mukainen menetelmä, joka lisäksi käsittää, jos transaktion pyytäjällä on oikeus käyttää reaaliaikaisen transaktion suorittamiseen lähteestä tarvittavia salattuja pyytäjäkohtaisia tietoja:

vastaanotetaan (3-34) reaaliaikaiselta transaktiosuoritusjärjestelmältä lisätietopyyntö, joka käsittää yhden tai useamman tietyn tiedon pyynnön reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisesti;

tunnistetaan (3-35) parseria käyttäen pyydetyt yksi tai useampi tietty tieto;

haetaan reaaliaikaisen transaktion suorittamiseen tarvittavien salattujen pyytäjäkohtaisten tietojen salauksesta puretusta osasta pyydetyt yksi tai useampi tietty tieto;

muodostetaan (3-39) parseria käyttäen reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukainen vastaus, joka sisältää pyydetyt yhden tai useamman tietyn tiedon;

lähetetään (3-40) vastaus suojatulla yhteydellä tai salattuna suojaamattomalla yhteydellä reaaliaikaiselle transaktiosuoritusjärjestelmälle.

3. Patenttivaatimuksen 2 mukainen menetelmä, joka lisäksi käsittää pyydetyn yhden tai useamman tiedon salauksen purkamisen, jos ne ovat vielä salattuja.

4. Jonkin edellisen patenttivaatimuksen mukainen menetelmä, joka lisäksi käsittää:

ylläpidetään kahta tai useampaa reaaliaikaista transaktiosuoritusjärjestelmää varten kullekin transaktiosuoritusjärjestelmälle parseria reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten transaktioon liittyvien pyyntöjen luomiseksi ja vastausten sisällön tunnistamiseksi;

valitaan (3-4) transaktion reaaliaikainen transaktiosuoritusjärjestelmä kahden tai useamman reaaliaikaisen transaktiosuoritusjärjestelmän joukosta; ja

käytetään valitun transaktion reaaliaikaisen transaktiosuoritusjärjestelmän parseria.

5. Jonkin edellisen patenttivaatimuksen mukainen menetelmä, joka lisäksi käsittää:

pyydetään (209) käyttöliittymän välityksellä transaktion pyytäjältä vahvistus reaaliaikaiselle transaktiolle; ja

lähetetään reaaliaikainen transaktiopyyntö vain, jos käyttöliittymän kautta saadaan vahvistus.

6. Jonkin edellisen patenttivaatimuksen mukainen menetelmä, joka lisäksi käsittää, jos reaaliaikainen transaktio voidaan suorittaa useammasta eri lähteestä:

pyydetään (206) käyttöliittymän välityksellä transaktion pyytäjää valitsemaan lähde; ja

muodostetaan suojaamaton tai suojattu yhteys siihen reaaliaikaiseen transaktiosuoritusjärjestelmään, jossa valittu lähde on.

7. Jonkin edellisen patenttivaatimuksen mukainen menetelmä, joka lisäksi käsittää:

jos transaktiopyyntö suoritettiin onnistuneesti, lähetetään (3-48, 10) tieto reaaliaikaisen transaktion onnistumisesta myös laitteelle, jolta vastaanotettiin vastike ja vastikkeen saajan tietoja.

8. Jonkin edellisen patenttivaatimuksen mukainen menetelmä, jossa suojatulla yhteydellä lähetetty transaktiopyyntö sisältää ainakin yhden tiedon, jota pidetään salattuna suojatussa muistissa salaamattomassa muodossa.

9. Jonkin edellisen patenttivaatimuksen mukainen menetelmä, jossa vastike ja vastikkeen saajan tietoja vastaanotetaan (202, 3-3, 5) salaamattomassa muodossa.

10. Jonkin edellisen patenttivaatimuksen mukainen menetelmä, joka lisäksi käsittää:

jos transaktiopyyntö suoritettiin onnistuneesti, lähetetään (3-42) tieto reaaliaikaisen transaktion onnistumisesta myös vastikkeen saajan tietojen osoittamaan paikkaan.

11. Tietokoneohjelmistotuote, joka tietokoneella ajettuna aikaansaa tietokoneen suorittamaan jonkin edellisen patenttivaatimuksen 1-9 mukaisen menetelmän.

12. Laite (120,120a, 140), joka käsittää ainakin

tiedonsiirtovälineet (123, 143) reaaliaikaisen transaktiosuoritusjärjestelmän kanssa kommunikoimiseksi;

t u n n e t t u siitä, että laite käsittää lisäksi:

muistia (122, 142), jonne on tallennettu ainakin yksi parseri reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten pyyntöjen luomiseksi ja reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten vastausten sisällön tunnistamiseksi, ja josta muistista ainakin osa on suojattua muistia (122-3), jonne on tallennettu reaaliaikaisessa transaktiossa tarvittavia transaktion pyytäjäkohtaisia tietoja; ja

transaktiovälineet (122-1, 122-1a, 141) jonkin patenttivaatimuksen 1-9 mukaisen menetelmän suorittamiseksi.

13. Patenttivaatimuksen 12 mukainen laite (120, 120a, 140), joka on konfiguroitu varmentamaan transaktiovälineet ennen jonkin patenttivaatimuksen 1-9 mukaisen menetelmän suorittamista.

14. Patenttivaatimuksen 12 tai 13 mukainen laite (120, 120a, 140), joka on konfiguroitu varmentamaan, että jonkin patenttivaatimuksen 1-8 mukainen menetelmä on suoritettu validilla sovelluksella ja tietyn ajan sisällä.

15. Patenttivaatimuksen 12, 13 tai 14 mukainen laite (120, 120a, 140), joka on käyttäjän päätelaite, joka lisäksi käsittää käyttöliittymän tietojen välittämiseksi käyttäjälle ja syötteiden vastaanottamiseksi käyttäjältä.

16. Patenttivaatimuksen 12, 13 tai 14 mukainen laite (140), joka on siirtopalvelin, jossa tiedonsiirtovälineet on sovitettu kommunikoimaan käyttäjien päätelaitteiden kanssa suojatun yhteyden yli tai salattua tietoa käyttäen suojaamattoman yhteyden yli.

17. Laite (140), joka käsittää ainakin

tiedonsiirtovälineet (143) käyttäjän päätelaitteen (120) ja reaaliaikaisen transaktiosuoritusjärjestelmän kanssa kommunikoimiseksi;

t u n n e t t u siitä, että:

laite käsittää lisäksi kutakin reaaliaikaista transaktiosuoritusjärjestelmää varten parserin (142-2) reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten transaktioihin liittyvien pyyntöjen luomiseksi ja reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten vastausten sisällön tunnistamiseksi;

laitteen (140) ollessa sovitettu vastaanotettuaan käyttäjän päätelaitteelta vastikkeen, vastikkeen saajan tietoja, tietojen käsittäessä ainakin tiedon, minne vastike suoritetaan, ja tiedon valitusta reaaliaikaisesta transaktiosuoritusjärjestelmästä, jota päätelaitteen käyttäjä haluaa käyttää vastineen suorittamiseksi saajalle, valitsemaan transaktiosuoritusjärjestelmän sähköistä rajapintaa vastaavan parserin, muodostamaan suojattu tai suojaamaton yhteys reaaliaikaiseen transaktiosuoritusjärjestelmään (130), hakemaan käyttäjän päätelaitteesta tai laitteen muistista ainakin yhden muun, käyttäjäkohtaisen tarvittavan tiedon transaktion suorittamiseen käyttäjän lähteestä, joka ainakin yksi muu käyttäjäkohtainen tieto on eri kuin lähteen, josta transaktio suoritetaan, identifioiva tieto, jonka ainakin yhden muun käyttäjäkohtaisen tiedon avulla transaktiosuoritusjärjestelmä varmentaa käyttäjän oikeuden vastikkeen suorittamiseen, muodostamaan reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisen transaktiopyynnön, joka sisältää vastikkeen, vastikkeen saajan tietoja ja ainakin yhden transaktion suorittamiseen käyttäjän lähteestä tarvittavan tiedon, joka on eri kuin lähteen, josta transaktio suoritetaan, identifioiva tieto, lähettämään transaktiopyyntö reaaliaikaiseen transaktiosuoritusjärjestelmään (130) suojatulla yhteydellä tai salattuna suojaamattomalla yhteydellä ja vastaanottamaan transaktiovastaus, joka osoittaa, suoritettiinko transaktiopyyntö onnistuneesti.

18. Patenttivaatimuksen 17 mukainen laite (140), joka on sovitettu, kun käytetään suojattua yhteyttä, muodostamaan transaktiopyyntö niin, että ainakin yksi käyttäjäkohtainen tieto, jota tarvitaan transaktion suorittamiseen, on salaamaton.

19. Järjestelmä (100), joka käsittää

ainakin yhden käyttäjän päätelaitteen (120);

ainakin yhden siirtopalvelimen (140); ja

ainakin yhden transaktion reaaliaikaisen transaktiosuoritusjärjestelmän (130);

välineitä (122-2) ylläpitämään transaktion suorittamiseen tarvittavia, transaktion pyytäjäkohtaisia tietoja salattuna joko käyttäjän päätelaitteessa tai siirtopalvelimessa ja purkamaan tilapäisesti salaus ainakin osasta transaktion suorittamiseen tarvittavia tietoja;

jossa järjestelmässä (100)

käyttäjän päätelaite (120) sisältää suojatussa muistissa (122) reaaliaikaisen transaktion sovelluksen (122-1) ja ainakin transaktion pyytäjäkohtaisia tietoa (122-3), joita voidaan käyttää varmentamaan käyttäjän oikeus käyttää sovellusta, käyttäjän päätelaitteen ollessa sovitettu välittämään käyttäjän varmennuksen onnistumisen jälkeen siirtopalvelimelle ainakin maksusta vaadittavan vastikkeen ja vastikkeen saajan tietoja, jotka käsittävät ainakin tiedon, mihin vastike suoritetaan,

t u n n e t t u siitä, että

siirtopalvelin (140) on sovitettu olemaan yhteydessä käyttäjän päätelaitteeseen (120) ja reaaliaikaiseen transaktiosuoritusjärjestelmään (130) ja käsittää reaaliaikaista transaktiosuoritusjärjestelmää varten parserin (142-2) reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten transaktioon liittyvien pyyntöjen luomiseksi ja reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten vastausten sisällön tunnistamiseksi, siirtopalvelimen (140) ollessa sovitettu vastaanotettuaan käyttäjän päätelaitteelta maksusta vaadittavan vastikkeen ja vastikkeen saajan tietoja muodostamaan suojattu tai suojaamaton yhteys reaaliaikaiseen transaktiosuoritusjärjestelmään (130), hakemaan vaaditun vastikkeen ja saajan tietojen lisäksi ainakin yhden transaktion pyytäjäkohtaisen transaktion suorittamiseen tarvittavan käyttäjäkohtaisen tiedon, käyttäjäkohtaisen tiedon ollessa muu kuin lähteen, josta transaktio suoritetaan, identifioiva tieto, muodostamaan reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisen transaktiopyynnön, joka sisältää vastikkeen, saajan tietoja ja haetun, ainakin yhden transaktion suorittamiseen tarvittavan transaktion pyytäjäkohtaisen tiedon, lähettämään transaktiopyynnön reaaliaikaiseen transaktiosuoritusjärjestelmään (130) suojatulla yhteydellä tai salattuna suojaamattomalla yhteydellä ja vastaanottamaan transaktiovastauksen, joka osoittaa, suoritettiinko transaktiopyyntö onnistuneesti.

20. Patenttivaatimuksen 19 mukainen järjestelmä (100), jossa

siirtopalvelin (140) on sovitettu vastaanottamaan reaaliaikaiselta transaktiosuoritusjärjestelmältä (130) yhden tai useamman tunnuspyynnön, hakemaan pyydetyn tunnuksen tai tunnukset ja lähettämään ne transaktion reaaliaikaiseen transaktiosuoritusjärjestelmään (130) suojatulla yhteydellä tai salatussa muodossa suojaamattomalla yhteydellä.

21. Patenttivaatimuksen 19 tai 20 mukainen järjestelmä (100), joka käsittää ainakin kaksi eri transaktion reaaliaikaista transaktiosuoritusjärjestelmää (130); ja

siirtopalvelin (140) käsittää ainakin kutakin reaaliaikaista transaktiosuoritusjärjestelmää varten parserin (142-2) reaaliaikaisen transaktiosuoritusjärjestelmän asetusten mukaisten pyyntöjen ja vastausten luomiseksi, siirtopalvelimen ollessa sovitettu valitsemaan parserin valitun reaaliaikaisen transaktiosuoritusjärjestelmän perusteella.

22. Patenttivaatimuksen 19, 20 tai 21 mukainen järjestelmä (100), jossa transaktion reaaliaikainen transaktiosuoritusjärjestelmä (130) on verkkopankkijärjestelmä.

23. Patenttivaatimuksen 19, 20, 21 tai 22 mukainen järjestelmä (100), jossa siirtopalvelin (140) on sovitettu muodostamaan transaktiopyyntö niin, että ainakin yksi tiedoista on salaamaton.

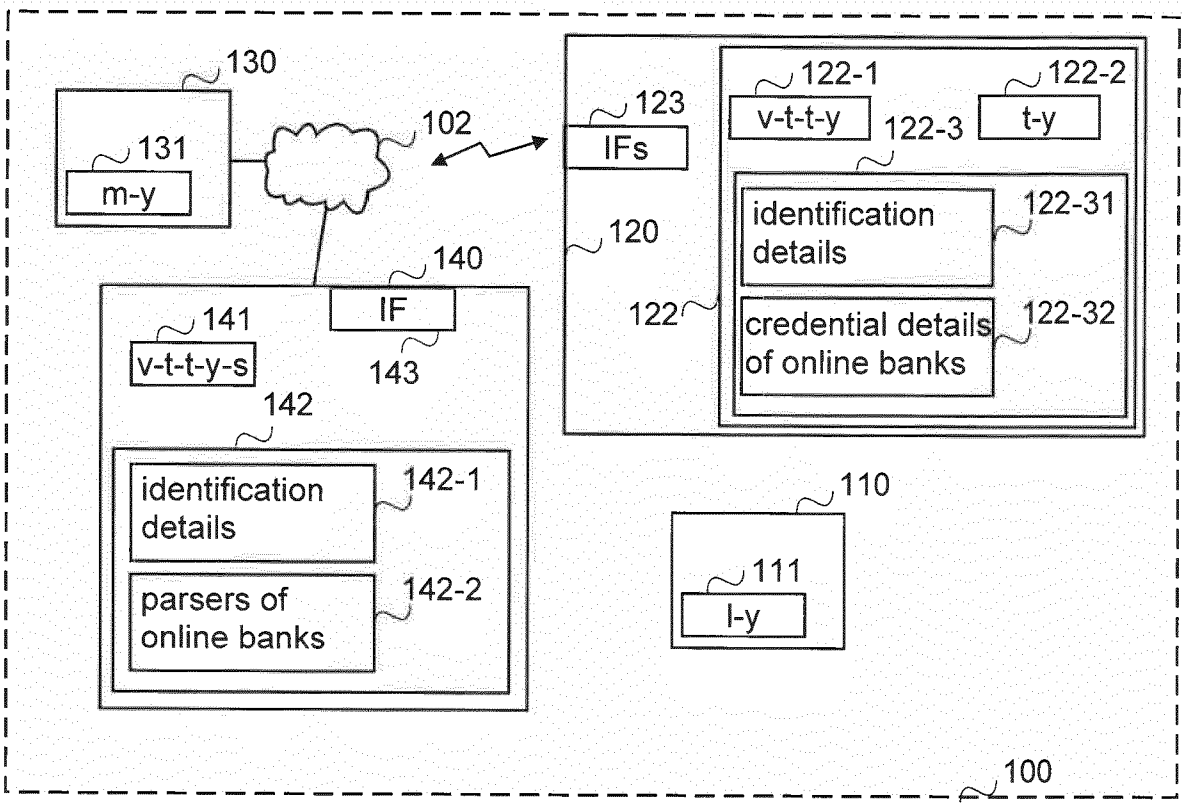


FIG.1A

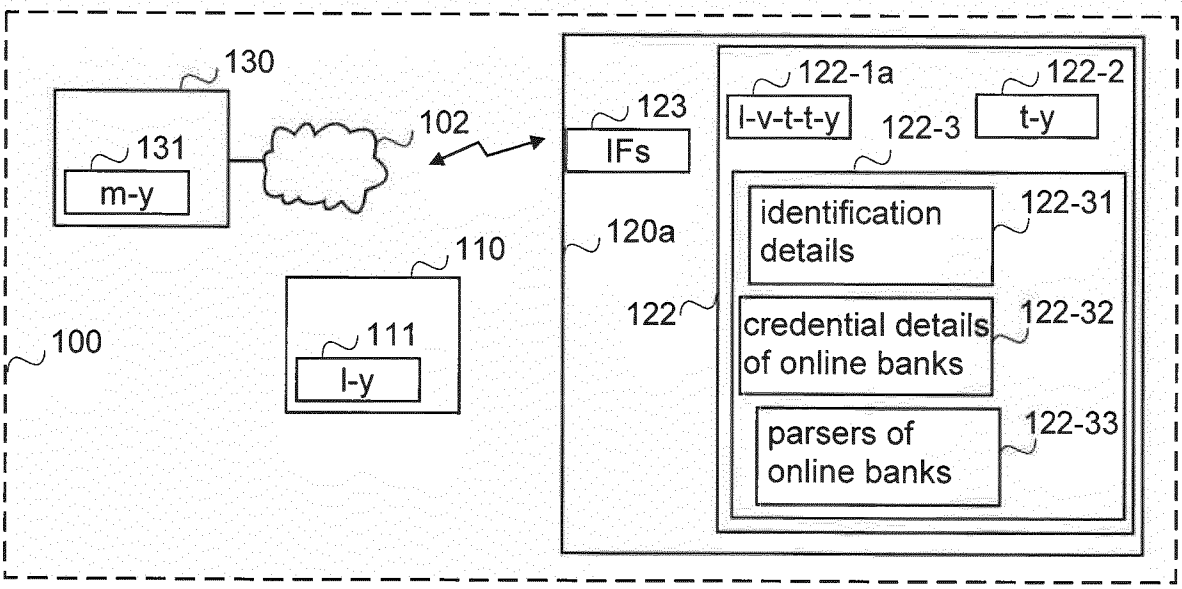


FIG.1B

20175372 PRH 25-04-2017

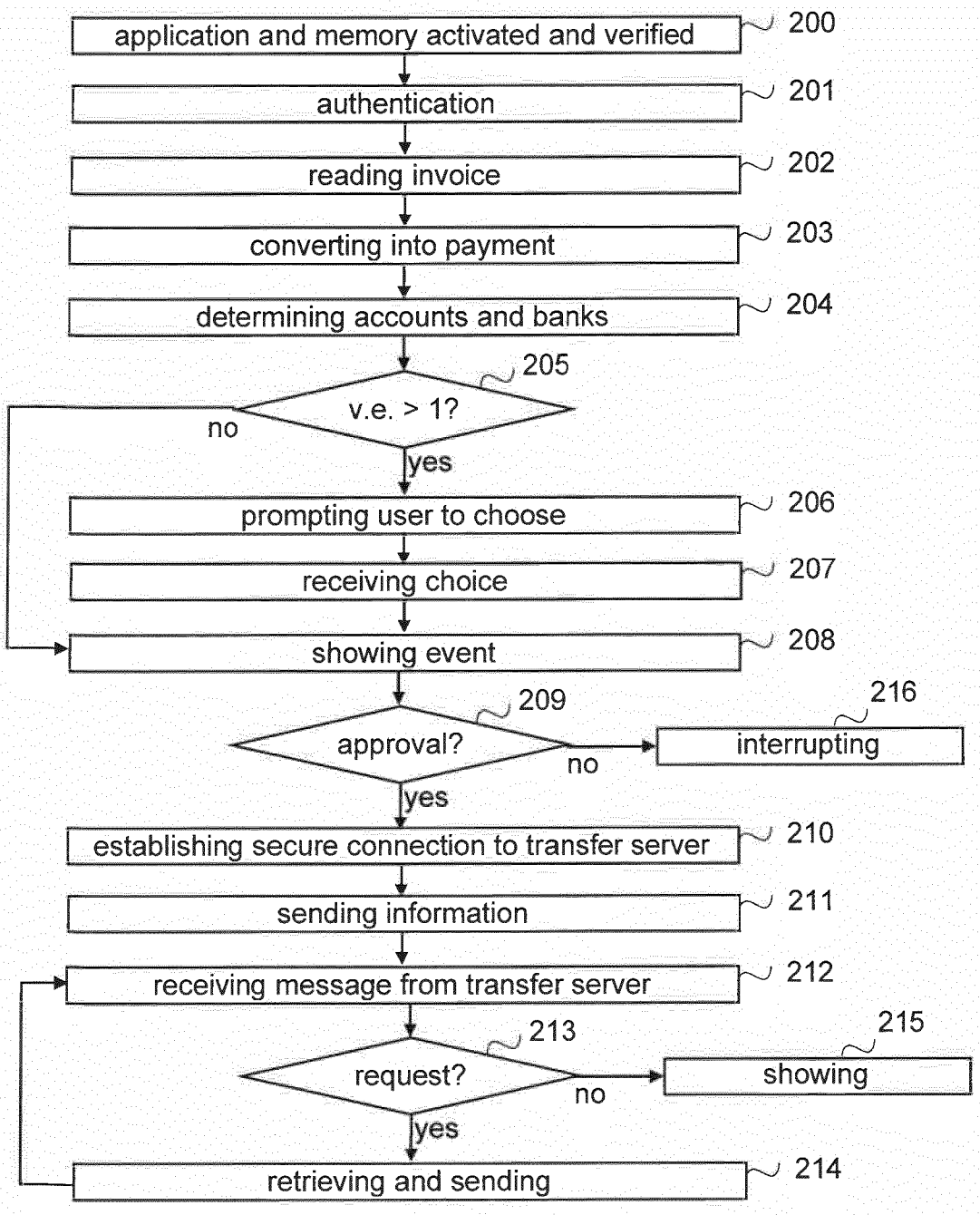


FIG.2A

20175372 PRH 25-04-2017

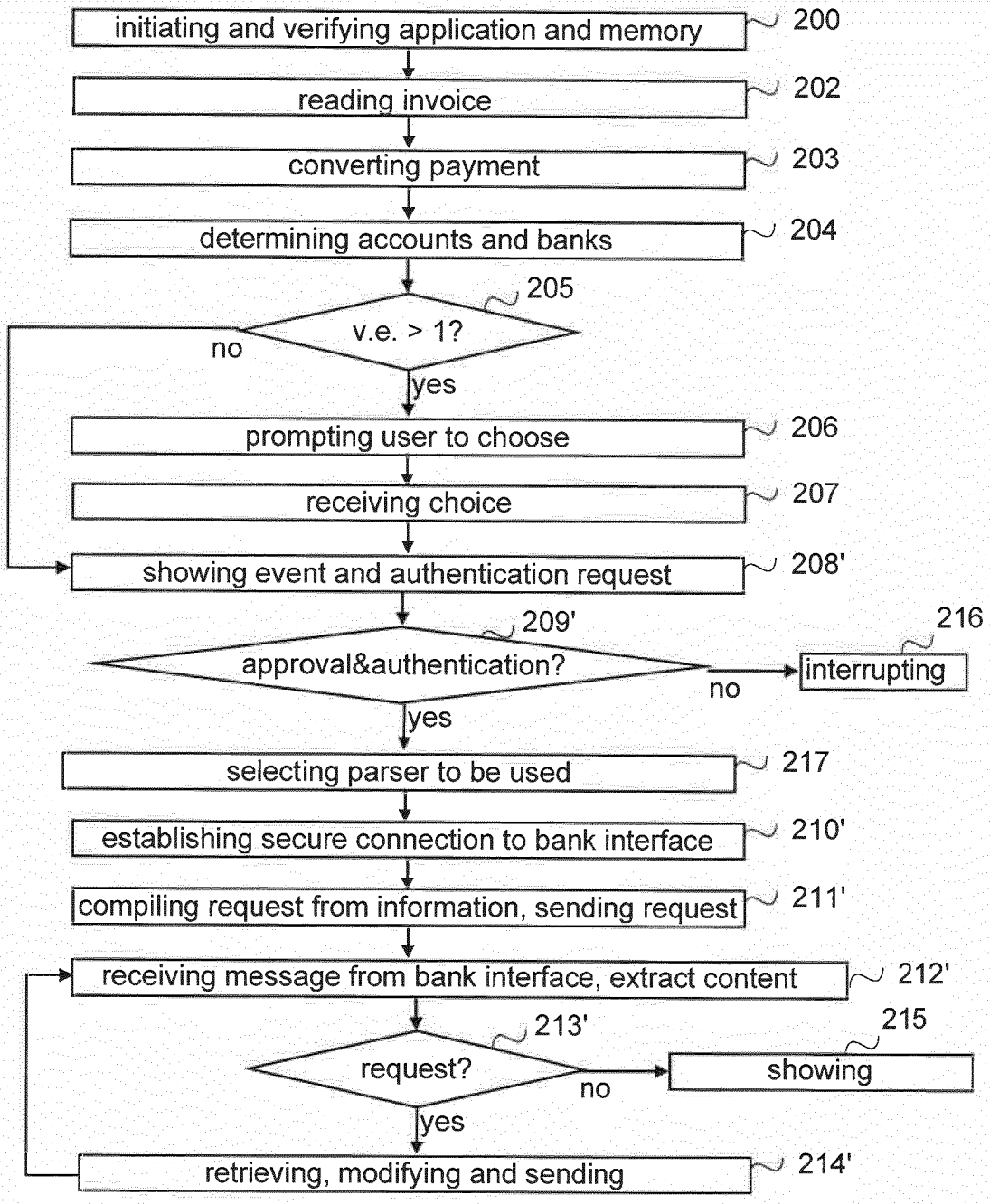


FIG.2B

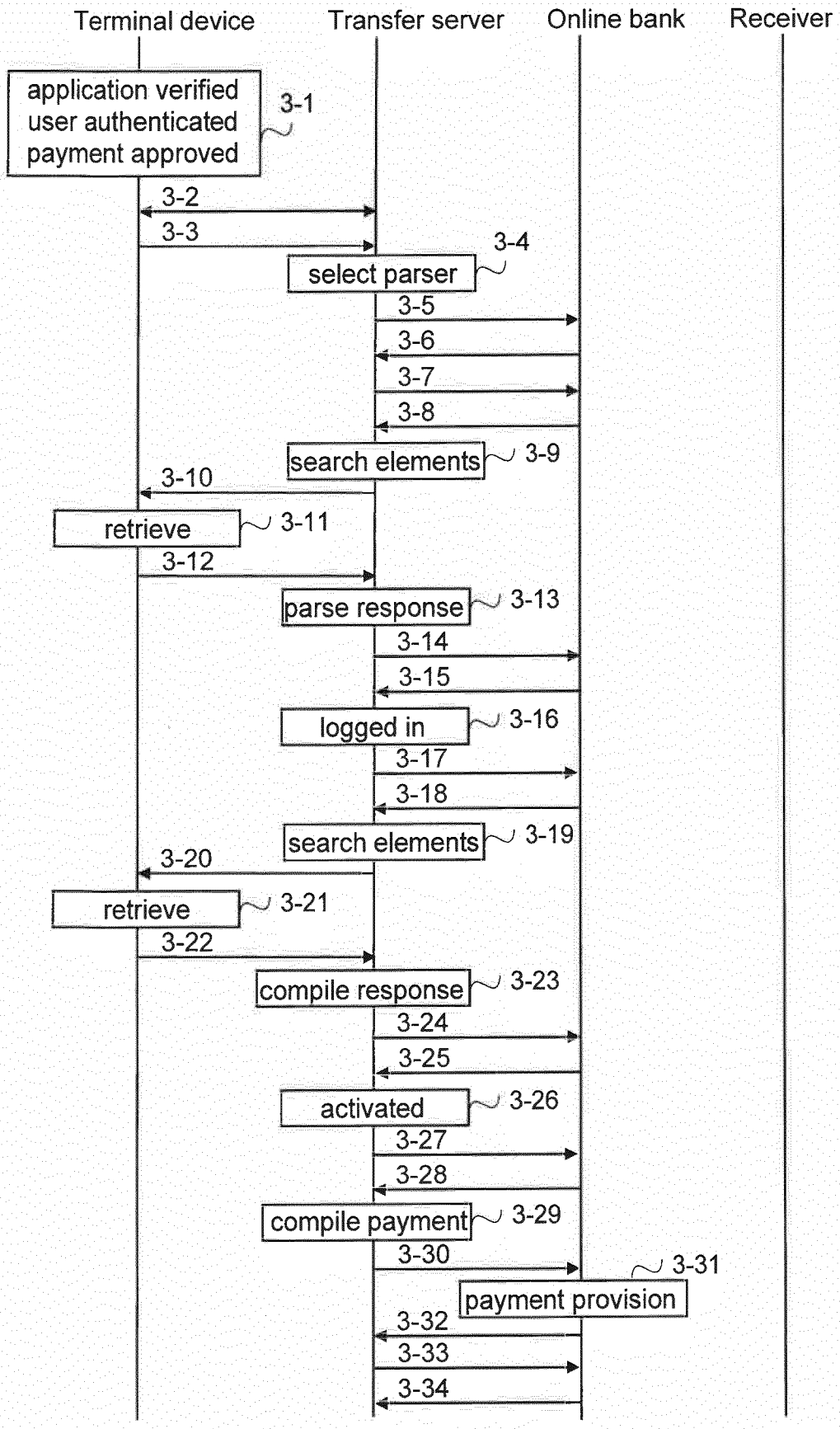


FIG.3A

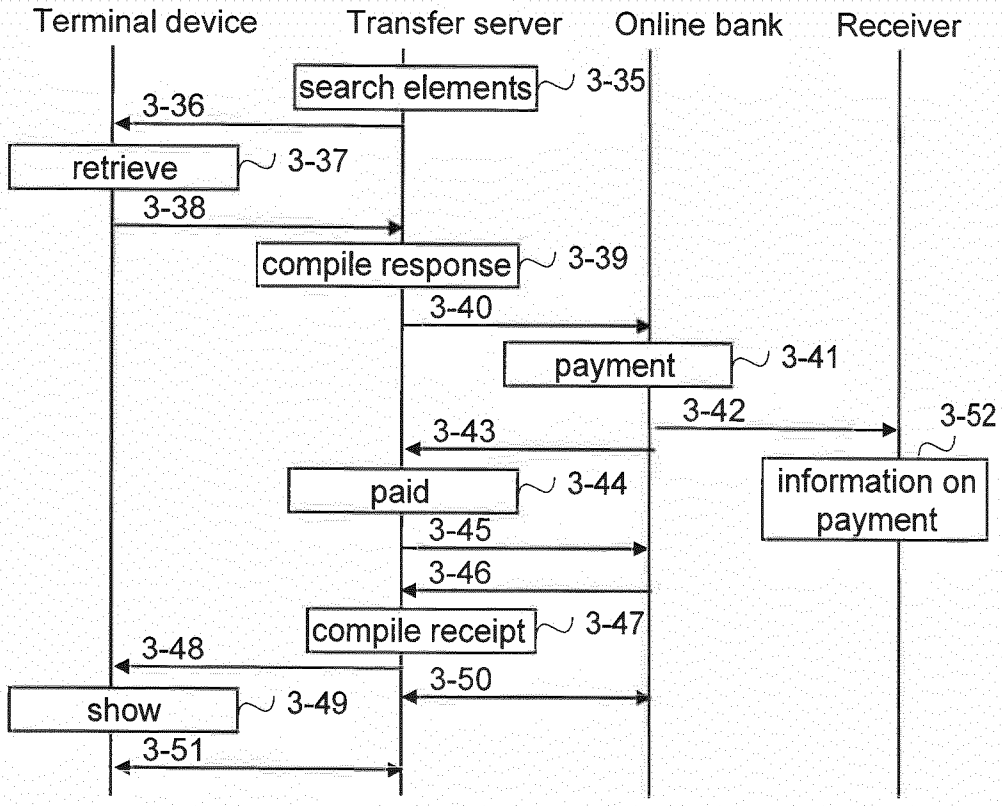


FIG.3B

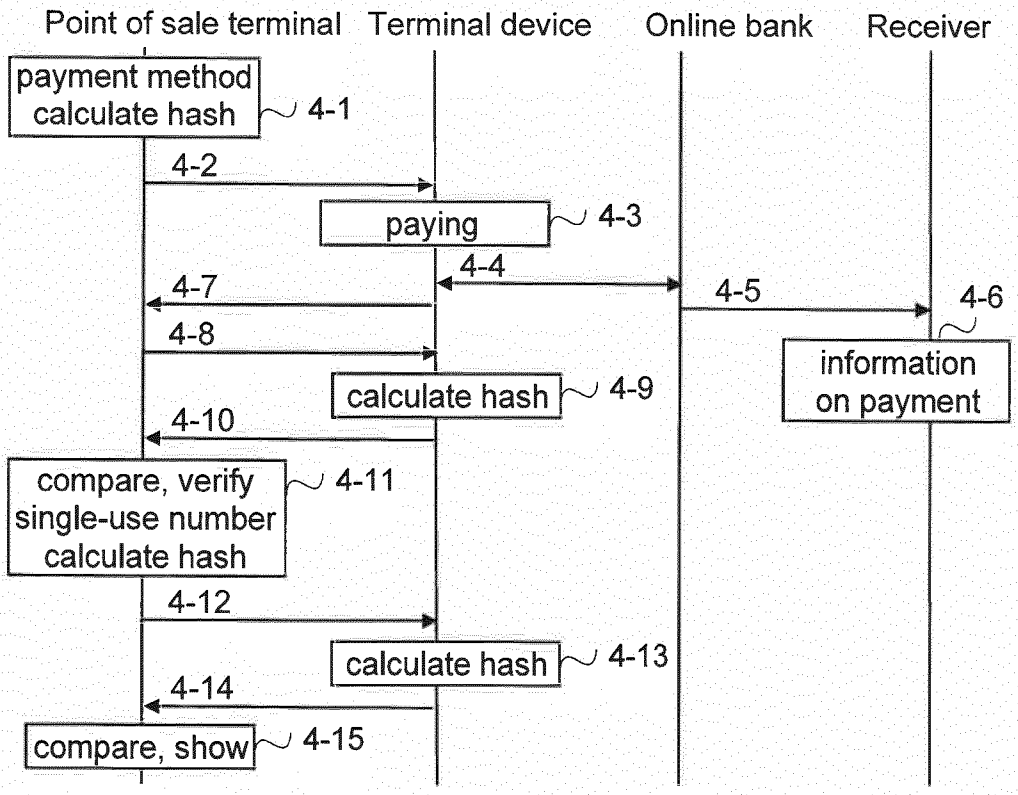


FIG.4

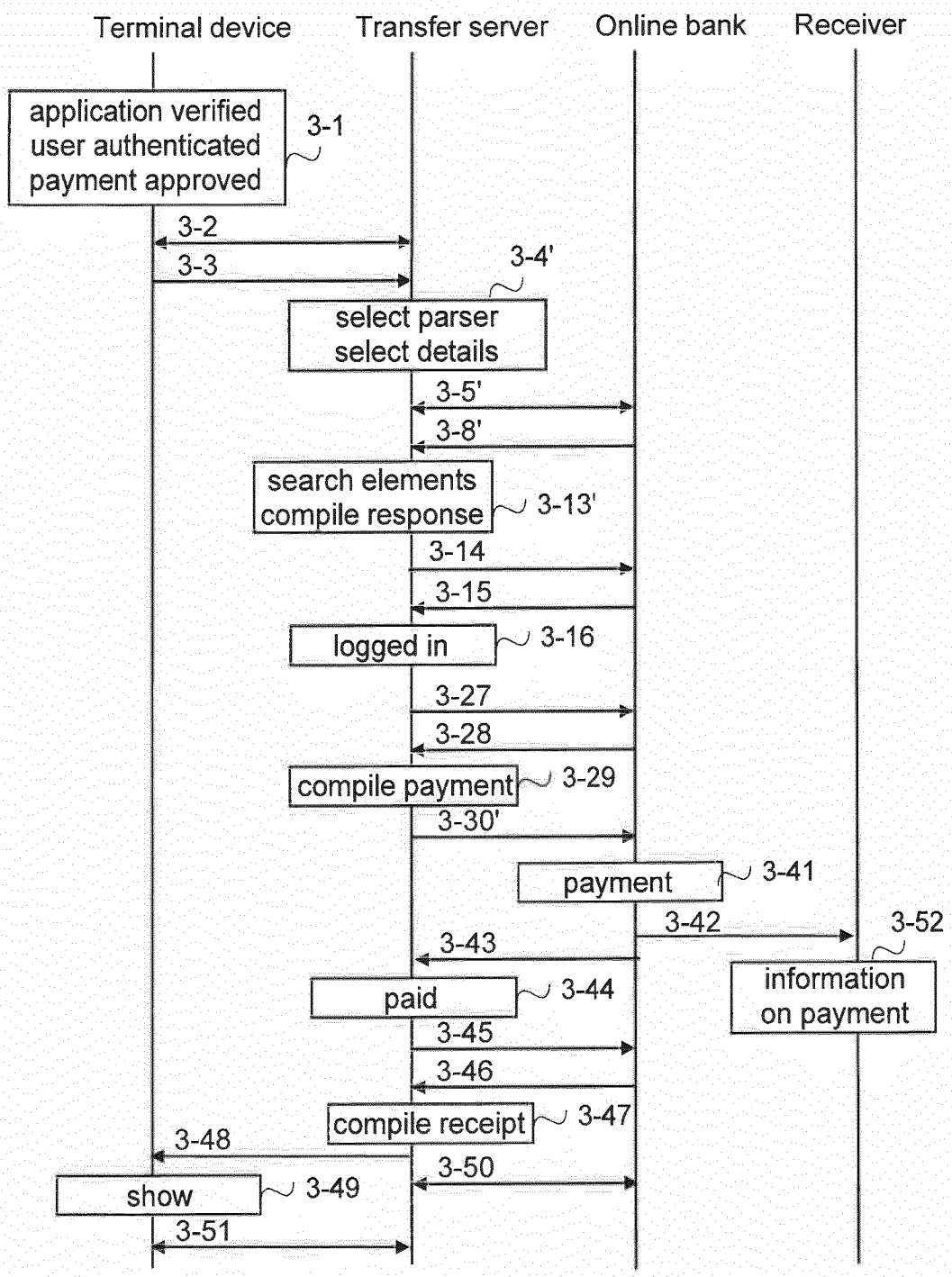


FIG.3C

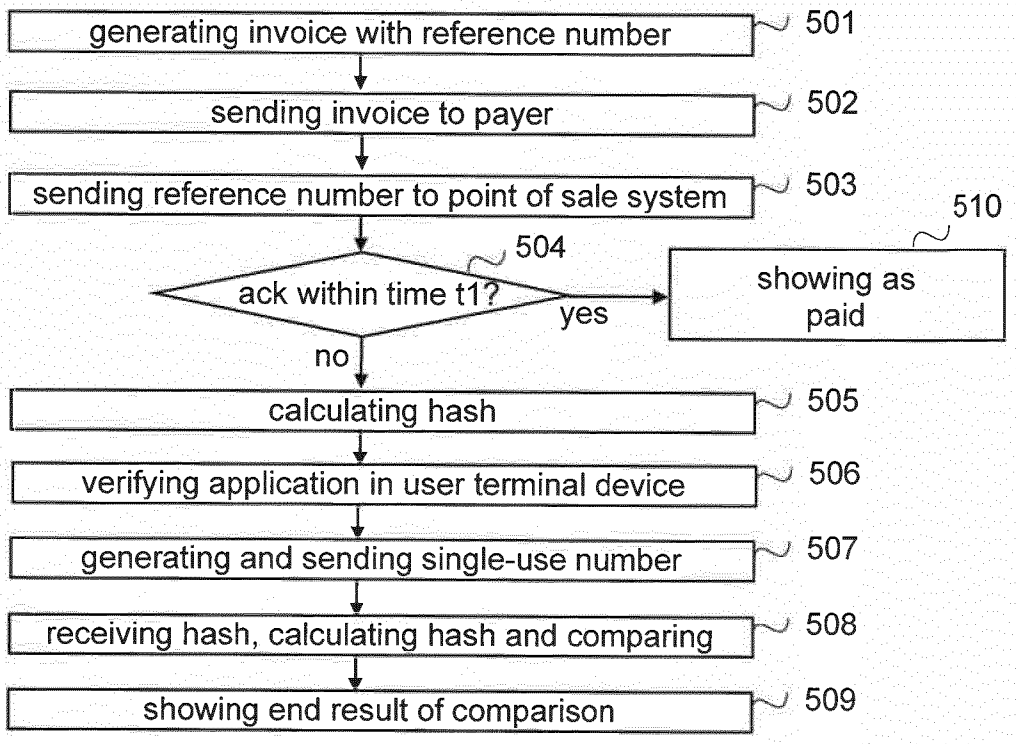


FIG.5A

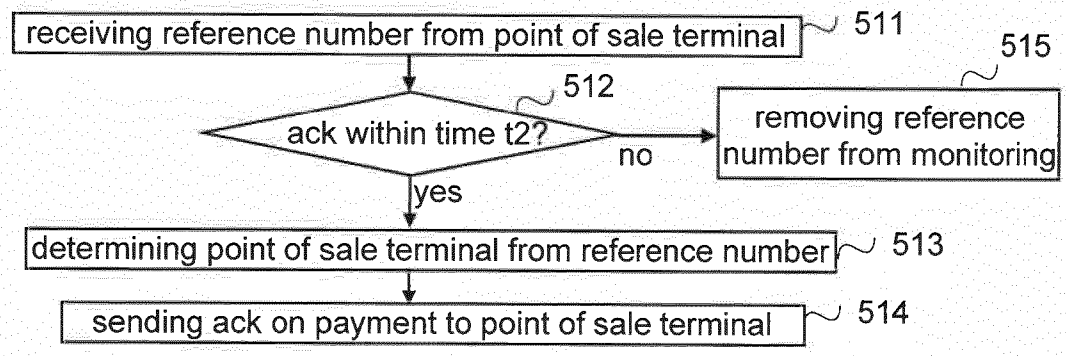


FIG.5B

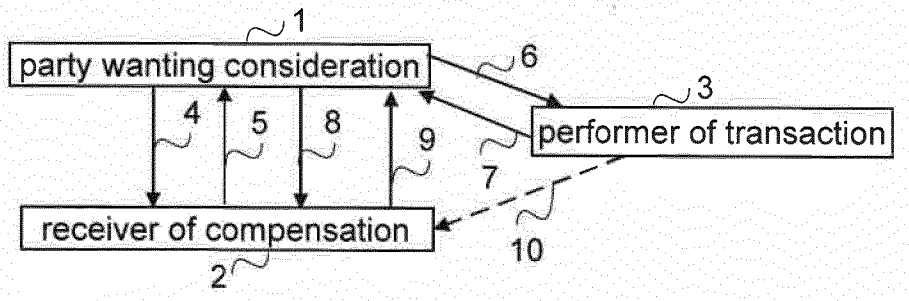


FIG.6

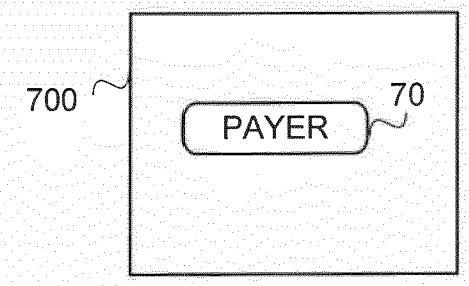


FIG.7A

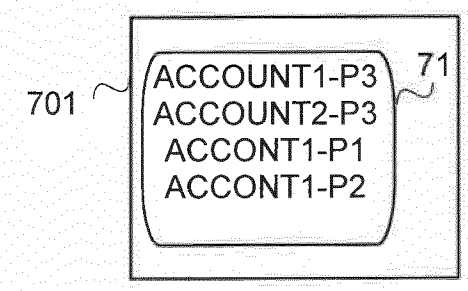


FIG.7B

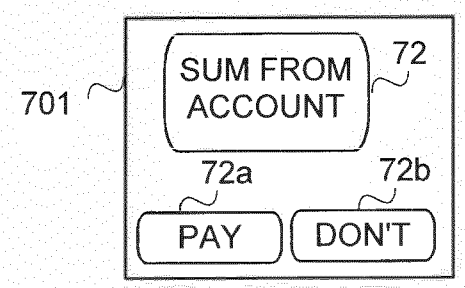


FIG.7C

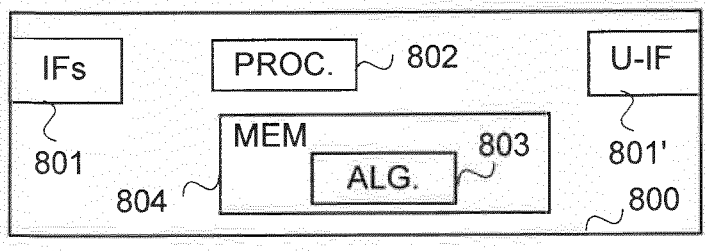


FIG.8

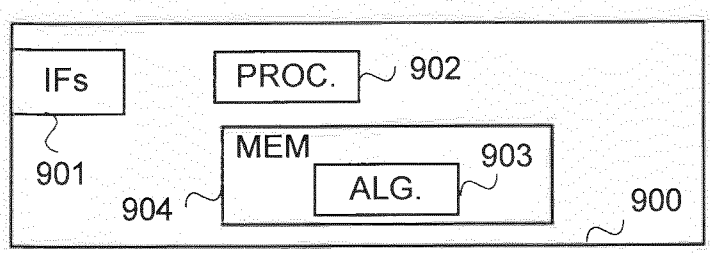


FIG.9

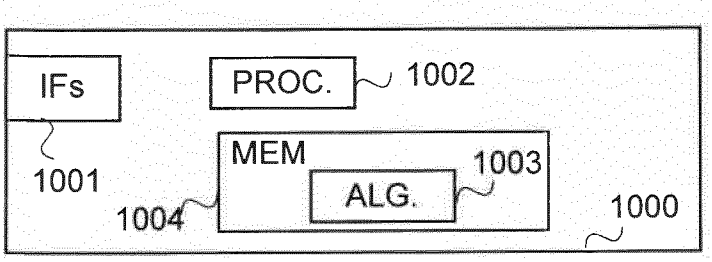


FIG.10

20175372 PRH 25-04-2017