



- (51) International Patent Classification:
H04L 9/08 (2006.01) *H04L 29/06* (2006.01)
- (21) International Application Number:
PCT/SE2015/051069
- (22) International Filing Date:
8 October 2015 (08.10.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
1451213-1 9 October 2014 (09.10.2014) SE
- (71) Applicant: **KELISEC AB** [SE/SE]; Box 14007, S-167 14 Bromma (SE).
- (72) Inventor: **REVELL, Elise**; Rånäs Bruksväg 68 B, S-762 96 Rånäs (SE).
- (74) Agent: **STRÖM & GULLIKSSON AB**; Box 5275, S-102 46 Stockholm (SE).

- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR ESTABLISHING A SECURE COMMUNICATION CHANNEL

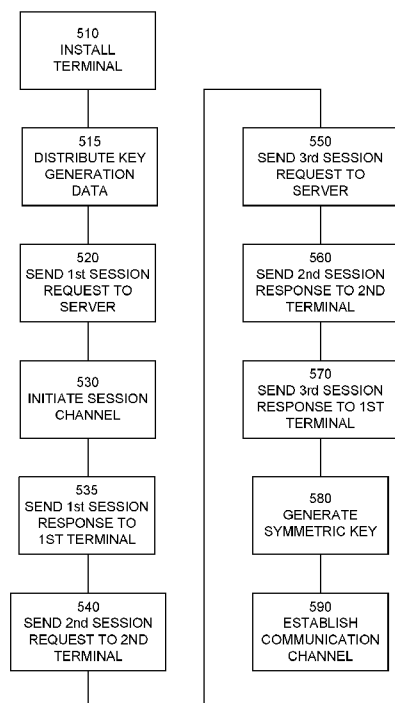


Fig. 5

(57) Abstract: A system comprising a first terminal and a second terminal, wherein the first terminal is configured to initiate a communication session with the second terminal by sending a first session request to a server for initiating a communication channel with the second terminal, receiving a first session response from the server, said first response including an identifier for a session channel and data relevant to the second terminal, sending a second session request to the second terminal including an identifier for the first terminal, receiving a third session response from the second terminal, and establishing a connection over the session channel, wherein the second terminal is configured for: receiving the second session request from the first terminal, sending a third session request to the server for initiating a communication channel with said first terminal, receiving a second session response from the server, said second response including the identifier for the session channel and data relevant to the first terminal, sending said third session response to the first terminal, and establish a connection over the session channel, whereby a communication channel is established between the first and the second terminal over the session channel.



Published:

— *with international search report (Art. 21(3))*

METHOD AND SYSTEM FOR ESTABLISHING A SECURE COMMUNICATION CHANNEL

by

5

Elise Revell

CO-EXISTING RELATED PATENT APPLICATIONS

10 This application is filed concurrently with the applications titled as per
below and filed by the same applicant. Details on specific issues in the various patent
applications are to be considered as incorporated into each one of the other related
patent applications.

 The related patent applications are titled:

15 IMPROVED INSTALLATION OF A TERMINAL IN A SECURE
SYSTEM;
 GENERATING A SYMMETRIC ENCRYPTION KEY;
 MUTUAL AUTHENTICATION; and
 IMPROVED SECURITY THROUGH AUTHENTICATION TOKENS.

20

TECHNICAL FIELD

 This application relates to a system, a method, an apparatus and a computer-
readable storage medium for improved security and safety, and in particular to a system,
a method, an apparatus and a computer-readable storage medium for establishing a
25 secure communication channel.

BACKGROUND

 In today's society digital communication is ever increasing and most people
feel comfortable performing tasks such as managing their bank accounts, declaring their
30 income, or exchanging other secret information using the Internet. In order to perform
such tasks while maintaining a satisfactory level of security many different security

solutions, based on encryption, have been suggested. Most such solutions involve some type of public key and private key and a user has to share his public key before a secure communication may be established with another party. In cryptography, a key may be seen as a piece of information that determines a functional output of a cryptographic
5 algorithm.

It is often said that in designing security systems, it is wise to assume that the details of a cryptographic algorithm may already be available to an attacker. This principle is known as Kerckhoffs' principle and thus it is only the secrecy of the key that provides security. This principle is based on the fact that it is difficult to keep the details
10 of a widely-used algorithm secret. A key is often easier to protect, since it is often a small piece of information compared to the encryption algorithm. However, it may also be difficult to keep the key a secret. If the attacker obtains the key in some way he or she may recover the original message from the encrypted data.

Encryption algorithms which use the same key for both encryption and
15 decryption are known as symmetric encryption key algorithms. There are also asymmetric encryption key algorithms which use a pair of keys, one to encrypt and one to decrypt. These asymmetric encryption key algorithms allow one key to be made public while retaining the private key in only one location. The asymmetric encryption keys are designed so that finding out the private key is extremely difficult, even if the
20 corresponding public key is known. A user can publish his or her public key, while keeping the private key secret, allowing anyone to send them an encrypted message.

In order for a key to be "safe" in conjunction with symmetric encryption algorithms a length of 80 bits is generally considered the minimum and 128-bit keys are commonly used and considered to be very strong. The keys used in public key
25 cryptography have some mathematical structure. For example, public keys used in the RSA system are the product of two prime numbers. Thus public key systems require longer key lengths than symmetric systems for an equivalent level of security. 3072 bits is the suggested key length for systems based on factoring and integer discrete logarithms which aim to have security equivalent to a 128 bit symmetric cipher.

30 As mentioned above it is possible to generate keys with a high degree of security, if they are long enough both for keys based on symmetric and asymmetric

algorithms. However, there may be a problem in key distribution. If, for example two parties want to communicate with each other using symmetric cryptography they first have to decide what key to use and then safely distribute it from one party to the other. Furthermore, the key has to be kept secret by both parties. The risk that an intruder may
5 find out the key increases with the time the key is in use. Therefore, a key is normally valid only during a limited time, for example six or twelve months. After this time a new key has to be distributed.

Also, the distribution of keys for asymmetric cryptography encryption encounters problems with key distribution when two parties want to communicate with
10 each other. In order to send information in both directions they generally need to exchange public keys with each other. Also in this case the keys usually have a limited time period during which they are valid. For a party that communicates with many different parties the management and distribution of valid public keys may be annoying. A typical example is when the validity of a key has expired when you need to send
15 some secret information urgently to another party or that you have not yet exchanged public keys.

One solution is provided in WO 2011/002412 which discloses a method for generating an encryption/decryption key, and especially for generating a one-time encryption/decryption key used for symmetric encryption, i.e. where the same key is
20 used both for encryption and decryption. In order to start key generation, a first terminal (A) sends a request to a central server (2) for setting up communication with a second terminal (B). The central server (2) sends a key generating file to both terminals (i.e. terminals A and B). Each terminal generates a different intermediate data set, i.e. a first and second data set. The first data set generated by the first terminal (A) is sent to the
25 second terminal (B), which generates, based on this data set, a third data set which is sent back to the first terminal. The generation of a first cryptographic key in terminal (A) is based on a bit by bit comparison between the third and the first intermediate data set and the generation of a second cryptographic key is based on a bit by bit comparison between the first and the second intermediate data set. The first and second
30 cryptographic keys are the same.

Although this manner has many advantages, it may offer challenges when implemented in systems having firewalls or other similar security measures. Given the above mentioned challenges the inventors have realized that there is a need for a method with which it is possible for two parties to communicate in a secure way with each other
5 without necessarily having to exchange keys with each other beforehand and which may be used in systems employing firewalls or other similar security measures.

Accordingly, there is thus a need for improved handling of the encryption keys.

10 SUMMARY

It is an object of the teachings of this application to overcome the problems listed above by providing a system comprising a first terminal and a second terminal, wherein the first terminal is configured to initiate a communication session with the second terminal by sending a first session request to a server for initiating a
15 communication channel with the second terminal, receiving a first session response from the server, said first response including an identifier for a session channel and data relevant to the second terminal, sending a second session request to the second terminal including an identifier for the first terminal, receiving a third session response from the second terminal, and establishing a connection over the session channel, wherein the
20 second terminal is configured for: receiving the second session request from the first terminal, sending a third session request to the server for initiating a communication channel with said first terminal, receiving a second session response from the server, said second response including the identifier for the session channel and data relevant to the first terminal, sending said third session response to the first terminal, and establish
25 a connection over the session channel, whereby a communication channel is established between the first and the second terminal over the session channel.

It is also an object of the teachings of this application to overcome the problems listed above by providing a method for establishing a secure communication channel in a system comprising a first terminal and a second terminal, wherein the
30 method comprises the first terminal sending a first session request to a server for initiating a communication channel with the second terminal; the first terminal receiving

a first session response from the server, said first response including an identifier for a session channel and data relevant to the second terminal; the first terminal sending a second session request to the second terminal including an identifier for the first terminal; the second terminal receiving the second session request from the first terminal; the second terminal sending a third session request to the server for initiating a communication channel with said first terminal; the second terminal receiving a second session response from the server, said second response including the identifier for the session channel and data relevant to the first terminal; the second terminal sending said third session response to the first terminal; the first terminal receiving a third session response from the second terminal; and the first terminal establishing a connection over the session channel and the second terminal establishing a connection over the session channel, whereby a communication channel is established between the first and the second terminal over the session channel.

It is also an object of the teachings of this application to overcome the problems listed above by providing a computer readable storage medium encoded with instructions that, when executed on a processor, performs the method according to above.

The teachings herein find use in systems for ensuring a secure connection, such as banking applications, but also in systems where one device needs to be authenticated to other devices, such as an external memory being connected to a computer.

Other features and advantages of the disclosed embodiments will appear from the following detailed disclosure, from the attached dependent claims as well as from the drawings.

Generally, all terms used in the claims are to be interpreted according to their ordinary meaning in the technical field, unless explicitly defined otherwise herein. All references to "a/an/the [element, device, component, means, step, etc]" are to be interpreted openly as referring to at least one instance of the element, device, component, means, step, etc., unless explicitly stated otherwise. The steps of any method disclosed herein do not have to be performed in the exact order disclosed, unless explicitly stated.

BRIEF DESCRIPTION OF DRAWINGS

The invention will be described in further detail with reference to the accompanying drawings in which:

5 Figure 1 is a schematic view of a terminal according to the teachings herein;

 Figure 2 is a schematic view of the components of a terminal according to the teachings herein;

10 Figure 3 is a schematic view of a general view of a computer network according to the teachings herein;

 Figure 4 shows a schematic view of a computer-readable medium according to one embodiment of the teachings of this application;

15 Figure 5 shows a flowchart for a method according to the teachings herein for establishing a communication channel between a first terminal and a second terminal in a system;

 Figure 6 shows a flowchart for a method according to the teachings herein for installing a terminal in a system;

 Figure 7 is a flowchart of a general method for mutual authentication of a terminal and a server according to the teachings herein;

20 Figure 8 is a schematic overview of the authentication transmissions being done during a mutual authentication according to the teachings herein;

 Figure 9 shows an example of a meta data string and a corresponding authentication token according to an embodiment of the teachings herein;

25 Figure 10A shows a general flowchart for generating an authentication token according to the teachings herein;

 Figure 10B shows a general flowchart for authenticating an authentication token according to the teachings herein;

 Figure 11 shows a flowchart for a method of generating a symmetric encryption key according to the teachings herein;

Figure 12 is a schematic overview of the transmissions being done during the generation of a symmetric encryption key according to the teachings herein;

Figure 13 is a flowchart for a general method according to herein for
5 generating a processing file from the key data for two terminals according to the teachings herein;

Figure 14 is a flowchart of a manner of generating the symmetric encryption key according to the teachings herein; and

Figure 15 shows a schematic overview of the system communication
10 according to the teachings herein.

DETAILED DESCRIPTION

The disclosed embodiments will now be described more fully hereinafter with reference to the accompanying drawings, in which certain
15 embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided by way of example so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art. Like numbers refer
20 to like elements throughout.

Figure 1 shows a communications apparatus 100 according to an embodiment herein. In one embodiment the communication apparatus 100 is configured for network communication, either wireless or wired. In one
25 embodiment the communication apparatus 100 is configured for network communication, both wireless and wired. Examples of such a communication apparatus 100 are: a personal computer, desktop or laptop, an internet tablet, a mobile telephone, a smart phone, a personal digital assistant, a server, electronic key, machine-to-machine device (M2M) and a work station to name a few.

The communication apparatus 100 will hereafter be exemplified and
30 described as being a personal computer 100. The personal computer or terminal 100 comprises a display 110 and a housing 120. The housing comprises a

controller or CPU (not shown) and one or more computer-readable storage
mediums (not shown), such as storage units and internal memory. Examples of
storage units are disk drives or hard drives. The terminal 100 further comprises at
least one data port. Data ports can be wired and/or wireless. Examples of data
5 ports are USB (Universal Serial Bus) ports, Ethernet ports or WiFi (according to
IEEE standard 802.11) ports. Data ports are configured to enable a terminal 100
to connect with other terminals or a server.

The terminal 100 further comprises at least one input unit such as a
keyboard 130. Other examples of input units are computer mouse, touch pads,
10 touch screens or joysticks to name a few.

Figure 2 shows a schematic view of a general structure of a computing
device. The computing device may be a communication apparatus as in figure 1
or a server. In one embodiment the computing device is a computing apparatus as
will be discussed in further detail below. In the description below the computing
15 device will be disclosed as being a terminal according to figure 1. The terminal
100 comprises a controller 210 which is responsible for the overall operation of
the terminal 100 and is preferably implemented by any commercially available
CPU ("Central Processing Unit"), DSP ("Digital Signal Processor") or any other
electronic programmable logic device. The controller 210 may be implemented
20 using instructions that enable hardware functionality, for example, by using
executable computer program instructions in a general-purpose or special-
purpose processor that may be stored on a computer readable storage medium
(disk, memory etc) 240 to be executed by such a processor. The controller 210 is
configured to read instructions from the memory 240 and execute these
25 instructions to control the operation of the terminal 100. The memory 240 may be
implemented using any commonly known technology for computer-readable
memories such as ROM, RAM, SRAM, DRAM, CMOS, FLASH, DDR,
EEPROM memory, flash memory, hard drive, optical storage or any combination
thereof.

30 The terminal 100 may also be connected to an external device such as
an external memory 245, possibly in the form of a USB stick (Universal Serial

Bus) or an external harddrive. Upon connecting such an external device to a terminal, then being an apparatus comprising the terminal and an external terminal or device 245. Any driver update or download that needs to be effected to enable the external device to cooperate with the terminal 100 may be
5 performed by a driver application executed in the controller or by another controller not shown.

The terminal 100 further comprises one or more applications 250. The applications are sets of instructions that when executed by the controller 210 control the operation of the terminal 100. The memory 240 is used for various
10 purposes by the controller 210, one of them being for storing application data and program instructions 250 for various software modules in the terminal 100. The software modules include a real-time operating system, drivers for a man-machine interface 220, an application handler as well as various applications 250. The applications 250 can include a messaging application such as electronic mail,
15 a browsing application, a banking application, as well as various other applications 250.

The terminal 100 may further comprise a user interface 220, which in the terminal 100 of figure of 1, is comprised of the display 110, the keypad 130. The display 110 may be a touch display upon which virtual keys (not shown) may
20 be displayed and operated. Such virtual keys may replace any or all of the physical keys 130 and no difference will be made herein if not explicitly stated.

The user interface (UI) 220 also includes one or more hardware controllers, which together with the UI drivers cooperate with the display 110, keypad 130, as well as various other I/O devices such as sound system, LED
25 indicator, etc. As is commonly known, the user may operate the terminal 100 through the man-machine interface thus formed.

The terminal 100 may further comprise a radio frequency interface 230, which is adapted to allow the terminal to communicate with other devices through a radio frequency band through the use of different radio frequency
30 technologies. Examples of such technologies are WIFI, Bluetooth®, W-CDMA, GSM, UTRAN, LTE, NMT to name a few.

The terminal 100 may further comprise a wired interface 235, which is adapted to allow the terminal to communicate with other devices through the use of different network technologies. Examples of such technologies are USB, Ethernet, Local Area Network, TCP/IP (Transport Control Protocol/Internet
5 Protocol) to name a few.

The controller 210 is configured to operably execute applications 250 such as the web browsing or email application through the RF interface 230 and/or the wired interface 235 using software stored in the memory 240 which software includes various modules, protocol stacks, drivers, etc. to provide
10 communication services (such as transport, network and connectivity) for the RF interface 230 and the wired interface 235, and optionally a Bluetooth interface and/or an IrDA interface for local connectivity. The RF interface 230 comprises an internal or external antenna as well as appropriate radio circuitry for establishing and maintaining a wireless link to a base station. As is well known to
15 a person skilled in the art, the radio circuitry comprises a series of analogue and digital electronic components, together forming a radio receiver and transmitter. These components include, i.e., band pass filters, amplifiers, mixers, local oscillators, low pass filters, AD/DA converters, etc.

Figure 3 shows a schematic overview of a computer network 300
20 according to an embodiment herein. Two terminals 100a, 100b, such as the terminals 100 of figures 1 or 2, are connected through a network or other connection 330 in one example via the internet. The two terminals may be connected through a wired connection or a wireless connection or any combination of known connection methods for example through dedicated
25 networks or connections. In figure 3 there are two terminals 100a and 100b, of which one is a desktop computer 100a and the other is a smartphone 100b. It should be noted that any terminal may be connected to the internet and the number and type of terminals 100 in figure 3 should not be construed as limiting. The computer network 300 further comprises at least one server 340. In figure 3
30 only one server 340 is shown, but it should be noted that any number of servers 340 may be implemented in a computer network 300. Generally a server is a

physical computer (a hardware system) dedicated to running one or more services (as a host), to serve the needs of users of the other computers or terminals 100a, 100b on the network 300. Depending on the computing service that it offers it could be a database server, file server, mail server, print server, web server, or
5 other.

In one embodiment the server 340 is a web server 340. Generally, a web server 340 can refer to either hardware (a computer) or software (a computer application) that helps to deliver content that can be accessed through a mutual communication network, such as the Internet 330.

10 The Internet 330 is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/IP – Transmission Control protocol/Internet Protocol) to serve billions of users worldwide. It is a network of networks that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a
15 broad array of electronic, wireless and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents of the World Wide Web (WWW).

As would be apparent to a skilled reader, the internet is full of possibilities and variations of how to connect two terminals and the embodiments
20 disclosed herein are for purely exemplary purposes and should not be construed to be limiting.

Even if the disclosure herein is focused on communications networks between terminals and a server, the teachings herein may also be used within an apparatus 300 for establishing a secure connection between two devices, such as
25 an external memory being connected to a personal computer. With reference to figure 3 a first device 100a is then to be connected to a second device 100b. The communication is to be established over a data bus or other internal communication network, as is known to a skilled person. A server application running in one (or both) of the devices 100a and 100b or alternatively in another
30 device not shown such as a controller of the apparatus 30 corresponds to the server 340 of figure 3. With reference to figure 2, such an apparatus 300 may be

conceived as a terminal 100 to which an external device 245 is to be connected to. The establishment of a secure communication channel as per the teachings herein may then be effected by the controller, serving as a server possibly through a server application being executed by the controller 210, or by another
5 drive controller. Possibly the server is an external security server which is contacted through a wired or wireless communication channel to handle the establishment of the communication exchanging addresses for the local bus or other intra-device communication hardware.

It should be noted that an external device in this context may in one
10 embodiment be understood to be a device to be connected to a terminal. It may thereafter be a more or less permanent component of, and possibly also internal to, the terminal 100.

Figure 4 shows a schematic view of a computer-readable medium as described in the above. The computer-readable medium 40 is in this embodiment
15 a data disc 40. In one embodiment the data disc 40 is a magnetic data storage disc. The data disc 40 is configured to carry instructions 41 that when loaded into a controller, such as a processor, executes a method or procedure according to the embodiments disclosed above. The data disc 40 is arranged to be connected to or within and read by a reading device 42, for loading the instructions into the
20 controller. One such example of a reading device 42 in combination with one (or several) data disc(s) 40 is a hard drive. It should be noted that the computer-readable medium can also be other mediums such as compact discs, digital video discs, flash memories or other memory technologies commonly used. In such an embodiment the data disc 40 is one type of a tangible computer-readable medium
25 40.

The instructions 41 may also be downloaded to a computer data reading device 44, such as a computer or other device capable of reading computer coded data on a computer-readable medium, by comprising the instructions 41 in a computer-readable signal 43 which is transmitted via a
30 wireless (or wired) interface (for example via the Internet) to the computer data reading device 44 for loading the instructions 41 into a controller. In such an

embodiment the computer-readable signal 43 is one type of a non-tangible computer-readable medium 40.

The instructions may be stored in a memory (not shown explicitly in figure 4, but referenced 240 in figure 2) of the computer 44.

5 References to computer program, instructions, code etc. should be understood to encompass software for a programmable processor or firmware such as, for example, the programmable content of a hardware device whether instructions for a processor, or configuration settings for a fixed-function device, gate array or programmable logic device etc.

10 The inventors have realized that a manner similar to that of WO 2011/002412 may be used also in systems using security measures such as firewalls, if the manner is modified in a clever and insightful manner. Figure 5 shows an overview of such a manner for establishing a mutually secure connection between two terminals.

15 Figure 5 shows a flowchart for a method according to the teachings herein for establishing a communication channel between a first terminal 100a and a second terminal 100b in a system 300 as in figure 3.

 Before a connection is to be established a terminal, such as either the first terminal 100a and/or the second terminal 100b of figure 3, is installed or
20 initiated 510 in the communication network 300 for the relevant application. Such an application may be a banking application, a secure communication application, or any other application requiring that both parties in a communication are identified. In one embodiment, such an application is when one device is to be connected to another device of an apparatus.

25 The installation is performed by a terminal 100 sending a request to be installed to a server 340. The server 340 responds by sending back information or data regarding key generation to the terminal 100 which receives the key generation data. Such key generation data is thereby distributed 515.

 More details regarding the installation of a terminal 100 into the
30 system 300 will be given with reference to figure 6.

When a first terminal 100a has been installed, it may want to set up a secure communication with another terminal 100b.

In one example embodiment, where the overall application is a banking application, the first terminal 100a represents the client and the second
5 terminal 100b represents the bank. The server 340 may be hosted by a mutual bank cooperation, the bank itself or a secure connections provider.

In order to setup such a secure connection, the first terminal 100a initiates the setup by sending a first session request 520 to the server 340. The first session request includes an identifier for the second terminal 100b with
10 which the first terminal wants to setup a communication with. The identifier for the channel data may in one embodiment be an address (such as a uniform resource locator address (URL) or Internet Protocol (IP) address) to an internet protocol communication channel.

The server 340 initiates or establishes a session channel 530 and
15 responds with a first session response 535 indicating an identifier for the session channel and key data for the second terminal 100b, which is received by the first terminal 100a. The first terminal 100a then sends a second session request to the second terminal 100b. The second session request includes the identity of the first terminal 100a and a key encryption seed for the first terminal 100a.

20 As the second terminal 100b receives the second session request from the first terminal 100a, the second terminal 100b sends a third session request 550 to the server 340. The third session request includes an identifier for the first terminal 100a and/or an identifier for the current session. The server 340 determines whether the first terminal 100a is allowed to connect with the second
25 terminal 100b and if so sends a second session response 560 to the second terminal 100b, wherein the second session response includes the identifier for the session channel and also data regarding key creation for the first terminal 100a.

As the second terminal 100b receives the second session response, the second terminal 100b may be sure that the first terminal is allowed to establish a
30 connection with the second terminal 100b. The second terminal 100b thereafter

sends a third session response 570 to the first terminal 100a, which third session response includes key data and a key encryption seed for terminal 100b.

The second terminal 100b and the first terminal 100a thus both have received key data from the server 340 as well as each other's key encryption
5 seed. Based on this, the first terminal and the second terminal generate 580 a symmetric encryption key for encrypting data to be sent on the initiated session channel and a channel is thereby established 590.

More details on how the symmetric encryption key is generated is given with reference to figures 12, 13 and 14.

10 One alternative manner of generating such a symmetric encryption key is to be found in the teachings of WO 2011/002412.

In this manner, both the first terminal 100a and the second terminal 100b can be sure of the other terminal's identity and credibility as both terminals have been authenticated by the server 340. The terminals are also enabled to
15 generate a symmetric encryption key without having to exchange complete keys with one another and without having to exchange a complete key, such as the symmetric key, with the server 340.

Furthermore, the server is authenticated by both terminals 100a and 100b. In prior art examples, it is only the server that authenticates the terminals,
20 not the other way around as per the teachings herein.

More details on such mutual authentication are given with reference to figures 6 and 7 under Mutual Authentication.

Note that the whole setup of the secure communication channel is performed by active prompts from a terminal to which a response is provided
25 thus enabling the communication systems setup to be used in systems employing security measures such as a firewall.

Installation of a terminal

In order to ensure that a terminal that is to be installed in the system 300 is to be trusted and in order to ensure that any installation data transmitted
30 between the terminal and the server can not be misused by a third party, the authors of this application have designed a manner for a secure installation where

the key data is not sent in full thereby making it useless to any third party intercepting the communication. Figure 6 shows a flowchart for a general method according to the teachings herein for installing a terminal 100 in a system 300 as in figure 3.

5 As a terminal 100 is to be installed in the system, such as when installing a terminal referenced 510 in figure 5, the server sets up an account for the terminal 100 by generating a random identification number, NodeID, 511 and a random encryption key, referenced Y, 512.

10 The random identification number and the random encryption key is then distributed in a safe manner to the terminal 100 to be installed. They may be distributed by regular mail, by email, by text message (such as short message service (SMS) messaging, or other secure channel.

15 The random identification number and the random encryption key are entered 513 into the terminal 100, possibly by a user, or by (another) secure application, where they are checked to see if they are correct, possibly through the use of a checksum being part of the random identification number and the random encryption key.

 The terminal 100 then identifies 514 itself to the server 300 through the random identifier, NodeID, and is thus identified by the server 340.

20 The server then generates two random encryption key seeds, a key data seed and a metadata seed, commonly referred to as a data seed.

 The key data seed may be of a length of for example 128 or 256 bits. The metadata seed may be of a length of for example 128 or 256 bits.

25 Using these seeds, the server generates key data and metadata. The key data is the data that is used to generate the symmetric encryption key and the metadata is used together with or without the key data to generate an authentication token, which is to be described further below.

30 The key data may be of a length of 128, 256, 512 or 1024 bits. The metadata may be of a length of 8192 bits.

To safeguard against eavesdropping the server 340 only distributes 515 the key generating seeds to the terminal 100, that is, only the key data seed and the meta data seed are distributed to the terminal 100. This ensures that the actual key data and the meta data are kept secret in the server 340.

5 The key data and the meta data are encrypted using the random encryption key before being distributed to the terminal 100. And the terminal 100 decrypts the key data and the meta data using the random encryption key.

 The terminal 100 then generates the key data and the metadata by using the same function as the server 340 used. The function may be distributed 10 along with the random identifier and the random encryption key, or it may be distributed separately, wherein the random identifier and/or the random encryption key are parameters of the function, and/or the function may be implemented in the terminal and/or server. In one embodiment the function is a cryptographic hash function, such as a Feistel network.

15 Several functions for generating numbers are known and will not be discussed in further detail herein. The only feature of importance to the function is that it is the same function that is used by the terminal 100 as by the server 340.

 The seeds are as such much smaller than the corresponding data to be 20 generated and the server 340 only needs to transmit the smaller seeds to the terminal which saves bandwidth and also increases security as the seeds are worthless without the function.

 The key data and the metadata are therefore generated 516 by both the server 340 and the terminal 100.

25 An authentication token is generated 517 by the terminal 100 based on the key data and/or the meta data and the authentication token is sent to the server 340 which authenticates 518 the authentication token, the server having access to the key data and the metadata on which the authentication token is based.

30 More details on how an authentication token is generated and also authenticated will be given below with reference to figures 10A and 10B.

If the authentication token is authenticated, an identity for the terminal 100, such as an identification, is stored in the server 340. In one embodiment, the identification for the terminal 100 is the NodeID. In one embodiment, the identification for the terminal 100 is the IMSI (International Mobile Subscriber Identity), IMEI (International mobile Equipment Identifier) and/or serial number(s) of the terminal 100. In one embodiment, the identification for the terminal 100 is an application identifier provided with an application utilizing the secure communication channels as taught by the teachings herein.

For the embodiment where a terminal is an external device, the NodeID for the external device may be supplied along with the external device or be sent to a user or to the external device upon registration of the external device 245.

Mutual Authentication

As a terminal 100a has been installed and attempts to set up a communication channel with a second terminal 100b, such as a bank application server, where the communication channel is administered by a security server 340, possibly being part of or implemented in the same bank server, the terminal 100a is to be authenticated by the authenticated by the security server 340, hereafter and as above referred to as the server 340. After the terminal 100a is authenticated by the server 340 it is provided with key data necessary for establishing the secure communication with the second terminal 100b.

However, in today's society many scams are run over the internet, which is the reason for the secure communication channel. However, the inventors have realized that there is an underlying problem inherent to all such secure communication systems in that using today's advanced technology neither the first terminal 100a (nor the second terminal 100b) can be sure that the server 340 is actually to be trusted. The address of the server 340 may actually have been hijacked by a malicious scam operator who may provide key data leading to an encryption which the scam operator may then be able decrypt.

Realizing that this problem exists and through insightful reasoning and inventive thinking, the inventors of this application have designed a manner where also the server 340 is authenticated.

For the embodiment of installing an external device 245 in an apparatus 300, such as a terminal 100, such an authentication serves to protect that sensitive data is not read by another device being part of or later connected to the apparatus 300 or by a replaced operating software. Therefore mutual authentication serves to protect from gaining access to a protected device simply by replacing some components (memory carrying software code) or other devices.

Figure 7 is a flowchart of a general method for mutual authentication of a terminal 100 and a server 340 according to the teachings herein and figure 8 is a schematic overview of the authentication transmissions being done during such a mutual authentication. A detailed description of such authentication will be given with simultaneous reference to figures 7 and 8 in the below.

The first terminal 100a initiates the authentication by generating an authentication token AT 1-S (reference indicating that it is an authentication token from terminal 1 to server S) and sending it 710 to the server 340 possibly along with a session request. Alternatively, the session request is sent after the authentication has been completed successfully. It should be noted that any of the terminals 100a, 100b may initiate the authentication. The server 340 may also initiate the authentication. The server may initiate the authentication by prompting the first terminal 100a (or the second) to provide an authentication token.

The server 340 receives the authentication token AT1-S and authenticates it 715.

The authentication token may be an identifier (such as the NodeID) for the terminal, possibly encrypted with the previously received random encryption number, known only to the first terminal 100a and the server 340. The authentication token may also be another string or number known only to the

server 340 and the terminal 100a possibly determined through a known protocol or algorithm, and may also be encrypted with the random encryption number.

Being able to secure that the authentication is safe it is beneficial to use a large string as such are more difficult to guess. Such a string is the meta data which is known only to the server 340 and the terminal 100. A problem in this is that a large file has to be sent every time the authentication is to be made, and as will be disclosed in the below, the meta data would also be shared with the second terminal thereby leaking the meta data to the second terminal who should not have knowledge of the meta data for the first terminal 100a. However, using the meta data would be beneficial as it is a large string that is difficult to maliciously derive and is also known only to the server 340 and the terminal 100.

The inventors have thus designed a clever and simple manner of devising an authentication token from the meta data, wherein a plurality of chunks or portions is chosen at random from the meta data.

Authentication token

Figure 9 shows an example of a meta data string MTS and a corresponding authentication token AT. In the example of figure 9, the meta data string is comprised of only 40 characters, however it should be noted that the actual length of a meta data string MTS is preferably much longer. In one embodiment the length is 8192 bits. Furthermore, the number of portions used in this example is four and the length of each portion is 2, but in a real implementation the length of a portion should be 64-128 bits, and in one embodiment the portion length is 96 bits. It should be noted that the length of a portion may vary also within an authentication token, as may the number of portions. In an embodiment where the number of portions varies, a header to the authentication token may be used to indicate the number of portions. In an embodiment where the length of one or more portions vary, a header to the authentication token may be used to indicate the length of the portions or, each portion may be initiated by the length of the portion.

The four portions are referenced by indexes. The indexes may be chosen randomly or through a known function, such as a random number

generator, e.g. using a seed. Examples of such a function have been given above in the disclosure of how a terminal is installed in a system.

In such an embodiment where a seed is used for a function to generate the indexes, only the seed may be necessary to transmit in the authentication
5 token.

An index should have a length that is enough to adequately address the larger data set, i.e. the meta data string (MTS). For a meta data of 8192 bits, an index has to have a length of 13 bits. An index may also be given as a number of steps to the next portion, wherein the index length can be reduced by one bit
10 assuming that the next portion is at least within the next half of the meta data.

The indexes or the seed may be given in a header of the authentication token. The indexes may also be given as a header to each portion. Figure 9 shows three examples for such authentication tokens, where a seed is given in a header, four indexes are given in a header and where the four indexes are given as
15 headers to each portion. In all three examples, the seed is 24, and the indexes are 3, 7, 13 and 16 and the corresponding portions are “as”, “or”, “Ip” and “it” respectively.

It should be noted, as in the above, that the example indexes and portion lengths as well as the length of the metadata string MTS are made short
20 to illustrate the teachings herein more easily.

By only sending portions of the meta data it is assured that a terminal 100 and a server 340 may be authenticated without wasting bandwidth and without divulging sensitive information as only portions of the meta data are transmitted.

25 An authentication token AT may then be authenticated by simply comparing the portions to corresponding portions in the meta data. The corresponding portions are retrieved through the indexes which are either received as part of the authentication token or retrieved as a function of the seed which is part of the authentication token.

The indexes may also be given based on a timed function, such as the time of day or a counter. The counter has to be incremented at a speed during which an authentication token may be generated, sent and received.

Figure 10A shows a general flowchart for generating an authentication token according to the teachings herein and Figure 10B shows a general flowchart for authenticating an authentication token according to the teachings herein.

An authentication token AT is generated by generating at least one index 1010. The number of indexes may vary within an implementation and also between implementations as has been discussed above. In one embodiment the number of indexes is 4.

The indexes may be generated as randomly chosen numbers, or through a randomly chosen seed being used with a function.

Thereafter at least one portion is retrieved 1020 from a larger data set, such as the meta data (MTS in figure 9) and packaged along with the indexes (or the seed) in a data structure 1030 such as shown in figure 9.

As an authentication token is received 1040, the indexes are retrieved 1050 (either from the authentication token or by using a function based on a seed) and the corresponding portion(s) are retrieved 1060 from a locally stored data set, such as the meta data. The corresponding portion(s) are then compared 1080 to the received portion(s) and if they match the authentication token is authenticated.

The comparison may be made bitwise or by any other known comparison. A bitwise comparison has the benefit that it is fast to execute.

If an index is chosen to be too close to an end of the data set so that there are not enough characters or bits left to complete a portion, further or missing characters or bits may be taken from the start of the data set, i.e. the data set is looped.

In the above no difference has been made between a data set containing characters and a data set containing bits. As would be understood by a

skilled person such entities may easily be replaced within the scope of the teachings herein.

For a meta data set of 8192 bits and with four portions or chunks of each 96 bits an authentication token of $4 \times 96 + 4 \times 13$ (an index needs to be at least 13 bits to properly address 8192 bits) = 436 bits. So not only is sending the authentication token alone safer than sending the meta data as the meta data is kept secret, the bandwidth needed is also reduced by approximately 95 %.

In one embodiment, the index is chosen to be able to address more than the meta data and the corresponding portion can then easily be found by addressing the meta data at the index modulo the metadata size. In the example of the meta data having 8192 bits, the address to retrieve a portion is thus to be found at the index modulo 8192.

As is clear from the description above an authentication token may be generated and authenticated by both a terminal and a server. In fact, an authentication token may be generated by any computing device, the terminals of figures 1 and 2 and the server of figure 3 being examples of such a computing device. An external device may also be such a computing device.

Returning to figures 7 and 8 and the mutual authentication, as the server 340 has authenticated the authentication token AT 1-S received from the first terminal 100a, the server generates 720 one authentication token AT S-1 directed to the first terminal 100a and one authentication token AT S-2 directed to the second terminal 100b and sends 725 both authentication tokens AT S-1, AT S-2 to the first terminal 100a.

In one embodiment, the server 340 encrypts the authentication token AT S-2 directed for the second terminal 100b. The authentication token may be encrypted using a portion of the meta data for the second terminal 100b and in such an embodiment, the authentication token may be configured to comprise a further field indicating an encryption index which indicates where in the meta data the encryption key for encrypting the authentication token is to be found. The encryption index is therefore transmitted unencrypted. This prevents any

malicious attacker posing as the second terminal 100b from decrypting the authentication token AT S-2.

As the first terminal 100a receives 730 the two authentication tokens AT S-1 and AT S-2, the first terminal 100a authenticates 735 the first authentication token AT S-1 received from the server 340.

As the first terminal 100a does not have access to the second terminal's 100b meta data it cannot authenticate the second authentication token AT S-2 received from the server 340. However, by the server 340 sending an authentication token aimed at the second terminal 100b to the first terminal 100a, an additional authentication of the server 340 is achieved through the second terminal 100b.

If the authentication of the first authentication token AT S-1 received from the server 340 is successful, the first terminal 100b proceeds with the additional authentication which is effected by the second terminal 100b. To this end, the first terminal sends 740 the second authentication token AT S-2 received from the server 340 to the second terminal 100b. The second terminal 100b receives and authenticates 745 the authentication token AT S-2. If the authentication is successful, the server 340 has been authenticated through both the first terminal 100a and the second terminal 100b. A mutual authentication of a high degree of security has thus been achieved.

However, by realizing that if a malicious scam operator has been able to hijack a communication channel between the first terminal 100a and the server 340, the authentication token for the second terminal 2 could potentially also be compromised. In order to increase the security of the authentication even further, the second terminal 100b is advantageously configured to also authenticate the server 340.

As the second terminal 100b has authenticated the authentication token AT S-2 received from the server 340 through the first terminal 100a and aimed for the second terminal 100b, the second terminal proceeds to authenticate the server 340 using its own communication channel to the server 340.

The second terminal therefore generates an authentication token AT 2-S for the server 340 and sends 750 it to the server 340. The server 340 receives the authentication token AT 2-S and authenticates it 755. If the authentication is successful, then the second terminal 100b is authenticated by the server 340 and the server 340 generates 760 one authentication token AT S-1 directed to the first terminal 100a and one authentication token AT S-2 directed to the second terminal 100b and sends 765 both authentication tokens AT S-1, AT S-2 to the second terminal 100b.

As the second terminal 100b receives 770 the two authentication tokens AT S-1 and AT S-2, the second terminal 100b authenticates the second authentication token AT S-2 received from the server 340, and if successful sends 775 the second authentication token AT S-1 to the first terminal 100a which authenticates 780 the authentication token AT S-1 aimed for it received from the server 340 through the second terminal 100b.

Not only is the server 340 thus authenticated by the first terminal 100a, but it is also authenticated on behalf of the first terminal 100a by the second terminal 100b. The two terminals thus assist each other in authenticating the server 340. Furthermore, the second terminal 100b is authenticated implicitly by the first terminal 100 through the authentication tokens received from the server 340 via the second terminal 100b. Also, the server 340 is authenticated using two different communication channels thereby making it very hard for a scam operator to maliciously identify itself as the security server 340.

Generation of a symmetric encryption key

As the two terminals 100 and the server 340 have been authenticated, a symmetric encryption key is generated. With reference to figure 5, a terminal 100 receives 515 key generation data when it is installed and based on this generates 516 key data. The key data is to be used for generating a symmetric encryption key as will be disclosed below. The symmetric encryption key is to be used encrypting for communication between a first terminal 100a and a second terminal 100b. The first terminal 100a may be, as in the above, a client to a secure service, and the second terminal 100b may be the provider of the service.

Alternatively, the first terminal 100a may be an external device 245 to be connected to an apparatus 300, wherein the second terminal 100b represents the apparatus 300.

The key data generated is known only to the first terminal 100a and
5 the server 340 and the second terminal 100b does not know of this key data. This enables for an increased security as the communication connection to be established is established without exchanging respective terminal's key data, thus making it difficult for a malicious scam operator to intercept any key data.

For this purpose, the inventors have devised a simple and clever
10 manner of producing a symmetric encryption key from the respective terminal's key data without actually sharing the key data.

Figure 11 shows a flowchart for a method of generating such a symmetric encryption key and figure 12 is a schematic overview of the transmissions being done during the generation of a symmetric encryption key
15 according to the teachings herein.

The first terminal 100a starts by sending a request for a symmetric encryption key to be generated 580. This request may be sent as part of a session setup request as per figure 5 or it may be sent on its own. The request may also be implicit in previous requests. The server 340 then proceeds with generating
20 581 a first processing file PF1, which is returned 582 to the first terminal 100a.

The first terminal 100a then extracts 583 a combined key data from the first processing file PF1. The first terminal 100a then generates a first random key seed which is encrypted and sent to the second terminal 584. The second terminal 100b also sends a request 580 for generating a symmetric encryption
25 key to the server 340 and the server 340 generates and sends 585 a second processing file PF2 to the second terminal 100b.

The second terminal 100b then extracts 586 the combined key data from the second processing file PF2 and generates a second random key seed which is sent 587 to the first terminal 100a. Both the first and the second
30 terminals 100 then generates the symmetric encryption key, each on their own 588, 589.

Figure 13 is a flowchart for a general method according to herein for generating a processing file PF from the key data for two terminals. The key data for a first terminal will be referenced KD1 and the key data for a second terminal will be referenced KD2.

- 5 As a processing file is to be generated 581, a combined key data (referenced CKD) is generated 5811 as a bitwise AND operation on the key datas for each terminal, that is:

$$\text{CKD} = \text{KD1 AND KD2}.$$

10

The inverse of the key data for the relevant terminal is then generated 5812, I.E. $\sim\text{KD}$ is generated and subjected 5813 to a function F where each 1 bit is changed to a 0 bit with a probability of for example 10 %, 25%, 50 % or a probability in any of the ranges of 10-80%, 20-70 % or 25- 50%. The resultant
15 changed bit string may be referenced as F($\sim\text{KD}$). The processing file is then generated 5814 as the logical OR between the combined key data and the resultant changed bit string. That is:

$$\begin{aligned} \text{PF1} &= \text{CKD OR F}(\sim\text{KD1}) = (\text{KD1 AND KD2}) \text{ OR F}(\sim\text{KD1}), \text{ and} \\ \text{PF2} &= \text{CKD OR F}(\sim\text{KD2}) = (\text{KD1 AND KD2}) \text{ OR F}(\sim\text{KD2}) \end{aligned}$$

20

The generated processing file PF is thus completely different from the key data and also from the combined key data and as such, the respective key data are kept secret from the other terminal and both the combined key data and
25 the respective key data are kept secret from the outside world.

As the processing file has been received, the combined key data is extracted (583/586 in figure 11) by simply performing the logical or bitwise AND of the processing file and the key data of the respective terminal, that is:

$$\begin{aligned} \text{CKD} &= \text{PF1 AND KD1 and} \\ \text{CKD} &= \text{PF2 AND KD2.} \end{aligned}$$

30

In order to illustrate the above, an example is presented below. It should be noted that the example is restricted to 8 bits for illustrative purposes, whereas in a real life implementation longer strings should be used, for example

5 128 bits, 256, 512, 1024 bits as more bits provides for a higher security.

Example

KD1	10001110
KD2	01111101
CKD = KD1 AND KD2	00001100
~KD1	01110001
F(~KD1)	01010000
PF1 = CKD OR F(~KD1)	01011100

Extraction

CKD = PF1 AND KD1	00001100
-------------------	----------

As the combined key data has been extracted, while maintaining it securely hidden in the processing file during transmission, the symmetric encryption key is to be generated. Figure 14 is a flowchart of an example manner

10 for generating the symmetric encryption key according to the teachings herein.

The symmetric encryption key is finally generated 588/589 by both terminals 100, by inputting the combined key data and a random key seed previously generated (584 for the first terminal and 587 for the second terminal) for each terminal as seeds in a function such as a cryptographic hash function or

15 an exclusive or function (XOR) 5801 for each terminal 100. In one embodiment, the function is to repeat the random key seed 8 times to become an extended random key seed of for example 512 bits and perform an exclusive OR function on the combined key data and the extended random key seed. The half Symmetric Key is created as 128 bits taken from the XORed value possibly using a modulo

20 function. This step is performed by both terminals 100 having received the other terminal's random key seed previously. In one embodiment, the result is a bit

string of for example 128 bits for each terminal and the two bits strings are concatenated 5802 into the thereby generated symmetric encryption key.

The symmetric encryption key is the result of the session setup and is to be used during the communication between the first terminal 100a and the
5 second terminal 100b for encoding and decoding the communication.

It should be noted that even though the flows for the mutual authentications and for generating the symmetric encryption keys have been disclosed as separate flows they may be parts of the same flow and data to be sent in transmissions in either flow may be combined into a common
10 transmission to reduce bandwidth and also parallelise the authentication flow.

Mutation of the key data

To protect against malicious attacker trying to hijack key data or other encryption data, a simple solution of mutating or changing the key data is proposed by the inventors of this application. It should be noted that the meta
15 data, the key data and the symmetric encryption key may all be mutated individually or collectively.

In one embodiment, the meta data is mutated once a day as it is mainly used for authentication purposes.

In one embodiment, the key data is mutated at every session setup or
20 at intervals or upon prompts from the first terminal during a session.

In one embodiment, the symmetric encryption key is mutated at intervals or upon prompts from the first terminal during a session.

One way to mutate the symmetric encryption key is to set up a new session. During the session setup the key data will be mutated and hence also the
25 symmetric encryption key. However, this consumes both computer resources and time to perform a complete session setup. A faster way is to only mutate the symmetric encryption key. This is only done at the terminals; the server is not involved.

It is up to the secure application to decide when the mutation is to be
30 done. In a chat application, for instance, it might be appropriate to mutate the symmetric encryption key for each message that is sent to the other terminal. In

an application that is time critical it might be better to perform the mutation at, for instance, every tenth package, or not at all.

It is up to the initiating terminal (sender) to control whether a new package should have a mutated Symmetric Encryption Key or not. This is done
5 by adding a version number, the Symmetric Encryption Key Version, to each communication package. If the initiating terminal wants to use this feature then it must increase the variable for each package that is sent.

The Symmetric Encryption Key Version is initially set to 0 at server setup. If the sender never changes this value no mutation will take place. If the
10 sender, on the other hand, for instance uses the values 3, 7, 13, 14 in the first four packages then the responding terminal needs to mutate 3, 4, 6, 1 times for the respective package.

The receiving first terminal always checks the version number of each communication package it receives. If it has been changed then the receiving
15 terminal has to mutate the Symmetric Encryption Key accordingly before decrypting the message. If the value is larger than the previous package then it just mutates the appropriate number of times. If it is smaller, then it must use the Symmetric Encryption Key with Symmetric Encryption Key Version = 0 and start all over again.

20 The mutation of the Symmetric Encryption Key may be done using the 512 bit cryptographic hash function using the current Symmetric Encryption Key as input.

Overview of the system

Figure 15 shows a schematic overview of the system communication
25 according to the teachings herein. The system 300 shown in figure 15 may be a system 300 or apparatus 300 as described above.

The first terminal 100a initiates the setup with the server 340 and the second terminal 100b, which is a responding terminal, as has been disclosed above with reference to figures 5, 6, 7, 8, 9, 10A, 10B, 11, 12, 13 and 14.

As the symmetric encryption key has been generated the first terminal 100a and the second terminal 100b can communicate securely over the communication channel established.

One benefit of the teachings herein is that a symmetric encryption key
5 may be generated by both communicating terminals without having to share it between the terminals or other devices.

The invention has mainly been described above with reference to a few embodiments. However, as is readily appreciated by a person skilled in the art, other
10 embodiments than the ones disclosed above are equally possible within the scope of the invention, as defined by the appended patent claims.

CLAIMS

1. A system (300) comprising a first terminal (100a) and a second terminal (100b), wherein

5 the first terminal (100a) is configured to initiate a communication session with the second terminal (100b) by:

 sending (520) a first session request to a server (340) for initiating a communication channel with the second terminal (100b);

 receiving (535) a first session response from the server (340), said first
10 response including an identifier for a session channel and data relevant to the second terminal (100b);

 sending (540) a second session request to the second terminal (100b) including an identifier for the first terminal (100a);

 receiving (570) a third session response from the second terminal (100b);
15 and

 establishing a connection over the session channel, and

wherein the second terminal (100b) is configured for:

 receiving (540) the second session request from the first terminal (100a);

20 sending (550) a third session request to the server (340) for initiating a communication channel with said first terminal (100a);

 receiving (560) a second session response from the server (340), said second response including the identifier for the session channel and data relevant to the first terminal (100a);

25 sending (570) said third session response to the first terminal (100a); and

 establishing (590) a connection over the session channel, whereby a communication channel is established between the first and the second terminal (100a, 100b) over the session channel.

30 2. The system according to claim 1, wherein

the first session response includes a combination of key data for the first terminal (100a) and key data for the second terminal (100b),

the second session request includes a key encryption seed for the first terminal (100a),

5 the third session response includes a key encryption seed for the second terminal (100b),

the second session response includes a combination of key data for the first terminal (100a) and key data for the second terminal (100b), and wherein

10 the first terminal (100a) is further configured to generate a same symmetric encryption key based on the key encryption seed for the first terminal (100a) and the combination of key data for the first terminal (100a) and key data for the second terminal (100b) in combination with the key encryption seed for the second terminal (100b) and the combination of key data for the first terminal (100a) and key data for the second terminal (100b), and

15 the second terminal (100b) is further configured to generate a symmetric encryption key based on the key encryption seed for the second terminal (100b) and the combination of key data for the first terminal (100a) and key data for the second terminal (100b) in combination with and the key encryption seed for the first terminal (100a) and the combination of key data for the first terminal (100a) and key data for the second terminal (100b),

20 whereby the same symmetric encryption key is generated for the first terminal (100a) and the second terminal (100b).

3. The system (300) according to claim 1 or 2, wherein the first terminal
25 (100a) is configured to be installed in the system (300) by the server (340) being configured to:

identify the first terminal (100a);

generate key generation data, comprising at least one data seed;

distribute the at least one data seed to the first terminal (100a);

30 generate key data and meta data based on said at least one data seed and a function, such as a cryptographic hash function;

store an identifier for the first terminal (100a) along with the key data and the meta data for the first terminal (100a), wherein the first terminal (100a) is arranged to

receive the at least one data seed from the server (340);

5 generate key data and meta data based on said at least one data seed and the same function;

store the key data and the meta data, wherein the key data and the meta data stored in the first terminal (100a) are the same as the key data and the meta data stored in the server (340).

10

4. The system (300) according to any preceding claim, wherein the server (340) is configured to

generate a first processing file (PF1) for the first terminal (100a) based on the key data for the first terminal (KD1) and a combination (CKD) of key data for the
15 first terminal (KD1) and key data for the second terminal (KD2);

generate a second processing file (PF2) for the second terminal (100b) based on the key data for the second terminal (KD2) and a combination (CKD) of key data for the first terminal (KD1) and key data for the second terminal (KD2); and

send the first processing file (PF1) to the first terminal (100a) and the
20 second processing file (PF2) to the second terminal (100b);

wherein the first terminal (100a) is configured to

receive the first processing file (PF1);

extract combined key data (CKD) from the first processing file (PF1);

generate a first random key seed and send it to the second terminal (100b);

25 receive a second random key seed from the second terminal (100b);

wherein the second terminal (100b) is configured to

receive the second processing file (PF2);

extract combined key data (CKD) from the second processing file (PF2);

generate the second random key seed and send it to the first terminal (100a);

30 receive the first random key seed from the first terminal (100a);

whereby the first terminal (100a) and the second terminal (100b) are each configured to:

input the combined key data (CKD) and the first random seed into a function;

5 input the combined key data (CKD) and the second random seed into the same function; and

concatenate the results of the functions into the symmetric encryption key, each terminal (100) thereby generating each copy of the symmetric encryption key.

10 5. The system (300) according to any preceding claim, wherein the first terminal (100a) is configured to authenticate the server (340) by:

generating a first authentication token (AT), being a data structure to be used for authenticating a first computing device (100a, 100b, 340), such as a terminal (100) to a second computing device (100a, 100b, 340), such as a server (340); and

15 sending the first authentication token (AT) to the server (340);

wherein the server (340) is configured for:

receiving the first authentication token (AT) and authenticate it;

generating a second authentication token for the first terminal (100a); and

20 send the second authentication token (AT) to the first terminal (100a),

whereby the first terminal (100a) is configured for

receiving the second authentication token (AT) and authenticate it, thereby performing a mutual authentication of the first terminal (100a) and the server (340), and

25 wherein the server (340) is further configured to generate a third authentication token (AT) for a second terminal (100b) and send it to the first terminal (100a), wherein the first terminal (100a) is further configured for receiving the third authentication token (AT) and sending it to the second terminal (100b), wherein the second terminal (100b) is configured for receiving and authenticating the third authentication token (AT)
30 thereby performing an authentication of the server (340) and the first terminal (100a).

6. The system (300) according to any preceding claim, wherein the first terminal (100a), the second terminal (100b) and/or the server (340) are arranged for generating an authentication token (AT) being a data structure to be used for authenticating a first computing device (100, 340), such as a terminal (100) to a second
5 computing device (100a, 340), such as a server (340), wherein the first terminal (100a), the second terminal (100b) and/or the server (340) are configured to:

generate at least one index;
retrieve a portion of the meta data (MTS), the portion being a portion of the data set starting at a position in the data set given by the index; and
10 include the portion in the authentication token (AT).

7. The system (300) according to any preceding claim, wherein the first terminal (100a), the second terminal (100b) and/or the server (340) are arranged to authenticate an authentication token being a data structure to be used for authenticating
15 one of the first terminal (100a), the second terminal (100b) and/or the server (340) to another one of the first terminal (100a), the second terminal (100b) and/or the server (340), wherein the first terminal (100a), the second terminal (100b) and/or the server (340) are configured to:

receive an authentication token (AT) from another of the first terminal
20 (100a), the second terminal (100b) and/or the server (340);
retrieve at least one index;
retrieve corresponding portion(s) from the meta data (MTS);
compare the corresponding portion(s) to portion(s) in the authentication token; and
25 authenticate the other of the first terminal (100a), the second terminal (100b) and/or the server (340) if the corresponding portion(s) match(es) the portion(s) of the authentication token (AT).

8. The system (100, 300) according to any preceding claim, wherein the
30 system (100, 300) is an apparatus (100) comprising at least one second terminal (100b)

and the first terminal (100a) is an external device (245) to be connected to the second terminal (100b).

9. The system according to claim 8, wherein said apparatus (300) is
5 configured for executing a server application (340).

10. The system according to claim 8, wherein said apparatus (300) is configured for communicating with the server (340).

10 11. A method for establishing a secure communication channel in a system (300) comprising a first terminal (100a) and a second terminal (100b), wherein the method comprises:

the first terminal (100a) sending a first session request to a server (340) for initiating a communication channel with the second terminal (100b);

15 the first terminal (100a) receiving a first session response from the server (340), said first response including an identifier for a session channel and data relevant to the second terminal (100b);

the first terminal (100a) sending a second session request to the second terminal (100b) including an identifier for the first terminal (100a);

20 the second terminal (100b) receiving the second session request from the first terminal (100a);

the second terminal (100b) sending a third session request to the server (340) for initiating a communication channel with said first terminal (100a);

25 the second terminal (100b) receiving a second session response from the server (340), said second response including the identifier for the session channel and data relevant to the first terminal (100a);

the second terminal (100b) sending said third session response to the first terminal (100a);

30 the first terminal (100a) receiving said third session response from the second terminal (100b); and

the first terminal (100a) establishing a connection over the session channel and the second terminal (100b) establishing a connection over the session channel, whereby a communication channel is established between the first and the second terminal (100a, 100b) over the session channel.

5

12. The method according to claim 11, wherein

the first session response includes a combination of key data for the first terminal (100a) and key data for the second terminal (100b),

the second session request includes a key encryption seed for the first terminal (100a),

the third session response includes a key encryption seed for the second terminal (100b),

the second session response includes key data for the first terminal (100a), and wherein the method further comprises

the first terminal (100a) generating the same symmetric encryption key based on the key encryption seed for the first terminal (100a) and the combination of key data for the first terminal (100a) and key data for the second terminal (100b) in combination with the the key encryption seed for the second terminal (100b) and the combination of key data for the first terminal (100a) and key data for the second terminal (100b), and

the second terminal (100b) generating a symmetric encryption key based on the key encryption seed for the second terminal (100b) and the combination of key data for the first terminal (100a) and key data for the second terminal (100b) in combination with the key encryption seed for the first terminal (100a) and the combination of key data for the first terminal (100a) and key data for the second terminal (100b),

whereby the same symmetric encryption key is generated for the first terminal (100a) and the second terminal (100b).

13. A computer readable storage medium (40) encoded with instructions (41) that, when executed on a processor, performs the method according to claim 11 or 12.

1/11

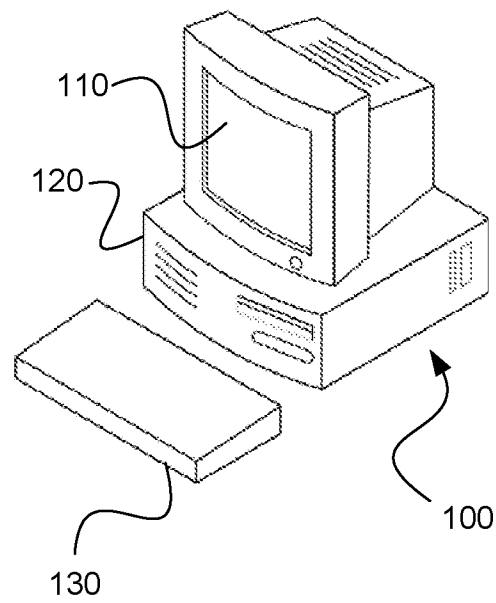


Fig. 1

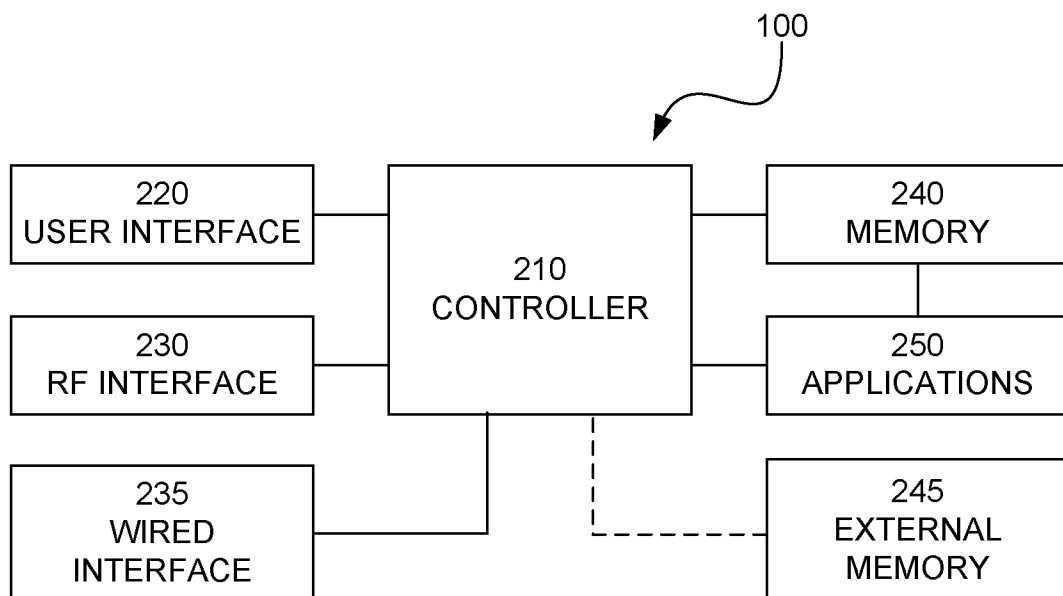


Fig. 2

2/11

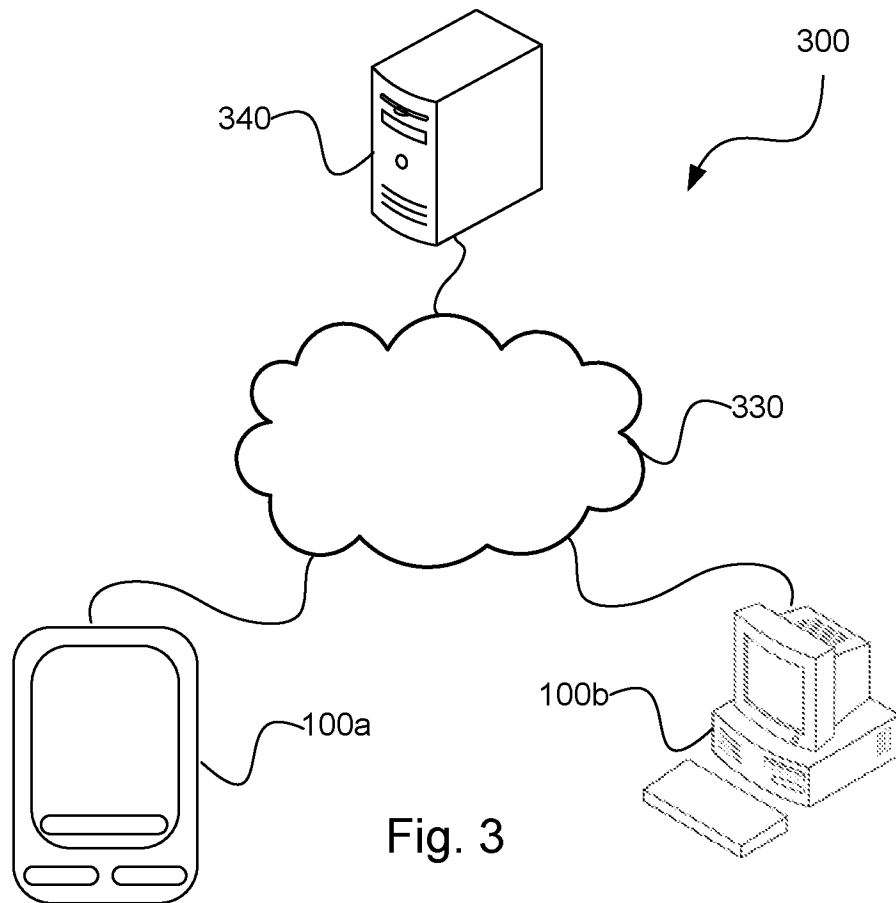


Fig. 3

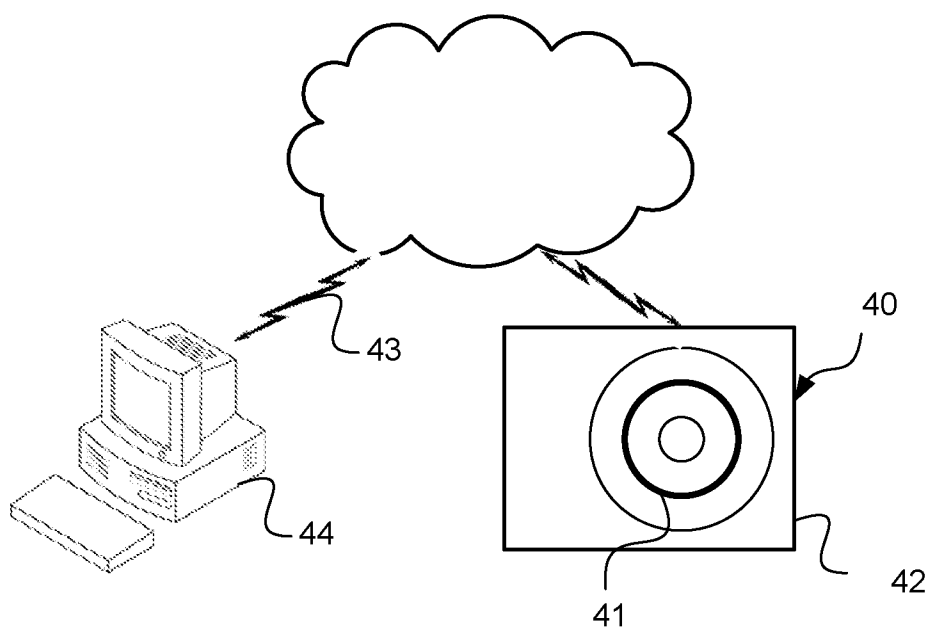


Fig. 4

3/11

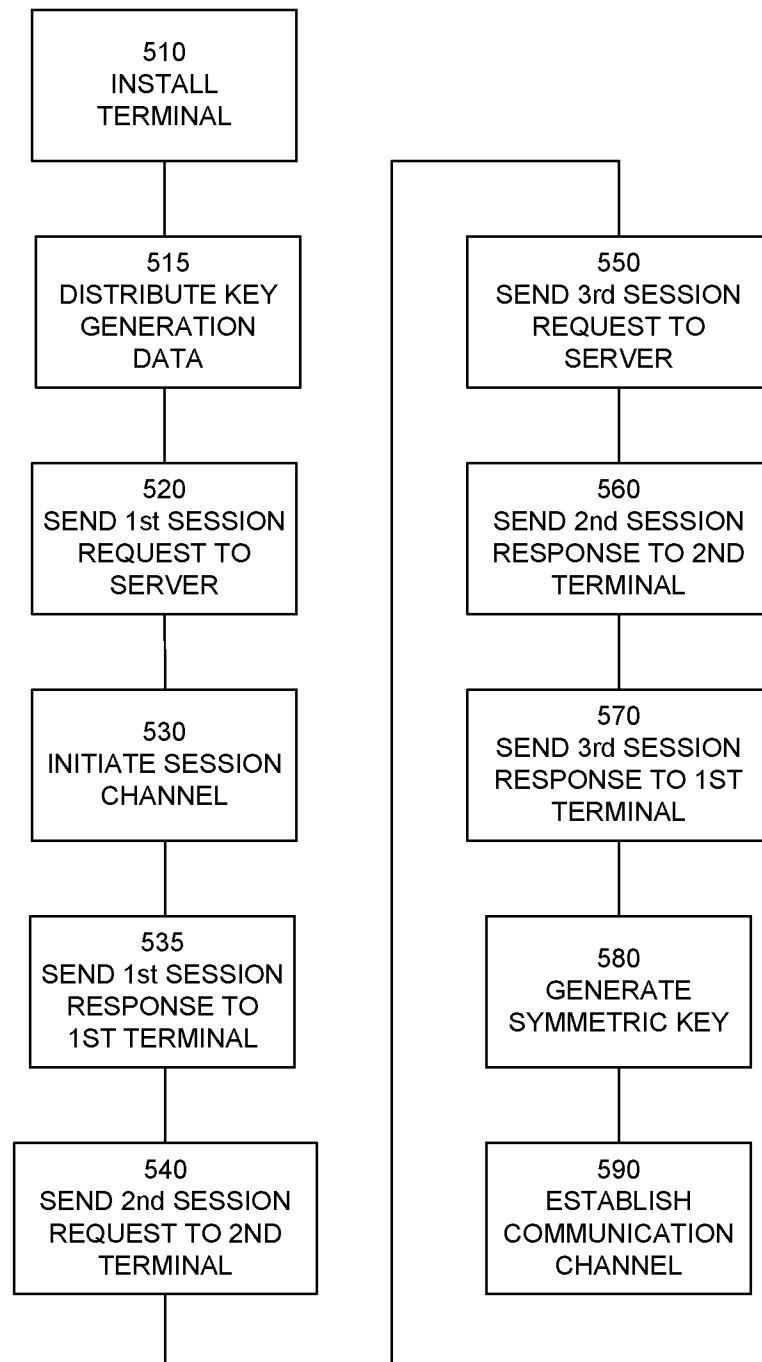


Fig. 5

4/11

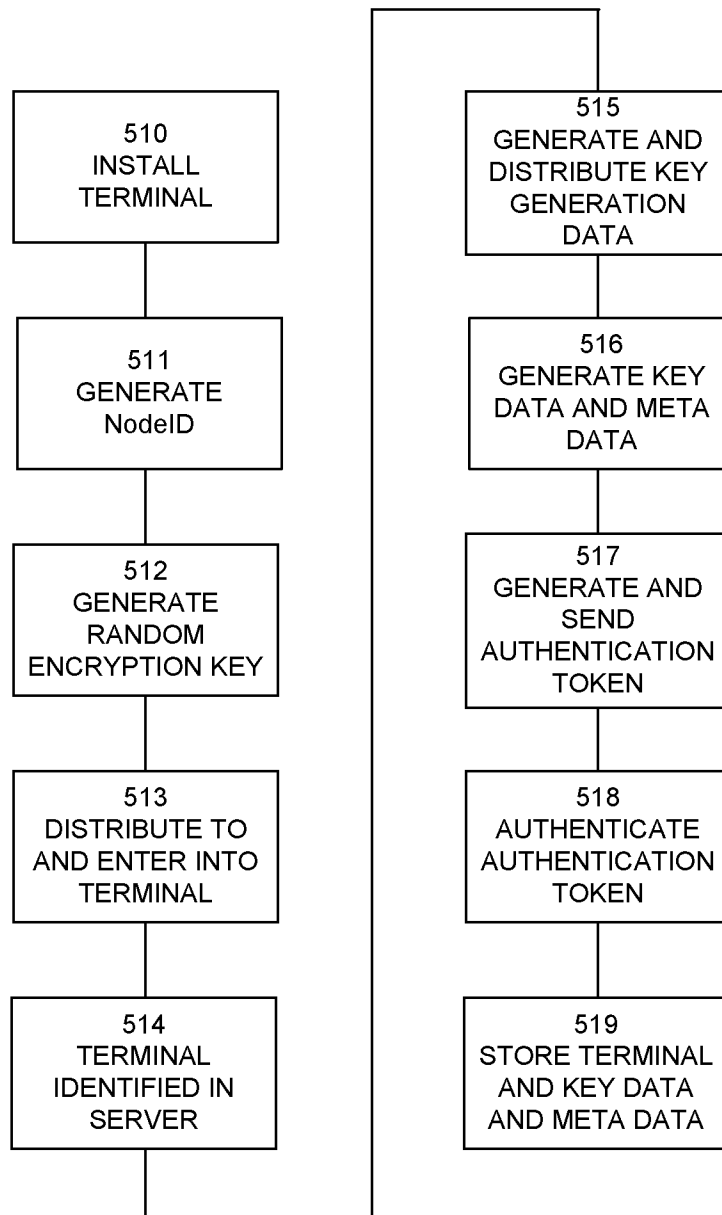


Fig. 6

5/11

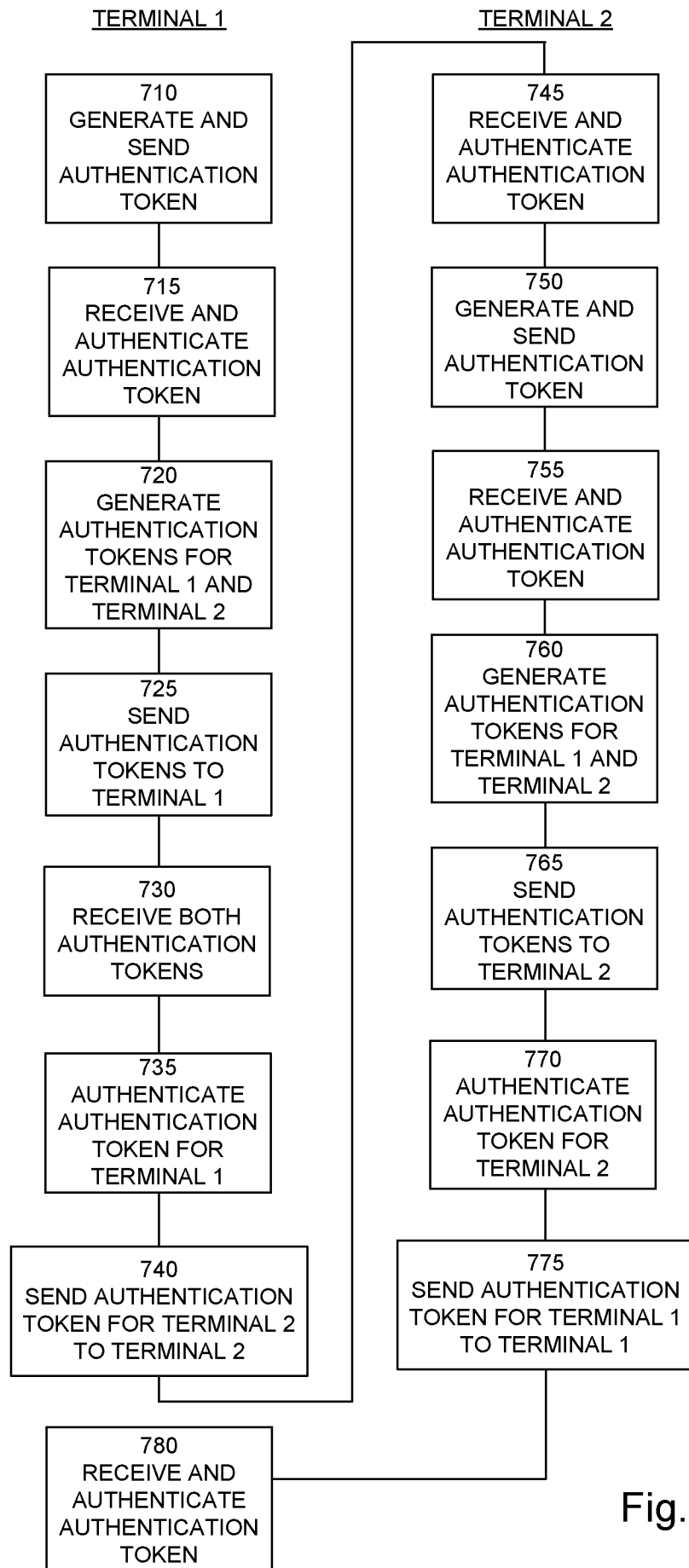


Fig. 7

6/11

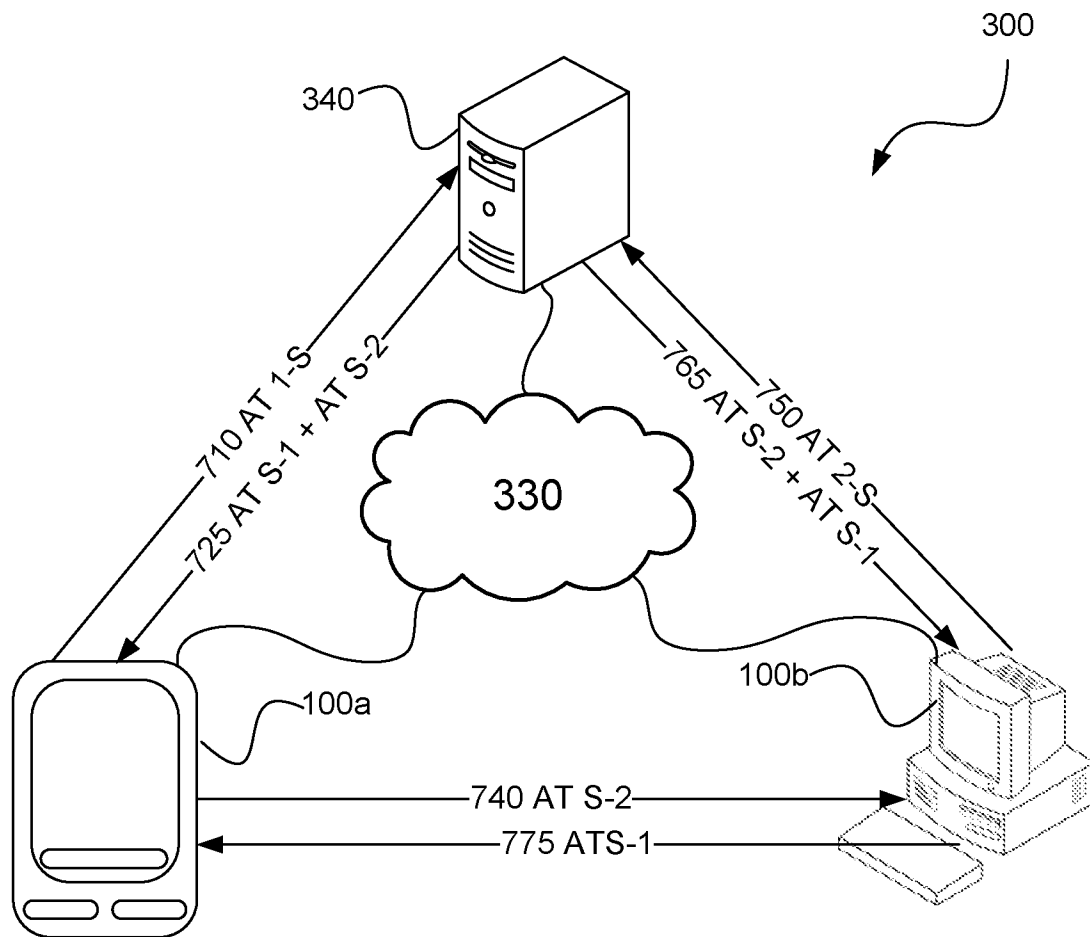


Fig. 8

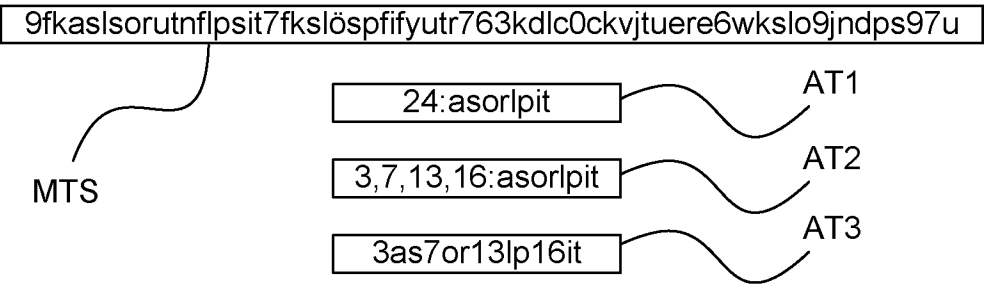


Fig. 9

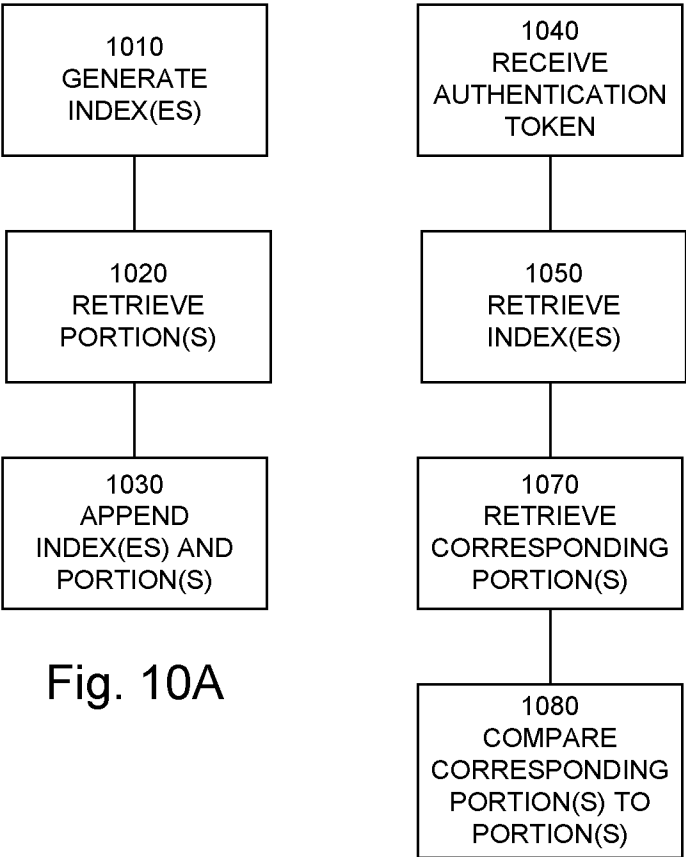


Fig. 10A

Fig. 10B

8/11

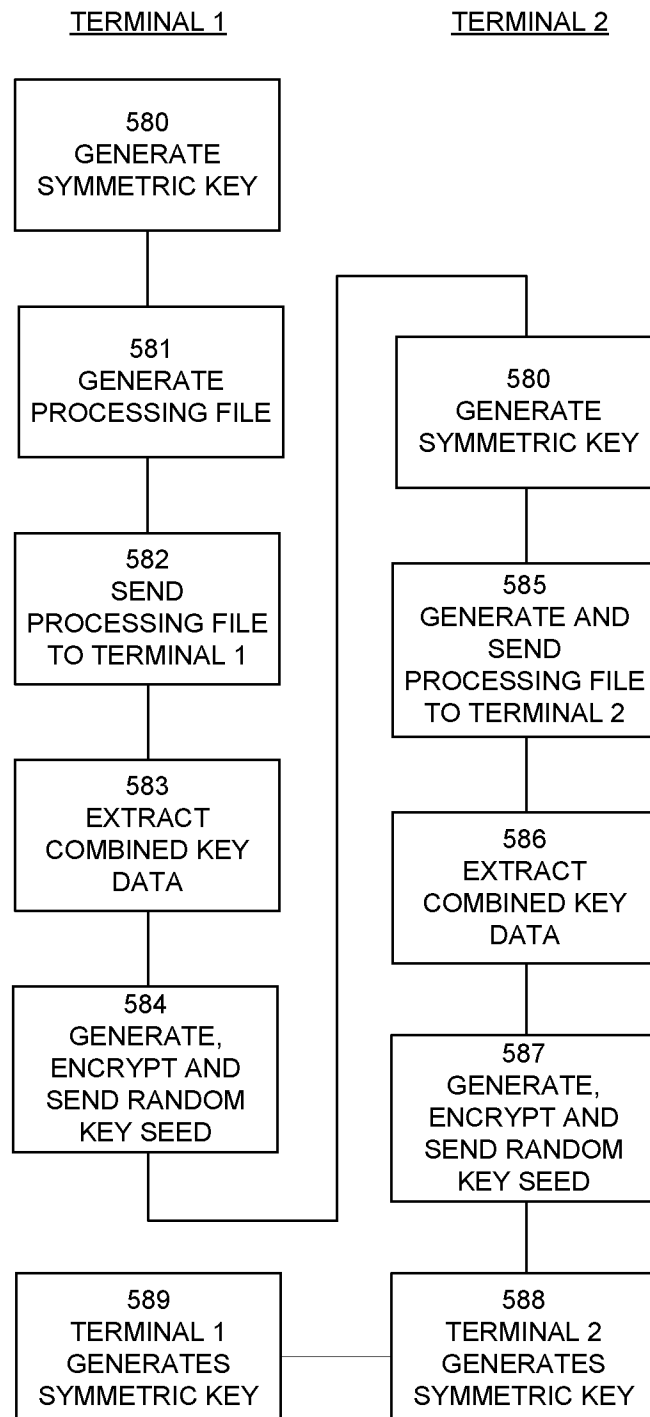


Fig. 11

9/11

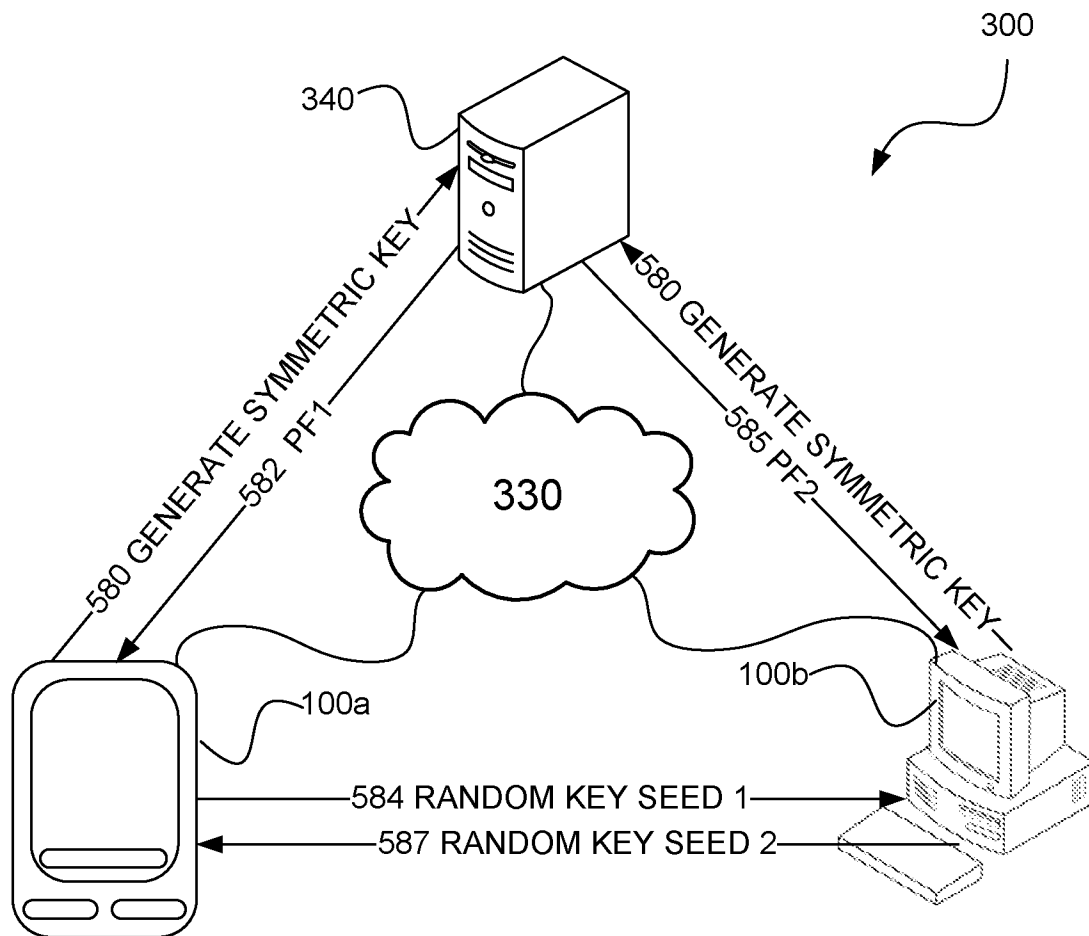


Fig. 12

10/11

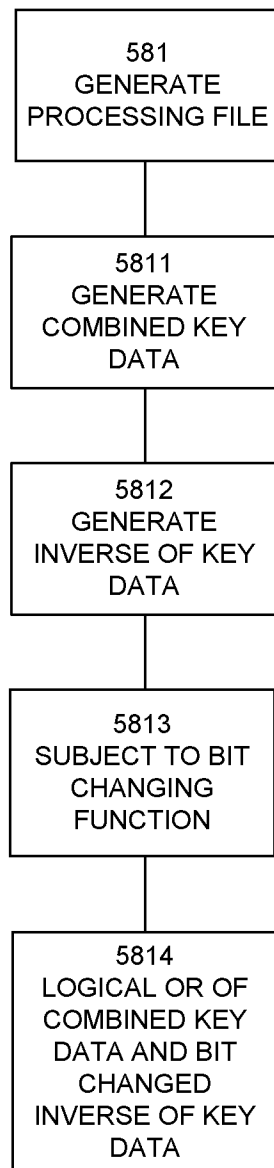


Fig. 13

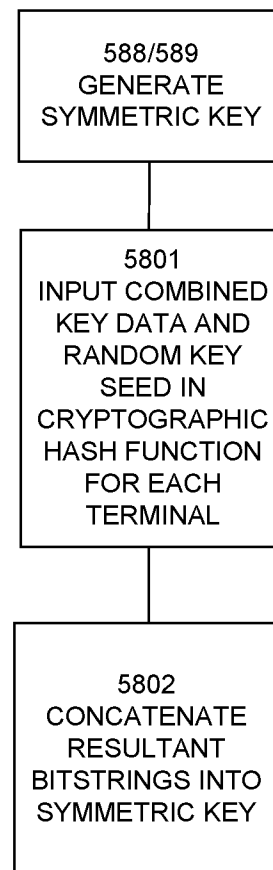


Fig. 14

11/11

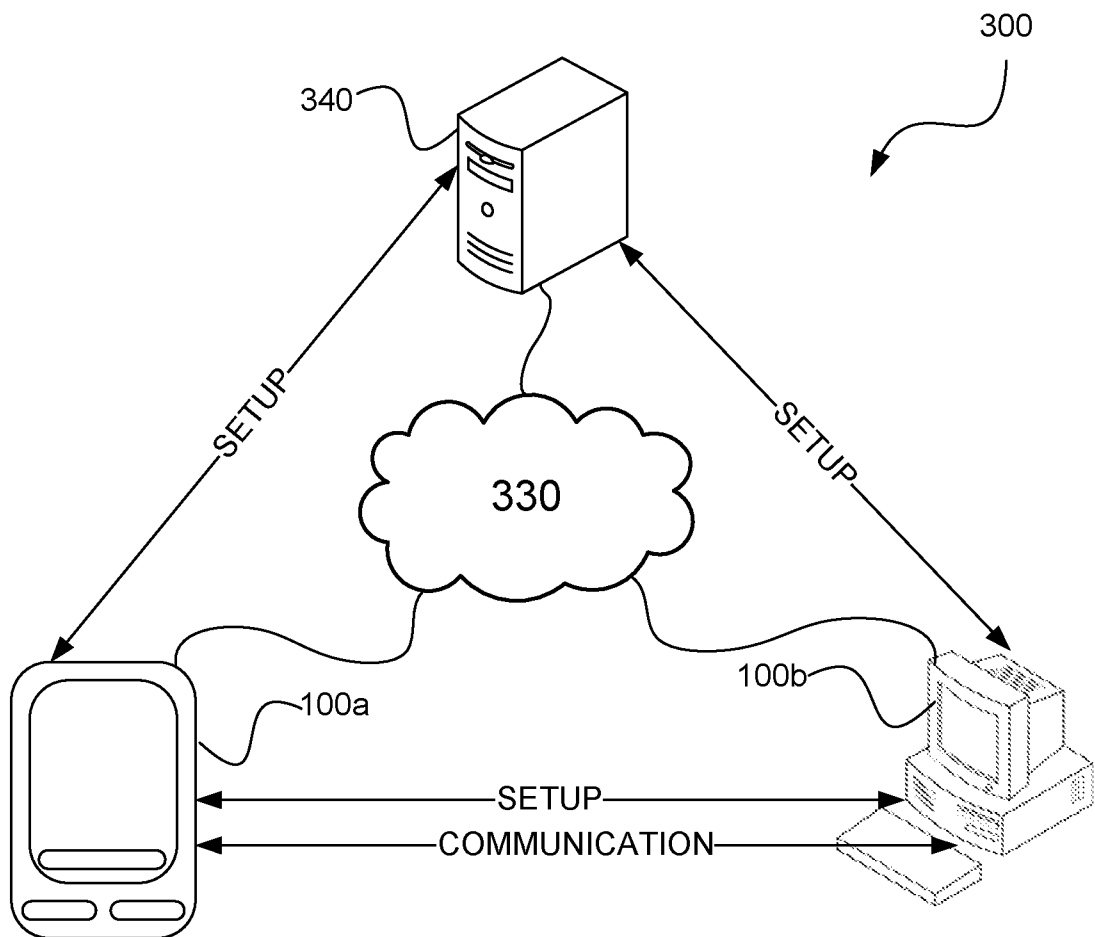


Fig. 15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE2015/051069

A. CLASSIFICATION OF SUBJECT MATTER

IPC: see extra sheet

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04L

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

SE, DK, FI, NO classes as above

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

EPO-Internal, PAJ, WPI data, COMPENDEX, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Alexander Hudic et al: "A Generation Method of Cryptographic Keys for Enterprise Communication Systems", 2013 International Conference on Availability, Reliability and Security, IEEE, 2013-09-02.; figure 3; Section III.B --	1-4, 8-13
A	WO 2011002412 A1 (URAEUS COMM SYSTEMS AB ET AL), 6 January 2011 (2011-01-06); abstract; claim 1 --	1-13
A	US 20100254533 A1 (MCCULLOUGH JEFFREY D ET AL), 7 October 2010 (2010-10-07); abstract -- -----	1-13

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

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

30-11-2015

Date of mailing of the international search report

01-12-2015

Name and mailing address of the ISA/SE

Patent- och registreringsverket

Box 5055

S-102 42 STOCKHOLM

Facsimile No. + 46 8 666 02 86

Authorized officer

Ralf Boström

Telephone No. + 46 8 782 28 00

Continuation of: second sheet**International Patent Classification (IPC)***H04L 9/08* (2006.01)*H04L 29/06* (2006.01)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/SE2015/051069

WO	2011002412 A1	06/01/2011	AU	2010266760 B2	10/04/2014
			CA	2747891 A1	06/01/2011
			CN	102227887 A	26/10/2011
			EA	201100887 A1	31/10/2011
			EA	019411 B1	31/03/2014
			EG	26868 A	10/11/2014
			EP	2361462 A4	23/11/2011
			HK	1157972 A1	19/06/2015
			IL	216897 A	30/07/2015
			JP	2012532486 A	13/12/2012
			JP	2015092745 A	14/05/2015
			KR	20120040127 A	26/04/2012
			MX	2012000104 A	25/06/2012
			NZ	596935 A	22/02/2013
			SE	0900918 A1	04/01/2011
			SE	534384 C2	02/08/2011
			SG	176707 A1	30/01/2012
			US	8433066 B2	30/04/2013
			US	20120087495 A1	12/04/2012
US	20100254533 A1	07/10/2010	CA	2698942 A1	06/10/2010
			MX	2010003709 A	14/10/2010
			US	8588410 B2	19/11/2013