The Patent Office
Innovation

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation and Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that the attached document is a printed office copy of the specification in respect of which the patent was granted on the application identified therein.

I also certify that subject to the payment of the prescribed renewal fees, the patent will remain in force for a period of twenty years from the date of the filing of the application.

I further certify that attached hereto is a true copy of the entries made to date in the Register of Patents in respect of the patent which is in force.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has been re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or the inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 24 October 2006

(12)UK Patent (19)GB (11)2 411 804 (13)B

(45) Date of publication: **11.10.2006**

(54) Title of the invention: **Method and apparatus for sending data from a server to a communication device**

(51) INT CL: **H04L 9/32** (2006.01) **H04L 29/06** (2006.01)

(21) Application No: **0509948.6**

(22) Date of Filing: **12.12.2003**

(30) Priority Data:
(31) **10334851** (32) **31.12.2002** (33) **US**

(86) International Application Data:
PCT/US2003/039726 En 12.12.2003

(87) International Publication Data:
WO2004/062189 En 22.07.2004

(43) Date A Publication: **07.09.2005**

(52) UK CL (Edition X):
H4P PDCSA PPEB

(56) Documents Cited:
EP 0539726 A
MENEZES et al, "Handbook of Applied Cryptography", 1997, CRC Press LLC, US, pp. 321-322,330,359-361,364-366,515-516
SCHNEIER B., "Applied Cryptography", Second edition, 1996, John Wiley & Sons, US, pp. 58-59

(58) Field of Search:
As for published application 2411804 A *viz:*
INT CL[7] **H04L**
Other: **EPO- INTERNAL, PAJ, INSPEC**
updated as appropriate

(72) Inventor(s):
Paul Drews
David Wheeler

(73) Proprietor(s):
Intel Corporation
(Incorporated in USA - Delaware)
2200 Mission College Boulevard,
Santa Clara, California 95052,
United States of America

(74) Agent and/or Address for Service:
Beresford & Co
16 High Holborn, LONDON, WC1V 6BX,
United Kingdom

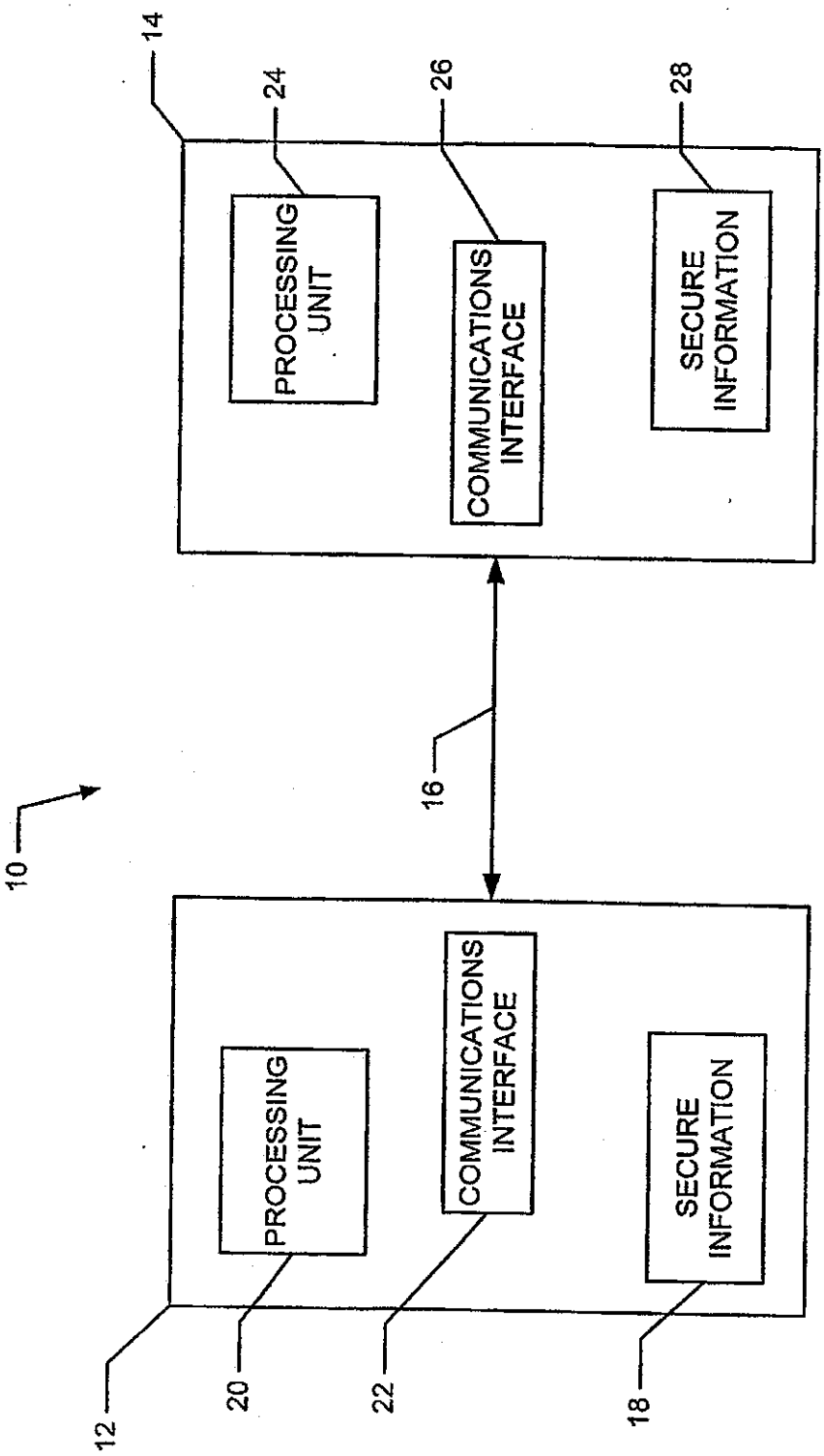
14
24
26
28
PROCESSING UNIT
COMMUNICATIONS INTERFACE
SECURE INFORMATION
16
10
PROCESSING UNIT
COMMUNICATIONS INTERFACE
SECURE INFORMATION
12
20
22
18

FIG. 1

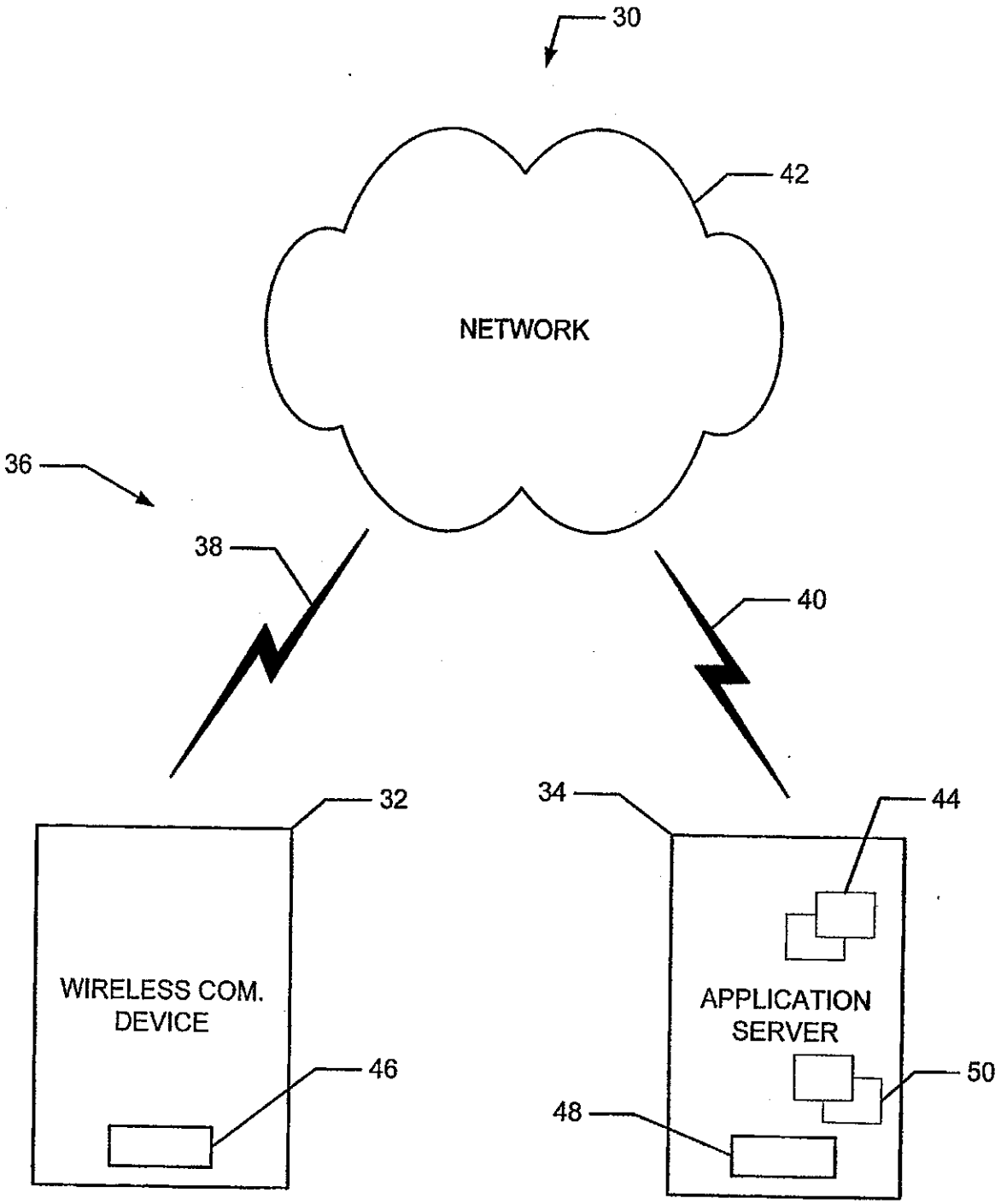
30
42
NETWORK
36
38
40
32
34
44
WIRELESS COM. DEVICE
APPLICATION SERVER
46
48
50

FIG. 2

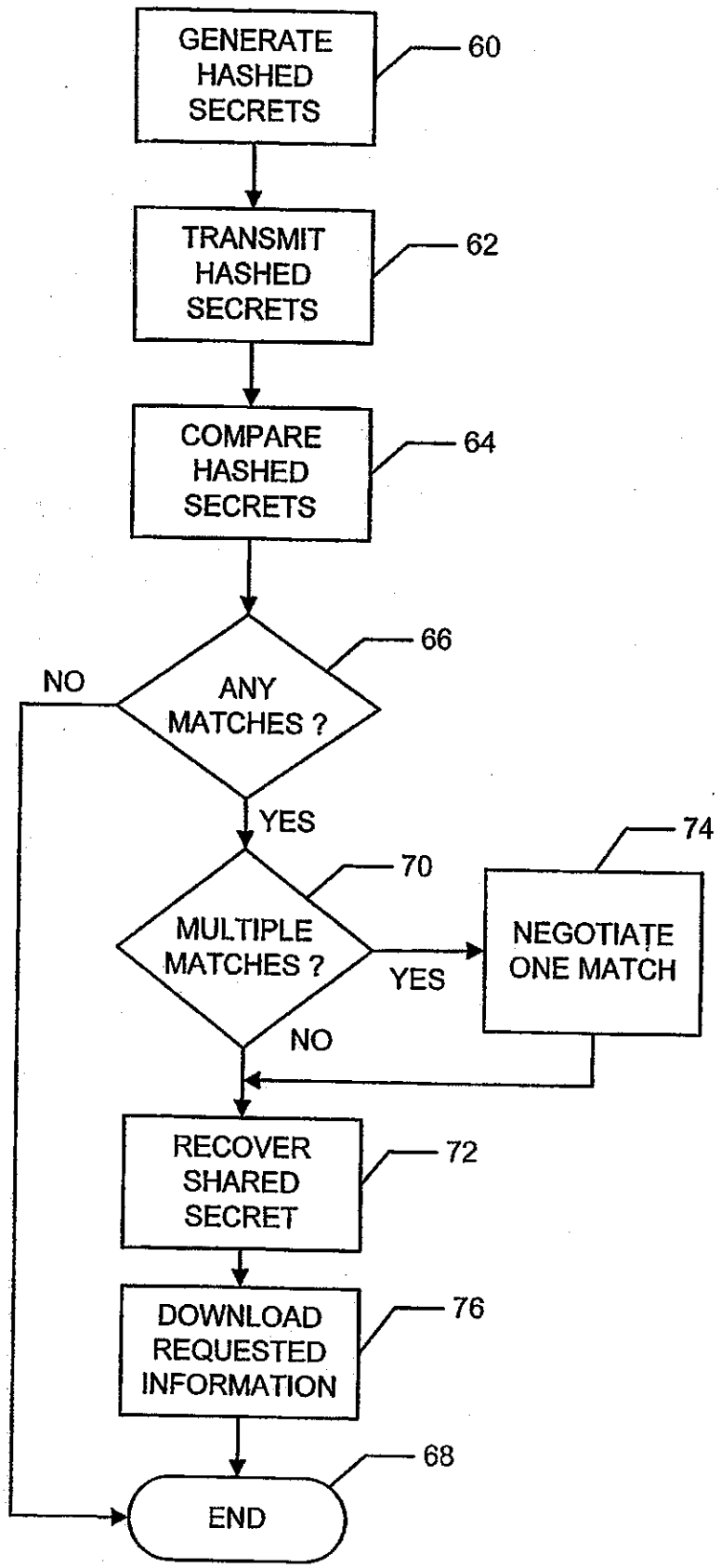
GENERATE HASHED SECRETS
60
TRANSMIT HASHED SECRETS
62
COMPARE HASHED SECRETS
64
ANY MATCHES ?
66
NO
YES
MULTIPLE MATCHES ?
70
YES
NO
NEGOTIATE ONE MATCH
74
RECOVER SHARED SECRET
72
DOWNLOAD REQUESTED INFORMATION
76
END
68

FIG. 3

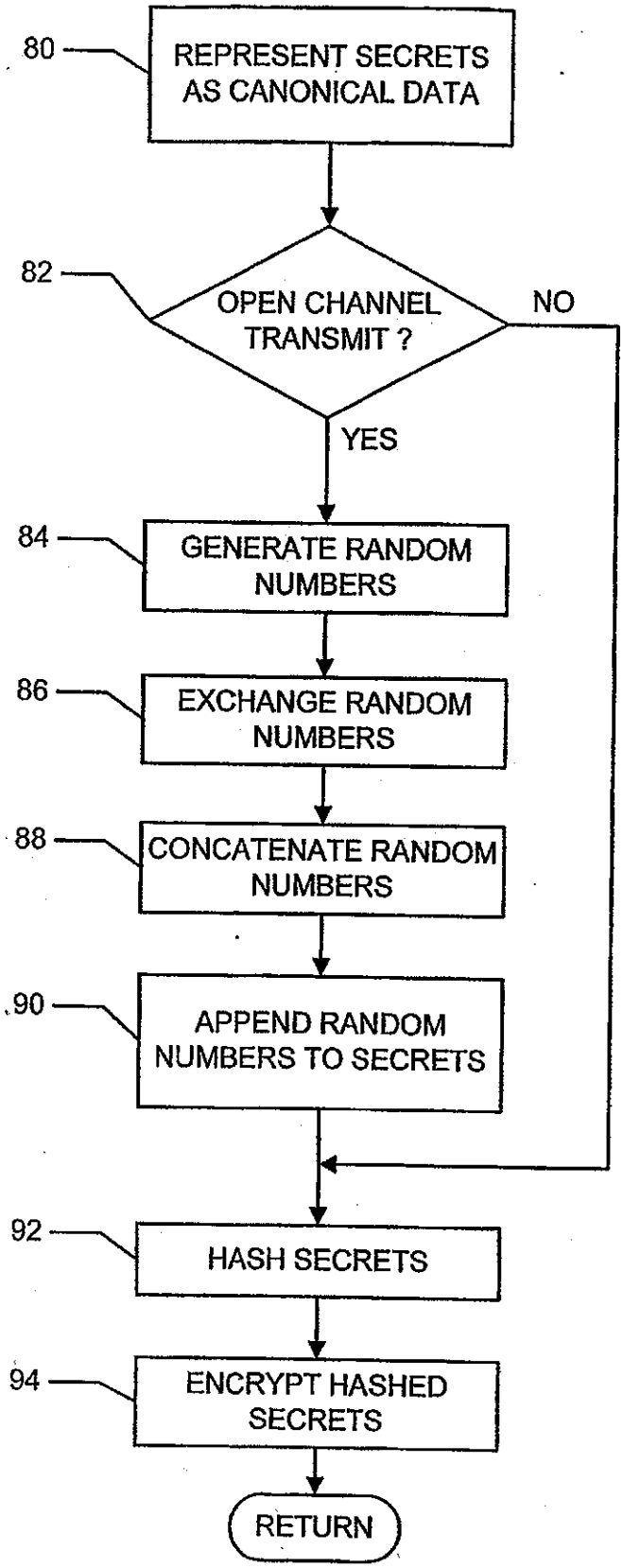
80
REPRESENT SECRETS AS CANONICAL DATA
82
OPEN CHANNEL TRANSMIT ?
NO
YES
84
GENERATE RANDOM NUMBERS
86
EXCHANGE RANDOM NUMBERS
88
CONCATENATE RANDOM NUMBERS
90
APPEND RANDOM NUMBERS TO SECRETS
92
HASH SECRETS
94
ENCRYPT HASHED SECRETS
RETURN

FIG. 4

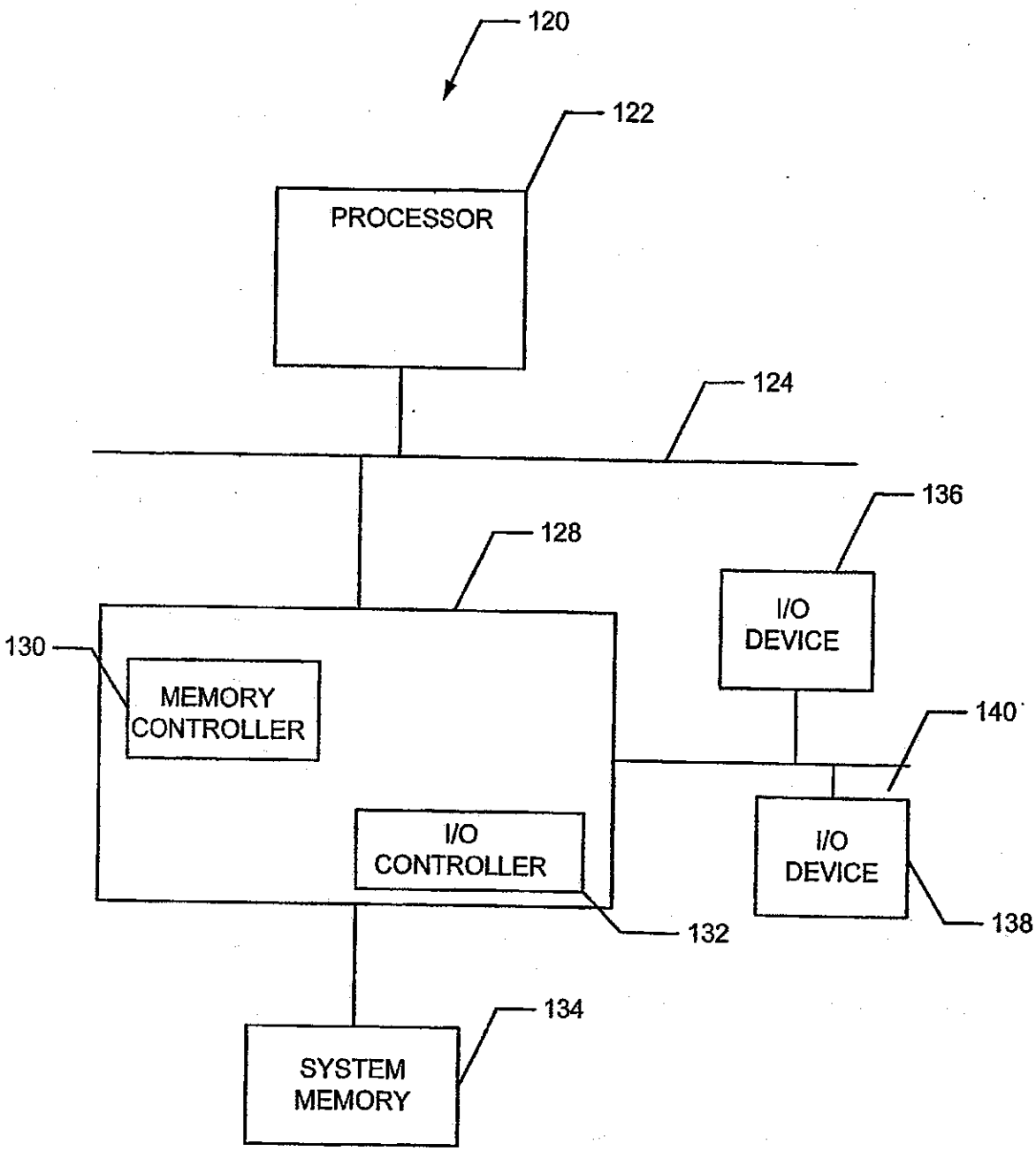
120
122
PROCESSOR
124
136
128
I/O
DEVICE
130
MEMORY
CONTROLLER
140
I/O
CONTROLLER
I/O
DEVICE
132
138
134
SYSTEM
MEMORY

FIG. 5

METHODS AND APPARATUS FOR FINDING A SHARED SECRET WITHOUT COMPROMISING NON-SHARED SECRETS

FIELD OF THE DISCLOSURE

[0001] The present disclosure relates generally to secure data communications and, more specifically, to methods and apparatus for finding a shared secret without compromising non-shared secrets.

BACKGROUND

[0002] The need for secure data communications or transfers has increased dramatically as the use of networked communications, particularly wireless communications (e.g., cellular communications), has become more accessible and widespread. For example, many mobile communication systems enable mobile clients (e.g., smart mobile phones, personal data assistants, etc.) to download software and access other data and/or services provided by data/application servers.

[0003] To provide a more secure environment for these mobile communication systems, mobile clients and servers may use a managed execution environment that provides a security function, which may be used to prevent unauthorized users from gaining access to data within the mobile clients and/or the data/application servers. For example, the Mobile Station Application Execution Environment (MExE) is a well-known wireless communication protocol that may be used with smart mobile phones and other mobile client devices to increase the security of data transactions between the mobile devices and the data/application servers.

[0004] Under the MExE security model, each mobile client device holds one or more digital certificates that designate the identity of the application server that must digitally sign software to enable that mobile client device to download and execute software from that application server. In other words, for a mobile client to download and execute an

application provided by a server, the mobile client must hold a digital certificate that corresponds to (e.g., is identical to) a digital certificate held by the server. Typically, application servers that supply software to mobile clients have multiple digital signatures of the software available for downloading. Each of these digital signatures may be created using a different digital certificate associated with a party authorized to create the software (e.g., a device manufacturer, a service provider, a software provider, etc.).

[0005] As is well known, a digital signature is typically generated by encrypting (e.g., using a private key from a public-private key combination) the hash of a message (e.g., a software application, a document, etc.) to be sent. In this manner, a digital signature can be used by a receiving entity to determine the identity of the originating entity and to determine that the received message has not been altered from what was sent by the originating entity. A digital certificate, on the other hand, typically contains a name (e.g., a user name), a serial number, a public key for encrypting data, expiration dates, and the signature of a certifying authority (certificate authority). In general, a digital certificate may be used to establish the credentials of an entity within a communication system or network and the public key portion of the certificate may be used to check or verify digital signatures.

[0006] In many mobile communication systems, the mobile clients freely provide or publish their digital certificates when negotiating a data transfer (i.e., establishing a communication link for data exchange) with another party (e.g., an application server). Similarly, application servers within these mobile communication systems may freely provide information relating to the digital signatures of available software or other data to clients requesting access to that software or other data.

[0007] Although it is generally desirable to provide digital signature and digital certificate information only to known authorized entities within the communication

network to maintain a high degree of network security, existing systems typically require this information to be released during the initial stages of a data transfer negotiation. Unfortunately, the release of digital certificate information or digital signature information during initial negotiations between two or more parties within a communication network can compromise the security of the network. In particular, the party releasing the digital certificate or signature information is typically unable to distinguish an authorized requesting entity from an attacker. Thus, if an attacker determines what digital certificates are authorized by, for example, a particular client device, the attacker can concentrate its efforts on overcoming a specific digital certificate. Likewise, if the attacker determines what digital signatures are authorized by a particular server, the attacker can concentrate its efforts on overcoming a specific digital signature.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Fig. 1 is a block diagram of an example system that may use the methods described herein to find one or more shared secrets without compromising non-shared secrets between entities negotiating a data transfer.

[0009] Fig. 2 is a block diagram depicting an example manner in which the system shown in Fig. 1 may be implemented.

[0010] Fig. 3 is a flow diagram of an example manner in which the systems shown in Figs. 1 and 2 may be configured to find one or more shared secrets without compromising non-shared secrets when negotiating a data transfer between entities.

[0011] Fig. 4 is a detailed flow diagram of an example manner in which hashed secrets may be generated for use with the example method shown in Fig. 3.

[0012] Fig. 5 is a block diagram of an example processor system that may be used to implement the apparatus and methods described herein.

DETAILED DESCRIPTION

[0013] As used herein, the term "secret" generally refers to secure data or information that is not published or freely distributed within a communication system or network. In some example systems, a secret may be a digital signature, a digital certificate, or some other information associated with a communication device, server, etc. that is not freely transmitted to entities requesting information. A "shared secret" refers to a secret that is common to (i.e., held by) more than one entity within the communication network or system. For example, a communication device and an application or data server within a communication system or network may both hold identical or corresponding digital certificates, thereby authorizing the communication device to download application information and/or other data from the server. However, entities sharing one or more secrets do not necessarily know that they share those secrets with another entity.

[0014] On the other hand, a "non-shared" secret refers to a secret that is not shared or commonly held between two or more entities. However, a non-shared secret for a first group of entities may be a shared secret for a second group of entities, even if the first and second groups of entities have common entity members. Thus, a determination of whether a secret is shared or non-shared must be referenced to a particular group of entities.

[0015] Fig. 1 is a block diagram of an example system 10 that may use the methods described herein to find one or more shared secrets without compromising non-shared secrets between entities negotiating a data transfer. The example system 10 includes a first party or entity 12 and a second party or entity 14, each of which is coupled to a communication channel 16. The communication channel 16 may include a packet-switched network such as, for example, the Internet, telephone lines, a wireless communication network(s) such as, for example, a cellular communications network, satellite links, etc. More generally, the communication channel 16 may be implemented

using any combination of media, platforms and/or protocols that enable the conveyance of information. Thus, the communication channel 16 may be public (e.g., in a case where the communication channel 16 includes the Internet) or private (e.g., in a case where the communication channel 16 uses only a local area network, phone lines, etc.)

[0016] The first party or entity 12 includes secure information 18, which preferably includes secrets such as data or information, which may be in the form of a list, table, etc., that is not freely transmitted on the communication channel 16 and which is not exposed or revealed to the second entity 14, or any other entities (not shown) coupled to the system 10. The first party or entity 12 also includes a processing unit 20, which may be implemented using any circuit or device capable of executing data or instructions to carry out the methods described herein, such as the example processor system shown and described in connection with Fig. 5 below,. In addition, the first entity 12 includes a communications interface 22, which enables the first entity 12 to communicate, if needed, via the communication channel 16. For example, in a case where the communication channel 16 includes phone lines, the communications interface 22 may include a modem and, in a case where the communication channel 16 includes a wireless network, the communications interface 22 may include a wireless transceiver.

[0017] Similar to the first party or entity 12, the second party or entity 14 includes secure information 28, a processing unit 24 and a communications interface 26. However, some or all of the secure information 28 held by the second entity 14 may be different from the secure information 18 held by the first entity 12. Further, while Fig. 1 depicts the system 10 as having only two entities or parties, additional entities or parties may be included and coupled to the communication channel 16.

[0018] Fig. 2 is a block diagram depicting an example manner in which the system 10 shown in Fig. 1 may be implemented. The example shown in Fig. 2 is a wireless

communication system 30 that includes a wireless mobile communication device or client 32 that can communicate with an application server 34 via a communication channel 36. The communication channel 36 includes one or more wireless communication links 38 and 40 and a switched-packet network 42, which may, for example, be the Internet.

[0019] The mobile communication device 32 may be, for example, a smart cellular phone, a PDA, a laptop computer having a wireless communications interface, etc. that is configured to communicate as a client of the application server 34 via the communication channel 36. The application server 34 may be implemented using a workstation or any other computer or processing system. The application server 34 includes one or more software applications and/or other data or information 44 available for downloading to and execution or use by mobile communication devices such as, for example, the mobile communication device 32.

[0020] The mobile communication device 32 includes or holds one or more digital certificates 46, which have not been revealed to (i.e., have been kept secret from) at least some other entities (e.g., the server 34) within the system 30. The digital certificates 46 secretly held by the communication device 32 correspond to one or more software applications or other data or information that the communication device 32 is authorized to transfer or download from other entities within the system 30. For purposes of discussion, at least one of the digital certificates 46 held by the communication device 32 authorizes the communication device 32 to download at least one of the software applications or other data or information 44 stored within the server 34. Thus, the mobile communication device 32 may hold a plurality of digital certificates, each of which may, for example, authorize the communication device 32 to download a different one or subset of the software applications or other data 44 stored within the server 34. However, in other examples, the digital certificates 46 held by the communication device 32 may not

authorize the communication device 32 to download any of the software applications or other data 44 from the application server 34. Instead, the communication device 32 may be authorized to transfer (e.g., download) data, applications, etc. from other entities or parties (e.g., other servers, which are not shown) within the system 30.

[0021] Similar to the wireless communication device 32, the application server 34 holds one or more digital certificates 48 that have not been revealed to at least some other entities (i.e., have been kept secret or secure) within the system 30. The digital certificates 48 held by the application server 34 correspond to the one or more applications or other data 44 available for download within the system 30. Thus, each of the digital certificates 48 may correspond to a particular one or subset of the applications or other data 44. As noted above, for purposes of discussion, at least one of the digital certificates 48 corresponds to a one or subset of the applications or other data 44 that the communication device 32 is authorized to download from the server 34. In addition to the software applications or other data 44, the application server 34 also includes digital signatures 50 of the applications or other data 44. Each of the digital signatures 50 is generated using a private key that corresponds to a public key associated with one of the digital certificates 48.

[0022] Fig. 3 is a flow diagram of an example manner in which the systems 10 and 30 shown in Figs. 1 and 2 may be configured to find one or more shared secrets without compromising non-shared secrets while negotiating a data transfer. For purposes of discussion, the method depicted in Fig. 3 is described in connection with the example system 30 shown in Fig. 2. However, the method depicted in Fig. 3 is generally applicable to any system in which two or more entities negotiate to find a secret that each entity holds.

[0023] In a case where the communication device 32 is in communication with the server 34 via the communication link 36 and initiates a negotiation for a transfer of one of the applications or other data 44, the communication device 32 and the application server 34 generate hashed secrets by generating hashed versions of their respective digital certificates 46 and 48 (block 60). Any desired hashing function such as, for example, SHA-1, may be used to hash the digital certificates 46 and 48. As is known, a hashing function typically computes a mathematical summary of the information being hashed. As a result, hashing a digital certificate composed of 4096 bytes of information may result in a hashed version of that digital certificate that can be represented using only twenty bytes. Hashing functions and their operation are well known and, thus, are not described in greater detail herein.

[0024] The hashed versions of the digital certificates 46 (i.e., the secrets) held by the communication device 32 are sent or transmitted to the application server 34 (block 62). As a result, the application server 34 holds a hashed version of its digital certificates 48 and a hashed version of the digital certificates 46 received from the communication device 32. The application server 34 then compares the hashed version of its digital certificates 48 to the hashed version of the digital certificates 46 it receives from the communication device 32 (block 64). Based on the comparison, the application server 34 determines if there are any matching hashed digital certificates (i.e., matching hashed secrets) (block 66). If there are no matching hashed secrets (i.e., the negotiating entities do not share any secrets) (block 66), the wireless communication device 32 is not authorized to download the requested information (e.g., one or more of the applications or other data 44) and the negotiation ends (block 68). On the other hand, if at least one match is found at block 66, the application server 34 determines if there are a plurality of matches (block 70) and, if only one match is found, the application server 34 uses the known relationship between

the matched hashed digital certificate and the original digital certificate to recover the original shared digital certificate (i.e., the original shared secret) (block 72). Once the match has been found and the shared secret recovered, the communication device 32 uses the recovered secret to download the requested information (e.g., one or more of the applications or other data 44) from the application server 34 (block 76). Preferably, the application server 34 uses the matched original digital certificate to find a corresponding one of the applications or other data 44, as well the corresponding one of the digital signatures, and downloads the corresponding application along with its digital signature to the communication device 32.

[0025] If more than one match is found (block 70), the wireless communication device 32 and the application server 34 negotiate to select a single match (block 74) prior to recovering the matched or shared digital certificate (block 72). To facilitate the selection of a single match at block 74, the communication device 32 may generate hashes of its digital certificates 46 using a predefined prioritization scheme. For example, digital certificates associated with manufacturers may be hashed first, digital certificates associated with carriers may be hashed second, and digital certificates associated with third-parties may be hashed last. The hashed digital certificates may then be stored in a list according to their priority (e.g., their order of hashing). In this manner, when the application server 34 receives the prioritized hashed list from the communication device 32 and compares the hashed digital certificates in that list to the hashed version of its digital certificates 48 (block 64), it can select the earliest found match as the single, negotiated match (block 74), despite the fact that other lower priority matches may remain in the list. In addition to the technique described above, there are many well-known techniques (which are not discussed in more detail herein) that may be used to negotiate one match (block 74).

[0026] Once a single match has been found and agreed on by the communication device 32 and the application server 34 (i.e., the negotiating parties or entities) (block 74) and the original secret has been recovered (block 72), the communication device 32 is authorized to download the requested one of the applications or other data 44 that corresponds to the matched certificate. Accordingly, the application server 34 downloads the requested one of the applications or other data 44 and, if requested, its corresponding one of the digital signatures 50 to the communication device 32 via the communication link 36.

[0027] Although it is not necessary, it may be desirable in some examples for each negotiating entity to receive a copy of the hashed digital certificates from the other negotiating entity (i.e., for the entities to exchange hashed secrets) and for each of the entities to make a comparison of its own hashed secrets to the hashed version of the secrets received from the other entity. Thus, in the case of the example system 30 shown in Fig. 3, the application server 34 may send or transmit (at block 62 or in a subsequent message) a hashed version of the digital certificates 48 to the communication device 32. In that case, the communication device 32 may make its own comparison of hashed secrets (block 64), determine if there are one or more matches (blocks 66 and 70) and may select a single match (block 74), independent from or in cooperation with the application server 34.

[0028] Fig. 4 is a detailed flow diagram of an example manner in which hashed secrets (e.g., hashed digital certificates) may be generated for use with the technique shown in Fig. 3. The entities generating hashed secrets (e.g., the communication device 32 and the application server 34) represent their secrets (e.g., their respective digital certificates 46 and 48) as canonical data (i.e., in a predefined format) (block 80). The entities involved in the negotiation for the data transfer then determine whether the data is to be transferred

(e.g., that an application is to be downloaded) via an open communication channel (block 82). For example, in the case where the communication device 32 is negotiating or requesting the application server 34 to download one of the applications or other data 44 via the communication link 36, an open channel transmission is involved. Namely, because the Internet 42 is an open communication channel, the entire communication channel 36 is considered open.

[0029] If it is determined at block 82 that the information is to be transferred via an open communication channel, the entities augment their hashed secrets with a random number by, for example, carrying out the steps described in connection with blocks 84 through 90 below. In particular, each of the entities (e.g., the communication device 32 and the application server 34) generates a random number (block 84) and exchange their random numbers via the open communication channel (e.g., the communication channel 36) (block 86). Preferably, but not necessarily, each of the random numbers is composed of at least sixty-four bits. The communication device 32 and the application server 34 then concatenate the random numbers (e.g., to form a one hundred twenty-eight bit number) that is used as a key (block 88). The concatenated random numbers are then appended to the secrets (block 90).

[0030] The secrets are hashed at block 92. In the case where an open channel transmission of the secrets is to occur (block 82) and, thus, random numbers are used to augment the secrets (e.g., by carrying out the activities described in connection with blocks 84 to 90), the concatenated random numbers are hashed together with the secrets to which they are appended. On the other hand, in the case where an open channel transmission of the secrets will not occur, the secrets are directly hashed (i.e., without any random number augmentation) at block 92.

[0031] After augmentation, the hashed secrets may optionally be encrypted (block 94) using, for example, the public key (from a private-public key combination) from the entity to which the hashed encrypted secrets are to be sent. For example, the communication device 32 would use the public key for the application server 34 to encrypt the hashed version of the secrets 46 (i.e., the digital certificates). The hashed secrets may be encrypted at block 94 as a group or individually. Following the transmission or exchange of encrypted hashed secrets (Fig. 3, block 62), in a case where the secrets have been augmented with one or more random numbers and where the hashed secrets have been encrypted, the entities may compare the hashed secrets to identify matches in the usual manner (i.e., block 64).

[0032] The use of random number augmentation (blocks 84 to 90) and encryption (block 94) for open channel communications can prevent an attacking entity from acquiring a copy of transmitted hashed secrets and using those hashed secrets to discover the secrets of other entities within the communication system. For example, an attacking entity could conceivably obtain a copy of transmitted hashed secrets and compare those hashed secrets to hashed versions of secrets it already has, thereby possibly enabling the attacking entity to gain access to information in the entity that originated the transmitted hashed secrets. Additionally or alternatively, the attacking entity could use the copy of the transmitted hashed secrets in a subsequent communication to falsely claim that a match was found (block 66 of Fig. 2), thereby tricking the entity that originated the transmitted hashed secrets into revealing the original secret. While the example method shown in Fig. 4 uses augmentation with a random number to provide replay protection (i.e., protection from an attacker trying to reuse captured or old information), any other technique that modifies the hashed secrets in a unique way that is verifiable by a receiving entity (and not by an attacker) could be used instead.

[0033] Further, the combination of random number augmentation with encryption provides both replay protection and confidentiality in the event that a data transfer is being negotiated over an open communication channel. In particular, random number augmentation provides replay protection because the random number changes for each data transactions and, thus, is essentially only good for one transaction. Encryption, on the other hand, provides a degree of confidentiality but, in the event an attacker overcomes the encryption, does not provide replay or reuse protection as does random number augmentation.

[0034] In cases where two entities or parties negotiate repeatedly and exchange secrets after negotiation, one of the entities could conceivably save a hashed secret value received from the other entity during an earlier negotiation and use that saved hashed secret to cause a false match at block 66. To prevent such a false mismatch, the well known Diffie-Hellman technique may be used to generate unique shared secret values for each negotiation. In particular, when generating hashed data, the entities or parties to the negotiation may use their unique shared secret values to perform a keyed hashing of the secrets associated with the entities. Keyed hashing techniques are well known and, thus, are not described further herein. Those having ordinary skill in the art will recognize that techniques other than Diffie-Hellman and keyed hashes may be used. In particular, any techniques that provide a secret that is shared by negotiating entities and which is unique to the negotiation may be used instead of the Diffie-Hellman technique. Additionally, any techniques that modify a resulting hash in a repeatable, non-reversible manner may be used instead of keyed hashes.

[0035] The example methods described in connection with Figs. 3 and 4 can be easily adapted for use with shared secret negotiations involving more than two entities or parties. For example, a first party may negotiate with a second party to develop a reduced list of

all matching hashed secrets. The first party may then negotiate with a third party, starting with the reduced list, to form a new reduced list containing matches between the list of the third party and the reduced list resulting from the negotiation between the first and second parties. This process continues until all parties involved in the shared secret negotiation have negotiated with the first party, which results in a final reduced list of hashed secrets that are shared between all the parties involved. Of course, if the involved parties do not have at least one common shared secret, the final list will be empty.

[0036] Fig. 5 is a block diagram of an example processor system 120 that may be used to implement the apparatus and methods described herein. As shown in Fig. 5, the processor system 120 includes a processor 122 that is coupled to an interconnection bus or network 124. The processor 122 may be any suitable processor, processing unit or microprocessor such as, for example, a processor from the Intel Itanium® family, Intel X-Scale® family, the Intel Pentium® family, etc. Although not shown in Fig. 5, the system 120 may be a multi-processor system and, thus, may include one or more additional processors that are identical or similar to the processor 122 and which are coupled to the interconnection bus or network 124.

[0037] The processor 122 of Fig. 5 is coupled to a chipset 128, which includes a memory controller 130 and an input/output (I/O) controller 132. As is well known, a chipset typically provides I/O and memory management functions as well as a plurality of general purpose and/or special purpose registers, timers, etc. that are accessible or used by one or more processors coupled to the chipset. The memory controller 130 performs functions that enable the processor 122 (or processors if there are multiple processors) to access a system memory 134, which may include any desired type of volatile memory such as, for example, static random access memory (SRAM), dynamic random access memory (DRAM), etc. The I/O controller 132 performs functions that enable the

processor 122 to communicate with peripheral input/output (I/O) devices 136 and 138 via an I/O bus 140. The I/O devices 136 and 138 may be any desired type of I/O device such as, for example, a keyboard, a video display or monitor, a mouse, etc. While the memory controller 130 and the I/O controller 132 are depicted in Fig. 5 as separate functional blocks within the chipset 128, the functions performed by these blocks may be integrated within a single semiconductor circuit or may be implemented using two or more separate integrated circuits.

[0038] As can be seen from the examples described herein, the use of hashed secrets enables a plurality of entities or parties (e.g., communication devices, data/application servers, etc.) that interact with one another (e.g., to carry out a data transfer) to discover shared secrets without revealing non-shared secrets. Such secrets may be digital certificates, digital signatures, message authentication code keys, ephemeral Diffie-Hellman parameters, etc. In general, the secrets may be any information that enables a receiving device (e.g., a client device) to authenticate the veracity of the origin of the information (e.g, data, a program, etc.) being received. Cryptographic techniques and replay protection (e.g., random number augmentation) may be used to exchange the hashed secrets via a public communication channel to prevent unauthorized or attacking entities from discovering secrets. The parties may compare their hashed secrets to the hashed secrets received from the other party to identify any matches. A single best match may be agreed upon and the parties may use knowledge of their respective hashing methods to recover the original secret from the hashed matching secret and to enable subsequent use of the shared secret (e.g., to carry out a data transfer) to occur. The use of hashed data (e.g., hashed digital certificates) reduces the amount of data that must be communicated via the communication channel during the shared secret negotiation

process. In addition, the apparatus and methods described herein may be more generally applied to shared secret negotiations involving more than two parties.

[0039] While the examples described herein focus on the downloading of applications (i.e., executable programs), the apparatus and methods described herein can be generally applied to any type of data. For example, ring tones for phones, small amounts of data for applications that are already installed on phones, PDAs, etc. could be downloaded as well.

[0040] Although certain methods and apparatus have been described herein, the scope of coverage of this patent is not limited thereto. To the contrary, this patent covers all methods, apparatus and articles of manufacture fairly falling within the scope of the appended claims either literally or under the doctrine of equivalents.

What is claimed is:

1. A method of finding a shared secret, comprising:

hashing first information associated with a first entity coupled to a communication channel to form a first hashed secret;

hashing second information associated with a second entity coupled to the communication channel to form a second hashed secret;

sending the first hashed secret to the second entity via the communication channel; and

comparing the first and second hashed secrets.

2. A method as defined in claim 1, wherein hashing the first information includes hashing a digital certificate.

3. A method as defined in claim 1, wherein hashing the second information includes hashing a digital certificate.

4. A method as defined in claim 1, including representing the first and second information in a canonical form prior to hashing the first and second information.

5. A method as defined in claim 1, including respectively augmenting the first and second information with first and second random numbers prior to sending the first hashed secret to the second entity via the communication channel.

6. A method as defined in claim 5, wherein augmenting the first and second information with the first and second random numbers includes concatenating the first and second random numbers to form a concatenated random number and appending the concatenated random number to the first and second information to form first and second augmented information.

7. A method as defined in claim 6, including hashing the first and second augmented information to form the first and second hashed secrets.

8. A method as defined in claim 5, including determining if the communication channel is an open channel prior to augmenting the first and second information with the first and second random numbers.

9. A method as defined in claim 1, including encrypting the first hashed secret prior to sending the first hashed secret to the second entity via the communication channel.

10. A method as defined in claim 9, wherein encrypting the first hashed secret includes using a first public key associated with the second entity to encrypt the first hashed secret.

11. A method as defined in claim 1, wherein comparing the first and second hashed secrets includes determining that the first and second hashed secrets match.

12. A method as defined in claim 1, wherein the first information includes a first group of secrets associated with the first entity and the second information includes a second group of secrets associated with the second entity.

13. A method as defined in claim 12, wherein the first group of secrets includes a first group of digital certificates and the second group of secrets includes a second group of digital certificates.

14. A method as defined in claim 13, wherein comparing the first and second hashed secrets includes finding a first digital certificate from the first group of digital certificates that matches a second digital certificate from the second group of digital certificates.

15. A method as defined in claim 1, wherein the first and second entities negotiate to select a unique secret value.

16. A method as defined in claim 15, wherein the first and second entities negotiate using a Diffie-Hellman technique.

17. A method as defined in claim 15, wherein the first and second hashed secrets are uniquely associated with the unique secret value.

18. A method as defined in claim 17, wherein the first and second hashed secrets are keyed hashes based on the unique secret value.

19. A system for finding a shared secret, comprising:

a first entity coupled to a communication channel, wherein the first entity hashes first information to form a first hashed secret; and

a second entity coupled to the communication channel, wherein the second entity hashes second information to form a second hashed secret, and wherein the first entity is adapted to send the first hashed secret to the second entity via the communication channel and the second entity is adapted to compare the first and second hashed secrets.

20. A system as defined in claim 19, wherein first information includes a first group of digital certificates and the second information includes a second group of digital certificates.

21. A system as defined in claim 19, wherein the first and second information is in a canonical form.

22. A system as defined in claim 19, wherein the first and second entities are adapted to augment their first and second information with respective first and second random numbers.

23. A system as defined in claim 19, wherein the first entity is adapted to encrypt the first hashed secrets.

24. A system as defined in claim 19, wherein the first and second entities are adapted to negotiate to select a unique secret value.

25. A system as defined in claim 24, wherein the first and second entities negotiate using a Diffie-Hellman technique.

26. A system as defined in claim 24, wherein the first and second hashed secrets are uniquely associated with the unique secret value.

27. A system as defined in claim 26, wherein the first and second hashed secrets are keyed hashes based on the unique secret value.

28. A machine accessible medium having data stored thereon that, when executed, causes a machine to:

hash first information associated with a first entity coupled to a communication channel to form a first hashed secret;

receive a second hashed secret at the first entity from the second entity via the communication channel; and

compare the first and second hashed secrets to find a shared secret.

29. A machine accessible medium as defined in claim 28, wherein the first information comprises a digital certificate.

30. A machine accessible medium as defined in claim 28, wherein the second hashed secret is formed by hashing a digital certificate.

31. A machine accessible medium as defined in claim 28 having data stored thereon that, when executed, causes the machine to augment the first information with a first random number.

32. A machine accessible medium as defined in claim 31 having data stored thereon that, when executed, causes the machine to determine if the communication channel is an open channel prior to augmenting the first information with the first random number.

33. A machine accessible medium as defined in claim 28 having data stored thereon that, when executed, causes the machine to encrypt the first hashed secret prior to sending the first hashed secret to the second entity via the communication channel.

34. A machine accessible medium as defined in claim 28 the first and second hashed secrets match.

35. A machine accessible medium as defined in claim 34, wherein the first hashed secret corresponds to a first digital certificate from a first group of digital certificates and the second hashed secret corresponds to a second digital certificate from a second group of digital certificates.

36. A machine accessible medium as defined in claim 28 having data stored thereon that, when executed, causes the machine to negotiate to select a unique secret value.

37. A machine accessible medium as defined in claim 36, wherein the negotiation is based on a Diffie-Hellman technique.

38. A method as defined in claim 36, wherein the first and second hashed secrets are uniquely associated with the unique secret value.

39. A method as defined in claim 38, wherein the first and second hashed secrets are keyed hashes based on the unique secret value.

40. A system for negotiating a data transfer via a communication channel, comprising:

a first entity coupled to the communication channel and having first secure information stored therein; and

a second entity coupled to the communication channel and having second secure information stored therein, wherein the first and second entities are programmed to:

hash the first and second secure information to form first and second hashed information;

transmit at least one of the first and second hashed information via the communication channel;

compare the first and second hashed information to find matching secure information; and

initiate the data transfer in response to finding the matching secure information.

41. A system as defined in claim 40, wherein the first entity is a communication device and the second entity is a server.

42. A system as defined in claim 41, wherein the server includes at least one of an application, data and information, at least one of which the communication device is authorized to download.

43. A system as defined in claim 42, wherein the server includes a plurality of digital signatures corresponding to the plurality of applications.

44. A system as defined in claim 41, wherein the communication device is a mobile communication device and the server is an application server.

45. A system as defined in claim 40, wherein the first and second secure information includes first and second respective groups of digital certificates.

46. A system as defined in claim 45, wherein at least one digital certificate within the first group of digital certificates matches a digital certificate within the second group of digital certificates.

47. A system as defined in claim 40, wherein the communication channel includes a wireless communication link.

48. A system as defined in claim 40, wherein the communication channel includes a packet-switched network.

49. A system as defined in claim 40, wherein the first and second secure information is stored in a canonical form.

50. A system as defined in claim 40, wherein the first and second entities are programmed to augment the first and second secure information with random numbers prior to transmitting the at least one of the first and second hashed information via the communication channel.

51. A system as defined in claim 40, wherein the first and second entities are programmed to encrypt the at least one of the first and second hashed information prior to exchanging the at least one of the first and second hashed information via the communication channel.

52. A system as defined in claim 40, wherein the data transfer includes the downloading at least one of an application, data and information from the second entity to the first entity.

53. A method of sending an application from an application server to a communication device via a communication channel, comprising:

receiving a first group of hashed secrets from the communication device;

comparing the first group of hashed secrets to a second group of hashed secrets associated with the application server;

identifying a shared secret among the first and second groups of hashed secrets; and

sending an application associated with the shared secret to the communication device via the communication channel.

54. A method as defined in claim 53, wherein receiving the first group of hashed secrets from the communication device includes receiving hashed digital certificates associated with the communication device.

55. A method as defined in claim 53, wherein identifying the shared secret among the first and second groups of hashed secrets includes identifying a shared digital certificate.

56. A method as defined in claim 53, wherein receiving the first group of hashed secrets includes receiving secrets that have been augmented with a random number.

57. A method as defined in claim 53, wherein receiving the first group of hashed secrets includes receiving encrypted secrets.

58. An application server, comprising:

secure information stored within the application server;

a communications interface adapted to be coupled to a communication channel; and

a processor unit programmed to cause the application server to:

receive a first group of hashed information from a communication device via the communications interface;

hash at least a portion of the secure information to form a second group of hashed information;

compare the first group of hashed information to the second group of hashed information; and

identify a shared secret among the first and second groups of hashed information.

59. A system as defined in claim 58, wherein the secure information includes a plurality of digital certificates.

60. A system as defined in claim 58, wherein the communication channel includes a public communication channel.

61. A machine accessible medium having data stored thereon that, when executed, causes a machine to:

receive a first group of hashed secrets from a communication device;

compare the first group of hashed secrets to a second group of hashed secrets associated with an application server;

identify a shared secret among the first and second groups of hashed secrets; and

send an application associated with the shared secret to the communication device via the communication channel.

62. A machine accessible medium as defined in claim 61 having data stored thereon that, when executed, causes the machine to receive the first group of hashed secrets from the communication device by receiving hashed digital certificates associated with the communication device.

63. A machine accessible medium as defined in claim 61 having data stored thereon that, when executed, causes the machine to identify the shared secret among the first and second groups of hashed secrets by identifying a shared digital certificate.

64. A communication device, comprising:

secure information stored within the communication device;

a communications interface adapted to communicate via a communication channel; and

a processor unit programmed to cause the communication device to:

receive a first group of hashed information via the communication channel;

hash at least a portion of the secure information to form a second group of hashed information;

compare the first and second groups of hashed information to identify matching information; and

receive data via the communication channel based on the matching information.

65. A communication device as defined in claim 64, wherein the secure information includes a digital certificate associated with the communication device.

66. A communication device as defined in claim 64, wherein the communications interface includes a wireless communication interface.

67. A method of finding a shared secret, comprising:

receiving encoded data via a communication channel;

encoding local data; and

comparing the received encoded data to the encoded local data to find the shared secret.

68. A method as defined in claim 67, wherein the received encoded data is associated with a first digital certificate and wherein the encoded local data is associated with a second digital certificate.

RENEWAL DETAILS

PUBLICATION NUMBER GB2411804

PROPRIETOR(S)

Intel Corporation, Incorporated in USA - Delaware, 2200 Mission College Boulevard, Santa Clara, California 95052, United States of America

DATE FILED 12.12.2003

DATE GRANTED 11.10.2006

DATE NEXT RENEWAL DUE 12.12.2007

DATE NOT IN FORCE

DATE OF LAST RENEWAL

YEAR OF LAST RENEWAL 00

STATUS PATENT IN FORCE

**** END OF REPORT ****

REGISTER ENTRY FOR GB2411804

Form NP1 Application No GB0509948.6 filing date 12.12.2003

Lodged on 16.05.2005

Priority claimed:
31.12.2002 in United States of America - doc: 10334851

PCT NATIONAL PHASE
PCT Application PCT/US2003/039726 filed on 12.12.2003 in English
Publication No WO2004/062189 on 22.07.2004 in English

Title METHODS AND APPARATUS FOR FINDING A SHARED SECRET WITHOUT COMPROMISING NON-SHARED SECRETS

Applicant/Proprietor
INTEL CORPORATION, Incorporated in USA - Delaware, 2200 Mission College Boulevard, Santa Clara, California 95052, United States of America
[ADP No. 00518647002]

Inventors
PAUL DREWS, 2190 NW Phillips Road, Gaston, OR 97119, United States of America
[ADP No. 09097510001]

DAVID WHEELER, 516 East Jasper Drive, Gilbert, AZ 85296, United States of America
[ADP No. 09097528001]

Classified to
H4P
H04L

Address for Service
BERESFORD & CO, 16 High Holborn, LONDON, WC1V 6BX, United Kingdom
[ADP No. 00001826001]

Publication No GB2411804 dated 07.09.2005

Examination requested 16.05.2005

Grant of Patent (Notification under Section 18(4)) 12.09.2006
Publication of notice in the Patents and Designs Journal (Section 25(1)) 11.10.2006
Title of Granted Patent METHOD AND APPARATUS FOR SENDING DATA FROM A SERVER TO A COMMUNICATION DEVICE

**** END OF REGISTER ENTRY ****