**Title:** METHOD AND APPARATUS FOR SECURE COMMUNICATION

**Abstract:** A many to many data exchange system comprises: a network comprising at least two DSCs (601, 603) and at least two hosts (610, 621) in which each of the at least two hosts share at least one key material with at least one of the at least two DSCs; data exchange means for performing a data exchange process, the data exchange process utilising a subset of at least two of the at least two DSCs and comprising: the first host interacting with at least one DSC, each of which is adapted to share at least one key material with the first host; and the second host interacting with at least one DSC each of which is adapted to share at least one key material with the second host; the data exchange means comprising: secure communication paths between the first host and its corresponding DSC it shares key material with, the secure communications paths adapted to use a cryptographic primitive keyed with the shared key material; and secure communication paths between the second host and its corresponding DSC it shares key material with, the secure communications paths adapted to use a cryptographic primitive keyed with shared key material; and at least two distinct communications paths for transporting data between the first and second host wherein at least one of the first host's DSC that it shares key material with is on the first distinct communications path and at least one of the second host's DSC that it shares key material with is on the second distinct communications path; and in which the first host sends to the second host: at least one first data over the first distinct communications path; and at least one second data over the second distinct communications paths.
METHOD AND APPARATUS FOR SECURE COMMUNICATION
RELATED APPLICATIONS

The present application claims priority from Australian provisional patent applications in the name of Synaptic Laboratories Limited:

5 Number 2007906076, filed on 5 November 2007, entitled "Method and Apparatus for Secure Communication"; and

Number 2008901696, filed on 8 April 2008, entitled "Method and Apparatus for Secure Communication",
the contents of each of which is incorporated herein.

FIELD OF THE INVENTION

The present invention relates to computer network security. In one form the present invention relates to a method and system for cryptographic key exchange. The present specification describes a preferred embodiment of the invention in relation to performing a candidate post quantum secure key exchange over the Internet between a first and second party with the assistance of at least two third parties, where each third party is trusted by at least the first or second party and when acting honestly should not be able to decrypt the communications of the session established using the key exchange between the first and second party. However, it should be appreciated that the invention is not limited to that use, only.

BACKGROUND OF THE INVENTION

It is likely that none of the mainstream public key encryption cryptographic systems based on factoring, on the discrete logarithm problem or elliptic curves will be adequately secure in times of quantum super computers.

Embodiments of the present invention accordingly aim at improving the security of key exchanges.

Throughout the specification we use the terms:
"post quantum secure" (PQS) to refer to a communication that is secure against both classical computing attacks and quantum super computing attacks;
"extranet security" to describe the encryption of messages flowing across a network such that the content of the messages may be discovered by, or directly exposed to, authorised nodes of the network
but not by unauthorised nodes on the network or other adversaries monitoring the network;

In this description and the appended claims the term "key material" is to be taken as meaning a portion or portions of information used to form a secret session key and which may include the use of the entirety of the information used to form a secret session key for use in secure communications. Key material is a type of data.

In this description and the appended claims the term "key support centre" means an element that receives key material, and transforms the key material using a cryptographic primitive on behalf of one or more parties. A KSC is similar to and may be adapted to perform some of the functions of a KTC (key translation centre), a KDC (key data centre), or both. However, a KSC need not be a absolutely trusted third party. Herein the term KSC is used to indicate one key support centre (singular) or more than one (plural) key support centre. Depending on context a KSC may refer to either the process, the process running on an apparatus, or the apparatus running the process.

In this description and the appended claims the term "data support centre" (DSC) means a more general implementation of a KSC such that is adapted to perform cryptographic transformations on data. It is always valid to describe a KSC as a DSC. When the data being cryptographically transformed comprises key material it is then also valid to describe the DSC as a KSC. Herein the term DSC is used to indicate one data support centre (singular) or more than one (plural) data support centre. Depending on context a DSC may refer to either the process, the process running on an apparatus, or the apparatus running the process.

In this description and the appended claims the term "host" means a computing device connected to a network. Every host is a node on the network, but not every node is a host on the network. For example, network nodes such as modems and network switches are not considered hosts. We exclusively use the term "host" in the context of a network session established between at least two nodes according to preferred embodiments of the present invention. In this context two host nodes may establish a network session according to a preferred embodiment of the present invention that is
facilitated by other nodes on the network, including but not limited to switch nodes, router nodes, and nodes running network services such as DSC and KSC according to preferred embodiments of the present invention. Depending on the context a host may refer to either a process, the process running on an apparatus, or the apparatus running the process.

Depending on the context the term "user" means either the human controller of a host node, or the host node itself. In this way, it is permissible to talk about a network comprising 6 nodes where 2 of the 6 nodes are "users", such that a first user communicates with a second user by relaying messages across at least one of the other 4 remaining nodes connecting the 2 nodes the users are controlling. It is also permissible to talk about a human operator controlling a node, such that a user configures a host node to trust one or more other nodes such as a DSC node.

Any discussion of documents, devices, acts or knowledge in this specification is included to explain the context of the invention. It should not be taken as an admission that any of the material forms a part of the prior art base or the common general knowledge in the relevant art on or before the priority date of the present application.

SUMMARY OF THE INVENTION

In one aspect preferred embodiments of the present invention provide a many to many key exchange system comprising:

a network comprising at least two DSCs and at least two hosts in which each of the at least two hosts share at least one key material with at least one of the at least two DSCs;

data exchange means for performing a data exchange process, the data exchange process utilising a subset of at least two of the at least two DSCs and comprising:

the first host interacting with at least one DSC, each of which is adapted to share at least one key material with the first host;

and

the second host interacting with at least one DSC each of which is adapted to share at least one key material with the second host;

the data exchange means comprising:
secure communication paths between the first host and its corresponding DSC it shares key material with, the secure communications paths adapted to use a cryptographic primitive keyed with the shared key material; and

secure communication paths between the second host and its corresponding DSC it shares key material with, the secure communications paths adapted to use a cryptographic primitive keyed with shared key material; and

at least two distinct communications paths for transporting data between the first and second host wherein at least one of the first host's DSC that it shares key material with is on the first distinct communications path and at least one of the second host's DSC that it shares key material with is on the second distinct communications path;

and in which the first host sends to the second host:

at least one first data over the first distinct communications path; and

at least one second data over the second distinct communications.

It is preferred that the system further comprises initiating means for invoking an initiating process between a first host and a second host.

It is preferred that each of the DSC on the first and second distinct communications paths are only on one of the at least two distinct communications paths.

It is preferred that:

at least one key material which is shared between a host and a DSC, is a symmetric key material that has been negotiated between that host and that DSC in a post quantum secure method, and the secure communications path between that host and that DSC is a post quantum secure path that is adapted to use a symmetrically keyed cryptographic primitive.
It is preferred that the network is an overlay network.

It is preferred that there are at least three DSCs in the overlay network, and wherein at least three of the at least three DSCs are associated with a unique set of hosts with respect to any other at least three DSCs.

It is preferred that the value of the data which is sent between the first host and the second host across one distinct communication path is always encrypted. Alternatively, it is preferred that the value of the data transmitted across each of at least two distinct communications paths is always encrypted.

It is preferred that the initiation of communications between the first host and the second host comprises a process of negotiation by the first host and the second host that results in an agreement on the selection of two or more DSCs.

It is preferred that the first host and second host exchange information identifying the DSC with which that host shares a key.

It is preferred that there are at least four DSCs, and at least two federations, where each of the at least four DSCs is assigned to one of the at least two federations and where at least two DSCs in one federation can establish secure communications with each other.

It is preferred that at least one distinct communications path between the first host and the second host comprises:

- a DSC in a federation that shares a key with the first host;
- a DSC in the same federation that shares a key with the second host;

where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish secure communications between each other.
It is preferred that:

   at least one first distinct communications path between the first host and the second host comprises:
      a DSC in a first federation that shares a key with the first host;
      a DSC in the first federation that shares a key with the second host;
       where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish secure communications between each other; and
   at least one second distinct communications path between the first host and the second host comprises:
      a DSC in a second federation that shares a key with the first host;
      a DSC in the second federation that shares a key with the second host;
       where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish secure communications between each other; and

   where the first federation and second federation are different federations.

It is preferred that:

   the data sent between the first host and the second host is a message, the message:
      comprises a component which is derived from a random number;
      is encoded under an all-or-nothing transformation; and
      is partitioned into at least two data parts where the at least two data parts are exchanged over different distinct communications paths.

It is preferred that many to many data exchange system further comprises:

   the exchange of key material from the first host to the second host to create a session key:
in which the key material which is exchanged from the first host to the second host is partitioned into at least one first data part and at least one second data part; and the session key is derived from a portion of the at least one first data part and the at least one second data part.

In another aspect preferred embodiments of the present invention provide a method of communicating data between at least two hosts in a network which comprises:

- at least two DSCs, and
- at least two distinct communications paths for transporting data between a first host and second host;

the method comprising the steps of:

- communicating at least one first data between the first host and a first DSC on a first distinct communications path, where the at least one first data is encrypted using a key which is shared between the first host and the first DSC;
- communicating at least one second data between the second host and a second DSC on a second distinct communications path, where the at least one second data is encrypted using a key which is shared between the second host and the least one second DSC; and

the first host sending to the second host:

- at least one first data over the first distinct communications path; and
- at least one second data over the second distinct communications; and

where each of the DSC on the first and second distinct communications paths are only on one of the at least two distinct communications paths.

It is preferred that the method of communicating data between at least two hosts further comprises initiating communications between the first host...
and the second host.

It is preferred that the network is an overlay network.

5

It is preferred that there are at least three DSCs in the overlay network, and wherein at least three of the at least three DSCs are associated with a unique set of hosts with respect to any other of the at least three DSCs.

It is preferred that the value of the data which is sent between the first host and the second host across one distinct communication path is always encrypted.

It is preferred that the value of the data transmitted across each of the at least two distinct communications paths is always encrypted.

10

It is preferred that the initiation of communications between the first host and the second host comprises a process of negotiation by the first host and the second host that results in an agreement on the selection of two or more DSCs.

15

It is preferred that the first host and second host exchange information identifying the DSC with which that host shares a key.

20

It is preferred that there are at least four DSCs, and at least two federations, where each of the at least four DSCs is assigned to one of the at least two federations and where at least two DSCs in one federation can establish secure communications with each other.

25

It is preferred that the at least one distinct communications path between the first host and the second host comprises:

- a DSC in a federation that shares a key with the first host;
- a DSC in the same federation that shares a key with the second host;

where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish
secure communications between each other.

It is preferred that:

at least one first distinct communications path between the first host and the second host comprises:

- a DSC in a first federation that shares a key with the first host;
- a DSC in the first federation that shares a key with the second host;

where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish secure communications between each other; and

at least one second distinct communications path between the first host and the second host comprises:

- a DSC in a second federation that shares a key with the first host;
- a DSC in the second federation that shares a key with the second host;

where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish secure communications between each other; where the first federation and second federation are different federations.

It is preferred that:

- the exchange of key material from the first host to the second host to

the data sent between the first host and the second host is a message, the message:

- comprises a component which is derived from a random number;
- is encoded under an all-or-nothing transformation; and

is partitioned into at least two data parts where the at least two data parts are exchanged over different distinct communications paths.

It is preferred that:

- the exchange of key material from the first host to the second host to
create a session key:
in which the key material which is exchanged from the first host
to the second host is partitioned into at least one first data part
and at least one second data part; and

the session key is derived from a portion of the at least one first
data part and the at least one second data part.

It should be understood that the detailed description and specific examples, while indicating preferred embodiments of the invention, are given by way of illustration only, since various changes and modifications within the spirit and scope of the disclosure herein will become apparent to those skilled in the art from this detailed description.

**BRIEF DESCRIPTION OF THE DRAWINGS**

In order that the present invention be more readily understood preferred embodiments of it are described with reference to the accompanying drawings, which are given by way of illustration only, and thus are not limitative of the disclosure herein, and in which:

Figure 1 is a schematic diagram illustrating an overlay network depicting KSC nodes and user nodes according to a preferred embodiment of the current invention;

Figure 2 is a schematic diagram illustrating a data exchange between two hosts with the support of two DSC of figure 1 according to a preferred embodiment of the current invention.

Figure 3 is a schematic diagram of a network depicting the communication paths between users and server according to a preferred embodiment of the current invention;

Figure 4 is a simplified message flow diagram depicting message flows between users and servers on network;

Figure 5 is a part illustration of a hybrid trust and network model depicting trust relationships and communication paths between users and servers on network of figures 3 and 4 according to a preferred embodiment of the current invention;

Figure 6 is a part illustration of a hybrid organizational and network model depicting communication paths between two users and a KSC on the network of figures 3 and 4 according to a preferred embodiment.
of the current invention;

Figure 7 is a schematic diagram of a relationship model depicting a federation 1101 with four member organizations in accordance with a preferred embodiment;

Figure 8 is a schematic diagram of a network model depicting an organization 1201 of figure 7;

Figure 9 is a schematic diagram of a hybrid organizational and network model depicting network users, KSCs, organizations, and federations in accordance with a preferred embodiment;

Figure 10 is a schematic diagram of a hybrid organizational and network model as in figure 9 that illustrates the communication paths between parties in accordance with a preferred embodiment;

Figure 11 is a part illustration of a hybrid trust and network model depicting communication paths between two users and at least one KSC on the network of figure 10 according to a preferred embodiment;

Figure 12 is a schematic diagram of a hybrid organizational and network model as in figure 9 that illustrates communication paths between parties in accordance with a preferred embodiment;

Figure 13 illustrates a preferred embodiment of exchanging keys between at least one KSC within a federation;

Figure 14 illustrates a communication system incorporating hardware in accordance with a preferred embodiment;

Figure 15 is a block schematic diagram of example general-purpose computers used in the communication system of figure 14;

Figure 16 is a block schematic diagram of exemplary USB tokens used in the communication system of figure 14;

Figure 17 is a block schematic diagram of a general-purpose server used in the communication system of figure 14;

Figure 18 illustrates an exemplary structure according to a preferred embodiment of aggregated key material used in a key exchange;

Figure 19 illustrates the structure of a first key material, a second key material and a third key material of the aggregated key material of figure 18;

Figure 20 is a flow chart of an authentication process used within a
preferred embodiment of the present invention;
Figure 21 is a flowchart of a further authentication process used within a preferred embodiment of the present invention;
Figure 22 is a flowchart illustrating another authentication process of a preferred embodiment
Figures 23, 24 and 25 are flow-charts illustrating key exchange processes conducted between negotiating parties for secure communication according preferred embodiments;
Figure 26 illustrates an element of a set (not illustrated) stored by each of the negotiating parties, and the fields in that element in accordance with a preferred embodiment;
Figure 27 illustrates two instantiated sets of elements of type 3001 of figure 26 for three parties;
Figure 28 illustrates parts of a message passed between negotiating parties, and the fields in that message in accordance with a preferred embodiment;
Figure 29 illustrates an instantiated set of two or more KSC selections achieved by each of the negotiating parties in accordance with a preferred embodiment;
Figure 30 is a flow chart of a process for multi-path negotiation used within a preferred embodiment of the present invention.

DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

Figure 1 is a schematic diagram 600 illustrating an overlay network 605 depicting DSC nodes and host nodes according to a preferred embodiment of the current invention. The overlay network 605 typically has at least three DSC nodes. Labels 601, 602, 603 and 604 illustrate 4 DSC nodes within an overlay network 605. Each DSC can have an arbitrary large number of host nodes.

Labels 610, 611, 612 and 613 illustrate 4 host nodes in group 614 that each share at least one key with DSC 601 and DSC 604. Labels 620, 621, 622 and 623 illustrate 4 host nodes in group 624 that each share at least one key with DSC 602 and DSC 603. Labels 630, 631 illustrate 2 host nodes in group 632 that each share at least one key with DSC 601 and DSC 603. Labels 640, 641 illustrate 2 host nodes in group 642 that each share at least
one key with DSC 604 and 602. In preferred embodiments of the present invention at least one key shared between a host and a DSC is a secret symmetric key. In a further preferred embodiments the secret symmetric key shared between the host node and the DSC node is negotiated between the host and the DSC. In a further preferred embodiment the negotiated secret symmetric key is derived from the output of a true random number generator.

Group 614 and 632 share one common DSC 601, and two different DSC 604 and 603. Group 614 and 642 share one common DSC 604, and two different DSC 601 and 602. Group 632 and 624 share one common DSC 603, and two different DSC 601 and 602. Group 624 and 642 share one common DSC 602, and two different DSC 603 and 604. Group 614 and 624 share no common DSC. Group 632 and 642 share no common DSC.

Figure 1 illustrates the overlay network 605 has three or more DSC, where each host node is registered with at least two DSC within 605, wherein at least three of the three or more DSC manage a distinct set of host nodes.

Figure 2 is a schematic diagram illustrating a data exchange between two hosts 610, 621 with the support of two DSC 601, 603 of figure 1 according to a preferred embodiment of the current invention. Figure 2 also illustrates two additional DSC 602, 604.

A data exchange performed on the overlay network 600 involves a first host 610, at least one DSC 601, 604 that shares key material with the first host 610, a second host 621, at least one DSC 602, 603 that shares key material with the second host 621, where there is a cryptographically secure path between the first host 610 and each of its participating DSC 601 that it shares key material with, and where there is a cryptographically secure path between the second host 621 and each of its participating DSC 602 that it shares key material with, and where there is at least two distinct communications paths 201, 202 for transporting data between the first host 610 and the second host 621 wherein at least one of the first host's DSC 601 is on the first distinct communications path 201 and at least one of the second host's DSC 603 is on the second distinct communications path 202.

An implementation of a preferred embodiment is where a data exchange performed on the overlay network 600 involves a first host 610, at least one DSC 601, 604 that shares key material with the first host 610, a
second host 621, at least one DSC 602, 603 that shares key material with the
second host 621, where there is a cryptographically secure path between the
first host 610 and each of its participating DSC 601 that it shares key material
with, and where there is a cryptographically secure path between the second
host 621 and each of its participating DSC 603 that it shares key material
with, and where there is at least two distinct communications paths 201, 202
for transporting data between the first host 610 and the second host 621
wherein at least one of the first host's DSC 601 is on the first distinct
communications path 201 and at least one of the second host's DSC 603 is
on the second distinct communications path 202.

In a further preferred embodiment the secure communications path
between at least one host 610 and at least one of its DSC 601 is a candidate
PQS communications path.

In a further preferred embodiment in which the value of the data
transmitted which is sent between the first host 610 and the second host 621
across one distinct communications path 201 is always encrypted so that
parties other than 610, 601, 621 monitoring the links \{610, 601\} and \{6102,
621\} cannot obtain the value of the data. In a further preferred embodiment in
which the value of the data transmitted across each of the at least two distinct
communications paths 201, 202 is always encrypted.

In a further preferred embodiment all communications paths \{601, 610\},
\{601, 621\}, \{603, 610\}, \{603, 621\} involving DSC 601, 603 in a data exchange
are secured use cryptographic primitives that are candidate PQS.

Figure 3 is a schematic diagram 700 of a network depicting the
communication paths between host nodes and KSC nodes according to a
preferred embodiment of the current invention.

Label 201, 202, 203, 204 are Internet service providers (ISP). Each of
the four ISPs are connected with the remaining three ISPs with optical fiber
connections as illustrated by the thin black lines. Label 210, 211, 212, 213,
214, 215, 216, 217 are nodes participating in the network. Each of the eight
nodes are connected to one ISP over a single communications link. Nodes
210 and 217 are connected to ISP 201. Nodes 211 and 212 are connected to
ISP 202. Nodes 213 and 214 are connected to ISP 203. Nodes 215 and 216
are connected to ISP 204. Node 215 is illustrated as being a negative agent
with the thick black border whereas users with thin black borders such as 212 are not. Link (217, 201) is illustrated as being monitored by an attacker 215 (compromised) with a solid square black box, where as links such as (211, 202) are not.

Label 217 is a host referred to as the first party. Label 213 is a host referred to as the second party. Label 215 is a KSC trusted by the first and second party. Labels 210 and 211 are 2 KSC trusted by the first party. Label 214 is a KSC trusted by the second party. The communications paths (211, 217), (213, 215) and (215, 217) are secure against classical and quantum adversaries. The communication paths (211, 213), (213, 217), (214, 217) are not encrypted.

Figure 4 is a simplified message flow diagram 800 depicting message flows between hosts and KSCs on network 700 of figure 3 across paths previously described.

First party 217 and second party 213 negotiate which KSC they wish to participate in the key exchange by exchanging messages over unencrypted but authenticated message flow 801. Authentication is achieved using a candidate PQS digital signature algorithm. First party 217 selects three KSC 215, 210, 211. Second party 213 selects two KSC 215 and 214. After negotiation, the two parties have agreed on selecting three KSC 215, 211 and 214 as participating KSC, each participating KSC will act as a key transfer centre.

In this way we have described how the initiation of communications between the first host 217 and the second host 213 comprises a process of mutual negotiation by the first host 217 and the second host 213 that results in an agreement on the selection of two or more participating DSC 211, 214, 215.

First party 217 generates and transmits key material (217, 213) to the second party 213 in unencrypted message flow 802. Second party 213 generates and transmits key material (213, 217) to the first party 217 in unencrypted message flow 803.

First party 217 securely authenticates itself with KSC 215 in message flow 804. Second party 213 securely authenticates itself with KSC 215 in message flow 805. First party 217 generates and transmits key material (217,
215, 213) across encrypted message flow 806 to KSC 215 which relays the
key material across encrypted message flow 807 and is received by the
second party 213. Second party 213 generates and transmits key material
{213, 215, 217} across encrypted message flow 808 to KSC 215 which relays
the key material across encrypted message flow 809 and is received by the
first party 217.

First party 217 securely authenticates itself with KSC 211 over
message flow 811.

First party 217 generates and transmits key material {217, 211, 213}
across encrypted message flow 812 to KSC 211 which establishes an
unencrypted connection with second party 213 and relays the key material
across unencrypted message flow 813 and is received by the second party
213. Second party 213 generates and transmits key material {213, 211, 217}
across unencrypted message flow 814 to KSC 211 which relays the key
material over encrypted link 815.

Second party 213 securely authenticates itself with KSC 214 over
message flow 817 to first party 217.

Second party 213 generates and transmits key material {213, 214, 217}
across encrypted message flow 818 to KSC 214 which establishes an
unencrypted connection with first party 217 and relays the key material across
unencrypted message flow 819 and is received by the first party 217. First
party 217 generates and transmits key material {217, 214, 213} across
unencrypted message flow 820 to KSC 214 which relays the key material over
encrypted link 821 to second party 213.

At the end of this protocol the following 8 parts of key material has
been exchanged, {217, 213}, {213, 217}, {217, 215, 213}, {213, 215, 217},
{217, 211, 213}, {213, 211, 217}, {217, 214, 213}, {213, 214, 217}. In a
preferred embodiment of the current invention, the eight parts are combined
using a candidate PQS collision resistant hash, such as SHA-512, to create
the session key between first party 217 and second party 213. Both the first
217 and second 213 parties can request additional key material to be exchanged over different paths, or with additional KSC as desired.

In the present embodiment of the current invention, each host party
217, 213 generates half of the key material over each distinct path to ensure
freshness and to protect, for example, against a weak pseudo-random number generator in either parties system. In a preferred embodiment of the current invention the entropy supplied by each host 217, 213 across each distinct communications path should be sufficient to ensure the session key is cryptographically secure.

In a preferred embodiment of description figure 4, each of the participating KSC 211, 214, 215 are further adapted to attest to the identity of each sender 217, 213 by sending an identifier value along with the key material it is relaying. In this way, the key transfer centre is enhanced to perform a function commonly associated with a KDC.

Figure 5 is a partially illustrated hybrid trust and network model 900 depicting trust relationships and communication paths between host nodes and KSC nodes on network 700 of figures 3 and 4 according to a preferred embodiment of the current invention. Label 910 illustrates five KSC 210, 211, 920, 921, and 922 in the first ring of trust for first party 217. Label 911 illustrates one KSC 215 in the second ring of trust for the first party 217. Label 930 illustrates six KSC 214, 941, 942, 943, 944 and 945 in the second party's 213 first ring of trust. Label 931 illustrates one KSC 215 in the second ring of trust for the second party 213.

A extranet secure communications path has been established between user 217 and 213 through common trusted KSC 215. That is, only the first party 217, the second party 213 and the trusted KSC 215 are authorized to know the unencrypted value of the key material sent between the first 217 and second 213 party across that path {213, 215, 217}.

In this illustration, first party 217 trusts the KSC 210, 211, 920, 921, 922, 923 in the first ring of trust 910 not to collude with KSC 215 and second party 213 trusts the KSC 214, 941, 942, 943, 944, 945, in its first ring of trust 930 not to collude with KSC 215. The first party 217 and second party 213 have not made an explicit opinion as to the likelihood that the others party's selected KSC 214 and 211 will collude with KSC 215. However, the first 217 and second 213 party each anticipate the other party is likely to choose a KSC that may act honestly in respect to that other party and thus not collude with KSC 215.

As each user has several KSC in their respective first rings of trust it is
possible that one of the possible combinations of trusted KSC to KSC paths might not be monitored by one or more colluding attackers.

The communication paths \{213, 217\}, \{214, 217\}, \{211, 213\} are monitored by KSC 215. However, the communication path between the KSC 211 and KSC 214 is an unencrypted communications path that in this example is not monitored by KSC 215. This implies that KSC 215 cannot gain access to all key material exchanged between the first party 217 and the second party 213 and therefore cannot later decrypt data encrypted with the session key generated by the first and second party.

The key exchange protocols described above do not rely on any shared secrets between the two parties 217 and 213. In the case where a session key has been previously exchanged between 217 and 213 using embodiments of the current invention, the result of a fresh key exchange may be combined with the value of the previous session key.

In this respect, consider the scenario where a user A employs the use of a personal smart card token to perform all key exchange operations according to a preferred embodiment of figures 3, 4 and 5. If the user A uses the smart card token to perform key exchanges at work, at home and when travelling overseas then this increases the probability that at least one session key was established without being monitored. If a single key exchange is successful, at one level of reasoning, every other key exchange that is derived from this key material is also successful. In a preferred embodiment when both parties agree to use previously exchanged key material, all new key material transmitted to each party is encrypted using the previously exchanged key material. This ensures that even if a link is ‘monitored’ that no useable information is leaked if they do not have access to the currently value of the previously exchange key material.

In a preferred embodiment where the key exchange is performed by a smart card token, the smart card token stores the successful session keys in a remotely hosted database which is encrypted using a key known only to the smart card. This allows a smart card to store a large number of session keys, and associated information, that could not otherwise be stored in the smart card non-volatile memory. In a preferred embodiment, after an initial symmetric secret key has been exchange it is possible to perform
authenticated key exchanges directly using the secret key without using DSC. In a preferred embodiment long-lived network sessions perform additional key exchanges with KSC not previously used to amplify the security of the current session key.

In a preferred embodiment host 217 belongs to a large global corporation that maintains KSC server 210 and 211 which is operating on a different ISP to host 217. This type of approach allows a large global corporation to run many such KSC, each KSC located at a different ISP in a different country and registering each host with each KSC.

In a preferred embodiment, a large global corporation automates the process of having each company employee's smart card trust every other company employee's smart card as a KSC as illustrated in 910. A user performing a key exchange under one ISP can randomly select another company employee located at a different ISP that is currently online and who has previously exchanged key material with to act as a trusted KTC on its behalf. This property is similar to peer-to-peer networks, where users perform services for other users on the overlay network.

Figure 6 is a partially illustrated hybrid organizational and network model 1000 depicting communication paths between host nodes 213 and 217 and KSC node 215 on network 700 of figures 3 and 4 according to a preferred embodiment of the current invention. In a preferred embodiment of the present invention KSC 215 presents itself as a single logical node that is implemented using a federated system of four KSC 1001, 1002, 1003 and 1004, also known herein as sub KSC nodes. In a preferred embodiment the four sub KSC nodes 1001 through 1004 are all located roughly in the same geographical location and have low communication latencies between each other on the order of 1 ms to less than approximately 50 ms. Each of the sub KSC 1001, 1002, 1003, 1004 nodes share large symmetric pair-wise keys, where large is on the order of 256 to 1024 bits long. Secure communications between each of the sub KSC nodes is achieved using key material derived from those symmetric keys. Host 213 shares a large symmetric secret with sub KSC 1004 of federated KSC 215. Host 217 shares a large symmetric secret with sub KSC 1002. Communications between host 217 is securely relayed by 1002 and 1004 to host 213. This simple type of federation allows
the number of hosts supported by a single logical KSC node to be increased. It also potentially improves reliability against partial system failure. A similar type of recursive subdivision of function can also occur within each sub KSC node.

Figures 7 through to 13 illustrate further preferred embodiments of the overlay network 605 of figure 1 such that:

- there are at least two distinct extranet secure communications paths. This property helps mitigate against a large-scale collusion attack of the type described in figure 5.

- the use of extensive compartmentalization which improves the overall scalability, reliability and the users ability to reason concerning possible collusion.

Figure 7 is a schematic diagram of a relationship model depicting a federation 1101 with four member organizations 1105, 1106, 1107, 1108, and relationships between the organizations and the federation. In this illustration, a federation is a distinctly managed body, which means that the humans involved in the management of the operations of one federation ideally should not be involved in the management of the operations of any other federation at the same level. A federation has one or more member organizations.

Point to point communications between members 1105, 1106, 1107 and 1108 is secured using authenticated encryption operations using large symmetric keys preferably on the order of 256 to 1024 bits. In a preferred embodiment, the key management for communications between organizations are negotiated using techniques described in the Australian provisional patent application 2008903827 in the name of Synaptic Laboratories Limited, filed on 28 July 2008 entitled "Method and Apparatus for Secure Communication".

In another preferred embodiment, the key management for communications between organizations includes the use of a distributed threshold secret sharing scheme.

In a further preferred embodiment, a global data or key exchange system may have at least three global federations and one or more private federations. It is envisioned that there will be a relatively small number of global federations to ensure interoperability between potentially billions of users and potentially millions or more privately managed federations to
provide higher assurance of security for their respective members.

Figure 8 is a schematic diagram of a network model depicting an organization 1201 which could be any one of organizations 1105, 1106, 1107, 1108 of figure 7. In this illustration organization 1201 has four DSCs 1205, 1206, 1207, 1208. It is preferred that an organization is distinctly managed, which means that the humans involved in managing the operations of one organization ideally should not be involved in managing any other organizations or other federations at the same level. The concept of federation levels is more fully described with assistance of figure 9. It is strongly preferred that each organization participates in one and only one federation at any level, e.g., global, national, regional, industry group, private. Each organization has at least one DSC. Each DSC may be a cluster of two or more computers as depicted in figure 6. Each DSC within an organization may be located in a geographically different location and distinctly managed by a distinct group of humans. In the present embodiment, each DSC is responsible for managing a potentially different set of parties/users/hosts. In the present embodiment, each host can establish a secure connection with each of the DSC it shares secrets with. In a preferred embodiment of the current invention, each DSC shares a large symmetric key with every other DSC within its organization and this key is used for secure communications between DSC.

In a preferred embodiment, the key management for communications between DSC within an organization, or between at least two different organizations, are negotiated using techniques described in the above referenced Australian provisional patent application 2008903827.

In a preferred embodiment, the key management for communications between DSC within an organization includes the use of a distributed threshold secret sharing scheme.

A DSC can be adapted to operate as a key transfer centre that does not attest to the identity of a user/host. This reduced level of responsibility or authority on the part of the DSC allows a user/host to share symmetric key secrets with the DSC even if the user/host does not trust them to behave honestly with respect to identity operations. There are known security risks associated with a single trusted third party attesting to the identity of a party.
A preferred method of adapting two DSC to attest as to the identity of a user/host is described in the associated Australian provisional patent application 2008904612 in the name of Synaptic Laboratories Limited, filed on 5 September 2008, entitled "Threshold Proxy Operations".

In a preferred embodiment, a DSC may reassign one or more of the hosts it manages, and the key material for them, to another DSC within the same organization. This may serve two purposes, (a) load sharing work across all available DSC, (b) protecting against attacks that correlate an identity with which DSC they regularly log in to.

A DSC can be securely deployed on any network topology and on any node in the network topology.

Figure 9 is a schematic diagram 1300 of a hybrid organizational and network model depicting network host nodes, DSCs, organizations, and federations. Figure 9 illustrates a highly regular structure to simplify description of embodiments of the invention. In this illustration there are four federations 1301, 1302, 1303, 1304 with each of the four federations having 4 organizations, each of the 4 organizations having four DSCs resulting in a total of 16 DSC in each federation. Each of the 4 federations are participating as part of one large overlay network 1300. The 64 DSCs are all connected to a common communications network (not illustrated) such as the Internet. Each of the 64 DSC may be located in a physically different building. Each DSC shares, or can establish, a large symmetric key with every other DSC within its federation. This may be established by each of the DSC establishing a secure connection using the symmetric key shared at the organizational level to exchange key material. A preferred embodiment of exchanging keys between DSC within a federation is described in figure 13. There are 3 host nodes called parties illustrated as 1305, 1306 and 1307 that are adapted to perform data exchanges or key exchanges on the overlay network 1300. In this example, the federation 1301 is a privately managed federation where as federations 1302, 1303, and 1304 are public federations operating on a global level. In a preferred embodiment, there are federations that operate on a national level. Conceptually an organization may participate within a national federation level and on a global federation level, but never twice in the same level of federation.
Figure 10 is a schematic diagram of a hybrid organizational and network model as in figure 9 that illustrates the communication paths between host nodes {1305, 1306} and {1305, 1307}. We will refer to the host nodes 1305, 1306, 1307 as parties. Parties {1305, 1306} and {1305, 1307} have each established an initial unencrypted network session 1410 and 1411, respectively. In the presently described embodiment the five DSCs 1401, 1402, 1403, 1404, 1405 are adapted to relay data and key material. When relaying key material they are operating as key transfer centres.

Party 1305 has prior established a large symmetric secret key with each of 1401, 1402, 1403 and can establish secure authenticated communications paths with each of 1401, 1402, 1403. Party 1306 has prior established a large symmetric secret key with each of 1404, 1405 and can establish secure authenticated paths with each of 1404 and 1405. Party 1307 has prior established a large symmetric secret key with each of 1403, 1404 and can establish secure authenticated communications paths with each of 1403 and 1404. DSC 1401 and 1405 are within the same federation and can establish secure authenticated communications paths between themselves as previously described. DSC 1402 and 1404 are within the same federation and can also establish secure authenticated communications paths between themselves as previously described.

Figure 11 is a partially illustrated hybrid trust and network model 1500 depicting communication paths between two parties and DSC 1305, 1306 on network 1400 of figure 10 according to a preferred embodiment of the current invention. Label 1510 illustrates the first ring of trust for the party 1305 which includes DSC 1403 of federation 1304, DSC 1401 of federation 1301. Label 1511 illustrates the second ring of trust for the party 1305 which includes DSC 1402 of federation 1303 and DSC 1405 of federation 1301. Label 1520 illustrates the first ring of trust for the party 1306 which includes DSC 1404 of federation 1303. Label 1521 illustrates the second ring of trust for the party 1306 which includes DSC 1402 of federation 1303 and DSC 1401 and 1405 of federation 1301.

A first distinct extranet secure communications path has been established securely between party 1305 and 1306 through federation 1303 including DSC 1402 and 1404. A second distinct extranet secure
communications path has been established between party 1305 and 1306 through federation 1301 including DSC 1401 and 1405. In this illustration party 1306 does not trust, or does not have a relationship with federation 1304, and in this illustration federation 1304 is not participating in the illustrated key exchange operation. Party 1306 has a high level of trust in federation 1303, where as party 1305 has a high level of trust in federation 1301. Each party trusts that the federations in their inner ring of trust will not collude with federations in their outer rings of trust.

It is possible that a human user does not have a high level of trust in any of the DSC in the four federations 1301, 1302, 1303 and 1304, however the human user may trust that certain combinations of federations, or certain combination of DSC from different federations, are unlikely to all collude together and that this is sufficient to ensure a secure key exchange.

The ability to group organizations of similar alliances within a single federation, and the ability for a human user to establish their own individual reasoning concerning the likely chances of collusion between federations provides a new level of empowerment to human users not present in the related art.

The use of associative rules, a technique well understood in the related art, can be used to build a set of rules that must be followed when performing a key exchange. A human user may specify the rules using a human readable language for the purpose, or with a graphical user interface. The rules of each host node may be different, reflecting the unique trust relationships of the human user and its host nodes. To execute some of the rules it may be required for each host to authenticate itself in some capacity as part of the negotiation process when selecting DSC to participate in a key exchange. An example of a key negotiation process is illustrated in figures 26 through 30.

When a group of human users wish to establish the most secure communications paths between each other, the group can agree to select from a set of mandatory organizations and federations that each host node must join so as to enable a secure data or key exchange. Furthermore, each member of the group can manage their own private federation, which every host node must be registered with. This allows each member of the group to
ensure the key exchange is secure against external adversaries, and simultaneously against internal corruption. This can be achieved if there are three or more DSC, and where each user can choose which DSC they wish to participate in the key exchange. Choice may also be expressed through negotiation of a secret with a DSC, by electing that a DSC not participate, or other related methods.

Figure 12 is a schematic diagram of a hybrid organizational and network model as in figure 9 that illustrates communication paths between parties \{1610, 1611\}. Parties \{1610, 1611\} have established an initial network session 1620. In the presently described embodiment the two DSCs 1601, 1602 are operating as KTC and the two DSC 1603 and 1604 are each operating as KDC. Party 1610 has prior established a large symmetric secret with each of 1601, 1604 and can establish secure authenticated communications paths with 1601 and 1604. Party 1611 has prior established a large symmetric secret with 1602 and 1603 and can establish secure authenticated communications paths with 1602 and 1603. KSC 1601 and 1602 are within the same federation 1303 and can establish secure authenticated communications paths between themselves as previously described. DSC 1603 and 1604 are within the same federation 1301 and can establish secure authenticated communications paths between themselves as previously described.

In the embodiment of figure 12 KDC 1603 and KDC 1604 are running a federated variation of a candidate PQS implementation of the Needham-Schroeder shared-key protocol, described in [009], ensuring Tag-Length-Value encoding, as described in [010], for all fields used in the protocol.

Party 1611 has a secure communications path with KDC 1603. KDC 1603 also has a secure communications path with KDC 1604. KDC 1603 and 1604 authenticate party 1611 with party 1610. The session key generated by KDC 1604 for parties 1610 and 1611 is combined with the other key material exchanged between party 1610 and 1611 to create the session key used to secure communications over 1620.

In this way figure 12 illustrates that a KSC adapted to operate as a KDC and a KSC adapted to operate as a KTC can be securely combined to create a PQS key exchange.
In an alternate description of figure 12, KSC 1603 and 1604 are both KTC. The logical path 1621 \{1604, 1610\} is achieved by routing packets through the nodes \{1610, 1611, 1603, 1604\}. This illustrates that a user 1610 does not necessarily need to establish direct communications links with each of its DSC that it shares secrets with. This property may result in faster key exchanges and a simplified implementation for computationally constrained parties wanting to exchange keys with a computationally more powerful party.

Figure 13 is a schematic diagram of a hybrid organizational and network model as in figure 9 that illustrates two smart cards 1702 and 1712 and communication paths between two DSC \{1701, 1711\} performing a key exchange. Figure 13 illustrates how two DSC, irrespective of their respective memberships within an organization or within a federation, can establish pairwise keys. There are three federations 1002, 1003, 1004.

A first DSC 1701 is illustrated as belonging to a different organization than the second DSC 1711 within the same federation 1002. The first DSC 1701 has a smart card 1702 that is a host node. The second DSC 1711 has a smart card 1712 that is a host node. DSC 170 and 1715 belong to federation 1002. DSC 1706 and 1716 belong to federation 1003. DSC 1707 and 1717 belong to federation 1004. The smart card 1702 has exchanged secrets with DSC 1705 that belongs to a different organization than 1701 within the same federation, with DSC 1706 and DSC 1707 which both belong to two other federations. The smart card 1712 has exchanged secrets with DSC 1715, 1716 and 1717 that each belong to two other organizations and in the latter two cases other federations. Key exchanges between DSC 1711 and 1701 are performed using the smart card as previously described. In this way, we have illustrated how any DSC can use other DSC belonging to different organizations and federations to exchange keys with any other DSC. Also, in a similar fashion, any host can register with any other DSC that it has not previously established secrets with. In a preferred embodiment, enrolment of a new KSC with a host triggers a process of key reinforcement by the host, that process uses the new KSC to exchange additional key material between the host and each of its existing DSC.

More detailed description follows with reference to figures 14 to 30.
Figure 14 illustrates a communication system 1800 incorporating a preferred embodiment of the present invention. The embodiment of figure 14 comprises a general-purpose computer 1801, which can interface with a secure module or device, which is preferably a USB smart card token 1802, a general-purpose computer 1803, which can interface with another USB smart card token 1804, a general-purpose computer 1806, which can interface with another USB smart card token 1807, a general-purpose server 1808, which can interface with a database 1809. The general-purpose computers 1801, 1803, 1806 and general-purpose server 1808 can communicate with each other over public network 1811.

Figure 15 is a block schematic diagram 1900 of the general-purpose computers 1801, 1803 and 1806 of figure 14. In figure 15, the general-purpose computers 1801, 1803 and 1806 comprise a central processing unit 1901, random access memory 1902, an input / output system 1903 and a video card 1904.

Figure 16 is a block schematic diagram 2000 of the USB tokens 1802, 1804 and 1807 of figure 14. The USB tokens 1802, 1804 and 1807 comprise a central processing unit 2001, random access memory 2002, a communication interface such as a universal bus (USB) 2003, a dedicated cryptographic processor 2004, and non volatile memory 2005.

Figure 17 is a block schematic diagram 2100 of the general-purpose server 1808 of figure 14. The general-purpose server 1808 comprises a central processing unit 2101 of figure 17, random access memory 2102, an input / output system 2103, and a dedicated cryptographic processor 2104.

Figures 18 and 19 illustrate portions of messages passed among the components of figure 14, and fields in those messages.

Figure 18 illustrates 2200 the structure of aggregated key material 2201 shared between two hosts.

The aggregated key material 2201 contains six fields 2211, 2212, 2213, 2214, 2215 and 2216 which are each 256-bits in length.

Figure 19 illustrates 2300 the structure of a first key material 2301, a second key material 2302 and a third key material 2303. In the presently-described embodiment of figure 19 the first key material 2301 contains two fields 2211 and 2212 from figure 18, the second key material 2302 contains...
two fields 2213 and 2214 from figure 18 and the third key material 2303 contains two fields 2215 and 2216 from figure 18. The aggregated key material of 2201 shown in figure 18 comprises 6 units of random numbers functioning as the key material. The structure of figure 19 shows the aggregate key material 2201 partitioned into 3 different key materials 2301, 2302, 2304 corresponding to a possible 3 distinct paths between the two hosts, each of the three different key materials further partitioned into 2 portions corresponding to a portion generated by one of the two hosts and sent to the other host over that distinct path. In this illustration key material is exchanged by each host generating three portions of key material, and sending a copy of each of the portions to the other host over a different path, the total of six portions of key material exchanged are then aggregated by both hosts into key material 2201. This key material 2201 may be combined with further key material. In this illustration key material 2201 is then hashed to generate the shared session key between the two hosts. In this way, multipath key exchange and multipath key reinforcement may be facilitated.

In an alternate embodiment each host can generate a first key material by encoding both key and other data using an "all or nothing" transformation, see [008], the output is then split into two or more parts, and each part is transmitted to the other host over a different path. Data derived from these parts can then be used to generate the shared session key between the two hosts.

Figure 20 is a flow chart 2400 of a process used within a preferred embodiment of the present invention. The left-hand side 2410 of the flow chart 2400 illustrates the client process for a cryptographic challenge-response protocol. The right-hand side 2420 of the flow chart 2400 illustrates the server process. Process 2410 and process 2420 may run concurrently. The arrows 2403, 2404 and 2405 illustrate message flows between processes 2410 and 2420.

The cryptographic challenge response protocol performs an authentication operation that is anonymous in respect to third parties monitoring the communications 2403, 2404 and 2405. The authentication operation also establishes a new session key between the client and server process.
The context to the flow chart 2400 is as follows:

- The processes 2410 and 2420 have established a bidirectional communications session.

The process 2400 begins in steps 2410 and 2420. In step 2411 a first 256-bit random number is generated. In step 2421 a second 256-bit random number is generated. In step 2412 a copy of the first random number generated is transmitted 2403 and successfully received in step 2422 of process 2420. In step 2423 a copy of the second random number generated is transmitted 2404 and successfully received in step 2413.

The first and second random numbers are referred to as the challenge. The client process 2410 hashes the challenge in step 2414, the output of this hash is referred to as the solution to the challenge. The hash in step 2414 is keyed with a unique identifier which identifies the client process. This identifier is a shared secret between the client and server process.

In step 2415 a copy of the client's response to the challenge calculated in step 2414 is transmitted in 2405 and successfully received in step 2424.

The server process now needs to establish if the challenge is generated by a client managed by the server. In step 2425 the server prepares a list of all the unique identifiers of each of the clients known by the server. The first entry in the list is selected.

In step 2426 if there are no more entries in the list to process, the step 2427 returns a status value indicating the server process 2420 could not identify the author of the response generated in step 2414.

In step 2428 the server process 2402 performs a keyed hash of the challenge, using the current unique identifier of the client in the list prepared in step 2425.

In step 2429 if the output of the hash performed in step 2428 matches the response received in step 2424 the process continues at step 2430, otherwise the process selects the next entry in the list and the process continues at step 2426.

At the start of step 2430 the server process 2420 has identified and authenticated the client.

In step 2416 and 2430 a session key is derived by supplying the two random numbers generated in 2411 and 2421 and the shared secret identifier.
as a seed to a secure pseudo random number generator which generates output.

The process stops at step 2417 and 2431.

The process 2400 describes an anonymous identification operation with a fixed secret key identifier that deterministically identifies client 2410 with server 2420. In a preferred embodiment a further challenge-response process using a rolling key associated with the first identity can be performed to mitigate against device cloning attacks that reverse engineer the fixed secret key identifier from a device and use the same identifier in several devices.

For another anonymous authentication method that can be adapted for use in the present invention see Australian provisional patent application 2008904694 in the name of Synaptic Laboratories Limited, filed on 10 September 2008, entitled "Cryptographic system, process and apparatus".

Figure 21 is a flowchart 2500 of a further authentication process used within a preferred embodiment of the present invention. The left-hand side 2510 of the flow chart 2500 illustrates the client process for a cryptographic authentication process. The right-hand 2520 of the flow chart 2500 illustrate the server process. The process 2510 and process 2520 may run concurrently. The arrows 2503, 2504 and 2505 illustrate message flows between processes 2510 and 2520.

The cryptographic challenge response process performs an authentication operation that is not anonymous in respect to third parties monitoring the communications 2503, 2504 and 2505. The authentication operation does not establish a new session key.

The context to the flow chart 2500 is as follows:

• The processes 2510 and 2520 have established a bidirectional communications session.

The process 2500 begins in step 2510 and 2520. In step 2511 a first 256-bit random number is generated. In step 2521 a second 256-bit random number is generated. In step 2512 a copy of the first random number generated is transmitted 2503 and successfully received in step 2522. In step 2523 a copy of the second random number generated is transmitted 2504 and successfully received in step 2513.
In step 2514 associated data such as Internet address and port number or a public unique identifier are selected and used to key the following hashing operation.

The value of the first and second random numbers and the value of the associated data will now be referred to as the message. The client process 2510 hashes the message in step 2515, the output of this hash is then digitally signed using a candidate PQS digital signature. In the presently-described embodiment the candidate PQS digital signature chosen is the Coronado Merkle Signature Scheme (CMSS), see [007], implemented with the NIST SHA-512 hash function.

In step 2516 a copy of the message and digital signature is transmitted 2505 and successfully received in step 2524.

In step 2525 the digital signature is verified against the message in the usual way according to CMSS specifications. The result of the verification is returned as the result in step 2526. If the verification of the message is successful, the message is also returned in the result of step 2526.

The process stops at step 2517 and 2526.

In a further preferred embodiment, an identity has a digital certificate associated with it. See above mentioned Australian provisional patent application 2008904612.

Figure 22 is a flowchart 2600 illustrating another authentication process of a preferred embodiment. Flowchart 2600 illustrates a successful token authentication process. In the present embodiment each party 1305, 1306, 1307 of figure 10 has a token, each token is a host node and stores prior established secrets with two or more DSC in which at least two DSC are participating in a different organization and federation.

The right-hand 2610 of the flow chart 2600 illustrates the token process. The middle 2620 of the flow chart 2600 illustrates the card acceptance device (CAD) process for token process 2610. The left-hand side 2630 of the flow chart 2600 illustrates the server process. The process 2610, 2620 and 2630 may run concurrently. The arrows 2601, 2602 and 2604 illustrate message flows between processes 2610 and 2620. The arrows 2603 and 2605 illustrate message flows between process 2620 and 2630.
The cryptographic data transmitted during the token authentication operation is anonymous in respect to third parties monitoring the communications 2601, 2602, 2603, 2604 and 2605. The token authentication operation establishes a new session key between the token process 2610 and the server process 2630.

The process starts in step 2610, 2620, 2630.

In step 2621 the CAD process is waiting for the insertion of a USB token. In step 2611 a USB token is inserted. A message is supplied to the CAD process 2620 over message flow 2601 indicating that the USB token was inserted. In step 2612 and 2622 the CAD process interrogates the USB token over bi-directional message flow 2602 and selects the applet executing process 2610.

In step 2623 the CAD process 2620 establishes a bidirectional communications channel 2603 with server process 2630 as instructed by the applet.

In step 2613 the token process 2610 executes process 2401 of figure 20. In step 2632 the server process 2630 executes process 2402 of figure 21. In step 2624 the CAD process relays the messages between process 2601 and 2603.

After this process of figure 21 completes token process 2610 is successfully identified and authenticated with the server 2630 managing it. A secure channel has been established between the token process 2610 and the server 2630.

In step 2633 the server proceeds to register the online status of the token in its database.

The process stops in steps 2614, 2625, 2634.

Figure 23 is a flowchart 2700 illustrating a process used during a key exchange process of a preferred embodiment. Flowchart 2700 illustrates a first token process 2710 discovering a second token 2750 and subsequent authentication operation between the two tokens. The first and second tokens are both hosts that have not previously performed a key exchange between each other. The token discovery and authentication operation does not negotiate a new session key between the first token process 2710 and the second token process 2750.
Process 2720 is the CAD process for the first token process 2710. Process 2730 is a server process. Process 2730 is also known as a DSC operating as a KTC. Process 2740 is a CAD process for the second token process 2750.

The processes 2710, 2720, 2730, 2740 and 2750 may run concurrently. The arrows 2701, 2702 and 2709 illustrate message flows between processes 2710 and 2720. The arrows 2703 and 2708 illustrate message flows between process 2720 and 2730. The arrows 2704 and 2707 illustrate message flows between process 2730 and 2740. The arrows 2705 and 2706 illustrate message flows between process 2740 and 2750.

The token discovery and authentication operations are anonymous in respect to third parties monitoring the communications 2701 through 2709.

The context to the flow chart 2700 in the current embodiment is as follows:

- The first token process 2710 and second token 2750 have authenticated themselves with the server process 2730 as described in figure 22.

  In step 2721 the first CAD process 2720 forwards a request to the first token process 2710 requesting a key exchange with a token by a public identifier associated with the second token 2750. The request is successfully received in step 2711 by the first token process 2710.

  In step 2712 the first token process sends a request to the server 2731 to establish a secure communications relay path with the second token 2750. Specifically, this communications relay path is secure in respect to third parties monitoring messages 2701 through 2709. The request is forwarded across the secure communications path previously established with the server 2730. In step 2722 process 2720 relays this message. In step 2731 the message is successfully received.

  In step 2732 the server process 2730 searches its database of online tokens to find the second token 2750. The server successfully finds a match as both tokens are online and currently authenticated with the server. The server process can now act as a message router, securely relaying messages between the first token process 2710 and the second token process 2750.
The server process must not disclose messages relayed between tokens to other parties.

In step 2733 the server process securely relays the connection request it received in step 2731 to the second CAD process 2740. The request is successfully received in step 2741.

In step 2742 the second CAD process forwards 2705 the request to the second token process 2750. The message is successfully received in step 2751.

In step 2752 and 2713 the authentication process 2401 and 2402 of figure 20 respectively is performed. In the current embodiment the first and second token mutually authenticate each other. No key exchange has occurred. The steps 2723, 2734, 2743 act as secure relays forwarding the authentication messages between 2752 and 2713.

The process stops in step 2714, 2724, 2735, 2745 and 2753.

Figure 23 illustrates how it is possible for a requesting party to establish a secure connection with a target party, even when the current network address of the target party is unknown to the requesting party. By establishing secure communications between the two parties, it is possible for two clients to actively negotiate further connectivity details. If a target party has a DSC publicly associated with their identity, it is possible to adapt the protocol to relay messages from the requesting party's DSC to the target party's DSC with this information.

Figure 24 is a flowchart 2800 illustrating a process used during a key exchange process of a preferred embodiment. Flowchart 2800 illustrates a first token process 2810 performing a key exchange operation with the second token process 2850. The first and second tokens have not previously performed a key exchange.

Process 2820 is the CAD process for the first token process 2810. Process 2830 is a server process. Process 2830 is also known as a DSC operating as a KTC. Process 2840 is a CAD process for the second token process 2850.

The process 2810, 2820, 2830, 2840 and 2850 may run concurrently. The arrows 2802 and 2805 illustrate message flows between processes 2810 and 2820. The arrow 2806 illustrates a message flow between process 2820
and 2830. The arrows 2801 and 2803 illustrate message flow between process 2820 and 2840. The arrow 2807 illustrates a message flow between process 2830 and 2840. The arrows 2804 and 2808 illustrate message flows between process 2840 and 2850.

The context to the flow chart 2800 in the current embodiment is as follows:

- The first token process 2810 and second token process 2850 have authenticated themselves with the server process 2830 as described in figure 22.

- The first token process 2810 and second token process 2850 have mutually authenticated each other as described in figure 23. In this and preferred embodiments, as part of the authentication process, a network address and port addresses for both tokens were exchanged.

In step 2821 the first CAD process 2820 has established an Internet socket on the Internet address and port number as supplied during mutual token authentications described in figure 23. In step 2841 the second CAD process 2840 has opened an Internet server socket connection on the Internet address and port number as supplied during mutual token authentications described in figure 23.

In step 2822 and 2842 an Internet network session 2801 is established across a public network.

In step 2811 a first key material 2211, a second key material 2213 and a third key material 2215 are generated using a secure true RNG.

In step 2812 a copy of the first key material 2211 is transmitted 2802 and successfully received in step 2851. The first key material is transmitted as ciphertext 2802, and as cleartext 2803 and as ciphertext 2804, being relayed in step 2823 and 2843.

In step 2813 a copy of the second key material 2213 is transmitted 2805 and successfully received in step 2852. The second key material is transmitted as ciphertext 2805, 2806, 2807 and 2808, being relayed in step 2824, 2831 and 2844.

In step 2853 and 2814 session key material is derived from the first and second key material 2211 and 2213.
In step 2815, 2825, 2832, 2845 and 2854 the process 2800 stops. The security of the system may rely upon the server process 2830:

- not publishing the second key material 2213; and
- not actively seeking the value of the first key material 2211; and
- not actively seeking the value of other key material such as 2215 if it is exchanged between the first token process 2810 and the second token process 2850.

The act of the server process 2830 actively seeking the value of other key material may require that it collude with one or more other parties to monitor network communications.

A party monitoring the communications between processes 2810, 2820, 2830, 2840 and 2850 that is not the KSC 2830 cannot determine the value of the session key.

In a preferred embodiment the second token process 2850 reciprocates the above described process for the first token 2810 by securely generating key material 2212 and key material 2214 and sending this to token 2810 over the two distinct paths, and where the session key is then derived from key materials 2211, 2212, 2213, and 2214.

In a preferred embodiment, the session key material further relies on previous session keys established between the first token process 2810 and the second token process 2850.

Figure 25 describes a key exchange performed over an overlay network that uses the building blocks previously described for a key exchange as illustrated in figure 7, 8 and 9. Figure 25 is a flowchart 2900 illustrating a further key exchange process according to a preferred embodiment. Flowchart 2900 illustrates a first token process having performed a key exchange according to figure 24 with a second token process requesting an additional key exchange step with the assistance of a third token process. The third token process is adapted to perform as a KTC.

Process 2910 is the combined operations of a first token process and the CAD process for the first token process. Process 2920 is the combined operations of a third token process and the CAD process for the third token process. Process 2930 is the combined operations of a second token process and the CAD process for the second token process.
The process 2910, 2920 and 2930 may run concurrently. The arrows 2901 and 2902 illustrate message flows between process 2910 and 2930. The arrow 2903 illustrates a message flow between process 2910 and 2920. The arrows 2904 and 2905 illustrate message flows between process 2920 and 2930.

The context to the flow chart 2900 is as follows:

- The first token process 2910, second token process 2930 and third token process 2920 have each authenticated themselves with one or more common server processes according to the process described in figure 15.
- The first token process 2910 and second token process 2930 have performed a key exchange according to the process described in figure 17.
- The first token process 2910 and the third token process 2920 have performed a key exchange according to the process described in figure 24.

The process starts in step 2910, 2920, 2930.

In step 2911 the first token process 2910 transmits 2901 an additional key exchange request with the second token process 2930. The request is successfully received in step 2931.

In step 2912 a 512-bit random number x is generated. In the present embodiment the random number x is equivalent in purpose to key material 2215.

In step 2932 a random number y is generated. The random number y is used to uniquely identify this additional exchange request.

In step 2933 a copy of the random number y is transmitted 2902 and successfully received in step 2913.

In step 2914 the first token process requests the third token process establish a network connection with the second token process using the same connection details previously used to establish a network connection between the first and second token. As part of this request, the first token process also securely transmits 2903 a copy of the random numbers x and y to the third token. The data transmitted 2903 is successfully received in step 2921.
In step 2922 and 2934 the third token opens a socket and establishes an Internet connection with the second token using the information supplied in 2903.

In step 2923 a copy of the random numbers x and y are transmitted 2905 as cleartext and successfully received in step 2935.

In step 2936 the value of y is received in step 2935 and is used to identify the additional key material x as being associated with this key exchange operation.

In steps 2915 and 2936 an updated session key is derived from the current session key and the key material x.

The process stops at step 2916, 2924, 2937.

In a preferred embodiment of the current invention, as part of a key exchange each token requires key material be sent through at least one KSC of its selection. For example, key material for a key exchange may be transmitted across the following paths as illustrated in figure 5 and figure 10:

- token A, mutually selected KSC, token B
- token A, token A's selected KSC, token B's selected KSC party, token B.
- token A, token A's selected KSC, token B.
- token A, token B's selected KSC, token B.
- token A, token B.

The previous processes described can be readily adapted by those skilled in the art to achieve all possible relay combinations.

Figures 26 through to 30 illustrate a preferred embodiment that allows both tokens to negotiate the choice of DSC such that the negotiated subset of proposed DSC satisfies the trust requirements of both tokens.

Figure 26 illustrates an element 3001 of a set (not illustrated) stored by each of the negotiating parties, and the fields in that element. The element 3001 contains 4 fields 3010, 3011, 3012 and 3013. In the presently described embodiment the field 3010 contains the identity of a organization, which federation(s) the organization is a participant in and this may be multiple if the organization is present in multiple levels of federation. Field 3011 contains the identity and details of a DSC. Field 3012 contains a public identifier of the
DSC. Field 3013 contains a private key of the party registered with the DSC respectively.

Figure 27 illustrates 3100 two instantiated sets of elements of type 3001 of figure 26 for three parties. The first party has instantiated sets 3101 and 3105. The second party has instantiated sets 3102 and 3106. The third party has instantiated sets 3103 and 3107. Instantiated set 3101 is populated with four distinct elements 3110, 3111, 3112, 3113. Instantiated set 3105 is populated with 1 distinct element 3115. Instantiated set 3102 is populated with three distinct elements 3120, 3121, 3123. Instantiated set 3106 is populated with two distinct elements 3125, 3126. Instantiated set 3103 is populated with three distinct elements 3130, 3131, 3132. Instantiated set 3107 is populated with zero elements illustrated by the empty place holder 3135. The instantiated set 3101 stores the details of the relationship between the first party and the one or more DSCs it has exchanged secrets with. The instantiated sets 3102 and 3103 store equivalent relationships for the second and third party respectively. The instantiated set 3105 stores the details of relationships between the first party and the zero or more federations, organizations or DSCs that the first party does not wish to include in any key exchange transaction. The instantiated set 3106 and 3107 store equivalent relationships for the second and third party respectively. This illustrates how each party may have unique relationships both in the DSC they have prior shared secrets with and the DSC they explicitly refuse to interact with.

Figure 28 illustrates parts of a message passed between negotiating parties, and the fields in that message 3201. In the presently-described embodiment field 3210 stores the number of sub fields 3202 and 3203 that are present in the message 3201. The field 3202 has two fields 3211 and 3212 that store fields of type 3010 and 3011 respectively. The field 3203 if present has two fields 3213 and 3214 that store fields of type 3010 and 3011 respectively. This message allows one party to inform one or more other parties as to a selection of one or more DSC.

Figure 29 illustrates an instantiated set 3301 of two or more DSC selections achieved by each of the negotiating parties. The field 3302 has two elements 3310 and 3311 that store fields of type 3010 and 3011.
respectively. The field 3303 has two elements 3312 and 3313 that store fields of type 3010 and 3011 respectively.

Figure 30 is a flow chart 3400 of a process used within a preferred embodiment of the present invention. The left-hand side 3410 illustrates a first participant's process of a multi-path negotiation process. The right-hand side 3430 illustrates a second participant's process of a multi-path negotiation process. The processes 3410 and 3430 may run concurrently. The three directed arrows 3405, 3406, and 3407 illustrate bidirectional message flows between processes 3410 and 3430.

The multi-path negotiation process allows each participant to select one or more KSC to participate in the key exchange (using message type 3201), for each party to evaluate the suitability of each other participants recommendation on KSC to participate, for each party to communicate to the other parties its acceptable subset of suggested KSC (also using message type 3201), and to allow each party to make a decision as to if it wishes to continue participating with the collective aggregated set of approved KSC 3301. The present embodiment illustrates the negotiation of KSC between two parties.

The context to flow chart 27 is:

- The processes 3410 and 3430 have established a bidirectional communications session but may not have exchanged key material over one or more KSCs. In the presently-described embodiment this link is initially unencrypted. In a preferred embodiment all unencrypted messages transmitted across the unencrypted link are authenticated using candidate post quantum secure digital signatures to prevent against man-in-the-middle attacks. In the presently-described embodiment the two parties have mutually authenticated each other as described in figure 21.

The process 3400 begins in step 3410 and 3430.

In step 3411 process 3410 sends a set of at least one KSC selected by process 3410 in message type 3201 to process 3430. Process 3430 receives the messages successfully in step 3431.
In step 3431 process 3430 sends a set of at least one KSC selected by process 3430 to process 3410. Process 3410 receives the message successfully in steps 3411.

In step 3412 the process 3410 prepared to iterate through the set of at least one KSC selected by process 3430. The first entry in the set is selected. An empty second set is created to store the approved KSC.

Step 3413 is a node.

In step 3414 if there are no more entries in the list to process the process continues at step 3416. The selected entry is checked against the list of unacceptable KSC for process 3410. If the selected entry in the set is appropriate the process continues at step 3415, otherwise the next entry in the list is selected and the process continues at step 3413.

In step 3415 the selected entry is appended to the approved set of KSC. The next entry in the list is selected. The process continues at step 3413.

In step 3416 the accepted set of elements is prepared using message format 3201, sent and successfully received by process 3430 in step 3436.

In step 3432 the process 3430 prepares to iterate through the set of at least one KSC selected by process 3410. The first entry in the set is selected. An empty second set is created to store the approved KSC.

Step 3433 is a node.

In step 3434 if there are no more entries in the list to process the process continues at step 3436. The selected entry is checked against the list of unacceptable KSCs for process 3430. If the selected entry in the set is appropriate the process continues at step 3435, otherwise the process selects the next entry in the list and continues at step 3433.

In step 3435 the selected entry is appended to the approved set of KSC. The next entry in the list is selected. The process continues at 3433.

In step 3436 the accepted set of elements is prepared using message format 3201, sent and successfully received by process 3410 in step 3416.

In step 3417 and 3437 the elements of the two sets of acceptable KSC are merged into a single set 3301.
In step 3418 a decision is made if the elements in the set 3301 satisfy the trust requirements for a secure key exchange according to process 3410. The trust requirements may include ensuring:

• that there are at least two distinct federations participating in the key exchange;
• that at least two portions of key material will be transmitted securely between the participants;
• any user selected requirements;
• that the process 3410 trusts that at least two participating KSCs will not collude to share key material.

In step 3438 a decision is made if the elements in the set 3301 satisfy the trust requirements for a secure key exchange according to process 3430. The trust requirements may include ensuring:

• that there are at least two distinct federations participating in the key exchange;
• that at least two portions of key material will be transmitted securely between the participants;
• any user selected requirements;
• that the process 3430 trusts that at least two participating KSCs will not collude to share key material.

In step 3419 process 3410 communicates the result of decision 3418. If the decision is to proceed, further information required to establish the negotiated multi-path communications to complete the key exchange is sent to process 3430.

In step 3439 process 3430 communicates the result of decision 3438. If the decision is to proceed, further information required to establish the negotiated multi-path communications to complete the key exchange is sent to process 3410.

The processes 3410 and 3430 stop at step 3420 and 3440.

**Key exchange as described with SSL**

As previously described random nonces exchanged between users may not be entirely composed of randomly generated numbers.

Nonces may be composed of a Secure Socket Layer (SSL) messages which can be encrypted under an 'all-or-nothing' transformation. Portions of
the SSL message may derive from freshly generated random numbers in which case additional nonce data need not be included in the transformation. An all-or-nothing transformation can be used to expand any message into a larger message, such that the message cannot be decoded if any of the information is missing. This allows SSL messages to be nested within multi-path relayed data between two users. It is envisaged that if an experimental PQS key exchange based on public key technologies becomes ready for field trials, it is possible to embed this experimental key exchange technique within embodiments of the present invention with reduced risk. Alternatively, short-term solutions such as using D&H-20000 instead of D&H-1024 used in SSL may provide an additional few more years of security for high-end workstations until sufficiently large quantum computers exist to break it.

**The use of public key cryptography between a KSC and a user**

In a further preferred embodiment a KSC server has a dedicated hardware circuit that implements arithmetic operations on integers well over 20,000 bits in length. The hardware circuit searches for large prime numbers. Large primes are then used by a hardware circuit capable of fast D&H or RSA operations on large fields to generate an asymmetric key pair and perform encryption or decryption operations. The large public key is signed using a PQS signature and sent with a large nonce generated by the KSC to any user that requests it. Each user receives the nonce, public key and certificate, verifies the certificate, generates a fresh large nonce, and encrypts the users nonce using the KSC public key within the desktop processing environment. The encrypted nonce is sent to the KSC, decrypted, hashed with the servers nonce and associated with the user. The user also hashes the two nonces to generate its copy of the session key. In a preferred embodiment the user and KSC have previously exchange key material as described in figure 13 and this new key material is hashed with the existing key material to establish the current shared secret. In a further preferred embodiment new key material is exchanged between the user and the KSC as described in figure 13 and combined with the key material generated by the public key operation. In a further preferred embodiment, the nonce and encrypted nonce previously described are encoded under an all-or-nothing transformation and are split.
into parts and encrypted under two or more distinct end-to-end secure paths as described in figure 13.

These preferred embodiments allows the slow public key generation operation to be performed in hardware once by KSC on behalf of thousands or millions of users, and the relatively cheap asymmetric encryption operation to be performed by any one of the users wishing to register with the KSC. By way of illustration, the encryption operation may be 2 hours on a users desktop, where generation of a public key by the user might take several weeks in software and the KSC may be able to generate the key in a few days and decrypt the operation in minutes using hardware. The most expensive operations are confined to the KSC and the remaining processing costs are distributed across any additional user-to-user key exchanges performed with the assistance of the KSC.

**Offline communications such as email**

Store and forward systems can be used to simulate the behaviour of online communications systems. One such store and forward example is an email server. Another such example is where a KSC adapted to behave as a KTC which is additionally adapted as a store and forward system. It is possible to transmit key exchange messages using store and forward systems without loss of generality. After a successful key exchange between the sender and the target, two additional session keys used specifically for sending emails may be negotiated. Authenticated encrypted messages can then be sent to each other without incurring additional delays using a stateless communications protocol where the IV of each encrypted packet is a large random value.

See also above mentioned Australian provisional patent application 2008904612 for more information on using the DSC to perform secure email operations.

**Further improving performance**

The descriptions of the embodiments above illustrate a correctly operating key exchange system that can scale. Several generic network and network application optimisation techniques known in the art can be used to improve the run-time network performance of large-scale implementations.
For example, in a preferred embodiment of the present invention, users will remain logged in to their trusted DSC participating in global federations in between data and key exchange operations with different users. For example in figure 5, KSC 630 authenticates with KSC 601 and 602 before establishing two or more key exchanges with two or more users. This ensures that the communication overhead to establish these connections is distributed over several connections and simultaneously reducing the latencies when performing additional key exchange operations with other users.

In a further preferred embodiment of the present invention, KSC ensure communications paths with other frequently used KSC remain available (connection pooling / session pooling), in this way establishing a new secure communications path is not required for each user’s key exchange operation. For example in figure 6, the DSC 1001, 1002, 1003, 1004 are in one building and the network sessions between the DSC 1001, 1002, 1003, 1004 are permanently maintained.

In a further preferred embodiment, DSC located in a first building have a first aggregating relay server located in that building that aggregates communications between it and a second aggregating relay server located in a second building, the second aggregating relay sever aggregating requests to DSC located in that second building.

The embodiments described above provide a candidate PQS data exchange and a candidate PQS key exchange (PQSKX) protocol that is a pragmatic solution that employs a fundamentally different but complementary technique to existing public key technologies. The PQSKX exchange ensures that no KSC in the transaction can monitor all the key material exchanged between users without large-scale collusion. Additionally the load of a key exchange between any two users is isolated to only the KSC participating in a key exchange. A preferred embodiment of the invention enables the system to scale to support an arbitrary number of KSC, users and key exchanges on a single overlay network. Of great advantage, a preferred embodiment of the PQSKX system allows users to take advantage of large-scale infrastructure while simultaneously establishing their own trust relationships to prevent likely collusions occurring. Specifically, embodiments of the present invention can achieve the necessary trust levels of each user by allowing each user to
individually assess if collusion between all participating parties is probable. These properties allow the preferred QSKX system to scale to be accepted by a very large number of users across a federated system of KSC servers.

In further preferred embodiments of the present invention all fields in a cryptographic protocol employ Tag-Length-Value encoding of all fields, see [010].

In further preferred embodiments of the present invention key material sent over at least one segment of distinct path between two hosts is encrypted using an entropically secure encryption function as described in the publication [011].

While this invention has been described in connection with specific embodiments thereof, it will be understood that it is capable of further modification(s). This application is intended to cover any variations uses or adaptations of the invention following in general, the principles of the invention and including such departures from the present disclosure as come within known or customary practice within the art to which the invention pertains and as may be applied to the essential features hereinbefore set forth.

As the present invention may be embodied in several forms without departing from the spirit of the essential characteristics of the invention, it should be understood that the above described embodiments are not to limit the present invention unless otherwise specified, but rather should be construed broadly within the spirit and scope of the invention as defined in the appended claims. The described embodiments are to be considered in all respects as illustrative only and not restrictive.

Various modifications and equivalent arrangements are intended to be included within the spirit and scope of the invention and appended claims. For example, with respect to the embodiments described above with reference to figures 7, smart card tokens are not essential in as much as the software processes that may run on a smart card token could also run on a processor that runs the card access device (CAD) processes. In this respect, the smart card process and card access device processes could be merged as a single process without affecting the scope of the embodiments so described.
Therefore, the specific embodiments are to be understood to be illustrative of the many ways in which the principles of the present invention may be practiced. In the following claims, means-plus-function clauses are intended to cover structures as performing the defined function and not only structural equivalents, but also equivalent structures. For example, although a nail and a screw may not be structural equivalents in that a nail employs a cylindrical surface to secure wooden parts together, whereas a screw employs a helical surface to secure wooden parts together, in the environment of fastening wooden parts, a nail and a screw are equivalent structures.

It should be noted that where the terms "server", "secure server" or similar terms are used herein, a communication device is described that may be used in a communication system, unless the context otherwise requires, and should not be construed to limit the present invention to any particular communication device type. Thus, a communication device may include, without limitation, a bridge, router, bridge-router (router), switch, node, or other communication device, which may or may not be secure.

It should also be noted that where a flowchart or equivalent illustration of processes or systems is used herein to demonstrate various aspects of the invention, it should not be construed to limit the present invention to any particular logic flow or logic implementation. The described logic may be partitioned into different logic blocks (e.g., programs, modules, functions, or subroutines) without changing the overall results or otherwise departing from the true scope of the invention. Often, logic elements may be added, modified, omitted, performed in a different order, or implemented using different logic constructs (e.g., logic gates, looping primitives, conditional logic, and other logic constructs) without changing the overall results or otherwise departing from the true scope of the invention.

Various embodiments of the invention may be embodied in many different forms, including computer program logic for use with a processor (e.g., a microprocessor, microcontroller, digital signal processor, or general purpose computer), programmable logic for use with a programmable logic device (e.g., a Field Programmable Gate Array (FPGA) or other PLD), discrete components, integrated circuitry (e.g., an Application Specific
Integrated Circuit (ASIC)), or any other means including any combination thereof. In an exemplary embodiment of the present invention, predominantly all of the communication between users and any other communication device, such as a server for example, is implemented as a set of computer program instructions that is converted into a computer executable form, stored as such in a computer readable medium, and executed by a microprocessor under the control of an operating system.

Computer program logic implementing all or part of the functionality where described herein may be embodied in various forms, including a source code form, a computer executable form, and various intermediate forms (e.g., forms generated by an assembler, compiler, linker, or locator). Source code may include a series of computer program instructions implemented in any of various programming languages (e.g., an object code, an assembly language, or a high-level language such as Fortran, C, C++, JAVA, or HTML) for use with various operating systems or operating environments. The source code may define and use various data structures and communication messages. The source code may be in a computer executable form (e.g., via an interpreter), or the source code may be converted (e.g., via a translator, assembler, or compiler) into a computer executable form.

A computer program(s) implementing all or part of the functionality where described herein may be fixed in any form (e.g., source code form, computer executable form, or an intermediate form) either permanently or transitorily in a tangible storage medium, such as a semiconductor memory device (e.g., a RAM, ROM, PROM, EEPROM, or Flash-Programmable RAM), a magnetic memory device (e.g., a diskette or fixed disk), an optical memory device (e.g., a CD-ROM or DVD-ROM), a PC card (e.g., PCMCIA card), or other memory device. The computer program may be fixed in any form in a signal that is transmittable to a computer using any of various communication technologies, including, but in no way limited to, analog technologies, digital technologies, optical technologies, wireless technologies (e.g., Bluetooth), networking technologies, and inter-networking technologies. The computer program may be distributed in any form as a removable storage medium with accompanying printed or electronic documentation (e.g., shrink wrapped software), preloaded with a computer system (e.g., on system ROM or fixed
disk), or distributed from a server or electronic bulletin board over the communication system (e.g., the Internet or World Wide Web).

Hardware logic (comprising programmable logic for use with a programmable logic device) implementing all or part of the functionality where described herein may be designed using traditional manual methods, or may be designed, captured, simulated, or documented electronically using various tools, such as Computer Aided Design (CAD), a hardware description language (e.g., VHDL or AHDL), or a PLD programming language (e.g., PALASM, ABEL, or CUPL).

Programmable logic may be fixed either permanently or transitorily in a tangible storage medium, such as a semiconductor memory device (e.g., a RAM, ROM, PROM, EEPROM, or Flash-Programmable RAM), a magnetic memory device (e.g., a diskette or fixed disk), an optical memory device (e.g., a CD-ROM or DVD-ROM), or other memory device. The programmable logic may be fixed in a signal that is transmittable to a computer using any of various communication technologies, including, but in no way limited to, analog technologies, digital technologies, optical technologies, wireless technologies (e.g., Bluetooth), networking technologies, and Internetworking technologies. The programmable logic may be distributed as a removable storage medium with accompanying printed or electronic documentation (e.g., shrink wrapped software), preloaded with a computer system (e.g., on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the communication system (e.g., the Internet or World Wide Web).

"Comprises/comprising" when used in this specification is taken to specify the presence of stated features, integers, steps or components but does not preclude the presence or addition of one or more other features, integers, steps, components or groups thereof. Thus, unless the context clearly requires otherwise, throughout the description and the claims, the words 'comprise', 'comprising', and the like are to be construed in an inclusive sense as opposed to an exclusive or exhaustive sense; that is to say, in the sense of "including, but not limited to".

References


CLAIMS:
1. A many to many data exchange system comprising:
   a network comprising at least two DSCs and at least two hosts in which each of the at least two hosts share at least one key material with at least one of the at least two DSCs;
   data exchange means for performing a data exchange process, the data exchange process utilising a subset of at least two of the at least two DSCs and comprising:
   the first host interacting with at least one DSC, each of which is adapted to share at least one key material with the first host;
   and
   the second host interacting with at least one DSC each of which is adapted to share at least one key material with the second host;
   the data exchange means comprising:
   secure communication paths between the first host and its corresponding DSC it shares key material with, the secure communications paths adapted to use a cryptographic primitive keyed with the shared key material; and
   secure communication paths between the second host and its corresponding DSC it shares key material with, the secure communications paths adapted to use a cryptographic primitive keyed with shared key material; and
   at least two distinct communications paths for transporting data between the first and second host wherein at least one of the first host's DSC that it shares key material with is on the first distinct communications path and at least one of the second host's DSC that it shares key material with is on the second distinct communications path;
   and in which the first host sends to the second host:
   at least one first data over the first distinct communications path; and
   at least one second data over the second distinct communications.
2. A many to many data exchange system as claimed in claim 1 further comprising initiating means for invoking an initiating process between a first host and a second host.

3. A many to many data exchange system as claimed in claim 1 or claim 2 in which each of the DSC on the first and second distinct communications paths are only on one of the at least two distinct communications paths.

4. A many to many data exchange system as claimed in claim 2 or claim 3 in which:
   at least one key material which is shared between a host and a DSC, is a symmetric key material that has been negotiated between that host and that DSC in a post quantum secure method, and the secure communications path between that host and that DSC is a post quantum secure path that is adapted to use a symmetrically keyed cryptographic primitive.

5. A many to many data exchange system as claimed in any one of claims 2, 3 and 4 in which the network is an overlay network.

6. A many to many data exchange system as claimed in claim 5 where there are at least three DSCs in the overlay network, and wherein at least three of the at least three DSCs are associated with a unique set of hosts with respect to any other at least three DSCs.

7. A many to many data exchange system as claimed in claim 5 or claim 6 in which the value of the data which is sent between the first host and the second host across one distinct communication path is always encrypted.

8. A many to many data exchange system as claimed in claim 5 or claim 6 in which the value of the data transmitted across each of at least two distinct communications paths is always encrypted.
9. A many to many data exchange system as claimed in any one of the claims 6 to 8 in which the initiation of communications between the first host and the second host comprises a process of negotiation by the first host and the second host that results in an agreement on the selection of two or more DSCs.

10. A many to many data exchange system as claimed in claim 9 in which the first host and second host exchange information identifying the DSC with which that host shares a key.

11. A many to many data exchange system as claimed in any one of claims 5 to 10 in which there are at least four DSCs, and at least two federations, where each of the at least four DSCs is assigned to one of the at least two federations and where at least two DSCs in one federation can establish secure communications with each other.

12. A many to many data exchange system as claimed in claim 11 in which the at least one distinct communications path between the first host and the second host comprises:

20 a DSC in a federation that shares a key with the first host;
a DSC in the same federation that shares a key with the second host;
where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish secure communications between each other.

13. A many to many data exchange system as claimed in claim 11 in which:

30 at least one first distinct communications path between the first host and the second host comprises:
a DSC in a first federation that shares a key with the first host;
a DSC in the first federation that shares a key with the second host;
where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host
can establish secure communications between each other; and at least one second distinct communications path between the first host and the second host comprises:

a DSC in a second federation that shares a key with the first host;

a DSC in the second federation that shares a key with the second host;

where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish secure communications between each other;

where the first federation and second federation are different federations.

14. A many to many data exchange system as claimed in any one of claims 5 to 13 in which:

the data sent between the first host and the second host is a message, the message:

comprises a component which is derived from a random number;

is encoded under an all-or-nothing transformation; and

is partitioned into at least two data parts where the at least two data parts are exchanged over different distinct communications paths.

15. A many to many data exchange system as claimed in any one of claims 5 to 13 further comprising:

the exchange of key material from the first host to the second host to create a session key:

in which the key material which is exchanged from the first host to the second host is partitioned into at least one first data part and at least one second data part; and

the session key is derived from a portion of the at least one first data part and the at least one second data part.
16. A method of communicating data between at least two hosts in a network which comprises:
   at least two DSCs, and
   at least two distinct communications paths for transporting data between a first host and second host;
the method comprising the steps of:
   communicating at least one first data between the first host and a first DSC on a first distinct communications path, where the at least one first data is encrypted using a key which is shared between the first host and the first DSC;
   communicating at least one second data between the second host and a second DSC on a second distinct communications path, where the at least one second data is encrypted using a key which is shared between the second host and the least one second DSC; and
   the first host sending to the second host:
   at least one first data over the first distinct communications path;
   and
   at least one second data over the second distinct communications; and
where each of the DSC on the first and second distinct communications paths are only on one of the at least two distinct communications paths;

17. A method of communicating data between at least two hosts in a network as claimed in claim 5, the method further comprising initiating communications between the first host and the second host.

18. A method of communicating data between at least two hosts in a network as claimed in claim 17, in which the network is an overlay network.

19. A method of communicating data between at least two hosts in a network as claimed in claim 18 where there are at least three DSCs in the overlay network, and wherein at least three of the at least three DSCs are associated with a unique set of hosts with respect to any other of the at least three DSCs.
20. A method of communicating data between at least two hosts in a network as claimed in claim 18 or claim 19, in which the value of the data which is sent between the first host and the second host across one distinct communication path is always encrypted.

21. A method of communicating data between at least two hosts in a network as claimed in claim 18 or claim 19, in which the value of the data transmitted across each of the at least two distinct communications paths is always encrypted.

22. A method of communicating data between at least two hosts in a network as claimed in any one of the claims 18 to 21 in which the initiation of communications between the first host and the second host comprises a process of negotiation by the first host and the second host that results in an agreement on the selection of two or more DSCs.

23. A method of communicating data between at least two hosts in a network as claimed in claim 22 in which the first host and second host exchange information identifying the DSC with which that host shares a key.

24. A method of communicating data between at least two hosts in a network as claimed in any one of claims 18 to 24, in which there are at least four DSCs, and at least two federations, where each of the at least four DSCs is assigned to one of the at least two federations and where at least two DSCs in one federation can establish secure communications with each other.

25. A method of communicating data between at least two hosts in a network as claimed in claim 25, in which the at least one distinct communications path between the first host and the second host comprises:

- a DSC in a federation that shares a key with the first host;
- a DSC in the same federation that shares a key with the second host;
- where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish...
secure communications between each other.

26. A method of communicating data between at least two hosts in a network as claimed in claim 25 in which:

5 at least one first distinct communications path between the first host and the second host comprises:

a DSC in a first federation that shares a key with the first host;

a DSC in the first federation that shares a key with the second host;

10 where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish secure communications between each other; and

at least one second distinct communications path between the first host and the second host comprises:

15 a DSC in a second federation that shares a key with the first host;

a DSC in the second federation that shares a key with the second host;

20 where the DSC that shares a key with the first host and the DSC in the same federation that shares a key with the second host can establish secure communications between each other;

where the first federation and second federation are different federations.

25 27. A method of communicating data between at least two hosts in a network as claimed in any one of claims 18 to 26, in which:

the data sent between the first host and the second host is a message, the message:

comprises a component which is derived from a random number;

is encoded under an all-or-nothing transformation; and

is partitioned into at least two data parts where the at least two data parts are exchanged over different distinct communications paths.
28. A method of communicating data between at least two hosts in a network as claimed in any one of claims 18 to 26, further comprising:
   the exchange of key material from the first host to the second host to create a session key:
   in which the key material which is exchanged from the first host to the second host is partitioned into at least one first data part and at least one second data part; and
   the session key is derived from a portion of the at least one first data part and the at least one second data part.

29. A host which is adapted for use as a host within a many to many data exchange system as defined in any one of claims 1 to 4.

30. A host which is adapted for use as a host within a many to many data exchange system as defined in any one of claims 5 to 15.

31. A DSC which is adapted for use as a DSC within a many to many data exchange system as defined in any one of claims 1 to 4.

32. A DSC which is adapted for use as a DSC within a many to many data exchange system as defined in any one of claims 5 to 15.

33. A many to many data exchange system as claimed in any one of claims 1 to 15, substantially as described with reference to any one or more of the drawings.

34. A many to many data exchange system, substantially as described with reference to the drawings.

35. A method of communicating data between at least two hosts in a network as claimed in any one of claims 16 to 28, substantially as described with reference to any one or more of the drawings.

36. A method of communicating data between at least two hosts in a
network, substantially as described with reference to any one or more of the drawings.
A. CLASSIFICATION OF SUBJECT MATTER

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

B. DOCUMENTS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L  H04W  G06P

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, COMPENDEX, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>figures 1-3,6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0008]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0033]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0047]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0052]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0060] - paragraph [0061]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>claim 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0025] - paragraph [0034]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0071] - paragraph [0075]</td>
<td></td>
</tr>
</tbody>
</table>

X Further documents are listed in the continuation of Box C

See patent family annex

Date of the actual completion of the international search:

7 April 2009

Date of mailing of the international search report:

16/04/2009

Name and mailing address of the ISA:

European Patent Office, P B 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel (+31-70) 340-2040,
Fax (+31-70) 340-3016

Authorized officer:

Schlossmaier, KL aus
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>PEEV M ET AL: &quot;Quantenkryptographisch gesicherte Netzwerke zur Schlüsselverteilung: Grundlagen, Topologien und das Quantum-Back-Bone des Integrierten Projekts = Quantum key distribution networks: basic concepts, topologies and the SECOQC quantum back bone&quot;</td>
<td>1-36</td>
</tr>
<tr>
<td></td>
<td>ISSN: 0932-383X, Section 6 and 7</td>
<td></td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AT 412932 B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU 2003285968 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1563638 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2006507735 T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2004120528 A1</td>
<td>24-06-2004</td>
<td>NONE</td>
<td></td>
</tr>
</tbody>
</table>