## Title: SECURE MESSAGE MANAGEMENT SYSTEM

A messaging system for transferring messages between any two of a plurality of user machines. The system includes a message manager for overseeing secure message operations. The message manager is associated with a management database which stores separate secure channel properties for each message channel between the manager and each of the user machines. Additionally, per user machine, the device includes a local secure message processor which secures messages. The processor is associated with a local database. The local database stores secure channel properties per channel between the user machine and each other user machine with which the user machine communicates, including the manager.
<table>
<thead>
<tr>
<th>Code</th>
<th>Country</th>
<th>Code</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>Albania</td>
<td>ES</td>
<td>Spain</td>
</tr>
<tr>
<td>AM</td>
<td>Armenia</td>
<td>FI</td>
<td>Finland</td>
</tr>
<tr>
<td>AT</td>
<td>Austria</td>
<td>FR</td>
<td>France</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
<td>GA</td>
<td>Gabon</td>
</tr>
<tr>
<td>AZ</td>
<td>Azerbaijan</td>
<td>GB</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>BA</td>
<td>Bosnia and Herzegovina</td>
<td>GE</td>
<td>Georgia</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
<td>GH</td>
<td>Ghana</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
<td>GN</td>
<td>Guinea</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
<td>GR</td>
<td>Greece</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
<td>HU</td>
<td>Hungary</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
<td>IE</td>
<td>Ireland</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
<td>IL</td>
<td>Israel</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
<td>IS</td>
<td>Iceland</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
<td>IT</td>
<td>Italy</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
<td>JP</td>
<td>Japan</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
<td>KE</td>
<td>Kenya</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
<td>KG</td>
<td>Kyrgyzstan</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d'Ivoire</td>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
<td>KR</td>
<td>Republic of Korea</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
<td>KZ</td>
<td>Kazakhstan</td>
</tr>
<tr>
<td>CU</td>
<td>Cuba</td>
<td>LC</td>
<td>Saint Lucia</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
<td>LI</td>
<td>Liechtenstein</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
<td>LK</td>
<td>Sri Lanka</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
<td>LR</td>
<td>Liberia</td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
<td>LS</td>
<td>Lesotho</td>
</tr>
<tr>
<td>LT</td>
<td>Lithuania</td>
<td>LU</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>LV</td>
<td>Latvia</td>
<td>MC</td>
<td>Monaco</td>
</tr>
<tr>
<td>MD</td>
<td>Republic of Moldova</td>
<td>MG</td>
<td>Madagascar</td>
</tr>
<tr>
<td>MK</td>
<td>The former Yugoslavia</td>
<td>ML</td>
<td>Mali</td>
</tr>
<tr>
<td>MN</td>
<td>Mongolia</td>
<td>MR</td>
<td>Mauritania</td>
</tr>
<tr>
<td>MW</td>
<td>Malawi</td>
<td>MX</td>
<td>Mexico</td>
</tr>
<tr>
<td>NE</td>
<td>Niger</td>
<td>NL</td>
<td>Netherlands</td>
</tr>
<tr>
<td>NO</td>
<td>Norway</td>
<td>NZ</td>
<td>New Zealand</td>
</tr>
<tr>
<td>PL</td>
<td>Poland</td>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td>RO</td>
<td>Romania</td>
<td>RU</td>
<td>Russian Federation</td>
</tr>
<tr>
<td>SD</td>
<td>Sudan</td>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td>SI</td>
<td>Slovenia</td>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>SN</td>
<td>Senegal</td>
<td>SZ</td>
<td>Swaziland</td>
</tr>
<tr>
<td>TD</td>
<td>Chad</td>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>TJ</td>
<td>Tajikistan</td>
<td>TM</td>
<td>Turkmenistan</td>
</tr>
<tr>
<td>TR</td>
<td>Turkey</td>
<td>TT</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td>UA</td>
<td>Ukraine</td>
<td>UG</td>
<td>Uganda</td>
</tr>
<tr>
<td>US</td>
<td>United States of America</td>
<td>US</td>
<td>United States of America</td>
</tr>
<tr>
<td>UZ</td>
<td>Uzbekistan</td>
<td>VN</td>
<td>Viet Nam</td>
</tr>
<tr>
<td>YU</td>
<td>Yugoslavia</td>
<td>ZW</td>
<td>Zimbabwe</td>
</tr>
</tbody>
</table>
SECURE MESSAGE MANAGEMENT SYSTEM

FIELD OF THE INVENTION

The present invention relates generally to electronic messaging management systems and, in particular, to secure e-mail communication over the public digital network.

BACKGROUND OF THE INVENTION

One of the major uses of the electronic data communication is the transfer of messages, which can include the transfer of e-mail, documents, notes, etc., and any other type of electronic data. A message contains at least the source address, the destination address and the message data. When a message is transferred over a telecommunication network it may be sent directly or relayed between one or more servers on its journey from a source user to a destination user. This opens up the possibility of message, including e-mail, exposure or theft by a multiplicity of unknown users, foreign entities and others.

E-mail security is a major issue for digital network users, such as Internet users. Individuals are concerned that unknown persons may intercept their personal messages. Companies are concerned about possible leaks or theft of e-mail containing proprietary trade secrets.

Fig. 1A details an e-mail network architecture 10 designed to secure privacy of e-mail communication.
System 10 comprises a plurality of e-mail clients, known as Users, and a multiplicity of Internet servers 12, generally designated in the drawings as server 12A, server 12B, and so on. In an exemplary situation, when user 1 transfers e-mail to user 2, the e-mail package is forwarded through the Internet, from server 12A to server 12B, until it reaches its destination, user 2.

Generally, the destination user, in this example user 2, has a mailbox 13, which is located at the server 12B which services user 2. Mail box 13 temporarily stores user 2's mail while user 2 is off-line. When user 2 logs-on, he retrieves his mail from mailbox 13.

The users have at their disposal a computer 26 installed with an operating system 28, such as Windows. Additionally installed in computer 26, and sitting on top of the operating system 28, is an e-mail package 20, a secured channel data base (SCDB) 22 and a mail securer 30.

When e-mail is sent securely from one user to another, it is sent along a secure channel. A secured channel between any two users is set of properties, known to just the two users, such as the agreed data and key encryption algorithms, keys, key lengths etc..

Fig. 1A shows only two users, user 1 and user 2; however, it is understood that e-mail transfer can be performed between any number of users.

Additionally, although detail is shown in Fig. 1 for user 1 only, it is apparent that all other users, including user 2, comprise similarly configured elements.

The e-mail package 20 is any commercially available e-mail package such as Outlook, Eudora, or Netscape. E-mail package 20 comprises a general electronic address book 18, an inbox 23 and an outbox 24, of the type generally
available with commercial e-mail packages. These elements are employed in the method typically used for commercially available elements of this type.

The mail securer 30 is any commercially available secure e-mail managing system such as MAILguardian, available from Vanguard Security Technologies of Haifa, Israel, the common assignee of the present invention. Generally, the mail securer 30 secures the e-mail packet by performing the following 4 steps:

1) A digital signature is attached to the e-mail packet;

2) the e-mail packet and the digital signature are encrypted, using a random number as a key and any of the commercially available encryption algorithm, such Data Encryption Standard (DES) or Blowfish;

3) the random key is encrypted using a predefined key and the above chosen encryption algorithm, and is attached to the encrypted packet; and

4) The now secured e-mail packet is encapsulated in a regular e-mail message, which displays only the minimal information needed for the transport.

SCDB 22 sits on mail securer 30, and contains only the addresses of those users which have security clearance to receive and send secured e-mail. Usually, these secured users are known as members. There are generally two methods to add users or members to the SCDB 22.

Method 1) User 1 contacts User 2 using any method (i.e. telephone), and together they create a secured channel by exchanging keys. This
procedure, known as the shared secret method, must be repeated for each new user added to the SCDB 22.

Method 2) Alternatively, if user 1 receives an e-mail from user 2, user 1's mail securer 30 will notice that user 2 is using the same security software and automatically creates a secure channel. This attempt will be achieved by a handshake process using internal e-mail messages between User 1's mail securer 30 and User 2's mail securer 30. Although this method avoids the need for user interface as in method 1, one or more e-mail packets may pass through non-secured before the secured channel is created for the two with the appropriate keys.

A secured channel that is created by either of the two above methods is dedicated to the two users who are paired members of that channel. As an example, when user 1 sends e-mail to user 2 along their secured channel, securer 30 of user 1 secures the e-mail packet with the properties of the secure channel, or alternately, this is known as sending a e-mail along the User 1 - User 2 secured channel.

When an e-mail packet is sent to more than one user or addressee, which is known as a multicast packet, each addressee receives the packet via its own secure channel with the sender. Additionally, a multicast can be sent to user members for which secured channel exists, along with non-user members for which there are no secured channels.

Reference is now made to Fig. 1B, a flow chart of an exemplary e-mail transfer via the e-mail network system of Fig. 1A.
User 1, using e-mail package 20, writes (step 40) an e-mail to user 2.

User 1 refers to address book 18 and retrieves (step 42) the address of user 2. User 1 completes the e-mail and request to "SEND". The e-mail package 20 puts (step 44) the e-mail in the outbox 24, and initiates (step 46) connection to the server 12. The e-mail packet passes (step 48) from the e-mail package 20 level to the communication protocol layer (i.e. TCP Socket) at the operating system 28 level. An interceptor (i.e. Layered Socket Program) hook observes the packet before or while in the communication protocol layer, and identifies the packet as e-mail being sent. The Interceptor hook provides (step 49) the packet to the mail securer 30.

Mail securer 30 tries to identify (step 50) the sender and the receiver, user 1 and user 2, respectively, as being a paired member of the SCDB 22. Once they are identified (step 52) as paired members, they are classified as being able to send/receive secured e-mail. Mail securer 30 secures (step 54) the e-mail packet, as detailed hereinabove.

However, if the mail securer 30 does not identify (step 56) user 1 and user 2 as being paired members of the SCDB 22, than the packet is not classified to be sent over a secured channel. At this point mail securer 30 may send (step 57) a warning to user 1, notifying him that the packet is about to be sent non-secured, and waits for instructions. User 1 can elect to either to send (step 58) the packet unsecured along open (unsecured) Internet channels or discard the packet (step 59).

The secured e-mail packet is injected back (step 60) to the operating system 28. The packet travels (step 62) through the telecommunication media
(i.e. Internet), and arrives at user 2's mailbox 13 at the server 12B which services user 2. The packet waits (step 64) at user 2's mailbox 13 until retrieval by user 2's e-mail package 20. The message is intercepted (step 66) at the communication protocol layer and provided to user 2's securer 30. If the pair user 1 - user 2 are members in the user 2's SCDB.22, the e-mail is decrypted (step 67) and injected back to user 2's Operating System 28. User 2's e-mail package 20 receives (step 68) the e-mail in plain text, puts the mail in Inbox 23, and notifies user 2.
SUMMARY OF THE PRESENT INVENTION

It is an object of the present invention to automatically provide security for an electronic messaging system from generally the first moment of installation.

There is therefore provided, in accordance with a preferred embodiment of the present invention, a messaging system for transferring messages between any two of a plurality of user machines. The system includes a message manager for overseeing secure message operations. The message manager is associated with a management database which stores separate secure channel properties for each message channel between the manager and each of the user machines.

Additionally, per user machine, the device includes a local secure message processor which secures messages. The processor is associated with a local database. The local database stores secure channel properties per channel between the user machine and each other user machine with which the user machine communicates, including the manager.

There is additionally provided a message manager which includes means for installing the local secure message processor on at least one of the user machine and provides the associated local database with at least the secure channel properties of the channel between the manager and the user machine, thereby ensuring secure message operations from generally the moment of installation. The local secure message processor includes means for sending messages to the message manager when the secure channel properties of a user to be communicated with are unknown or when the local secure message processor cannot decrypt a message received from another user machine.
There is therefore provided in accordance with a preferred embodiment of the present invention, a method of insuring the transfer of generally secure message between any two of a plurality of user machines. The method including the steps of:

5 generating a secure channel properties between a message manager and each of the plurality of user machines;

storing in a management data base the secure channel properties for each secure channel between the message manager and each of the plurality of user machine;

10 installing a local secure message processor on at least one of the plurality of user machines and providing the processor with an associated local database;

storing in the associated local database secure channel properties between the at least one plurality of user machines and at least the manager, and secure channel properties per channel for each of the secure channels between each of the plurality of user machines and each other user machine; and

15 securing messages between any of the plurality of user machines from generally the moment of installation.

There is additionally provided a method which includes the steps of logging on at more than one the plurality of user machines, and transferring generally secure message by sending messages to the message manager.

20 There is furthermore provided a method wherein generating a secure channel properties includes employing automatic authentication with an automatic key generation, and/or employing an authentication code.
The method additionally includes the steps of sending messages to the message manager when the secure channel properties of a user to be communicated with are unknown or when the local secure message processor cannot decrypt a message packet received from another user machine.

The method furthermore includes the step of regenerating the secure channel properties for the secure channel between the any two of a plurality of user machines without interrupting user communication flow. Alternatively, regenerating the secure channel properties includes regenerating the secure channel properties every predetermined time period, and/or when the local secure message processor cannot decrypt a message packet received from another user machine.
BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood and appreciated more fully from the following detail description taken in conjunction with the drawings in which:

Fig. 1A is a schematic illustration of the architecture of a prior art secured e-mail network;

Fig. 1B is a flow chart of the e-mail transfer via the e-mail network of Fig. 1A;

Fig. 2 is a schematic illustration of the architecture of a secured e-mail network constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 3A is a flow chart of a secured e-mail transfer procedure of the e-mail network of Fig. 2;

Fig. 3B is a flow chart of an alternative secured e-mail transfer procedure of secured e-mail management system of Fig. 2;

Fig. 4A is a flow chart of an auto-authenticated installation procedure of the e-mail network of Fig. 2;

Fig. 4B is a flow chart of a user authenticated installation procedure of e-mail network of Fig. 2;

Fig. 5A is a schematic illustration of the architecture of a multiple location user system used in the secured e-mail network of Fig. 2;

Fig. 5B is a flow chart of a multiple location user e-mail transfer procedure of the secured e-mail network of Fig. 2;
Fig. 6 is a flow chart of a damaged key replacement procedure of the secured e-mail network of Fig. 2; and

Fig. 7 is a flow chart of an aged key refreshing procedure of the secured e-mail network of Fig. 2.
DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Reference is now made to Fig. 2 which is a schematic illustration of the architecture of a secured e-mail management system 100, constructed and operative in accordance with a preferred embodiment of the present invention.

From the moment of installation of the agent at the user's desktop, the present invention provides end-to-end secured e-mail communication channels. In contrast to prior art systems, system 100 is transparent to the nominal user.

Similar to system 10, system 100 includes a multiplicity of users, which have at their disposal computers 26, installed with an operating system 128. The multiplicity of users transmit e-mail over the Internet via servers 12. Each user has e-mail package 20, electronic address book 18, inbox 23, outbox 24, and mailbox 13.

Operating system 128 is any available operating system, such as UNIX, MAC or Windows.

System 100 further includes a mail security agent 130, a transparent local secured channel data base (LSCDB) 122, and a mail security manager 112 which includes a global management secured channel data base (MSCDB) 114.

The basic functions of each mail security agent 130 are similar to those of mail securer 30. Additionally, in a manner described in detail below, each agent 130, along with manager 112, automatically installs members in each LSCDB 122.

Similar to prior art, each LSCDB 122 is located locally per user and includes the addresses and keys to the other members with whom the local user communicates. In a preferred embodiment of the present invention, user 1 no
longer needs to input secured users to the LSCDB 122, and hence, the local secured channel database is transparent to him. In an alternative embodiment, LSCDB 122 can be interfaced with in a manner similar to prior art, and hence, LSCDB 122 is also available with a viewable and/or editable option.

Manager 112 is a user in system 100, and is serviced by one of the servers 12. Manager 112 holds the MSCDB 114, which is created and imported with members during set-up and continuously updated, as to be described hereinbelow in detail. MSCDB 114 holds the channel properties and keys to all the users in system 100. Thus, inasmuch as manager 112 holds (via the MSCDB 114) the channel properties and keys to all the users in system 100, manager 112 communicates with all the users of system 100. However, MSCDB 114 does not store any of the properties for the secured channel between any two users, these properties are stored only in the LSCDB of the two users involved.

The users, or members, of the MSCDB 114 are defined by the company which has installed system 100. System 100 services any user with whom the company wishes to include as a recipient/transmitter of secured e-mail, regardless of their physical location or affiliation. As an example, the members of MSCDB 114 can be the employees of the engaging company, who may or may not sit at the same physical site as the company. Alternatively, the member list can also include clients or associates of the company, or any other user with which the company wishes to communicate with on a secured channel.

Since manager 112 is a user of system 100 and is serviced by a server 12, it can communicate with any user anywhere on the Internet network. Once a
user's name and address is added as a member of the MSCDB 114, messages sent to the user are secured, regardless of his location on the Internet.

Manager 112 has the additional task of communicating with each agent 130 via control messages, which can be sent in both directions, from manager to agent, and from agent to manager. Control messages include commands such as "refresh key", "installation complete", "recovered key" "message recovered", etc (to be described in detail hereinbelow). Additionally, the engaging company has the ability to generate a company policy data base 129. Company policy data base 129 is located at the computer 26 of manager 112 and automatically distributed to computers 26 of each user and regulates the operations of illegal or non-secure e-mail transfer.

On a general basis, manager 112 does not participate in message transfer between two users. Manager 112 does participate in message transfers between users if the transfer is the first communication between the users or if there are problems in the transfer.

Reference is now made to Figs. 3A and 3B, which illustrate two types of e-mail transfer. Fig. 3A illustrates direct transfer of e-mail without the participation of manager 112. Fig. 3B illustrates transfer of e-mail with the participation of manager 112. Steps of this embodiment which are similar to those which have been described hereinabove in prior art are similarly numbered, and will not be further described.

Referring now to Fig. 3A, transfer of e-mail from user 1 to user 2 functions generally similar to the prior art of Fig. 1B, except that all steps previously performed by the securer 30 (steps 49, 50, 52, 54, 60, 66 and 67) of
Fig. 1B are presently performed by agent 130 (steps 149, 150, 152, 154, 160, 166 and 168) of Fig. 3A. Additionally, instead of referring in step 49 of Fig. 1B to the SCDB 22, the present invention refers in step 149 of Fig. 3A to the LSCDB 122.

Referring now to Fig. 3B, which illustrates e-mail transfer with the participation of manager 112. User 1 uses e-mail package 20 and writes (step 40) an e-mail to user 2. User 1 refers to address book 18 and retrieves (step 42) the address of user 2. User 1 completes the e-mail. The e-mail package 20 puts (step 44) the e-mail in the outbox 24, and initiates (step 46) connection to the server 12.

The e-mail packet passes (step 48) to the communication protocol layer at the operating system 28 level. The interceptor hook observes that the packet is in the communication protocol layer, and provides (step 149) the packet to the mail security agent 130.

Mail security agent 130 tries to identify (step 150) the sender and the receiver, user 1 and user 2, respectively, as being a paired member of the LSCDB 122. Agent 130 does not recognize (step 155) user 1 and user 2 as paired members in the LSCDB 122. Agent 130 secures (step 156) the e-mail packet with a predefined set of properties for the secure channel between user 1 and manager 112, and sends the packet to manager 112. Manager 112 refers (step 157) to the MSCDB 114 and tries to identify user 2.

If user 2 is found (step 158) in the MSCDB 114 and identified, then manager 112 secures (step 159) the e-mail packet with a predefined set of properties for the secure channel between user 2 and manager 112. In parallel, the manager 112 generates (step 160) and sends a secured channel key for user
1 - user 2 to both user 1 and user 2. The generation and exchange of the keys allows the next e-mail transmission between user 1 - user 2 (or user 2 – user 1) to take place unaided by the manager 112.

Manager 112 sends (steps 161, 62, 64, 166, and 167) the packet onto user 2, where it is decrypted. The plain text e-mail packet is received (step 168) by Inbox 23 of user 2, which notifies user 2 of its existence. The plain text message does not contain any sign of the manager intervention. It resembles messages sent directly from User 1 to User 2 message, as described previously for Fig. 3A.

If user 2 is not found (step 170) in the MSCDB, then manager 112 follows (step 172) company policy as listed in database 129, which may include sending a non secured message, deferring the transfer until such a secured channel is created, discarding the message or returning an error message to user 1, etc.

Reference is now made to Fig 4A, a flow chart of an automatic authentication installation procedure for the users of system 100 in accordance with a preferred embodiment of the present invention.

System 100 creates a secured channel between at least the manager 112 and each user of system 100. As to be described hereinbelow in detail, the MSCDB 114 is imported and updated with at least the properties of the secured channel for manager 112 and each user of system 100. As additionally described in detail hereinbelow, the LSCDB 122 of each user is automatically installed with at least the properties of the secured channel for manager 112 and said user.
The installation procedures commences with installation (step 202) of the manager 112 on the system 100. The company, which has engaged system 100, selects the members/users of system 100 and imports (step 204) their names into the MSCDB 114. If, at a later date, the engaging company selects to include additional users onto system 100, this step can be repeated for any new users.

Manager 112 then commences the process of securing the members per each user that the manager's operator decide to secure. This process begins by generating (step 206) a Diffie Hellmen (DH) private and public keys for each user member. The flow chart in Fig. 4 details the distribution and generation for a single member, user 1; however, it is apparent that this procedure is repeated by manager 112 for all users imported in step 204.

Manager 112 saves the DH private key and sends (step 208) the DH public key to user 1, along with an installation program of agent 130. The installation software can be distributed in many various way, including as an attachment of an e-mail message. User 1 starts the installation program by activating the attachment (i.e. double clicking on the attachment), commencing (step 210) with installation of agent 130. Agent 130 generates (step 212) its own DH public and private keys. Agent 130 then generates (step 214) from its own private key and the received public key a whole DH key, and use it to generate the secured channel. Agent 130 automatically sends (step 216) (from user 1) to the manager 112, along the now secured channel, a secured "Installation Complete" message, preceded with its DH public key (which is sent in clear text). Manager 112 retrieves (step 218) User 1's DH public key and generates (step 220) from the
retrieved key and the saved DH private key the same whole DH key. This allows manager 112 to read and decrypt (step 222) the secured "Installation Complete" message. If the "Installation Complete" message is correctly decrypted, Manager 112 marks off (step 224) user 1 as being installed.

Reference is now made to Fig. 4B, a flow chart of the user authentication installation procedure for system 100 in accordance with a preferred embodiment of the present invention.

Instead of generating Diffie Hellman (DH) private and public keys, as described in Fig. 4A, the option is available for the company which have engaged system 100 secure one or more of the imported members by providing them with an Authentication Code, which can be any character string. This is done via the manager 112.

Steps 202 and 204 from Fig. 4A are repeated for this embodiment even though they might be avoided if they where already performed earlier. However, instead of generating the Diffie Hellman (DH) Private and Public keys, the manager 112 generates (step 250) a random Authentication Code, and provides (step 252) it to user 1. The manager 112 then uses the Authentication Code and creates (step 254) a key for the manager - user 1 secure channel. Manager 112 records (step 256) the properties of the secure channel for sending between user 1 and manager 112 in the MSCDB 114 and sends (step 258) the agent 130 software installation program to user 1, as an example, as an attachment of an e-mail message to user 1.

User 1 activates (step 262) the installation program which commences (step 264) with installation of agent 130. Agent 130 asks (step 266) for the
Authentication Code and uses it to generate (step 268) a key for the secured channel. Agent 130 automatically sends (step 270) (from User 1) to the manager 112, along the now secured channel, a secured "Installation Complete" message.

Manager 112 reads and decrypts (step 272) the secured "Installation Complete" message using the key saved in step 254. If the "Installation Complete" message is correctly decrypted, Manager 112 marks off (step 274) User 1 as being installed.

In a preferred embodiment of the present invention, an e-mail packet is secured by performing at least the following steps.

1) A digital signature is attached to the e-mail packet;

2) the e-mail packet and the digital signature are encrypted, using a random number as a key and the secure channel data encryption algorithm which can be any of the commercially available encryption algorithm, such as Data Encryption Standard (DES) or Blowfish;

3) the random (number) key is encrypted using a predefined key of a secure channel and the secure channel key encryption algorithm, which can be any of the commercially available encryption algorithm, such as Data Encryption Standard (DES) or Blowfish, and is attached to the encrypted packet;

4) the random key is also encrypted using the key for the User 1 – manager secure channel and the above chosen data encryption algorithm and is also attached to the encrypted packet; and
5) the now secured e-mail packet is encapsulated in a regular e-mail message, which displays only the minimal information needed for the transport.

Since system 100 provides manager 112 with the ability to decrypt the e-mail random number key, system 100 provides the flexibility of the following alternative options;

1. Users are able to receive secured mail at multiple locations;
2. damaged keys can be automatically repaired; and
3. aged keys can be automatically refreshed.

These three options are now going to be explained in detail.

**Multiple locations**

Fig. 5A shows a schematic diagram of system 100 with one or more users in a multiple location setup. As may occur from time to time, one or more users may log-on at one or more locations. As illustrated in the exemplary situation in Fig. 5A, user 2 is registered at two different locations; location A and location B. However, both locations of user 2 use the same server 12 and the same mailbox 13.

System 100 is aware that user 2 has two different log-on sites; however, the system does not know where user 2 will be the next time that he logs on. Moreover, it is not practical for Manager 130 to update LSCDB 122 for each site user 2 is using.

Once system 100 receives an “Installation Complete” message from user 2 with a new, or second, computer ID, manager 112 declares the user as a
multi-location user. Manager 112 provides user 2 at both locations with the same key for User 2 - manager secured channel. Additionally, manager 112 ceases to create secure channels for users 2 with other member users. Moreover, manager 112 also tries to ask users 2 to clear all their secure channels with other users and leave only the secure channel with the manager 112 active. In this manner, user 2 can always receive his e-mail, secured, through his secure channel with manager 112.

Reference is now made to Fig. 5B, a flow chart illustrating the e-mail process for Users with Multiple locations. Steps of this embodiment which are similar to those illustrated in Fig. 3A have been described hereinabove in detail, will not be further described.

In an exemplary situation, user 1 writes (step 302) e-mail to User 2. The e-mail packet is encrypted (step 304) to be delivered to User 2. However, there is no user 1- user 2 secured channel, and the e-mail is detoured (step 306) and sent to manager 112.

Manager 112 receives (step 314) the e-mail packet from User 1. Manager 112 then decrypts (step 316) the e-mail packet using his user 1-manager 112 key.

Manager 112 then re-encrypts (step 318) the opened e-mail packet with the properties of the user 2-manager secure channel, and sends (step 320) the now encrypted e-mail packet onto user 2. User 2, which has a key for manager 112, receives (322) and decrypts the packet.

Repairs of a damaged key
Reference is now made to Fig. 6, a flow chart illustrating the steps to be followed in repair of a damaged key.

User 1 sends (step 350) a secured e-mail packet to user 2. The e-mail packet is encrypted (step 352) to be delivered to user 2. User 2 receives (step 354) the e-mail however, user 2 cannot open (step 356) the packet. Either the key held by user 1 is damaged, or the key held by user 2 is damaged.

The agent 130 at user 2 sends (step 360) the unopened packet to manager 112, along with a message informing the manager that he (user 2) has mail he cannot open.

As noted in the description hereinabove, all e-mail packets are encrypted with a key for the user - manager 112 secured line. Thus, manager 112 received a key for the e-mail packet on the user 1 - Manager secured line in step 352, the original transfer of the e-mail packet. Hence, when manager 112 receives (step 362) the message and packet from user 2, manager 112 can decrypt (step 364) the packet using the user 1 - manager 112 key carried by the original messages.

Manager 112 re-encrypts (step 366) the opened e-mail packet with the properties of the user 2-manager secure channel and sends the now encrypted e-mail packet onto user 2. User 2, which has a key for the manager 112, receives (step 368) and decrypts the packet.

Manager 112 selects a random new key for (step 372) to user 1 and user 2. The exchange of the replacement keys allows the next e-mail transmission between user 1 and user 2 to take place unaided by the manager 112.
Refresh of an aged key

Reference is now made to Fig. 7, a flow chart illustrating the steps to be following in refreshing of an aged key. The principle is similar to that of repairing of the damaged key.

Manager 112 updates keys on a predetermined cycle, as an example, every three weeks. Upon the conclusion of the cycle, user 1 asks to refresh his key with user 2. Manager 112 updates (step 402) the user 1 – user 2 key by generating a new random key and sending it to the two users. Each new key is appointed with the manager's creation time, which is the manager's computer time of the key generation of the key. While user 1 receives the new key with the actual creation time, user 2 will receive a later creation time to reduce the probability that the two users will ask for key refresh at the same time, and thus create collision.

User 1 and user 2 can continue their communication without noticing the key refresh process being performed in the background. If user 1 receives the new key and uses it before user 2 received the new key, manager 112 still can recover the message.

In the exemplary situation of Fig. 7, user 1 receives (step 403) its new key. User 1 sends (step 404) a secured e-mail packet to user 2. The e-mail packet is encrypted (step 406) using the new key and delivered to user 2. User 2 receives (step 408) the e-mail, however, since user 2 does not have the updated key, he cannot open (step 410) the packet.

The agent 130 at user 2 recognizes that the message is secured using a new version of the key. User 2 sends (step 414) the unopened packet to
manager 112, along with a message informing the manager that he (User 2) has mail he cannot open because it is encrypted using a new version of the key.

When manager 112 receives (step 416) the message from User 2 about the e-mail packet from User 1, manager 112 decrypts (step 418) the packet using his user 1 – manager 112 key.

Manager 112 re-encrypts (step 420) the packet with the properties of the user 2-manager secure channel.

Manager 112 sends (step 424) the packet onto User 2. User 2 receives (step 426) and decrypts the packet.

If User 2 sends a message to User 1 using the old key, User 1 can still decrypt it. Each user preserves old keys in the LSCDB until the first message is decrypted with the new key.

It will be appreciated by persons skilled in the art that the present invention is not limited to what has been particularly shown and described hereinabove. Specifically, although the invention herein describes e-mail transfer, it is applicable for any type of electronic messages over digital networks used for communication applications. Rather the scope of the present invention is defined only by the claims which follow:
CLAIMS

1. A messaging system for transferring messages between any two of a plurality of user machines, the system comprising:

   a message manager for overseeing secure message operations,

   the message manager being associated with a management database storing separate secure channel properties for each message channel between said manager and each said user machine; and

   per user machine, a local secure message processor for securing messages, said processor being associated with a local database storing secure channel properties per channel between said user machine and each other user machine, including said manager, with which said user machine communicates.

2. A messaging system according to claim 1 and wherein said message manager installs said local secure message processor on at least one of said user machine and provides said associated local database with at least the secure channel properties of the channel between said manager and said user machine, thereby ensuring secure message operations from generally the moment of installation.

3. A messaging system according to claim 1 and wherein said local secure message processor includes means for sending messages to said message manager when the secure channel properties of a user to be
communicated with are unknown or when said local secure message processor cannot decrypt a message received from another user machine.

4. A messaging manager for overseeing secure message operations between any two of a plurality of user machines,

5. A message manager according to claim 4 and wherein said message manager is associated with a management database storing separate secure channel properties for each message channel between said manager and each said user machine.

6. A message manager according to claim 4; and wherein said message manager installs a local secure message processor and an associated local database on at least one of said user machine and provides said associated local database with at least the secure channel properties of the channel between said manager and said user machine, thereby ensuring secure message operations from generally the moment of installation.

7. A message manager according to claim 4 and wherein said secure manager recovers a message when the secure channel properties of a user to be communicated with are unknown or when said local secure message processor cannot decrypt a message received from another user machine.
8. A method of insuring the transfer of generally secure message between any two of a plurality of user machines, the method comprising the steps of:

   generating a secure channel properties between a message manager and each of said plurality of user machines;

   storing in a management data base said secure channel properties for each secure channel between said message manager and each of said plurality of user machine;

   installing a local secure message processor on at least one of said plurality of user machines and providing said processor with an associated local database;

   storing in said associated local database secure channel properties between said at least one plurality of user machines and at least said manager, and secure channel properties per channel for each said secure channel between each of said plurality of user machines and each other user machine; and

   securing messages between any of said plurality of user machines from generally the moment of installation.

9. A method according to claim 8 and comprising the steps of:

   logging on at more than one said plurality of user machines; and

   transferring generally secure message by sending messages to said message manager.
10. A method according to claim 8 and whereas generating a secure channel properties comprises employing an automatic authentication with an automatic key generation.

11. A method according to claim 8 and whereas generating a secure channel properties comprises employing an authentication code.

12. A method according to claim 8 and comprising the steps of sending messages to said message manager when the secure channel properties of a user to be communicated with are unknown or when said local secure message processor cannot decrypt an message packet received from another user machine.

13. A method according to claim 8 and comprising the step of regenerating said secure channel properties for said secure channel between said any two of a plurality of user machines without interrupting user communication flow.

14. A method according to claim 13 wherein regenerating said secure channel properties includes regenerating said secure channel properties every predetermined time period.

15. A method according to claim 13 wherein regenerating said secure channel properties includes regenerating said secure channel properties when said local secure message processor cannot decrypt an message packet received from another user machine.
User 1 writes e-mail to user 2

Retrieve address from address book 18

Send to out box 24

Initiates connection to server 12

e-mail sent to TCP socket

Intercepted and provided to securer 30

Securer 30 refers to SCDB 22 and tries to identify users

Identified - Secured

Keyed, signed, encrypted

No ID - Non-secure

Send warning to user 1

Send non-secure

Don't send

Returned to operating system 28

Sent over Internet

Waits in user 2 mail box 13 until retrieval

Intercepted and provided to securer 30

Decrypted

Received by inbox 23 of user 2

FIG. 1B
Prior Art
4/12

User 1 writes e-mail to user 2

Retrieve address from address book 18

Send to out box 24

Initiates connection to server 12

e-mail sent to TCP socket

Intercepted and provided to agent 130

Agent 130 refers to LSCDB 122 and tries to identify users

Identified - Secured

Signed, encrypted

Returned to operating system 128

Sent over Internet

Waits in user 2 mail box 13 until retrieval

Intercepted and provided to agent 130

Decrypted

Received and decoded by user 2

FIG. 3A
5/12

User 1 writes e-mail to user 2

Retrieve Address from address book 18

Send to out box 24

Initiates connection to server 12

e-mail sent to TCP socket

Intercepted and provided to agent 130

Agent 130 refers to LSCDB 122 and tries to identify users

Agent 130 does not recognize users

Secures and sent to manager 112

Manager refers to MSCDB 114

Identifies user 2

Secures e-mail for user 2

FIG. 3B
6/12

160
Creates keys and sends to user 1 and user 2

161
Returns e-mail to operating system 128

62
Sent over Internet

64
Waits in user 2 mail box 13 until retrieval

166
Intercepted and provided to agent 130

167
Decrypted

168
Received by inbox 23 of user 2

170
User 2 not identified

172
Per company policy; discard, return to user or other

FIG. 3B cont.
Install manager 112

Import users

Generate DH public and private keys

Send DH private key and agent 130 installation program to user 1

Start installation of agent 130 at user 1

Agent 130 generates DH private and public keys at user 1

User 1 generates whole DH key

Send "installation complete" message and DH public key to manager 112

Manager 112 retrieves DH public key

Manager 112 generates whole DH key

Decrypt "installation complete" message

Manager 112 marks user 1 as installed

FIG. 4A
8/12

202
Install manager 112

204
Import users

220
Manager 112 generates authentication code

252
Manager 112 provides authentication code to user 1

254
Manager 112 creates user 1-manager 112 key

256
Manager 112 records properties

258
Sends installation program for agent 130 to user 1

262
User 1 activates program

264
Starts installation program

266
Agent 130 asks for authentication code

268
Agent 130 generates key

270
Agent 130 sends "installation complete" message to manager 112

272
Manager 112 reads and decrypts

274
User 1 is marked as installed

FIG. 4B
10/12

302
User 1 writes e-mail to User 2

304
Mail is encrypted to user 2

306
No user 1 - user 2 channel; detoured to manager 112

314
Manager 112 receives e-mail from user

316
Manager 112 encrypts e-mail

318
Re-encrypts message for User 2

320
Sends to User 2

322
User 2 receives and opens

Multiple locations

FIG. 5B
11/12

User 1 writes e-mail to user 2

Mail is encrypted to user 2 and includes a manager 112 - user 1 key

Mail received by User 2

User 2 unable to open

Sends message and unopened packet to manager 112

Manager 112 receives unopened packet

Manager 112 uses the manager 112 - user 1 key to open

Re-encrypts the packet and sends to user 2

User 2 receives and opens

Manager 112 selects a new random key for user 1 and 2

FIG. 6
12/12

User 1 needs an updated key

Manager 112 updates user 1- user 2 key

User 1 receives new key

User 1 writes e-mail to user 2

Mail is encrypted with new key

Mail received by user 2

User 2 cannot open

Sends unopened packet and message to manager 112

Manager 112 retrieves

Manager 112 decrypts the packet

Manager 112 re-encrypts for user 2

Manager 112 adds to e-mail new user 1 key

Sends to user 2

User 2 receives and decrypts the packet

FIG. 7