(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 August 2002 (01.08.2002)

PCT

(10) International Publication Number
WO 02/059713 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number: PCT/US01/47615

(22) International Filing Date:
7 November 2001 (07.11.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/246,420 7 November 2000 (07.11.2000) US

(71) Applicant *(for all designated States except US)*: **ASPSECURE CORPORATION** [US/US]; dba TrustData Solutions, 146 Clover Way, Los Gatos, CA 95032 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CATO, Miles, A.** [US/US]; 1808 Pacific Avenue, #304, San Francisco, CA 94019 (US). **MURRAY, Timothy, J.** [US/US]; 12420 Sierra Rojo, Valley Center, CA 92082 (US). **HALLFORD, Catherine, M.** [US/US]; 4818 Highlands Drive, McKinney, TX 75070 (US). **NEAL, Dwight, E.** [US/US]; 1701 Park Central Dr. #1008, McKinney, TX 75069 (US). **PATTERSON, Eldon, R.** [US/US]; 741 FM 2194, Farmersville, TX 75442 (US). **WILLIAMS, Dana, N.** [US/US]; 725 Lakeview Dr., Lucas, TX 75002 (US). **SUDDUTH, Tammy, W.** [US/US]; 319 Fountain Gate Dr., Allen, TX 75002 (US). **WEBER, Robert, P.** [US/US]; 215 Waverley Street, #4, Menlo Park, CA 94025 (US).

(74) Common Representative: **CATO, Miles, A.**; 1808 Pacific Avenue, #304, San Francisco, CA 94019 (US).

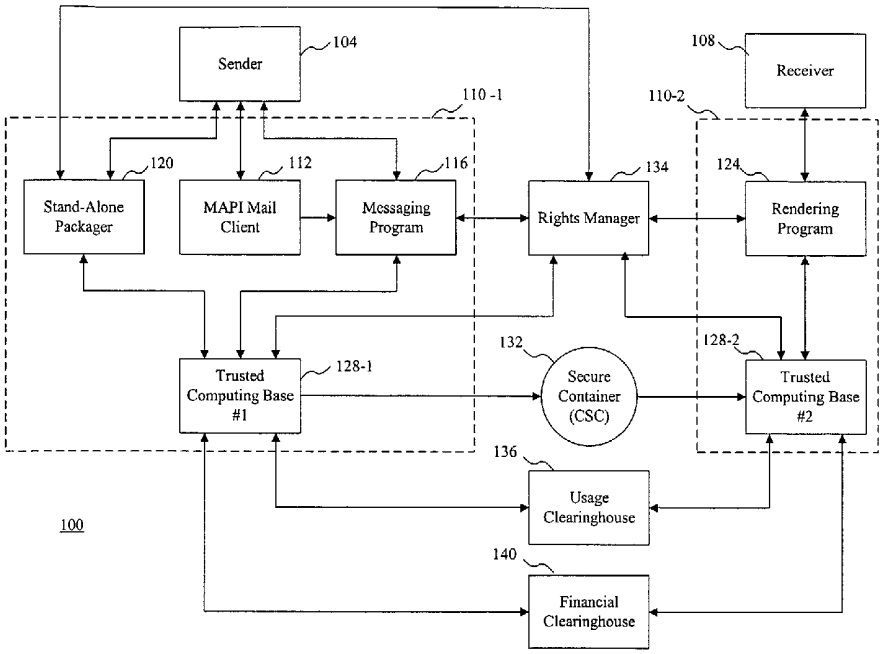
(81) Designated States *(national)*: CA, CN, JP, KR, SG, US.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: METHODS FOR TRUSTED MESSAGING



(57) Abstract: A method for securing electronic messages, comprises of (i) a packager for packaging an electronic message prepared by a sender, (ii) a first trusted computing base for securing and passing said packaged electronic message in accordance with specified rules; (iii) a second trusted computing base for receiving and decoding said secured electronic message in accordance with said rules; and (iv) a rendering program for manipulating said decoded electronic message in accordance with said rules.

WO 02/059713 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Specification

METHODS FOR TRUSTED MESSAGING

PRIORITY CLAIM

This application claims priority to a provisional application entitled "Trusted Messaging" filed on November 7, 2000, having an application number 60/246,420.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention generally relates to methods for exchanging electronic messages and, in particular, methods for the secure and trusted exchange of electronic messages and of electronic rules associated with said electronic messages.

Description of the Prior Art

Email is rapidly becoming a ubiquitous form of communication and may represent one of the, if not the fastest technology adoption trend in history. In order to stay competitive, every organization must adapt to the email explosion while simultaneously protecting themselves from the risks of open communications. Example risks include the unauthorized access to and/or use of email systems and message content, stolen or misappropriated individual and/or organizational identities, and the unauthorized modification of email messages including attachments. Governments have similar concerns and may have more stringent security and trust requirements in many instances.

Given the increasing ubiquity of electronic messaging and email, individuals share similar concerns and risks as well. For example, individuals may wish to preserve the privacy or confidentiality of their email messages and attachments and may wish to ensure that their electronic identities are not stolen or otherwise misused.

Though email is fast, cost-efficient, and convenient, as typically implemented, email may expose enterprises and individuals to extreme risk. Some of these risks include: 1) the inability to control email messages and attachments after transmission, 2) the loss of communications privacy and trust, and 3) increased corporate and individual liability from uncontrolled digital communications. The inability to control message content, for example, may lead to the disclosure of sensitive personal, corporate, and/or governmental information. Email users may not trust messages when they cannot be certain of the integrity of message content and/or the identity of the sender or receiver. Lack of message control, therefore, may lead to increased liability, for example, when individual healthcare or financial records are received and read by parties lacking the authority to do so.

More specifically, referring to Fig. 1, a scenario illustrating present day email communication is illustrated. Here, the sender 10 wishes to communicate via email a private message 12 to an intended receiver 14 (or group) and not beyond, and wishes to be assured that the message only be used in a manner specified. The private message travels from a first mail server 16 to the internet and to the mail server 20 of the intended receiver. However, referring to Fig. 2, the intended receiver can use the private message in ways unintended by the sender. For example, the intended receiver may distribute electronic or print copies of the message, reply to the email message and including unintended receivers, or forwarding the message to unintended or unauthorized receivers. Furthermore, referring to Fig. 3, the privacy of the email message can be compromised by its context alone (i.e., subject / sender identity). For example, in an open work environment 24, unintended personnel can view the email message just from the sender or subject line information 26 presented on the screen to glean private and confidential information. Also, referring to Fig. 4, an email message could be unintentionally sent to an unintended receiver with a mistake in the email address; valuable company, governmental, and/or individual information could be lost and/or greatly compromised. The lack of predictability, the lack of certainty regarding email messages, their senders, recipients, contents, attachments, and/or intermediaries remain a concern for individuals and organizations. All of these problems associated with unsecured and untrustable email messages provide the impetus and desire for a secure and trusted email system.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide methods for protecting electronic messages from unauthorized access and use.

It is another object of the present invention to provide methods for governing the transport of protected electronic messages.

It is another object of the present invention to provide methods for authorized access of electronic messages in accordance with rules associated with said messages.

It is another object of the present invention to provide methods for managing the rights to use a packager for securing electronic messages and a rendering program for decoding and manipulating electronic messages.

It is another object of the present invention to control access to subject line information in accordance with rules associated with said subject line information.

It is still another object of the present invention to provide methods for controlling the viewing, the forwarding, or the replying of an electronic message.

Briefly, in a presently preferred embodiments of the present invention, a method for securing electronic messages, comprising of (i) a packager for packaging an electronic message prepared by a sender; (ii) a first trusted computing base for securing and passing said packaged electronic message in accordance with specified rules; (iii) a second trusted computing base for receiving and decoding said

secured electronic message in accordance with said rules; and (iv) a rendering program for manipulating said decoded electronic message in accordance with said rules.

An advantage of the present invention is that it provides methods for securely transporting electronic messages.

Another advantage of the present invention is that it provides methods for managing the rights to use a packager for securing electronic messages and a rendering program for decoding and manipulating electronic messages.

Still another advantage of the present invention is that it provides methods for controlling the viewing, the forwarding, or the replying of an electronic message.

Still another advantage of the present invention is that it provides methods for controlling the viewing of message subject line information.

Note that the current invention transitions common email from a public chat to a private dialogue. An email sender can protect selected email messages and attachments by specifying rules that a recipient must comply with in order to access the message and attachments. Messages and attachments are packaged into secure containers along with rules governing their use by an authorized messaging program. The secure containers are transmitted using normal email channels, however upon receipt the body and attachments can only be accessed when using an authorized rendering program and handled by the recipient in a manner specified by the sender. For example, if the sender specifies a rule that states a recipient can only view the body and attachments of an email, then the recipient is prohibited by the trusted email rendering program from cutting or copying the message or from printing it. The system provides an approach to managing the trusted authorization of message packaging and rendering programs. Furthermore, the system provides a tracking function that can monitor and report the status of email transmissions. The system does not require that a company or individual scrap their current email system. Secure email seamlessly integrates into existing mail clients and requires no special processing by backend mail servers. This allows the existing email network becomes the foundation to integrate secure communications process and persistent content protection into email communications, inside and outside an organization.

The rule-based nature of secure email can implement virtually any privacy policy as required by governments, contracts, markets or end users. Rules attached to e-email protect the rights to the intellectual property and ideas contained in email messages and attachments. At the time the email is generated, access and usage rules, as defined by the sender, are attached to the email. The sender can control an extensible list that defines things like how and when the email and attachments can be used, e.g., a sender can provide direct control over intended uses that may include viewing, printing, copying, editing and duplication. Trusted email can maximize the efficiency of email communications while providing the full confidence that intellectual property; confidential communication, and financial information are exchanged and shared with minimum risk.

Regardless of where the secure emailand attachments are sent, the contents remain protected and the systems persistently controls access to and use of email messages, both on-line and off-line. Even after the message is delivered, and the recipients have downloaded and opened the email on their local machines, the email is still governed by rule set by the sender.

Secure email has a built in audit trail that can monitor and report the status of email transmissions. This audit trail could be used to verify that a message was delivered, in authentic and unaltered form. The audit function can also determine and report the fact that a message was opened at the destination. The trusted email audit features are significant in those circumstances where message delivery status and acceptance may be required. The audit trail provides a full-featured protection process in case an intended recipient repudiates message delivery.

Note that the present invention relates to exchanging electronic messages in a cryptographically secure container (non-limiting examples of which are disclosed US Pat. No. 5,715,403 issued to Stefik on 2/03/1998; US Pat. No. 5,638,443 issued to Stefik et al. on 6/10/1997; US Pat. No. 5,634,012 issued to Stefik et al. on 5/27/1997; US Pat. No. 5,629,980 issued to Stefik et al. on 5/13/1997; and, US Pat. No. 5,845,281 issued to Benson et al. on 12/1/1998; which are incorporated herein by reference). One embodiment that provides offline availability for secured information and offline evaluation and/or enforcement of rules associated with the secured information, is disclosed in US patents issued to InterTrust™ ("the InterTrust patents") incorporated by reference herein and includes the following patents: U.S. Patent No. 6,185,683 to Ginter, et al.; U.S. Patent No. 6,138,119 to Hall et al.; U.S. Patent No. 6,112,181 to Shear et al.; U.S. Patent No. 5,982,891 to Ginter et al.; U.S. Patent No. 5,949,876 to Ginter et al.; U.S. Patent No. 5,920,861 to Hall et al.; U.S. Patent No. 5,917,912 to Ginter et al.; U.S. Patent No. 5,915,019 to Ginter et al.; U.S. Patent No. 5,910,987 to Ginter et al.; U.S. Patent No. 5,892,900 to Ginter et al.; and pending application WIPO Publication WO9,810,381 A1 to Shear et al.

The processes described herein may also be compatible with applications based on and/or interacting with software and/or hardware instantiations of the Common Security Data Architecture, its APIs and services, as described, for example, in CDSA Explained, ISBN 1-85912-231-0, Document Number 901, UK: The Open Group, January 1999; Common Security: CDSA and CSSM, Version 2 (with corrigenda) ISBN: 1-85812-202-7, Document Number C914, UK: The Open Group, May 2000, presently available at: http://www.opengroup.org/pubs/catalog/c914.htm#medium3.

The present invention also relates to electronic rules securely associated with electronic messages and means for enforcing those rules whenever and wherever access and/or use of protected information is requested.

These and other features and advantages of the present invention will become well understood upon examining the figures and reading the following detailed description of the invention.

IN THE DRAWINGS

Fig. 1 is an illustration of a prior art process for sending email from a creator/sender to a receiver(s);

Fig. 2 is an illustration of a prior art process where an unintended receiver reads forwarded email;

Fig. 3 is an illustration of a prior art process where portions of the email message are compromised in an open work environment;

Fig. 4 is an illustration of a prior art process where the email is mistakenly addressed for the wrong person;

Fig. 5 is a block diagram of an embodiment of a messaging system; and

Fig. 6 is a block architecture diagram of an embodiment of a trusted messaging system client.

Fig. 7 is a block diagram of an embodiment for creating a secured email message;

Fig. 8 is a block diagram of an embodiment for viewing a secured email message;

Fig. 9 is a screen shot of an embodiment of a settings panel that includes profiles, intents and message field replacement; and

Fig. 10 is a block diagram of a rule selection process.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In a presently preferred embodiments of the present invention, an non-limiting example embodiment of a messaging system 100 is shown in Fig. 5. A sender 104 and receiver 108 have a trusted computing base ("TCB") 128 associated with each of their computers, said TCB 128 having a software tamper resistant secure execution facility known to those skilled in the arts. A TCB 128 may rely in part on hardware tamper resistant techniques known to those skilled in the arts. One example of a TCB 128 is disclosed in US patent No. 5,715,403 issued to Stefik on 2/03/1998; US Patent No. 5,638,443 issued to Stefik et al. on 6/10/1997; US Pat. No. 5,634,012 issued to Stefik et al. on 5/27/1997; and, US Pat. No. 5,629,980 issued to Stefik et al. on 5/13/1997 which are incorporated herein by reference. Another example is disclosed in US Pat. No. 5,845,281 issued to Benson et al. on 12/1/1998 which is also incorporated herein by reference.

One embodiment of TCB 128 is an "InterRights Point" software installation that includes a protected processing environment as disclosed in the InterTrust patents. In a related embodiment, the TCB functions are more fully integrated with the operating system, one example of which is the Virtual Distribution Environment (VDE) disclosed in the InterTrust patents. Using a messaging program 116, the sender 104 composes a message, which may include one or plural attachments. The messaging program 116 formats and passes the message to the TCB 128-1 using the application program interface (API) of the TCB 128. Some embodiments could gather some message and/or packager information from a MAPI mail client 112 or a stand-alone packager application program 120. In some embodiments the messaging program 116, MAPI mail client 112, and/or stand-alone packager program 120 would operate together with TCB 128-1 on a common computer/appliance 110-1. In other embodiments, the

messaging program 116, MAPI mail client 112, and/or stand-alone packager would be distributed on separate computer/appliance and interact remotely with TCB 128-1.

The messaging program 116 includes a packager routine and rules compiler routine. The packager routine gathers the message and any attachments and sends them to the API of the TCB 128-1. The rules associated with the packaged message and/or attachments are compiled and passed to the TCB 128-1 for inclusion in a cryptographically secure container ("CSC") 132. The CSC protects the message and attachments during transport through the Internet, private networks, including corporate networks ("intranets") and private networks among value chain parts, often referred to as "extranets," for example.

Once in the CSC 132, the message is passed to the receiving party 108. With the assistance of a rendering program 124, the TCB 128-2 of the receiver decodes the CSC 132 and controls subsequent access to the message and/or attachments. The rendering program 124 allows the receiver 108 to view and manipulate the message and any attachments in accordance with the rules provided by the sender 104. In some embodiments the rendering program 124 would operate together with TCB 128-2 on a common computer/appliance 110-2. 108. In other embodiments, the rendering program 124 would be distributed on a separate computer/appliance and interact remotely with TCB 128-2.

The messaging program 116 and/or the stand-alone packager 120 is authorized to exchange messages with the rendering program 124. Authorization to exchange messages is based in part on authorization to use the TCB 128 and the presence of a protected digital data structure(s) stored within TCB 128. One or more protected digital data structures authorizes messaging program 116 and/or the stand-alone packager 120 to allow sender 104 according to rules to encode messages and attachments within a CSC 132 and to include within the CSC 132 a protected digital control mechanism that allows an authorized rendering program 124 to authenticate that the receiver 108 of the message is the intended recipient. Similarly, the presence of one or more protected digital structures stored within TCB 128 authorizes the rendering program 120 using the protected digital control mechanism transported within CSC 132 to only decode and render messages that are intended for receiver 108.

In one embodiment authorization to exchange messages is provided by a centralized rights manager program ("RM") 134. RM 134 may create, modify, store, and distribute protected digital data structures used by TCBs to manage rights. Non-limiting examples of rights managed within a protected digital data structure include authorizations, membership cards, budgets, and persistent ownership rights. In this example, a user authorized to use TCB 128, must also be authorized by the RM 134 to use messaging program 116, stand-alone packager 120, and/or rendering program 124. In some embodiments the RM 134 may be a stand-alone program or in other embodiments the functions of RM 134 may be incorporated via API's or other suitable mechanisms into existing centralized authorization systems such as Novell's directory services products, or other directory systems using enterprise directory system (e.g. x.500), or Lightweight Directory Access Protocol (LDAP). The entity that controls the RM 134 may

provide certain policies to authenticate the identity of the user including their email address. Non-limiting examples of the policies used by the controlling entity may include procedures that incorporate x.509 certificates or other digital certificate authority schemes. Some embodiments of the RM may include API's to facilitate this access to 3rd party systems that provide certificate and key management services. The RM 134 controlling entity is also responsible for causing the RM 134 to distribute securely a protected digital data structure to TCB 128, which grants authorization to use the messaging program 116, the stand-alone packager 120, and/or the rendering program 124.

In another embodiment authorization to exchange messages is provided in an initial peer-to-peer exchange of authorization information between the messaging program 116 and the rendering program 124. In this example, a sender 104 and a receiver 108 must first be authorized to use TCB 128-1 and TCB 128-2 respectively, the sender 104 and the receiver 108 securely registers certain identifying information about themselves including their email address to TCB 128-1 and TCB 128-2 respectively, the sender 104 then invites the receiver 108 to exchange messages. Authorization is passed within a protected digital data structure inside a CSC 132 from the receiver 108 back to the sender 104.

In another embodiment, at least some rules and/or associated "objects" or files related to the message may be sent in a CSC to receiver 108 by a third party. Non-limiting examples of said third parties include providers of financial clearing and/or processing services. Non-limiting examples of the financial clearing rules and/or associated information include one or more instantiations of electronic money, payment, stored value, credit, notational currency, digital cash, and other currency and payment related electronic files or "objects". An example financial clearinghouse 140 may send a rule or payment object in a CSC 132 to the message receiver 108. In one embodiment, the TCB 128 includes means for receiving, assessing the integrity and authorized source of the payment-related information, and as necessary, associating the payment information with rules provided by sender 104 if payment for access and/or use of the message and/or one or plural attachments is required by sender's 104 rules.

The sender 104 can choose rules to regulate access to the message and attachments by the receiver 108 and/or other parties. The messaging program 116 gathers the rules and passes them to the TCB 128 for inclusion with the CSC 132. These rules are persistent, such that the receiver 108 or other parties cannot violate the rules at any time in the future. For example, a rule may prevent the receiver 108 from forwarding, printing, copying or saving the message such that another recipient who is not authorized by sender 104 is prevented from accessing and/or using the message and one or plural attachments. Some embodiments can sunset and sunrise the rules to make them applicable during a time period.

The rules are embodied in intents or verbs that the TCB 128 applies to the message and/or attachments inside the CSC 132. Non-limiting example embodiments of verbs or intents are disclosed in US Pat. No. 5,715,403 issued to Stefik on 2/03/1998; US Pat. No. 5,638,443 issued to Stefik et al. on 6/10/1997; US Pat. No. 5,634,012 issued to Stefik et al. on 5/27/1997; and US Pat. No. 5,629,980 issued

to Stefik et al. on 5/13/1997. Other intent or verb embodiments are disclosed in the InterTrust patents. For example, a "save" intent could be specified to allow the receiver 108 to save an attachment locally outside the protection of the TCB and any CSCs. Once saved, the control of the unprotected saved version is lost.

A usage clearinghouse 136 audits a message as it is packaged and sent by sender 104 and/or received and accessed by receiver 108. Information from the TCB 128 of the sender and receiver is available to the clearinghouse 136. This audit information can be used to check compliance with the rules, and as a foundation for value-added services, such as "certified delivery", "non-repudiation", and other value-added services.

In another embodiment, the messaging program accesses the facilities of the TCB through abstraction layer as shown in Figure 6. The abstraction layer provides an API consisting of messaging related functions, non-limiting examples of which include "package this message", "package this attachment" "package these rules", and any other trusted functions and/or capabilities provided by messaging program 116. As illustrated in Figure 6, computer or appliance is a hardware context on which operating systems and applications are run. Non-limiting examples of computers or appliances include PCs, Macs, Windows CE machines, cell phones, personal digital assistants, game machines, digital cable boxes, WebTV, and other devices. The operating system controls the hardware and peripheral devices and provides services to and a software execution context for applications and other software. Operating systems include various Microsoft Windows versions including XP, MAC OS, Linux, Unix, and other operating systems. In one embodiment, the TCB 212 may be thought of as a middleware operating system extension that provides security and trust related services to applications that use its services which are accessed through a TCB API 216.

Conventional mail clients make no use of TCB services, for example a first mail client 224-1 interfaces with the OS 208 exclusively. Figure 6 also shows that the trusted messaging system may be integrated with any number of mail clients. In one embodiment, a second mail client 224-2 may obtain services from the TCB through the TCB API 216. However, in some embodiments many TCB API calls are required to accomplish common and repetitive tasks, such as packaging rules, messages, and/or attachments in one or more CSCs. A third and n^{th} mail clients 224-3, 224-n are enabled to make use of TCB services through a trusted mail client abstraction layer API 220, which is in contrast to the second mail client 224-2 that makes use of the TCB services directly through the TCB-API 216. The first mail client 224-1 makes no use of the TCB 212.

Figure 6 also shows that a mail client may present its information and services to a user through a graphical user interface (GUI) 232 that may vary from mail-client-to-mail-client. In addition, there may be a need to "localize" the presentation for different languages using a localization method, such as files that contain certain presentation information in the language of choice. The information in these localization files is then used by the mail client GUI interface 232.

One benefit of the abstraction layer API 220 is that it makes the task of creating an interface between a mail client 224 and TCB services 212 much easier because higher level functional calls to abstraction layer API 220 may hide the underlying complexity of multiple TCB API 216 calls per function. The programmer then is free to concentrate on the particular issues in interfacing with the mail client at hand. An advantage of the present invention(s) is that programmers who are already familiar with mail clients are less likely to have to learn the details and complexities of the TCB API 216. In addition, enhancement, modification, bug fixing are enhanced since in many cases one need only change the abstraction layer 220 once to provide these improvements to users of many different mail clients 224, assuming that the abstraction layer calls themselves do not require alteration.

Solutions to Contextual Problems

In the implementation of the preferred embodiments, a number of issues are presented and resolved including the resolution of contextual problems. More specifically, the description provided below provide methods used by messaging program 116, stand-alone packager 120 and/or rendering program 124 to to resolve various issues related to protecting electronic messages from unauthorized access and use .

Solutions to Contextual Problems – Providing for auditable email path tracking, credentialed and trusted email applications, and email clearinghouse services

The chain of repackaging authority for the contents of an email message can be traced. This is to be accomplished by associating two attributes for each content object (or some other suitable object) within a message. Each repackaging of the contents of a message by messaging program 116 and/or by the stand-alone packager 120 will causes an instantiation of one or more content objects. The first attribute will be assigned a value containing a unique Object ID for the latest instantiation of the content object. The second attribute will be assigned the full parentage (each Object ID which has included it in this path) of all prior instantiations. In the case of an object that is a re-packaging of previously packaged content, the newly packaged content's second attribute will get the full Object ID set from the prior packaging's second attribute (the second attribute's value) as well as the Object ID of that previously packaged content (the first attribute's value). Repackaging authority, within messaging program 116 and/or the stand alone client 120, is handled by a custom intent, that is, by a verb or controlled action that is not standard verb of, and/or known to a TCB, but which in the preferred TCB embodiment has been in part programmed and certified by the provider of the preferred embodiment TCB.

One or more application credentials are relied upon for the security of the messaging program 116, the stand-alone packager 120, and the rendering program 124 applications In order to ensure the integrity of the said applications using a TCB instance, the provider of the TCB may evaluate the security, trustedness, and/or functionality of a proposed trusted application, and upon determining that

the application is secure, trustable, and performs certain controlled actions in accordance with the TCB verb(s) or intent(s), the TCB provider associates with the certifiable application one or more trusted credential such as a "digital certificate" or signed object. This certificate or signed object is referred to as an "application credential" whose presence and/or validity may be checked by the TCB at least one or more times as the certified or credentialed application uses capabilities of the TCB through the TCB API 216 or through an abstraction layer API comprised at least in part of higher level or more holistic function call. In turn, the abstraction layer uses one or more TCB API 216 calls to accomplish the higher level tasks. In one embodiment, the application credential(s) associated with messaging program 116 and/or stand-alone packager 120 are known in a trusted and protected manner by rendering program 124. The application credential(s) associated with the certified messaging program 116 and/or the stand-alone packager 120 are stored in each CSC prepared by said programs so that the rendering program 124 can readily tell CSCs prepared by said programs. In the presently preferred embodiment the rendering program 124 will only be process those CSCs created by messaging program 116 and/or stand-alone packager 120. Similarly, other 3rd-party CSC rendering programs are not to process CSCs created by messaging program 116 and/or stand-alone packager 120. In the presently preferred embodiment, one non-limiting example of a CSC is an InterTrust™ DigiBox secure container.

A clearinghouse needs to be involved for processing and reporting the audit trail, which in the presently preferred embodiment is at least in part described in the aforementioned WIPO publication WO9,810,381, Shear, et al. Audit/usage records are created at the time the email is secured within the CSC by the TCB (see Fig. 7) and again when the email is first viewed (the CSC is opened and content is presented after the authorization rules are met) (see Fig. 8). At the sender's option, an audit record can be produced each time the email or attached files are viewed, printed, or saved.

Solutions to Contextual Problems – Providing for email Subject Line replacement

The real subject line of an email message is saved in a protected fashion and replaced by a non-informative subject line during public transport, after receipt by the intended recipient the original subject line is restored In one embodiment, this is accomplished in messaging program 116 and/or stand-alone packager 120 by providing the sender 104 with a dialog (something similar to Fig. 9) to specify the non-informative subject line to use. In some embodiments the non-informative text may be stored as a default subject line available to sender 104 each time stand-alone packager 120 and/or messaging program 116 is used. The messaging program 116 and/or the stand-alone packager 120, creates a replacement message that contains the non-informative subject line along with an attached CSC file that contains the original subject line along with other components of the message. The replacement message is then submitted by messaging program 116 and/or stand-alone packager 120 to sender 104 email client via some standard interface, for example the Messaging Application Program Interface (MAPI) proposed by Microsoft™ and now implemented by many email system vendors or some other suitable interface. It is noted that

MAPI includes functions for harvesting the real subject line of a message, thus an advantage of subject line replacement is to protect the sender 104 and receiver 108 from unwanted tracking of the subject of exchanged messages. Upon receipt of the replacement message by receiver 108, the rendering program 124, according to rules decodes the message in the CSC and the real subject line is made available by replacing the default subject line in the rendered message.

Solutions to Contextual Problems – Providing for Sender Name/Address Replacement

The real sender's name and address is saved in a protected fashion and replaced by a non-informative name and/or address during public transport, after receipt by the intended recipient the original name and address is restored. In one embodiment, this is accomplished in messaging program 116 and/or stand-alone packager 120 by providing the sender 104 with a dialog (something similar to Fig. 9) to specify the non-informative name and address to use.

How to modify the name and email address for the "sender" varies somewhat by email client. Willful replacement of sender information with incorrect or invalid sender information is known as spoofing, and may at some point be considered illegal or subject to control by some governmental bodies. Conventional mail clients allow changing the sender information for all messages through a manual process. For example, the name and email address of the sender is established via a properties panel in Microsoft™ Outlook 2000™. The panel for Outlook 2000 is available via "Tools"/"Services"/"Properties" where the User Information section is changed to supply a different name and email Address. In one embodiment, it is stored in the Registry in entry "Office\Outlook\OMI Account Manager\Accounts\00000001".

In one embodiment, this is accomplished in messaging program 116 and/or stand-alone packager 120 by providing the sender 104 with a dialog (something similar to Fig. 9) to specify the non-informative name and address to use. The messaging program 116 and/or the stand-alone packager 120, creates a replacement message that contains the non-informative name and address along with an attached CSC file that contains the original name address and along with other components of the message. The replacement message is then submitted by messaging program 116 and/or stand-alone packager 120 to sender 104 email client via some standard interface, for example the Messaging Application Program Interface (MAPI) proposed by Microsoft™ and now implemented by many email system vendors or some other suitable interface. Upon receipt of the replacement message by receiver 108, the rendering program 124, according to rules decodes the message in the CSC and the real subject name and address is made available by replacing the spoofed name and address in the rendered message.

Solutions to Contextual Problems - Other Fields

Other fields in the message header could be replaced by default values. The rendering application in conjunction with the TCB would replace the default values with the real values. For

example, for a message sent to many receivers could have the other receivers suppressed until viewing such that viewing the message during transport would only reveal one of the receivers. In one example, a medical specialist's doctor's office may send a trusted email to a patient and choose to mask who the sender is less so that some interloper cannot infer from the sender what medical condition(s) the patient might have.

Solutions to Contextual Problems - Rule profiles (named collections of rules)

A rule profile is a named collection of rules, specified by a sender 104 or established by some other enterprise, governmental or institutional body and selected by sender 104, for controlling how a receiver 108 may use an email. Selection of one of the predetermined groups of rules applies those rules to the message more efficiently and consistently. Rules specify how and when an email may be used. Rules include but are not limited to specification for the handling of messages that allow or disallow for printing, viewing, saving into the clear (unencrypted), as well as a time range when such actions are valid, so called "sunrise" and "sunset" dates, and the consequences of these actions, e.g. auditing or financial payment,. For example, a patient profile may allow viewing a note from the attending physician, but not printing the note, the note is available for the next 10 days, and if the note is view an audit record is created.

In one embodiment, sender 104 provides a list of e-mail recipients to messaging program 116 and/or stand-alone packager 120 and selects a rule profile. Messaging program 116 and/or stand-alone packager 120 processes the recipient list and applies the rules contained in the selected rules profile to message packaged for each recipient.

Rule profiles are stored in Extensible Markup Language (XML) (non-limiting example details of which are provided in Appendices 1, 2, and 3) anticipating that the set of rules/options covered in a profile will likely grow over time (see Figs. 9 and 10). XML is defined via http://www.w3.org/XML/ as a vendor-independent standard. Microsoft extensions may also be used in the implementation and they are documented at http://msdn.microsoft.com/XML.

Referring to Appendix 1, the first line identifies this as an Email Settings XML and identifies the namespace to use (for datatypes). The second line begins the definition of the "Individual" settings, which in one embodiment indicates which of a possible hierarchy of rules profiles the present instance belongs. Rules profiles might reflect any class distinction(s), one example of which might be a hierarchical Corporate/Department/Individual settings of options in a rules profile. In this example, Corporate settings would overrule Department settings which would overrule Individual settings. Distinctions would be made between "defaults" being specified at a higher level vs. a required value being specified at a higher level. For example, Corporate settings in a given profile might default secured emails to expire (sunset) after 1 year while requiring that those emails NOT have SAVE intent enabled.

This would allow Individuals to set their corresponding profile to a different sunset date but would NOT allow them to enable the SAVE intent. The third line describes the "Ind_Print" option attribute. It is labeled "Ind_Print" anticipating the hierarchy of options described above to allow for "Corp_Print" and "Dept_Print" option attributes. It is defined as an integer ("int") datatype using the specified XML NameSpace ("xmlns"). It is assigned a value of 1 and the attribute description is terminated (</Ind_Print>). In interpreting the XML representation of a rules profile, the messaging program 116 and/or the stand-alone packager 120 interprets a value of 1 to mean that the intent is enabled and a value of 0 means the intent is disabled. These interpretations in the preferred embodiment then drive the building of a rule or rules by the messaging program 116 and/or the stand-alone packager 120 that can be stored and transported in a CSC These rules are subsequently interpreted and enforced by the TCB whenever the CSC is opened. The fourth line describes the "Ind_View" intent option attribute. See the description of the third line for the details. The 1 indicates "View" is to be enabled. The fifth line describes the "Ind_Edit" intent option attribute. The value of 0 indicates the "Edit" option is NOT to be enabled. The sixth line describes the "Ind_Save" intent option attribute. The value of 1 indicates the "Save" option is to be enabled. The seventh line describes the "Ind_Reply" intent option attribute. The value of 0 indicates the "Reply" option is NOT to be enabled. The eighth line describes the "Ind_Forward" intent option attribute. The value of 0 indicates the "Forward" option is NOT to be enabled. The ninth line describes the "Ind_Sender_Name" replacement value. It is defined as a string. It has the value "Default" which would become the sender name on the email if the next attribute (Ind_Sender_Name_Override) is enabled (has a value of 1). The tenth line describes the "Ind_Sender_Name_Override" option attribute. It is an integer ("int") assigned a value of 1. When interpreting the XML representation of a rules profile, the messaging program 116 and/or the stand-alone packager 120 considers the value of 1 for this attribute to request that the sender's name on the email be replaced during the "Make Secure" operation so that the value specified in the attribute "Ind_Sender_Name" is shown instead. Only after the rendering program 124 caused the successful opening by a TCB of a CSC according to rules will the actual sender name be displayed when the email is opened. The eleventh line describes the "Ind_Subject" line option attribute. It is defined as a string and the value assigned, "TRUSTDATA Secure Message" in this case, is to be the value assigned as the subject line of the email replacing the actual subject line. The actual subject line would only be visible once the CSC was opened by the trusted viewer or rendering application. The value is only used to replace the actual value if the next attribute "Ind_Subject_Override" has the value of 1. The twelfth line describes the "Ind_Subject_Override" option attribute. It is defined as an integer. When it is assigned a value of 1 (as in this example), it indicates the value of the attribute "Ind_Subject" should be used to override the actual entered subject line for the email message. A value of 0 for "Subject_Override" indicates the actual entered subject line should be used. The thirteenth line describes the "Ind_Sunrise" option attribute. It is defined as a dateTime attribute. The value assigned indicates the earliest time

(GMT) at which the email may opened. The fourteenth line describes the "Ind_Sunset" option attribute. It is defined as a dateTime attribute. The value assigned indicates the latest time (GMT) at which the email may be opened. The fifteenth and sixteenth lines provide closure to the elements defined. In another embodiment the XML representation includes: (1) validity date in days from packaging date/time - this option would allow a number of days to be specified for the validity period. The sunrise date would be assigned as the packaging date and the sunset days would be calculated to be the number of days as specified in this attribute after the packaging date. (2) usage auditing options - What audit records should be prepared from use of this email Options include but are not limited to combinations of; at packaging time, on first open, on every open, on first save, on every save, on every print, on first print, no auditing at all.

In one embodiment the messaging program 116, stand-alone packager 120, and/or rendering program 124 will provide a recipient level ID test will authenticate that the specified recipient email address has one or more associated valid TCB UserIDs. An email address may have one or more valid TCB UserIDs This is defined as allowing many UserIDs for one email address to allow for eventual web-based email access and shared file access where the intended recipient may have UserIDs on many different TCBs for the same user (e.g., work computer, home computer, notebook computer, etc.). The alternative of many email addresses associated with the same UserID seems attractive at first glance (lots of people have more than one email address). However, it would make for a not readily detected means of getting unauthorized access to emails intended for others if someone could add their own email address to a list for someone else's UserID. In general, secure control of the address book mapping of email address and UserID is important. The selected profile(s) chosen will be authenticated against the selected recipients at the time the CSC is formed to ensure class and/or group membership. This requires not just an address book but something that lets the messaging program tell what groups a given recipient belongs to and what groups and profiles are valid combinations. In the preferred embodiment, a protected digital data structure that we refer to as a "Digital Credential Object" may represent, signify, warrant, and/or attest to any one or more identity characteristics, attributes, and/or memberships. We refer to these identity characteristics, attributes, and/or memberships equivalently as "memberships." The DCO contains data elements for the name of the DCO, one or more memberships, information that may include cryptographic information indicating the authority that associated the membership with the entity, and information that may include cryptographic information indicating the entity with whom the one or more memberships are associated, and any other fields providing relevant information. Since the DCO is never released from or directly exposed for manipulation outside the control of the TCB, there is no requirement for the calculation of either an integrity check, such as a checksum or cryptographic message digest, and/or the validation of an integrity check using other cryptographic means. In the presently preferred embodiment these DCOs are "Membership Cards" as described, for example, in the

aforementioned U.S. Patent No. 6,112,181 to Shear, et al.

Solutions to Contextual Problems - Membership identification

In one embodiment the messaging program 116, stand-alone packager 120, and/or rendering program 124 will allow email to be sent to general membership groups where that group could expand or contract over time and the access to attachments sent with historical messages to be enabled without having to be a specific recipient. There is both a revocation mechanism (the presence of a revocation of a membership indicates the membership is no longer valid) as well as a way of expiring memberships (date range limits). A given individual can have many memberships. A general membership management application could be of use beyond secure emails and files. A membership can be granted for free or for some information (filling out a survey for example) or for some payment. This membership mechanism may be used to assign a membership specific to each authorized user of our secure email products and automatically include a test for this membership in every rule of every email/file we package. One non limiting benefit of this mechanism would allow RM 134 to quickly revoke that user's access to all secure emails/files should circumstances warrant.

Solutions to Contextual Problems - Each email message may have one or more specific membership cards.

In one embodiment RM 134 will provide membership cards in the form of a protected digital data structure with expiration dates. This will be coordinated with the use of the date validity range mechanism and may be revoked using appropriate revocation token(s) also provide by RM 134. Messaging program 116, stand-alone packager 120 and rendering program 124 will process membership cards according to rules. The use of memberships per email message is primarily to enable revocation of a given recipient's authorization to read a given email message. In some embodiments the memberships would be delivered to the recipients via the same CSC that included the email message. In other embodiments, revocations could be sent in CSCs later either from the sender or potentially from an administrator, such as RM 134. All of this requires that the rules defined in the CSC for the email include a test for the membership in addition to whatever other tests are appropriate.

Expired and/or revoked membership cards and other objects may be deleted by a utility application. In one embodiment, function calls in the API allow membership cards (or other objects) to be found and examined for validity. If invalid, the membership card is deleted. For example, a membership card may have exceeded its validity date. Upon its next use, the membership card is deleted.

Solutions to Contextual Problems - Unique application ID used to restrict access to a particular rendering application(s)

In one embodiment an identifying application ID or application IDs would be associated with

rendering program 124 and would protect the email messages. This mechanism is managed between the messaging program 124 or the stand-alone 120 and the TCB. The messaging program 124 and the stand-alone packager 120 always includes an identifying application ID(s) in CSCs said program cause to be created by a TCB. The rendering program 124 and TCB always test for the presence of that application ID(s) (see Figs. 7 and 8) before the receiver 108 can view the message and any attachments. Third party rendering programs do not work with CSCs created by the messaging program because the third party rendering program does not have the proper application ID. By use of the application ID, a rendering program that may be generally be able to process a CSC is prevented from processing CSC from messaging program 116 and stand-alone packager 120.

Solutions to Contextual Problems - Body of the message replaced by non-informative body

In one embodiment the messaging program 116 and/or the stand-alone packager 120 causes replacement message to include instruction or directions for obtaining a TCB and/or appropriate rights and controls, perhaps from RM 134. The TCB is software that can be downloaded for use in decoding a message. Using the MAPI functions, the messaging program 116 and/or the stand-alone packager 120 obtains the body of the email message and includes it in the CSC as part of the content. The messaging program 116 and/or the stand-alone packager 120 then replaces that body portion of the email message with either blanks (an empty message) or with user/administrator supplied text (if there is some provided) (see Fig. 7).

Solutions to Contextual Problems - Secure "Reply" capability

In one embodiment the messaging program 116, stand-alone packager 120, and/or rendering program 124 provide for the specification and enforcement of rules that specify that the set of "reply" recipients cannot be expanded. Reply can be to some or all, but not more than, the list of authorized recipients. The set of recipients and the UserID test that we infer to include in the rules that govern access to the message is constrained to be just those in the original email's recipient list and the sender. Even if the email user sends the message in a CSC to some other email address, the other receiving email address won't have a UserID that is in the rules specified and therefore won't be able to access the message.

In one embodiment this is implemented as a protected object within the CSC that is removed from the container by the rendering program 124.

Reply capability can be specified with or without decoding of the message such that a reply can be made without a TCB. Without decoding, the recipient cannot read the body of the message. Rules in the reply email limit it to being read by the same set of people who received the original or a proper subset of that group. The email could be delivered to others (perhaps by mistake) but the rules enforced by the TCB would prevent those others from reading the message.

Edit capability can be specified as to identified parts of the email. Such that some portions of the header or body of the message can be edited while other portions cannot. For example, the attachments could be editable while the email body could not.

Reply without Edit capability means that reply text is entered without any edit access to the original email text. This is most likely to be done by leveraging the fact that we do have editing capability for the body of emails with the messaging program. A rule will allow viewing and replying to the message, but will not allow editing or deleting text in the email body.

Reply with Edit capability is more complex since that implies that the body of the email can be changed. This could range from limiting changes to being additions (no modification of existing text but allow in-line addition) to full edit capability (deleting original text or modifying it). There may be distinct options for those types of controls. When the options include modification and deletion of original text, it is essentially the creation of a new document more so than a forwarding of the original and should imply a SAVE intent for the forwarding author.

Attachments have more complex processing. There are content viewers for a wide variety of attachment file types (e.g., the content viewer from INSO™). These content viewers do not, however, support CSC that are opened by TCBs. Some editors have options for tracking changes (MS Word for example) but these off-the-shelf editors are not used. We could use "Contrast" sorts of tools to identify differences for text but the range of files types is large enough we would need a lot of such tools. The "Contrast" sort of approach is also "after the fact" which is not generally a well-received ergonomics choice.

Among the non-limiting example uses of the trusted messaging system disclosed herein is to conduct asynchronous electronic commerce transactions. One non-limiting example is a trusted messaging-based Electronic Data Interchange ("EDI") system. Generally, an EDI system may in part be comprised of a data structure or form with standardized fields. There may be plural forms defined either because there are plural variant forms for a given purpose and/or transaction, and/or there may be many forms-based transactions necessary to accomplish the commerce functions of the system. For example, there may be variant healthcare claim forms that differ in the arrangement and sequencing of the fields. In another example, there may be claim forms and payment forms, the latter resulting in, and/or constituting instructions for a funds transfer system, two non-limiting examples of which are the S.W.I.F.T network and the Automated Clearinghouse ("ACH") system. The trusted messaging system disclosed here provides the basic capabilities of prior art EDI systems. CSCs can be used to securely transport the EDI information from one party to another and to provide authentication and non-repudiation of sending and receiving parties. Using the trusted messaging system disclosed herein, the sender may create rules that govern the access and/or used of field specific information. In one non-limiting example, a healthcare claims form may be sent to a person or process that can only view and/or modify selected portions of the information contained in the form. Such differential access and rights

may depend at least in part on rules that are conditional on the presence and/or absence of credential information securely associated with the receiving person or process. Said credential information may be represented, contained, conveyed, and/or stored in various kinds of objects, non-limiting examples of which are digital certificates known to those skilled in the relevant arts and "membership cards" or "membership objects" is, for example, disclosed in aforementioned U.S. patent No. 6,112,181, Shear, et al. The present invention(s) enable "acceptance of terms" using protected digital objects comprised at least in part by HTML, XML, JAVA, and/or Active X code. In addition, one or plural preferred embodiment clearinghouse(s) may handle charges for non-digital goods and services via the payment processing systems interfaces disclosed in the InterTrust Patents. The security of EDI transactions would be enhanced though the use of time-based thresholds such as "sunrise/sunset" parameters, In this non-limiting example, a given EDI transaction would be valid only during a certain pre-specified time interval.

Solutions to Contextual Problems - Secure "Forward" capability

In one embodiment the messaging program 116, stand-alone packager 120, and/or rendering program 124 provide for the specification of rules that grant authority to forward a message. Authority to forward may be limited only to new recipients within a specified set of domains. Domain in this context means the email domain (the part after the @ in the email address). For example, dconley@trustdatasolutions.com is in the domain trustdatasolutions.com. This technique may be used to limit the forwarding to addresses within a company or other organization, for example.

The rights of the original receiver define the maximum rights of anyone the original receiver forwards the message to. Rules may require subsequent receivers get a more limited set of intents. The forward intent can be specified with or without an edit intent (same concepts as with Reply above). The edit intent can be specified as to identified parts of the email, such as: attachments and email body.

Solutions to Contextual Problems - Code Reuse

In one embodiment the messaging program 116 provides for the reuse of the rules profiles and rules selection dialogs found in the messaging program 116 to allow creation of a CSC for any file. These dialogs are used in a stand-alone packager. This allows the stand-alone packager 120 to apply rules governing their use to any file whether sent as email attachments or not. In another embodiment, The code that invokes the packager routine and rules compiler routine of the messaging program is also reused. Since there isn't a "recipients" list per se with files, we add dialogs to prompt for authorized UserIDs or memberships to the stand-alone packager. The file packager works both as a plug-in to some MS Office products (e.g., Word, Excel, etc.) in addition to being available as a standalone executable, which can prompt for the file to be packaged (most software routines are reused across these two settings).

Plug-in to many email clients. For the purpose of viewing the secured package as received by many different types of email clients, we have built the presently preferred embodiment of a rendering program on the common Messaging Application Programming Interface (MAPI) implemented by all the major email clients; however, other rendering program embodiments are also envisioned as well.

Solutions to Contextual Problems - Usage (view, print, save, etc.) tracking/auditing.

Usage can be both governed in terms of who is allowed to use (view, print, save are separate types of usage intents with separate authorization rules) and what if any audit trail (usage records) are created to document this (see Fig. 8).

Solutions to Contextual Problems - Authentication of intended recipient

In one embodiment the messaging program 116, stand-alone packager 120, and/or rendering program 124 utilize the user ID of the TCB (i.e., UserID) and Password for authentication of the user. Support degrees of authentication, including one or more of the following: password, biometrics, cardkey, and others. Each of these techniques require our rendering programs to test for a higher level of authentication. In the default case, we trust the UserID/PASSWORD scheme of authentication used by the TCB. When additional authentication is required, we note this requirement in the packaged CSC (as a value in an additional attribute on the container object (or perhaps the content object)). Our Rendering program then tests for that attribute value to decide if additional authentication is required.

There are at least two different non-limiting choices for managing the additional authentication. One choice is to place the signature(s) to match (biometric or other) inside the CSC and test against them before invoking the other authorization tests. A second choice is to interact with an authentication service (perhaps via a directory service) which would manage the signature(s) to be tested against. In any case, we would interact with the biometric device or cardkey or other in the conventional ways suggested by the makers of those products.

Solutions to Contextual Problems - Full usage rights for email "originator"

In one embodiment the messaging program 116, stand-alone packager 120, and/or rendering program 124 provide for the specification and enforcement of full usage rights of the embedded content may always be granted to the originator of the new secured email. Rule may be applied by default to allow the "originator" full access to the content regardless of additional specified rules. The originator must have an option to allow "sunset" emails to become unreadable by the originator also.

While the present invention has been described with reference to certain preferred embodiments, it is to be understood that the present invention is not to be limited to such specific embodiments. Rather, it is the inventor's intention that the invention be understood and construed in its broadest meaning as reflected by the following claims. Thus, these claims are to be understood as incorporating and not only

the preferred embodiment described herein but all those other and further alterations and modifications as would be apparent to those of ordinary skill in the art.

APPENDIX 1 XML-Encoded Default Rules Profile

```
Email_Settings xmlns:dt="urn:schemas-microsoft-com:datatypes">
     <Individual>
          <Ind_Print dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_Print>
          <Ind_View dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_View>
          <Ind_Edit dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Edit>
          <Ind_Save dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_Save>
          <Ind_Reply dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Reply>
          <Ind_Forward dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Forward>
          <Ind_Sender_Name dt:dt="string" xmlns:dt="urn:schemas-
microsoft-com:datatypes">Default</Ind_Sender_Name>
          <Ind_Sender_Name_Override dt:dt="int"
xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_Sender_Name_Override>
          <Ind_Subject dt:dt="string" xmlns:dt="urn:schemas-
microsoft-com:datatypes">TRUSTDATA Secure Message</Ind_Subject>
          <Ind_Subject_Override dt:dt="int" xmlns:dt="urn:schemas-
microsoft-com:datatypes">1</Ind_Subject_Override>
          <Ind_Sunrise dt:dt="dateTime" xmlns:dt="urn:schemas-
microsoft-com:datatypes">2000-09-25T12:12:41.000</Ind_Sunrise>
          <Ind_Sunset dt:dt="dateTime" xmlns:dt="urn:schemas-
microsoft-com:datatypes">2000-12-14T12:12:37.000</Ind_Sunset>
     </Individual>
</Email_Settings>
```

APPENDIX 2 XML-Encoded Healthcare Provider Rules Profile

```
<Email_Settings xmlns:dt="urn:schemas-microsoft-com:datatypes">
     <Individual>
          <Ind_Print dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Print>
          <Ind_View dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_View>
          <Ind_Edit dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Edit>
          <Ind_Save dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_Save>
          <Ind_Reply dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Reply>
          <Ind_Forward dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Forward>
          <Ind_Sender_Name dt:dt="string" xmlns:dt="urn:schemas-
microsoft-com:datatypes">Dr. XYZ</Ind_Sender_Name>
          <Ind_Sender_Name_Override dt:dt="int"
xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_Sender_Name_Override>
          <Ind_Subject dt:dt="string" xmlns:dt="urn:schemas-
microsoft-com:datatypes">HIV Results</Ind_Subject>
          <Ind_Subject_Override dt:dt="int" xmlns:dt="urn:schemas-
microsoft-com:datatypes">1</Ind_Subject_Override>
          <Ind_Sunrise dt:dt="dateTime" xmlns:dt="urn:schemas-
microsoft-com:datatypes">2000-09-18T10:26:13.000</Ind_Sunrise>
          <Ind_Sunset dt:dt="dateTime" xmlns:dt="urn:schemas-
microsoft-com:datatypes">2001-02-08T10:26:08.000</Ind_Sunset>
     </Individual>
</Email_Settings>
```

APPENDIX 3 XML-Encoded Healthcare Patient Rules Profile

```
<Email_Settings xmlns:dt="urn:schemas-microsoft-com:datatypes">
    <Individual>
        <Ind_Print dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_Print>
        <Ind_View dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_View>
        <Ind_Edit dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Edit>
        <Ind_Save dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_Save>
        <Ind_Reply dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Reply>
        <Ind_Forward dt:dt="int" xmlns:dt="urn:schemas-microsoft-
com:datatypes">0</Ind_Forward>
        <Ind_Sender_Name dt:dt="string" xmlns:dt="urn:schemas-
microsoft-com:datatypes">John Doe</Ind_Sender_Name>
        <Ind_Sender_Name_Override dt:dt="int"
xmlns:dt="urn:schemas-microsoft-
com:datatypes">1</Ind_Sender_Name_Override>
        <Ind_Subject dt:dt="string" xmlns:dt="urn:schemas-
microsoft-com:datatypes">Payment Method</Ind_Subject>
        <Ind_Subject_Override dt:dt="int" xmlns:dt="urn:schemas-
microsoft-com:datatypes">1</Ind_Subject_Override>
        <Ind_Sunrise dt:dt="dateTime" xmlns:dt="urn:schemas-
microsoft-com:datatypes">2000-12-06T11:10:32.000</Ind_Sunrise>
        <Ind_Sunset dt:dt="dateTime" xmlns:dt="urn:schemas-
microsoft-com:datatypes">2000-12-20T11:10:36.000</Ind_Sunset>
    </Individual>
</Email_Settings>
```

What we claim are:

CLAIMS

1. A method for securing electronic messages, comprising of:

a packager for packaging an electronic message prepared by a sender;

a first trusted computing base for securing and passing said packaged electronic message in accordance with specified rules;

a second trusted computing base for receiving and decoding said secured electronic message in accordance with said rules; and

a rendering program for manipulating said decoded electronic message in accordance with said rules.

2. A method as recited in claim 1 wherein said packager comprises of a mail client and a messaging program.

3. A method as recited in claim 2 wherein said messaging program includes a packager routine and a rules compiler for compiling said rules.

4. A method as recited in claim 1 further including a secured container securely transporting said message from said first trusted computing base to said second trusted computing base.

5. A method as recited in claim 1 further including a clearinghouse for providing additional rules to said first and second trusted computing bases.

6. A method as recited in claim 5 wherein said additional rules are usage rules for manipulating said message.

7. A method as recited in claim 5 wherein said additional rules are financial rules for manipulating financial information.

8. A method as recited in claim 1 further including a clearinghouse for tracking information of said message.

9. A method as recited in claim 1 further including a rights manager for authorizing the usage of said

packager.

10. A method as recited in claim 1 further including a rights manager for authorizing the usage of sai rendering program.

11. A method as recited in claim 1 further including a rights manager for communicating rules with sai first and second trusted computing base.

12. A method as recited in claim 1 wherein said rendering program controls the forwarding and th replying of said message in accordance with said rules.

13. A method for securing electronic messages, comprising of:

a rights manager for authorizing the usage of a packager and a rendering program;

said packager for packaging an electronic message prepared by a sender;

a first trusted computing base for securing and passing said packaged electronic message i accordance with specified rules;

a second trusted computing base for receiving and decoding said secured electronic message i accordance with said rules; and

said rendering program for manipulating said decoded electronic message in accordance with sai rules.

14. A method as recited in claim 13 wherein said packager comprises of a mail client and a messagin program.

15. A method as recited in claim 14 wherein said messaging program includes a packager routine and rules compiler for compiling said rules.

16. A method as recited in claim 13 further including a secured container securely transporting sai message from said first trusted computing base to said second trusted computing base.

17. A method as recited in claim 13 further including a clearinghouse for providing additional rules t said first and second trusted computing bases.

18. A method as recited in claim 17 wherein said additional rules are usage rules for manipulating said message.

19. A method as recited in claim 17 wherein said additional rules are financial rules for manipulating financial information.

20. A method as recited in claim 13 further including a clearinghouse for tracking information of said message.

21. A method as recited in claim 13 further including a rights manager for communicating rules with said first and second trusted computing base.

22. A method as recited in claim 13 wherein said rendering program controls the forwarding and the replying of said message in accordance with said rules.

23. A method for securing electronic messages, comprising of:

a rights manager for authorizing the usage of a packager and a rendering program;

said packager for packaging an electronic message prepared by a sender;

a first trusted computing base for securing and passing said packaged electronic message in accordance with specified rules;

a secured container securely transporting said secured electronic message from said first trusted computing base to said second trusted computing base;

a second trusted computing base for receiving and decoding said secured electronic message in accordance with said rules; and

said rendering program for manipulating said decoded electronic message in accordance with said rules.

24. A method as recited in claim 23 wherein said packager comprises of a mail client and a messaging program.

25. A method as recited in claim 24 wherein said messaging program includes a packager routine and a rules compiler for compiling said rules.

26. A method as recited in claim 23 further including a clearinghouse for providing additional rules to said first and second trusted computing bases.

27. A method as recited in claim 26 wherein said additional rules are usage rules for manipulating said message.

28. A method as recited in claim 26 wherein said additional rules are financial rules for manipulating financial information.

29. A method as recited in claim 23 further including a clearinghouse for tracking information of said message.

30. A method as recited in claim 23 further including a rights manager for communicating rules with said first and second trusted computing base.

31. A method as recited in claim 23 wherein said rendering program controls the forwarding and the replying of said message in accordance with said rules.

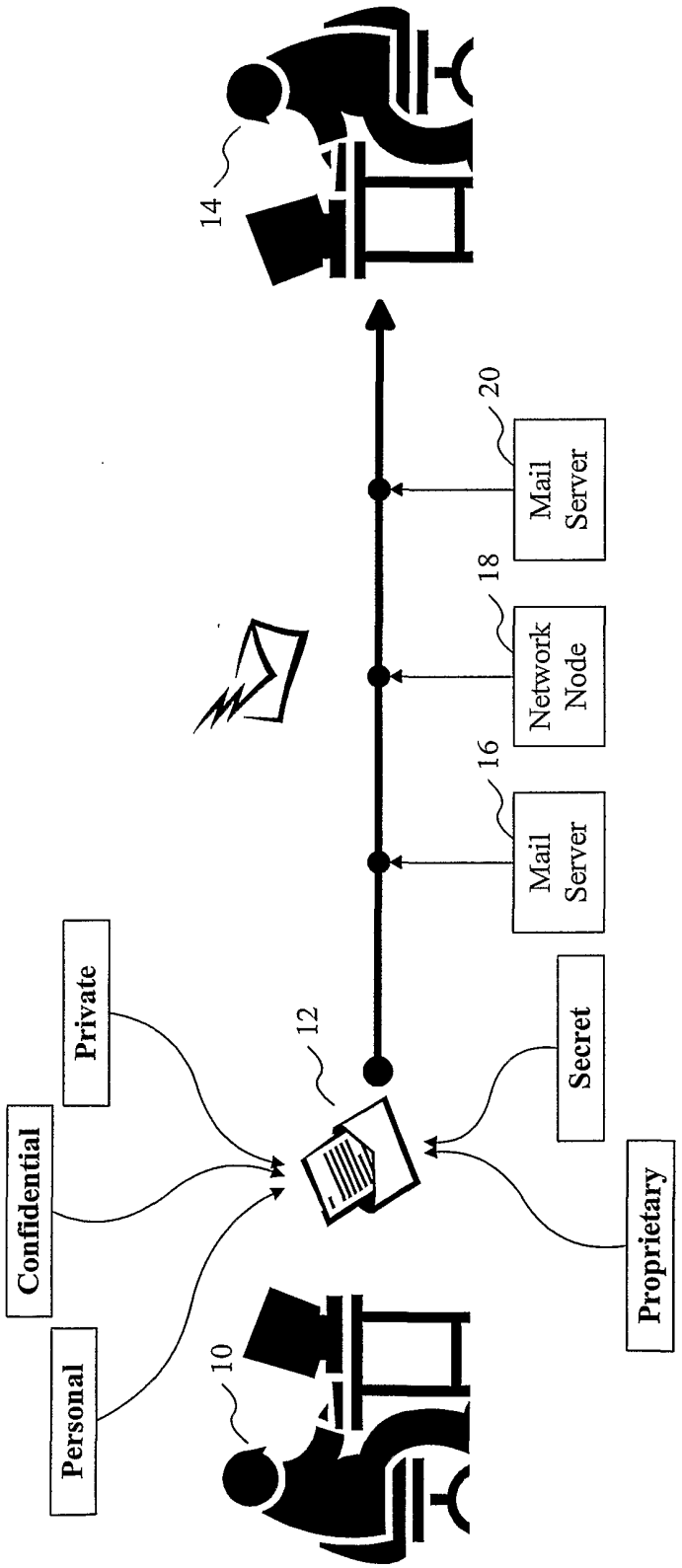

14
20
Mail Server
18
Network Node
16
Mail Server
12
Private
Confidential
Personal
Secret
Proprietary
10

Fig. 1

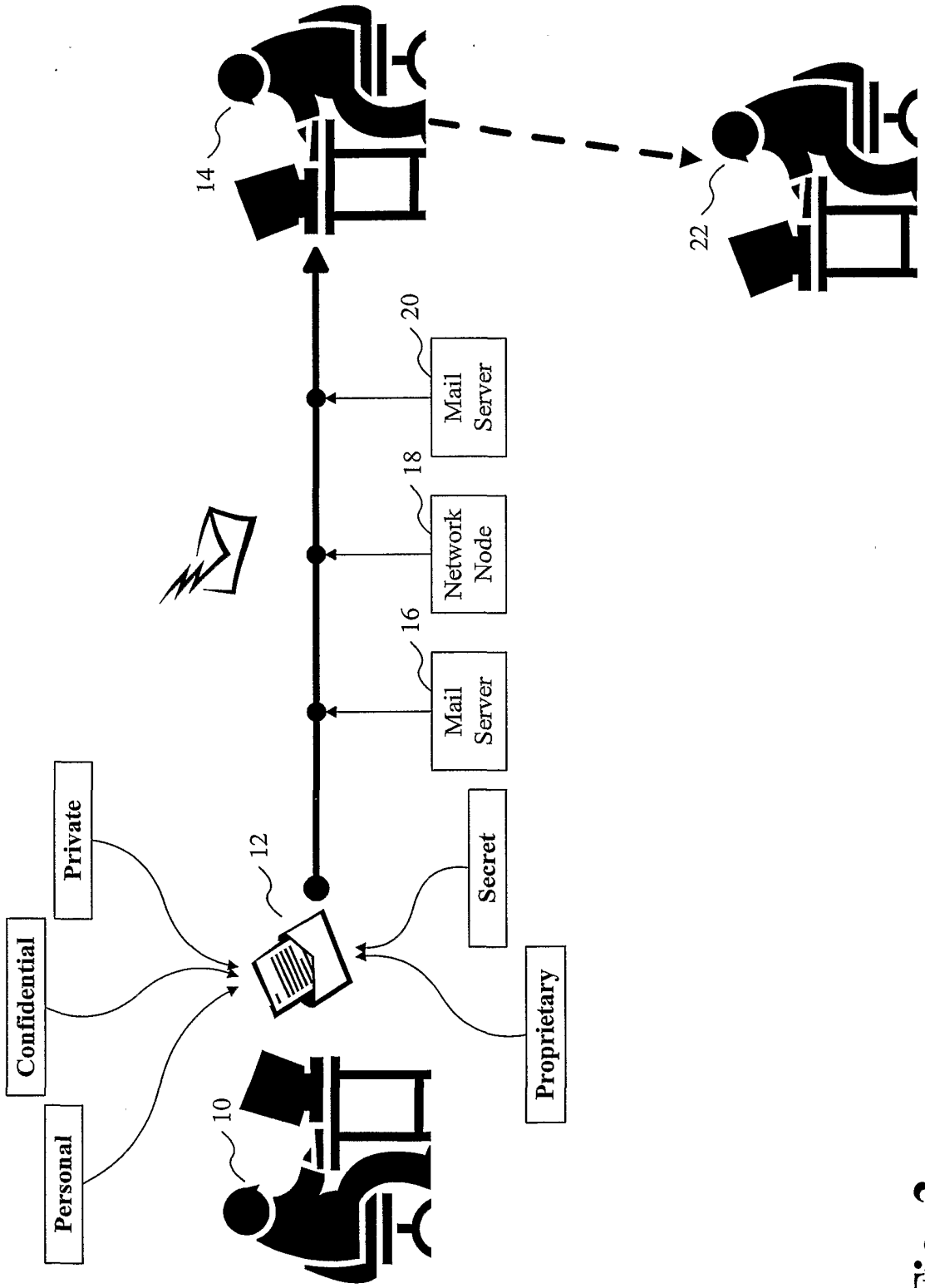

14
22
20
Mail Server
18
Network Node
16
Mail Server
12
10
Private
Confidential
Personal
Secret
Proprietary

Fig. 2

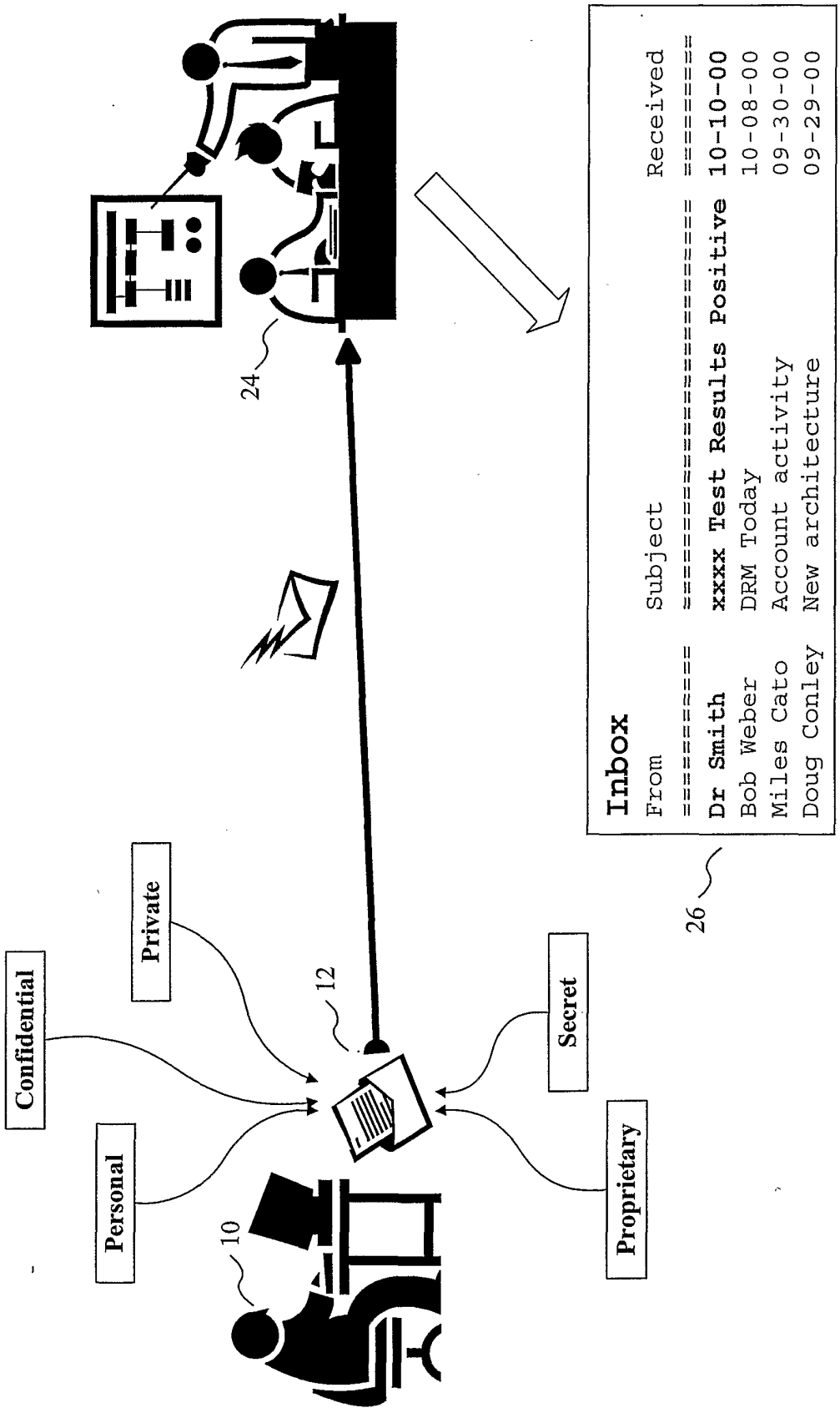

Inbox
From Subject Received
Dr Smith xxxx Test Results Positive 10-10-00
Bob Weber DRM Today 10-08-00
Miles Cato Account activity 09-30-00
Doug Conley New architecture 09-29-00
24
26
12
10
Confidential
Private
Personal
Secret
Proprietary

Fig. 3

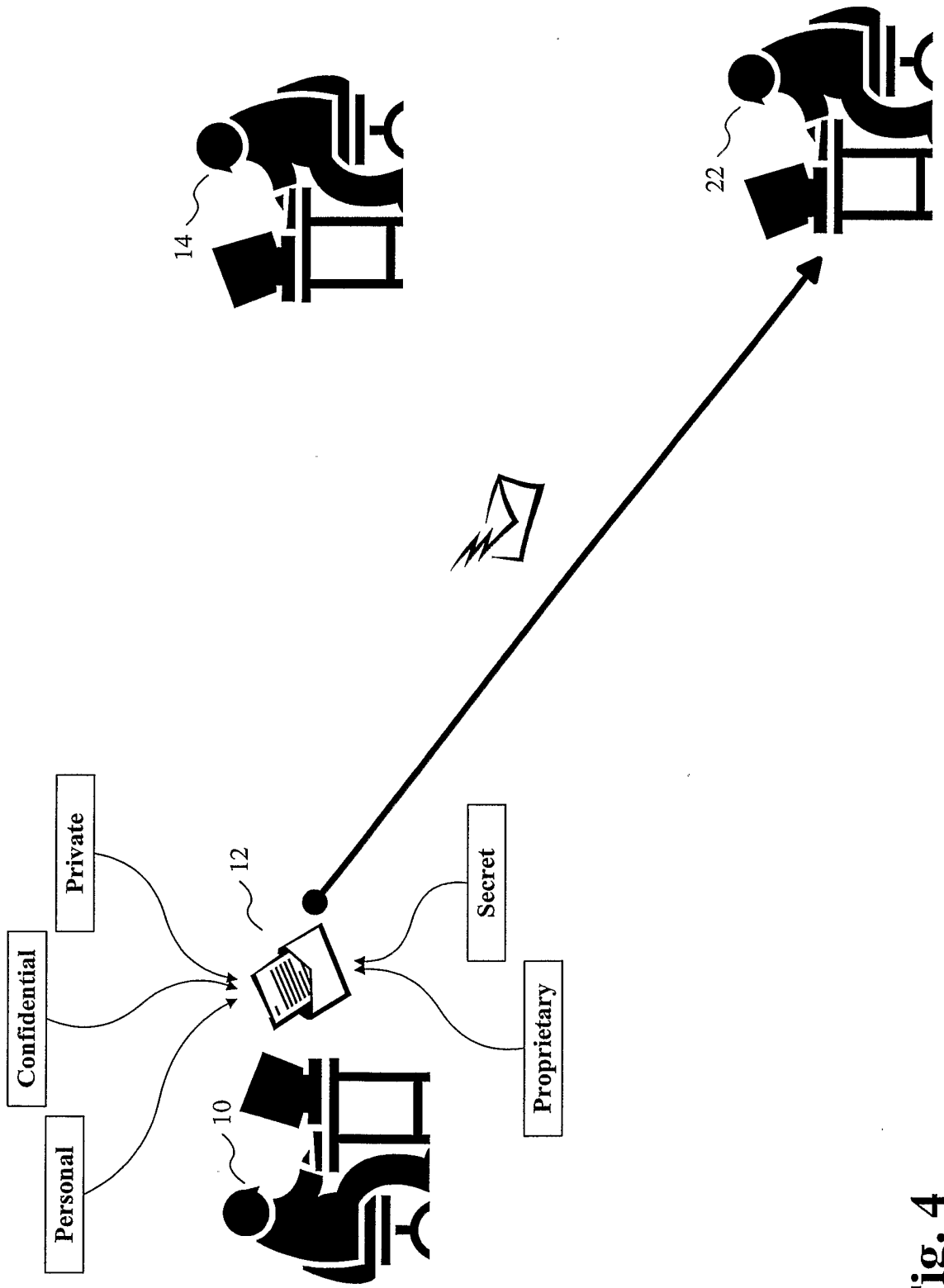

14
22
12
10
Private
Confidential
Personal
Secret
Proprietary

Fig. 4

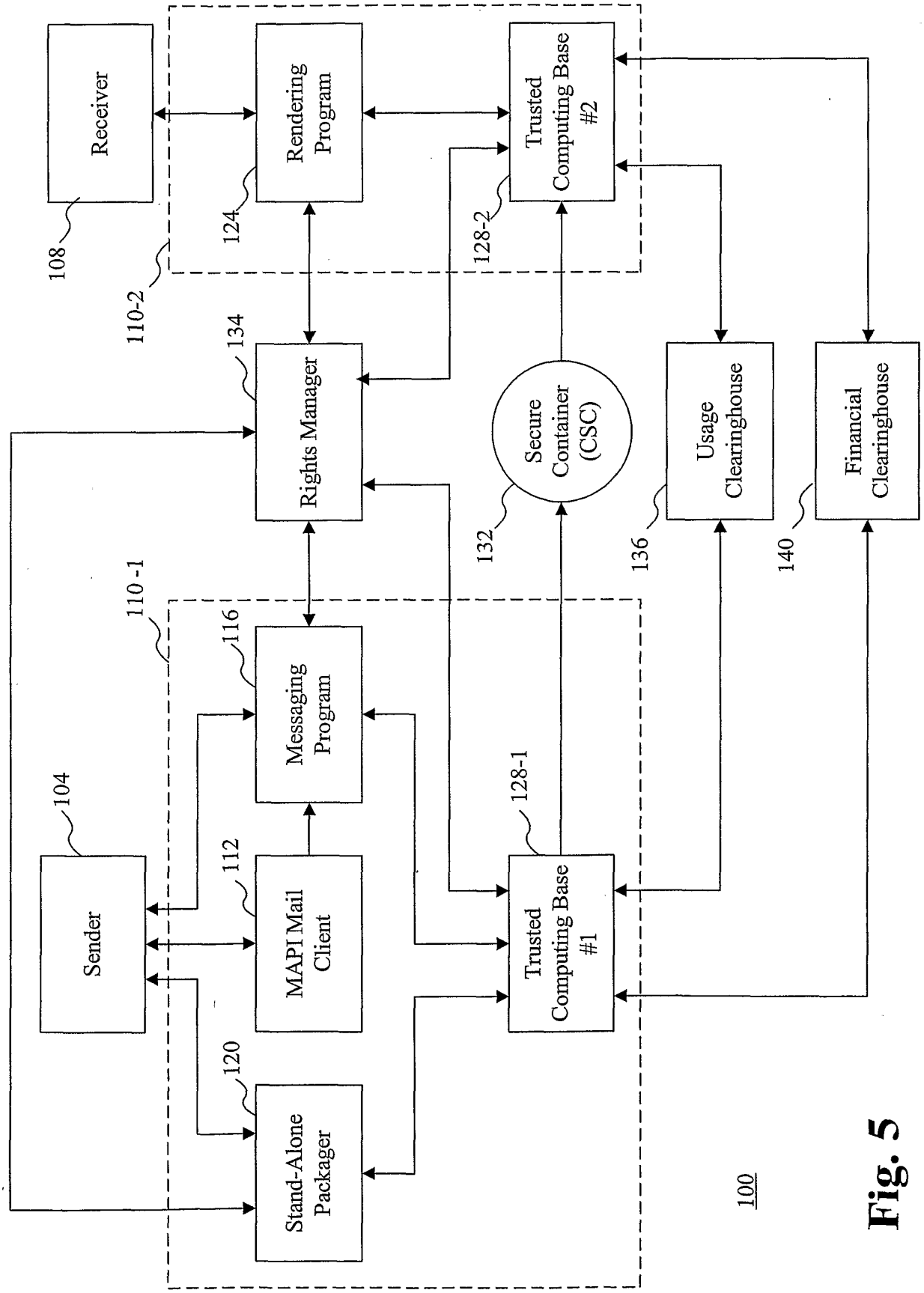

Receiver
108
Rendering Program
124
Trusted Computing Base #2
128-2
110-2
134
Rights Manager
Secure Container (CSC)
132
Usage Clearinghouse
136
Financial Clearinghouse
140
110-1
116
Messaging Program
104
Sender
112
MAPI Mail Client
128-1
Trusted Computing Base #1
120
Stand-Alone Packager
100

Fig. 5

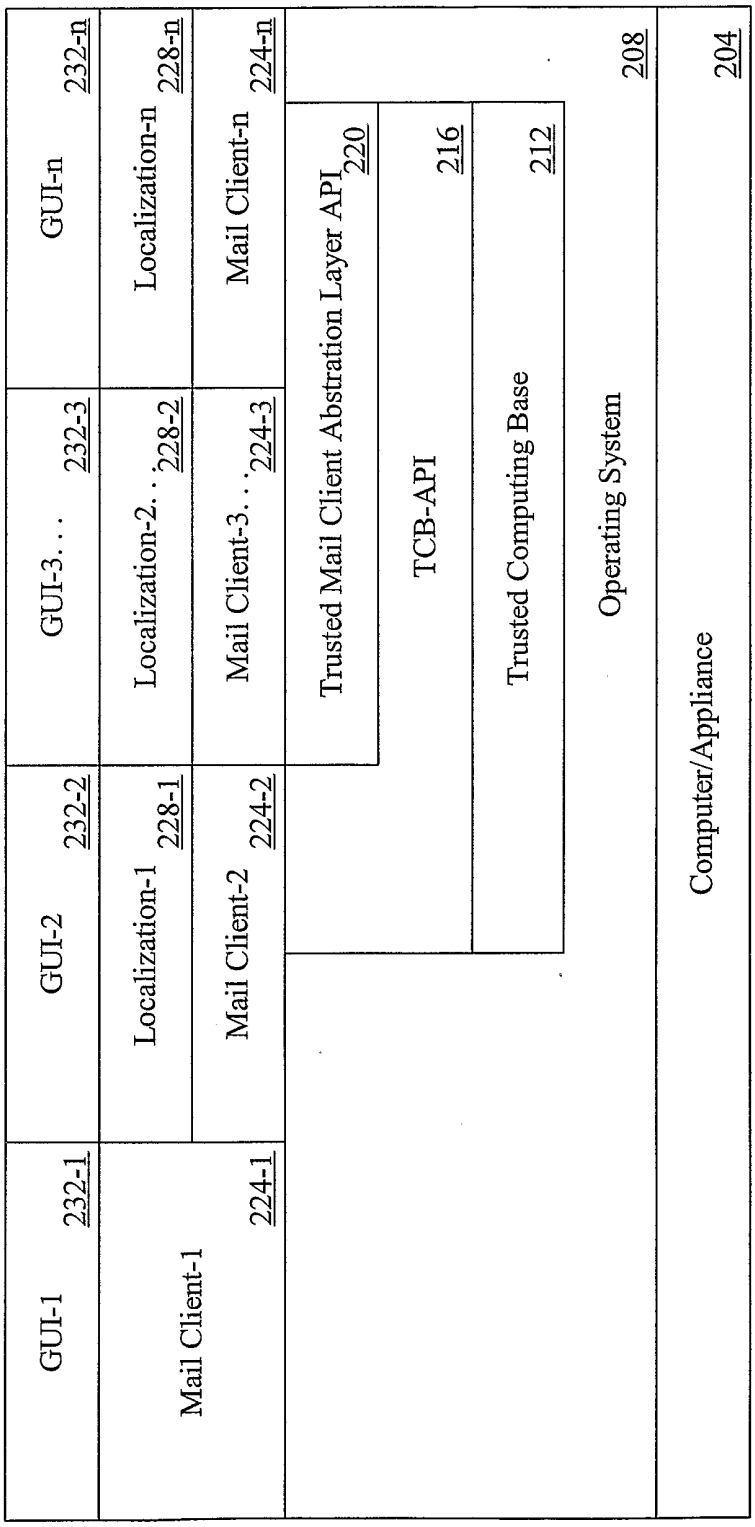

GUI-1 232-1
GUI-2 232-2
GUI-3. . . 232-3
GUI-n 232-n
Mail Client-1 224-1
Localization-1 228-1
Mail Client-2 224-2
Localization-2. . . 228-2
Mail Client-3. . . 224-3
Localization-n 228-n
Mail Client-n 224-n
Trusted Mail Client Abstration Layer API 220
TCB-API 216
Trusted Computing Base 212
Operating System 208
Computer/Appliance 204

Fig. 6

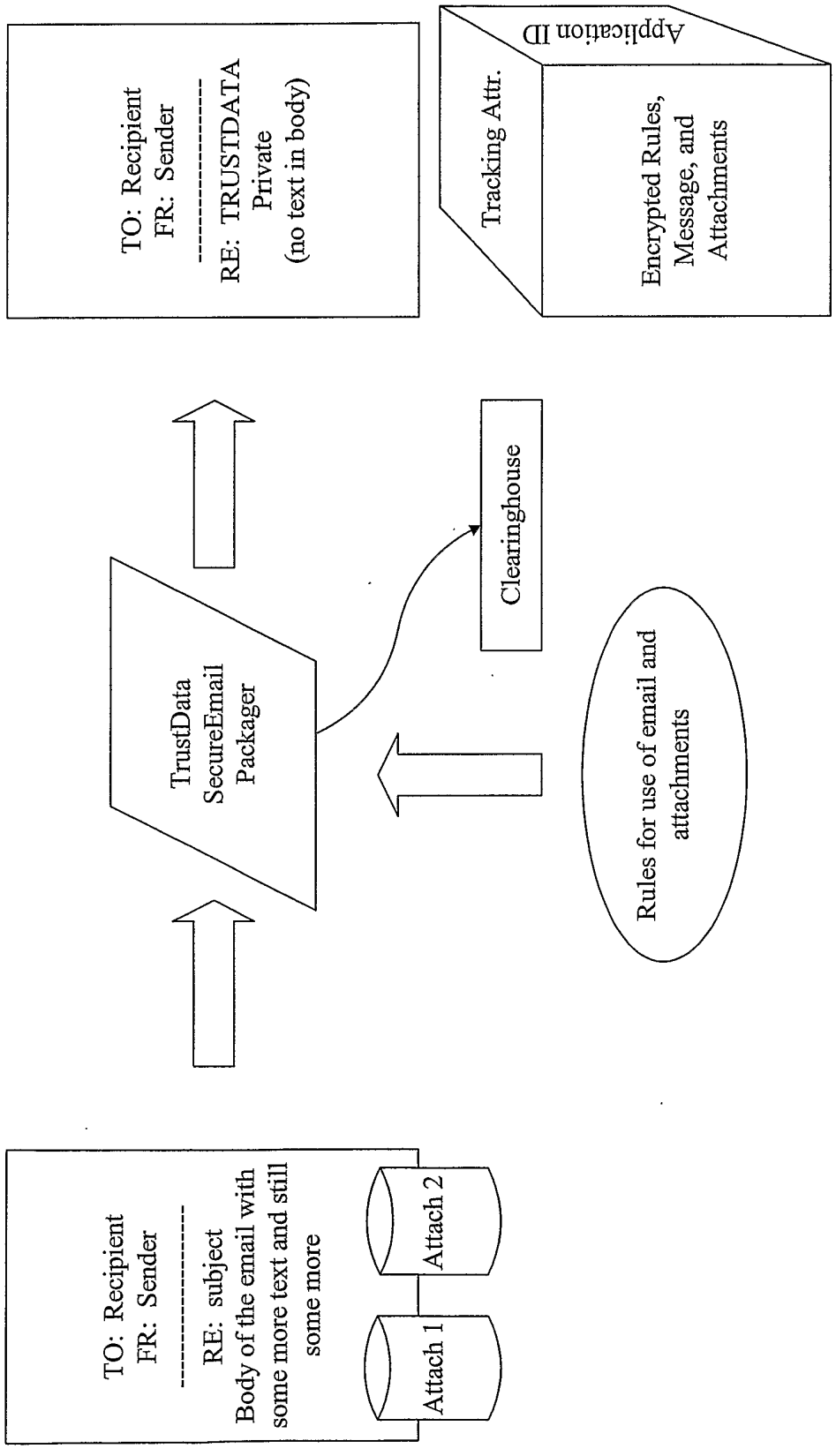

TO: Recipient
FR: Sender
RE: subject
Body of the email with some more text and still some more
Attach 1
Attach 2
TrustData SecureEmail Packager
Clearinghouse
Rules for use of email and attachments
TO: Recipient
FR: Sender
RE: TRUSTDATA Private
(no text in body)
Application ID
Tracking Attr.
Encrypted Rules, Message, and Attachments

Fig. 7

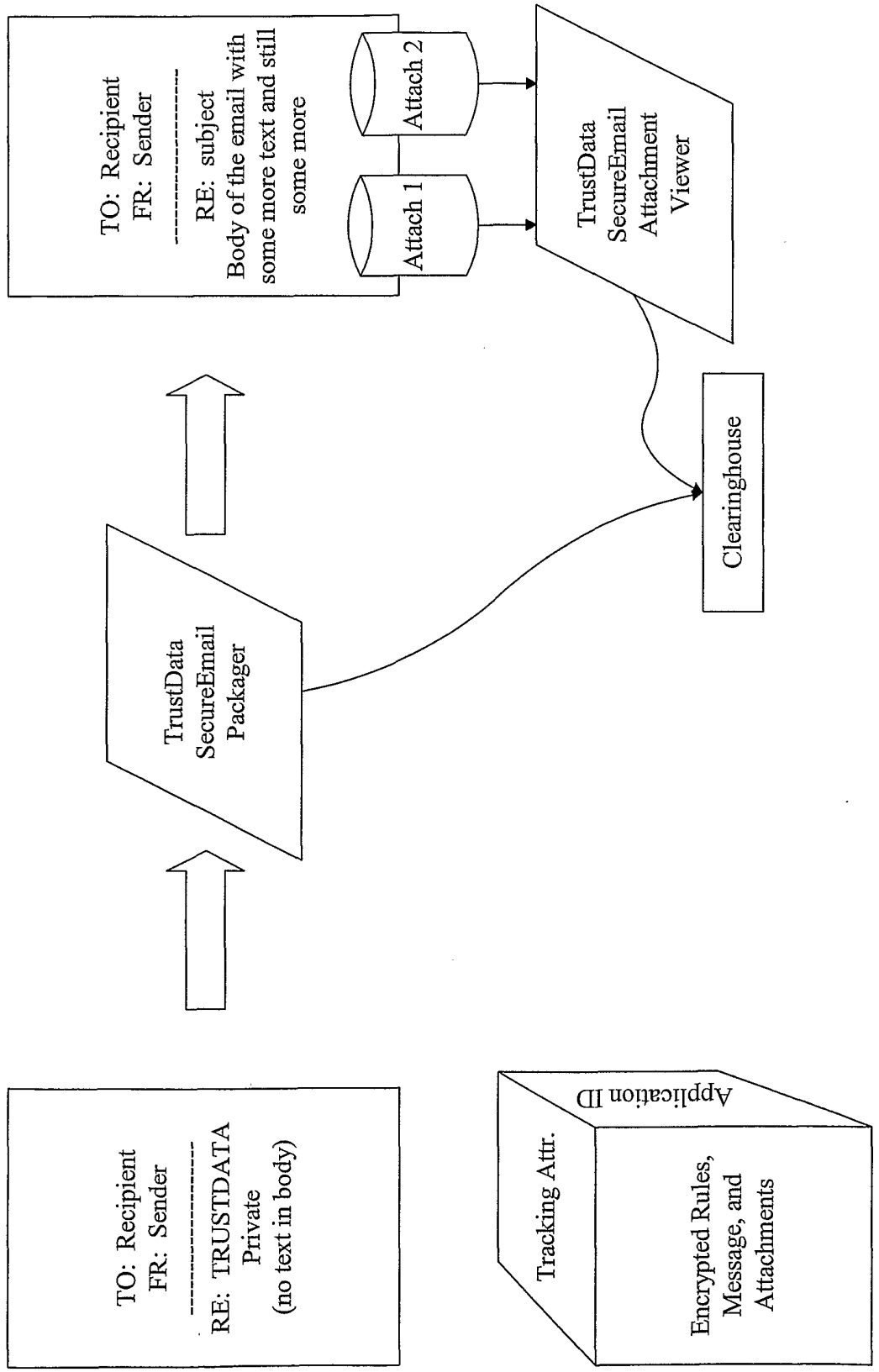

TO: Recipient
FR: Sender
RE: TRUSTDATA
Private
(no text in body)
TrustData
SecureEmail
Packager
TO: Recipient
FR: Sender
RE: subject
Body of the email with
some more text and still
some more
Attach 1
Attach 2
TrustData
SecureEmail
Attachment
Viewer
Clearinghouse
Application ID
Tracking Attr.
Encrypted Rules,
Message, and
Attachments

Fig. 8

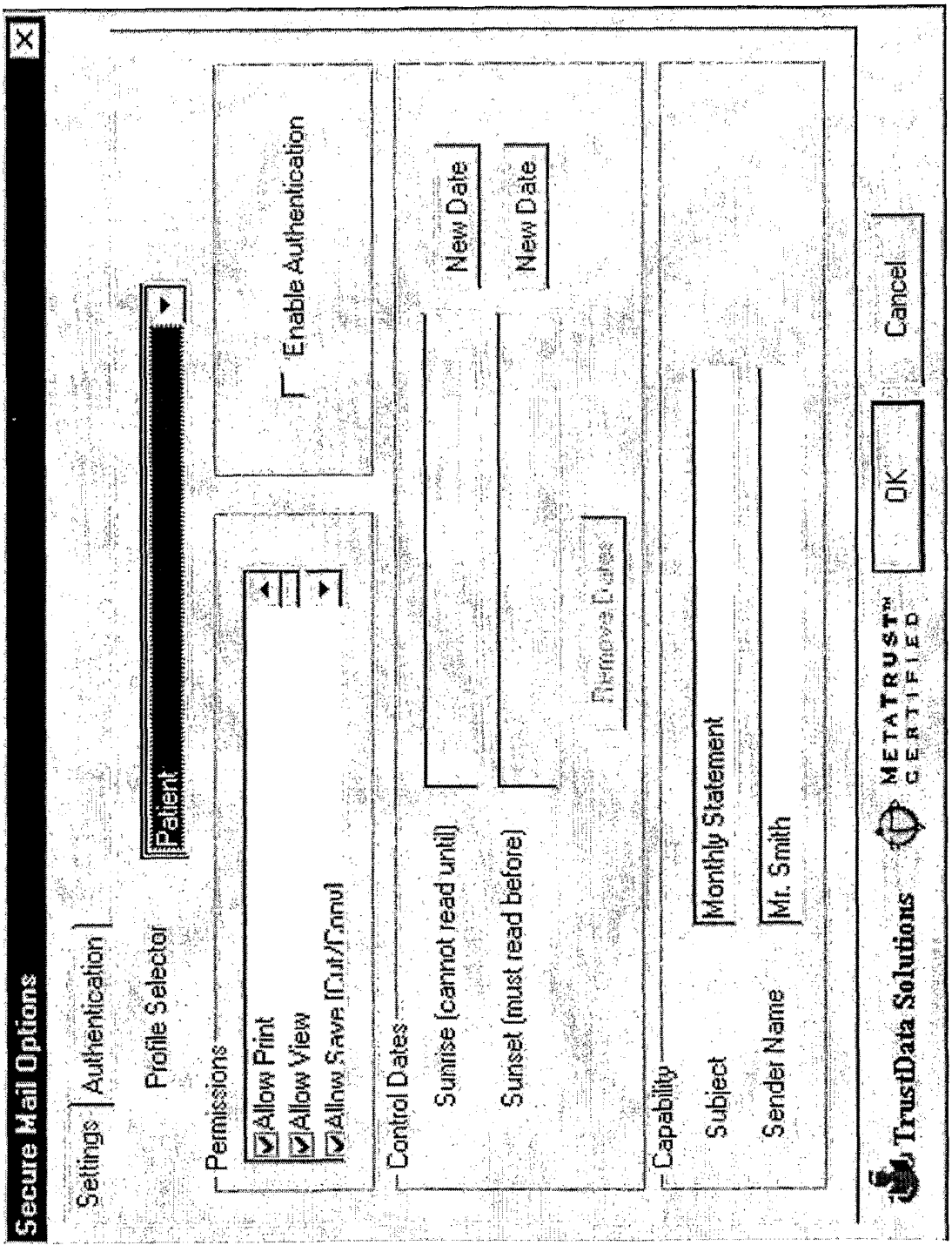

Secure Mail Options
Settings
Authentication
Profile Selector
Patient
Permissions
Allow Print
Allow View
Enable Authentication
Control Dates
Sunrise (cannot read until)
New Date
Sunset (must read before)
New Date
Capability
Subject
Monthly Statement
Sender Name
Mr. Smith
TrustData Solutions
METATRUST™ CERTIFIED
OK
Cancel

Fig. 9

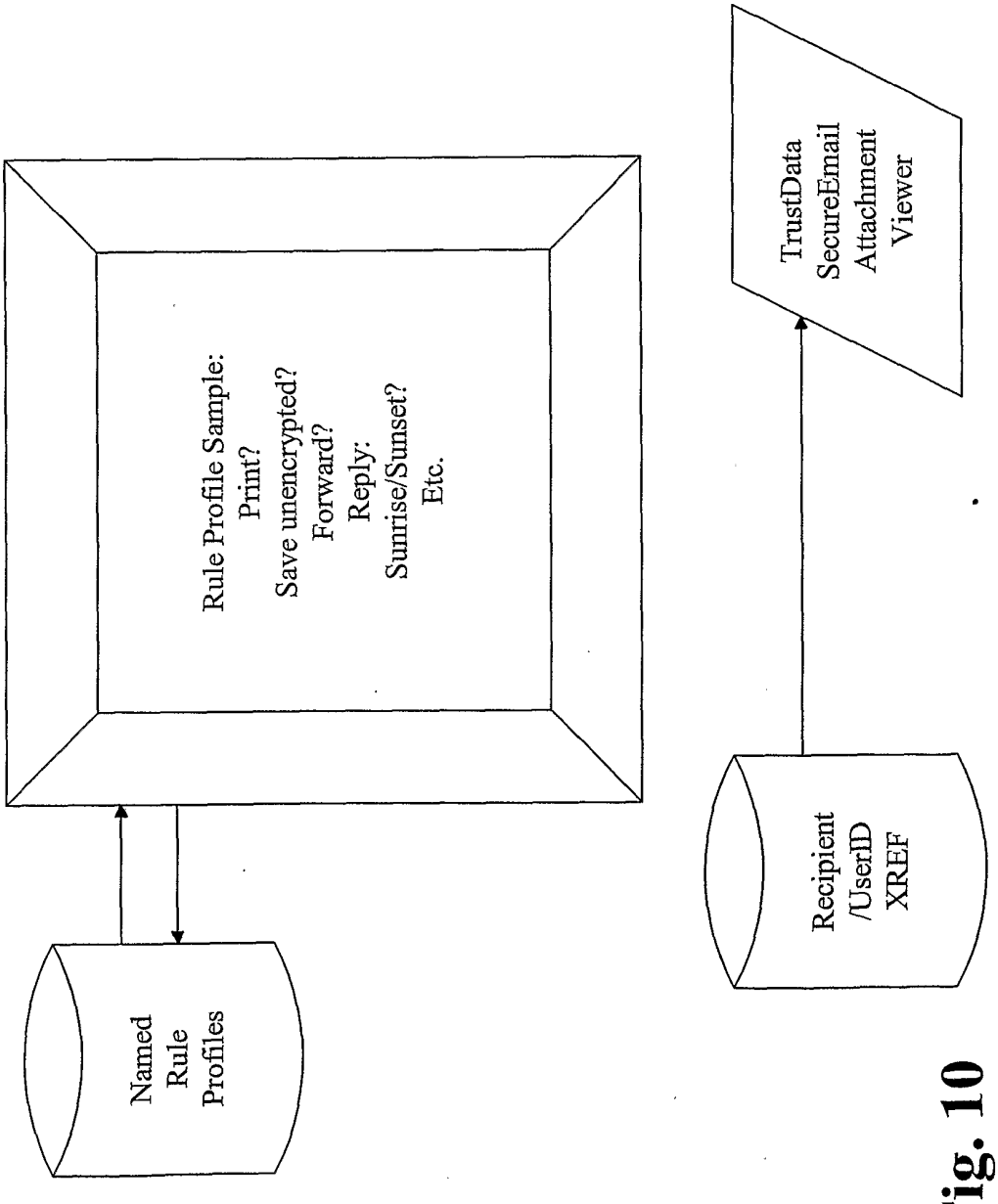

Rule Profile Sample:
Print?
Save unencrypted?
Forward?
Reply:
Sunrise/Sunset?
Etc.
TrustData
SecureEmail
Attachment
Viewer
Named
Rule
Profiles
Recipient
/UserID
XREF

Fig. 10