PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : **H04L 12/58, 29/06**	A2	(11) International Publication Number: **WO 00/22787** (43) International Publication Date: 20 April 2000 (20.04.00)

(21) International Application Number: PCT/US99/23453

(22) International Filing Date: 8 October 1999 (08.10.99)

(30) Priority Data:
09/168,936 9 October 1998 (09.10.98) US

(71) Applicant: BANKERS TRUST COMPANY [US/US]; 130 Liberty Street, Mail Stop 2310, New York, NY 10006 (US).

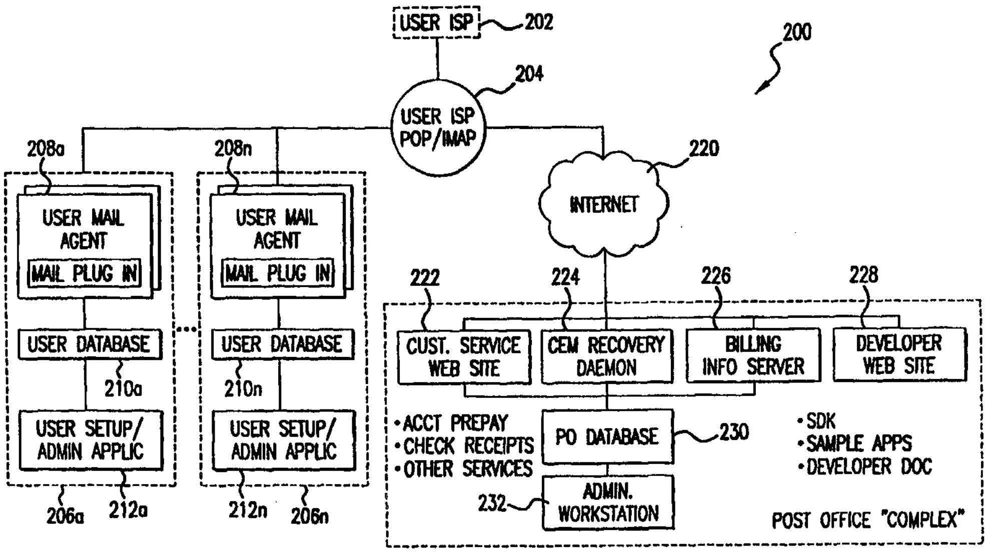
(72) Inventor: SUDIA, Frank, Wells; Apartment 14s, 25 Broad Street, New York, NY 10004 (US).

(74) Agents: RAY, Michael, B. et al.; Sterne, Kessler, Goldstein & Fox P.L.L.C., Suite 600, 1100 New York Avenue, Washington, DC 20005-3934 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: METHOD, SYSTEM, AND COMPUTER PROGRAM PRODUCT FOR PROVIDING ENHANCED ELECTRONIC MAIL SERVICES



(57) Abstract

A method, system and computer program product for providing enhanced electronic mail services such as certified electronic mail. The method facilitates a recipient's reconstruction of the initial message, eliminates full resend of the message, minimizes communication during the recovery step, eliminates the need for super encryption, allows the parties to delegate performance to agents (208a, 208n), and provides explicit system enrollments by user certificate authorities (CA's). The system includes a plurality of user sites and a central post office complex (240).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

METHOD, SYSTEM, AND COMPUTER PROGRAM PRODUCT FOR PROVIDING ENHANCED ELECTRONIC MAIL SERVICES

Background of the Invention

Field of the Invention

The present invention relates generally to simultaneous electronic transactions, and more particularly to electronic authorization functions such as certified electronic mail and the like.

Related Art

The explosion in the size and use of the global Internet has given rise to the ability for people to instantaneously communicate via such means as electronic mail. Such communications, originally used mostly for academic and personal uses, have increasingly come to be used in the commercial and business arenas. For many, electronic mail has come to replace postal mail as the primary mechanism for communication. Electronic mail allows the high-speed and convenient transfer of text, graphics, and voice data and thus allows the delivery of many types of documents. These characteristics have made electronic mail the most popular and most widely-used service over the Internet.

With the rise in the use of electronic mail (and the Internet in general) for commercial uses, an entire electronic commerce industry has developed. Electronic commerce includes businesses, individual entrepreneurs, organizations, and the like who offer their services and products to people all over the world via the Internet. The Internet and electronic mail communications with potential clients and customers, who may be located anywhere in the world, greatly expands the opportunities to offer services and products as well as disseminate information.

Electronic commerce, like traditional commerce, depends on the ability of parties to perform transactions (e.g., an exchange of goods for some value). Thus, communications and electronic mail in general, must develop certain protocols in order to ensure transactions occur in an orderly and secure fashion. Similar to traditional commerce, parties must be able to assure the identity of those they deal with, verify the integrity of messages (e.g., orders) they receive, expect that any desired privacy of a transaction be maintained, and rely on the finality of a transaction by exchanging receipts and payments. In sum, the communications mechanism of electronic commerce, electronic mail, must provide the basic assurances found in traditional commerce.

Many have attempted to solve the problem of making electronic commerce as secure and trustworthy as traditional commerce, thereby giving participants a high comfort level that will allow proper exploitation of electronic commerce technology. The use of digital signatures and public-key encryption in electronic transactions have brought security to the field of electronic commerce. This has been the case especially in communications involving financial transactions (e.g., payments, transfers, etc.). Furthermore, advances in telecommunications equipment and computing have brought about increased speed to the field of electronic commerce.

One problem with electronic transactions, not solved until recently, is that of "simultaneity." With traditional face-to-face transactions, parties are simultaneously able to exchange value (i.e., cash, check, etc.) for a good or service received. In an electronic transaction, however, one party to a transaction may not be willing to send a valuable digital message or instruction without first receiving the other party's message. Conversely, a party may not be willing to send an electronic receipt without first receiving the other party's message.

An electronic communication method for simultaneous electronic transactions (SET) was described in U.S. Patent No. 5,666,420 issued to Micali ("Micali"). Micali addressed the simultaneity problem in a two-party context. Referring to **FIG. 1**, a flow diagram briefly summarizing Micali's extended certified mail (ECM) protocol 100, which accomplishes SET between two parties with minimal reliance and support of a third party is shown.

The description of **FIG. 1** below uses notations that will be apparent to one skilled in the relevant art(s) (e.g., cryptography). However, for completeness, the cryptographic notations used in **FIG. 1** as well as throughout herein are summarized in **Table 1**. A detailed discussion of Cryptography and its associated notations can be found in B. Schneier, *Applied Cryptography*, John Wiley & Sons, 2nd ed. 1996 (USA) ("Schneier"). Furthermore, it is assumed, for simplicity, that the parties A and B are using a secure public-key encryption algorithm (e.g., RSA) and that the parties are able to communicate electronically via a computer network or the like.

TABLE 1

NOTATION	DEFINITION
M	Plain text message
M_n	Transmitted Encrypted (i.e., ciphertext) message number n
$S_X(M)$	A message M which is signed by a party X (includes the digital signature and may also include the message)
R	A receipt; same as $S_x(M)$, generally just the signature plus related info
K^-_X	The private signature key of user X
K^+_{PO}	The public encryption key of the "Post Office"
$E_X(M)$	Encryption of a message M using the public key of a party X
$\mid$	Concatenation operation
$\oplus$	Exclusive Or (XOR) operation
A➔B	Transmission of an electronic message in the direction of from a party A to a party B
H(M)	Computing a cryptographic hash function of variable-length M (the pre-image) to obtain a fixed-length hash value
$Cert_X$	The (digital) certificate of user X, issued by a certificate authority
$D_X(M_X)$	Decryption using user X's private decryption key, K^-_X

NOTATION	DEFINITION
Info	Generally, the associated transaction ID of the sender and/or recipient, and any other (usually brief) data attributes for transaction control, message routing, and legal semantics, as defined herein or known to those skilled in the relevant art(s)

TABLE 1

FIG. 1 illustrates a party A who desires to send a message to another party B and obtain a receipt. The ECM protocol uses an "invisible" trusted third-party, referred to as a post office (PO), to facilitate the transaction. In step 102, party A first takes her message M and encrypts it using party B's public key, thereby obtaining ciphertext as signified by $M_B = E_B(M)$. Then the ciphertext is further encrypted by encrypting the concatenation of the triplet M_B, and the identifiers A and B. This second encryption process of the inner message, M_B, is called super-encryption. It provides an added layer of "conditional access" such that one who removes the outer envelope(i.e., the second layer of encryption) cannot read the inner message M. conversely, one having the key to the inner envelope (i.e., the first layer of encryption) cannot act until the outer layer has first been removed. The super-encrypted message is indicated by $M_0 = E_{PO}(A \mid B \mid M_B)$.

Upon receiving M_0, in step 104, party B signs the super-encrypted message and forwards their signature to A as a receipt (as indicated by $R = S_B(M_0)$). If A receives a properly signed receipt, then A will forward to B, in step 106, the inner message $E_B(M)$. If B does not receive the inner message from A within a pre-determined period of time, a recovery process is needed where B requests assistance from the invisible post office. In step 108, B forwards the super-encrypted message, M_0, plus its signed receipt, R, to the post office. If B's signature on R is correct, the post office will then, in step 110, send to B the inner message, $E_B(M)$. Then, in step 112, the post office will send to A, B's signed receipt R. The above-described Micali protocol (steps 102-112) is summarized in **Table 2** below. While only the transmissions between party A and B are labeled "steps," it will be apparent to one skilled in the relevant art(s) that the intermediary steps shown in **Table 2** are also part of the Micali protocol.

PARTY	ACTION	STEP
A:	$M_B = E_B(M)$	
	$M_0 = E_{PO}(A \mid B \mid M_B)$	
A→B:	M_0	102
B:	$R = S_B(M_0)$	
B→A:	R	104
A:	Verify R (sig of B over M_0) -- stop if invalid	
A→B:	M_B	106
B:	Uses K^{+}_{PO} and "A, B" Info to reconstruct M_0	
	Verify M_0 is the same as one first received -- stop if not same	
	$M = D_B(M_B)$	
B:	If A fails to send M_B, B applies to PO for help:	
B→PO:	R, M_0	108
PO:	Verify R (sig of B over M_0) -- stop if invalid	
	Decrypt PO Data = $D_{PO}(M_0)$ = "A, B, M_B"	
	Verify A and B are proper parties	
PO→B:	M_B	110
B:	Uses K^{+}_{PO} and "A, B" Info to reconstruct M_0	
	Verify M_0 same as one first received -- stop if not same	
	$M = D_B(M_B)$	
PO→A:	R	112

TABLE 2 - PRIOR ART

Given the ECM protocol described in Micali, the receipt of the message by B and the receipt of the signed receipt by A are "logically simultaneous," while the post office never views the message in the clear but only recovers and forwards the ciphertext $E_B(M)$. Furthermore, if A and B deal honestly with each other during their electronic transaction (i.e., protocol 100), only steps 102-106 are needed and the post office remains "invisible." Else, steps 108-112 are needed. These steps (i.e., steps 108-112) are known as the "recovery" process.

Despite the "logical simultaneity" addressed by Micali, as described above, shortcomings still exist in electronic mail communications aimed at quick and secure financial electronic commerce transactions. Currently, there exist no electronic mail protocol that facilitates recipient B's reconstruction of the initial message, eliminates full resend of the message, minimizes communication during the recovery step, eliminates the double (super) encryption, allows the parties to delegate performance to agents, and/or supports an "L-shaped" model that forces B to recover by default. Further, there exist no protocol that supports integrated multi-step transactions and adds declared value and/or other extensions.

Therefore, what is needed is a method, system and computer program product for providing enhanced electronic mail services that meets the above-mentioned needs.

Summary of the Invention

The present invention is directed to a method, system, and computer program product for providing enhanced electronic mail services. The system includes a plurality of user sites, and a central post office (i.e., trusted third-party) complex. The method includes an enhanced electronic mail protocol with several embodiments that allows, for example, facilitation of recipient's reconstruction of electronic mail messages, eliminates full resend of messages, minimizes communication during the recovery process, eliminates the need for super encryption, and allows parties (both recipients and senders) to delegate performance to (proxy) agents.

Features and advantages of the invention as well as the structure and operation of various embodiments of the present invention are described in detail below with reference to the accompanying drawings.

Brief Description of the Figures

The features and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference numbers indicate identical or functionally similar elements. Additionally, the left-most digit of a reference number identifies the drawing in which the reference number first appears.

FIG. 1 is a flow diagram illustrating a conventional simultaneous electronic transaction;

FIG. 2 is a block diagram representing the overall system architecture according to a preferred embodiment of the present invention;

FIGS. 3A and 3B are flow diagrams illustrating the use of agents within a protocol according to a preferred embodiment of the present invention;

FIG. 4 is an L-Shaped Model of a protocol according to a preferred embodiment of the present invention;

FIGS. 5A and 5B are a flow diagrams representing an enhanced electronic mail protocol implementing billing and policy signals according to an embodiment of the present invention; and

FIG. 6 is a block diagram of an exemplary computer system useful for implementing the present invention.

Detailed Description of the Preferred Embodiments

I. Overview

The present invention relates to a method, system, and computer program product for providing enhanced electronic mail (EEM) services. As the field of electronic commerce expands rapidly within the Internet environment, business-to-business, customer-to-supplier, and business-to-consumer communications require not only simultaneity, but also security against many operation risks. The use of EEM allows electronic commerce participants (i.e., "users") to exchange actual value (e.g., electronic cash and checks), confidential information, formal notifications, orders, etc. while addressing both simultaneity and security. Thus, in a preferred embodiment of the present invention an organization provides an infrastructure, protocol, and facilities so that electronic commerce participants may utilize EEM to address their electronic commerce communications needs. More specifically, the EEM providing organization would furnish users with software and documentation to implement a particular embodiment of an EEM protocol, maintain one or more trusted third-party (post office) facilities, provide user customer service and support, as well as provide customer billing.

The present invention is described in terms of the above example. This is for convenience only and is not intended to limit the application of the present invention. In fact, after reading the following description, it will be apparent to one skilled in the relevant art how to implement the following invention in alternative embodiments (e.g., various electronic payment schemes, delivery of legal papers and summonses, tax filings, military orders, etc.). Furthermore, it will be appreciated by one skilled in the relevant art(s), that the protocol described herein can be applied to many other communications besides electronic mail. For example, the present invention may be applied to electronic data interchange (EDI), communication system protocols such as the Simple Network Management Protocol (SNMP), electronic process or machine control protocols, electronic bidding or auctions, etc.

It should be noted that the symbol "M" or term "message," as used herein can refer to any digital data, or may be null (e.g., a message of zero length containing no data). In addition, M can refer to a set of files that are being sent as a group, which in existing secure mail systems, such as NetDox™, is sometimes called a "package." This is analogous to a conventional express mail envelope that may contain several different documents or attachments. Most secure and non-secure mail or messaging systems, in practice, may involve a given message that contains any number of attachments. These are typically referred to as "attached files" having stated or recommended file names and file types, which can be "detached" and reconstituted as named files on the receiving computer system.

Furthermore, in alternative embodiments of the present invention, M may also contain a pointer to a message, file, or document. For example, M may be a uniform resource locator (URL) address pointing to where the actual message, file, or document is stored. It is preferable that the URL be accompanied by a hash value of the message, file, or document to which the URL is pointing, and potentially also a decryption key which can decrypt it after it has been retrieved, and optionally a user ID and password with which to retrieve it. In this manner, the present invention could be modified such that B, upon signing the receipt R, opens M_B which preferably contains an URL, associated hash value, decryption key, user ID, and password. B can then use those values to retrieve the message, file, or document and decrypt it, and verify that the content matches the hash value.

II. EEM System Architecture

Referring to **FIG. 2**, a general enhanced electronic mail (EEM) system architecture 200 for implementing the present invention is shown. It should be understood that the particular system architecture 200 in **FIG. 2** is provided as an example of the preferred embodiment of the present invention and is not intended to limit the scope of the invention. The system architecture 200 includes a plurality of users (or participant) "sites" 206 (shown as sites 206a-206n in **FIG. 2**). A user A or user B, as used herein, refers to a party who can access enhanced electronic mail (EEM) service provided by a EEM organization as explained above. Each user of the EEM system architecture 200 would employ a user mail agent 208, a user database 210, and a user set-up and administration application 212. Of course, any given user can normally both send and receive EEM messages.

Each user site 206 is linked to the Internet 220 via an Internet Service Provider (ISP) 204 or in large companies an internal enterprise Service Provider (ESP) with substantially similar functions. A user ISP proxy 202 is also linked to the ISP provider 204 and constitutes a value added service of the ISP (or ESP) 204 whereby the user will appoint the ISP (or ESP) 104 to assist him or her in completing one (or both) sides of the user processing for the EEM protocol. This is desirable in view of the potential delays in completing the protocol (due to uncertain transit times) and the propensity of users to turn off their computers making them unavailable to complete the protocol.

A trusted third-party Post Office (PO) "complex" 240 is furnished by the EEM provider. The PO complex 240 provides an infrastructure support for the "invisible" protocol, and administrative functions so that electronic commerce participants may utilize EEM services. As mentioned above, the PO complex 240 remains 'invisible" to the users until any recovery features of a EEM protocol are required.

The PO complex 240 includes a customer service Web site server 222 that allows users to access their accounts for billing, receipts, and other like services. The PO complex 240 also includes an EEM recovery daemon 224, a billing information server 226, a developer Web site server 228, a PO database 230, and an administrative workstation 232. The administrative workstation 232 allows EEM provider personnel to monitor the performance of the EEM recovery daemon 224 process and perform various administrative tasks associated with providing EEM services.

The functions of the individual components of the PO complex 240, as well as the EEM system architecture 200 in general, are described more fully below. Furthermore, while only one PO complex 240 is shown in **FIG. 2**, it will be apparent to one skilled in the relevant art(s) that a plurality of PO complexes 240 (each maintained by the same or a different EEM provider) may be connected to the Internet 220, all using interoperable embodiments of the EEM protocol of the present invention under a "technical standard." This is described in more detail below.

III. Efficiency Enhancements of the Present Invention

The system, method, and computer program product for providing enhanced electronic mail (EEM) services of the present invention provides benefits not found in the Micali ECM protocol described above with reference to **FIG. 1** and **Table 2**, nor in the conventional electronic commerce services. The different embodiments of the present invention provide different enhancements to the Micali ECM protocol 100. The following discussion highlights these enhancements and their respective benefits with reference to the EEM system architecture 200 (as shown in **FIG. 2**).

A. Facilitates B's Reconstruction of Initial Message

In one embodiment of the present invention, user B's reconstruction of the initial message, M, is facilitated in comparison with Micali ECM protocol. At the end of each round of the Micali protocol, B must use the purported inner message M_B he has received to attempt to reconstruct the original message M_0, in order to verify whether he now possesses the actual message for which he has signed. The results of encrypting M with K^+_{PO} can be deterministic, when the "full message recovery" public key encryption method is used, but this is impractical for messages of normal size, in which case it is common to "envelope" the message with a random DES key and then encrypt that DES key using K^+_{PO}, a process known as key transport (as in the popular RSA-DES mode of operation known as "RSA Key Transport").

Hence if user B is given only the inner message M and K^+_{PO} he cannot reconstruct the outer message M_0, without knowing the values of the random DES key, and any initialization vector, etc. (collectively referred to herein as the "symmetric key bitstring") used by user A. Thus, the present invention allows various ways to provide that information to B which can readily enable the "enveloping" method. In a first embodiment, user A may return the bitstring along with $E_B(M)$ on the third pass (i.e., step 106 of **FIG. 1**), preferably also encrypted using K^+_B. In an alternative embodiment, the bitstring may be formed deterministically using bits taken from the inner message, either M or $E_B(M)$ (e.g., H(M) may be used as the bitstring), because these values are unknown to "outsiders." In yet another alternative embodiment, the bitstring values (e.g., a DES key) needed to reconstruct M_0 may be included in a separate field inside the inner envelope, along with M, such as $E_B($ M, $K_{DES})$, where B can easily obtain it.

B. Eliminates Full Resend of Message

As disclosed in the Micali ECM protocol 100, after A receives the receipt, user A re-sends the inner message $E_B(M)$ (step 102 of **FIG. 1**), in its entirety. Since this inner message is almost as long as the original super-encrypted message, M_0, it may be inefficient, especially for very large messages, such as delivery of video content, etc. For M_0, the outer envelope is encrypted using K^+_{PO}. However, for speed, this outer encryption will always be performed via a fast symmetric algorithm, such as DES, with only the DES key being wrapped under K^+_{PO}.

In one embodiment of the present invention, upon receiving R, party A more efficiently sends B a second copy of that DES key, preferably in the form of another message addressed to B. That message may also contain a linkage indicator (e.g., the hash of the original inner message) and be encrypted using that same DES key formerly under K^+_{PO}, except now that same DES key is being sent to B under K^+_B. In an alternative embodiment, A may send the third message encrypted under K^-_B using yet another random DES key, yet containing the original DES key and the hash as a separate short message, such as:

$$A \rightarrow B: E_B\,(\,K_{DES} \mid H(M)\,)$$

In yet another alternative embodiment, as all communications from A bear an external digital signature of A, the third message from A→B may be a "detached substitute mail header" addressed to B, that may be merely affixed onto the front of the original message by B, such that B is now made a recipient of the original message, whereupon he opens the outer (PO) envelope as if it had been originally addressed to him. Both headers may be stored with the message for archival purposes.

With each of the above embodiments, the inner message remains encrypted (i.e., double encrypted) such that even if the PO 240 comes into possession of, and decrypts the outer envelope, the inner message to B remains unreadable by it. This enhances B's confidentiality and decreases the PO's potential liability for improper disclosure.

C. Eliminates Double Encryption, Substituting a Split Key Scheme

Based on the above description of the Micali ECM protocol 100, an embodiment of the present invention that does not require double encryption (or super-encryption) of M (as seen in step 102 of **FIG. 1**) can be implemented. Thus, the PO 240 recovery process is reduced to one which B merely sends the header and signature blocks of M_0, without sending the actual super-encrypted message body.

With M being encrypted only once, using a DES key that is a blended value (e.g., using the XOR "⊕" operation) of two DES keys, one available to both user A and the PO, and the other available only to B. This allows the final decryption of M to remain under B's sole control, whether he receives the other key from A or from the PO 240 under the recovery process.

A sends M_0 originally with a 2-addressee encrypted mail header. B can open his side of the header, but now he gets only a partial DES key K_{DES1}. Upon signing the required receipt R, he is then sent the other partial DES key K_{DES2}, as a detached mail header, addressed now to himself, B. By now combining the 2 DES keys, B can decrypt M in a single step. The advantage of the embodiment, in this respect (i.e., single-step decryption), can be summarized as shown in **Table 4** below. (Random number and linkage values are omitted for clarity.)

PARTY	ACTION	DESCRIPTION
A:	$K_{DES1} = \text{Rand1}$ $K_{DES2} = \text{Rand2}$	A generates 2 random numbers
	$K_{DES} = K_{DES1} \oplus K_{DES2}$	
A→B:	$M_0 = [\ E_{PO}(\ K_{DES1} \mid A \mid B\ ) \mid E_B(\ K_{DES2}\ ) \mid E_{DES}(\ M\ ) \mid S_A(\ Q\ )\ ]$	
B:	$R = S_B(\ Q\ )$	$Q = M_0$ minus A's signature
B→A:	R	
A→B:	$E_B(\ K_{DES1}\ )$	
B:	$K_{DES} = K_{DES1} \mid K_{DES2}$	B can now open and read M
	If B requires intervention from the PO to recover the message, then:	
B:	$R = S_B(\ Q\ )$	$Q = M_0$ minus A's signature
B→PO:	$R \mid M_0$	
PO:	Verify parties and verify R is B's signature over M_0	
PO→B:	$E_B(\ K_{DES1}\ )$	
B:	$K_{DES} = K_{DES1} \oplus K_{DES2}$	B can now open and read M

TABLE 4

Except for very long messages, this above-described advantage of the present invention does not provide a large computational advantage for the average user. This is becauѕe DES is a relatively fast algorithm and decrypting M twice does not generally consume noticeably more resources than decrypting it only once. This is especially true if the Micali protocol is implemented such that both the outer and inner decryptions are processed simultaneously, on a streaming basis, eliminating the need to fully buffer the intermediate result. This single decryption embodiment of the EEM protocol, however, does provide a significant advantage to a party that must rapidly process a large number of long messages.

D. Minimizes Communication During Recovery Step

As disclosed in Micali and shown in **FIG. 1**, when A fails to complete the ECM protocol 100, B sends to the PO 240 both B's receipt and the entire original message. The PO 240 opens the original message, checks the ID's of A and B, verifies the receipt of B, and sends back the inner message. By using am embodiment of the present invention, the same effect may be achieved with much less communication, albeit at the cost of deviating from and thus, modifying the standard mail header formats.

If all information needed for recovery was contained in the message header created by A, then B may receive service from the PO 240 by merely sending that detached header to the PO, along with B's receipt, rather then B having to send M_0 in its entirety to the PO. This reformatting seeks to: (a) provide the PO 240 enough information to permit recovery without sending M_0 in its entirety, (b) convince that PO 240 that A really sent M_0, and (c) provide a receipt R signed by B that will be convincing to A. This can be achieved by formatting M_0 in detail as follows:

$$M_0 = [\{ E(K^{+}_{PO}, (K_{DES} \mid (A \mid B \mid H(M)))) \mid E(K_{DES}, E_B(M))\} = Q \mid S_A(H(M) \mid H(Q))]$$

$$\text{where } Q = \{ E(K^{+}_{PO}, (K_{DES} \mid (A \mid B \mid H(M)))) \mid E(K_{DES}, E_B(M))\}$$

Rather than using as a header $E(K^{+}_{PO}, K_{DES})$ as is more common in contemporary secure messaging protocols, A adds "A | B | H(M)" into the area where normally just the DES key for the message would be encrypted.

Similar to the embodiment shown in **Table 4**, Q represents the header and message body up to but not including the double signature. This example envisions an RSA-type full message recovery signature process. whereby the signature verification step returns the original and complete values of H(M) | H(Q), as known to have been signed by A.

Due to the long modulus lengths of many public key encryption algorithms, there may be plenty of space available to fit both hash values. For example, as will be apparent to one skilled in the relevant art(s), with a 1024-bit modulus, the size of a message which can be fully recovered is 128 bytes. After deducting 24 bytes for a 3-DES key and 20 bytes for the SHA-1 output of H(M), this leaves 84 bytes to hold the ID's of the 2 parties. Longer modulus lengths provide even more such "cargo" space. As is well known in the relevant art(s), SHA-1, is the Secure Hash Algorithm developed by the U.S. National Institute of Standards and Technology (NIST) that maps bitstrings to 160-bit hash-codes.

The preferred method to form the double signature of A over both M and Q is to make H(M) and H(Q) into "signature attributes," which may then fit within a signature block as defined by most popular technical standards.

As a result of the present invention's enhancements (to the Micali system), when B seeks recovery from the PO 240, he can omit the main message, and send merely the message header and A's signature block, along with his own receipt R = S_B(Q). Because, Q is the entirety of the original message header and body, S_B(Q) convinces PO 240 that B has really signed a valid receipt, since PO 240 can obtain the original value of Q from A's detached signature block. It should be noted that if the value for Q in A's signature does not match the message, the message was malformed (and unsigned) and B should have rejected it without ever signing the receipt.

After verifying both A's signature and B's receipt, the PO 240 opens the header and obtains the inner values, which include most or all the information it would have gotten from opening M_0 under the Micali system. The PO 240 checks the ID's of A and B, and retrieves H(M) which is the hash of the inner message which it has not received. However, it can check this H(M) against the one that it has recovered by verifying A's signature block, which provides adequate comfort to the PO 240 that A really did send the message that B is seeking to recover. After making these two checks, the PO 240 is ready to allow B to recover the message (which is still in B's possession) and the PO 240 does this by sending $E_B(K_{DES} | \text{Info})$ to user B.

In a commercial implementation, the EEM protocol of the present invention would include various methods to easily link all message components, including the placement of sequence numbers, record ID types (e.g., object identifiers), date-time stamps, and random number values in all related message components, as will be apparent and is well known to those skilled in the relevant art(s).

All users of system architecture 200 are equipped with public-private key pairs and certificates, and sign all protocol messages that they originate. Certificates to enable verification of these signatures are either sent along with the messages, or made available via a readily accessible directory service. Where needed, unique names or ID's would be provided for all system architecture 200 participants. Each message and signature may also contain or refer to a policy-ID that contains or points to a legal policy or rule that governs the use or interpretation of that message or component.

This enhancement of the present invention has the further advantage that the PO 240 never possesses any form of the inner message. M. which may further limit its risks of legal liability to either party for accidental disclosure of M. However, as before, the PO 240 still receives the identities of A and B, which expose it to possibly unwanted "traffic analysis" data about its clients.

E. Combined Split Key and Minimized Communication EEM

It is desirable to provide a version of the EEM protocol that supports both a split key access to a "double condition" encryption layer (or multi-condition access) as described in Section III.C above, as well as minimized communication with the PO 240 (compact recovery) in the event intervention is required by the PO 240 to provide recovery service for B as described in Section III.D above. An embodiment which combines the two methods described above is the EEM protocol presented in **Table 5**.

PARTY	ACTION	DESCRIPTION
A:	$K_{DES1} = Rand1$ $K_{DES2} = Rand2$	Generate 2 random numbers
	$K_{DES} = K_{DES1} \oplus K_{DES2}$	XOR to produce message encryption key
	$M_B = E_{DES}(M)$	Encrypt M using combined key
	$HDR_{PO} = E_{PO}(A \mid B \mid H(M) \mid K_{DES1})$	PO gets K_{DES1}
	$HDR_B = E_B(H(M) \mid H(K_{DES1}) \mid K_{DES2})$	PO gets K_{DES2} plus hash of K_{DES1}
	$Q = (HDR_B \mid HDR_{PO} \mid M_B)$	The basic unsigned message
	$M_0 = [\,(HDR_B \mid HDR_{PO} \mid M_B) \mid S_A(H(M) \mid H(Q)) \mid H(HDR_{PO})\,]$	H's in S_A are signature attributes
A→B:	M_0	
B:	$R = S_B(H(Q) \mid Info)$	Note: $Q = M_0$ minus A's signature
B→A:	R	
A→B:	$E_B(K_{DES1} \mid Info)$	
B:	$K_{DES} = K_{DES1} \oplus K_{DES2}$	B can now decrypt M_B
	Alternatively, if A fails to complete the protocol:	
B→PO:	$[\,R \mid HDR_{PO} \mid S_A(H(M) \mid H(Q) \mid H(HDR_{PO}))\,]$	
PO:	Verify A's signature over HDR_{PO}	
	Verify B's signature on R	
	Verify H(Q) in both S_A and R	
	Open HDR_{PO} and check party IDs	
	H(M) from HDR_{PO} same as H(M) from A's sig block?	
A→ B:	$E_B(K_{DES1} \mid Info)$	
B:	$K_{DES} = K_{DES1} \oplus K_{DES2}$	B can now decrypt M_B

TABLE 5

This embodiment will be preferable for larger messages where the processing required to decrypt M twice, or retransmit M_0 to and from A and from the PO 240, begin to create long time delays and other inefficiencies.

As with the "PO header only" method, the PO 240 need not receive nor store the M_0, which may be very large, and the double or multi-condition layer concept is preserved so that the PO 240 cannot read M without the active participation of B. The private decryption key of B is also required, thus minimizing the PO's potential liability for improper disclosure of M as in the basic Micali protocol.

Certain new elements have been added to this embodiment of the protocol, as shown in **Table 5**. First, the signature of A, S_A, over M_0 is enhanced to include signature attributes, which are delivered along with the signature block, containing separate hash values for: (1) H(M) - placing the hash of the inner content, M, into the signature links A to M in the eyes of both B and the PO 240; (2) H(Q) - A signs over the entire message; and (3) H(HDR_{PO}) - A signs the PO header separately, to enable compact recovery, so that it is enough to send to the PO HDR_{PO} and S_A, without having to send the much larger M_B. If the PO header is not signed separately it cannot be detached from M_0 and sent to the PO 240, as the PO 240 cannot verify it separately from M_0.

Second, B's header, HDR_B, along with K_{DES2}, also includes H(M) and $H(K_{DES1})$. H(M) further links B's header to the content to which B receives access when he signs and returns the receipt. $H(K_{DES1})$ helps to inform B whether A (or PO) has sent him the correct second access key, and its presence in the header, signed over by A within Q, constitutes a representation by A that a symmetric key hashable to that value will be provided, breach of which will cause B's receipt to become legally void.

Third, an "Info" element has been specified at various points. This represents generic information that a competent protocol designer or programmer would consider useful for tracking messages, transactions, and their components, matching and reassembling separated components, creating or updating entries in the local databases of each participant, generating payment system messages and records of monetary payments and balances, reconciling each message component with any pre-existing entries, etc. This topic will be covered in much greater depth elsewhere, but some potential elements of the "Info" field include, but are not limited to: (1) each participant's system or user generated message tracking sequence reference numbers; (2) the date and time of the last action, and any prior actions for the same message; (3) hash values or sequence numbers of any prior or related message components; (4) random numbers or other nonce values used to link components together; and (5) software and protocol version release numbers and related version data.

As will be apparent to one skilled in the relevant art(s), any symmetric cipher, other than the DES algorithm specified herein, suited to securely encrypting comparably sized blocks of information may be used with equal efficacy.

There are many different methods to generate a two-party or multi-party conditional access scheme. The idea of creating two fragmentary DES keys being only the simplest. Another approach would be to utilize a threshold encryption system, in which the parties A, B, PO, etc. must act in concert (on a two of three basis) to gain access. Another would be to integrate these methods with a key-agreement methodology, such as Diffie-Hellman, such that A and B can each contribute some random material, and "agree" on a symmetric (e.g., DES) key, thereby achieving multi-party conditional access.

F. Key Agreement Protocol Integration

In another embodiment of the present invention, a key agreement protocol, such as the above-mentioned Diffie-Hellman (D-H) protocol which is well known in the relevant art(s), can be introduced into the EEM protocol. This embodiment permits both A and B to contribute random material to the symmetric key that will be used to encrypt M, the content. This approach is preferred in a military or intelligence grade communication system because the recipient B may not trust A to wisely, competently, or honestly choose a highly random symmetric (e.g., DES) key. Hence B wants to "participate" with B in generating the message key.

The D-H protocol, is described in detail in Schneier cited above. For completeness, however, the D-H protocol is described briefly herein. First, A and B both pre-agree to use two system-wide values: (1) g--the base; and (2) P--the modulus, which should be a prime number. Then A and B both choose random numbers (x and y, respectively) and perform the steps outlined in **Table 5**.

PARTY	ACTION	DESCRIPTION
A:	$X = g^x \bmod P$	X is also sometimes called "DH1"
B:	$Y = g^y \bmod P$	Y is also sometimes called "DH2"
B→ A:	Y	Y can also be delivered to A via B's certificate
A:	$K = Y^x \bmod P$	A uses Y and x to generate K (a symmetric key)
	C = Encr(K, M)	use K to encrypt message, forming C (ciphertext)
A→ B:	C | X	send C to B, along with X
B:	$K' = X^Y \bmod P$	B uses X to generate K' = K
	M = Decr(K', C)	decrypts the ciphertext and reads the message

TABLE 5

In an embodiment of the present invention, D-H is used as the public key encryption method for the basic Micali protocol 100. This is illustrated in **Table 6.**

PARTY	ACTION	DESCRIPTION
A:	x = random $X = g^x \bmod P$	A generates a one-time D-H private and public key pair for this message
B:	y = random $Y = g^y \bmod P$	B's keys, could be fixed or one-time
B → A:	Y	Y can also be delivered to A via B's certificate

PARTY	ACTION	DESCRIPTION
A:	$K = Y^x \bmod P$	Create K using Y and x (A's temp private key)
	$M_B = Encr(K, M)$	Encrypt M using K
	$M_0 = E_{PO}(A \mid B \mid M_B \mid X)$	
A → B:	M_0	
B:	$R = S_B(M_0)$	
B → A:	R	Preferably a detached signature
A:	verify R	
A→ B:	$M_B \mid X$	Send M_B per basic version, along with X
B:	Use K^+_{PO} and "$A \mid B \mid M_B \mid X$" Info to reconstruct M_0	
	Verify M_0 same as one first received stop if not same	
	$K' = X^y \bmod P$	B uses X and Y to generate K' = K
	$M = Decr(K', M_B)$	Decrypts the ciphertext and reads the message

TABLE 6

If A fails to send M_B, B's recovery process with the PO proceeds as before. Note that the embodiment presented in **Table 6** integrates the well-known D-H public key encryption with protocol 100, at a point where that protocol requires use of a public key encryption scheme.

The D-H public key encryption scheme can also be used for the PO 240 encryption layer, in which case, the PO would generally be required to publish its D-H public key in its public key certificate to permit multiple entities to use the services of the PO 240 over a period of time.

Many variants of the D-H scheme exist, in which the parties may generate one or more sets of temporary intermediate public and private keys to minimize exposure of their long term keys, and preferably the output of the D-H function is also passed through a hash function (keyed or unkeyed) to avoid any direct use of the actual D-H output material.

In alternative embodiments, other "key agreement" protocols may be used. D-H, however, is the best known. An embodiment that employs D-H will be suitable for military and national security users, where B typically desires to participate with A in generating K_{DES}, to assure a high quality key.

G. Multi-Condition Access via Threshold Cryptosystems

According to the basic Micali protocol 100, as enhanced by the split key embodiment described in Section III.C above, it is desirable to have A send to B an initial message M_0 that is readable upon the occurrence of 2 fixed conditions: (1) access granted by A (or the PO) upon getting B's receipt R; and (2) subsequent access by B's private key. The guarantee of access by a named PO assures B it is okay to sign the receipt R without fear of cheating or non-performance by A, and the requirement of further access by B assures B that only he can read the inner message M in any event.

In another embodiment, similar results can be achieved using a threshold cryptosystem in which any two of three parties can act together to gain access to the inner message M. Thus A and B can together grant access to B in the normal case, and in the recovery case, the PO 240 can act together with B to grant B access to M.

Party A should utilize an encryption scheme that can allow "2-of-3" access by A, B, and the PO to decrypt the inner message. This may require that the three parties initially generate a temporary master public key for any given message (or sequence of messages involving these three parties) that can be accessed by any two of their three private keys (which may be temporary or long term). Also, it will be desirable to select a threshold scheme that permits each party to perform their part of the "access" computation separately, forwarding their partial results on to the next party for completion.

It may also be possible to utilize a threshold computer system in which each party generates a public-private key pair, and then (in this case) the three ("little") public keys are combined, to produce a master public key that can encrypt data where two of the three participants may act together to decrypt the data.

1. First Threshold Scheme

In a first embodiment, to send a message, A generates a symmetric message encryption key, typically a random number K_{DES}, uses K_{DES} to encrypt the inner message M, and then wraps K_{DES} using the 2-of-3 master public key K^{+}_{3}. Generally A also wraps the recovery information "A | B | Info" for the PO in a field that can be read by the PO without B's assistance, thus yielding, in a first embodiment the scheme in **Table 7**.

PARTY	ACTION	DESCRIPTION
A:	$M_B = E(K_{DES}, M)$	Encrypt M with random DES key
	$KK = E_{K3}(K_{DES})$	Vrap K_{DES} with master public key K^{+}_{3}
	$M_0 = [\ E_{PO}(A \mid B \mid Info \mid H(M) \mid H(K_{DES})) \mid KK \mid M_B\ ]$	
A →B:	M_0	
B → A:	$R = S_B(H(M_0) \mid Info)$	
A:	$PD_A = partial\_decrypt(K^{-}_{A}, KK)$	
A → B:	PD_A	
B:	$K_{DES} = final\_decrypt(K^{-}_{B}, PD_A)$	
	$M = D(K_{DES}, M_B)$	
	Or if A fails to perform, then PO can step in and perform as follows:	
B:	$R = S_B(H(M_0))$	
B→ PO:	$R \mid M_0$	See Note***.
PO:	$PD_{PO} = partial\_decrypt(K^{-}_{PO}, KK)$	
PO→ B:	PD_{PO}	

PARTY	ACTION	DESCRIPTION
B:	K_{DES} = final_decrypt(K^-_B, PD_{PO})	
	M = D(K_{DES}, M_B)	

TABLE 7

Note***: Preferably, this method will be combined with the "eliminate full resend" embodiment described above in Section III.B. However, it is shown in a simplified form, with a full send of M_0 to the PO, to reduce the complexity of the protocol and make it easier to see the use of the threshold enhancement.

2. *Second Threshold Scheme*

In a second closely related embodiment, rather than encrypting the PO header using only the key of the PO 240, A might separately encrypt both KK and the PO header using the same scheme, thereby requiring B's involvement for the PO 240 to access the PO header. This would strengthen B's control over the recovery process.

As will be apparent to those skilled in the relevant art(s), A can encrypt both M_B and the PO header using the shared master public key K_3^+, such that the participation of at least two parties (of A, B, and C) is required to access both M_B and the PO header. Access to the PO header is not required unless B requests recovery by the PO 240. If B requires assistance from the PO 240 to recover the message, he sends R and the PO header to the PO 240, along with his own partial decryption of the PO header access point, PD_B (analogous to the quantity PD_A in the prior example). The PO 240 then performs a final decryption of PD_B to recover the symmetric (e.g., DES) key giving access to the PO header.

H. Provides Explicit System Enrollments by User CA's

The present invention, with an "invisible" PO 240, provides a convenient method for A and B to exchange a message, with A getting strong proof of receipt from B, and B getting strong assurance that he will get the message, without routine involvement by any third party. However, when recovery is desired by B, in the event of A's non-performance, it will be desirable for all system architecture 200 participants to have an unambiguous way to identify all the other possible participants, the role they will play, the services they will provide, for whom, the fees that will be billed and paid, and who will be liable for any damages in the event of non-performance or system failures. Most importantly, prior to signing the receipt, B will want to know that the PO 240 selected by A in fact stands ready to provide reliable recovery service to B if requested, at a reasonable price.

The PO 240 may be chosen by either A or B. A might prefer a PO 240 that charges a lower fee for a recovery that would be billed to her, or one located in the same country as her and subject to the same national laws. In contrast, B may prefer a PO 240 he deemed more reliable, in the event he needed to look to it for recovery, or possibly a cheaper one, if he would be required to pay the recovery fee himself. Also, if A initiated a given message using the key of a given PO, where that PO 240 required either A or B to be a member to receive recovery services, then each needs to know that the other is in fact a member of the particular PO 240 chosen.

Alternatively, the PO 240 choice may be imposed by a third party, such as the employer of A or B, or a PO 240 might refuse service to some class of users or messages. In a non-public system, such as the proprietary corporate electronic mail system of a given company, the system may be a closed system, such that the PO 240 cannot recover messages unless one or both of the parties are employees, agents, customers, or suppliers of that company, and their certificates clearly evidence that relationship (e.g., the user's e-mail addresses must belong to the same domain name as the company PO).

Also, there is a variety of ways to allocate the cost of recovery services, as well as any per message costs that may be billed by the system, even when recovery is not requested, and the parties need to know how these costs will be allocated. These allocations may be specified by A upon sending the message, in an area readable to B, or they may be pre-coded into A's certificate.

I. Support Multiple Senders and Recipients

The present invention allows multiple recipients and/or senders of the same message, which may further enhance the basic Micali protocol 100.

1. Multiple Recipients

The recipient's name inside and outside of the initial message M_0 can be enhanced to contain a list of several possible recipients $[B_1, B_2 \dots B_N]$, any or all of whom can sign a receipt R and individually complete the protocol with sender A, and/or request and receive recovery services from the PO 240. This may be expressed in notational form as:

$$M_B = E_B(M)$$

$$M_0 = E_{PO}(A \mid [B_1, B_2 \dots B_N] \mid M_B)$$

Normally, all intended B's (i.e., recipients) are also listed in the external message header so that the communication system will deliver a copy of M_0 to each B_X. Then each B_X sends his receipt R_X to A, and A individually completes the protocol with each B_X by sending him $E_{Bx}(M)$, which is M encrypted with B_X's public key, and each B_X decodes and reads the message M.

In cases where there is a separate header and possibly a partial DES key for each B_X, then each such header will normally be encrypted using B^+_X, the public key of B_X and will contain the same partial DES key. Alternatively, each header may contain a different partial DES key in which case, A must retain the matching partial DES key to be sent to recipient, and place each separate partial DES key into the PO header, associated with the name and ID data of each B_X. A may also create multi-way key splits that would require two or more recipients to combine their partial DES keys and act in concert to access the message M.

2. *Multiple Senders*

Normally a requirement for multiple senders is not anticipated. However, this embodiment of the present invention is included for completeness. A message could also be originated by a group of senders, $A_1, A_2 \ldots A_N$, such that each one represents to B that she can complete the protocol and deliver $E_B(M)$ to B upon receipt of R signed by B. This is expressed in notational form as:

$$M_B = E_B(\ M\ )$$

$$M_0 = E_{PO}(\ [A_1, A_2 \ldots A_N] \mid B \mid M_B\ )$$

Further, in this embodiment, upon receiving B's receipt R, and after completing the protocol on the A_X's behalf, by returning $E_B(M)$ to B, the PO would preferably forward a copy of R to each A_X.

If it were an objective to guarantee that each A_X gets a copy of B's receipt R, then the L-shaped model described in Section IV.C below might be preferred. This is because the L-shaped model would force the process to flow through the PO 240 which can be relied on to provide each A_X with a copy of R. Alternatively, if the L-shaped model is not used or required, then all of the A's must agree among themselves that whichever one of them receives R she will send a copy of R to all the other A_X's.

Yet another approach might be to require by contract that B will send a copy of R to all the A_X's. Then to avoid getting N copies of $E_B(M)$, B (or the A's) might designate one A_X as the lead A_X to complete the protocol by sending a copy of $E_B(M)$. This could be further enhanced by causing the message M_B to be encrypted using a DES key that is generated by securely combining partial DES keys from each (or some quorum of) senders. That would require B to send the receipt R to all (or a quorum of) A_X's, before each would send B their partial DES key needed to read the message. To recover, B would send a receipt R_X for each A_X to the PO 240.

3. Multiple Post Offices

For various reasons, the parties (i.e., users at sites 206) using the architecture 200 may wish to specify more than one PO 240 that can perform the recovery service, should it become necessary. For example, party B might indicate to party A that he desires A to utilize both a primary and backup PO 240, if the need should arise. Or perhaps A might choose one PO 240 and B another. B might select one that operates using his preferred language and Jurisdiction's laws, whereas A might specify one that she believes offers a more reliable service, and hence is less likely to create a service problem that could reflect badly on A.

Once A and/or B have made an appropriate selection of 2 (or more) POs, A can proceed as usual to construct M_0, except A will create as many additional PO headers as needed, each one encrypted using the public key K^{+}_{PO} of its respective PO 240, and identifying the PO name in readable external form, and containing the recovery information needed by that PO 240. Party A may, if needed, form each PO header using a different version release number and PO header data format of the EEM protocol, as each PO 240 may specify in its public key certificate the EEM version it is currently capable of processing.

4. Multiple Messages for Very Large Files

Users 206 may wish to break very large files into pieces so they may be transported as separate messages. In this case, each separate installment can be handled, signed for, and (potentially) recovered as a separate M_0 with a message M-of-N indicator preferably both inside and outside the outer encryption layer of M_0 for tracking purposes. B would then sign a separate receipt R for each message, and would receive a separate send from A of either the inner message M_B (as in the basic Micali protocol) or the key or key share which allows B to access the message (in the enhanced versions of the present invention). It may also be desirable to place a hash value of the prior message in each successive message, or a cumulative hash value of all the messages sent so far.

In a related embodiment, the EEM protocol may be modified such that B only needs to sign one receipt for all the files in a large related series of transmissions. This may be preferred by B, who would rather not sign for anything until assured that he will in fact receive the entire shipment (i.e., all components). In this case, A might encrypt the entire shipment of messages with a single access key (or key split), and upon receiving B's single receipt, send to B a single message containing the access key (or key split) that grants to B access to read the message.

Likewise the proof-of-decryption (POD) and post-on-send (POS) embodiments described in Section VI below may be further enhanced in accordance with this embodiment so that they are configured to work with either a single setup and fulfillment of each respective protocol for the entire transmission of related messages, or a separate setup and fulfillment of each respective protocol for each individual message within the series.

IV. Proxy Agents

Most large scale business-to-business messaging systems are highly robust and fault tolerant, and can routinely operate reliably on a continuous (i.e., twenty-four hours per day and 7 days per week) basis as needed to participate in the EEM services of the present invention. However, many if not most human users and their desktop or portable systems do not operate in a reliable, robust, and continuous way, and therefore these types of users may wish to delegate their performance of many steps of the EEM protocol to an agent (e.g., their ISP or corporate mail system). In particular, by using an embodiment of the present invention: (a) party A may delegate the step of verifying party B's receipt and forwarding the final message or key; or (b) party B may delegate the process of listening for certified mail and signing the receipts. As will be apparent to one skilled in the relevant art(s), an "agent" can be any entity selected by the primary parties (i.e., users A and B), and that are also connected to the same communications network. Generally, these agent entities will be "trusted" in the sense that they will be subject to contractual (or other legal) obligations to perform. Thus, in the case of nonperformance, these entities will be subject to financial or other legal remedies for breach, improper disclosure of confidential information, etc.

A. Sender (User A) Appointing an Agent

Referring to **FIG. 3A**, an "agent" process "C" is created on the mail server used by user A, to allow A to appoint C as her agent. In one embodiment, when user A sends M_0 to B, she sends C the information needed to complete the protocol in her absence, and then tells B to send his receipt to C. This is relatively easy, as user A need not entrust C with her private signing key, nor with her plaintext message (M).

In regard to user A's use of agent C, at least four embodiments may be distinguished. In a first embodiment, A sends M_0 to B and B's message (E_B(M)), to C, as shown in **FIG. 3A**. C may just intervene directly to receive the receipt from A's mailbox and complete the protocol, where C is a process running on A's mail server which has access to her account, or C may be named in either the message header or in A's certificate as A's "protocol completion agent." In a second embodiment, user A may construct and send both M_0 and E_B(M) to C, and have C send M_0 on to B. Alternatively, in a third embodiment, user A may merely send E_B(M) to C, specifying which PO 240 to use, and let the PO 240 construct and send the M_0.

In yet another embodiment, user A could send the unencrypted inner message M to agent C, telling C to obtain B's key and form E_B(M) and also inform C as to the desired PO 240 whose key (K^+_{PO}) should be used to form M_0. This should preferably be done only if there is a secure link between A and C to avoid potential comprises of the message. Or better yet, A may send M and the identities of B and the PO 240 to C, using K^{+}_{C} to protect it in transit.

In an alternative embodiment, it may be desirable for user A to signal to user B that A is using C as her agent, but this is mainly a processing convenience. That is, A may tell B to send his receipt to C, rather than to A, and B's system may need a suitable message identifier to tell it that a message, $E_B(M)$ really originates from A and is associated to the prior M_0 and R. This would allow B's user software and database to tag and store the messages correctly.

B. Recipient (User B) Appointing an Agent

Referring to **FIG. 3B**, an "agent" process "D" is created on the mail server used by user B in order to allow B to appoint an agent. When B receives a message, D can sign and return the receipt on his behalf, and then receive and hold the inner message when it arrives. Further, if the inner message fails to arrive during a pre-defined time, D may also apply automatically to the PO 240 for recovery of the message for which it has signed.

Where the recipient (user B) appoints an agent, a more troublesome situation arises than when a sending party appoints an agent. This is because B may have to entrust his private signing key to D, to enable D to sign receipts in his name. That problem can be alleviated, however, by B giving D a certificate authorizing D to sign only certified electronic mail receipts for B, using D's private key. This e-mail delegation certificate may be signed by B, or by a CA acting on B's behalf. D may then attach this authorization certificate to its receipt, along with its own certificate, and A (or A's agent C) would accept D's signature, because the authorization certificate causes B to be bound by D's signature on the receipt. (In the field of commercial banking for example, the authority to sign receipts is considered among the "lower" types of risks and therefore is more frequently granted to signing parties.)

When B employs D as his agent to sign the receipt, B's signaling of this fact to A must be more explicit and should preferably convince A that B is legally bound by a receipt R signed by D. There are several ways that B may convey to A information about his delegation of authority to D.

In a first embodiment, B may issue to D an authorization or delegation certificate formally delegating to D authority to sign receipts on B's behalf. This may take the form of a user-issued attribute certificate, signed by B, naming D as the subject, and including possibly D's public key, a hash of D's public key, or the CA name and serial number on D's public key certificate that will be used for signing and verifying receipts. It may also contain an attribute calling out "EEM Receipts" as a class of transactions for which authority is being conferred, and also possibly including an expiration date, and referencing an external document containing a text of all or part of the EEM system 200 rules agreement, and/or any other legal understanding between B, D, (and A).

However, user issued digital certificates are difficult to manage, because the user who grants or delegates may not have enough information (e.g., about some compromise of the agent) or technical capability to effectively revoke them when required. Therefore, in a preferred embodiment, the attribute certificate conferring authority on D will be issued by B's CA (or possibly D's CA, or any other CA selected by the parties), and will contain a representation satisfactory to A that B's consent for D to act on his behalf is on file (and of course that B retains the power to revoke that consent if desired). In this manner, for example, if D's certificate becomes invalid or failed to perform, the CA in question can act quickly to revoke other delegations to D, and publish notice of such revocation.

In an alternative embodiment, B may cause its CA to issue (or reissue) B's base public key certificate with an extension naming D and stating that D will have authority to sign receipts for B. This extension may contain any or all of the elements of the user-issued attribute certificate described above. It may also contain D's public key D^+, so that A will not need to search for it.

In yet another alternative embodiment, D's authority to sign may be listed in a database or directory, publicly accessible using a secure protocol, where A (or C) may look it up when needed. This is similar to the standard practice for corporate authorities in England, where a similar database would merely be hosted on a Web server. It would not involve the creation, issuance, management, or revocation of any authority certificates.

In cases where an authority certificate granted to D is not issued by D's CA, it will be desirable for the issuing CA to contract with D's CA to be notified whenever D's key certificate is revoked, so that the authority can also be revoked. In theory this should not be necessary, since no one should accept a receipt from D once D's key certificate had been revoked, but as a matter of prudence the authority certificates associated with a revoked ID certificate should also be revoked as soon as possible.

C. Supports an "L-shaped" Model That Forces B to Recover by Default (PO serves as A's Proxy)

In some cases a sender A may not wish to complete the protocol, either herself or via a proxy agent (C). A may place a flag in the message M_0 telling B that B's only recourse will be to recover from the designated PO 240.

Referring to **FIG. 4**, an L-Shaped Model 400 according to an embodiment of the present invention is shown. A signals B to recover from the PO 240 by default. B will receive M_0, read this signal flag, and send the message plus his receipt R to the PO 240, which will (in effect) play the role of A's proxy agent to complete A's duties in the transaction.

The PO 240 will verify with the proper parties that M_0 is properly formed and authorized and that R is a valid receipt signed by B or his agent. Then, the PO 240 will (1) send $E_B(M)$ (or the recovery data, from another variant) back to B, and (2) either forward B's receipt R to A, or hold it for A to pick up.

In addition, A may simultaneously signal her intent to accept the recovery charges charged by the PO 240, if any, (i.e., "bill sender"), since she is forcing B to utilize the recovery center (the PO 240) involuntarily. This embodiment may be the preferred approach when A is using an undesirable end-user PC and has no sender proxy agent (e.g., C) of its own. If A expects to use this model repetitively, the PO 240 may give A a bulk discount on the recovery charges.

V. Policy and Billing Signals

A. Post Office Complex Extensions

The present invention, making use of the PO complex 240, is readily extensible by defining inner and outer vectors using a sequence of attributes. Within the well-known Secure Multipurpose Internet Mail Extensions (S-MIME) and the Internet Engineering Task Force's (IETF) cryptographic message syntax (CMS) protocols, these elements can also be placed in the signature blocks of the inner and outer envelopes. As explained below, the sequence of attributes allows the EEM protocol to ensure that all participants (i.e., system 200 users) have a high degree of confidence and make electronic commerce as secure and trustworthy as traditional commerce.

The embodiments of the present invention presented above contemplates abuses and the related concerns that senders and recipients of certified electronic mail (collectively the user sites 206 as shown in **FIG. 2**) may face. For example, before signing the receipt, user B (a recipient) may wish to be assured that: (1) the PO 240 is a valid post office complex that will actually deliver the inner message; (2) the enveloped message was really encrypted using the public key of the PO complex 240; (3) the message will be in a language he can read (e.g., English rather than, say Japanese); (4) there will be no unreasonable or unanticipated message recovery charges; and (5) if a user A has agreed to "accept charges," that A is a valid subscriber of the PO complex 240.

The EEM protocol and EEM architecture 200 of the present invention address these concerns. First, by identifying the PO 240 in the outer vector and providing access to the certificate of the PO 240, and by defining an "is-cem-po" extension in the PO's certificate to allow major certificate authorities (CA's) to certify them, user B is assured that the PO 240 is a valid post office complex that will actually deliver the inner message.

CA's are well known in the relevant arts(s). For completeness, however, a particular certificate authority is an entity (i.e., a service provider) that issues digital certificates to other entities (organizations or individuals) to allow them to prove their identity to others. A certificate authority might be a separate company that offers digital certificate services to the public or an internal organization such as a management information systems department within a larger enterprise.

Second, because user B can not be assured that the enveloped message was really encrypted using the public key of the PO complex 240 before signing the receipt, the participants of the EEM system 200 must agree (e.g., via a EEM System 200 Subscriber Rules Agreement) that the recipient's receipt is not valid or binding if the PO 240 declared by user A in the outer vector cannot actually recover the message, does not exists, etc., or if the inner vector does not match as to A, B, etc.

Third, user B can be assured that the message will be in a language he can read by having the sender declare the language (L) and also provide (e.g., via the EEM System 200 Subscriber Rules Agreement) that the receipt will not bind the recipient if the actual message language differs from the declared language. In an alternative embodiment, the PO complex 240 may allow L to be multi-valued so that a message written in several languages (e.g., a bi-lingual contract) may be sent.

Fourth, user B can be assured that there will be no unreasonable or unanticipated message recovery charges through the EEM protocol. As shown in **Table 3**, B cannot see "$" because it's hidden in the inner vector. However, if A has agreed to "accept the charges" in the outer vector, B has no concerns. Since return receipt service mainly benefits party A (the sender), most serious commercial senders will subscribe to PO 240 and agree to accept recovery charges. If A has declined to pay for any needed recovery, B can decide that he trusts A in the ordinary course of business. In his PO-request B declares a maximum value $ for which he will pay recovery charges. In his receipt, B can state that he has limited the maximum recovery value for which he will pay, without saying to what value. This tells A that B's receipt might not be valid.

Fifth, user B can be assured that A is a valid subscriber of the PO complex 240 when user A has agreed to "accept charges," by assigning a "membership ID" attribute to the certificate to A when user A subscribes to PO 240. In an alternative embodiment, User A may also have this information re-certified into his identity certificate. In A's certificate an extension:

is-cem-subscriber/po=a_cem_co/until=12-31-99

can be defined to enable major commercial CA's to certify EEM subscribers. This is easy where the PO complex 240 is the same entity as the CA.

1. Declared Language and Data Format

The intended recipient B may be unwilling to sign a receipt for an electronic mail message under the EEM system described herein unless provided with a representation by A that the message is in a known language and/or data format that B can properly interpret. B may be unwilling to provide A with a receipt, which A can use to hold B responsible for having read the content, unless he is assured, for example, that the document is in, for example, English (US), and in, for example, Microsoft Word 97 Format.

B needs this information to be available prior to signing the receipt and decrypting M_0, hence it must be provided using one or more attributes outside the encryption of the external PO envelope. For example, in the popular S-MIME or IETF-CMS formats, it could be contained in a signature attribute within A's signature block over the outer envelope.

The system rules agreement will preferably provide that A is bound to honor this promise, and B's receipt R is not valid against B if A has breached this representation as to language and format. This language/format attribute may also be replicated in the "Info" fields inside M_0 and possibly within the receipt R, etc.

A may wish to ascertain what languages and formats are acceptable to B prior to sending M_0. To aid A in making this determination, B may place one or more attributes or extensions in his public key certificate, or may publish such information in a directory that is accessible to A.

2. Declared Value and Time Limit

The PO may wish to require A to state (or declare, or legally represent) the monetary value of the message M, *inter alia* to place a cap on the PO's liability in the event of failure by the PO to perform the recovery operation.

Party A may also wish to state a time limit, "deadline," or "stale date," after which she will no longer be liable for non-performance of an underlying obligation to deliver M to B.

The monetary value field must be placed in the "Info" vector inside the PO layer of M_0, and should also preferably be repeated inside B's message M_B, so B can confirm that what was decrypted by the PO 240 matches what A represented.

To give effect to these declarations, the PO 240 (or the EEM system 200) may charge A and/or B a higher fee to cover the PO 240's added insurance costs for higher value messages, and risk of non-performance in view of a shorter deadline for final delivery.

Such higher fees should preferably be billed on a per-message basis, rather than a per recovery basis, to better reflect the PO's true outstanding exposure to such potential liabilities.

The PO 240 may wish to control which parties can expose it to such liabilities, especially to assure itself that such parties have access to more reliable communications facilities, so the PO 240 need not assume the risk of failures by communications providers over which it has no control.

B. Support Policy Signals to Proxy Agents

The parties (A and/or B) who elect to utilize the services of a proxy agent © and/or D) to assist them in performing the EEM protocol, will wish to exercise care to provide detailed instructions to the agents about how they want certain decisions to be made.

1. Proxy Agent C's Policies of Interest to A

Party A, the sender, will be concerned with the quality of the receipt to be received from party B, the intended recipient, prior to handing over (or granting conditional access to) the inner message M_B. Party B may have many different e-mail accounts and public key certificates. Typically party A will have selected one (or more) specific public key certificates and utilized the certified public encryption keys contained therein to construct M_B and M_0.

One (default) policy is to require that the receipt R be signed using a private key of B whose digital signatures are verifiable using a public signature verification key contained within one of the certificates initially selected by A. (Current secure electronic mail systems, such as NetDox™, generally require that each party have the other party's exact certificate prior to initiating any communication.) This policy of requiring in advance the exact certificate of B, however, will become burdensome as system use grows and as users demand increased flexibility. In addition, if party B decides to use a proxy agent D to sign for him, then the quality of D's signature on the receipt must be addressed.

When party A sends a message to proxy C intending to appoint C to act as A's agent for a given message, or a series of messages, party A may send one or more electronic instructions to proxy C regarding: (1) the type of certificate C may accept from B; (2) ancillary actions to be performed by C; and (3) quality of delegation of authority that C may accept in cases where B may have delegated his receipt-signing authority such as to his proxy agent D. First, party A may send messages to C relating to the type of certificate that C may accept as constituting a signature of B on the receipt R. For example, A may specify one or more of the following rules:

-Require exact certificate match (initial default)
-Allow any certificate with exact user name and employer name match
-Allow any certificate with identical last name and lexically similar first name
-Require that certificate be issued by a specific CA (such as B's employer)
-Require that B's CA adhere to a specified minimum set of policies
-Require that B's CA provide at least a stated minimum financial guarantee on B's certificate
-Require that B's CA be located in a given geographic area (nation or region)

Second, party A may send messages to C relating to ancillary actions to be performed by C on A's behalf. For example A may send C electronic instructions directing C to perform one or more of the following actions:

-Store outgoing message in an archive for N years
-Store returning receipt in an archive for N years
-Notify A if B does not return the receipt within N days, hours or minutes
-Notify some third party X if B does not return R within N days, hours or minutes
-Resend the original M_0 if B does not return R within N days, hours or minutes

Third, party A may send messages to C relating to quality of delegation of authority to sign receipt R granted from B to D that will be acceptable to A. A will be concerned that B must not be in a position to legally deny or disavow the receipt R signed by D. For example. A may send electronic instructions to C directing C to apply one or more of the following rules:

-Disallow delegation and require exact match to B's certificate (default)

-Allow delegation where D is certified by B's employer or B's bank

-Allow delegation where B signs a separate delegation power for D and D uses the same certificate as before

-Allow delegation where D created a separate and unique key pair and certificate, certified properly by B's CA, for the sole purpose of signing receipts on B's behalf.

-Require that B's delegation to D is imbedded as an attribute or extension in B's certificate signed by B's CA

-Require that B's delegation to D is imbedded as an attribute or extension in D's certificate signed by D's CA

-Require that whatever form of delegation D possesses be approved according to a specified set of legal rules, contracts, or agreements

-Require that any delegation certificate or document actually be signed by B

-Require that any delegation certificate or document which is not actually signed by B must instead be signed by an entity that pledges adequate security or collateral to cover any loss that may be sustained by A in the event B successfully disavows a receipt signed by D.

The above three requirements placed on C's activities may also be specified in whole or in part by a system administrator S_A, empowered by A's employer or other organizational sponsor, who may alter or override similar instructions from A. S_A may entirely control the profile of instructions given to C on A's behalf, may specify some things and leave others for A to specify, or may specify, with regard to any given parameter, certain ranges of options that can be selected by A.

2. *Proxy Agent D's Policies of Interest to B*

Party B, the recipient, will be concerned with carefully limiting the delegation of authority to sign receipts for B, to ensure that D does not sign for any messages that B does not wish to receive or acknowledge. First, B may send an electronic instruction to D instructing D not to sign receipts or acknowledgments for messages that appear to have been sent by:

-Specific named senders, or a list of persons or organizations, specified by B

-A list of persons or organizations, specified by reference to a list published by one or more third parties

-Senders whose certificates do not meet a specified level or quality or legal or financial responsibility

-Senders located in certain geographic regions

-Senders whose names may have bad credit or other reports filed against them, or be found on any database of delinquent or suspended persons or organizations

-Senders whose personal or organizational names contain pre-specified words or character strings

Second, B may also send an electronic instruction to D instructing D not to sign receipts or acknowledgments for messages where:

-The title line contains pre-specified words or character strings

-The declared language or declared data format are not acceptable to B

-The declared value (if known) is outside a range specified by B, which may be different for different senders or classes of senders

-The encryption, message digest, key exchange, or digital signature algorithms employed by A in forming the message are not readable by or acceptable to B.

The above requirements placed on D's activities can also be specified in whole or in part by a system administrator S_B, empowered by B's employer or other organizational sponsor, who may alter or override similar instructions from B. S_B may entirely control the profile of instructions given to D on B's behalf, or may specify some things, and leave others for B to specify, or may specify, with regard to any given parameter, certain ranges of options that can be selected by B.

C. Accounting, and System Management via Proxy Servers Enterprise Billing

It will be advantageous to provide a usage billing feature in the proxy agent systems used by parties A and B. If users elect, or are required by company policy to utilize the proxy agents to send and receive all messages, this can improve corporate message tracking, backup, and accountability, as well as provide a suitable point at which to assess usage based (per message) billing and royalty payments.

This is especially desirable because the liability of the post office and other system operators will be proportional to the total volume and value of traffic handled by the system. However, because the certified e-mail embodiments of the present invention are generally off-line systems, utilizing an invisible post office, the post office typically is not aware of any messages unless called upon to perform recovery services. Hence the PO 240 cannot know how many messages have been sent at any given time, or their value, or even whether they were ever completed, in cases where B either never received the message, declines to sign the receipt, or declines after signing the receipt and not getting M_B from A, and for some reason does not bother to request recovery.

When a proxy agent acts on behalf of either A or B, or when a user is signed up for a given mail or message system, the proxy server can, for example, impose usage charges one or more of the following ways:

-The charges assessed by the proxy can be billed to either the sender or recipient, or the sender or recipient's organizational sponsor or their department within the sponsor;

-The financial account to be billed can be prepaid, accrued and invoiced, or can be billed to an outside electronic payment or billing service, such as a digital wire transfer, ACH debit, e-check, credit card, direct debit, digital coin, subscription, electronic scrip, or an invoice by proper mail service;

-The choice of the account to be billed can be fixed, or may vary depending on the identity of the sender, receiver, sender's sponsor, receiver's sponsor, sender's preferred PO, receiver's preferred PO, message size or type, message priority, level of insurance or other financial assurance requested or required;

-The amounts to be billed to the designated account can be fixed per message, based on the declared value field (assuming it is readable or else separately stated by the sender), based on volume discounts, based on time of day or day of week discounts, or may vary depending on the type of quality of communications network used or to be used;

-The users can be billed on a per use basis, with an optional monthly cap (to limit the size of the bill they can run up), or on a pre-paid subscription basis, where an amount is deducted from their financial account inside or (or associated with) the proxy agent server, for each use;

-The amounts to be billed may reflect a discount for prepayment on a subscription basis, or may differ depending on which form of payment is used, and whether that payment is immediate, same day, several days later, or end of month plus 15 days (e.g., T+45 days).

When a customer is about to accrue a charge that exceeds their subscription account balance, monthly credit limit, or the like, or where the declared value is in excess of that allowed by the predefined rules of the system, the system may notify a "reviewing officer," which could be a human or artificial agent that is empowered to review such a request and approve or deny it.

In one embodiment, several different accounts may be established, either inside the proxy agent server, or associated with it, to which fee payments are to be credited, and eventually settled and remitted. Such accounts may be for the benefit of the operator of the proxy server, a sponsoring post office (if any) which accredits the proxy agent and may assume liability for its reliable and correct operation, a seller or licensor of software (for collecting pay-per-use fees) or a licensor of intellectual property rights (for collecting royalties), or for the user's sponsor or for the user themselves (for accruing refunds, credits, or loyalty points, such as frequent flyer miles, as a reward for using the system).

The proxy agent and its financial accounting system can also notify system users of available user-software upgrades, respond to requests for upgrades, and bill the price of those upgrades (if any) to the user's individual or organizational accounts, in accord with any applicable pricing plan that may be in effect with regard to that sponsoring organization.

The accounting capabilities of the proxy agent servers can be enhanced to include registers, counters, or accounts for recording such data as:

-Total number of messages sent or received

-Total declared value of messages sent

-Count of messages sent and received by declared language and data format

-Total number of receipts received or sent

-Total receipts not signed, by reason codes, including user discretion, bad certificates, bad message types, kill lists, and so on,

-Total receipts received but not accepted, due to signed by wrong person, not same certificate for B as used by A, name form not followed the rules, improper delegation, etc.

-Total receipts sent but no messages M_B received from A

-Total recoveries requested

From time to time the sponsoring post office, or some other system-wide authority, may issue an electronic instruction to the proxy agent requesting it to transmit the data it has collected to a centralized location for further collation, summarization, and processing to monitor and assess system use and performance characteristics and systemic risks. Such data will not be deleted from the memory of the proxy agent, nor counters reset to zero for any given period, until the proxy agent receives a confirmation of successful receipt and decryption from the sponsoring post office or other system-wide authority that issued the request. Alternatively, the proxy agent may be instructed to forward such information spontaneously to the sponsoring post office, or other system-wide authority, on a daily, weekly, monthly, quarterly, etc. basis.

D. More Detailed EEM Protocol

Referring to **FIGs. 5A** and **5B**, a flow diagram 500 representing an EEM protocol according to an embodiment of the present invention is shown. The EEM protocol is handled by the EEM system architecture 200 presented above. More specifically, **FIGs. 5A** and **5B** illustrate the Micali protocol as enhanced by the policy and billing signals discussed above in Sections V.A to V.E (i.e., an embodiment of the EEM protocol of the present invention) with respect to parties A and B at user sites 206a and 206b, respectively, and the PO 240. The normal steps (i.e., non-recovery) of the embodiment of the EEM protocol presented in **FIGs. 5A** and **5B** is presented and summarized (with examples) in **Table 3** below. In this embodiment, a different notation is utilized where M_0 is the cleartext and the M_X's are the larger constructs.

TABLE 3

PARTY	ACTION	DESCRIPTION
A:	V_A = eem_protocol_version	A's EEM protocol version number
	T_A = create_datetime	sender creation date (including GMT offset)
	N_A = sender_unique_msg_id	sender unique message ID (e.g., H(A)+T_1+seqno)
	L = msg_language_code	lets recipient know if he'll be able to read it
	A = sender_name	sender name: "Alice Apple" <alice@an_isp_service.com>

TABLE 3

PARTY	ACTION	DESCRIPTION
	B = recipient_name	recipient name: "Bob Barton" <bob@an_isp.net>
	P = post_office_name	declared PO: "EEM_PO" <eempo@trustco.com>
	$ = declared_value	[value, curr_type, exponent], for billing
	K_A = bill_sender_consent	T/F?: A's consent for PO to bill sender
	S_A = sender_stale_date	date after which PO no longer helps B
	$Q_1 = V_A, N_A, L, T_A, A, B, P, \$, K_A,$ [...]	inner message attribute vector
	M_0 = message content	cleartext to be sent
	$M_B = E_B(M_0)$	"B's message" - the ciphertext B will receive
	$M_1 = E_{PO}(Q_1, M_B)$	post office envelope: inner block and message
	$Q_2 = V_A, P, N_A, L, T_A, S_A, K_A,$ [...]	outer message attribute vector
	H_1 = hash(M_2)	store this now for later use (see M_2, next)
	Write to disk: N_A, B, V, P, T_A, H_1, M_B	what A needs to complete the task
A→B:	$M_2 = S_A(Q_2, M_1)$, [$Cert_A$, $Cert_{PO}$]	A's complete message to B
B:	Use $Cert_A$ to check signature of A over (Q_2, M_1)	
	Verify $Cert_{PO}$ to see if P is a valid post office	
	Inspect $Cert_A$ to see if A is a pre-paid subscriber to P	
	Check L to see if message is in an understandable language	
	Check K_A to see if the sender will pick up any PO charges	
	V_B = eem_protocol_version	B's EEM protocol version number
	H_1 = hash(M_2)	don't send back whole message
	N_B = recip_unique_rcpt_id	recipient's unique receipt ID, e.g., H(B)+T_2+seqno

TABLE 3		
PARTY	ACTION	DESCRIPTION
	T_2 = receipt_datetime	datetime of receipt (including GMT offset)
	$Q_3 = V_B, N_A, N_B, T_2, P, [...]$	receipt attribute vector
	Write to disk: $N_B, N_A, P, L, T_1, T_2, H_1$	what B needs to complete the task
B→A:	$R = S_B(Q_2, H_1), [Cert_B]$	B's receipt to A
A:	Use N to retrieve B, V, P, T_1, H_1, M_B	
	Compare Q2 against retrieved data	
	Check B's signature over receipt	

VI. Advanced EEM Embodiments

A. Nested M_0 and Multi-hop EEM Transmissions

In various situations a sequence of parties will wish to receive intermediate receipts from others "down the line" relating to the progress of a single message.

1. Controlled Sequence / Chains Back to A / Proof-of-Sending

A may wish to receive an intermediate receipt R_1 from an intermediate party V (such as a value added network (VAN) or internal corporate mail server), and also a final receipt R_2 from B, the ultimate recipient. To achieve this result, A can form a nested version of M_0 that first requires a receipt from V, and once V gets access, V can forward an inner M_0 on to B, which will then require a second receipt signed by B prior to A completing the protocol with B.

The sequence listed in **Table 8** illustrates chaining back to A.

PARTY	ACTION
A:	$M_B = E_B(M)$
	$M_0 = E_{PO}(A \mid B \mid M_B)$
	$M_V = E_{PO}(A \mid V \mid M_0)$
A→ V:	M_V
V:	$R_V = S_V(M_V)$
V→ A:	R_V
A→ V:	M_0
V→ B:	M_0
B:	$R_0 = S_B(M_0)$
B→ A:	R_0
A→ B:	M_B

TABLE 8

This embodiment may be useful when A wants proof-of-sending as of a particular time and fears that B may not respond in a timely manner. A may use this multi-hop embodiment to send to the intermediate party V. V is a service provider under contract with A to provide prompt receipts according to a contract-specified service level. Also, according to A's contract with V, party V will put A's message on the wire to B and mark the time of sending a journal. By using an independent party V to send A's message, A can now prove that she in fact sent her message at a given time to one who was obligated to re-forward it immediately.

Alternatively, because A already trusts V to return a proof of send receipt,, it will often be sufficient for A's contract with V to specify that V will send A an ordinary receipt (signed or unsigned) upon receiving M_0 from A.

2. *Compound Nesting / Chains Back to V*

Alternatively, A may wish to receive an intermediate receipt R_1 from an intermediate party V (such as a value added network or internal corporate mail server), but further direct that B send the final receipt R_2 to V. To achieve this result, A can form a nested version of M_0 that first requires a receipt from V, and once V gets access, V can forward an inner M_0 on to B, which will then require a second receipt signed by B prior to V's completing the protocol with B.

The sequence listed in **Table 9** illustrates chaining back to V, under the direction of A:

PARTY	ACTION
A:	$M_B = E_B(M)$
	$M_0 = E_{PO}(A \| V \| M_B)$
	$M_V = E_{PO}(V \| B \| M_0)$
A→ V:	M_V
V:	$R_V = S_V(M_V)$ V substituted for A in M_0
V→ A:	R_V
A→ V:	M_0
V→ B:	M_0
B:	$R_0 = S_B(M_0)$
B→ V:	R_0 -- V performs final delivery
V→B:	M_B

TABLE 9

One concern is whether the system is capable of enforcing A's decision to use only V as the party who is must complete the protocol with B. If V applied to the PO 240 to recover his own message, and recovery were granted, V could re-wrap M_B with someone (anyone) else as the sender.

In the split-key embodiment of the EEM protocol described in Section III.C above. V could recover his own fragment of the symmetric (DES) key, and then, although V could not read M_B without also having B's symmetric key fragment, V could form a new EEM message encrypting his own former fragment in the new header under another party's public encryption key, thereby redirecting the protocol completion steps to that other party, perhaps subverting A's intent that V serve as the protocol completion party.

Therefore, it will be desirable to place a signal flag inside M_V telling the PO 240 that the original sender was A and that A has appointed V to complete the protocol. At that point, under contract, the PO might refuse recovery to the intermediary V, while still allowing it for B. While under normal circumstances this is not a concern, when designing these kinds of systems it is desirable to account for all possibilities.

3. Simple Chaining of EEM Forwards

In another embodiment V can simply reapply the EEM protocol when forwarding to B a message received from A.

B. Support Integrated Transactions

Other desirable enhancements to the EEM protocol may be effected by providing direct integration of other business functions beyond the "fair exchange" provided by the EEM protocol.

1. Cash on Delivery (COD) Extension and Linked Receipt

In one embodiment, A can enhance M_0 to include a request for payment, which B may be required or permitted to remit back to A in the receipt R. Alternatively, A may state that A's completion of the exchange protocol is conditioned on prior receipt by A of monetary payment or credit (or other exchange of value or rights) by a means acceptable to A.

To the super-encrypted message M_0 an optional unencrypted (visible) attribute or extension is added to signal B that payment is required, along with the amount and other remittance instructions, such as the payment technology (e.g., digital cash mechanism), and A's bank number and account number, or the like. This "COD Extension" should appear outside the outer (PO) envelope, but within the scope of A's external signature over the outer envelope, preferably for current secure mail systems as a signature attribute to A's external signature.

Where a split key scheme is used, there is no "outer envelope" as such, because there is only one envelope requiring two keys to open. Hence it might be more accurate to call it the "double condition layer" to signify that the removal of the layer (in both cases) is subject to two conditions. This coinage has the drawback of being relatively abstract. Hence either term may be used with the understanding that they are interchangeable.

If B does not wish to make the requested payment, he is advised to reject M_0 without signing the receipt or return a negative ("DECLINE") receipt with the reason of "declined to accept COD charges." B's software system will request user confirmation and prompt B, showing the nature of the request and seeking B's authorization to initiate the payment instruction,

If B elects to make the requested payment, there are several ways it may be carried out. B can initiate the payment using any payment methodology (even mailing a paper check) and then reference A's M_0 message number and his own R_B receipt serial number on that payment in the memo field. Such a memo reference will allow both A and B to reconcile their payment records with their messages and receipts. This approach is less desirable, as it may be unduly difficult to reconcile disconnected payments with receipts.

In the alternative, the EEM receipt may be modified to include a "payment memo" field, in which B, prior to signing the receipt, may include a payment system identifier and transaction reference number or optionally a hash value of the payment message. Although payment to A still flows through the unrelated channel of the payment system, at least now the receipt provides information to facilitate automated linkage and reconciliation with the statements and/or remittance advices of the payment system.

Further, with several digital payment methodologies, it may be possible to directly integrate B's payment into the receipt R, thereby providing the highest degree of binding between the two related acts of performance by B (B's signed receipt and B's payment). In this case a "payment transmittal" extension may be provided to B's receipt message R, which will identify the payment system type and further contain the actual digital payment, valid for redemption by or credit to A, generated by or on behalf of B under the rules and procedures of that payment system.

Further to the COD payment method, the following digital payment systems may be suitable for use to provide a COD payment functionality within the EEM receipt:

DigiCash™, offered in the US by Mark Twain Bank (discontinued in September of 1998), provides a digital payment token that emulates a coin of fixed value, redeemable by A.

CyberCash™, an independent payment service located in Reston, VA, provides a payment message telling the recipient (usually a merchant who also subscribes to the CyberCash system) that CyberCash will credit his account with the funds.

GC Tech™, an unreleased payment system designed by a French firm, provides B with a payment advice he can send to A, which is signed by a bank (or an agent of a bank), advising A that she will receive payment in good funds into her account (as designated her the "COD Extension" field within M_0) at a time certain, usually next business day.

Electronic Monetary System (EMS™), proposed by Citibank, NA of New York, provides a form of digital message which emulates a piece of currency, and which can be sent by A's digital wallet and received and redeemed by A.

Secure Electronic Transactions (SET™) sponsored by VISA and MasterCard, provides a method to transmit a message authorizing A to debit B's credit card account, signed by B, and transmitted to A's merchant accepting bank for payment to A's account.

Mondex™, a global electronic cash system sponsored by Mondex International, provides a message signed by B's payment smart card that will be accepted by A's smart card a proof that the desired funds have been debited from B's account (the balance of which is maintained on B's card), and may be redeemed by A upon transmission of a redemption request message from A's smart card to A's participating bank.

Financial EDI™, a general methodology for creating and processing financial payment and advice messages, provides a payment advice message, called an "820" transaction, which when signed by B, may assure A that B has or will make the payment as described in the 820, and if signed by B's bank, would provide further assurance that said payment will occur in due course.

These and other (present and/or future) payment systems share the desirable feature that B's primary action of delivering value to A occurs in a single payment message, which is compatible with direct integration into the signed EEM receipt. While the COD processing is not limited to "single step" payment protocols, these are more convenient to integrate, manage, and reconcile. If a more complex payment protocol is used, the goal of "integrating" the receipt and payment could be partially achieved by inserting at least one step (message) of that protocol into the receipt R.

The COD payment extension field may contain information that must be encrypted prior to transmission from B to A, either to protect the confidentiality of the parties (e.g., as to transaction description), or to preserve the security and integrity of the payment system. In such a case B may obtain the public encryption key of A and use it to protect any such information, prior to forming the receipt R and transmitting it to A.

The EEM protocol need not, and probably should not, make any alterations to the primary payment message (or value message) as specified and generated by the payment system used. Preferably, B's software will copy those value messages verbatim into the payload section of the "COD Payment" extension in the receipt R, possibly with the addition by B of an overlying layer of encryption readily removable by A. After A removes such encryption, if any, and extracts the value message from the COD payment extension, A will further process said value message (as or if required) and submit it to his bank or other payment service provider for conversion into good funds.

Some payment systems may require or permit other messages between B and A in addition to those described above.

As a further enhancement, B's EEM user software may provide a function allowing B's payment software to request information regarding the transaction for which payment is being requested, including at least the transaction description and A's message reference number(s) present within M_0. In addition the EEM user software can pre-generate the receipt ID numbers that it plans to use in the receipt that will be returned to A and that will contain the payment or value message. Based on this information, the payment software can place such transaction data and message reference numbers into the memo field of the payment or value message, form the rest of the value message, apply the appropriate digital signature, and then convey the entire value message (now containing the transaction data) to the EEM user software to be integrated with the receipt R.

2. *Multi-step Transaction Support*

In another series of embodiments, multi-step (complex) transaction support may be provided by combining (1) receipt of transaction 1 with initial send of transaction 2 and/or (2) receipt of transaction 2 with fulfillment of transaction 1.

Consider the following example in **Table 10**, which illustrates both of these customizations. Because there are two of them, references to M_0 are replaced by M_X and M_Y.

PARTY	ACTION	DESCRIPTION
A:	$M_B = E_B(M)$	A's message intended for B
	$M_X = E_{PO}(A \mid B \mid M_B)$	A prepares her initial send, M_X
A→ B:	M_X	
B:	$M_A = E_A(M)$	B's message intended for A
	$M_Y = E_{PO}(A \mid B \mid M_B)$	B prepares his initial send, M_Y
	$R_B = S_B(H(M_X) \mid M_Y)$	Embed M_Y in R_B
B→A:	R_B	Send as combined message
	Verify R_B, extract M_Y	
A:	$R_A = S_A(H(M_Y) \mid M_B)$	A combines M_B with R_A
A→B:	R_A	
B:	Verify R_A, extract M_B	A's send to B is now complete
B→A:	M_A	B's send to A is now complete

TABLE 10

As suggested in **Table 10**, for economy in forming receipts containing embedded materials from other steps of other transactions, the following formatting construct would be preferable:

$$R_B = \{ S_B(H(M_X) \mid H(M_Y)) \mid M_Y \}$$

This has the benefit that $H(M_Y)$ is embedded within the region signed by B, but the entire message M_Y, which could be quite large, is not itself embedded, but rather is appended, or perhaps transmitted via other means. This can keep the receipt RB small enough to be readily stored in a commercial database system. In this case, it is preferable to embed an appropriate "external attachment" extension in the receipt, that provides transaction or message reference numbers, to help the recipient (i.e., A) associate $H(M_Y)$ with M_Y.

C. Support Proof-of-Decryption (POD)

It is desirable to add a "proof of decryption" feature to the EEM protocol and system 200. This enhancement can achieve two related objectives: (a) to provide proof to A that B has decrypted the message M_B; and (b) to link this proof to the original M_0 message header, so that such proof can also be furnished to the PO, or (c) in another embodiment, to provide this "has-read" proof to any pre-determined party.

1. Integrate POD into Basic Protocol

The following example (slightly simplified) in **Table 11** shows how this feature can be integrated into the basic Micali protocol 100.

PARTY	ACTION
A:	Z = random_nonce
	$M_B = E_B(Z \mid M)$
	$M_0 = E_{PO}(A \mid B \mid H(Z) \mid M_B)$
A→ B:	M_0
B:	$R = H(M_0)$
B→A:	R
A→B:	M_B
B:	$Z \mid M = D_B(M_B)$
B→A:	Z | Info
A:	check Z against H(Z) in M_0
	Or, if A fails to send M_B and B must apply to PO for help:
B→PO:	R, M_0

PARTY	ACTION
PO:	Verify R and M_0
PO→B:	M_B
B:	$[\, Z \mid M \,] = D_B(\, M_B \,)$
B→PO:	$Z \mid \text{Info}$
A:	check Z against H(Z) in M_0
PO→A:	$R \mid Z \mid \text{Info}$
A:	check Z against H(Z) in M_0

TABLE 11

As will be apparent to one skilled in the relevant art(s), this POD feature can be incorporated into any of the EEM protocol embodiments presented herein. B cannot be required to decrypt M_B, to send back Z, or to examine or use M, but this mechanism may be provided to allow B to assure A that decryption has taken place, and make it easy for A and the PO to verify this fact.

2. *A Directs B's POD to a Third Party*

In another embodiment, A can form M_0 in a manner that directs B to send the POD to another party. A does this by naming the third party (TP) and including a copy of (or a pointer to) TP's public encryption key, and providing an encapsulation of Z in a form that is verifiable only by TP.

The following example (slightly simplified) in **Table 12** shows how this feature can be integrated into the Micali protocol 100.

PARTY	ACTION
A:	Z = random_nonce
	$TP_1 = TP \mid K^{+}_{TP} \mid E_{TP}(A \mid B \mid H(Z) \mid \text{Info}\,)$
	$M_B = E_B(\, Z \mid TP_1 \mid M\,)$
	$M_0 = E_{PO}(\, A \mid B \mid H(Z) \mid M_B\,)$
A→B:	M_0

PARTY	ACTION
B:	$R = H(M_0)$
B→A:	R
A→B:	M_B
B:	$[Z \mid TP_1 \mid M] = D_B(M_B)$
	$TP_2 = E_{TP}(Z \mid Info)$
B→TP:	$TP_1 \mid TP_2$
TP:	decrypt both TP_1 and TP_2
	check Z in TP_2 against H(Z) in TP_1
	check that B from TP_1 is correct
	$TP_3 = E_A(A \mid B \mid Z \mid H(Z) \mid Info)$
TP→A:	TP_3
A:	verifies TP_3 against original M_0 send

TABLE 12

In **Table 12**, the PO recovery process step has been omitted, because B's actions under this embodiment are the same either way, regardless of how he receives M_B.

Note that the "Info" field is included as a generic concept, as used elsewhere within this document. It will typically contain such data elements as sender and receiver transaction and message ID numbers, date-time fields showing time of transmission or reception, hash values of various protocol elements for matching, and other system administrative control fields, such as billing information, as described in Section V above.

D. Support Registered Mail Concept

Normally, most electronic or paper messages can simply be resent if they fail to arrive. However, some messages may be more sensitive. In the electronic world, irreplaceable value bearing messages may exist, such as delivery of anonymous digital cash, airline tickets, passwords, etc. that justify additional tracking.

The proof of delivery (POD) embodiment described above can be combined with a "post on send" (POS) feature to produce a service level similar to "registered mail" as historically offered by the United States Postal Service. Unlike certified mail, which only undertakes to make delivery conditional on the signing of a receipt, registered mail also undertakes to track the entire process, and is used especially when the message may contain irreplaceable items of high intrinsic value, such as currency, negotiable securities, precious metals, etc.

The POS and POD tracking messages may be sent and received by a third party monitor service, which could be the PO 240, or A's proxy, even though the original message is never sent to the third party monitor service.

The following example in **Table 13** shows the Micali protocol 100 as enhanced by both the POS and POD methods of the present invention. To facilitate comprehension, the new POS/POD material is shown in **boldface**.

PARTY	ACTION	DESCRIPTION
A:	$Z = \text{random\_nonce}$	
	$TP_0 = E_{TP}(TP \mid A \mid B \mid H(Z) \mid Info)$	new POS message to TP
	$TP_1 = TP \mid K^{+}_{TP}$	omits recovery data, which is now in TP_0
	$M_B = E_B(Z \mid TP_1 \mid M)$	
	$M_0 = E_{PO}(A \mid B \mid H(Z) \mid M_B)$	
A→ B:	M_0	
A→TP:	**TP_0**	A posts data to TP on send, TP listens for B's response
B:	$R = S_B(M_0)$	
B→A:	R	
A→B:	M_B	
B:	$Z \mid TP1 \mid M = D_B(M_B)$	
	$TP_2 = E_{TP}(Z \mid Info)$	

PARTY	ACTION	DESCRIPTION
B→TP:	TP_2	B sends basic POD data to TP
TP:	decrypt TP_0 and TP_2	
	check Z in TP_2 against H(Z) in TP_0	
	check that B from TP_0 is correct	
	$TP_3 = E_A(A \mid B \mid Z \mid H(Z) \mid Info)$	
TP→A:	TP_3	
A:	verifies TP_3 data against original M_0 send	

TABLE 13

The "Info" field will contain the time of each prior action, which will be of interest to A, especially the declared time of A's send, the time TP_0 was received by TP, the declared time of B's processing of M_0, and the time TP_2 was received by TP. Because A's sending of TP_0 to TP occurs after A's sending of M_0 to B, TP_0 may be enhanced to attach or include any proof-of-sending receipt that A may have received from a "value added network provider" (VAN) or mailroom. Alternatively, A may send both M_0 and TP_0 to his mailroom simultaneously.

VII. Centerless Time Stamping

In a secure communication system with multiple parties, it is often highly desirable to reliably establish the date and time a given message was sent or received. For example it may be desirable for legal reasons to determine whether some given message or data: (1) existed in a fixed form at a given time; (2) was in fact sent to and/or received by another party at a given time; (3) was sent or received before or after another, i.e., relative time sequence; and/or (4) existed or was sent or received at or near some absolute date and time.

It is undesirable to allow a party which originates or receives a message to act as its own witness, and self-certify the time of sending or receipt, because the party may have reason and/or opportunity to alter the time, e.g., to show that it was in conformance with the time requirements of a contract or legal requirement, when actually it was not.

Hence, it has been commonly assumed that all transactions should be sent to an external trusted third party (TTP), e.g., a time stamp service, unrelated to the subject party, which will reliably affix the correct time, digitally sign the result (or secure it by other cryptographic means) and return the signed message. Such a TTP may be called a "heavy weight," because it interposes its costs and delays on every message. A time stamp service is described in detail in S. Haber and W.S. Stornetta, *How to Time-Stamp a Digital Document*, Advances in Cryptology, CRYPTO '90 Proceedings, Springer-Verlag, 1991, pp. 437-455.

In an electronic data interchange (EDI) system, as commonly utilized between long term trading partners in a commercial trade environment, the role of TTP is commonly played by a VAN which processes and logs each message and can provide independent evidence of the existence and timing of any message or transaction upon a request from a party to the transaction, an external fact finder or tribunal, or anyone else with an interest in the subject transaction.

Following Micali, this party is called a "heavy weight" because it must stand between the two actual parties in each and every case, thereby imposing delays due to communications overhead, the inevitable high costs of running the VAN as a separate business or legal organization, potential bottlenecks due to single points of failure or insufficient capacity to handle peak traffic, etc. Also, such heavy weight parties may typically exert an undue political influence over the structure of the industry, and gain access to large amounts of sensitive commercial data content and traffic analysis information.

In a banking environment, a trusted third party (TTP) might be housed in a different part of the same organization, provided that the service was fully independent and separated by legal and procedural "firewalls" to prevent improper influence on the service's integrity.

When there are many parties in a given system, and each of them sends or receives a very small number of messages per unit of time, the interposition of such an independent TTP in each transaction may be preferable.

However, when the parties are large organizations which send numerous messages for unit time (per day, per hour, etc.) and each completed transaction may typically comprise a "set" of several related messages (the parts of which may often overlap with the parts of other message sets, then it will be preferable to stop sending every message to the heavy weight TTP for purposes of time stamping, because in accordance with the present invention the parties can achieve commercially adequate proof of date-time without the physical, economic, or political overhead it may impose.

The present invention contemplates the following new methodology, which uses an "invisible" arbitrator who is only called by the parties in case of a dispute, similar to the post office in the Micali protocol 100. For purposes herein, it will be referred to as a Time Arbitration Office (TAO). In actual practice the TAO may often also be an EEM PO 240, but the technical processes are different, and are invoked at different times, and hence must be considered separately.

Further, although this method represents a substantial advance over the use of VANs and other heavy weight intermediaries, it does not entirely eliminate the need for an external, independent "time reference service" (TRS), which must be resorted to when transaction volumes drop too low for the new method to work effectively. Further, it is desirable for each party to "poll" the TRS on a regular basis, and insert an external absolute time reference record from such a TRS occasionally (e.g., hourly) within its relative transaction sequences, to strengthen their reference value.

A. Time Arbitration via Interleaved Time Chains

If the following conditions are present: (1) many parties in an electronic commerce system are sending multi-message transactions to many parties simultaneously; (2) every transaction and all its messages are uniquely numbered; and (3) each party's software takes adequate steps to provide secure evidence of time sequencing, then the message journals of all system participants, taken as a whole, can provide the necessary independent witnessing capability, without the need for active intermediation of a TTP on every message, provided there is a trusted party (the TAO) which, in the event of dispute or data loss, can request transaction and message sequencing information from those parties possessing "relevant evidence," analyze those transaction and message sequences, and render an impartial finding as to the actual sequencing and absolute time of any disputed messages.

The method can also work when at least one party to a given transaction or message is doing a high volume of traffic with other parties, even if the other party is not, because in this case, the TAO can rely principally on the evidence it gets from the high volume party (HVP), and especially the HVP's trading partners, despite the paucity of usable data from the low volume party (LVP) or its other trading partners.

B. Generating and Recording Interleaved Time Chains

Although the principles of the present invention would be the same regardless of the type and structure of the transactions and their component messages, for discussion purposes, a commercial system in which parties are sending each other EEM transactions as described herein is considered.

Consider an EEM transaction flow model as follows:

A→ B: M_0	-- initial EM message send
B→ A: R	-- receipt
A→ B: M_B	-- inner message provided to B
B→ A: POD	-- B provides proof of decryption

A comprehensive transaction numbering system is required to implement this. However, such numbering systems are commonplace, and in any event the burden of any added complexity is amply offset by the cost savings from not having to make payments to a third party processor.

Each party will have: (1) A system-wide unique party ID number; (2) their own transaction sequence number for the entire transaction; and (3) their own message sequence number for each message within a given transaction. Suppose the system-wide unique party ID numbers are:

A's Party ID = 15198

B's Party ID = 20901

and that A and B each maintains their own unique sequence counter for all transactions in accordance with which they designate this transaction (TXN) as:

A's TXN ID = "EEM2-1998-180995"

B's TXN ID = "EEM2-1998-097242"

Parties A and B assign each message of this EEM transaction their own sequence number (1, 2, 3, etc.) for message numbers within this transaction.

These three numbering systems are practiced uniformly by both parties, and when viewed together yield (in practice) the following presented in **Table 14** (or equivalent):

Step	Direction	Content	A's Message Number	B's Message Number
1	A→ B	M_0	15198-EEM2-1998-180995-1	20901- EEM2-1998-097242-1
2	B→ A	R	15198-EEM2-1998-180995-2	20901- EEM2-1998-097242-2
3	A→ B	M_B	15198-EEM2-1998-180995-3	20901- EEM2-1998-097242-3
4	B→ A	POD	15198-EEM2-1998-180995-4	20901- EEM2-1998-097242-4

TABLE 14

The foregoing numbering system of **Table 14**, while indicative of actual practice. is unwieldy for explanation purposes. and thus will be simplified as shown in **Table 15**. This is with the understanding that the former system of **Table 14** is closer to how the present invention actually operates.

Step	Direction	Content	A's Number	B's Number
1	A→ B	M_0	A-180-1	B-097-1
2	B→ A	R	A-180-2	B-097-2
3	A→ B	M_B	A-180-3	B-097-3
4	B→ A	POD	A-180-4	B-097-4

TABLE 15

When A and B send each other the above messages, preferably each one will include both its own reference number for that message step, and the other party's message number from the prior step.

Upon receiving the first message of a new transaction, each party will utilize its own "next" sequence number for the entire transaction, and record the other party's transaction sequence (TSN) number for reference.

Each party will record all messages sent and received in a journal, such as a computer database, indexed at least by the unique TSN they have assigned, and each message entry will also contain the other party's TSN.

It is generally assumed that each party has a well managed and secure commercial data processing system with adequate fault-tolerant backup and recovery systems. That means each party will in all cases possess the physical data records contemplated by the present invention.

Further, to perform the present invention, each party will insert a cryptographic checksum (or "hash") value derived from the prior message (in the same transaction sequence) into each subsequent message in that sequence, and each party will (in accordance with an embodiment of the EEM protocol) digitally sign each message they send, and store the digital signature of the counter-party from each message they receive.

It may be desirable to include one or more additional hash values which represent the cumulative hash value (the secure hash-chain concept of Haber and Stornetta) from the sequences of messages originated by only that party (only sent), or preferably both parties to that specific transaction (both sent and received), and/or, more importantly, all parties with which party X was exchanging other messages while waiting for all messages of the subject transaction to be completed.

However, while the Haber-Stornetta "chaining" method may add assurance, it is not required, for an equivalent result can be achieved by simply including in each message sent (and digitally signed by its sender) the prior (non-chained) hash value from each message meeting one or more of the above criteria.

Next, it is assumed that during the time that A and B are exchanging these messages, they are also exchanging other sets of similar messages with other parties, A with C and B with D, such that the messages from each transaction set are temporally interleaved with each other in the databases of the parties.

C. *A Detailed Example*

The above may yield a sequence of messages between the four parties such as shown in **Table 16**.

TIME				PARTY			
	C		A		B		D
1				----AB1----→			
2						← ---- DB1----	
3				← ---- AB2----			
4		← ---- AC1----					
5				---- AB3----→			
6						---- DB2----→	
7		---- AC2----→					
8						← ---- DB3----	
9				← ---- AB4----			
10		← ---- AC3----					
11						---- DB4----→	
12		---- AC4----→					

TABLE 16

It will be apparent to those skilled in the relevant art(s) that if either of the two parties have a disagreement about the relative or absolute time sequencing of any message, they can appeal to the TAO, which can generally resolve the matter by polling the other counter-parties of the two affected parties. And the greater the saturation level of the system, in terms of messages per unit time among a broad diversity of parties, the more complete and accurate the TAO's determination will be.

For purposes of this example, each party will maintain a journal of every transaction (both sent and received), in the following database format:

[rec_no]	not shown, system assigned unique record ID within database
time	time of each transaction (shd be yyyymmddhhmmss.mmm), represented as a series of integers for simplicity
descr	message description (optional)
direction	identifies sender, receiver and direction
my_num	concatenates party ID, my transaction sequence number, and the message sequence number within the transaction
in_num	same data as my_num, received or inferred from other party
this_hash	a hash or message digest of the current transaction
my_last_txn_hash	the hash of the current transaction's last message
my_last_abs_hash	the hash my exactly prior message in absolute sequence
my_time_ref_blk	concatenation of all prior fields listed above, possibly excluding the description field
in_time_ref_blk	the "time_ref_blk" received from the counter-party, which he took from his immediate (absolute) preceding journal entry

As a result of the foregoing transactions and messages, the database for each party (i.e., A, B, C, and D) now contains the entries shown in **Table 17**, **Table 18**, **Table 19**, and **Table 20**, respectively.

1. ***Party A's Database***

2.

Time	Descr	Direction	My Num	In Num	This Hash	My Last Txn H	My Last Abs H	My Time Ref Blk	In Time Ref Blk
1	AB1	A→B	A-180-1	*B-297-1	H1	N/A	N/A	= A1	
3	AB2	B→A	A-180-2	B-297-2	H3	H1	H1	= A2	**B2 (D)**
4	AC1	A→C	A-181-1	*C-327-1	H4	N/A	H3	= A3	
5	AB3	A→B	A-180-3	B-297-3	H5	H3	H4	= A4	
7	AC2	C→A	A-181-2	C-327-2	H7	H4	H5	= A5	**C1 (A)**
9	AB4	B→A	A-180-4	B-297-4	H9	H5	H7	= A6	**B6 (D)**
10	AC3	A→C	A-181-3	C-327-3	H10	H7	H8	= A7	
12	AC4	C→A	A-181-4	C-327-4	H12	H10	H10	= A8	**C3 (A)**

TABLE 17

The sender A that initiates a the first message of transaction set cannot know the intended recipient B's "next" transaction set number in advance, but A can derive it from B's return message and write it back to A's database record of A's initial message, if desired.

The hash values assigned (H1, H2, etc.) are intended as arbitrary unique values, such as cryptographic message digests. However, for clarity they have been numbered to align with the time interval in which they occur, to make them unique yet understandable within this example.

3. *Party B's Database*

Time	Descr	Direction	My Num	In Num	This Hash	My Last Txn H	My Last Abs H	My Time Ref Blk	In Time Ref Blk
1	AB1	A→B	B-297-1	A-180-1	H1	N/A	N/A	= B1	N/A
2	DB1	D→B	B-298-1	D-412-1	H2	N/A	H1	= B2	N/A
3	AB2	B→A	B-297-2	A-180-2	H3	H1	H2	= B3	
5	AB3	A→B	B-297-3	A-180-3	H5	H3	H3	= B4	**A3 (C)**
6	DB2	B→D	B-298-2	D-412-2	H6	H2	H5	= B5	
8	DB3	D→B	B-298-3	D-412-3	H8	H6	H6	= B6	**D2 (B)**
9	AB4	B→A	B-297-4	A-180-4	H9	H5	H8	= B7	
11	DB4	B→D	B-298-4	D-412-4	H11	H8	H9	= B8	

TABLE 18

When B is the recipient, as in cases AB1 and DB1 above, there is normally an in_time_ref_blk. However in this example, because the system has just "started," the counter-parties A and D have no prior transaction data to send, so the field contains "N/A." This condition would be unusual.

4. Party C's Database

Time	Descr	Direction	My Num	In Num	This Hash	My Last Txn H	My Last Abs H	My Time Ref Blk	In Time Ref Blk
4	AC1	A→C	C-327-1	A-181-1	H4	N/A	N/A	= C1	**A2 (B)**
7	AC2	C→A	C-327-2	A-181-2	H7	H4	H4	= C2	
10	AC3	A→C	C-327-3	A-181-3	H10	H7	H7	= C3	**A6 (B)**
12	AC4	C→A	C-327-4	A-181-4	H12	H10	H10	= C4	

TABLE 19

5. Party D's Database

Time	Descr	Direction	My Num	In Num	This Hash	My Last Txn H	My Last Abs H	My Time Ref Blk	In Time Ref Blk
2	DB1	D→B	D-412-1	*B-298-1	H2	N/A	N/A	= D1	
6	DB2	B→D	D-412-2	B-298-2	H6	H2	H2	= D2	**B4 (A)**
8	DB3	D→B	D-412-3	B-298-3	H8	H6	H6	= D3	
11	DB4	B→D	D-412-4	B-298-4	H11	H8	H8	= D4	**B7 (A)**

TABLE 20

D. Notes and Discussion

The variable my_time_ref_blk (=D1) is shorthand for the concatenation of the eight fields to the left. The variable in_time_ref_blk is the my_time_ref_blk received ("in") from the counter-party, which was taken from the counter-party's immediate (absolute) preceding journal entry, which contains the 8 fields from whatever message the counter-party was sending or receiving, to or from whomever, just prior to the subject message. The letter in parenthesis following the in_time_ref_blk, e.g., "(A)" provides an informative reference to the ID of the counter-party's "prior counter-party" which is referenced in the in_time_ref_blk.

Generally, the parties will be concerned about the confidentiality of the data contained in the reference time blocks which they send to one trading partner, which contain identity and traffic analysis data about their dealings with other unrelated parties. As a further enhancement, the TAO will provide a public key K^{-}_{TAO} to all system participants, which they may use to encrypt the reference time blocks they send. This will keep the recipients from reading the confidential traffic analysis data in the reference time blocks, while allowing the TAO to read them, when it is called upon to arbitrate.

In this embodiment, each time any party receives (but not sends) a message, they will receive a reference time block, in_ref_time_blk. These time blocks, which contain data about other contemporaneous transactions with unrelated parties, can be requested by the TAO in the event of any dispute among the parties, and upon being decrypted using the TAO's private decryption key, can allow the TAO to establish with certainty the relative time sequence of any given transaction, with reference to the interleaved transactions of the disputing parties with all their other unrelated counter-parties.

When a party is "starting" their system, and does not have a "prior" transaction in their journal upon which to base a reference time block, the TAO system rules will provide a standard format to be followed, so that the record will not be null. Preferably, the starting party will be instructed by the TAO to initiate a "first" transaction with an external time reference service (TRS) to acquire a starting record that contains an absolute time reference, and that record's data can then be used to form an appropriate reference time block for their first business message send.

E. Priming the System with Time-Beat Messages

At times when the system may have inadequate saturation of messages over some given time interval, or on a regular basis (such as every 15 minutes), the participants and the TAO may generate and send to one another heartbeat messages to help keep the system full of unique messages.

The parties may send such messages to their nearest counterparts, or the TAO may send them to the parties. For example, if the TAO had concerns about saturation of any party, or at any time, the TAO might send a message to each subscribing party. Further, such a message might contain an instruction to the recipient party telling it to send another heartbeat message to another party selected by the TAO, which may not be a party that the recipient normally trades with, to help improve that party's saturation level.

Further, the TAO's instruction might direct the recipient to initiate a cascade of messages. The TAO might direct A to send further heartbeat messages to a definite or descriptive list of other system participants, such as "B, C, D and E" or "all known participants whose names begin with the letter B." The TAO's instruction to A could contain a further embedded instruction, to be passed on by A, wherein each recipient of the cascade from A might be further directed to send a cascade to yet other participants, such as "two other participants chosen by algorithm from a random number to be generated by the recipient."

Such a heartbeat cascade process must be carefully controlled to prevent it from flooding the system with excessive messages and degrading performance. However, it is simple enough to set the maximum depth to a small number, such as three or four, and likewise control the numbers of sub-recipients, with the object of providing a "faint breeze" of heart beat messages, preferably of relatively random distribution, throughout the system, not exceeding a few such messages per recipient per 10 minute time interval, which should not create a noticeable burden on system resources.

F. Polling an External Time Reference Service (TRS)

In addition to generating heart beat messages to improve system saturation, it will also be useful if all parties from time to time poll an external time service (which may be the TAO or any other trusted third party time service) to request and receive a record of the current absolute time, which will be preferably entered and chained into that party's journal, to provide absolute time markers (signed by the TRS) inserted at regular intervals into the stream of relative interleaved markers generated by normal system traffic flow.

The TAO will provide all participants with a list of network names and addresses of approved TRS providers, and will require such participants by contract to poll one or more of such TRS's at pre-determined time intervals. Each participant may be assigned one or more primary TRS's, and be further directed to poll others on the list if their primary TRS is down or providing suspect data.

It may be further desirable to require all participants to poll different available TRS's at random, to reduce the possibility of damage to system integrity if one TRS begins to malfunction and provide incorrect absolute time data. In yet another embodiment, the TAO and its system participants may specify the use of a Byzantine Agreement protocol, whereby participants may poll several TRS's simultaneously, and in the event of any discrepancies, whether from malfunction or sabotage, employ a suitable algorithm to determine which time value to enter into their respective journals.

G. Monitoring the Adequacy of a Centerless Time System

Various parties, either the participants themselves, or the TAO, can request or receive information to enable them to determine whether any given party's transaction flow has adequate saturation and temporal distribution to provide adequate assurance of proper date-time determination, at a future time, to the TAO and its other subscribers.

Parties may be required to periodically provide information summarizing the total numbers of transactions, sub-messages, counter-parties dealt with, and distribution over different time intervals. The TAO can use this data to assess the party's adequacy of saturation, and the adequacy of their nearest partners and of the entire community, over various past, present, and future time frames. The TAO can publish or provide this information to the parties or to other users, on request, or by subscription, or via publication.

H. Reconstructing a Party's Lost Database (DRO)

It is a further objective of the present invention to provide a neutral third party (a Database Reconstruction Office or "DRO") which can, upon request, assist a subscribing system participant to reconstruct its database in the event of less. Since every message that a party has ever sent or received can be found in the journals of its counter-parties, then upon application by the Loss Party which has lost its database, the DRO can poll all parties in the system, recover from them their records relevant to the Loss Party, then the DRO can carefully scrutinize the records thus received to eliminate any impertinent ones that may have been sent by mistake, and to re-poll for any that may not have been supplied in response to the first request, and can then reconstruct, or help the Loss Party reconstruct, its lost database.

Such an innovation, if implemented and subscribed to across an entire trading population, might reduce or eliminate the high cost of maintaining separate backup and recovery facilities and procedures, because all parties would in effect serve as the backup centers for each other's business operations.

VIII. Multi-Party Self-Maintaining VAN-less EDI System

The essence of the present invention will be to combine the foregoing with a mechanism to continually refresh all party's databases with pertinent counter-party information, and to provide a method to perform complex multi-party trades (as are common in the world of global securities custody and settlement). Thus, all messages may be sent over the open Internet with the benefit of EEM (proof of delivery), centerless time service, and self-updating "cc" lists, such that all the traditional functions of a VAN can be dispensed with, and replaced with a message transfer agent (MTA)-- a EEM PO 240 combined with a TAO. Such lightweight parties remain invisible, cost far less to rely upon than heavyweight parties, and exert very little political or economic control over the business of the parties.

IX. Environment

The present invention may be implemented using hardware, software or a combination thereof and may be implemented in a computer system or other processing system. In fact, in one embodiment, the invention is directed toward one or more computer systems capable of carrying out the functionality described herein. An example of a computer system 600 is shown in **FIG. 6**. The computer system 600 includes one or more processors, such as processor 604. The processor 604 is connected to a communication bus 606. Various software embodiments are described in terms of this exemplary computer system. After reading this description, it will become apparent to a person skilled in the relevant art how to implement the invention using other computer systems and/or computer architectures.

Computer system 600 also includes a main memory 608, preferably random access memory (RAM), and may also include a secondary memory 610. The secondary memory 610 may include, for example, a hard disk drive 612 and/or a removable storage drive 614, representing a floppy disk drive, a magnetic tape drive, an optical disk drive, etc. The removable storage drive 614 reads from and/or writes to a removable storage unit 618 in a well known manner. Removable storage unit 618, represents a floppy disk, magnetic tape, optical disk, etc. which is read by and written to by removable storage drive 614. As will be appreciated, the removable storage unit 618 includes a computer usable storage medium having stored therein computer software and/or data.

In alternative embodiments, secondary memory 610 may include other similar means for allowing computer programs or other instructions to be loaded into computer system 600. Such means may include, for example, a removable storage unit 622 and an interface 620. Examples of such may include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units 622 and interfaces 620 which allow software and data to be transferred from the removable storage unit 622 to computer system 600.

Computer system 600 may also include a communications interface 624. Communications interface 624 allows software and data to be transferred between computer system 600 and external devices. Examples of communications interface 624 may include a modem, a network interface (such as an Ethernet card), a communications port, a PCMCIA slot and card, etc. Software and data transferred via communications interface 624 are in the form of signals 628 which may be electronic, electromagnetic, optical or other signals capable of being received by communications interface 624. These signals 628 are provided to communications interface 624 via a communications path (i.e., channel) 626. This channel 626 carries signals 628 and may be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link and other communications channels.

In this document, the terms "computer program medium" and "computer usable medium" are used to generally refer to media such as removable storage drive 614, a hard disk installed in hard disk drive 612, and signals 628. These computer program products are means for providing software to computer system 600. The invention is directed to such computer program products.

Computer programs (also called computer control logic) are stored in main memory 608 and/or secondary memory 610. Computer programs may also be received via communications interface 624. Such computer programs, when executed, enable the computer system 600 to perform the features of the present invention as discussed herein. In particular, the computer programs, when executed, enable the processor 604 to perform the features of the present invention. Accordingly, such computer programs represent controllers of the computer system 600.

In an embodiment where the invention is implemented using software, the software may be stored in a computer program product and loaded into computer system 600 using removable storage drive 614, hard drive 612 or communications interface 624. The control logic (software), when executed by the processor 604, causes the processor 604 to perform the functions of the invention as described herein.

In another embodiment, the invention is implemented primarily in hardware using, for example, hardware components such as application specific integrated circuits (ASICs). Implementation of the hardware state machine so as to perform the functions described herein will be apparent to persons skilled in the relevant art(s).

In yet another embodiment, the invention is implemented using a combination of both hardware and software.

X. Conclusion

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example, and not limitation. It will be apparent to persons skilled in the relevant art that various changes in form and detail can be made therein without departing from the spirit and scope of the invention. Thus the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What Is Claimed Is:

1. A method for providing enhanced electronic mail (EEM) recovery services by a post office trusted third-party, comprising the steps of:

(1) receiving an encrypted message from a first user, wherein said encrypted message was originally received by said first user from a second user;

(2) using said encrypted message to verify the identities of said first user and said second user;

(3) sending said first user an inner message extracted from said encrypted message; and

(4) sending said second user a receipt created from said encrypted message signed by said first user.

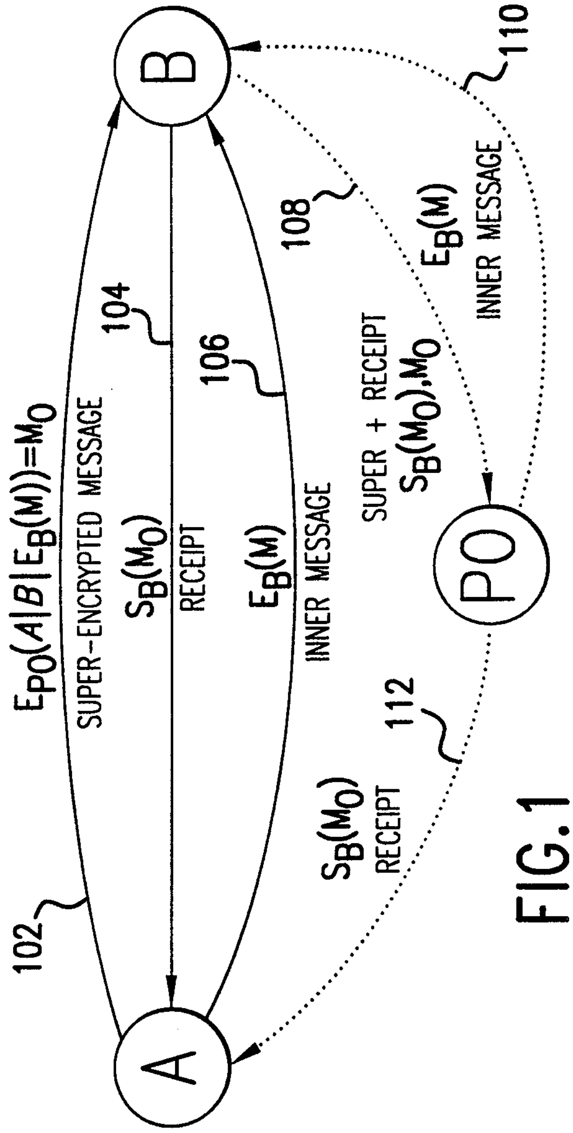

100
B
A
PO
102
$E_{PO}(A|B|E_B(M))=M_O$
SUPER-ENCRYPTED MESSAGE
104
$S_B(M_O)$
RECEIPT
106
$E_B(M)$
INNER MESSAGE
108
SUPER + RECEIPT
$S_B(M_O),M_O$
110
$E_B(M)$
INNER MESSAGE
112
$S_B(M_O)$
RECEIPT

FIG.1

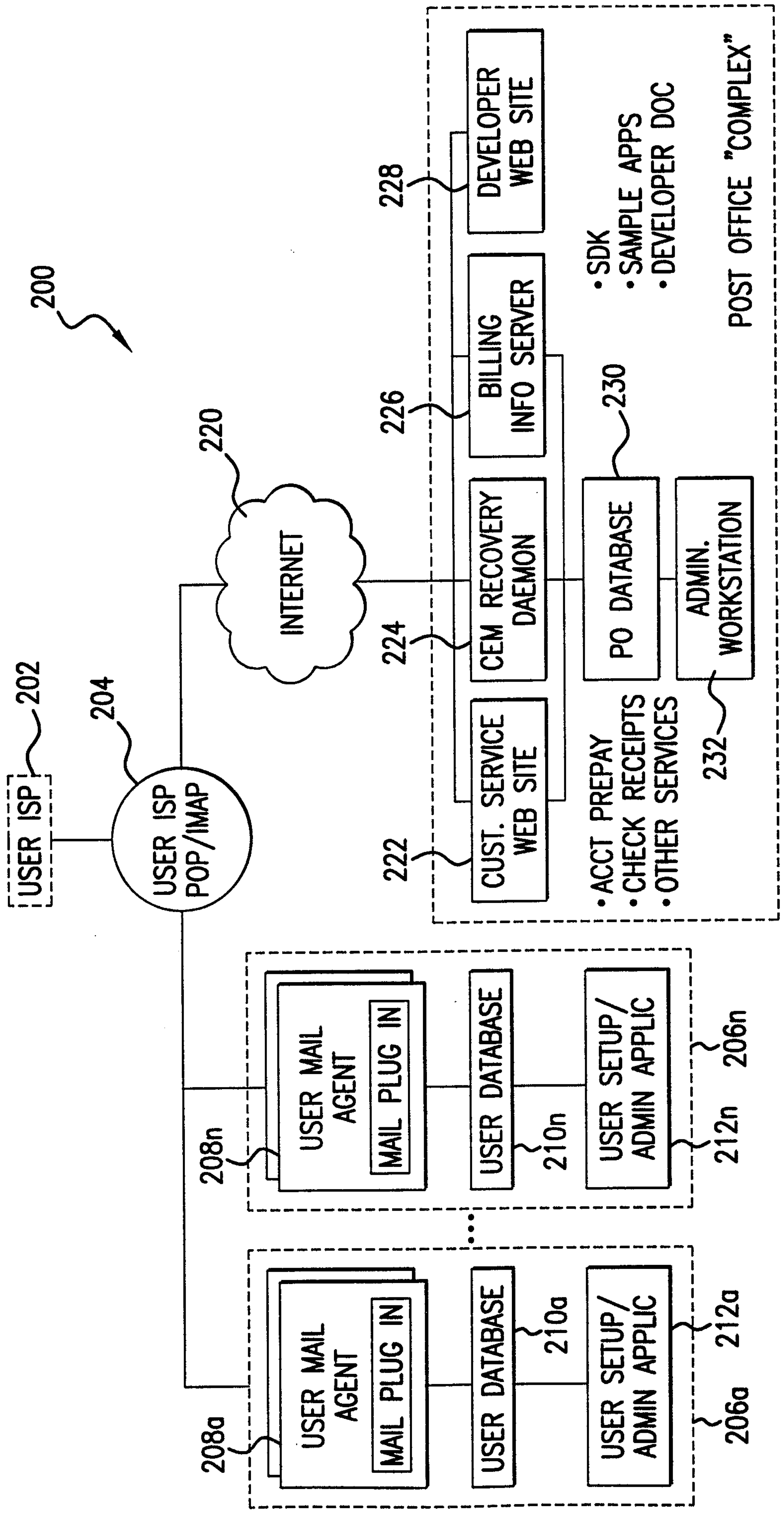

200
USER ISP
202
USER ISP POP/IMAP
204
INTERNET
220
CUST. SERVICE WEB SITE
222
CEM RECOVERY DAEMON
224
BILLING INFO SERVER
226
DEVELOPER WEB SITE
228
PO DATABASE
230
ADMIN. WORKSTATION
232
•ACCT PREPAY
•CHECK RECEIPTS
•OTHER SERVICES
•SDK
•SAMPLE APPS
•DEVELOPER DOC
POST OFFICE "COMPLEX"
USER MAIL AGENT
MAIL PLUG IN
208n
USER DATABASE
210n
USER SETUP/ ADMIN APPLIC
212n
206n
USER MAIL AGENT
MAIL PLUG IN
208a
USER DATABASE
210a
USER SETUP/ ADMIN APPLIC
212a
206a

FIG.2

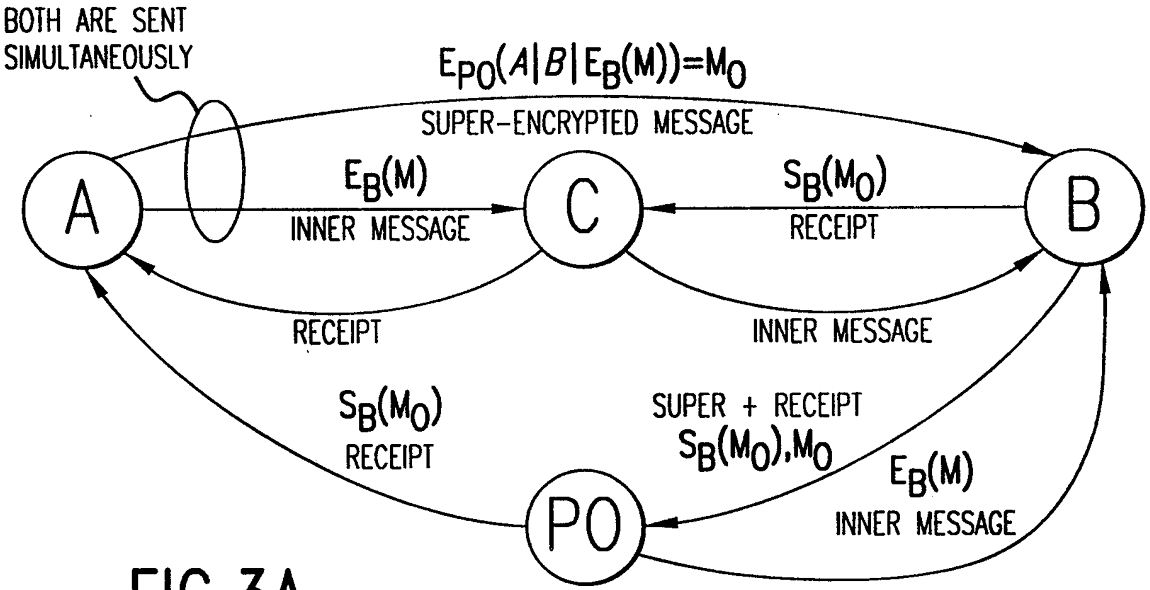

BOTH ARE SENT
SIMULTANEOUSLY
$E_{PO}(A|B|E_B(M))=M_O$
SUPER-ENCRYPTED MESSAGE
A
$E_B(M)$
INNER MESSAGE
C
$S_B(M_O)$
RECEIPT
B
RECEIPT
INNER MESSAGE
$S_B(M_O)$
RECEIPT
SUPER + RECEIPT
$S_B(M_O),M_O$
$E_B(M)$
INNER MESSAGE
PO

FIG.3A

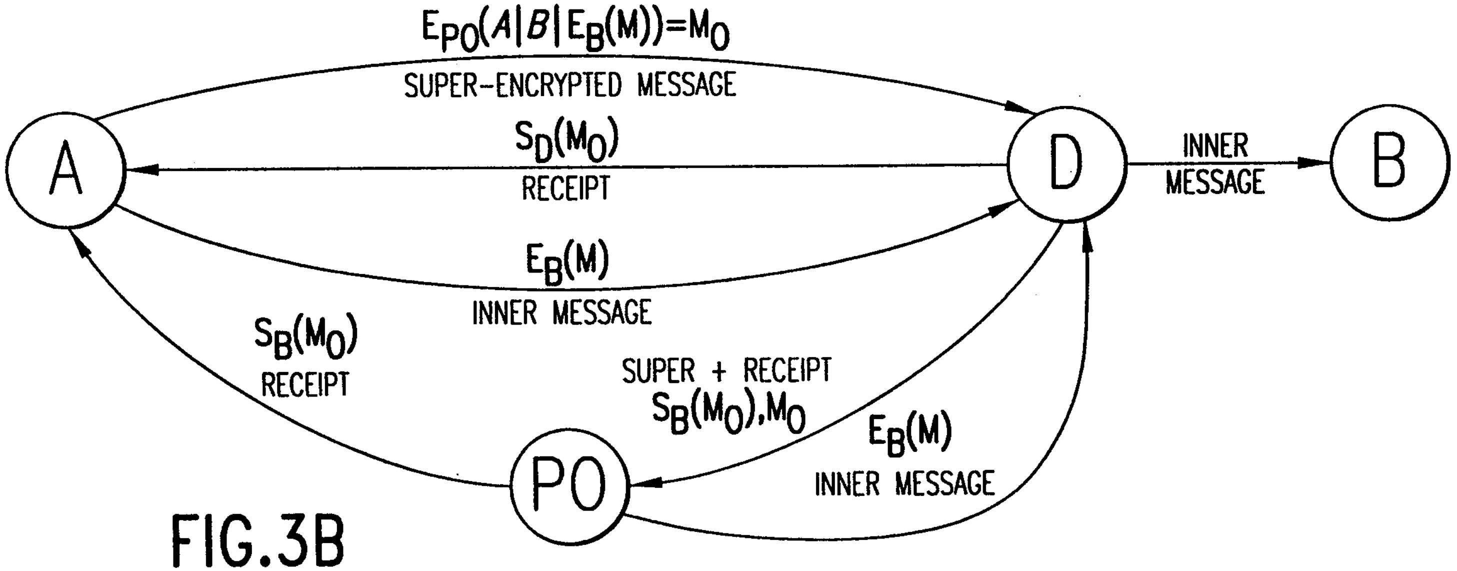

$E_{PO}(A|B|E_B(M))=M_O$
SUPER-ENCRYPTED MESSAGE
$S_D(M_O)$
RECEIPT
$E_B(M)$
INNER MESSAGE
A
D
INNER
MESSAGE
B
$S_B(M_O)$
RECEIPT
SUPER + RECEIPT
$S_B(M_O),M_O$
$E_B(M)$
INNER MESSAGE
PO

FIG.3B

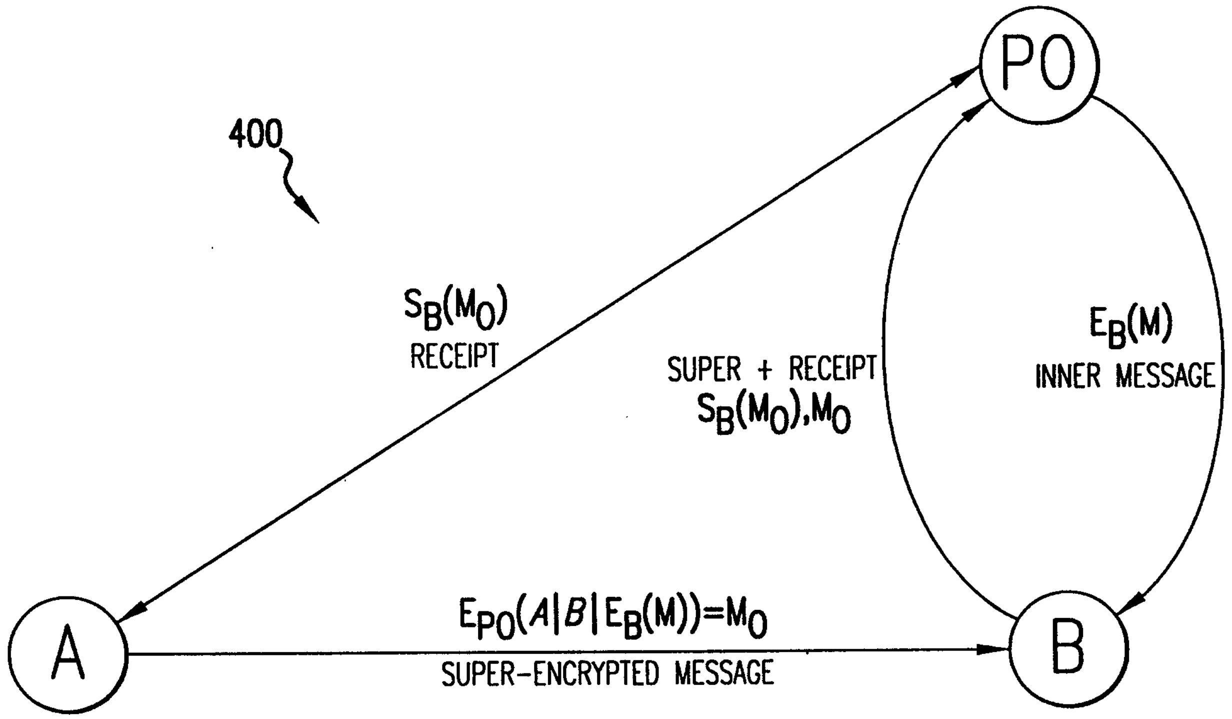

400
PO
A
B
$S_B(M_O)$
RECEIPT
SUPER + RECEIPT
$S_B(M_O),M_O$
$E_B(M)$
INNER MESSAGE
$E_{PO}(A|B|E_B(M))=M_O$
SUPER-ENCRYPTED MESSAGE

FIG.4

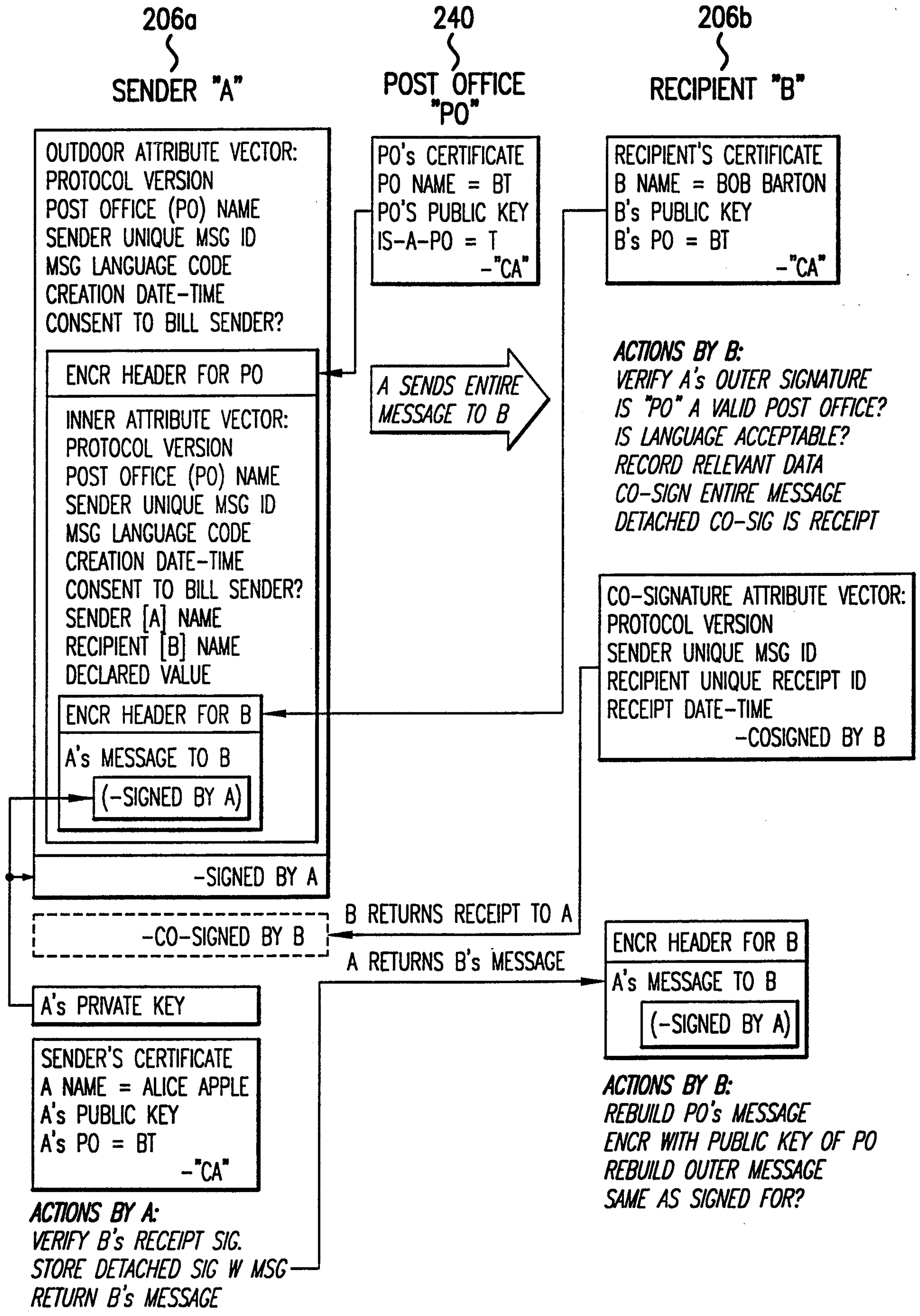

206a
SENDER "A"
240
POST OFFICE "PO"
206b
RECIPIENT "B"
OUTDOOR ATTRIBUTE VECTOR:
PROTOCOL VERSION
POST OFFICE (PO) NAME
SENDER UNIQUE MSG ID
MSG LANGUAGE CODE
CREATION DATE-TIME
CONSENT TO BILL SENDER?
ENCR HEADER FOR PO
INNER ATTRIBUTE VECTOR:
PROTOCOL VERSION
POST OFFICE (PO) NAME
SENDER UNIQUE MSG ID
MSG LANGUAGE CODE
CREATION DATE-TIME
CONSENT TO BILL SENDER?
SENDER [A] NAME
RECIPIENT [B] NAME
DECLARED VALUE
ENCR HEADER FOR B
A's MESSAGE TO B
(-SIGNED BY A)
-SIGNED BY A
-CO-SIGNED BY B
A's PRIVATE KEY
SENDER'S CERTIFICATE
A NAME = ALICE APPLE
A's PUBLIC KEY
A's PO = BT
-"CA"
ACTIONS BY A:
VERIFY B's RECEIPT SIG.
STORE DETACHED SIG W MSG
RETURN B's MESSAGE
PO's CERTIFICATE
PO NAME = BT
PO'S PUBLIC KEY
IS-A-PO = T
-"CA"
A SENDS ENTIRE MESSAGE TO B
B RETURNS RECEIPT TO A
A RETURNS B's MESSAGE
RECIPIENT'S CERTIFICATE
B NAME = BOB BARTON
B's PUBLIC KEY
B's PO = BT
-"CA"
ACTIONS BY B:
VERIFY A's OUTER SIGNATURE
IS "PO" A VALID POST OFFICE?
IS LANGUAGE ACCEPTABLE?
RECORD RELEVANT DATA
CO-SIGN ENTIRE MESSAGE
DETACHED CO-SIG IS RECEIPT
CO-SIGNATURE ATTRIBUTE VECTOR:
PROTOCOL VERSION
SENDER UNIQUE MSG ID
RECIPIENT UNIQUE RECEIPT ID
RECEIPT DATE-TIME
-COSIGNED BY B
ENCR HEADER FOR B
A's MESSAGE TO B
(-SIGNED BY A)
ACTIONS BY B:
REBUILD PO's MESSAGE
ENCR WITH PUBLIC KEY OF PO
REBUILD OUTER MESSAGE
SAME AS SIGNED FOR?

FIG.5A

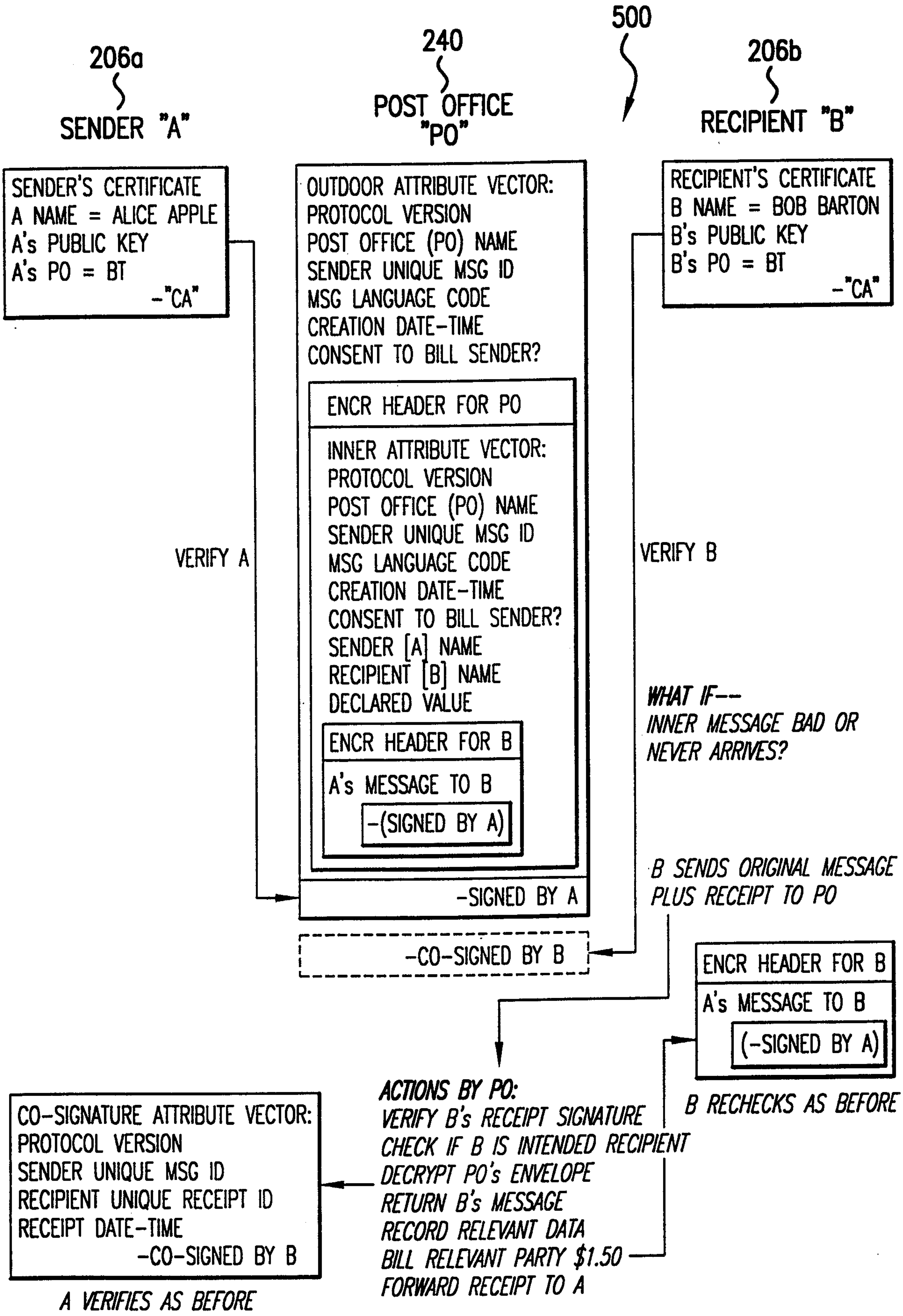

500
206a
SENDER "A"
240
POST OFFICE "PO"
206b
RECIPIENT "B"
SENDER'S CERTIFICATE
A NAME = ALICE APPLE
A's PUBLIC KEY
A's PO = BT
-"CA"
OUTDOOR ATTRIBUTE VECTOR:
PROTOCOL VERSION
POST OFFICE (PO) NAME
SENDER UNIQUE MSG ID
MSG LANGUAGE CODE
CREATION DATE-TIME
CONSENT TO BILL SENDER?
ENCR HEADER FOR PO
INNER ATTRIBUTE VECTOR:
PROTOCOL VERSION
POST OFFICE (PO) NAME
SENDER UNIQUE MSG ID
MSG LANGUAGE CODE
CREATION DATE-TIME
CONSENT TO BILL SENDER?
SENDER [A] NAME
RECIPIENT [B] NAME
DECLARED VALUE
ENCR HEADER FOR B
A's MESSAGE TO B
-(SIGNED BY A)
-SIGNED BY A
-CO-SIGNED BY B
RECIPIENT'S CERTIFICATE
B NAME = BOB BARTON
B's PUBLIC KEY
B's PO = BT
-"CA"
VERIFY A
VERIFY B
WHAT IF--
INNER MESSAGE BAD OR
NEVER ARRIVES?
B SENDS ORIGINAL MESSAGE
PLUS RECEIPT TO PO
ENCR HEADER FOR B
A's MESSAGE TO B
(-SIGNED BY A)
B RECHECKS AS BEFORE
ACTIONS BY PO:
VERIFY B's RECEIPT SIGNATURE
CHECK IF B IS INTENDED RECIPIENT
DECRYPT PO's ENVELOPE
RETURN B's MESSAGE
RECORD RELEVANT DATA
BILL RELEVANT PARTY $1.50
FORWARD RECEIPT TO A
CO-SIGNATURE ATTRIBUTE VECTOR:
PROTOCOL VERSION
SENDER UNIQUE MSG ID
RECIPIENT UNIQUE RECEIPT ID
RECEIPT DATE-TIME
-CO-SIGNED BY B
A VERIFIES AS BEFORE

FIG.5B

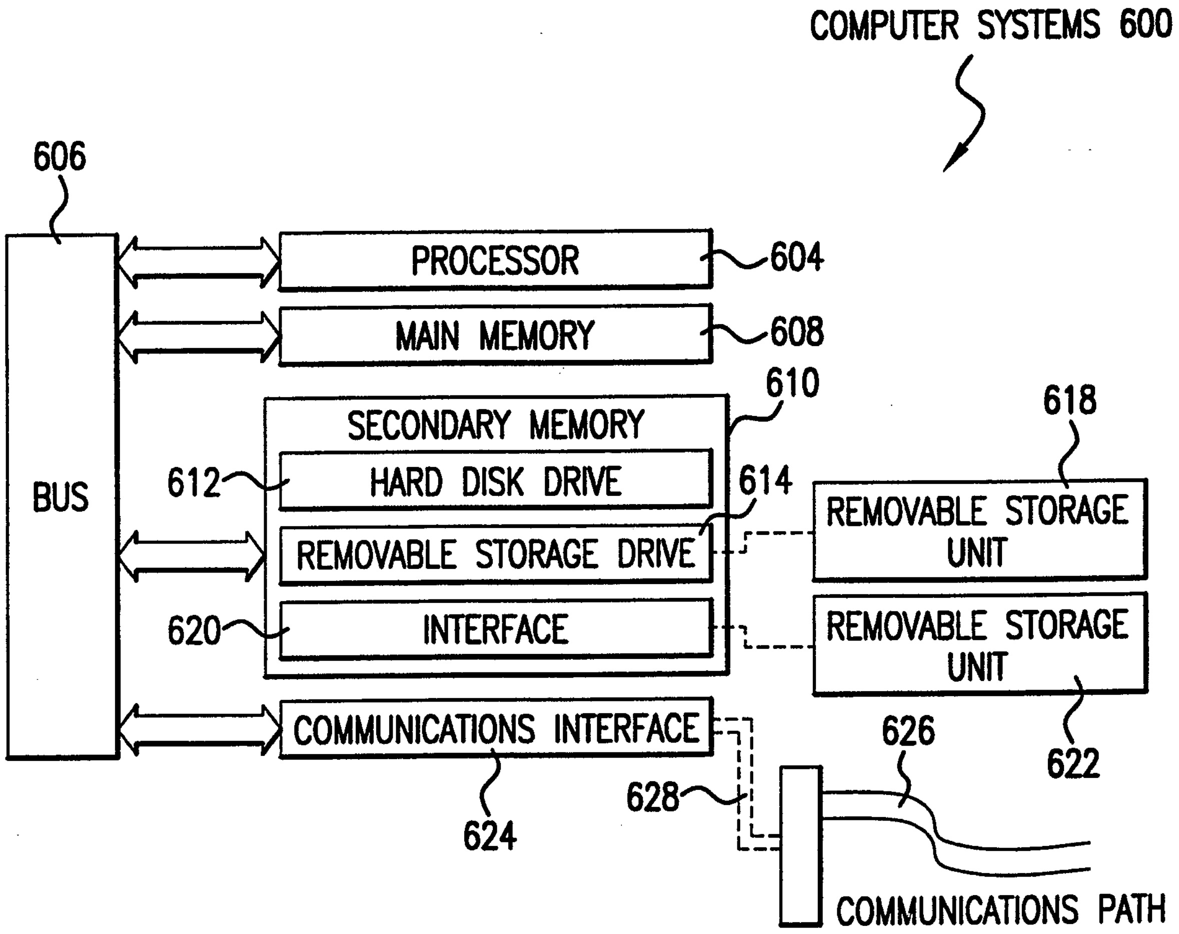

COMPUTER SYSTEMS 600
606
BUS
PROCESSOR
604
MAIN MEMORY
608
610
SECONDARY MEMORY
612
HARD DISK DRIVE
614
REMOVABLE STORAGE DRIVE
620
INTERFACE
618
REMOVABLE STORAGE UNIT
REMOVABLE STORAGE UNIT
622
COMMUNICATIONS INTERFACE
624
628
626
COMMUNICATIONS PATH

FIG.6