DIVIDED INFRINGEMENT

STRATEGIES FOR DRAFTING CLAIMS

DALE S. LAZAR
DLA PIPER LLP (US)

DIVIDED INFRINGEMENT

STRATEGIES FOR DRAFTING CLAIMS

Divided infringement issues are most likely to arise in the context of large systems and corresponding methods, where multiple parties may contribute to or participate in the system. A common instance where divided infringement is likely to arise is a network of computers where the functionality of the system is distributed across the computers of the network. The network employed to connect the computers may be a LAN/WAN and/or the Internet. Particularly where different computers perform different functions for the overall system, the different computers may be owned or operated by different parties, typically with each party not being controlled by other parties involved with the system.

Included at the end of this paper is a patent application directed to a "Reliance Server for a Transaction System". In many ways, the invention is typical of Internet-based inventions - inventions that are implemented over an open communications channel, such as the Internet. The application has been simplified to focus on only a few of the features of the invention. Please read

the application as a basis for understanding the following remarks concerning divided infringement.

CLAIM DRAFTING SUGGESTIONS

Draft System Claims as Well as Claims Directed to the Actions and Apparatus of Each Party in Multi-Party Systems

Frequently inventions implemented over the Internet involve the interactions of several parties. In the application at the end of this paper, the transaction system involves the interactions of a subscriber mechanism, a relying party mechanism, a reliance manager mechanism and a certification authority mechanism. In drafting claims to such a multi-party system, you should consider claims directed to the overall system, such as claims 1 and 5 in the application at the end of the paper. However, the Internet readily enables different parties to use a system. Those different parties may even be located in different countries. If different, unrelated parties operate different components of a system claim, who directly infringes? The problem only gets worse if the parties reside in different countries. Perhaps one entity or several entities under common control have provided software to all the parties with which the system is implemented. Thus the entity responsible

for the overall system might be an infringer of the system claims. Damage might be based on the value of the system.

However, the value of a patent is maximized by maximizing the possibilities of direct infringement. Thus, not only is it desirable to consider system claims, but it is also desirable to have claims specifically directed to the actions of each party or portion of a system. In the application at the end of the paper, claims 2, 6, 9, 12 and 15 are all directed to the actions of a relying party mechanism and the equipment that a relying party mechanism must have to implement the system. Thus, for example, method claim 2 simply comprises receiving signals representing a transaction, creating a reliance request message, and causing signals representing the reliance request message to be sent to a reliance server. These are the actions taken by a relying party mechanism in the transaction system. These claims do not require the actions of any other party. A relying party in the United States would directly infringe this claim and the other claims directed to the relying party.

Similarly, claims 3, 7, 10, 14 and 16 are all directed to the actions taken by the reliance manager mechanism and the

apparatus that a reliance manager mechanism would need to implement this system. For example, method claim 3 recites receiving signals representing a reliance request message, determining whether to provide transactional assurance, and generating signals representing an indication of whether transactional assurance is available. Since only the actions of a reliance manager are claimed, a reliance manager in the United States would directly infringe these claims.

Claims 4, 8, 11, 13 and 17 are directed to the actions of a certification authority mechanism and the apparatus that a certification authority would need to implement the system. These claims would be infringed by a certification authority in the United States operating within the system described in the application.

By separately claiming the actions taken or apparatus needed for each of the participants in a system, we have maximized the likelihood of finding a direct infringer of at least one claim of the patent and avoiding issues of divided infringement.

Draft Claims to Multiple Classes of Invention

The strength of a patent is maximized by claiming an invention in as many different ways as possible. Not only do we want to present claims directed to the actions of each participant in a multi-party system, but we also want to present claims directed to multiple classes of invention. This is demonstrated in the application at the end of this paper. Thus, for example, claims 5-8 are directed to a structured transaction system. These claims take a structural approach to various aspects of the system.

Claims 1-4 are method claims. Each of the method claims is directed to a different aspect of the system.

Claims 9-11 are directed to a computer-readable medium tangibly embodying a program of instructions executable by a computer to perform a method. The body of the claim recites the method that the computer can perform. Thus, these claims are directed to the combination of computer-readable media and software. By indicating that the software is executable by a program to perform a method, we have maximized the likelihood that the claim will be determined as reciting a practical utility, meeting the standard of §101.

Claims 12-14 are directed to a memory medium in a transaction system, in which the medium stores software programmed to implement a method. The body of the claim recites that method. As with claims 9-11, by suggesting that the software in the memory implements aspects of a structured transaction system, we have enhanced the likelihood that the claim will meet the requirements of §101.

Finally, claims 15-17 are directed to at least one computer programmed to execute a particular process. The body of the claim recites that process.

SAMPLE PATENT APPLICATION:
RELIANCE SERVER FOR A TRANSACTION SYSTEM

[0001] This invention relates to remote transactions, and, more particularly, to services supporting reliance on digital signature certificates and managing the risk of such certificates in a structured transaction system.

[0002] In the accompanying drawing:

[0003] **FIGURE 1** shows an overview of a structured transaction system according to an embodiment of the present invention; and

[0004] **FIGURES 2** and **3** are flowcharts depicting the operation of aspects of a transaction system according to embodiments the present invention.

[0005] The term "message" generally refers to a signal representing a digital message. As used herein, the term "mechanism" is used herein to represent hardware, software or any combination thereof. The mechanisms and servers described herein can be implemented on standard, general-purpose computers or they can be implemented as specialized devices. The

mechanisms and servers may operate electronically, optically or in any other fashion.

A. Overview

[0006] An overview of the structured transaction system **200** is described with reference to **FIGURE 1**. A subscriber mechanism **202** is issued one or more certificates **204** from a certification authority mechanism within an hierarchy of certification authority mechanisms **206** or from one of a number of sponsor mechanisms **208**. The certificates may serve to identify the subscriber mechanism **202** or to authorize certain transactions or types of transactions by the subscriber mechanism **202**. Copies of the certificates (or of relevant information from the certificates) is placed in repositories or directory mechanisms **210**. Each certification authority mechanism and sponsor mechanism may have its own directory mechanism **210**, or they may share directory mechanisms **210**.

[0007] The subscriber mechanism **202** transacts with a party mechanism **212** (hereinafter the relying party mechanism) by forming and digitally signing a message encompassing a

transaction **214** which includes those of the subscriber's certificates (or unique identifiers of the subscriber's certificates) required to identify the subscriber mechanism and to validate and authorize the transaction and then sending the transaction **214** to the relying party mechanism **212**.

[0008] Upon receipt of the signed transaction **214**, the relying party mechanism **212** verifies as much of the transaction **214** that it can or that it wishes to, and then composes a message **216** which it then sends to a reliance manager mechanism **218**. The message **216** can be one of various different kinds of messages, including either a signature guarantee request (SGR) message, and a status check message (SCM).

[0009] Either at this time or before a proposed transaction has been initiated, the relying party mechanism **212** has entered into a contract with the reliance manager mechanism **218**, for the reliance manager mechanism to perform services for the relying party mechanism and for allocating risks between the relying party mechanism and the reliance manager mechanism.

[0010] A purpose field can be included in each message **216** so that a reliance manager mechanism **218** knows which tasks

to perform. An SGR informs a reliance manager mechanism **218** that the relying party mechanism **212** will be relying on certain information included with the message (derived from the transaction **214**), and asks the reliance manager mechanism **218** to verify that the information is reliable and to guarantee the results of the check. For example, an SGR may specify that a relying party mechanism **212** will be relying on certain certificates for a $200 transaction, and the reliance manager mechanism **218** is requested to check that the transaction will be good for that amount.

[0011] A status check message is similar in form to an SGR, except that the relying party mechanism **212** does not actually request a guarantee, only an indication that such a guarantee would be given if requested.

[0012] When the message **216** is an SGR or an SCM, it contains enough information for the reliance manager mechanism **218** to verify the subscriber information in the transaction **214**. The relying party mechanism **212** may also specify in the message **216** a category of transaction as well as those aspects of the subscriber's information in the transaction **214** (or, more precisely,

in the certificates associated with transaction **214**) on which it will rely.

[0013] As noted, the relying party mechanism **212** can verify as much of the transaction **214** that it can or that it wishes to. Thus, for example, a relying party mechanism **212** may verify all signatures, certificates and attribute values within the transaction and then just request that the reliance manager mechanism **218** check the certificate serial numbers against CRLs. Alternatively, the relying party mechanism **212** may send the entire transaction **214** to the reliance manager mechanism **218** for verification, doing nothing itself. The cost of the verification services performed by the reliance manager mechanism **218** can depend on the amount of work it is requested to perform.

[0014] When the reliance manager mechanism **218** gets a message **216** from a relying party mechanism **212**, it first determines what kind of message it has received. If the message is an SGR or SCM, the reliance manager mechanism **218** tries to verify the information in the certificates provided by the subscriber mechanism **202** to the relying party mechanism **212** in the transaction **214**. To verify this information, the reliance manager

mechanism may check with certification authority mechanisms **206** and sponsor mechanisms **208**, or it may rely on information (e.g., CRLs or information from previous checks) that it has previously obtained from those parties or from elsewhere, e.g., from directory mechanisms **210**.

[0015] The reliance manager mechanism **218** tracks the cumulative liability of each certification authority mechanism **206** and sponsor mechanism **208**, and periodically notifies them of this liability. The regularity of this notification may depend on the arrangement between the reliance manager mechanism **218** and the parties, or it may depend on the type or size of the transaction or liability. For example, in some cases a certification authority mechanism may wish to be notified immediately of certain transactions or types of transactions, such as transactions exceeding a certain amount of money, transactions by particular entities, transactions which would cause its cumulative liability to exceed some value, transactions at certain times of day or any combination these and other conditions. In this way, a certification authority mechanism **206** or a sponsor mechanism **208** would be

able to act immediately, if necessary, to insure for liability against those transactions.

[0016] There may be more than one reliance manager mechanism **218**, and different transactions **212** or different parts of the same transaction **214** may have to be verified by different reliance manager mechanisms. In order for the various reliance manager mechanisms **218** to track the cumulative liability associated with each outstanding certificate, global liability tracking servers **220** are used. Each liability tracking server **220** acts as a global shared memory for the transaction system **200**, allowing cumulative liabilities associated with each outstanding certificate to be read and written by reliance manager mechanisms **218**. Only one liability tracking server **220** may be used for each certificate. The liability tracking servers **220** can be separate entities or they can be a part of the directory mechanisms **210**, the CA mechanisms **206**, or the sponsor mechanisms **208**. If a particular certificate can only be processed by one reliance manager mechanism, then that reliance manager mechanism can track the cumulative liability associated with that certificate. These liability tracking servers provide a general “inhibit” function

to detect and prevent over-reliance on a certificate. The "inhibit" service is generally performed by a high availability system under contract to the issuing CA mechanism.

[0017] Certificates can specify a reliance limit or a reliance limit per period of time, e.g., per hour, day, week, month, year, 24 hour period, weekday, etc. Thus, one certificate may have a reliance limit of $200 per day, while another has a reliance limit of $500. Similarly, certificates can specify a number of transactions per time period, e.g., per hour, day, week, month, year, 24 hour period, weekday etc. Thus a certificate may specify ten transactions per day. Combinations of these may apply, e.g., five transactions per day, not to exceed $500 per day.

[0018] The liability trackers **220** store the current cumulative liability and the number of transactions for each certificate for the period stated in the certificate. The certificates can be indexed based on their unique identity (issuer name and certificate serial number).

[0019] Since a certificate may specify a class or category of reliance manager mechanisms which can be used to validate that certificate, simultaneous attempts to read/write values for a

particular certificate at a liability tracker **220** are possible. That is, it is possible that more than one reliance manager mechanism **218** is processing a copy of the same certificate and that more than one reliance manager mechanism is requesting reliance based on that certificate. Accordingly, the liability trackers **220** use an appropriate locking mechanism to ensure consistent reading and updating of their records.

[0020] Each reliance manager mechanism **218**, certification authority mechanism **206** and sponsor mechanism **208** may, at any time, insure itself against some liability by obtaining insurance from an insurer mechanism **222**. The reliance manager mechanism **218** may be authorized to obtain insurance from insurer mechanisms **222** on behalf of a certification authority mechanism **206** or sponsor mechanism **208**, depending on such factors as that authority mechanism or sponsor mechanism's current pending cumulative liability. The reliance manager mechanism **218** may also obtain its own insurance from the insurer mechanisms **222**.

[0021] The reliance manager mechanism **218** bills the various parties for its services via billing service mechanism **224**.

The reliance manager mechanism **218** also bills the appropriate party for the use of the certificates being relied upon by the relying party mechanism **212**. This may take the form of requiring an immediate payment over a network, e.g., from an unknown relying party mechanism, debiting a pre-established deposit account, e.g., of a sponsor mechanism, or sending a periodic invoice to a sponsor mechanism or relying party mechanism with established credit and payment history.

[0022] Having processed a message **216** (e.g., verified an SGR or SCM), notified the appropriate parties, obtained the appropriate insurance and billed for the services provided, the reliance manager mechanism **218** then sends a reliance manager receipt (RMR) **226** to the relying party mechanism **212**. This RMR **226** informs the relying party mechanism **212** of the outcome of the status checks and of the amounts charged for those checks, or, in the case of an over-limit guarantee request, of the response to that request, which may be either a guarantee receipt or a reject message. The RMR receipt **226** can be digitally signed by the reliance manager mechanism **218**, with the date and time and a digest of the message, thereby acting as proof of the verification

performed by the reliance manager mechanism. The reliance manager mechanism **218** can, if needed, archive the message **216**, the signed receipt **226** and any other information related to the processing of that message **216**.

[0023] The transaction **214**, the message **216** and the RMR **226** can be digitally signed by an independent timestamp server **228** when created.

[0024] Upon receipt of the RMR **226**, the relying party mechanism **212** can store the RMR and, depending on the information in the RMR **226**, continue the transaction with the subscriber mechanism **202**.

[0025] While shown in **FIGURE 1** as separate entities, the billing service mechanism **224** and the reliance manager mechanism **218** can be part of the same entity, in which case the various parties to a transaction (e.g., certification authority mechanisms, sponsor mechanisms, subscriber mechanisms and vendors) can have accounts with the reliance manager mechanism **218**. The reliance manager mechanism will keep separate totals for status check fees owed to itself and reliance fees owed to CA mechanisms, sponsor mechanisms and insurer mechanisms, and

will periodically perform a separation and settlement of all these charges, collecting any funds due and remitting all funds collected to the appropriate parties, less any service fees.

[0026] Further, while shown in **FIGURE 1** as separate entities, the subscriber mechanism **202** may, in some circumstances, itself be the relying party mechanism **212**. For example, a subscriber mechanism **202** may wish to determine whether a particular transaction would be acceptable for a signature guarantee before sending that transaction to another party.

B. Detailed Description

1. Relying Party Mechanism Receives Transaction

[0027] When a relying party mechanism **212** receives a transaction **214** formed as described above, it can do a number of things, depending on how much information is contained in the transaction **214** and on how much it wants the reliance manager mechanism **218** to do.

[0028] With reference to **FIGURES 2** and **3**, the relying party mechanism **212** receives the transaction **214** (at **S200**) and performs some or all of the following steps:

[0029] If the relying party mechanism **212** wants the entire transaction **214** checked by a reliance manager mechanism **218**, it proceeds by determining the address(es) of the appropriate reliance manager mechanism(s) **218** to which the entire transaction **214** will be sent (at **S210**).

[0030] Otherwise, the relying party mechanism **212** extracts the copy of the subscriber's certificate **242** and the other certificates **244** (if any) from the transaction **214** (at **S202**). Recall that some or all of these certificates **242**, **244** may not be present. They may have been sent previously, be obtainable from directory mechanisms, or be left for the reliance manager mechanism to obtain. The transaction **214** should contain at least a unique identifying reference (e.g., issuer name and serial number) to the subscriber's certificate **204**.

[0031] At this point the relying party mechanism **212** can either further validate the transaction **214** itself, or it can go

directly to the address determination step (**S210**), letting the reliance manager mechanism **218** check the entire transaction **214**.

[0032] If the relying party mechanism **212** decides to further verify the transaction **214**, it then retrieves any missing certificates (at **S204**). The missing certificates are obtained from the appropriate directory mechanism **210** or the relying party mechanism **212** may have retained copies of them from prior retrievals. This certificate retrieval process is done by working upward from each provided certificate to find all parent certificates, working toward a known good root key.

[0033] Again, at this point, the relying party mechanism **212** can continue to verify the transaction **214** itself, or it can go directly to the address determination step (**S210**), giving the reliance manager mechanism **218** more of the transaction **214** to check.

[0034] Having obtained all the required certificates (at **S202**, **S204**), the recipient verifies the certificate chain (at **S206**), working downward from the root to the signer, verifying policy controls in the certificates on the way down. The relying party

2. Relying Party Mechanism Determines Address(es) of Status Service(s)

[0035] Before having any aspect of a transaction **214** verified by one or more reliance manager mechanisms **218**, the relying party mechanism **212** may determine which reliance manager mechanism(s) to use. Each subscriber certificate includes either the name of a status checking service (reliance manager mechanism **218**) at which that certificate can be checked as well as a suitable address for its directory mechanism **210**, or the relying party mechanism **212** may obtain that information from the appropriate certifying authority mechanism **206**. Thus, given a certificate, it is possible to determine which reliance manager mechanism to use to verify aspects of that certificate.

[0036] Accordingly, at **S210**, if the relying party mechanism **212** has not already obtained at least one appropriate certificate (at **S202**), it does so and determines from that certificate the name of a reliance manager mechanism **218** (or a class of

reliance manager mechanisms) at which that certificate can be checked.

[0037] On the other hand, if the relying party mechanism had already obtained at least one certificate (at **S202**), it uses that certificate to determine the name of a reliance manager mechanism **218** (or a class of reliance manager mechanisms) at which that certificate can be checked.

[0038] If the relying party mechanism obtained all certificates associated with the transaction, it determines the appropriate reliance manager mechanism **218** for each certificate.

3. Relying Party Mechanism Creates A Message

[0039] Having received a transaction **214** (at **S200**), and verified as much of that transaction as it desires (at **S202-S208**), the relying party mechanism **212** next creates one or more messages (either SGR, an SCM) **216** (at **S212**) to be sent to one or more reliance manager mechanisms **218**.

[0040] However, before creating any messages **216** (at **S212**), the relying party mechanism **212** may determine which reliance manager mechanism(s) **218** to send the message(s) **216** to.

Since there might be more than one message **216** if multiple reliance manager mechanisms are involved, the relying party mechanism **212** may first determine how many reliance manager mechanisms are involved.

[0041] When a certification authority mechanism **206** or sponsor mechanism **208** issues a certificate, it specifies, in the certificate, a status checking service (reliance manager mechanism **218**) at which this certificate can be checked. The status checking service can be specified by name, class of provider or in some other manner. Thus, a certificate issuer mechanism may determine that only a particular reliance manager mechanism **218** can verify its certificates (and so specify in its certificates), or it may allow its certificates to be verified by any reliance manager mechanism **218** that meets certain requirements (as specified by a reliance manager mechanism class).

[0042] The relying party mechanism **212** next analyzes the certificates requiring status checking in order to determine the address(es) of status services (reliance manager mechanism(s) **218**) to use for transaction verification (at **S210**).

[0043] The contents of the message(s) depend on what it is the relying party mechanism **212** wants the reliance manager mechanism(s) **218** to do. First, the relying party mechanism **212** may provide the reliance manager mechanism(s) **218** with the entire transaction **214** or with only parts of the transaction. Second, the relying party mechanism **212** may provide the reliance manager mechanism(s) **218** with all certificates associated with the transaction or it may provide only unique identifiers for only some of the certificates. Third, the relying party mechanism **212** may ask the reliance manager mechanism(s) **218** to validate the entire transaction or only some aspects thereof.

[0044] The various functions that the relying party mechanism **212** can request of the reliance manager mechanism(s) **218** include:

[0045] 1. Given only the unique identifiers of certificates, check whether or not those certificates have been revoked (i.e., are listed on CRLs) or suspended.

[0046] 2. Given a set of actual certificates, check whether or not they have been revoked or suspended.

[0047] 3. Given a combination of certificates and unique certificate identifiers, check whether or not those certificates have been revoked or suspended.

[0048] 4. Verify certificate chain to see if certificates actually verify each other.

[0049] 5. Same as above in 1-4, but check entire transaction, including the digital signature of the original subscriber who signed it.

[0050] As noted above, the message **216** can be one of various different kinds of messages, including either an SGR (which informs a reliance manager mechanism **218** that the relying party mechanism **212** will be relying on certain information included with the message and asks the reliance manager mechanism **218** to verify that the information is reliable); an SCM (which is similar in form to an SGR, except that the relying party mechanism **212** does not actually request a guarantee, only an indication that such a guarantee would be given if requested).

[0051] If the message **216** is an SGR or an SCM, it includes a monetary value (for reliance purposes) which the relying party mechanism **212** initially determines from the

transaction **214**. If no monetary value is provided in the transaction **214** or if the relying party mechanism wants to rely on a different value than that provided (for example, if the relying party mechanism **212** self insures for some amount), the relying party mechanism can set the monetary value in the message **216** accordingly. However, to prevent the relying party mechanism **212** from stating a monetary value in excess of the value stated in the transaction **214**, thereby perhaps prematurely exhausting the subscriber's maximum allowed limits, the transaction's signature attributes **248** should include the actual stated value of the transaction, and the message **216** should include this actual stated value by including the signature attributes **248**.

[0052] If only a status check is desired, either a status check bit can be set in the message or the monetary value in the message **216** is set to zero. A status check bit can be used to avoid overloading of data values with semantics. In this case the message **216** (an SCM) will be used to request confirmation of the status of the various certificates, but will not be used to purchase any guarantees.

[0053] A message **216** (SGR or SCM) contains the following (not necessarily in the given order):

[0054] 1. The name and network address of the relying party mechanism **212**.

[0055] 2. A unique sequence number so duplicate messages (RCMs in particular) can be rejected.

[0056] 3. Account information with the billing service mechanism 224 for billing purposes (even if the signer or sponsors pay).

[0057] 4. An optional intent to request extension if over the allowed/remaining reliance limit.

[0058] 5. A list of certificates or certificate unique identifiers (e.g., issuer names and serial numbers).

[0059] 6. A signature sequence number and timestamp (if present in the transaction 214).

[0060] 7. A hash of the message (transaction 214) being checked. Optionally the entire message can be appended for checking and/or archiving.

[0061] 8. A date and time of request can be (provided by timestamp server 228).

[0062] 9. An optional list of categories for this transaction.

[0063] 10. A request to archive even if status check fails; or to archive even if over reliance limit.

[0064] 11. An archive retrieval password, encrypted with reliance manager service's public key.

[0065] 12. A purpose for this message (guarantee request, status check billing approval, etc.).

[0066] 13. A role (relying party)

[0067] 14. A hash of the relying party mechanism's billing service certificate.

[0068] 15. The relying party mechanism's signature for charging the account at the billing service mechanism 222.

4. Relying Party Mechanism Sends Message(s)

[0069] Having determined the address(es) (at **S210**) and created the appropriate message(s) (at **S212**), the relying party mechanism **212** then determines the total of base checking fees requested and the total of reliance guarantee fees requested per dollar amount of reliance value (at **S214**).

[0070] With the messages **216** created, they are sent (at **S216**) to the appropriate reliance manager mechanism(s) **218**. The messages **216** are sent using whatever transport mechanism is specified in directory entry for the reliance manager mechanism, e.g., sockets, HTTP, e-mail, and the like.

5. Processing by Reliance Manager Mechanism

a. Receive Message

[0071] Upon receipt of a message **216** from a relying party mechanism **212** (at **S218**), the reliance manager mechanism **218** performs the following operations (with reference to **FIGURE 3**).

b. Verify Message Syntax

[0072] First the reliance manager mechanism **218** verifies (at **S220**) that the general syntax of the message **216** is correct. If the syntax is incorrect, the reliance manager mechanism **218** notifies the relying party mechanism **212** and ceases processing. A relying party mechanism **212** may be billed for a syntactically incorrect message.

[0073] If the syntax of the message **216** is correct, then, if the relying party mechanism **212** has an account with the reliance manager mechanism **218** (i.e., if the reliance manager mechanism and the billing service mechanism **224**) are the same entity, the reliance manager mechanism looks up the relying party mechanism **212** public key which it has stored locally, hashes the message **216** and verifies the relying party's signature.

[0074] If the relying party mechanism **212** does not have a pre-established account with the reliance manager mechanism **218**, the reliance manager mechanism may verify the relying party's signature on the message **216** by some conventional approach, typically verifying the chain of certificates which specify the relying party's public key.

c. Determine Minimum Fees

[0075] Next, the reliance manager mechanism **218** determines the minimum amount due on this verification transaction and verifies available funds in the relying party's account (at **S222**). Sometimes the reliance manager mechanism **218** cannot determine the full amount due on a transaction until the entire transaction has been verified. If the reliance manager

mechanism **218** and the billing service mechanism **224** are separate entities, then the reliance manager mechanism can either bill the relying party's account with the billing service mechanism or contact the billing service mechanism to determine whether the relying party mechanism is in good standing with them and has funds or credit available.

d. Determine What is being Requested

[0076] Subsequent activities of the reliance manager mechanism **218** depend on what the relying party mechanism **212** requested in the message **216**. The reliance manager mechanism **218** determines what it is that is being requested (at **S224**) and validates the message accordingly (at **S226**).

e. Validate Message

[0077] In one possible case, the reliance manager mechanism **218** is given (in the message **216**) the unique identifiers of various certificates and/or actual certificates, along with a requested reliance limit.

[0078] The reliance manager mechanism **218** checks whether or not those certificates have been revoked (i.e., whether or not they are listed on CRLs) or suspended.

[0079] The reliance manager mechanism **218** may first check to determine whether the requested reliance is less than or equal to the value of the transaction **214**. The signature attributes block **248** of the transaction includes the value of the transaction, and this block, along with the actual signature, are provided to the reliance manager mechanism along with the relying party mechanism's requested liability. If the requested reliance exceeds the value of the transaction, the reliance manager mechanism should reject the request. In such cases, the reliance manager mechanism should notify the subscriber mechanism **202** and other parties of the request.

[0080] The reliance manager mechanism can thus verify the signature and from the block can extract the subscriber's declared transaction value prior to utilizing the subscriber's available reliance limit. If the entire transaction has been submitted, the RM mechanism can also hash it to see if it matches the transaction hash contained in the signature attributes block.

These processes of checking the signature block or the entire message can generally be omitted for smaller value message, where it is appropriate to rely on the declared values provided by the relying party mechanism. In such cases, the signature guarantee or other transaction insurance provided by the reliance manager mechanism will simply be void if there is any misstatement by the relying party mechanism, so it is not in his interest to submit incorrect values.

[0081] For each certificate listed in the message (either by serial number or by being provided), the reliance manager mechanism **218** checks that certificate's serial number against the appropriate CRL, i.e., the CRL from the issuer of that certificate. If the certificate has been revoked or suspended, the reliance manager mechanism notes the invalidity of that certificate and continues with any remaining certificates. If any certificate is invalid the entire transaction is considered invalid.

[0082] For each certificate checked, the reliance manager mechanism **218** can notify the issuer of its use and of the reliance value being associated with that certificate (at **S228**). Such

notification can be monthly, daily, weekly, hourly, or per transaction above a pre-approved amount.

f. Update Global Reliance Limits

[0083] The reliance manager mechanism **218** can also read the current cumulative liability for that certificate from the appropriate global liability tracker **220** assuming more than one RM mechanism can validate. If the current cumulative liability exceeds the requested liability, the transaction is invalid and is rejected. The reliance manager mechanism notifies the relying party mechanism if it will consider processing an over-limit guarantee request. Otherwise, if the current cumulative liability in addition to the requested liability does not exceed the limit on that certificate, the cumulative liability should be updated to reflect the requested amount (at **S230**).

[0084] Because of potential synchronous update attempts of the cumulative liability, the reliance manager mechanism may need to obtain a lock on the values for all certificates used by the transaction with the corresponding liability trackers before doing any reading or updating of values. Alternatively, it can use an optimistic commit strategy, write new values as each certificate is

processed, and roll back any change in the event of a later failure (over limit). The updating of these records can be performed in any manner known in the art. Further, in some cases, only a few of the certificates have their associated liability checked. The last one (of the subscriber mechanism **202**) is the most important one to monitor closely.

[0085] In some cases the reliance manager mechanism **218** will be asked to check an entire transaction. In these cases the reliance manager mechanism will obtain all the certificates associated with the transaction, check them for validity and consistency and then process them as above with respect to the liability limits requested.

[0086] Once the reliance manager mechanism **218** has determined the status of a particular certificate, the reliance manager mechanism **218** can create a record for that certificate. This record can be created regardless of the certificate's status. The record lists all certificates on which this certificate depends and all certificate which depend on this certificate. In this way, chains of certificates can be verified in a single table lookup. Whenever a CRL is received or whenever the reliance manager

mechanism determines that a particular certificate has become invalid (revoked or suspended) it can update its records, invalidating chains as appropriate.

[0087] The reliance manager mechanism also records all parties who have relied on each certificate. Whenever a CRL is processed or whenever the reliance manager mechanism determines that a particular certificate is invalid, the reliance manager mechanism can inform all parties who have relied on that certificate within a certain period of time, such as 1-2 weeks. This is sometimes referred to a "lookback notification," and it can help parties who recently relied on a certificate prior to its revocation determine if there may be any problems with prior transactions, e.g., possible fraudulent transactions issued by a thief prior to discovery of the theft of the signer's key, and reporting to the CA for certificate revocation.

[0088] If all certificates associated with a message are acceptable, then the reliance limit for the appropriate period on each certificate is incremented according to the reliance requested in the SGR.

g. Determine Fees and Bill Parties

[0089] The reliance manager mechanism **218** then writes the transaction for later batch processing, during which it increments guarantee fees collected on each certificate and accumulates guarantee fees by issuing certification authority name (at **S232**).

[0090] If any certificate is not acceptable then the relying party mechanism **212** (or the subscriber mechanism **202**) is billed only for the base checking fee, and the reliance and guarantee fees do not apply.

h. Archive Message if Requested

[0091] If the message **216** requests an archive and the full document is attached then the document is rehashed and the signature as submitted is verified. If the result of this is not the same as the transaction submitted then the archive request is rejected, otherwise the message is archived (at **S234**.)

[0092] The relying party mechanism **212** is then billed for the initial archive period requested, the default period being six months. The relying party mechanism **212** is notified that he will be billed for successive archive periods. The receipt provided by

the reliance manager mechanism contains a transcript, a hash of the transaction, and is signed using a long "archival" signature key (of at least 1800 bits) so the user can archive the transaction anywhere. It is, however, convenient to let the RM mechanism validate store the transaction as a "one-step shopping" arrangement.

i. Verify Signing Device

[0093] If the signer of the transaction **214** was a device (or device-confined subscriber private key), then the reliance manager mechanism **218** checks for anomalies based on the history of signatures produced by this device (verify device, at **S236**).

j. Obtain Insurance

[0094] In some instances the reliance manager mechanism **218** may obtain insurance (at **S238**), either to cover its own assumed risks in validating a transaction or on behalf of a certification authority mechanism or sponsor mechanism.

k. The Reliance Manager Mechanism's Response to the Relying Party Mechanism

[0095] Having verified the message **216** (to the extent requested by the relying party mechanism **212**), and having billed and notified the appropriate parties, issued or purchased the appropriate insurance and stored the required records, the reliance manager mechanism **218** then creates and sends a reliance manager response (RMR) **226** which it digitally signs and sends back to the relying party mechanism **212** via same method as the message **216** was sent, e.g., sockets, HTTP, e-mail, etc. (at **S240**). The receipt **226** can be timestamped by timestamp server **228** prior to being sent to the relying party mechanism **212**. The timestamp server can be a different physical and legal entity, so that if one or the other (reliance manager mechanism or timestamp service mechanism) is compromised, it will at least be impossible to backdate seemingly valid transactions.

[0096] The receipt **226** includes the following information:

[0097] 1. The identity of the reliance manager mechanism 218.

[0098] 2. A unique identifier (sequence number) for this receipt.

[0099] 3. The identity of the relying party mechanism 212 (and optionally its address).

[00100] 4. The relying party mechanism's unique message sequence number.

[00101] 5. A hash of the message checked.

[00102] 6. The date and time processed (declared value).

[00103] 7. The results of the request, including:

[00104] a. whether the certificate status checks were acceptable

[00105] b. whether the amount was within requested reliance limits, and

[00106] c. whether the message was archived, and, if so, an archive retrieval identification number

[00107] 8. If any status checks failed, a list of status/reason codes by certificate.

[00108] 9. The relying party mechanism's requested reliance limit (might be zero).

[00109] 10. If the signer (subscriber mechanism 202) is over her limits, whether the service (reliance manager mechanism 218) supports over-limit exception processing.

[00110] 11. If over-limit process was requested by relying party mechanism, what method is used for this.

[00111] 12. If the status checks and reliance limit were acceptable, a list of fees paid by certificate checked and total fees billed to subscriber and/or third parties, and the total fees billed to relying party mechanism's account.

[00112] 13. Whether the message had been previously checked and whether the subscriber/sponsors were out of funds.

[00113] 14. A signature of the status check service 218.

[00114] 15. A signature and time of timestamp server 228.

[00115] The receipt **226** may be in the form of a secondary certificate. This secondary certificate is issued automatically, based on other certificates and perhaps additional information gathered and maintained by the reliance server.

l. Final Processing by Relying Party Mechanism

[00116] When the relying party mechanism **212** gets a receipt **226** from a reliance manager mechanism **218**, it first checks that the receipt **226** has been sent to the correct relying party mechanism **212** by looking at the identity of the relying party mechanism as stated in the receipt **226**. Next the relying party mechanism **212** associates the receipt **226** with a message **216** which it sent out. To make this association, the relying party mechanism **212** checks the value of the unique message identifier (sequence number) in the receipt **226**. If the relying party mechanism **212** does not find a message **216** corresponding to this receipt **226**, or if the receipt has been sent to the wrong relying party mechanism, the recipient can either notify the reliance manager mechanism of the inconsistency found or simply ignore the receipt and do nothing.

[00117] Having determined that it is the correct recipient of the receipt **226** and having found the message **216** corresponding to the receipt, the relying party mechanism **212** verifies the signature of the reliance manager mechanism and

evaluates the receipt to determine the outcome of the reliance manager mechanism's processing. In other words, the relying party mechanism **212** examines the receipt **226** to determine whether the message **216** passed the requested certificate status checks; whether the amount was within requested reliance limits; and whether the message was archived. If the message was archived, the relying party mechanism stores the archive retrieval identification number.

[00118] The relying party mechanism **212** then detaches and stores the receipt **226** which serves as an advice. Note that the receipt is good only with respect to this specific relying party mechanism and may not be relied upon by another party without payment of an additional signature guarantee fee. Where a prior receipt is submitted, the reliance manager mechanism may offer a discount to subsequent relying party mechanisms for the same document or transaction.

[00119] Next, if the receipt **226** indicates that the status checks and reliance limit were acceptable, the relying party mechanism **212** evaluates, records, and deducts the fees billed to its account with the billing service mechanism **224**.

[00120] If the status checks fail because the subscriber mechanism **202** is over its limits, the relying party mechanism determines from the receipt **226** whether the reliance manager mechanism **218** supports over-limit exception processing, and then decides whether to request such processing.

[00121] Recall that a particular transaction **214** may require that messages be sent to more than one reliance manager mechanism **218**. The relying party mechanism **212** may wait for all reliance manager mechanisms to respond with receipts **226** before it can make a final determination as to whether to proceed with the transaction **214**. Accordingly, the relying party mechanism **212** checks to determine whether there are any outstanding reliance manager requests for the transaction associated with this receipt/message pair. If not, the relying party mechanism can continue with the transaction **214** with the subscriber mechanism **202**, otherwise it continues to wait for replies from other reliance manager mechanisms to other messages **216** associated with the transaction **214**.

[00122] Based on the outcome reported in all receipts **226** (corresponding to a particular transaction), the relying party

mechanism **212** can continue with the transaction. That is, having performed all checks and received guarantees and assurances, it may now proceed to actually rely on the transaction.

[00123] Thus, a reliance manager mechanism for a transaction system is provided. One skilled in the art will appreciate that the present invention can be practiced by other than the described embodiments, which are presented for purposes of illustration and not limitation, and the present invention is limited only by the claims that follow.

What is claimed is:

1. A method of managing reliance in a transaction system, the method comprising:

a certification authority mechanism issuing a signal representing a time-based certificate to a subscriber mechanism;

forwarding, from the certification authority mechanism, a signal representing information about the certificate to a reliance server, the information including a unique identifier for the certificate and an actual reliance limit for the certificate;

the subscriber mechanism forming a signal representing a transaction based on the certificate and forwarding the transaction to a relying party mechanism;

the relying party mechanism sending a signal representing a reliance request message to the reliance server concerning the transaction;

the reliance server checking information in the reliance request message; and

based on the checking, issuing a signal representing a transactional certificate as a voucher to the relying party mechanism.

2. A method of managing reliance in an electronic transaction system in which subscriber mechanisms have certificates issued by certification authority mechanisms, the method comprising, by a relying party mechanism:

receiving a signal representing a transaction from a subscriber mechanism, the transaction including information regarding at least one certificate of that subscriber mechanism;

creating a message based on certificate information from the transaction, the message specifying an amount of the transaction upon which the relying party mechanism can rely; and

sending a signal representing the message to a reliance server requesting a guarantee for the amount of the transaction upon which the relying party mechanism can rely.

3. A method of managing reliance in a transaction system in which subscriber mechanisms have time-based

certificates issued by certification authority mechanisms, the method comprising, by a reliance server:

receiving a signal representing a reliance request message from a party mechanism, the message specifying an amount of a transaction upon which the party mechanism can rely and requesting a guarantee for the amount of the transaction, the message including certificate information derived from the transaction;

determining whether to provide a guarantee for the amount of the transaction; and

sending a signal representing a voucher to the relying party mechanism, the voucher including an indication of whether the reliance server guarantees the amount of the transaction.

4. A method of managing reliance in a transaction system, the method comprising, by a certification authority mechanism:

issuing a signal representing a time-based certificate to a subscriber mechanism, the certificate specifying a stated reliance limit; and

forwarding to a reliance server electronic signals representing an actual reliance limit for the certificate, the actual reliance limit being different from the stated reliance limit.

5. A transaction system comprising:

a certification authority mechanism issuing a signal representing certificates to subscriber mechanisms to the system; and

a reliance server connectable to the certification authority mechanism and receiving from the certification authority mechanism a signal representing information regarding the certificates issued by the certification authority mechanism, the reliance server issuing, upon request from relying party mechanisms, a signal representing secondary certificates to the relying party mechanisms, the issuing being based on the information provided by the certification authority mechanism and on information provided by the relying party mechanisms.

6. An apparatus for managing reliance in a transaction system in which subscriber mechanisms have certificates issued by certification authority mechanisms, the apparatus comprising:

a mechanism constructed and arranged to receive electronic signals representing a transaction from a subscriber mechanism, the transaction including information regarding at least one certificate of that subscriber mechanism;

a mechanism constructed and arranged to create a message based on certificate information from the transaction, the message specifying an amount of the transaction upon which a relying party mechanism can rely; and

a mechanism constructed and arranged to send electronic signals representing the message to a reliance server requesting a guarantee for the amount of the transaction upon which the relying party mechanism can rely.

7. An apparatus for managing reliance in an electronic transaction system in which subscriber mechanisms have digital time-based certificates issued by certification authority mechanisms, the apparatus comprising, by a reliance server:

a mechanism constructed and adapted to receive electronic signals representing a reliance request message from a party mechanism, the message specifying an amount of a transaction upon which the party mechanism can rely and requesting a guarantee for the amount of the transaction, the message including certificate information derived from the transaction;

a mechanism constructed and adapted to determine whether to provide a guarantee for the amount of the transaction; and

a mechanism constructed and adapted to send electronic signals representing a voucher to the relying party mechanism, the voucher including an indication of whether the reliance server guarantees the amount of the transaction.

8. An apparatus for managing reliance in an electronic transaction system, the apparatus comprising:

a mechanism constructed and arranged to issue electronic signals representing a time-based certificate to a subscriber mechanism, the certificate specifying a stated reliance limit; and

a mechanism constructed and arranged to forward to a reliance server electronic signals representing an actual reliance limit for the certificate, the actual reliance limit being different from the stated reliance limit.

9. Computer-readable media tangibly embodying a program of instructions executable by a computer to perform a method of managing reliance in an electronic transaction system in which subscriber mechanisms have digital certificates issued by certification authority mechanisms, the method comprising, by a relying party mechanism:

receiving electronic signals representing a transaction from a subscriber mechanism, the transaction including information regarding at least one certificate of that subscriber mechanism;

creating a message based on certificate information from the transaction, the message specifying an amount of the transaction upon which the relying party mechanism can rely; and

sending electronic signals representing the message to a reliance server requesting a guarantee for the amount of the transaction upon which the relying party mechanism can rely.

10. Computer-readable media tangibly embodying a program of instructions executable by a computer to perform a method of managing reliance in an electronic transaction system in which subscriber mechanisms have digital time-based certificates issued by certification authority mechanisms, the method comprising, by a reliance server:

receiving electronic signals representing a reliance request message from a party mechanism, the message specifying an amount of a transaction upon which the party mechanism can rely and requesting a guarantee for the amount of the transaction, the message including certificate information derived from the transaction;

determining whether to provide a guarantee for the amount of the transaction; and

sending electronic signals representing a voucher to the relying party mechanism, the voucher including an indication of whether the reliance server guarantees the amount of the transaction.

11. Computer-readable media tangibly embodying a program of instructions executable by a computer to perform a method of managing reliance in an electronic transaction system, the method comprising, by a certification authority mechanism:

issuing electronic signals representing a time-based certificate to a subscriber mechanism, the certificate specifying a stated reliance limit; and

forwarding to a reliance server electronic signals representing an actual reliance limit for the certificate, the actual reliance limit being different from the stated reliance limit.

12. In an electronic transaction system, in which subscriber mechanisms have digital certificates issued by certification authority mechanisms, a memory medium comprising software programmed to provide for reliance management by a method comprising:

by a relying party mechanism:

receiving electronic signals representing a transaction from a subscriber mechanism, the transaction including information regarding at least one certificate of that subscriber mechanism;

creating a message based on certificate information from the transaction, the message specifying an amount of the transaction upon which the relying party mechanism can rely; and

sending electronic signals representing the message to a reliance server requesting a guarantee for the amount of the transaction upon which the relying party mechanism can rely.

13. In an electronic transaction system, a memory medium comprising software programmed to provide for reliance management by a method comprising:

by a certification authority mechanism:

issuing electronic signals representing a time-based certificate to a subscriber mechanism, the certificate specifying a stated reliance limit; and

forwarding to a reliance server electronic signals representing an actual reliance limit for the certificate, the actual reliance limit being different from the stated reliance limit.

14. In an electronic transaction system, in which subscriber mechanisms have digital certificates, a memory medium

comprising software programmed to provide for reliance management by a method comprising, by a reliance server:

receiving electronic signals representing a message from a party mechanism thereby requesting a guarantee for an aspect of the transaction, the message including certificate information derived from the transaction;

validating information in the message to determine whether to provide the guarantee for the aspect of the transaction; and

sending electronic signals representing a reply receipt to the relying party mechanism, the reply receipt including an indication of whether the reliance server guarantees the aspect of the transaction.

15. At least one computer programmed to execute a process for managing reliance in an electronic transaction system in which subscriber mechanisms have digital certificates issued by certification authority mechanisms, the process comprising:

by a relying party mechanism:

receiving electronic signals representing a transaction from a subscriber mechanism, the transaction including information regarding at least one certificate of that subscriber mechanism;

creating a message based on certificate information from the transaction, the message specifying an amount of the transaction upon which the relying party mechanism can rely; and

sending electronic signals representing the message to a reliance server requesting a guarantee for the amount of the transaction upon which the relying party mechanism can rely.

16. At least one computer programmed to execute a process for managing reliance in an electronic transaction system in which subscriber mechanisms have digital time-based certificates issued by certification authority mechanisms, the process comprising, by a reliance server:

receiving electronic signals representing a reliance request message from a party mechanism, the message specifying an amount of a transaction upon which the party mechanism can rely and requesting a guarantee for the amount of the transaction, the

message including certificate information derived from the transaction;

determining whether to provide a guarantee for the amount of the transaction; and

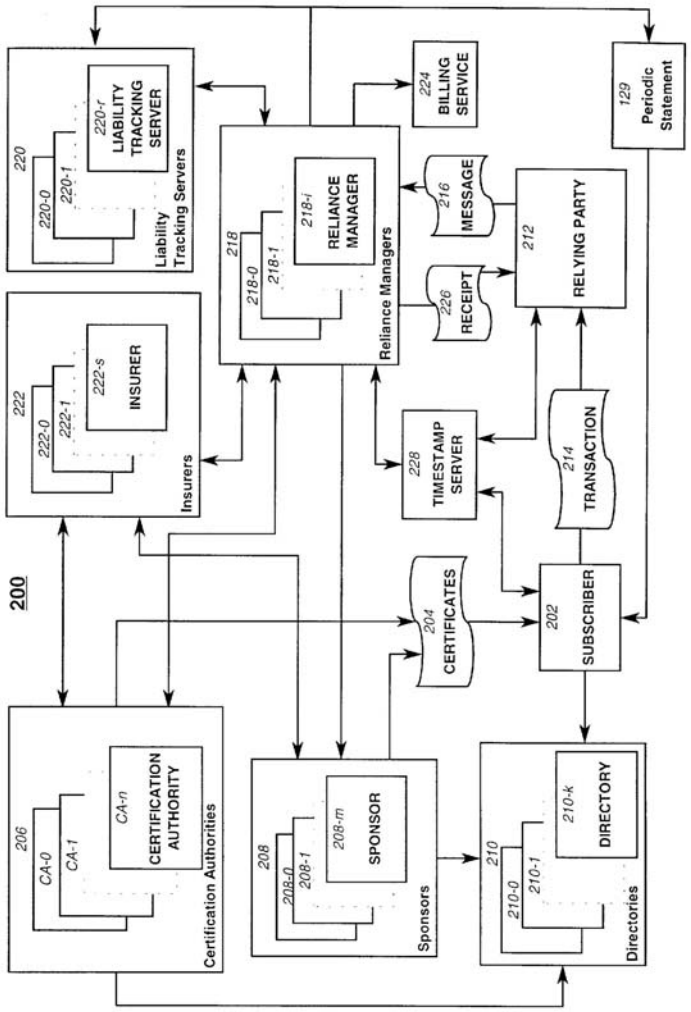
sending electronic signals representing a voucher to the relying party mechanism, the voucher including an indication of whether the reliance server guarantees the amount of the transaction.

17. At least one computer programmed to execute a process for managing reliance in an electronic transaction system, the process comprising, by a certification authority mechanism:

issuing electronic signals representing a time-based certificate to a subscriber mechanism, the certificate specifying a stated reliance limit; and

forwarding to a reliance server electronic signals representing an actual reliance limit for the certificate, the actual reliance limit being different from the stated reliance limit.

FIG. 1


200
220
220-0
220-1
220-r
LIABILITY TRACKING SERVER
Liability Tracking Servers
222
222-0
222-1
222-s
INSURER
Insurers
218
218-0
218-1
218-i
RELIANCE MANAGER
Reliance Managers
224
BILLING SERVICE
129
Periodic Statement
216
MESSAGE
226
RECEIPT
212
RELYING PARTY
228
TIMESTAMP SERVER
214
TRANSACTION
204
CERTIFICATES
202
SUBSCRIBER
206
CA-0
CA-1
CA-n
CERTIFICATION AUTHORITY
Certification Authorities
208
208-0
208-1
208-m
SPONSOR
Sponsors
210
210-0
210-1
210-k
DIRECTORY
Directories

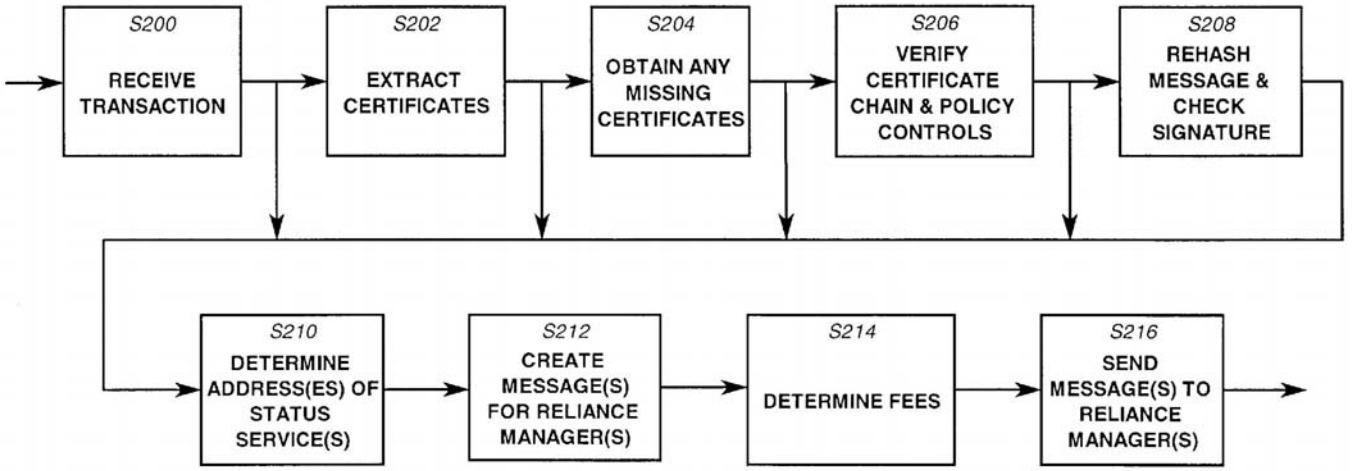
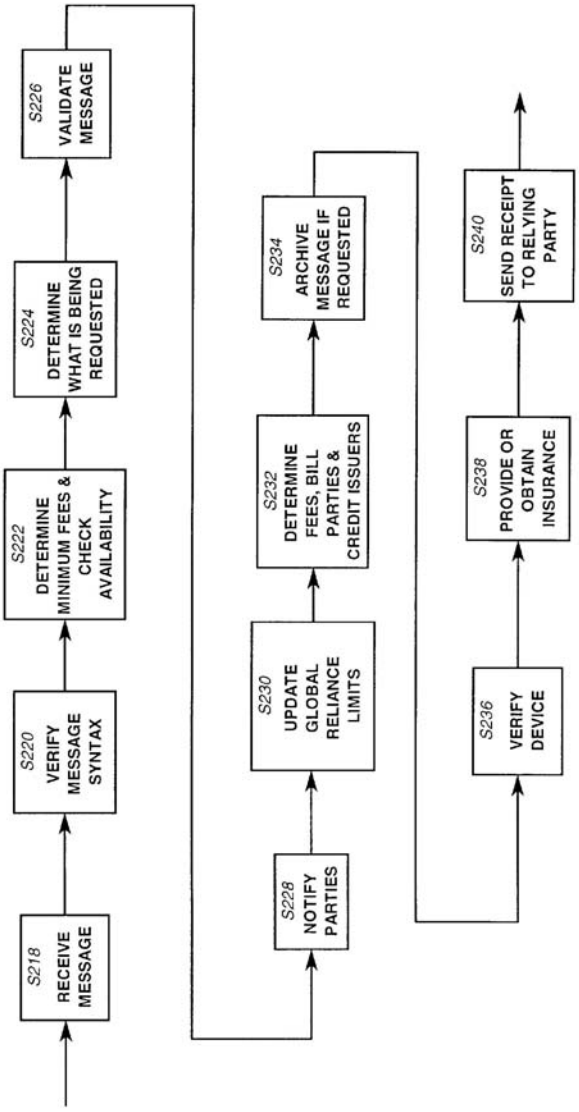

S200
RECEIVE TRANSACTION
S202
EXTRACT CERTIFICATES
S204
OBTAIN ANY MISSING CERTIFICATES
S206
VERIFY CERTIFICATE CHAIN & POLICY CONTROLS
S208
REHASH MESSAGE & CHECK SIGNATURE
S210
DETERMINE ADDRESS(ES) OF STATUS SERVICE(S)
S212
CREATE MESSAGE(S) FOR RELIANCE MANAGER(S)
S214
DETERMINE FEES
S216
SEND MESSAGE(S) TO RELIANCE MANAGER(S)

FIG. 2

FIG. 3


S218
RECEIVE MESSAGE
S220
VERIFY MESSAGE SYNTAX
S222
DETERMINE MINIMUM FEES & CHECK AVAILABILITY
S224
DETERMINE WHAT IS BEING REQUESTED
S226
VALIDATE MESSAGE
S228
NOTIFY PARTIES
S230
UPDATE GLOBAL RELIANCE LIMITS
S232
DETERMINE FEES, BILL PARTIES & CREDIT ISSUERS
S234
ARCHIVE MESSAGE IF REQUESTED
S236
VERIFY DEVICE
S238
PROVIDE OR OBTAIN INSURANCE
S240
SEND RECEIPT TO RELYING PARTY