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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 June 2005 (09.06.2005)

PCT

(10) International Publication Number
WO 2005/052752 A2

(51) International Patent Classification[7]: G06F

(21) International Application Number:
PCT/US2004/039126

(22) International Filing Date:
19 November 2004 (19.11.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/523,398 19 November 2003 (19.11.2003) US

(71) Applicant *(for all designated States except US)*: **CORESTREET, LTD.** [US/US]; One Alewife Center, Suite 200, Cambridge, MA 02140 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **ENGBERG, David** [US/US]; 38 Dover Street, Cambridge, MA 02140 (US).

(74) Agent: **MUIRHEAD, Donald, W.**; Choate, Hall & Stewart, Exchange Place, 53 State Street, Boston, MA 02109-2804 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, 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, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

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

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

WO 2005/052752 A2

(54) Title: DISTRIBUTED DELEGATED PATH DISCOVERY AND VALIDATION

(57) Abstract: Providing path validation information for a system includes determining paths between a subset of certificate of the system and at least one trust root, storing each of the paths in a table prior to a request for path validation information, and fetching the validation information stored in the table in response to a request for path validation information. Providing path validation information may also include digitally signing the validation information. Providing path validation information may also include applying constraints to the validation information and only providing validation information that is consistent with the constraints. Determining paths may include constructing a directed graph of trusted roots and the subset of certificates and performing a depth-first acyclic search of the graph.

DISTRIBUTED DELEGATED PATH DISCOVERY AND VALIDATION

Cross-Reference to Related Applications

This application claims priority to U.S. provisional application 60/523,398 filed on November 19, 2003, which is incorporated by reference herein.

Background of the Invention

1. Technical Field

This application relates to the field of data security and validation, and more particularly to the field of verifying and validating digital certificates and other information.

2. Description of Related Art

It is useful to be able to determine the status of a digital certificate, including determining whether the certificate was validly issued and/or whether the certificate has been revoked prior to the expiration thereof. There are a number of techniques for determining the status of an individual digital certificate. For example, U.S. Patents 5,666,416 and 5,717,758 describe techniques for providing individual certificate status. Other techniques for disseminating and ascertaining certificate status are also known, including Certificate Revocation Lists (CRL's), which are digitally-signed list of revoked certificates.

Verifying a digital certificate may require having to trust the issuer of the digital certificate and/or trust the signer of the revocation information, which may or may not be the same entity. In the case of a digital certificate, "trusting" an issuer and/or a

signer of revocation information may refer to the fact that the issuer and/or signer is a known authority having a valid public key corresponding to a secret key that was used to sign the certificate and/or the revocation information. For example, a user may receive a digital certificate that is digitally signed by an authority, A, and may also receive an up-to-date CRL (that does not contain the digital certificate) signed by a different authority, A'. However, the user would want to be able to trust both A and A' along with their public keys (corresponding to the secret keys used to sign the certificate and the CRL) in order to be able to honor the certificate.

There are mechanisms for facilitating dissemination and trust of otherwise unknown authorities that issue certificates and revocation information. In some instances, it is possible to have a trusted authority digitally sign information (or otherwise validate the information) to verify an otherwise unknown authority. Thereafter, the previously-unknown authority may present digitally-signed information (e.g., a digital certificate and/or revocation information) that may be verified using the public key of the previously-unknown authority. For example, if a user does not know or trust the digital signature of authorities A1 and A2, but the user does know and trust authority A3, then the user may obtain (or be presented with) information digitally signed by A3 (and thus vouched for by A3) indicating that A1 and A2 are trustworthy authorities. Thus, if such a user were presented with a digital certificate signed by A1 and a CRL (that does not list the digital certificate) signed by A2, the user would use the information vouched for by A3 to verify the validity of the presented certificate.

There are also nesting mechanisms that may be used in instances where authorities vouch for other authorities. For example, U.S. patent no. 5,717,759, which is incorporated by reference herein, discloses a technique where a first authority, A1, vouches for a second authority, A2, who vouches for a third authority, A3, etc. until the nesting reaches an authority trusted by a potential user. In some instances, vouching may include providing a digital signature of the vouching authority.

Although nesting and other mechanisms are useful, in some cases a user may be presented with a digital certificate and/or revocation information and/or some other information for which there is no straight-forward mechanism for determining whether the signer of the certificate/revocation information/other information may be trusted and thus the user may not be able to determine whether the digital certificate is currently valid. Accordingly, it would be useful to address this.

Summary of the Invention

According to the present invention, providing path validation information for a system includes determining paths between a subset of certificates of the system and at least one trust root, storing each of the paths in a table prior to a request for path validation information, and fetching the validation information stored in the table in response to a request for path validation information. Providing path validation information may also include digitally signing the validation information. Providing path validation information may also include applying constraints to the validation information and only providing validation information that is consistent with the constraints. Determining paths may include constructing a directed graph of trusted

roots and the subset of certificates and performing a depth-first acyclic search of the graph. The table may be indexed using the trusted roots or using the certificates. Providing path validation information may also include receiving proofs of revocation status for the subset of certificates, storing the proofs prior to a request for path validation information, and fetching the proofs along with the validation information in response to a request for path validation information. Providing path validation information may also include digitally signing the validation information and the proofs. Providing path validation may also include applying constraints to the validation information and only providing validation information that is consistent with the constraints. Determining paths may include constructing a directed graph of trusted roots and the subset of certificates and performing a depth-first acyclic search of the graph. The proofs may be stored in the table that contains the validation information. The table may be indexed using the trusted roots. The table may be indexed using the certificates. The proofs may be stored in an other table that is separate from the table that contains the validation information. The other table may be indexed using the trusted roots or using the certificates. The subset of certificates may include trusted roots, authorities that issue end user certificates, and authorities that vouch for other authorities. The subset of certificates may also include end user certificates.

According further to the present invention, a computer program product that provides path validation information for a system includes a storage medium that contains executable code for the computer program product, executable code that determines paths between a subset of certificates of the system and at least one trust root, executable code that stores each of the paths in a table prior to a request for path

validation information, and executable code that fetches the validation information stored in the table in response to a request for path validation information. The computer program product may also include executable code that digitally signs the validation information. The computer program product may also include executable code that applies constraints to the validation information and that only provides validation information that is consistent with the constraints. Executable code that determines paths may construct a directed graph of trusted roots and the subset of certificates and may perform a depth-first acyclic search of the graph. The table may be indexed using the trusted roots. The table may be indexed using the certificates. The computer program product may also include executable code that receives proofs of revocation status for the subset of certificates, executable code that stores the proofs prior to a request for path validation information, and executable code that fetches the proofs along with the validation information in response to a request for path validation information. The computer program product may also include executable code that digitally signs the validation information and the proofs. The computer program product may also include executable code that applies constraints to the validation information and only provides validation information that is consistent with the constraints. Executable code that determines paths may construct a directed graph of trusted roots and the subset of certificates and may perform a depth-first acyclic search of the graph. The proofs may be stored in the table that contains the validation information. The table may be indexed using the trusted roots or using the certificates. The proofs may be stored in an other table that is separate from the table that contains the validation information. The other table may be indexed using the trusted roots or using the certificates. The subset of certificates may include trusted roots, authorities

that issue end user certificates, and authorities that vouch for other authorities. The subset of certificates may also include end user certificates.

According further to the present invention, a server includes a processor, internal storage coupled to the processor, executable code, provided on the internal storage, that determines paths between a subset of certificates and at least one trust root, executable code, provided on the internal storage, that stores each of the paths in a table prior to a request for path validation information, and executable code, provided on the internal storage, that fetches the validation information stored in the table in response to a request for path validation information. The server may include executable code, provided on the internal storage, that digitally signs the validation information. The server may include executable code, provided on the internal storage that applies constraints to the validation information and that only provides validation information that is consistent with the constraints. Executable code that determines paths may construct a directed graph of trusted roots and the subset of certificates and performs a depth-first acyclic search of the graph. The table may be indexed using the trusted roots or using the certificates. The server may include executable code, provided on the internal storage, that receives proofs of revocation status for the subset of certificates, executable code, provided on the internal storage, that stores the proofs prior to a request for path validation information, and executable code, provided on the internal storage, that fetches the proofs along with the validation information in response to a request for path validation information. The server may include executable code, provided on the internal storage, that digitally signs the validation information and the proofs. The server may include executable code, provided on the internal storage, that

applies constraints to the validation information and only provides validation information that is consistent with the constraints. Executable code that determines paths may construct a directed graph of trusted roots and the subset of certificates and may perform a depth-first acyclic search of the graph. The proofs may be stored in the table that contains the validation information. The table may be indexed using the trusted roots or using the certificates. The proofs may be stored in an other table that is separate from the table that contains the validation information. The other table may be indexed using the trusted roots or using the certificates. The subset of certificates may include trusted roots, authorities that issue end user certificates, and authorities that vouch for other authorities. The subset of certificates may also include end user certificates.

Brief Description of Drawings

Figure 1 illustrates a untrusted delegated path/validation server according to an embodiment of the system described herein.

Figure 2 illustrates a trusted delegated path/validation server according to an embodiment of the system described herein.

Figure 3 illustrates two interconnected areas, each containing local certificate information according to an embodiment of the system described herein.

Figure 4 illustrates using a network to communicate certificate information to a user according to an embodiment of the system described herein.

Figure 5 is a flow chart illustrating pre-calculating trust paths according to an embodiment of the system described herein.

Figure 6 is a flow chart illustrating iterating through trust paths in connection with populating a table of certificates and trust paths according to the system described herein.

Figure 7A, 7B, and 7C illustrate tables containing trust paths and/or proofs according to an embodiment of the system described herein.

Figure 8 is a flow chart illustrating returning pre-calculated trust paths and/or proofs according to an embodiment of the system described herein.

Figure 9 is a flow chart illustrating processing performed by a trusted delegated path/validation server according to an embodiment of the system described herein.

Detailed Description of Various Embodiments

Techniques for providing individual certificate status information are known as are techniques for verifying certificates and/or authorities that digitally sign certificates. See, for example, the disclosure provided in U.S. Patents 5,420,927; 5,604,804; 5,610,982; 6,097,811; 6,301,659; 5,793,868; 5,717,758; 5,717,757; 6,487,658; and 5,717,759, all of which are incorporated by reference herein. The system described herein may use techniques disclosed in one or more of these patents, possibly in combination with one or more other appropriate techniques. Techniques that may be used include, alone or in any combination, full CRL's, partitioned CRL's, delta CRL's,

OCSP responses (individually and in groups), mini CRL's (bitwise compressed CRL's), VTokens (one-way hash chain), and various Merkle tree or other tree incarnations.

Path discovery relates to the process of finding a trust path (trust relationship) between an arbitrary certificate and one of a user's trust roots, which are digital certificates corresponding to authorities trusted by a user. A trust path is a chain of authorizations from one certificate to another having a target certificate at one end and a trust root at the other. For example, if certificate C1 corresponds to a trusted authority A1, and A1 signs C2 for unknown authority A2 and authority A2 signs certificate C3 for unknown authority A3 that signed the target certificate, then a trust path from C1 (the trust root) to the target certificate is C1 to C2, C2 to C3, and C3 to the target certificate.

In some cases, path discovery may be performed locally in a fairly straight-forward manner by the user whenever all of the certificates in a trust path are locally available. However, in some cases, including those with a federated trust model based on cross-certification, users may not have all of the information needed to perform local path discovery.

Note also that, even when a complete trust path has been found between a trust root and a target certificate, there still may be a concern that one or more of the certificates on the trust path may have been revoked since the issuance thereof. Path validation refers to confirming the current status of all of the certificates in a trust path,

including the validity thereof (revocation status) of the certificates as well as taking into account any and all rules and/or constraints corresponding to use of the certificates (e.g., checking security policies). Assembling the revocation status information for each certificate in the trust path may be relatively difficult for some users in some situations (e.g. those in special firewall settings preventing access to public networks). Accordingly, it is useful to be able to provide a service that can verify a certificate in a single transaction. This may be done by first determining the trust path, receiving revocation information (and/or constraint information) about the certificates in the trust path, and processing the received information to verify the target certificate.

Referring to Figure 1, an untrusted delegated path discovery/validation (UDPDV) server 30 receives, as input, information identifying one or more trust roots (certificates for trusted authorities) and an identifier corresponding to a target certificate (or perhaps some other information/data) that a user desires to validate. Optionally, the UDPDV server 30 may be pre-configured with trust roots in addition to, or instead of, receiving trust roots as input.

Of course, if the user already trusts an authority that issued the target certificate and the user has access to reliable revocation information for the target certificate (and possibly the trusted authority's certificate), then the user can verify the target certificate with that information and it may not be necessary to use the UDPDV server 30. Accordingly, the UDPDV server 30 is useful in instances where a user wants to verify the status of a target certificate for which the user does not trust (know) the authority that issued the target certificate and/or for which the user does not posses reliable

revocation information about the target certificate (and possibly the certificate of the authority that issued the target certificate).

The UDPDV server 30 processes the inputs thereto to determine one or more trust paths between the target certificate and a trust root (either input or preconfigured into the UDPDV server 30, as described above). Note that, in many instances, it may also be useful to confirm that none of the certificates in a trust path have been revoked (indicating, for example, compromise of an authority's secret key). In such a case, it may be useful to have the UDPDV server 30 also provide proofs in the form of authenticated revocation information (e.g., CRL's or OCSP responses). Accordingly, in some embodiments, the UDPDV server 30 may also provide proofs for certificates in a trust path.

Thus, in the embodiment disclosed in Figure 1, the UDPDV server 30 assembles the trust path information for the target certificate and, optionally, also assembles revocation information for path elements. The output of the UDPDV server 30 to the user may not be independently authenticated (e.g. signed). Instead, the UDPDV server 30 returns all of the individual elements of the trust path (and, optionally, proofs) to the user, who may then independently verify the correctness of the trust path and the validity of the elements (certificates) thereof. Operation of the UDPDV server 30 is described in more detail elsewhere herein.

Referring to Figure 2, a trusted delegated path discovery/validation server 40 receives, as input, a user's trust roots and an identifier corresponding to a target

certificate (or perhaps some other information/data) that a user desires to verify. Optionally, the trusted delegated path discovery/validation server 40 may be pre-configured with trust roots in addition to, or instead of, receiving the trust roots as input.

The trusted delegated path discovery/validation server 40 outputs a result indicating whether or not the target certificate (or other information/data) is valid. In an embodiment described herein, the server 40 performs its own cryptographic authentication of the information that is provided to the user. In this embodiment, the server 40 may sign responses provided thereby, which would allow the server 40 to provide a small volume of data to the user (e.g. "Yes, you can trust certificate X"). However, such an implementation provides that the user explicitly trust the integrity and correctness of the internal operations of the trusted delegated path discovery/validation server 40. If the server 40 is compromised, the server 40 could be improperly used to authenticate responses that validate any certificate, regardless of source or current status.

The servers 30, 40 may be provided by computer workstations having processors and internal storage for storing executable code and data, one or more software modules provided in a general purpose computer, dedicated hardware, or any combination of hardware and software capable of providing the functionality described herein. For the system described herein, the UDPDV server 30 may be provided by one or more lightweight servers that do not need to have authentication capabilities (e.g. private keys for digital signatures).

The UDPDV server 30 may be periodically configured with lists of information of two types. The first type of list contains a set of certificates which may be used in path validation and, in an embodiment herein, represents all or nearly all certificates accessible (e.g., via a network) to the UDPDV server 30. In another embodiment, the first type of list contains certificates only for authorities that issue certificates and revocation information and for authorities that vouch for those authorities, without necessarily contain all or even any end user certificates. The first type of lists may contain self-signed root certificates (trusted roots) which serve as trust anchors. The trusted roots may be signed by authorities trusted by the user(s) of the system. The first type of lists may also contain issuer (aka "certificate authority") certificates for authorities that issue certificates, authorities that issue revocation information, and/or authorities that vouch for other authorities. In one embodiment, the first type of lists may also contain end user certificates while in another embodiment, the first type of lists does not contain end user certificates. The first type of lists of certificates may be used to provide path discovery services, as described herein.

The second type of list provided in the UDPDV server 30 may include pre-generated certificate status proofs. Each proof may contain the status of one or more certificates (from the first list) for a fixed interval of time, and the proof may be securely authenticated, e.g. using a digital signature. The proofs may be used to provide path validation services, as described elsewhere herein. The proofs may be provided by any appropriate means, including CRL's, OCSP responses, VTokens, etc.

Checking a path between a trusted root and a target certificate is known in the art and well documented in certificate standards (e.g., RFC 3280). However, finding a trust path through a large number of certificates may take exponential time based on the number of certificates in the pool. This may be acceptable in some situations and/or for small collections of certificates, but may be unacceptably in other situations, such as a large community of federated authorities.

The system described herein is designed to perform path discovery in logarithmic or constant time at the time of a discovery request by first pre-calculating trust paths between each trusted root certificate and all other reachable certificates for authorities that issue certificates and/or vouch for other authorities. When the UDPDV server 30 receives a new list of certificates (or is coupled to a source of new certificates), the server may pre-compute an M by N matrix where M is the number of trusted roots and N is the total number of certificates. Each cell in the matrix (e.g., at row r1 and column c1) may contain one or more legitimate paths from the specific trusted root indicated by the row r1 to the specific certificate indicated by the column c1, or else contain an empty set to indicate that no paths are possible. Optionally, each cell (or each cell of an analogous table) may also contain appropriate proofs to provide validation.

When a user request for verification of a target certificate arrives, the UDPDV server 30 uses the pre-computed matrix to look up the trusted root(s) for the user and to look up the authority that issued the target certificate and, optionally, to an authority that issued revocation information for the target certificate (if different from the issuing

authority) to find one or more valid paths therebetween. This may be performed in a constant time look-up since the paths have been pre-calculated and stored in the UDPDV server 30. Note that if there is more than one possible trust path or if there are constraints associated with a trust path (e.g. name or policy constraints) that limit in any way the usage of any of the certificates in the trust path, then the path policies need to be applied against any trust path(s) found. By pre-computing all possible paths, the UDPDV server 30 may offer path discovery for large communities with relatively high performance and scalability.

In addition to path discovery, the UDPDV server 30 may also provide validation information (proofs) for elements (certificates) of a trust path. In one embodiment, the UDPDV server 30 obtains the proofs (e.g. in the form of a CRL or OCSP response) in real time for each element (certificate) in a trust path which is then provided to the user. Although it may be possible to improve performance by caching proofs, there still may be the potential for non-optimal performance at the time of a request when proofs are obtained in real time. In another embodiment, there may be a path validation mechanism in the UDPDV server 30 that receives at regular intervals pre-generated, fine-grained status proofs for each certificate used by the UDPDV server 30 in connection with generating trust paths. For this embodiment, the UDPDV server 30 may be able to quickly access all needed validation information that may be provided to relying parties without requiring any extra processing time to retrieve the certificate status information for validation at real time.

Note also that, if the proofs are pushed to the UDPDV server 30 (e.g. in the form of pre-signed OCSP responses), the UDPDV server 30 may have instant local access to the status of every certificate in a trust path. For some embodiments, the UDPDV server 30 confirms the status of the trust path before returning the trust path for path delegation. The UDPDV server 30 may also optionally return the status proofs to the user to permit local complete validation of the trust path by the user. The individualized, pre-generated nature of the proofs (e.g. pre-generated OCSP responses) may allow an efficient use of networking resources while avoiding any security risks that may be associated with trusted online servers.

Referring to Figure 3, a diagram 50 shows a first area 52 and a separate second area 54. The areas 52, 54 may be interconnected by any appropriate means (e.g., a network or direct connection) to enable an exchange of signals therebetween. The first area 52 includes a plurality of certificates 62-64. The second area 54 includes a plurality of different certificates 66-68. The first area 52 represents a locality that may be managed and accessed locally by a user therein. Similarly, the second area 54 represents a separate locality that may be managed and accessed locally by a different user therein. Thus, for example, a user in the area 52 may locally access any or all of the certificates 62-64.

At the area 52, the certificate 62 is a self-certifying root certificate (trusted root) for an authority A1. The certificate 62 also certifies the certificate 63 for an authority A2 by, for example, having A1 sign the certificate 63. The certificate 63 certifies the certificate 64 for an authority A3. Note that, in this example, if a user trusts A1, then

the user should also trust A2 (vouched for by A1) and should also trust A3 (vouched for by A2). At the area 54, the certificate 66 is a self-certifying root certificate (trusted root) for an authority A1'. The certificate 66 also certifies the certificate 67 for an authority A2' and the certificate 67 certifies the certificate 68 for an authority A3'. The certificate 62 is cross-certified with the certificate 66 so that each of the certificates 62, 66 certifies the other one of the certificates 62, 66.

If a user at the area 52 is presented with a target certificate signed by A3', the user may not be able to immediately verify that the target certificate is valid if the user at the area 52 only initially knows about (and trusts) A1, A2, and A3. However, note that the authority A2' vouches for the authority A3' and that the authority A1' vouches for the authority A2'. Note also that the authority A1 vouches for the authority A1'. Thus, assuming a local user at the area 52 trusts A1, then there is a trust path for the user from the certificate 62 to the certificate 66 to the certificate 67 to the certificate 68 to the target certificate that has been signed by A3'. Thus, the user may accept the target certificate signed by the authority A3' because of the trust path from the target certificate to the certificate 62 that has been signed by the trusted authority A1. Also, as discussed elsewhere herein, it is possible to provide validation information (proof of validity) for each of the certificates along the trust path as well as for any authority that issued revocation information (if different from the issuing authority) so that, in the example the user that receives the trust path may also receive up-to-date revocation information for the certificates 62, 66-68.

Referring to Figure 4, a diagram 80 illustrates a user 82 receiving certificate information (including trust path information and possibly validation information) from one or more of a plurality of data storage devices 84-86 coupled to the user 82 by a network 88. In an embodiment herein, the network 88 may be the Internet, although other suitable networks may be used, including networks that provide direct connections between the user and one or more of the data storage devices 84-86. Note also that the user 82 may locally store some certificate information. In an embodiment herein, the user 82 may be presented with one or more certificates that the user desires to determine the validity thereof. In some cases, the user 82 may determine the validity of the certificates using local data. However, as described elsewhere herein, it may be necessary in other cases for the user 82 to obtain certificate information from other sources like the data storage devices 84-86. In such cases, certificate information and requests therefor may be communicated between the user and the storage devices 84-86 over the network 88 in a straight-forward manner. Of course, any appropriate connectivity and information request/transmission techniques may be used to provided the functionality described herein.

Referring to Figure 5, a flow chart 100 illustrates initializing the UDPDV server 30 to contain all of the paths between the trusted roots (root certificates for trusted authorities) and one or more other certificates that issue certificates and/or vouch for other authorities. As discussed elsewhere herein, each of the trust paths may be pre-computed so that when a user presents a target certificate, the UDPDV server may consult a table, look up the authority that issued the target certificate (or look up the target certificate itself), and provided a pre-computed trust path from the target

certificate to a trusted root certificate. As also discussed herein, the UDPDV server 30 may optionally provide, for the certificates in the trust path, proofs indicating that the certificates have not been revoked.

Processing for the flow chart 100 begins at a first step 102 where a directed graph is constructed for all of the certificates for which information is to be stored. A directed graph is a mathematical construct that is well-known in the art. For an embodiment herein, all of the certificates in a system, including end user certificates, are used. For another embodiment, end user certificates are not included or, alternatively, only some end user certificates are included. At the step 102, the directed graph that is constructed represents a trust relationship between certificates where an edge (connecting line) of the graph indicates that a first authority (corresponding to a certificate connected at one end of the edge) has vouched for a second authority (corresponding to a certificate connected at the other end of the edge).

Following the step 102 is a step 104 where an index variable, I, is set equal to one. The index variable, I, is used to iterate through each of the trusted root certificates for the UDPDV server 30. As discussed elsewhere herein, the trusted root certificates are provided as an input to or are pre-configured in the UDPDV server 30.

Following the step 104 is a test step 106 where it is determined if the index variable, I, is greater than the number of trusted root certificates. If so, then processing is complete. Otherwise, control transfers from the test step 106 to a step 108 to determine all of the paths from the trusted root certificate (corresponding to the index

variable I) to all of the other certificates in the directed graph. The step 108 is discussed in more detail elsewhere herein. Following the step 108 is a step 112 wherein the index variable, I, is incremented. Following the step 112, control transfers back to the test step 106, discussed above.

Referring to Figure 6, a flow chart 120 illustrates in more detail processing performed in connection with the step 108 of the flow chart 100 of Figure 5. For each of the trusted roots, a depth-first acyclic search of the directed graph is performed to find all of the trust paths. For each certificate that is found in each path, an entry is made in a table indicating the trusted path from the certificate to the trusted root.

Processing begins at a first step 122, where it is determined if there are more paths to be examined. Note that, in some instances, it may be possible to have a trusted root with no paths thereto. However, in most cases, it is expected that each of the trusted roots will have at least one path thereto. If it is determined at the test step 122 that there are no more paths to be examined (processed), then processing is complete. Otherwise, control transfers from the test step 122 to a step 124 wherein the next path is determined using the depth-first acyclic search of the directed graph. Following the step 124 is a step 126 where a certificate pointer, CP, is set to point to the end of the path being examined (processed).

Following the step 126 is a test step 128 which determines if CP points to the trusted root, thus indicating that the entire path has been traversed. If so, then control transfers from the test step 128 back to the step 122, discussed above, to begin the next

iteration. Otherwise, control transfers from the test step 128 to a step 132 where the path from CP to the trusted root is recorded in the table (discussed elsewhere herein) that stores certificates and trusted paths thereto. In one embodiment, at a portion of the table indexed by the certificate pointed to by CP, the path from CP to the trusted root is recorded at the step 132. Following the step 132 is a step 134 where CP is set to point to the previous certificate in the path. Thus, CP initially points to the end of the path and then subsequently points to previous certificates in the path working backwards toward the trusted root. Following the step 134, control transfers back to the test step 128, discussed above.

Referring to Figure 7A, a table 140 includes a first portion 142 indexed by certificates in the system and having, as elements, trusted paths to each of the certificates. The table 140 may also include a second portion 144 that is also indexed by certificates in the system. The second portion has elements that describe proofs (e.g., OCSP responses, CRL's, etc.) indicating the revocation status for each of the corresponding certificates. In an embodiment described herein, the proofs provided in the second portion 144 correspond to the index certificate so that, for example, the table entry indexed by C2 corresponds to a proof for C2. In another embodiment, the proofs provided in the second portion 144 correspond to all of the certificates in all of the paths for a particular entry in the first portion 142. Thus, for example, the proofs provided in the second portion 144 in connection with certificate C2 correspond to all of the certificates in the trust paths provided in the entry for C2 in the first portion 142.

In some embodiments, it may be possible to eliminate one of the portions 142, 144. Thus, for example, in embodiments with only the portion 142, the UDPDV server 30 provides pre-calculated trust paths, but not proofs. For such embodiments, users may either forgo proofs altogether, obtain proofs in real time, or some combination thereof (i.e., obtain proofs for some certificates in the paths but not others). For embodiments with only the portion 144, trust paths may be determined in real time and the pre-calculated proofs from the portion 144 may be provided by the UDPDV server 30.

Referring to Figure 7B, another embodiment uses a single table 140' indexed according to certificates, where both the trust paths and the proofs are provided as elements of the table 140'.

Referring to Figure 7C, yet another embodiment illustrates another configuration for the table 140''. The table 140'' may be indexed according to a particular trusted root certificate TR1, TR2, ...TRN and other certificates C1, C2, ...CN. The elements at an entry of the table 140'' contain trust paths and (optionally) proofs "P/P" so that, for example, an entry for trusted root TRx and certificate Cy contains one or more trust paths (if any exist) from Cy to TRx and, optionally, one or more proofs for just Cy (in one embodiment) or for all of the certificates in the trust path(s) (in another embodiment).

Referring to Figure 8, a flow chart 150 illustrate steps for processing in connection with the UDPDV server 30 servicing a request from a user for a trust path

and, optionally, proofs for certificates thereof. As discussed elsewhere herein, the UDPDV server 30 may be presented with a particular target certificate for which the UDPDV server 30 provides a trust path and optionally proofs indicating the status out each certificate in the trust path. In one embodiment, the trust path is provided to a certificate for an authority that issued the target certificate and, optionally, to a certificate for an authority that issued revocation information for the target certificate (if different from the issuing authority). In another embodiment, the UDPDV server 30 provides a trust path and optional revocation information for the target certificate itself.

Processing begins at a first step 152 where it is determined if there are any trust paths available at all for the target certificate or the issuer thereof. If not, then control transfers from the step 152 to a step 154 where special processing is performed. The special processing performed at the step 154 may include posting an error message and/or indicating to a user in some other way that no trust path has been found for the target certificate. Following the step 154, processing is complete.

If it is determined at the test step 152 that there are trust paths available, then control transfers from the test step 152 to a step 156 where the next trust path is chosen for processing. In an embodiment herein, the system may iterate through each of the trust paths (provided by the table 140, the table 140', or the table 140'') to find an appropriate trust path between the target certificate (or issuer thereof) and one or more of the trusted root certificates. The next trust path chosen at the step 156 may be chosen using any of a number of possible criteria, such as shortest path, path containing particular certificates, etc.

Following step 156 is a test step 158 which determines if there are constraints on the particular trust path (certificates of the trust path) chosen at the step 156. As discussed elsewhere herein, there may be one or more constraints that prevent use of a particular trust path such as, for example, one or more certificates not being acceptable for certain purposes. If it is determined at the test step 158 that there are constraints that make the trust path being examined unacceptable, then control transfers from the test step 158 back to the step 152, discussed above. Otherwise, control transfers from the test step 158 to a step 162 where it is determined if proofs for the certificates in the trust path have been requested. If not, then control transfers from the step 162 to a step 164 where the trust path between the target certificate (or issuer thereof) and the trusted root certificate is returned to the user. Following step 164, processing is complete.

If it is determined at the test step 162 that proofs have been requested, then control transfers from the test step 162 to a step 166 where the proofs (from the table 140, the table 140', or the table 140'') are obtained. In other embodiments, the proofs may be obtained in real time. Following the step 166 is a step 168 where the trust path and the proofs are returned. Following step 168, processing is complete.

As discussed elsewhere herein, in some instances, a user may desire to use the trusted distributed delegated path discovery and validation server 40 that returns a signed (or otherwise authenticated) message indicating whether or not a particular certificate is valid. The server 40 may sign a message indicating that a particular certificate is acceptable or not. The user may rely on the signed response from the

server 40 without necessarily knowing or examining the path and/or the validation information first hand.

Referring to Figure 9, a flow chart 180 illustrates steps performed by the server 40 in connection with constructing and returning a signed message indicating whether or not a particular target certificate is valid. Processing begins at a first step 182 where the trusted path and proofs are obtained in accordance with the description elsewhere herein. Following the step 182 is a test step 184 wherein the trust path and the proofs are examined to determine if the target certificate is valid. The determination at the step 184 may be made using the trust path and proofs obtained at the step 182. If it is determined at the test step 184 that the target certificate is valid, then control transfers from the test step 184 to a step 186 where a positive message, indicating that the target certificate is valid, is created. Otherwise, if it is determined at the test step 184 that the target certificate is not valid, then control transfers from the test step 184 to a step 188 where a negative message is created.

Following either the step 186 or the step 188 is a step 192 where the message is digitally signed by the server 40. Following the step 192 is a step 184 where the signed result is returned to the user. Following step 194, processing is complete.

The system described herein may be implemented using any appropriate combination of hardware and/or software, including software provided in a storage medium (e.g., disk, tape, CD ROM, removable memory, etc.). The software may run

on specialized hardware, on a general purpose computer, or an appropriate combination thereof.

While the invention has been disclosed in connection with various embodiments, modifications thereon will be readily apparent to those skilled in the art. Accordingly, the spirit and scope of the invention is set forth in the following claims.

What is claimed is:

1. A method of providing path validation information for a system, comprising:

determining paths between a subset of certificates of the system and at least one trust root;

storing each of the paths in a table prior to a request for path validation information; and

fetching the validation information stored in the table in response to a request for path validation information.

2. A method, according to claim 1, further comprising:

digitally signing the validation information.

3. A method, according to claim 1, further comprising:

applying constraints to the validation information and only providing validation information that is consistent with the constraints.

4. A method, according to claim 1, wherein determining paths includes constructing a directed graph of trusted roots and the subset of certificates and performing a depth-first acyclic search of the graph.

5. A method, according to claim 1, wherein the table is indexed using the trusted roots.

6. A method, according to claim 1, wherein the table is indexed using the certificates.

7. A method, according to claim 1, further comprising:

receiving proofs of revocation status for the subset of certificates;

storing the proofs prior to a request for path validation information; and

fetching the proofs along with the validation information in response to a request for path validation information.

8. A method, according to claim 7, further comprising:

digitally signing the validation information and the proofs.

9. A method, according to claim 7, further comprising:

applying constraints to the validation information and only providing validation information that is consistent with the constraints.

10. A method, according to claim 7, wherein determining paths includes constructing a directed graph of trusted roots and the subset of certificates and performing a depth-first acyclic search of the graph.

11. A method, according to claim 7, wherein the proofs are stored in the table that contains the validation information.

12. A method, according to claim 11, wherein the table is indexed using the trusted roots.

13. A method, according to claim 11, wherein the table is indexed using the certificates.

14. A method, according to claim 7, wherein the proofs are stored in an other table that is separate from the table that contains the validation information.

15. A method, according to claim 14, wherein the other table is indexed using the trusted roots.

16. A method, according to claim 14, wherein the other table is indexed using the certificates.

17. A method, according to claim 1, wherein the subset of certificates includes trusted roots, authorities that issue end user certificates, and authorities that vouch for other authorities.

18. A method, according to claim 17, wherein the subset of certificates further includes end user certificates.

19. A computer program product that provides path validation information for a system, comprising:

a storage medium that contains executable code for the computer program product;

executable code that determines paths between a subset of certificates of the system and at least one trust root;

executable code that stores each of the paths in a table prior to a request for path validation information; and

executable code that fetches the validation information stored in the table in response to a request for path validation information.

20. A computer program product, according to claim 19, further comprising:

executable code that digitally signs the validation information.

21. A computer program product, according to claim 19, further comprising:

executable code that applies constraints to the validation information and that only provides validation information that is consistent with the constraints.

22. A computer program product, according to claim 19, wherein executable code that determines paths constructs a directed graph of trusted roots and the subset of certificates and performs a depth-first acyclic search of the graph.

23. A computer program product, according to claim 19, wherein the table is indexed using the trusted roots.

24. A computer program product, according to claim 19, wherein the table is indexed using the certificates.

25. A computer program product, according to claim 19, further comprising:

executable code that receives proofs of revocation status for the subset of certificates;

executable code that stores the proofs prior to a request for path validation information; and

executable code that fetches the proofs along with the validation information in response to a request for path validation information.

26. A computer program product, according to claim 25, further comprising:

executable code that digitally signs the validation information and the proofs.

27. A computer program product, according to claim 25, further comprising:

executable code that applies constraints to the validation information and only provides validation information that is consistent with the constraints.

28. A computer program product, according to claim 25, wherein executable code that determines paths constructs a directed graph of trusted roots and certificates in the system and performs a depth-first acyclic search of the graph.

29. A computer program product, according to claim 25, wherein the proofs are stored in the table that contains the validation information.

30. A computer program product, according to claim 29, wherein the table is indexed using the trusted roots.

31. A computer program product, according to claim 29, wherein the table is indexed using the certificates.

32. A computer program product, according to claim 25, wherein the proofs are stored in an other table that is separate from the table that contains the validation information.

33. A computer program product, according to claim 32, wherein the other table is indexed using the trusted roots.

34. A computer program product, according to claim 32, wherein the other table is indexed using the certificates.

35. A computer program product, according to claim 19, wherein the subset of certificates includes trusted roots, authorities that issue end user certificates, and authorities that vouch for other authorities.

36. A computer program product, according to claim 35, wherein the subset of certificates further includes end user certificates.

37. A server, comprising:

a processor;

internal storage coupled to the processor;

executable code, provided on the internal storage, that determines paths between a subset of certificates of the system and at least one trust root;

executable code, provided on the internal storage, that stores each of the paths in a table prior to a request for path validation information; and

executable code, provided on the internal storage, that fetches the validation information stored in the table in response to a request for path validation information.

38. A server, according to claim 37, further comprising:

executable code, provided on the internal storage, that digitally signs the validation information.

39. A server, according to claim 37, further comprising:

executable code, provided on the internal storage that applies constraints to the validation information and that only provides validation information that is consistent with the constraints.

40. A server, according to claim 37, wherein executable code that determines paths constructs a directed graph of trusted roots and the subset of certificates and performs a depth-first acyclic search of the graph.

41. A server, according to claim 37, wherein the table is indexed using the trusted roots.

42. A server, according to claim 37, wherein the table is indexed using the certificates.

43. A server, according to claim 37, further comprising:

executable code, provided on the internal storage, that receives proofs of revocation status for the subset of certificates;

executable code, provided on the internal storage, that stores the proofs prior to a request for path validation information; and

executable code, provided on the internal storage, that fetches the proofs along with the validation information in response to a request for path validation information.

44. A server, according to claim 43, further comprising:

executable code, provided on the internal storage, that digitally signs the validation information and the proofs.

45. A server, according to claim 43, further comprising:

executable code, provided on the internal storage, that applies constraints to the validation information and only provides validation information that is consistent with the constraints.

46. A server, according to claim 43, wherein executable code that determines paths constructs a directed graph of trusted roots and the subset of certificates and performs a depth-first acyclic search of the graph.

47. A server, according to claim 43, wherein the proofs are stored in the table that contains the validation information.

48. A server, according to claim 47, wherein the table is indexed using the trusted roots.

49. A server, according to claim 47, wherein the table is indexed using the certificates.

50. A server, according to claim 43, wherein the proofs are stored in an other table that is separate from the table that contains the validation information.

51. A server, according to claim 50, wherein the other table is indexed using the trusted roots.

52. A server, according to claim 50, wherein the other table is indexed using the certificates.

53. A server, according to claim 37, wherein the subset of certificates includes trusted roots, authorities that issue end user certificates, and authorities that vouch for other authorities.

54. A server, according to claim 53, wherein the subset of certificates further includes end user certificates.

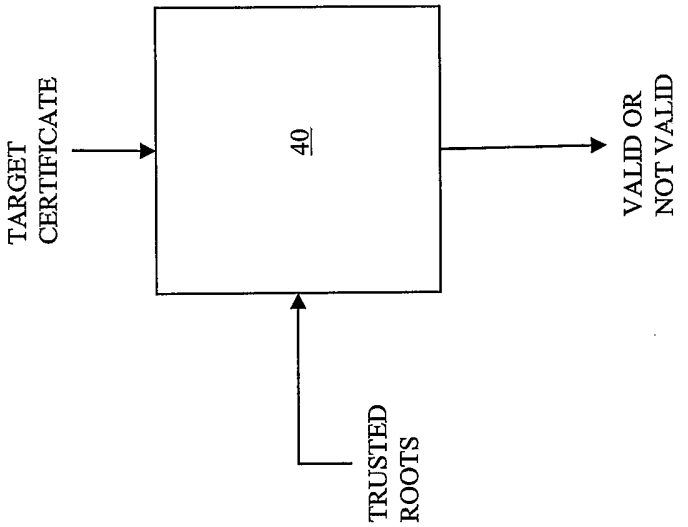


FIGURE 2

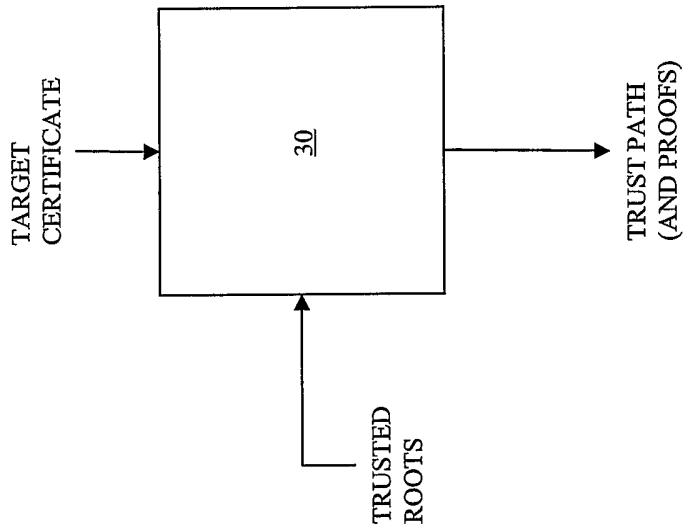


FIGURE 1

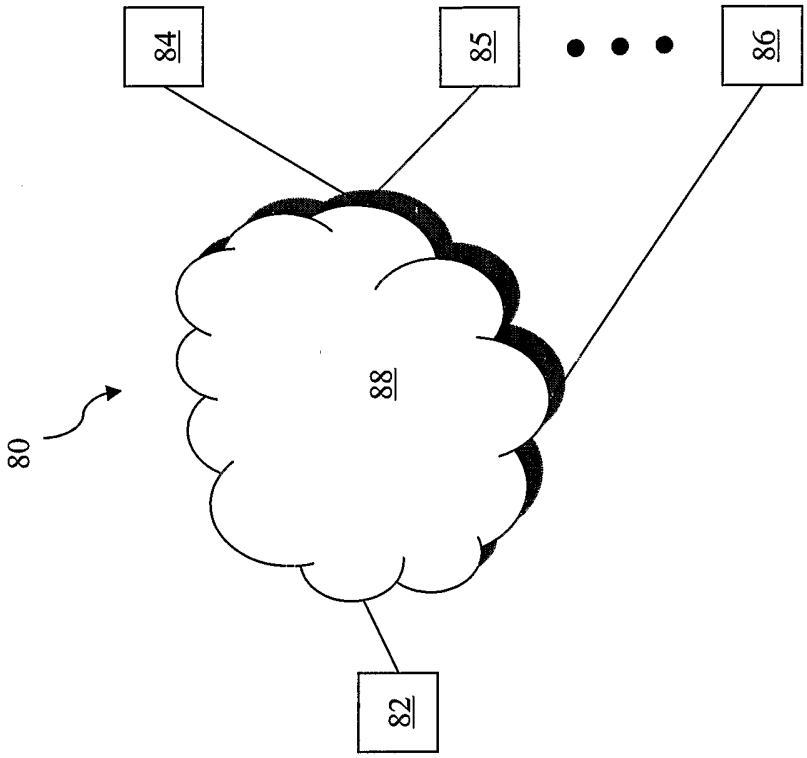


FIGURE 4

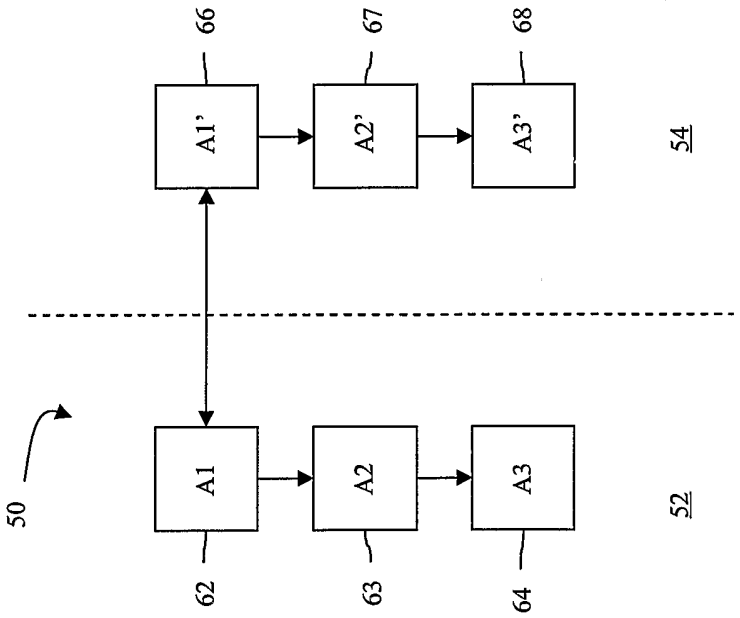


FIGURE 3

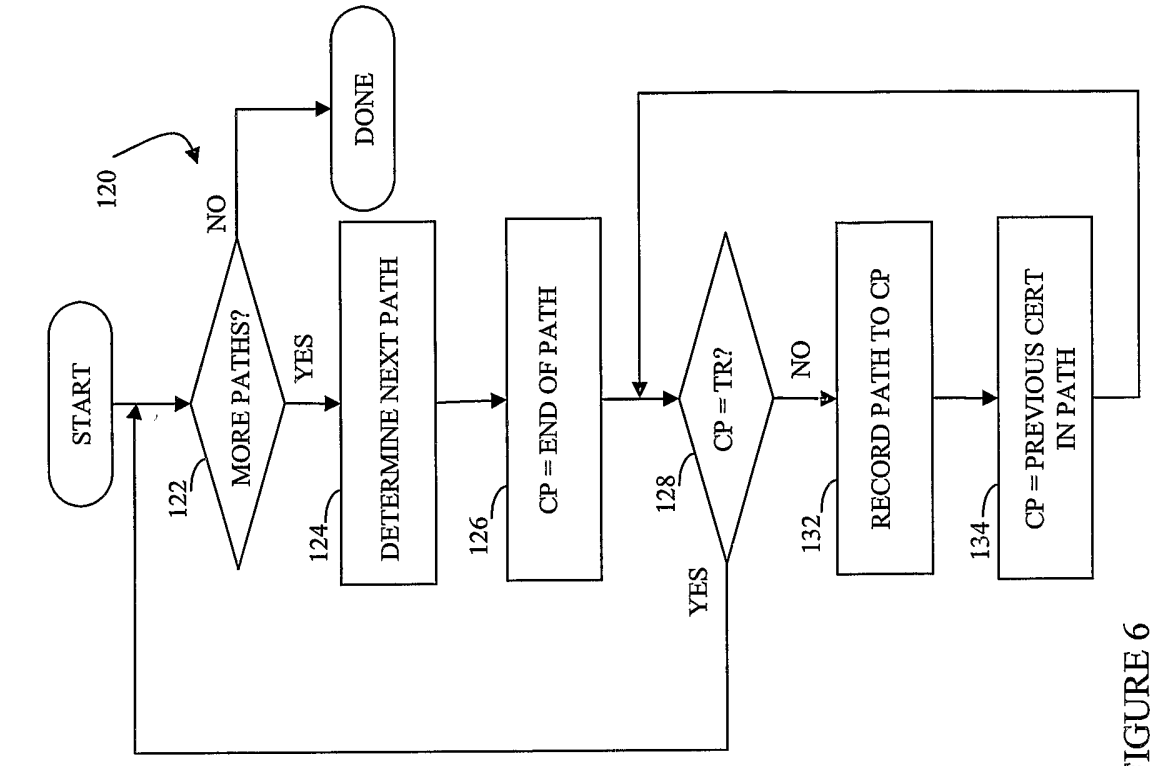


FIGURE 6

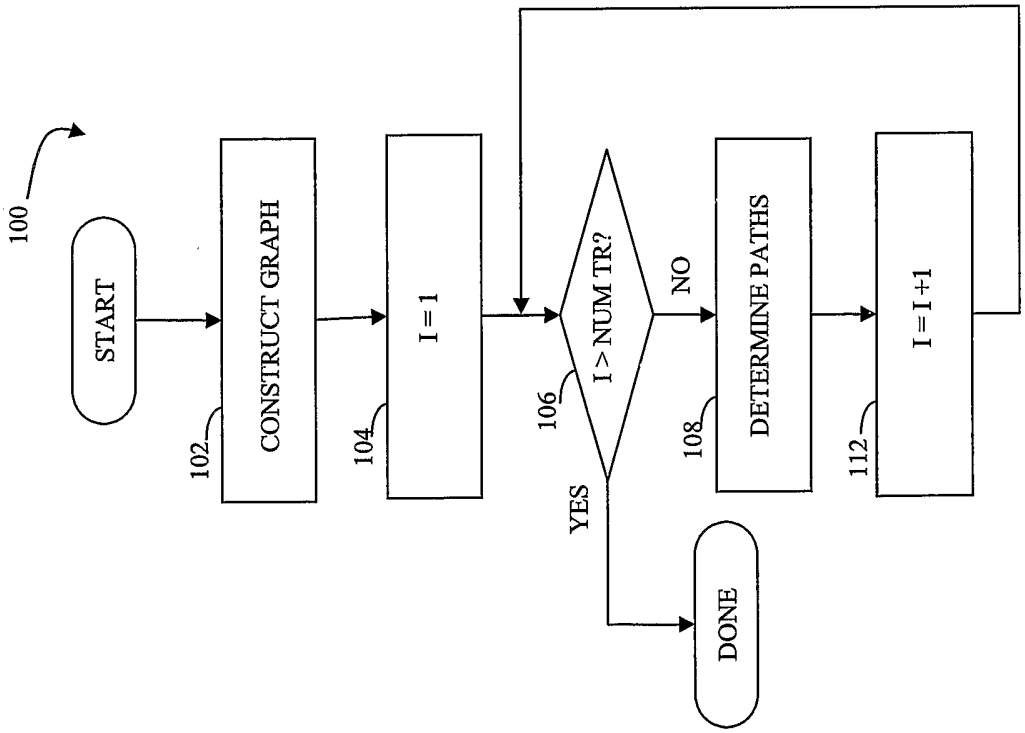


FIGURE 5

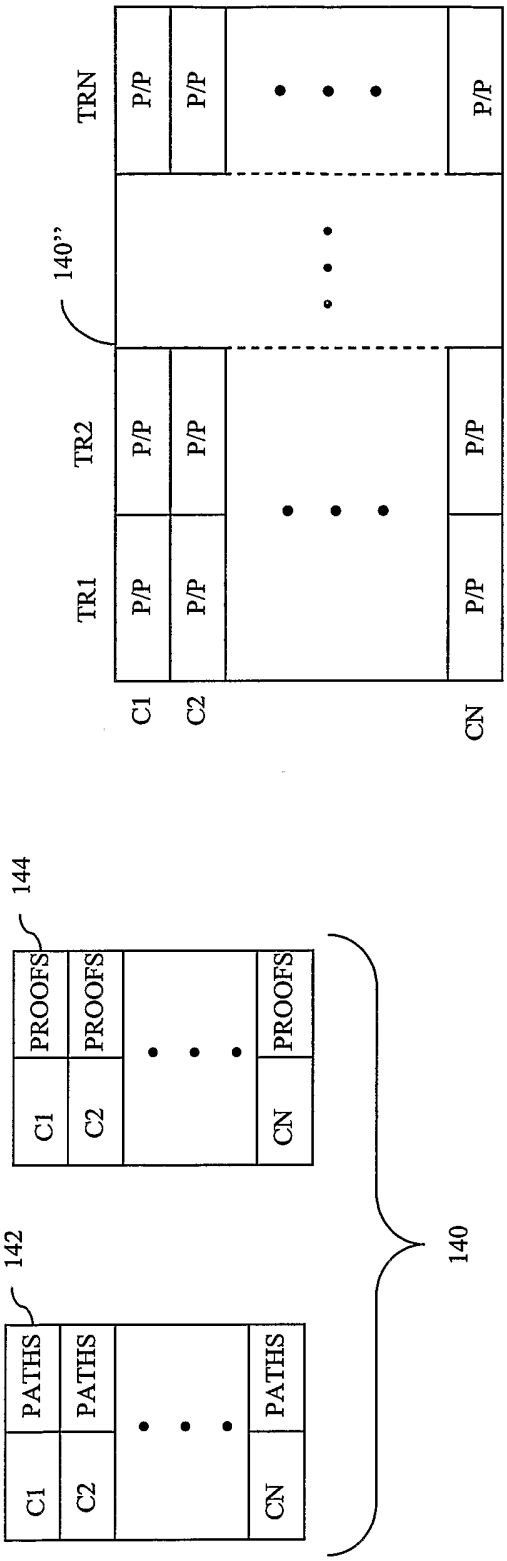


FIGURE 7A

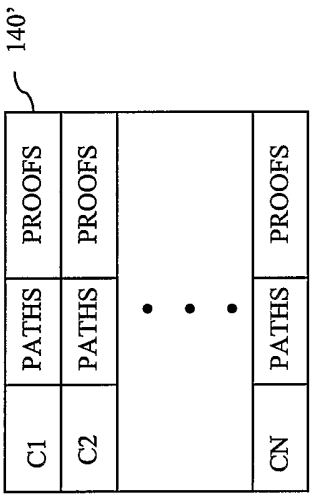


FIGURE 7B

FIGURE 7C

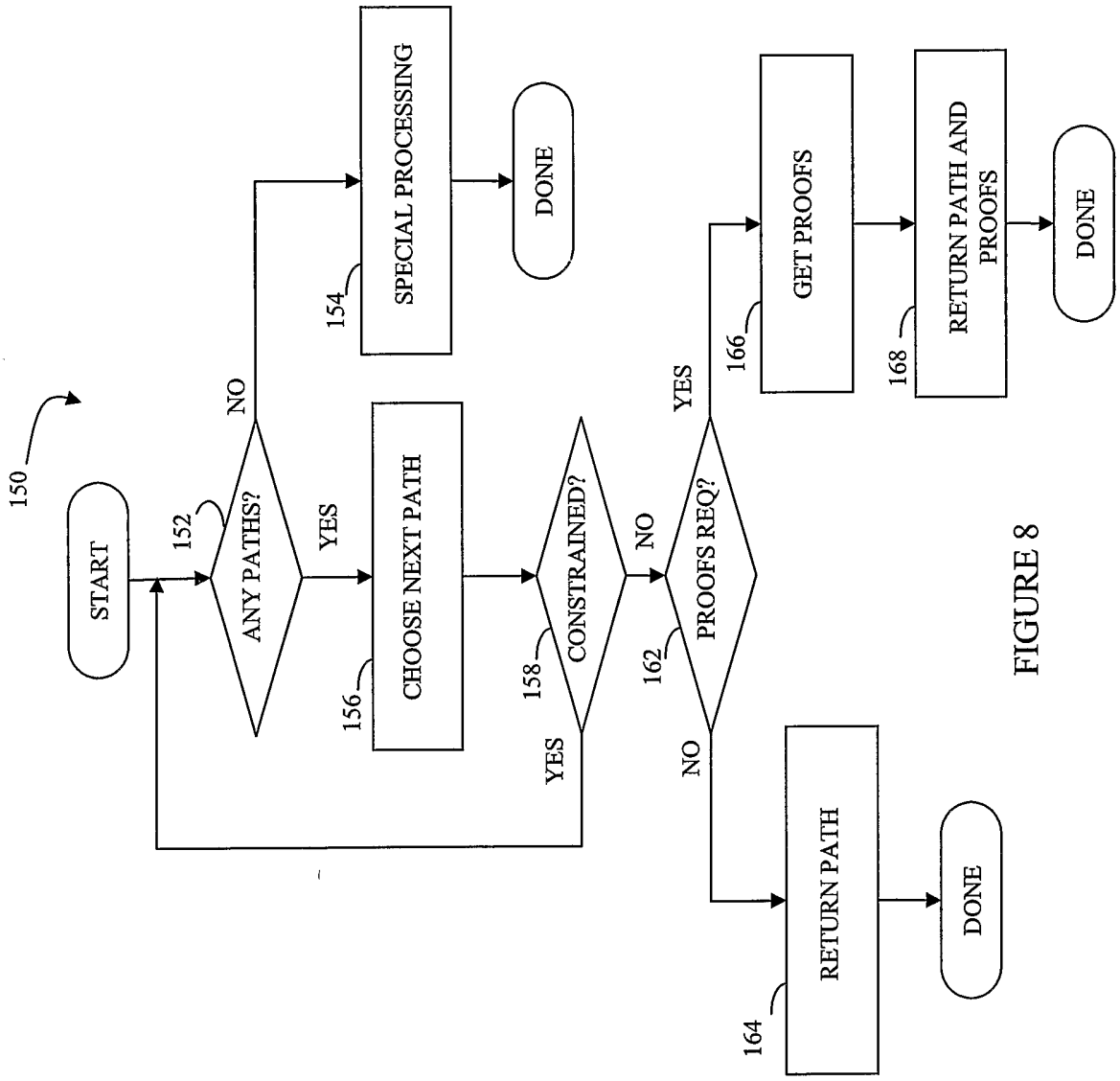

150
START
152
ANY PATHS?
NO
YES
154
SPECIAL PROCESSING
DONE
156
CHOOSE NEXT PATH
158
CONSTRAINED?
YES
NO
162
PROOFS REQ?
YES
166
GET PROOFS
168
RETURN PATH AND PROOFS
DONE
NO
164
RETURN PATH
DONE

FIGURE 8

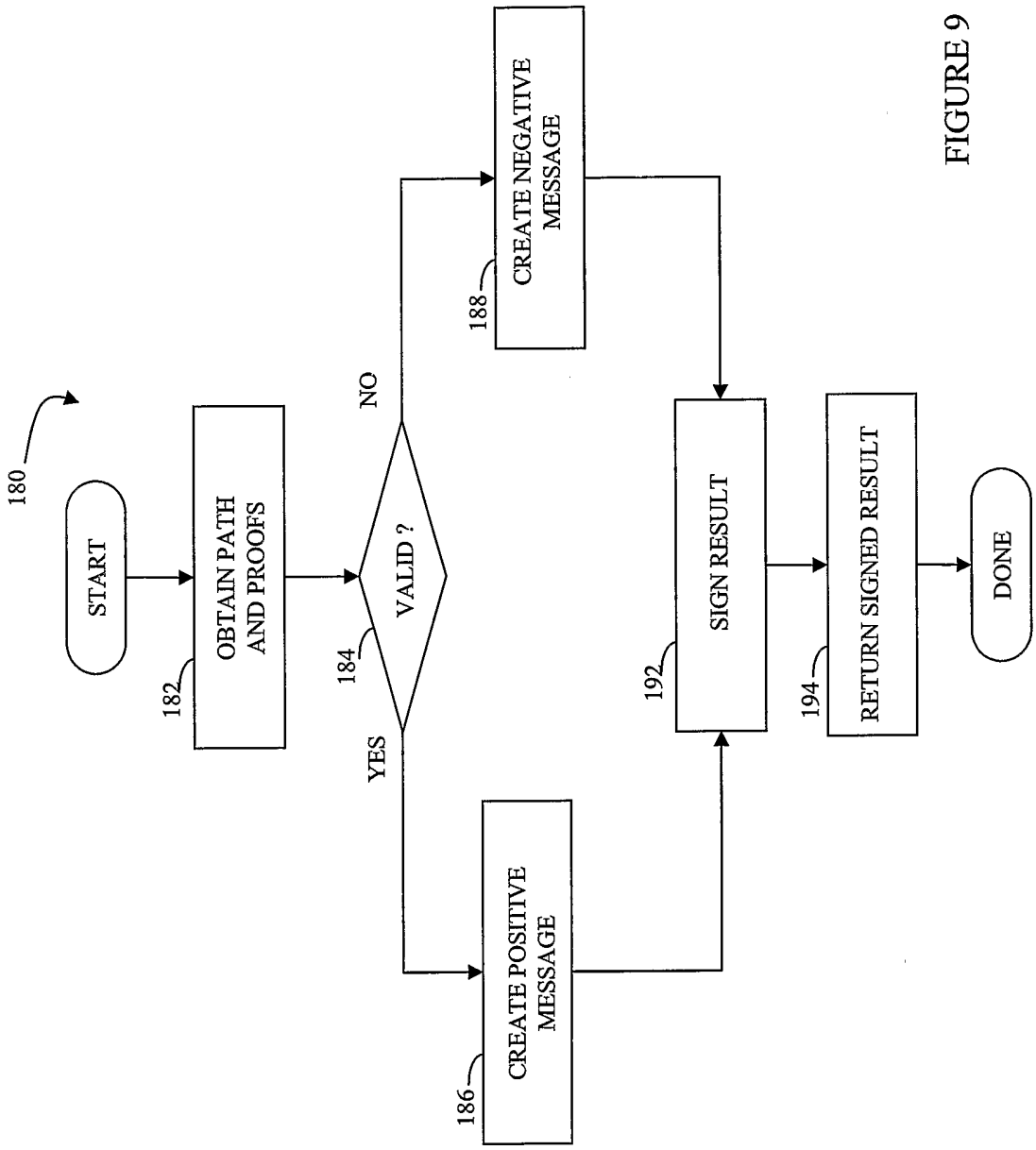

180
START
OBTAIN PATH AND PROOFS
182
VALID ?
184
NO
YES
CREATE NEGATIVE MESSAGE
188
CREATE POSITIVE MESSAGE
186
SIGN RESULT
192
RETURN SIGNED RESULT
194
DONE

FIGURE 9