APPARATUS AND METHOD FOR DEMONSTRATING AND CONFIRMING THE STATUS OF DIGITAL CERTIFICATES AND OTHER DATA

TECHNICAL FIELD

This invention relates generally to data security, information security, and cryptography, and specifically to systems for constructing digitally-signed lists and determining whether particular values are present on such lists. The invention has specific application to revocation of digital certificates or other types of digital data items and for determining whether such data items have been revoked.

BACKGROUND ART

Asymmetric (public key) cryptologic techniques are widely used in security protocols for applications such as secure e-mail, electronic commerce, encrypted voice and video communications, and security on the World Wide Web. The RSA cryptosystem, described in U.S. patent 4,405,829 to Rivest et al (1983), and the Digital Signature Algorithm (DSA), described in U.S. patent 5,231,668, to Kravitz, are examples of asymmetric functions. Asymmetric cryptosystems systems typically involve a secret private key, which is used for digital signing or decryption, and a non-confidential public key derived from the private key, which is used for signature verification or encryption. For general information about RSA, DSA, and other asymmetric cryptosystems, the reader is referred to *Applied Cryptography*.

Before using a public key to encrypt a confidential message or verify a signature, a party in an electronic communications protocol generally must confirm the identity of the party holding the private key. An electronic communications protocol defines conventions allowing two or more computers or other electronic devices to exchange digital data or messages via a communications channel. Without such confirmation, an attacker could substitute a legitimate public key with another for which the attacker knows the private key. Digital certificates are the most common solution to this problem. The holder of the private key provides its corresponding public key to a widely-trusted Certificate Authority (CA) along with acceptable identification. The CA then issues a certificate, which typically consists of a digital signature on a specially-formatted block of data containing the user's name, the user's public key, the certificate issuance and expiration dates, and the certificate serial number. The recipient of a digital certificate who trusts the issuing CA can use the CA's (already trusted) public key to verify the signature. If the signature is valid and if the CA is trustworthy, the recipient can trust that the person identified in the certificate holds the private key corresponding to the public key in the certificate. The ISO 9594-8 standard defines techniques and data formats for computing and verifying digital signatures and certificates.

Certificates often need to be revoked due to unexpected events such as compromise, theft, or loss of the device containing the private key. A certificate

might also need to be revoked if a user has lost the privileges granted by the certificate. In general, a certificate's status might be good, revoked, or pending, as well as other possibilities that will be appreciated by those skilled in the art.

In large open networks such as the Internet, certificate status determination, specifically certificate revocation, presents enormous challenges. The Internet is expected to have hundreds of millions of users worldwide soon. It is desirable that certificate revocation messages propagate as quickly as possible to all users who might otherwise accept the invalid certificate. There are thus difficult design constraints which a successful system must satisfy:

1. Network applications are sensitive to latency. A good solution should minimize the number of additional network connections and data exchanges required.

2. The system must work on a global scale and work on a broad range of systems with different levels of connectivity.

3. The system must be distributable so that critical information can be cached in many locations at once to minimize the number of long-distance network connections.

4. The system must be cryptographically secure.

Previous certificate revocation mechanisms, such as ISO 9594-8, use a type of digitally-signed structure called a Certificate Revocation List (CRL) which is issued periodically by the CA and lists the serial numbers of certificates issued by the CA which have been revoked. **FIG. 1** shows the structure of a typical CRL **101**, which consists of the issuer's name, a field identifying the signature algorithm, the date and time of issuance, and a list of revoked certificates, followed by a digital signature of the above information. To determine if a particular certificate is revoked, one obtains an up-to-date CRL from the appropriate CA, verifies that the digital signature in the CRL is valid, then searches the list of revoked certificates to determine whether the certificate in question is revoked. If the certificate is not on the list, it is assumed to be valid.

Because the complete CRL must be obtained and verified to determine the revocation status of a single certificate, CRLs do not scale well to large networks. In particular, existing certificate revocation mechanisms suffer from a number of disadvantages:

(a) CRLs can become extremely large, making them inefficient to transmit or process. For example, a very large system might have several million revoked certificates, resulting in CRLs which are many megabytes in size. To determine whether a particular certificate is valid, one must download a recent CRL in its entirety and process the entire list to verify the digital signature. For a large network, the required network bandwidth can be prohibitively large, especially if every

user needs to download new CRLs often. The time required to process a large list can also be an issue.

(b) Only mechanisms recognized and supported by the certificate recipient can be used to revoke certificates. In most cases, only revocations issued by the CA are accepted. Additional revocation mechanisms cannot be added easily.

(c) Because CAs are entrusted with both certificate issuance and revocation, physical destruction of a CA's private key could result in a situation where certificates could no longer be revoked without revoking the CA's public key.

(d) Verifiers must be able to obtain up-to-date CRLs from every supported CA. Certificate chaining makes this particularly difficult since there can easily be an extremely large number of CAs and multiple CAs per certificate chain.

Present techniques for determining whether types of data other than certificates are present on digitally-signed lists suffer from the same scalability problems. On large networks such as the Internet such systems will typically suffer from poor latency and extremely large bandwidth requirements. These limitations arise because existing techniques either require active network connections to a trusted server at transaction time or require replication of CRLs or other digitally-signed lists containing all elements of the list.

DISCLOSURE OF INVENTION

Accordingly, it is an object of the invention to reduce greatly the processing effort, network bandwidth, network latency, data storage, and data replication requirements needed to determine whether a particular certificate has been revoked. In particular, the invention allows certificate status to be determined without knowledge of the entire list of revoked certificates and without having to search the entire list of revoked certificates.

Another object of the invention is to simplify the addition of new revocation mechanisms, such as revocation by certificate holders, without altering existing revocation mechanisms.

Another object of the invention is to allow revocations from many CAs to be included efficiently in a single database, thereby allowing a single trusted source for certificate revocation messages.

Another object of the invention is to provide a certificate revocation system whose operation is open to public scrutiny to ensure that certificates are not maliciously revoked and that revoked certificates are not marked as valid.

In general, the invention can be used to determine whether data items of any type are present on a digitally-signed list without requiring that the verifier retrieve the entire list. It should be readily apparent

to a reader skilled in the art that the problem of determining securely whether a data item belongs to a list of data items has applications to many problems beyond certificate revocation. For example, an Internet user might want to determine whether a digitally-signed Java application has been revoked as having harmful side effects.

Briefly, the present invention typically includes at least one tree issuing device, one or more confirmation issuers, and at least one verification device.

The tree issuing device assembles a list of data items, which can have any content but would typically be a list of serial numbers identifying revoked digital certificates. The issuer sorts the list, optionally removes any duplicate entries, then adds a beginning-of-list marker and an end-of-list marker. Each pair of adjacent entries in this sorted list specifies a range between which there are no list entries. Except for the beginning and end markers, each list entry appears in two ranges, once as a minimum value and once as a maximum value. A hash tree is then constructed where leaf nodes correspond to ranges in the list. Because the tree's leaf nodes define intervals, this structure is referred to as an interval hash tree. A binary tree such as those described in U.S. patent 4,309,569 to Merkle (1982) would typically be used, but those skilled in the art will appreciate that a variety of other hash tree structures are also suitable (for example, as described in U.S. Patent 4,881,264 to Merkle

(1989)). Merkle uses hash trees to reduce the cost per signature when computing a large number of digital signatures by combining a large number of items into a single root node which can be digitally signed. Merkle's hash tree techniques produce assurances that particular items have been digitally signed. However, Merkle's hash trees do not provide the capability disclosed herein, of cryptographically demonstrating that particular items were not included in the tree (except in the highly inefficient case where the verifier obtains the entire tree and searches it for the particular candidate item).

The tree issuing device digitally signs the tree's root node (or nodes, if the chosen tree structure has multiple roots) with other data which would typically identify the issuer's identity and the date and time.

A confirmation issuer obtains the hash tree including the root node digital signature. The tree may be obtained either separately from, or together with, its digitally signed root. In the latter case, the hash tree shall be considered to include its digitally signed root(s). These values could be obtained directly from the tree issuer, be computed independently, or be obtained from other sources. The confirmation issuer can be the same device as the tree issuer, or could be an independent device connected to the tree issuer via a communications channel. Confirmation issuers might also be included in devices including, but not limited to, network servers, firewalls, or directory servers and might be assigned to serve a particular region of a network or subnetwork.

The verification device (verifier) begins with a "candidate data item" whose status on the list is to be determined. The verifier sends the candidate data item (or a representation thereof) to the confirmation issuer. The confirmation issuer locates a leaf node whose minimum range value is no larger than the candidate data item and whose maximum range value is no smaller than the candidate data item. The confirmation issuer then sends the verifier the appropriate leaf node, the digitally-signed root node, and the additional nodes needed to cryptographically derive the root node from the leaf node. By cryptographically determining that the particular leaf can be used to reconstruct the root node, the verifier gains cryptographic assurance that the leaf was part of the original tree whose root node was digitally signed. The leaf is said to be cryptographically bound to the root. Note that the confirmation issuer does not need a private key, or knowledge of tree issuer's private key, since it does not generate any new digital signatures.

The verifier confirms that the signature on the header is correct and comes from a trusted tree issuer; that the date, name, and other information included in the root node digital signature are appropriate; that the root node can be constructed from the leaf node using the specified supporting nodes; and that the candidate data item is within the range specified by the given leaf node. If any of the above verification steps fail, the assurance is bad and item's status on the list cannot be determined. If the verification steps are successful and

either range endpoint equals the data item, the verifier has cryptographic assurance that the data item is present on the list. If the smaller range endpoint is less than the data item and the larger endpoint is larger than the item, the verifier has cryptographic assurance that the item is not present on the list.

In addition to the embodiments described above, another illustrated embodiment does not use ranges, but rather uses a hash tree constructed from a sorted list of data items such that two adjacent leaves spanning a candidate data item provide cryptographic assurance that the candidate data item is not present on the list. Yet another illustrated embodiment does not use ranges, but rather uses the hash tree to build digitally signed assertions that specific items are not on the list. Still another illustrated embodiment does not use hash trees, but rather uses individually signed ranges.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows the contents of a traditional certificate revocation list which does not use the present invention.

FIG. 2 shows an exemplary technique for preprocessing data items.

FIG. 3 show a particular preprocessing technique appropriate for digital certificates.

FIG. 4 shows the construction of a set of ranges from a list of data items.

FIG. 5 shows a set of ranges made from a sorted list of data items.

FIG. 6 shows the structure of a binary interval hash tree made from a list containing six ranges.

FIG. 7 is a flowchart showing steps to construct an interval hash tree from a set of ranges.

FIG. 8 shows a degenerate interval hash tree.

FIG. 9 shows the contents of a digitally-signed root node.

FIG. 10 shows an alternate embodiment of the invention, without a hash tree, in which the individual ranges are signed directly.

FIG. 11 shows the steps taken by the confirmation issuer to issue a confirmation for a specified candidate item.

FIG. 12 is a flowchart showing steps to construct a list of supporting nodes cryptographically binding a leaf node to the root node.

FIG. 13 shows the steps typically taken by the verifier to determine whether a confirmation message provides acceptable cryptographic proof of the status of an item.

FIG. 14 is a flowchart showing how to use a set of supporting nodes to verify the cryptographic binding between a leaf node and the root node.

FIG. 15 outlines the operation of a communications system which a certificate holder uses to obtain confirmation messages which it provides to certificate acceptors as cryptographically-secure evidence that the certificate has not been revoked.

DETAILED DESCRIPTION OF THE INVENTION

Methods and apparatuses are disclosed for constructing efficient cryptographically secure assertions as to whether candidate item is present on a list. In one embodiment of the invention, data items on the list are converted into a set of ranges having data items as endpoints such that there are no data items on the list between the endpoints of any range. The ranges are used as leaf nodes to construct a hash tree, then the tree's root node is digitally signed by the tree issuer. A verifier can determine the validity of a leaf node by checking the path from the leaf to the root node and by checking the digital signature on the root node. A valid leaf node with a range endpoint equal to a candidate data item provides cryptographic assurance that the candidate data item is present on the list. A valid leaf node with one range endpoint larger than the candidate data item and one range endpoint smaller than the candidate data item provides cryptographic assurance that the candidate data item is not on the list.

Hash Tree Construction and Issuance

For certain kinds of data items, preprocessing using a collision-free function (CFF) may be performed before the data items are used for tree construction. A CFF is a function for which it is believed to be computationally unfeasible to find two different messages X and Y such that CFF(X)=CFF(Y). The identity function I(X)=X is a CFF and may be used, although for larger data items a cryptographic hash function is generally more

efficient. Cryptographic hash functions include SHA or MD5 and are used to reduce the size of messages (i.e. the size of H(X) is less than the size of X) yet H(X) is cryptographically collision-free. The exemplary preprocessing technique shown in **FIG. 2** uses a cryptographic hash function to reduce the size of the items. Each data item **201** is hashed at step **202** to produce a fixed-length processed item **203**. For example, if the data item consisted of a CA name ("Sample CA") followed by a four-byte serial number (decimal 123456789 with hexadecimal representation 07 5B CD 15), the processed item might be SHA("Sample CA" | 07 5B CD 15), where "|" denotes concatenation, or:

9D 76 7D 83 1D 85 A2 A8 35 95 08 DB 91 F2 AA DC D8 DD C4 AD.

Data items of specific types may use different kinds of preprocessing. **FIG. 3** illustrates a particular preprocessing technique appropriate for data items such as digital certificates. The certificate issuer name **301** is hashed at step **303**, and at step **304**, the hashed issuer name is concatenated with the certificate serial number **302** to produce the processed digital certificate **305**. (The serial number could also be hashed before concatenation.) For example, a certificate with a 32-bit (4-byte) serial number 123456789 whose CA name is "Sample CA" would have a 24-byte list entry consisting of SHA ("Sample CA") followed by the byte representation of 123456789. In particular, the hexadecimal representation would be:

E2 CA 64 56 40 BE 99 AC CA 9D 3A 9B 02 97 0D 1E F2 95 8E A0 07 5B CD 15

FIG. 4 shows one way to use a computer to convert the set of preprocessed data items into a set of ranges. At step **401**, the data items are assembled into a list stored in a computer-readable memory, which is then sorted in ascending order at step **402**. At step **403**, markers are added to denote the beginning and end of the list. If the data items consist of a 160-bit SHA output, the beginning-of-list marker might consist of 160 zero bits, and the end-of-list marker might consist of 160 one bits. Every pair of adjacent list entries then defines a range where the list entries are the range endpoints. There are no entries on the list which lie between these endpoints (if a value between the endpoints was present on the list, the sorting operation would have placed it between the range endpoints and the range endpoints would thus no longer be adjacent on the sorted list). At step **404**, a data structure is constructed specifying the range, typically encoding the range as the minimum (the lesser of the two list entries) and the range maximum (the greater list entry). Other formats for the range data structure are also possible, such as a range endpoint and length, a range midpoint and length, etc. Ranges can also be broken into subranges, although in this case range endpoints would not necessarily correspond to data items on the list. In some cases it is helpful to add additional markers in places other than the beginning and end of the list. For example, if the digital certificate preprocessing technique of **FIG. 3** is

used and certificates from multiple certificate issuers are present on the list, additional markers might be placed at the beginning and end of the list region corresponding to each certificate issuer. Ranges would only be issued within regions belonging to supported CAs. In general, additional start and stop signals are helpful to define ranges excluded from the list or which are otherwise noteworthy.

If the initial data set contains n data items, there will be n+2 entries in the sorted list due to the addition of the beginning and end markers. There are n+1 possible pairs of adjacent entries in the sorted list, so there will be n+1 range structures. **FIG. 5** shows a set of ranges constructed from a sorted list **501** of five data items $I_0..I_4$. The first range **502** goes from the beginning-of-list marker to the first data item, I_0. The next range **503** goes from I_0 to I_1. Subsequent ranges **504** are I_1-I_2, I_2-I_3, and I_3-I_4. The final range **505** is I_4 through the end-of-list marker.

A hash tree is then be built from the sorted list of ranges. A hash tree is a hierarchical data structure comprising a plurality of leaf nodes combined using a cryptographic function to form a lesser number of root nodes such that using one or more applications of the cryptographic hash function it is possible to cryptographically transform any leaf node into a root node. A hash tree where the leaves are intervals (ranges) is called an interval hash tree.

FIG. 6 shows the structure of a binary interval hash tree built using a set of six ranges **601**. The ranges are optionally transformed using a collision-free function to form the tree's leaf nodes **602**. The leaf nodes are then combined using a cryptographic hash function to form levels of intermediate nodes **603** and a single root node **604**.

FIG. 7 describes the steps required to use a computer to construct a binary interval hash tree given a set of n leaf nodes $N_{0,0}...N_{0,n-1}$ stored in a computer readable memory. At step **701**, the variable n is initialized to the number of leaf nodes and the tree level counter x is initialized to zero. At step **702**, the variable y is initialized to zero. At step **703**, the device determines whether two times y is smaller than n-1. If so, at least two more nodes are present at the current level x and, at step **704**, these are concatenated (as denoted by the symbol "|") and combined with a cryptographic hash function to produce one level x+1 node. At step **705** y is updated then the computer returns to step **703**. If the comparison at step **703** is not true, the computer determines at step **706** whether a single level x node remains. If so, the node is simply copied at step **707** to level x+1. The level x+1 nodes are now complete, so at step **708** the device adds one to x so that the next level of nodes can be done. Step **708** also replaces n with the number of nodes in the new level by dividing by 2 while rounding upward. At step **709** the device then determines whether more than one node remains. If so, it returns to step **702**. Otherwise the

device finishes at step **710**, returning the root node which is node $N_{x,0}$.

Although the preferred embodiment uses binary trees (see, e.g., U.S. patent 4,309,569 to Merkle (1982)), other tree structures are also possible. Variant tree structures, such as those having more than one root node, which combine more than two nodes at once, or which otherwise deviate from the binary tree are called degenerate trees. In some situations it may be desirable to have multiple roots since this can shorten the paths from leaves to roots. **FIG. 8** shows an example of a degenerate tree in which groups of three nodes **801** (instead of two) are combined, a level x node **802** (i.e., $N_{1,1}$) is used in computing more than one level x+1 nodes **802** (e.g., $N_{2,0}$ and $N_{2,1}$), and there are two root nodes **803** and **804**. It will be apparent to one skilled in the art that a wide variety of degenerate tree structures can be constructed. For example, U.S. patent 4,881,264 to Merkle (1989) describes several degenerate hash tree structures which may be used in connection with the present invention.

Referring to **FIG. 9**, after constructing the hash tree, the tree issuer uses RSA, DSA, or another signature algorithm to digitally sign the tree's root node **901**, the date and time of the tree's issuance **902**, the date and time of the next issuance (optional) **903**, and the total number of nodes in the tree **904**. The structure might also include other information, for example (but not limited to) the signing algorithm

identifier, the tree issuer's name, and the root node of the previous tree.

To summarize, the tree issuer thus performs the following steps:

1. Construct the list of items,

2. Convert list into a set of ranges,

3. Build an interval hash tree from the ranges,

4. Digitally sign the hash tree's root node, and

5. Publish the hash tree and signed root node.

The foregoing illustrates the preferred embodiment of the invention in which hash trees are used. Alternatively, **FIG. 10** shows a treeless variant in which the individual ranges **1001** are signed directly at **1002** to produce a set of signed ranges **1003**.

Confirmation Issuance

Corresponding to the embodiments of the invention disclosed in **FIGS. 2-9**, **FIG. 11** outlines a process for issuing cryptographic assurances as to whether specific items are on a list. The list is the plurality of data items used to generate the ranges for an interval hash tree, and the confirmation will directly demonstrate whether a candidate data item belongs to the plurality of items (i.e., is present on the list). At step **1101**, the confirmation issuer first obtains the

interval hash tree (or information allowing the confirmation issuer to construct at least the required portions of the tree) along with its digitally signed root node. At step **1102**, the confirmation issuer receives a candidate data item for which the confirmation is to be constructed. At step **1103**, the confirmation issuer performs any required preprocessing steps. At step **1104**, the confirmation issuer identifies a leaf node representative of the candidate item. In one embodiment of the invention, the identified leaf node has a lesser range endpoint which is no larger than the candidate item and a larger range endpoint which is no smaller than the candidate item. Next, at step **1105**, a list is made specifying the intermediate nodes needed to cryptographically reconstruct the path binding the leaf to the root node (**FIG. 12** describes in detail the steps required to locate the appropriate intermediate nodes). At step **1106**, the final confirmation including the contents of the leaf node whose range spans the candidate data item, the number of the specified leaf node in the tree, the intermediate nodes which cryptographically bind the leaf to the root node, and the digitally-signed root node, is produced. Note that the confirmation does not include the entire list of data items represented by the leaf nodes of the interval hash tree. Finally, at step **1107**, the confirmation issuer issues the confirmation to the party requesting it. The requesting party may be either a party wishing to know the status of the candidate item, or it may be an intermediate in a communications channel. For example, the requesting party might be the owner of a certificate which will

store the confirmation and supply it to any other parties which want verify the status of its certificate. In this case, the confirmation is communicated by the confirmation issuer to the verifier via the certificate holder. Alternatively, the confirmation might be requested by the party which wishes to verify the certificate's status.

FIG. 12 describes steps to find the intermediate nodes needed to cryptographically bind a leaf node ($N_{0,i}$) to the root node of a binary interval hash tree. The process begins at step **1201** by setting node counter n to the total number of leaf nodes in the tree and i to the vertical position of the leaf node to be bound to the root. At step 1202, x is initialized to zero. At step **1203**, the device computes $i \oplus 1$, where "$\oplus$" denotes an exclusive-OR operation. Equivalently, step **1203** could be defined as $y=i+1-2(i \bmod 2)$. At step **1204**, if y is less than the total number of level x nodes (hashes) then $N_{x,y}$ is added to the list of hashes binding the specified leaf to the root. At step **1205**, i is divided by 2 and rounded upward to find the vertical position of the level x+1 node leading to the root. At step **1206**, n is updated to equal the number of level x+1 nodes. At step **1207**, x is incremented. At step **1208**, the device tests whether n is larger than 1 and, if so, loops back to step **1203**.

To summarize, the confirmation issuer obtains the hash tree and signed root, receives a confirmation request, and constructs and issues the confirmation. When the hash tree is about to expire or a new tree is

available, the confirmation issuer obtains an updated tree.

Verification

FIG. 13 outlines the steps taken by a verifier to use a confirmation message stored in a computer readable memory to cryptographically determine the status of a candidate data item with respect to the list. At step **1301**, the verifier first uses the tree issuer's public key (which is assumed to be previously known and trusted) to check the digital-signature on the root node. At step **1302**, the verifier then checks the date and time of the tree's issuance, along with any other auxiliary information signed with the root, to ensure that the tree is acceptable. At step **1303**, the verifier confirms that the leaf node is representative of the data item. In particular, the leaf's lesser range endpoint should be no larger than the data item and the larger range endpoint should be no smaller than the data item. At step **1304**, the verifier uses the supporting nodes to check the cryptographic binding between the leaf node and the root node (**FIG. 14** shows the steps required to verify the cryptographically binding between a leaf node and the root node in a binary tree). If any steps fail, the verifier skips to step **1308** and the confirmation is invalid. If all steps above are successful, the verifier checks at step **1305** whether either endpoint of the leaf node's range equals the candidate data item. If the candidate data item lies between the endpoints of the range, the verifier concludes at step **1306** with assurance that the item is not on the list. If the item in

question equals one of the range endpoints, the verifier concludes at step **1307** with assurance that the item is on the list.

FIG. 14 outlines the process of using a set of supporting nodes to verify the cryptographic binding between the leaf node representative of the candidate data item and the root node. At step **1401**, the variable i is initialized to the number of the leaf node in the tree, n is initialized to the number of nodes in the tree, x is initialized to zero, R is initialized to the leaf node (i.e., $N_{0,i}$), and k is initialized to the number of supporting hashes provided. Note that n was checked along with the digital-signature on the root node. At step **1402**, the device checks whether i is even and equal to n-1. If so, the device skips directly to step **1408**, but otherwise the device proceeds to step **1403** and increments j. At step **1404**, the device ensures that j (the number of supporting hashes used) never exceeds k (the total number of supporting hashes provided). At step **1405**, the verifier determines whether i is even or odd. If i is odd, the verifier proceeds to step **1406** where the new value for R is found by concatenating the existing R with the next supporting hash ("|" denotes concatenation) and hashing the result with a cryptographic hash function. If i is even, the verifier proceeds instead to step **1407**, where the new value for R is found by concatenating the existing next supporting hash with the existing R (in the opposite order from step **1406**) and hashing the result with a cryptographic hash function. After step **1406** or **1407**, the verifier proceeds

to step **1408** and divides i by 2 (rounding upward), divides n by 2 (rounding upward), and increments x. At step **1409** the verifier checks whether the main loop is complete, returning to step **1402** if n has not yet reached one. If the loop has finished, the verifier finally checks, at step **1411**, whether R corresponds to the expected value of the root node from the confirmation. If R corresponds to the root node, the verifier concludes at step **1412** that the binding is good. If R does not correspond to the root node, the verifier concludes at step **1413** that the binding is bad.

FIG. 15 outlines the operation of a system which uses the invention to determine whether certificates have been revoked. The tree issuer **1501** constructs a list of revoked certificates by obtaining CRLs or other revocation messages **1502**. The tree issuer then constructs an interval hash tree including the digitally signed root node(s). The confirmation issuer **1503** obtains the tree's signed root from the tree issuer over a communications channel. The confirmation issuer also obtains, typically also from the tree issuer, the rest of the tree or the leaf nodes needed to reconstruct the tree. The owner of a certificate **1504** submits its certificate to the confirmation issuer, which responds with a confirmation. The certificate holder can then provide the confirmation along with its certificate to certificate acceptors **1505**, which each verify the confirmation message to confirm that the certificate has not been revoked. Participants in the protocol can verify the operation of the tree issuer to detect

attempts at fraud. In particular, the tree should include all revoked certificates with no unauthorized additions. For every certificate included in the tree, the tree issuer should be able to provide a CRL or other acceptable evidence of revocation. In some cases, such as if a CA stops issuing CRLs, the tree issuer can optionally define alternate mechanisms for adding entries to the list. In general, the tree issuer can determine the criteria for including items in the list, allowing the addition of new revocation mechanisms, such as revocation by certificate holders. Even so, the operation of the tree issuer is open to public scrutiny. In particular, third parties can verify that the tree's leaf nodes specify only properly-revoked certificates and that no revoked certificates were omitted. The third party can also confirm that the leaf nodes' ranges were constructed properly and detect any other abnormalities.

Those skilled in the art will appreciate that many simple variant forms of the invention are possible. For example, in another embodiment of the invention, the hash tree can be constructed using the sorted list of items (rather than ranges) as leaf nodes, in which case confirmations will consist of a pair of adjacent leaf nodes whose values span the candidate item, the intermediate nodes connecting the leaf nodes to the root, and the digitally-signed root. It is also possible to combine the functionality of the tree issuer and confirmation issuer within a single entity. In yet another embodiment of the invention, a trusted confirmation issuer can generate digitally-signed

assurances as to the status of candidate items on the list, rather than including a chain of nodes from the interval hash tree as part of the confirmation. Finally it is also possible for the confirmation issuer to issue confirmations without receiving an explicit request.

Conclusions

Accordingly, the reader will see that this invention can be used to efficiently and securely demonstrate the presence or absence of items on a list. Traditional solutions to this problem are inefficient in that an amount of data must be downloaded that is proportional to the number of items on the list. In contrast, the present invention provides tremendous savings for large lists. For example, by using binary interval hash trees, the amount of data required is proportional to the base-2 logarithm of the total list size. Construction of such hash trees is easy, requiring only (2n-1) cryptographic hash operations, where n is the number of leaves in the tree. Furthermore, construction of confirmation messages can be done efficiently and easily using insecure hardware since no private key operations are performed by the confirmation issuer.

Reliable and efficient verification of certificate revocation status is absolutely essential for secure World Wide Web transactions, Internet-based EDI, distributed design and manufacturing collaboration, financial transaction security, secure exchange of electronic mail, communication of medical records, and other applications requiring digital certificates. Using

this invention, parties holding digital certificates can obtain tree leaves corresponding to their own certificates and supply the tree and corresponding signature information along with the certificate. This eliminates the need for certificate recipients to independently download a CRL. Furthermore, the number of extra network connections involved in certificate revocation status is small, since a certificate holder can use a single nonrevocation proof for many transactions.

Although the description above contains many specificities, these should not be construed as limiting the scope of the invention but merely providing illustrations of some of the exemplary embodiments thereof. For example, the security system can be used with many sorts of data, including but not limited to lists of revoked certificates, revoked digital signatures on computer-executable code (such as executable code, object code, source code, interpreted code, etc.), lists of other types of revoked digital signatures of other types, lists of authorized users, lists of unauthorized users, lists of stolen credit card numbers, and lists of bad checks. In general, the invention is useful whenever it is necessary to determine in a secure manner whether or not a particular value is on a list. The system can be implemented using almost any computer technology, including almost any software language (Pascal, C, C++, assembly language, FORTRAN, etc.) or integrated circuit manufacturing technology.

CLAIMS

WHAT IS CLAIMED IS:

1. A computer-based method of generating an interval hash tree, for use in an electronic communications protocol for generating a cryptographically assured confirmation that a candidate data item does not belong to a plurality of data items, without including all of said data items in said confirmation, comprising the steps of:

(a) storing, in a memory device, a plurality of data items;

(b) deriving a plurality of leaf nodes from said data items;

(c) generating, from said leaf nodes, a plurality of intermediate nodes and at least one root node; and

(d) digitally signing said at least one root node.

2. The method of claim 1 wherein the step of deriving said plurality of leaf nodes includes operating on said data items with a collision-free function.

3. The method of claim 2 wherein said collision-free function is a cryptographic hash.

4. The method of claim 2 wherein said collision-free function is an identity transformation.

This Sheet Has Been Replaced

REPLACED SHEETS SG

☐ Description : paged ..... to paged .....
☑ Claim(s) : paged 28 to paged 47
☐ Drawing(s) : paged ..... to paged .....
☐ Abstract : paged ..... to paged .....

5. The method of claim 2 wherein each of said leaf nodes specifies a range in which there are no data items.

6. The method of claim 5 wherein said ranges are constructed by:

(a) ordering said operated on data items; and

(b) combining selected ones of adjacent said ordered data items.

7. The method of claim 2 wherein each of said leaf nodes represents one of said data items.

8. The method of claim 1 wherein the step of generating said intermediate and root nodes includes:

(a) combining at least two of said leaf nodes to form a first intermediate node; and

(b) forming additional ones of said intermediate and root nodes, by combining a plurality of nodes selected from the group consisting of said leaf nodes and previously formed intermediate nodes, until said at least one root node is formed.

9. The method of claim 1 wherein said interval hash tree is of binary type.

10. The method of claim 1 wherein said interval hash tree is of degenerate type.

This Sheet Has Been Replaced

11. The method of claim 1 wherein said data items represent digital signatures.

12. The method of claim 11 wherein said data items represent revoked digital certificates.

13. The method of claim 12 wherein said digital certificates originate from a plurality of certification authorities.

14. A computer-based method of generating an interval hash tree, for use in an electronic communications protocol for generating a cryptographically assured confirmation as to whether a candidate data item belongs to a plurality of data items, without including all of said data items in said confirmation, comprising the steps of:

(a) storing, in a memory device, a plurality of data items;

(b) deriving a plurality of leaf nodes from said data items;

(c) generating, from said leaf nodes, a plurality of intermediate nodes and at least one root node; and

(d) digitally signing said at least one root node.

15. An interval hash tree contained in a computer-readable memory, for use as part of an electronic communications protocol for cryptographically confirming that a candidate data item does not belong to

This Sheet Has Been Replaced

a plurality of data items, wherein selected portions of said interval hash tree are sufficient to cryptographically confirm that said candidate data item does not belong to said plurality of data items without specifying all of said data items, said interval hash tree including:

(a) a plurality of leaf nodes derived from said data items;

(b) a plurality of intermediate nodes; and

(c) at least one digitally signed root node.

16. The interval hash tree of claim 15 wherein each of said leaf nodes includes at least one of said data items that has been operated on with a collision-free function.

17. The interval hash tree of claim 16 wherein said collision-free function is a cryptographic hash.

18. The interval hash tree of claim 16 wherein said collision-free function is an identity transformation.

19. The interval hash tree of claim 16 wherein each of said leaf nodes specifies a range in which there are no data items.

20. The interval hash tree of claim 19 wherein each of said leaf nodes includes a plurality of adjacent

This Sheet Has Been Replaced

data items selected from the group consisting of an ordered sequence of said operated-on data items.

21. The interval hash tree of claim 16 wherein each of said leaf nodes represents one of said data items.

22. The interval hash tree of claim 15 wherein said intermediate and root nodes are formed by a process comprising the steps of:

(a) combining at least two of said leaf nodes to form a first intermediate node; and

(b) forming additional ones of said intermediate and root nodes, by combining a plurality of nodes selected from the group consisting of all said leaf nodes and all previously formed intermediate nodes, until said root node is formed.

23. The interval hash tree of claim 15 in which said interval hash tree is of binary type.

24. The interval hash tree of claim 15 in which said interval hash tree is of degenerate type.

25. The interval hash tree of claim 15 wherein said data items represent digital signatures.

26. The interval hash tree of claim 25 wherein said data items represent digital certificates.

This Sheet Has Been Replaced

27. The interval hash tree of claim 26 wherein said digital certificates are of revoked status.

28. The interval hash tree of claim 27 wherein said digital certificates originate from a plurality of certification authorities.

29. The interval hash tree of claim 27 wherein said revoked digital certificates are specified via certificate revocation lists of ISO 9594-8 format.

30. The interval hash tree of claim 26 wherein said digital certificates are of pending status.

31. The interval hash tree of claim 25 wherein said digital signature is on a segment of computer-executable code.

32. An interval hash tree contained in a computer-readable memory, for use as part of an electronic communications protocol for cryptographically confirming whether a candidate data item belongs to a plurality of data items, said interval hash tree comprising:

(a) a plurality of leaf nodes derived from said data items;

(b) a plurality of intermediate nodes; and

(c) at least one digitally signed root node;

This Sheet Has Been Replaced

wherein selected portions of said interval hash tree are sufficient to cryptographically assure whether said candidate data item belongs to said plurality of data items without specifying all of said data items.

33. A computer-based method of providing a confirmation that a candidate data item does not belong to a plurality of data items, for use in an electronic communications protocol, comprising the steps of:

(a) obtaining an interval hash tree representative of a plurality of data items, said interval hash tree including a plurality of leaf nodes derived from said data items, at least one digitally signed root node, and a plurality of intermediate nodes interrelating said leaf nodes and said root node;

(b) electronically generating a cryptographically assured confirmation that said candidate data item does not belong to said plurality of data items, without including all of said data items in said confirmation; and

(c) communicating said confirmation to a party wishing to know whether said candidate data item belongs to said plurality of data items.

34. The method of claim 33 wherein the step of communicating said confirmation is performed upon receiving a request identifying said candidate data item for which said confirmation is requested.

35. The method of claim 33 wherein said data items represent digital signatures.

This Sheet Has Been Replaced

36. The method of claim 35 wherein said candidate data item represents a digital certificate having a public key whose corresponding private key is held by an entity requesting said confirmation.

37. The method of claim 35 wherein said candidate data item represents a digital certificate having a public key whose corresponding private key is not known to an entity requesting said confirmation.

38. The method of claim 33 wherein the step of generating said confirmation includes:

(a) selecting a subset of said intermediate nodes that cryptographically bind said root node to at least one leaf node representative of said candidate data item; and

(b) combining said at least one leaf node representative of said candidate data item, said selected subset of intermediate nodes, and said root node to form said confirmation.

39. The method of claim 38 wherein said selecting said subset of intermediate nodes includes:

(a) identifying said at least one of said leaf nodes representative of said candidate data item; and

(b) adding additional ones of said nodes from said hash tree to said subset, such that each added additional node and at least one previously identified

This Sheet Has Been Replaced

node can be combined to identify a previously unidentified node, until said root node is identified.

40. The method of claim 33 wherein said interval hash tree is of conventional type.

41. The method of claim 33 wherein said interval hash tree is of degenerate type.

42. The method of claim 33 wherein each of said leaf nodes includes at least one of said data items that have been operated on with a collision-free function.

43. The method of claim 40 wherein said collision-free function is a cryptographic hashing function.

44. The method of claim 40 wherein said collision-free function is an identity transformation.

45. The method of claim 40 wherein each of said selected leaf nodes specifies a range in which there are no said operated-on data items.

46. The method of claim 40 wherein each of said leaf nodes represents one of said data items.

47. The method of claim 33 wherein the step of obtaining said interval hash tree includes:

This Sheet Has Been Replaced

(a) receiving said plurality of leaf nodes derived from said data items; and

(b) receiving said at least one digitally signed root;

(c) using said leaves to construct said intermediate nodes, wherein said intermediate nodes are capable of cryptographically binding said leaves to said digitally signed root.

48. The method of claim 33 wherein the step of obtaining said interval hash tree includes receiving said interval hash tree in its entirety from an external source.

49. The method of claim 33 wherein the step of obtaining said interval hash tree includes:

(a) constructing said interval hash tree; and

(b) digitally signing a root node of said interval hash tree to form said digitally signed root node.

50. A computer-based method of providing a confirmation as to whether a candidate data belongs to a plurality of data items, for use in an electronic communications protocol, comprising the steps of:

(a) obtaining an interval hash tree representative of a plurality of data items, said interval hash tree including a plurality of leaf nodes derived from said data items, at least one digitally

This Sheet Has Been Replaced

signed root node, and a plurality of intermediate nodes interrelating said leaf nodes and said root node;

(b) electronically generating a cryptographically assured confirmation as to whether a candidate data item belongs to said plurality of data items, without including all of said data items in said confirmation; and

(c) communicating said confirmation to a party wishing to know whether said candidate data item belongs to said plurality of data items.

51. A cryptographic confirmation, for use as part of an electronic communications protocol, that a candidate data item does not belong to a plurality of data items, said confirmation contained in a computer-readable memory and prepared by a process comprising the steps of:

(a) obtaining an interval hash tree representative of a plurality of data items, said interval hash tree including a plurality of leaf nodes derived from said data items, at least one digitally signed root node, and a plurality of intermediate nodes interrelating said leaf nodes and said root node;

(b) selecting a subset of said intermediate nodes that cryptographically bind said root node to at least one leaf node representative of said candidate data item; and

(c) combining said at least one leaf node representative of said candidate data item, said selected subset of intermediate nodes, and said root node to form

This Sheet Has Been Replaced

said confirmation, wherein said confirmation does not include all of said data items.

52. The confirmation of claim 51 wherein the step of selecting said subset of intermediate nodes includes:

(a) identifying said at least one of said leaf node representative of said candidate data item;

(b) identifying additional ones of said nodes from said hash tree, such that each identified additional node and at least one previously identified node can be combined to identify a previously unidentified node, until said root node is identified; and

(c) selecting at least those of said identified nodes that are not in a direct path between said root node and said leaf node representative of said candidate data item.

53. The confirmation of claim 51 wherein each of said leaf nodes includes at least one of said data items that has been operated on with a collision-free function.

54. The confirmation of claim 53 wherein said collision-free function is a cryptographic hash.

55. The confirmation of claim 53 wherein said collision-free function is an identity transformation.

This Sheet Has Been Replaced

56. The confirmation of claim 53 wherein each leaf node specifies a range in which there are no data items.

57. The confirmation of claim 53 wherein each of said leaf nodes represents one of said data items.

58. The confirmation of claim 51 wherein said interval hash tree is of binary type.

59. The confirmation of claim 51 wherein said interval hash tree is of degenerate type.

60. The confirmation of claim 51 wherein said data items represent digital signatures.

61. The confirmation of claim 60 wherein said data items represent digital certificates.

62. The confirmation of claim 61 wherein said digital certificates are of revoked status.

63. The confirmation of claim 62 wherein said digital certificates are specified via certificate revocation lists of ISO 9594-8 format.

64. The confirmation of claim 61 wherein said digital certificates are of pending status.

This Sheet Has Been Replaced

65. The confirmation of claim 60 wherein said digital signature is on a segment of computer-executable code.

66. A cryptographic confirmation, for use as part of an electronic communications protocol, as to whether a candidate data item belongs to a plurality of data items, said confirmation contained in a computer-readable memory and prepared by a process comprising the steps of:

(a) obtaining an interval hash tree representative of a plurality of data items, said interval hash tree including a plurality of leaf nodes derived from said data items, at least one digitally signed root node, and a plurality of intermediate nodes interrelating said leaf nodes and said root node;

(b) selecting a subset of said intermediate nodes that cryptographically bind said root node to at least one leaf node representative of said candidate data item; and

(c) combining said at least one leaf node representative of said candidate data item, said selected subset of intermediate nodes, and said root node to form said confirmation, wherein said confirmation does not include all of said data items.

67. A computer-based method of cryptographically verifying that a candidate data item does not belong to a plurality of data items, comprising the steps of:

This Sheet Has Been Replaced

(a) receiving, over an electronic communications channel, a cryptographically assured confirmation that said candidate data item does not belong to said plurality of data items, said confirmation including at least one leaf node, at least one intermediate node, and a digitally signed root node, wherein said confirmation does not include all of said data items; and

(b) using said computer to cryptographically verify the validity of said confirmation.

68. The method of claim 67 wherein the step of using said computer to cryptographically verify said confirmation includes:

(a) determining that each of said leaf nodes corresponds to said candidate data item;

(b) cryptographically constructing at least one expected root node from said leaf nodes using said confirmation;

(c) verifying that the digital signature of said digitally signed root node is valid and that said at least one digitally signed root node corresponds to said expected root node.

69. The method of claim 68 wherein the step of cryptographically constructing said expected root node includes:

(a) identifying at least two of said nodes included in said confirmation;

This Sheet Has Been Replaced

(b) identifying additional of said nodes belonging to said confirmation, such that the combination of each identified additional node and at least one node previously identified node identifies a previously unidentified node, until a root node is obtained.

70. The method of claim 68 wherein the step of constructing said expected root node includes at least four cryptographic hashing operations.

71. The method of claim 67 wherein said at least one received leaf node defines a range including an identifier of said candidate data item, thereby specifying that said candidate data item does not belong to said plurality of data items.

72. The method of claim 67 wherein said at least one received leaf node represents one of said data items.

73. The method of claim 67 wherein said data items represent digital signatures.

74. The method of claim 73 wherein said data items represent digital certificates.

75. The method of claim 74 wherein said digital certificates are of revoked status.

This Sheet Has Been Replaced

76. The method of claim 67 further comprising the step of verifying the time of creation of said digitally signed root node.

77. A computer-based method of cryptographically verifying whether a candidate data item belongs to a plurality of data items, comprising the steps of:

(a) receiving, over an electronic communications channel, a cryptographically assured confirmation as to whether said candidate data item belongs to said plurality of data items, said confirmation including at least one leaf node, at least one intermediate node, and a digitally signed root node, and wherein said confirmation does not include all of said data items; and

(b) using said computer to cryptographically verify the validity of said confirmation.

78. A computer-based method of providing a cryptographically assured message that a candidate data item does not belong to a plurality of data items, for use in an electronic communications protocol, comprising the steps of:

(a) receiving a request identifying said candidate data item;

(b) confirming that said candidate data item does not belong to said plurality of data items; and

This Sheet Has Been Replaced

(c) digitally signing a message that said candidate data item does not belong to said plurality of data items, and

(d) communicating said message to a party wishing to know whether said candidate data item belongs to said plurality of data items.

79. The method of claim 78 wherein said data items represent revoked digital certificates.

80. A computer-based method of providing cryptographically assured ranges of data items, comprising the steps of:

(a) ordering said data items;

(b) selecting adjacent pairs of said ordered data items to form a plurality of ranges, each said range having no data items besides one of said pairs of data items;

(d) digitally signing at least one of said ranges; and

(e) communicating said at least one signed range to a party wishing to use said signed range as part of an electronic communications protocol for confirming that said range contains none of said data items.

81. The method of claim 80 wherein said data items represent revoked digital certificates.

82. A computer-based method of cryptographically confirming that a candidate data item

This Sheet Has Been Replaced

does not belong to a plurality of data items, for use in an electronic communications protocol, comprising the steps of:

(a) receiving a request identifying said candidate data item;

(b) obtaining a digitally signed range of data items, said range spanning said candidate data item but none of said plurality of data items; and

(c) communicating said signed range to a party wishing to know whether said candidate data item belongs to said plurality of data items.

83. The method of claim 82 wherein said data items represent revoked digital certificates.

84. A cryptographic confirmation, for use as part of an electronic communications protocol, as to whether a candidate data item belongs to a plurality of data items, said confirmation comprising a digitally signed message that said candidate data item does not belong to said plurality of data items.

85. The method of claim 84 wherein said data items represent revoked digital certificates.

86. A cryptographic confirmation, for use as part of an electronic communications protocol, that a candidate data item does not belong to a plurality of data items, said confirmation comprising a digitally signed range of data items, said range spanning said

This Sheet Has Been Replaced

candidate data item but containing none of said plurality of data items.

87. The method of claim 86 wherein said data items represent revoked digital certificates.

88. A computer-based method of cryptographically verifying that a candidate data item does not belong to a plurality of data items, comprising the steps of:

(a) receiving, over an electronic communications channel, a digitally signed range of data items, said range spanning said candidate data item but none of said plurality of data items; and

(b) cryptographically verifying that the digital signature of said digitally signed range is valid and that said digitally signed range spans said candidate data item.

89. The method of claim 88 wherein said data items represent revoked digital certificates.

This Sheet Has Been Replaced

FIG. 1

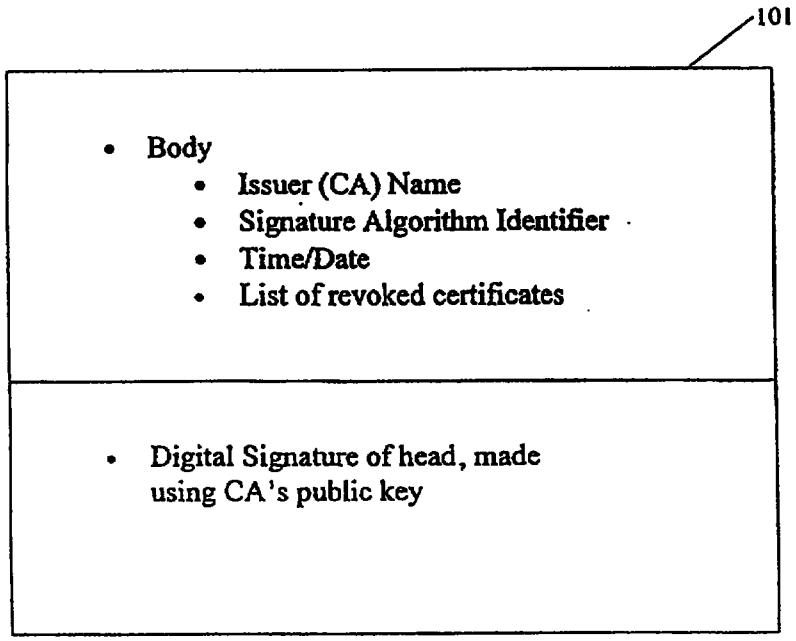

101
Body
Issuer (CA) Name
Signature Algorithm Identifier
Time/Date
List of revoked certificates
Digital Signature of head, made using CA's public key

FIG. 2

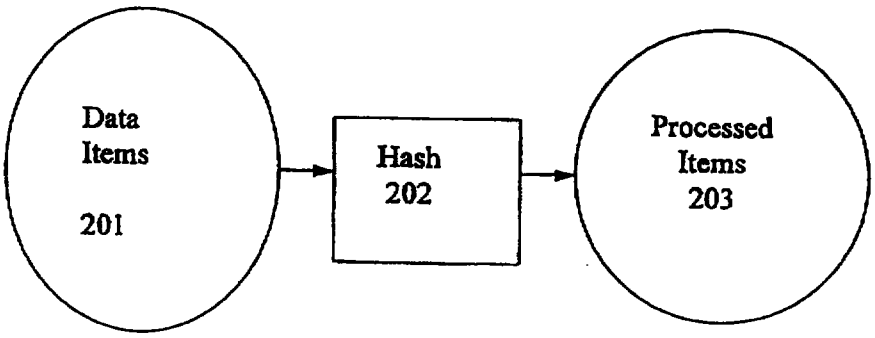


FIG. 3

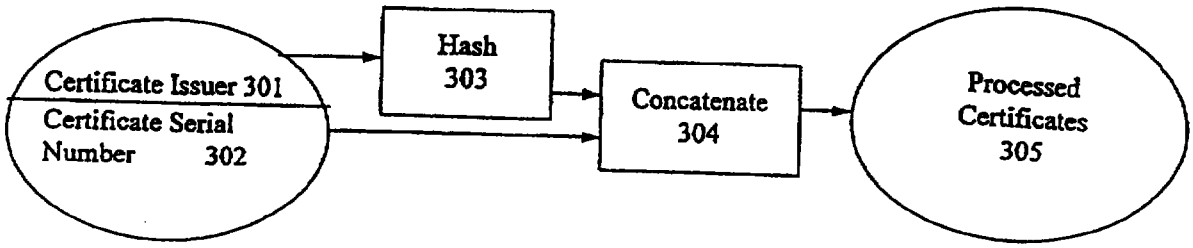

Certificate Issuer 301
Certificate Serial Number 302
Hash 303
Concatenate 304
Processed Certificates 305

FIG. 4

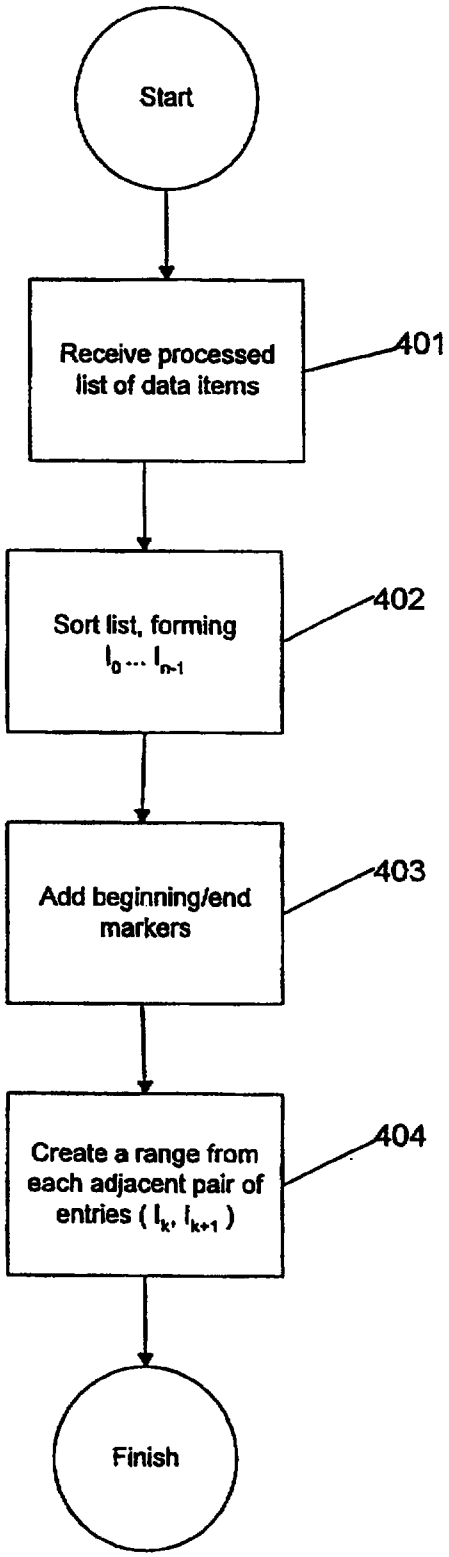

Start
Receive processed list of data items
401
Sort list, forming $I_0 \cdots I_{n-1}$
402
Add beginning/end markers
403
Create a range from each adjacent pair of entries (I_k, I_{k+1})
404
Finish

FIG. 5.

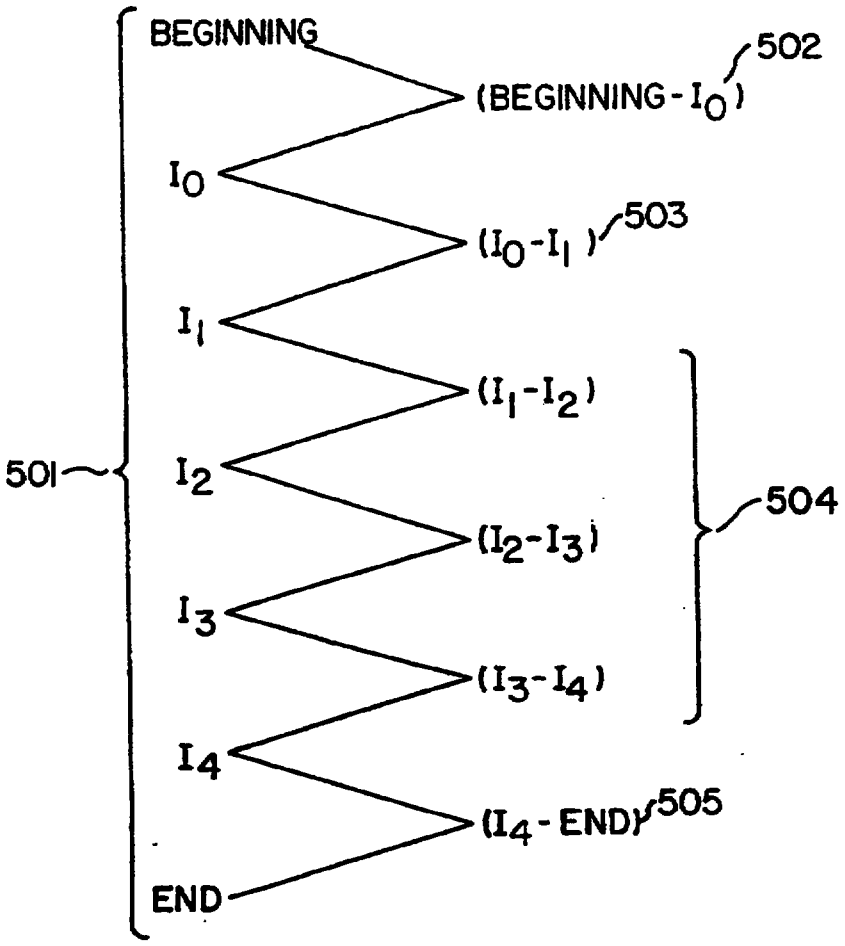

BEGINNING
(BEGINNING - I_0)
502
I_0
(I_0-I_1)
503
I_1
(I_1-I_2)
501
I_2
(I_2-I_3)
504
I_3
(I_3-I_4)
I_4
(I_4 - END)
505
END

FIG. 6.

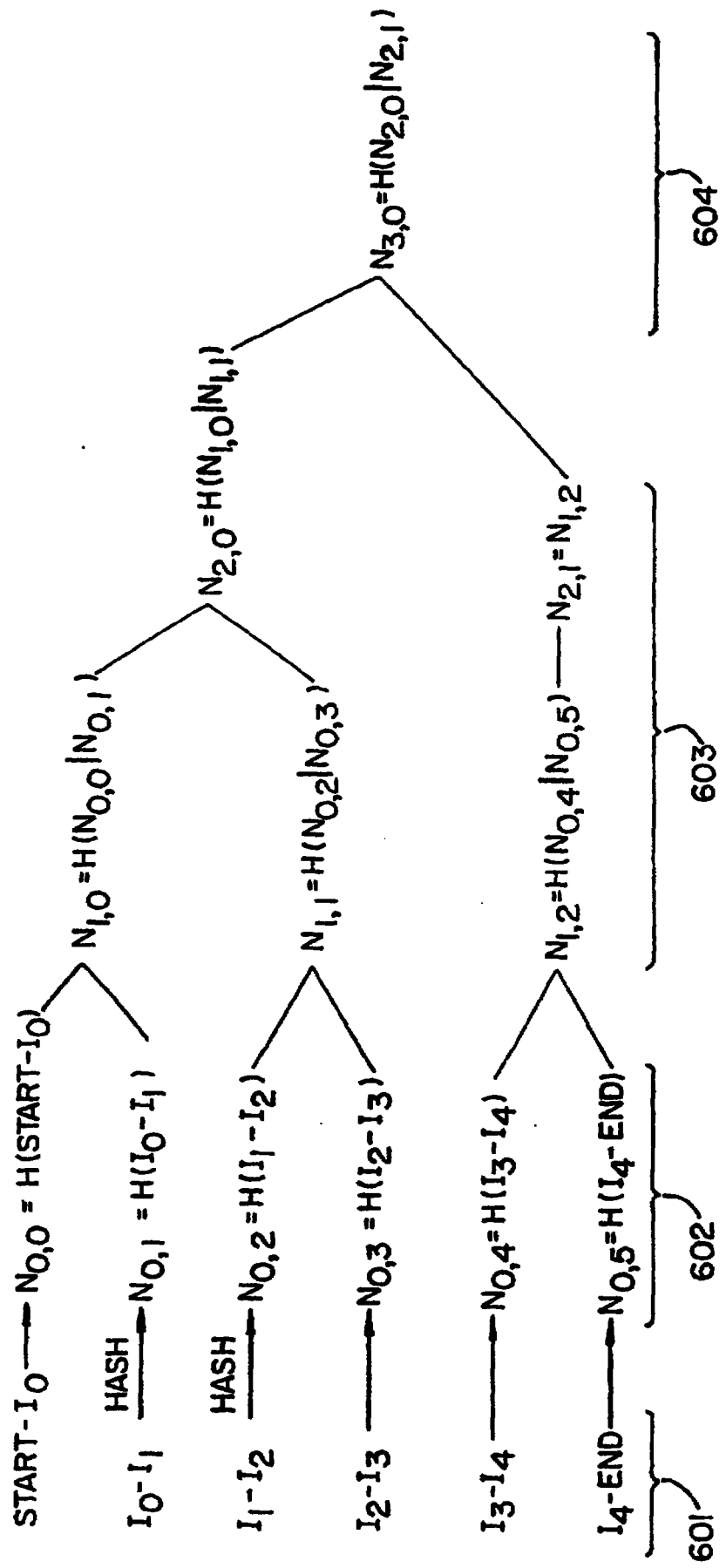

START-I0
N0,0 = H(START-I0)
I0-I1
HASH
N0,1 = H(I0-I1)
I1-I2
HASH
N0,2=H(I1-I2)
I2-I3
N0,3 =H(I2-I3)
I3-I4
N0,4=H(I3-I4)
I4-END
N0,5=H(I4-END)
N1,0=H(N0,0|N0,1)
N1,1=H(N0,2|N0,3)
N1,2=H(N0,4|N0,5)
N2,1=N1,2
N2,0=H(N1,0|N1,1)
N3,0=H(N2,0|N2,1)
601
602
603
604

FIG. 7

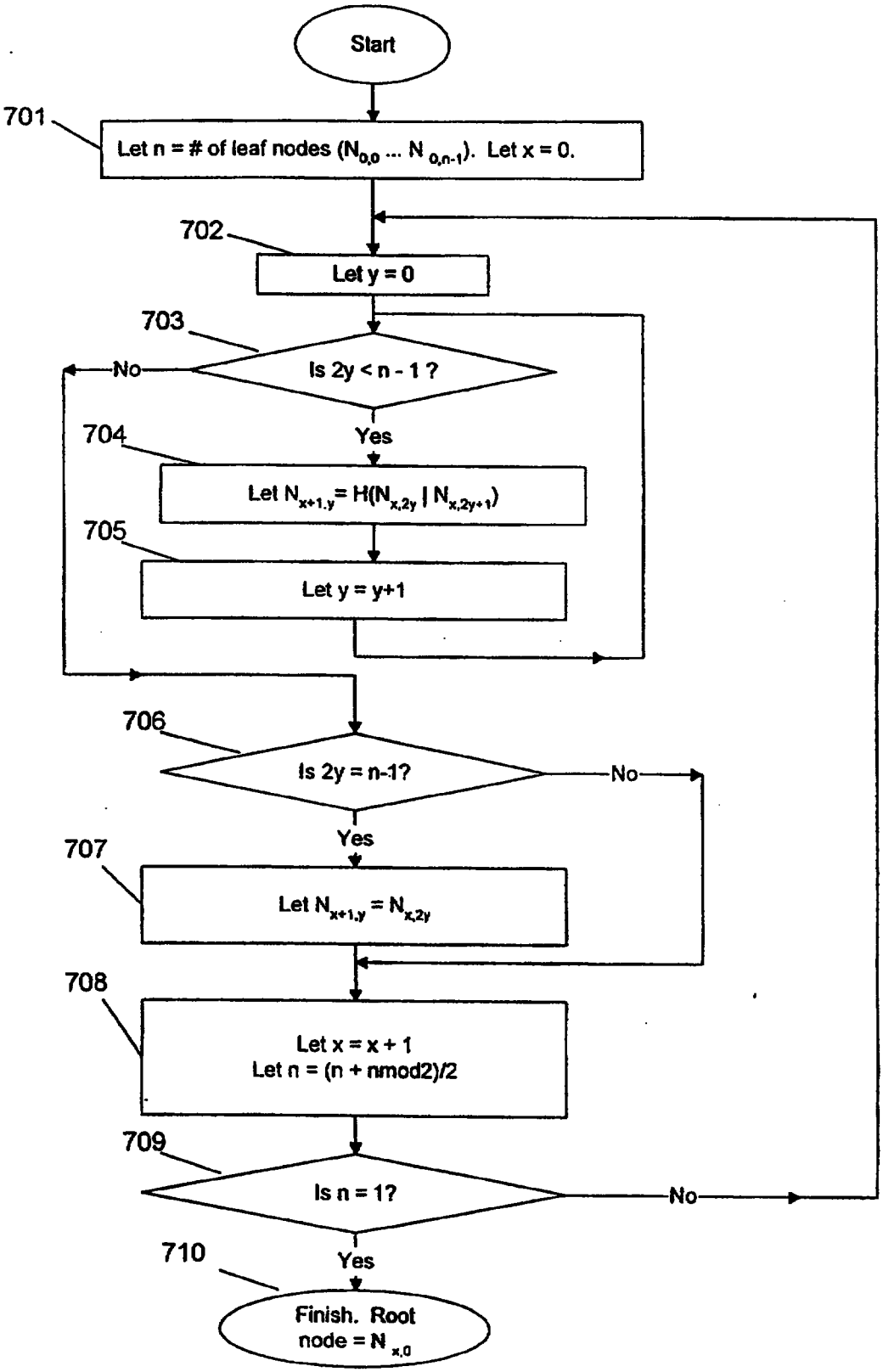

Start
701
Let n = # of leaf nodes ($N_{0,0}$ ... $N_{0,n-1}$). Let x = 0.
702
Let y = 0
703
Is 2y < n - 1 ?
No
Yes
704
Let $N_{x+1,y}$ = H($N_{x,2y}$ | $N_{x,2y+1}$)
705
Let y = y+1
706
Is 2y = n-1?
No
Yes
707
Let $N_{x+1,y}$ = $N_{x,2y}$
708
Let x = x + 1
Let n = (n + nmod2)/2
709
Is n = 1?
No
Yes
710
Finish. Root node = $N_{x,0}$

FIG. 8.

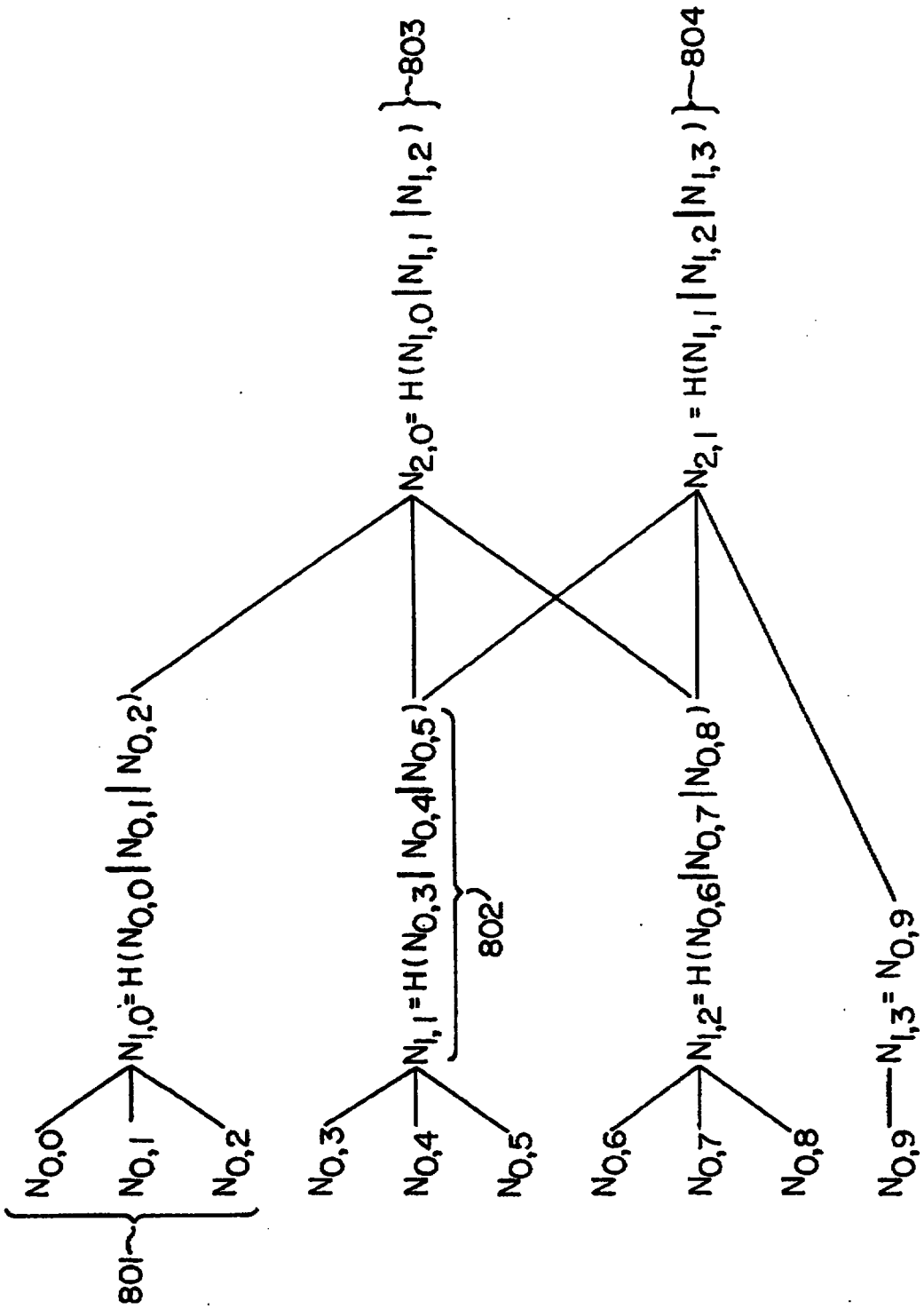

$N_{2,0}=H(N_{1,0}|N_{1,1}|N_{1,2})$
$N_{2,1}=H(N_{1,1}|N_{1,2}|N_{1,3})$
$N_{1,0}=H(N_{0,0}|N_{0,1}|N_{0,2})$
$N_{1,1}=H(N_{0,3}|N_{0,4}|N_{0,5})$
$N_{1,2}=H(N_{0,6}|N_{0,7}|N_{0,8})$
$N_{1,3}=N_{0,9}$
$N_{0,0}$ $N_{0,1}$ $N_{0,2}$ $N_{0,3}$ $N_{0,4}$ $N_{0,5}$ $N_{0,6}$ $N_{0,7}$ $N_{0,8}$ $N_{0,9}$
801
802
803
804

FIG 9

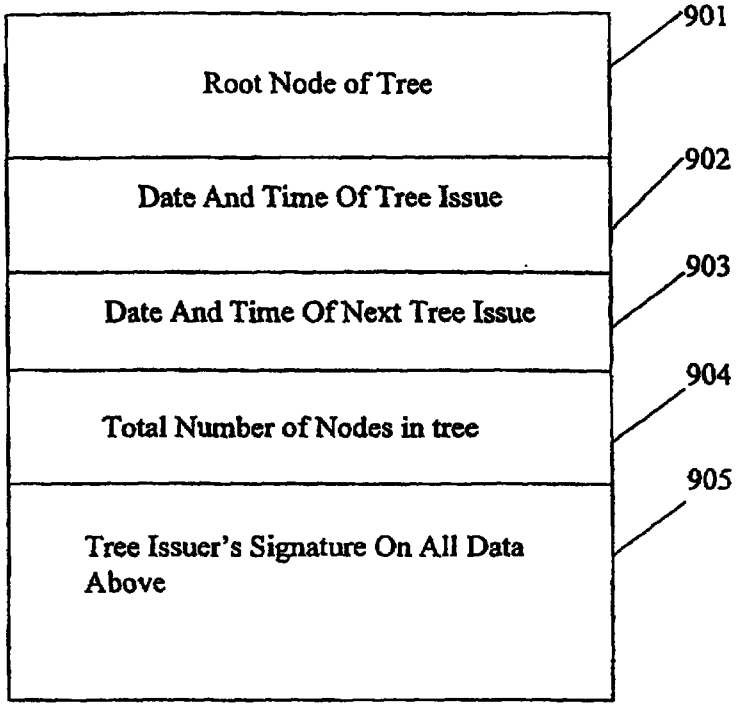

Root Node of Tree
Date And Time Of Tree Issue
Date And Time Of Next Tree Issue
Total Number of Nodes in tree
Tree Issuer's Signature On All Data Above
901
902
903
904
905

FIG. 10.

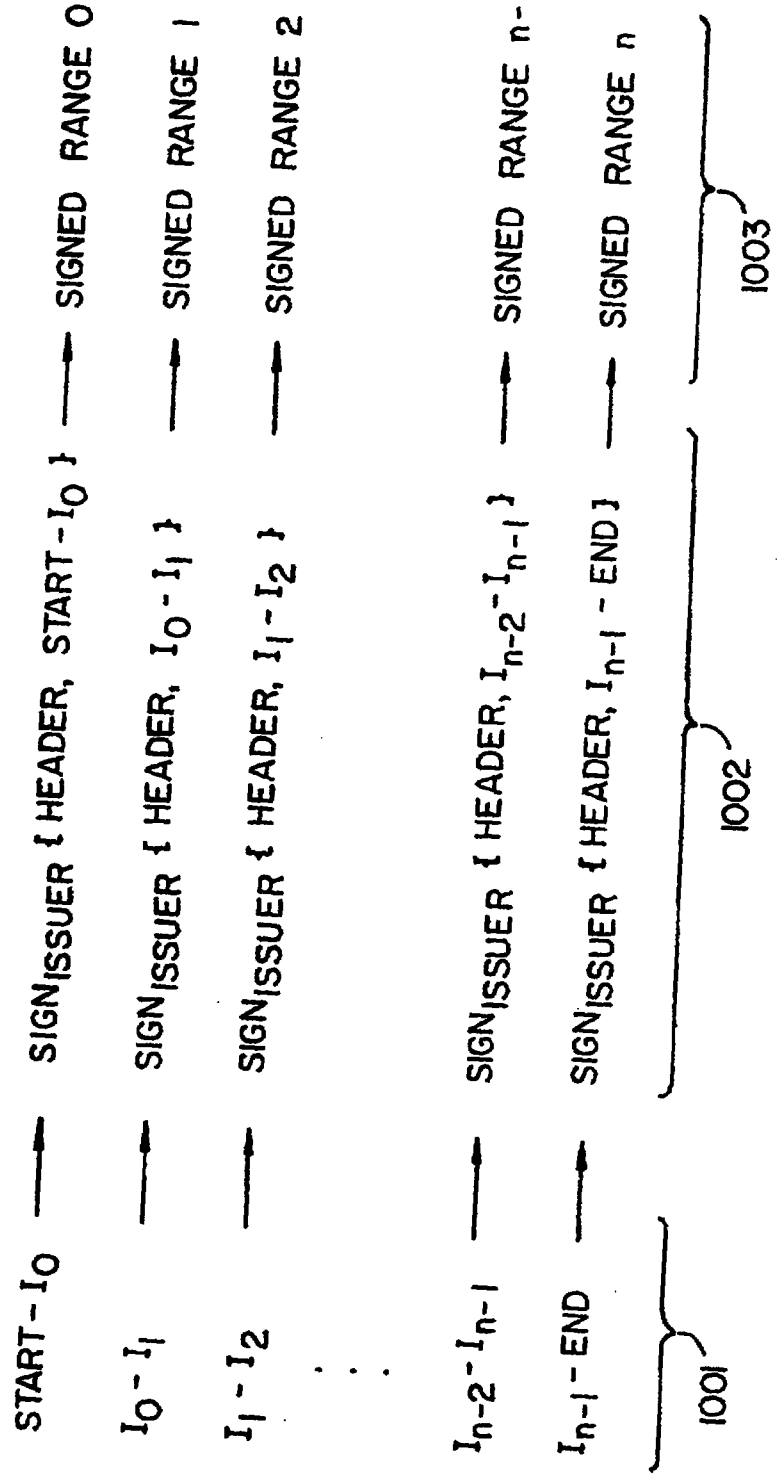

START - I_0
I_0 - I_1
I_1 - I_2
I_{n-2} - I_{n-1}
I_{n-1} - END
$SIGN_{ISSUER}$ { HEADER, START - I_0 } → SIGNED RANGE 0
$SIGN_{ISSUER}$ { HEADER, I_0 - I_1 } → SIGNED RANGE 1
$SIGN_{ISSUER}$ { HEADER, I_1 - I_2 } → SIGNED RANGE 2
$SIGN_{ISSUER}$ { HEADER, I_{n-2} - I_{n-1} } → SIGNED RANGE n-1
$SIGN_{ISSUER}$ { HEADER, I_{n-1} - END } → SIGNED RANGE n
1001
1002
1003

FIG. 11

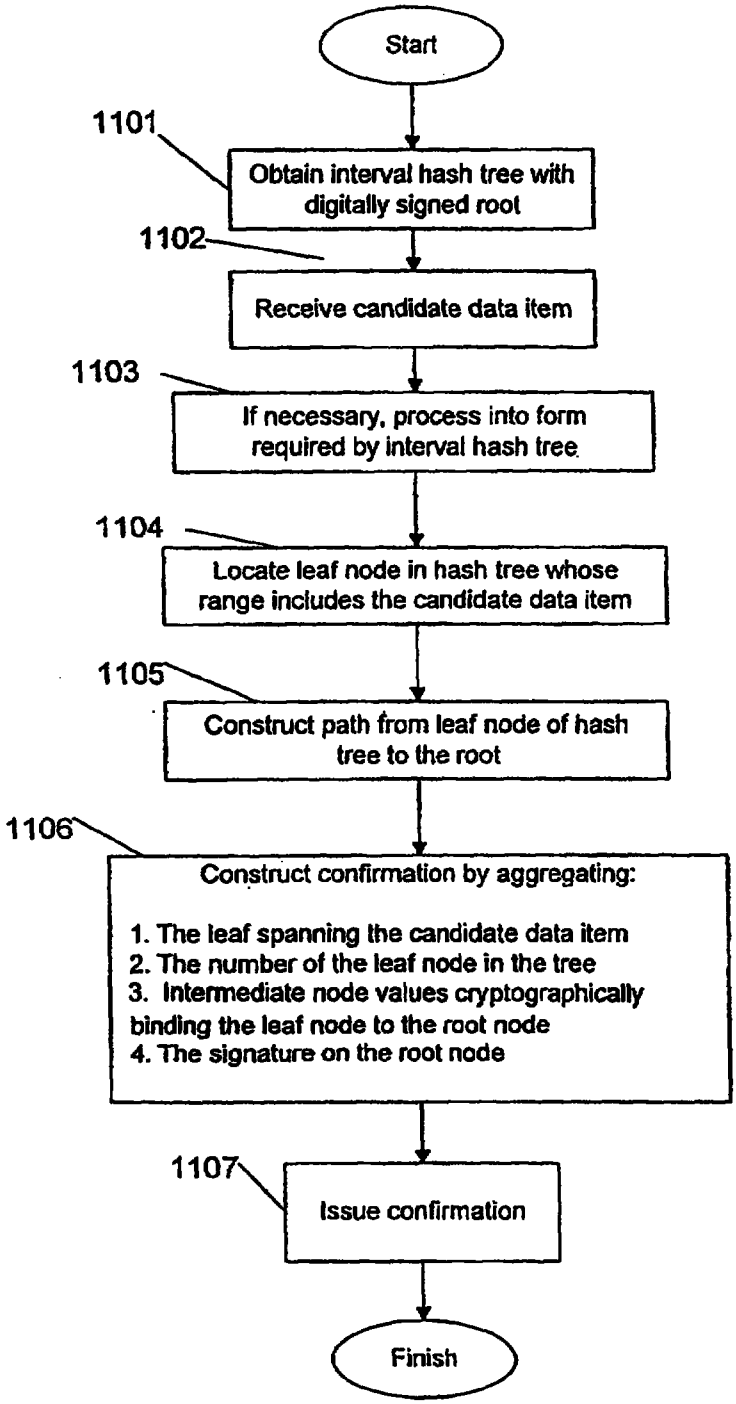

Start
1101
Obtain interval hash tree with digitally signed root
1102
Receive candidate data item
1103
If necessary, process into form required by interval hash tree
1104
Locate leaf node in hash tree whose range includes the candidate data item
1105
Construct path from leaf node of hash tree to the root
1106
Construct confirmation by aggregating:
1. The leaf spanning the candidate data item
2. The number of the leaf node in the tree
3. Intermediate node values cryptographically binding the leaf node to the root node
4. The signature on the root node
1107
Issue confirmation
Finish

FIG. 12

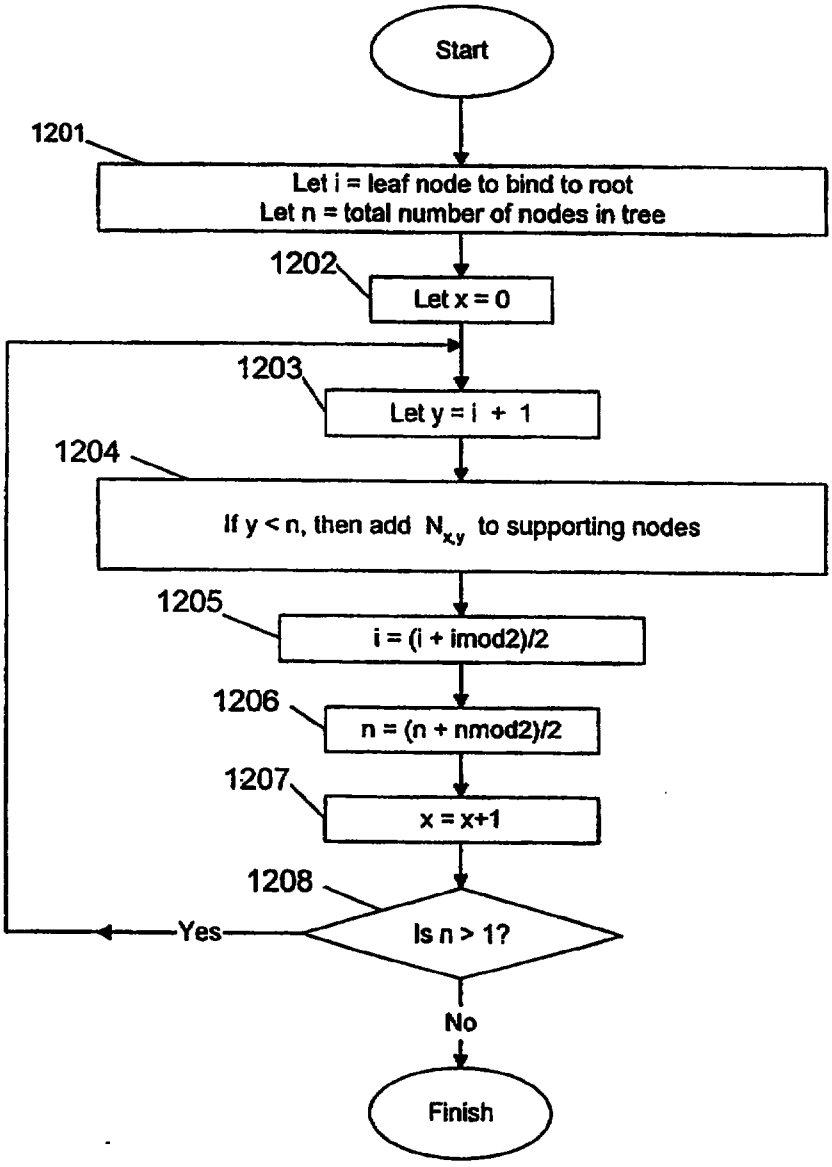

Start
1201
Let i = leaf node to bind to root
Let n = total number of nodes in tree
1202
Let x = 0
1203
Let y = i + 1
1204
If y < n, then add $N_{x,y}$ to supporting nodes
1205
i = (i + imod2)/2
1206
n = (n + nmod2)/2
1207
x = x+1
1208
Is n > 1?
Yes
No
Finish

FIG. 13

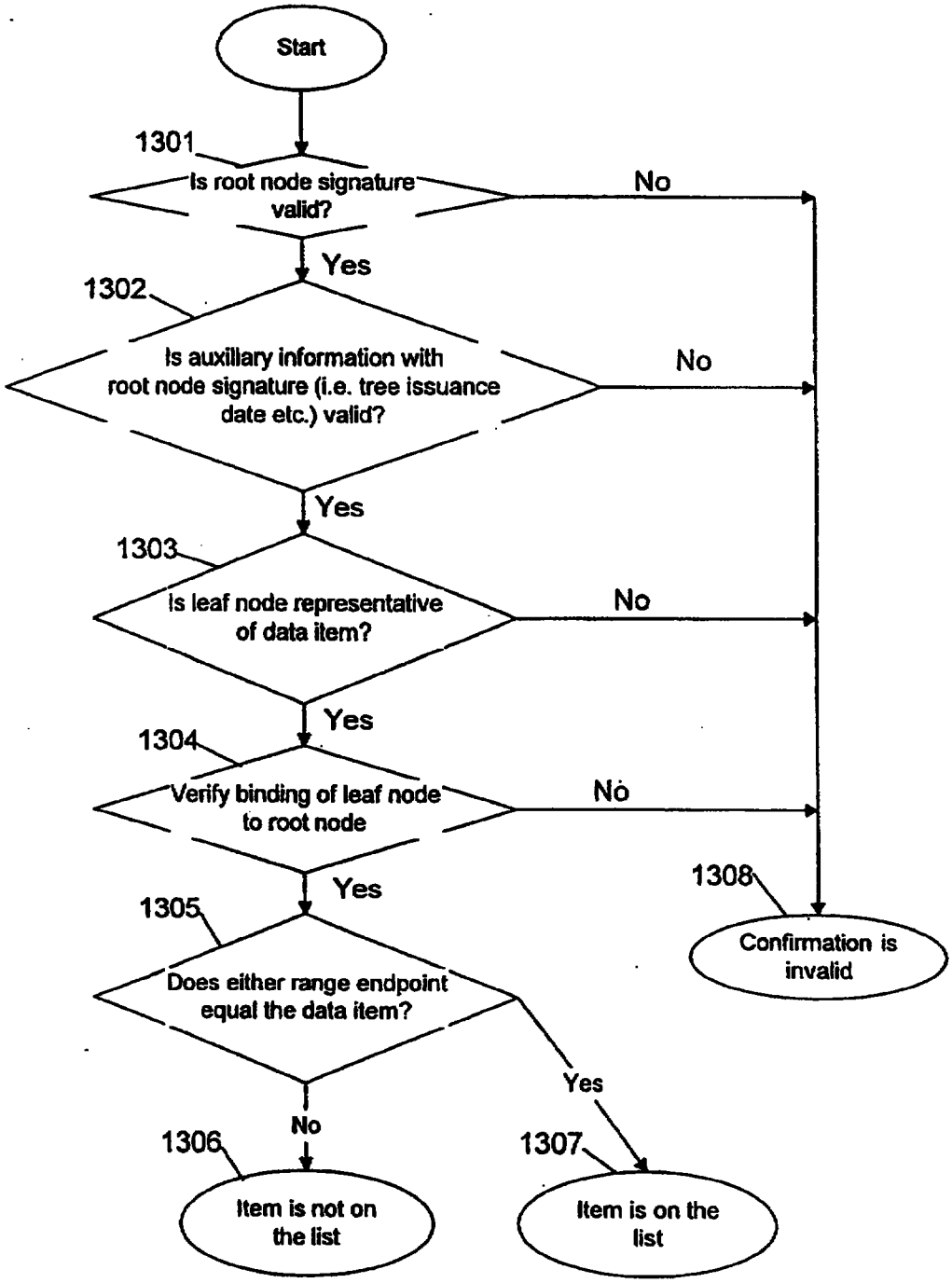

Start
1301
Is root node signature valid?
No
Yes
1302
Is auxillary information with root node signature (i.e. tree issuance date etc.) valid?
No
Yes
1303
Is leaf node representative of data item?
No
Yes
1304
Verify binding of leaf node to root node
No
Yes
1305
Does either range endpoint equal the data item?
Yes
No
1306
Item is not on the list
1307
Item is on the list
1308
Confirmation is invalid

FIG. 14

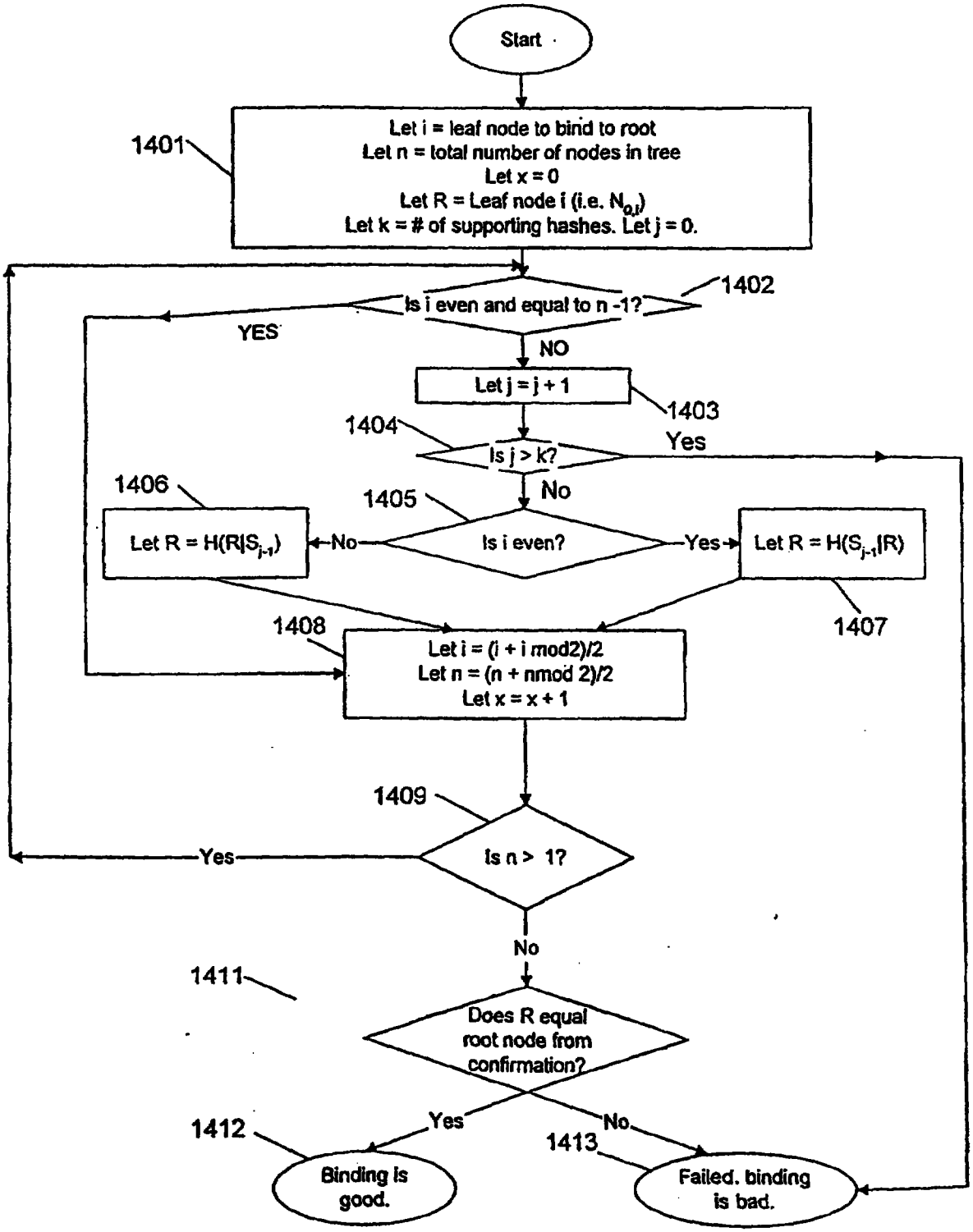
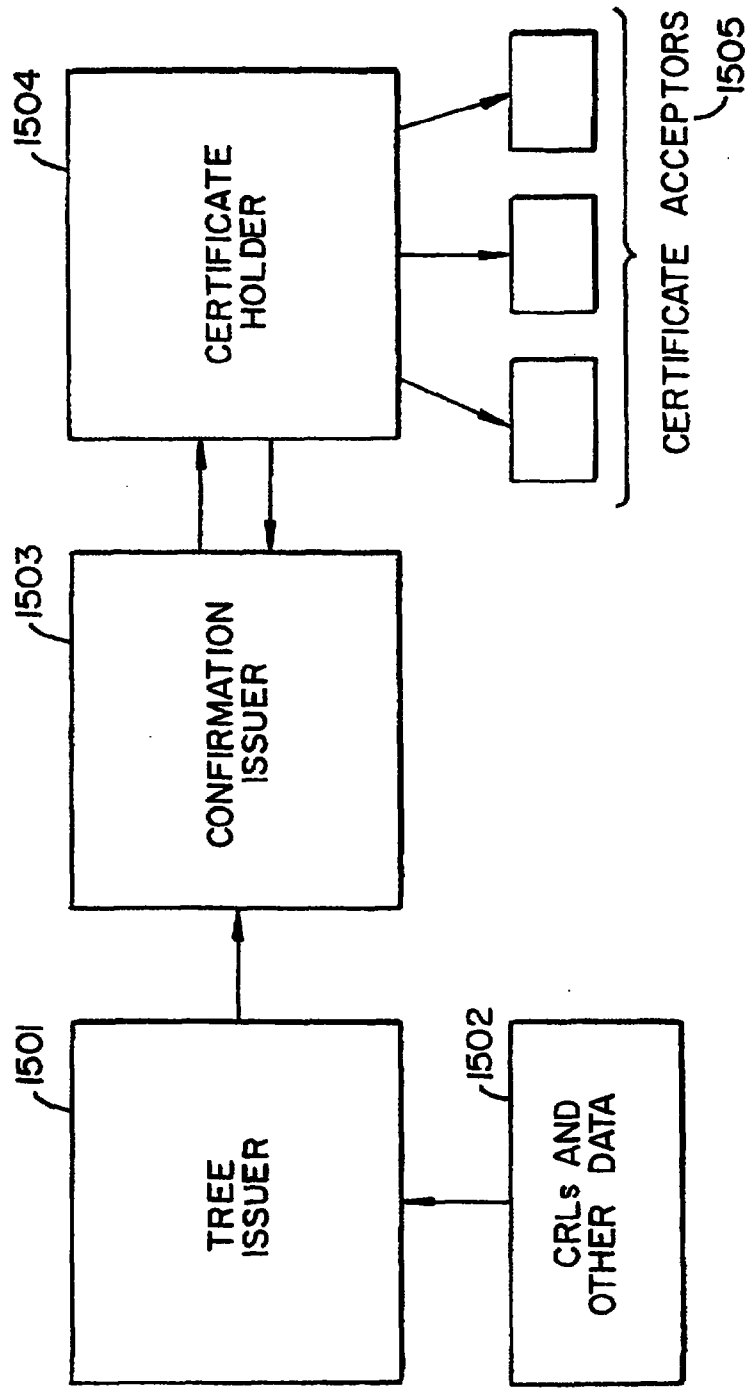

Start
1401
Let i = leaf node to bind to root
Let n = total number of nodes in tree
Let x = 0
Let R = Leaf node i (i.e. $N_{0,i}$)
Let k = # of supporting hashes. Let j = 0.
1402
Is i even and equal to n -1?
YES
NO
Let j = j + 1
1403
1404
Is j > k?
Yes
No
1406
1405
Let R = H(R|S_{j-1})
No
Is i even?
Yes
Let R = H(S_{j-1}|R)
1407
1408
Let i = (i + i mod2)/2
Let n = (n + nmod 2)/2
Let x = x + 1
1409
Is n > 1?
Yes
No
1411
Does R equal root node from confirmation?
Yes
No
1412
1413
Binding is good.
Failed. binding is bad.

FIG. 15.


1504
CERTIFICATE HOLDER
CERTIFICATE ACCEPTORS
1505
1503
CONFIRMATION ISSUER
1501
TREE ISSUER
1502
CRLs AND OTHER DATA

9805705-2
IPOS
REPLACEMENT SHEET
REPLACED SHEETS SG
☐ Description : paged ..... to paged .....
☒ Claim(s) : ~~paged~~ 01 to ~~paged~~ 26
☐ Drawing(s) : paged ..... to paged .....
☐ Abstract : paged ..... to paged .....
* C L M S *
New Claims
REPLACEMENT SHEET
29 OCT 1999

1. A method comprising the computer implemented steps of:

sorting a plurality of data items belonging to a superset of data items;

deriving a plurality of ranges from adjacent pairs of data items in said sorted plurality of data items;

generating cryptographically manipulated data from the plurality of ranges that cryptographically demonstrates whether any given data item is one of said plurality of data items; and

electronically transmitting at least parts of said cryptographically manipulated data onto a network for use in cryptographically demonstrating whether any given data item is one of said plurality of data items.

2. The method of claim 1, wherein said step of generating cryptographically manipulated data includes the step of digitally signing each of the plurality of ranges.

3. The method of claim 1, wherein said step of generating cryptographically manipulated data includes the steps of:

generating a tree having leaf nodes that represent the plurality of ranges; and

digitally signing a root node of the tree.

4. The method of claim 3, wherein said step of deriving the plurality of ranges includes the step of selecting said adjacent pairs of data items as endpoints.

5. The method of claim 1, wherein said step of deriving the plurality of ranges includes the step of selecting endpoints such that the plurality of ranges span the values

This Sheet Has Been Replaced

* C L M S *

REPLACEMENT SHEET

29 OCT 1999

REPLACEMENT SHEET

between said adjacent pairs of data items but do not include said adjacent pairs of data items.

6. The method of claim 1, wherein said plurality of data items identify digital certificates that are revoked.

7. A method comprising the computer implemented steps of:

receiving a request message requesting whether a first data item is one of a plurality of data items belonging to a superset of data items;

selecting a range that is derived from the pair of data items in said plurality of data items that defines the smallest range that includes said first data item;

forming a response message including cryptographically manipulated data derived from said range, said response message cryptographically demonstrating whether said first data item is one of said plurality of data items; and

electronically transmitting the response message onto a network.

8. The method of claim 7, wherein:

ranges are derived from adjacent pairs of data items in a sorted list of said plurality of data items; and

said step of forming includes the step of:

including a digitally signed representation of the selected range in said response message.

This Sheet Has Been Replaced

* C L M S *

REPLACEMENT SHEET
29 OCT 1999

9. The method of claim 7, wherein said step forming includes the steps of:

determining, for a tree having leaf nodes that represent ranges derived from said plurality of data items, a path through said tree from said selected range to a first of a set of root nodes; and

generating said response message based upon said path.

10. The method of claim 9, wherein said step of generating said response message based upon said path includes the steps of:

including data identifying said selected range in said response message;

including a set of nodes in said tree that cryptographically bind the selected range to the first root node; and

including a digitally signed representation of the first root node in said response message.

11. The method of claim 7, wherein said plurality of data items identify digital certificates that are revoked.

12. A method comprising the computer implemented steps of:

electronically transmitting a request message as to whether a first data item is one of a plurality of data items belonging to a superset of data items;

receiving a response message that cryptographically demonstrates whether said first data item is one of said plurality of data items, said response message identifying a range that is derived from the pair of data items in said plurality of data items that defines the smallest range that includes said first data item; and

This Sheet Has Been Replaced

* C L M S *

REPLACEMENT SHEET

29 OCT 1999

REPLACEMENT SHEET

determining if said first data item is one of said plurality of data items based on said range.

13. The method of claim 12, wherein:

said range is derived from an adjacent pair of data items in a sorted list of said plurality of data items; and

said response message includes a digitally signed representation of said range.

14. The method of claim 12, wherein said step of receiving the response message includes the steps of:

receiving a digitally signed representation submitted to be a root node of a tree having leaf nodes that represent ranges, the ranges derived from said plurality of data items;

receiving data identifying said range; and

receiving a set of nodes in said tree sufficient to generate said root node starting from said range.

15. The method of claim 12, wherein said plurality of data items identify digital certificates that are revoked.

16. A computer implemented method comprising:

selecting data that is derived from less than all the entries on a list identifying revoked digitally signed data items, wherein said data identifies whether at least a particular digitally signed data item has been revoked, and wherein said data is cryptographically manipulated in a manner that provides a

This Sheet Has Been Replaced

* C L M S *

REPLACEMENT SHEET

29 OCT 1999

REPLACEMENT SHEET

relative level of assurance to third parties that said data was supplied by an entity responsible for supplying revocation status information;

transmitting said data to an owner of the particular digitally signed data item;

the owner transmitting said data to a plurality of third parties; and

said plurality of third parties confirming whether the particular digitally signed data item has been revoked using said data.

17. The method of claim 16, further comprising:

selecting an expiration indicator that identifies when the representation of revocation status identified by the data expires;

transmitting said expiration indicator with said data to said owner;

the owner transmitting said expiration indicator with said data to said plurality of third parties; and

said plurality of third parties checking the representation of revocation status identified by said data has not expired using said expiration indicator.

18. The method of claim 16, wherein said digitally signed data items are digital certificates.

19. The method of claim 16, wherein said selecting said data comprises:

selecting a range that is derived from the pair of entries on said list that defines the smallest range that includes said particular digitally signed data item; and

deriving said data from said range.

This Sheet Has Been Replaced

* C L M S *

REPLACEMENT SHEET

29 OCT 1999

REPLACEMENT SHEET

20. The method of claim 19, wherein said deriving said data from said range comprises:

selecting a digitally signed representation of the range as part of said data.

21. The method of claim 19, wherein said deriving said data from said range comprises:

determining, for a tree having leaf nodes that represent ranges derived from the entries on said list, a path through said tree from said selected range to a root node of said tree; and

deriving said data from said path.

22. A computer implemented method comprising:

a trusted party collecting certificate revocation lists from multiple certificate issuers;

said trusted party forming a single list from the certificate revocation lists;

said trusted party cryptographically manipulating the single list to form a single structure that cryptographically demonstrates whether any given digitally signed data item is identified by any one of said certificate revocation lists and that provides a relative level of assurance to other parties which do not have the entire single structure that the trusted party formed the single list; and

electronically transmitting at least part of the single structure onto a network.

23. The method of claim 22, wherein:

said method further includes receiving a plurality of request messages from other parties as to status information regarding digital certificates issued by the certificate issuers; and

This Sheet Has Been Replaced

* C L M S *

REPLACEMENT SHEET

REPLACEMENT SHEET

29 OCT 1999

said electronically transmitting includes transmitting, in response to each of the plurality of request messages, a response message that includes the part of the single structure required to respond to that request message.

24. The method of claim 22, wherein a party with a pair of adjacent entries on a sorted version of the single list and only certain of the single structure that is associated with the pair of entries can verify the trusted party generated said single list and whether a plurality of digital certificates are on any of said certificate revocation lists.

25. The method of claim 22, wherein said trusted party cryptographically manipulating the single list to form the single structure comprises:

deriving from said single list a tree having at least one root node; and

digitally signing the at least one root node.

26. The method of claim 25, wherein said trusted party forming comprises:

inserting markers at the beginning and end of each region of the single sorted list corresponding to each trusted data item issuer.

This Sheet Has Been Replaced

980505-2 1902
REPLACEMENT SHEET
REPLACEMENT SHEETS SG
☐ Description : paged .... to paged ....
☒ Claim(s) : paged 28 to paged 34
☐ Drawing(s) : paged .... to paged ....
☐ Abstract : paged .... to paged ....
DATE OF RECEIPT 11/04/2000

New Claims

1. A method comprising the computer implemented steps of:

sorting a plurality of data items belonging to a superset of data items;

deriving a plurality of ranges from adjacent pairs of data items in said sorted plurality of data items;

generating cryptographically manipulated data from the plurality of ranges that cryptographically demonstrates whether any given data item is one of said plurality of data items; and

electronically transmitting at least parts of said cryptographically manipulated data onto a network for use in cryptographically demonstrating whether any given data item is one of said plurality of data items.

2. The method of claim 1, wherein said step of generating cryptographically manipulated data includes the step of digitally signing each of the plurality of ranges.

3. The method of claim 1, wherein said step of generating cryptographically manipulated data includes the steps of:

generating a tree having leaf nodes that represent the plurality of ranges; and

digitally signing a root node of the tree.

4. The method of claim 3, wherein said step of deriving the plurality of ranges includes the step of selecting said adjacent pairs of data items as endpoints.

5. The method of claim 1, wherein said step of deriving the plurality of ranges includes the step of selecting endpoints such that the plurality of ranges span the values

REPLACEMENT SHEET
11 APR 2000

between said adjacent pairs of data items but do not include said adjacent pairs of data items.

6. The method of claim 1, wherein said plurality of data items identify digital certificates that are revoked.

7. A method comprising the computer implemented steps of:

receiving a request message requesting whether a first data item is one of a plurality of data items belonging to a superset of data items;

selecting a range that is derived from the pair of data items in said plurality of data items that defines the smallest range that includes said first data item;

forming a response message including cryptographically manipulated data derived from said range, said response message cryptographically demonstrating whether said first data item is one of said plurality of data items; and

electronically transmitting the response message onto a network.

8. The method of claim 7, wherein:

ranges are derived from adjacent pairs of data items in a sorted list of said plurality of data items; and

said step of forming includes the step of:

including a digitally signed representation of the selected range in said response message.

REPLACEMENT SHEET

11 APR 2000

9. The method of claim 7, wherein said step forming includes the steps of:

determining, for a tree having leaf nodes that represent ranges derived from said plurality of data items, a path through said tree from said selected range to a first of a set of root nodes; and

generating said response message based upon said path.

10. The method of claim 9, wherein said step of generating said response message based upon said path includes the steps of:

including data identifying said selected range in said response message;

including a set of nodes in said tree that cryptographically bind the selected range to the first root node; and

including a digitally signed representation of the first root node in said response message.

11. The method of claim 7, wherein said plurality of data items identify digital certificates that are revoked.

12. A method comprising the computer implemented steps of:

electronically transmitting a request message as to whether a first data item is one of a plurality of data items belonging to a superset of data items;

receiving a response message that cryptographically demonstrates whether said first data item is one of said plurality of data items, said response message identifying a range that is derived from the pair of data items in said plurality of data items that defines the smallest range that includes said first data item; and

REPLACEMENT SHEET

11 APR 2000

* C L M S *

REPLACEMENT SHEET

determining if said first data item is one of said plurality of data items based on said range.

13. The method of claim 12, wherein:

said range is derived from an adjacent pair of data items in a sorted list of said plurality of data items; and

said response message includes a digitally signed representation of said range.

14. The method of claim 12, wherein said step of receiving the response message includes the steps of:

receiving a digitally signed representation submitted to be a root node of a tree having leaf nodes that represent ranges, the ranges derived from said plurality of data items;

receiving data identifying said range; and

receiving a set of nodes in said tree sufficient to generate said root node starting from said range.

15. The method of claim 12, wherein said plurality of data items identify digital certificates that are revoked.

16. A computer implemented method comprising:

selecting data that is derived from less than all the entries on a list identifying revoked digitally signed data items, wherein said data identifies whether at least a particular digitally signed data item has been revoked, and wherein said data is cryptographically manipulated in a manner that provides a

REPLACEMENT SHEET
11 APR 2000

relative level of assurance to third parties that said data was supplied by an entity responsible for supplying revocation status information;

transmitting said data to an owner of the particular digitally signed data item;

the owner transmitting said data to a plurality of third parties; and

said plurality of third parties confirming whether the particular digitally signed data item has been revoked using said data.

17. The method of claim 16, further comprising:

selecting an expiration indicator that identifies when the representation of revocation status identified by the data expires;

transmitting said expiration indicator with said data to said owner;

the owner transmitting said expiration indicator with said data to said plurality of third parties; and

said plurality of third parties checking the representation of revocation status identified by said data has not expired using said expiration indicator.

18. The method of claim 16, wherein said digitally signed data items are digital certificates.

19. The method of claim 16, wherein said selecting said data comprises:

selecting a range that is derived from the pair of entries on said list that defines the smallest range that includes said particular digitally signed data item; and

deriving said data from said range.

REPLACEMENT SHEET

11 APR 2000

20. The method of claim 19, wherein said deriving said data from said range comprises:

selecting a digitally signed representation of the range as part of said data.

21. The method of claim 19, wherein said deriving said data from said range comprises:

determining, for a tree having leaf nodes that represent ranges derived from the entries on said list, a path through said tree from said selected range to a root node of said tree; and

deriving said data from said path.

22. A computer implemented method comprising:

a trusted party collecting certificate revocation lists from multiple certificate issuers;

said trusted party forming a single list from the certificate revocation lists;

said trusted party cryptographically manipulating the single list to form a single structure that cryptographically demonstrates whether any given digitally signed data item is identified by any one of said certificate revocation lists and that provides a relative level of assurance to other parties which do not have the entire single structure that the trusted party formed the single list; and

electronically transmitting at least part of the single structure onto a network.

23. The method of claim 22, wherein:

said method further includes receiving a plurality of request messages from other parties as to status information regarding digital certificates issued by the certificate issuers; and

REPLACEMENT SHEET

11 APR 2000

said electronically transmitting includes transmitting, in response to each of the plurality of request messages, a response message that includes the part of the single structure required to respond to that request message.

24. The method of claim 22, wherein a party with a pair of adjacent entries on a sorted version of the single list and only certain of the single structure that is associated with the pair of entries can verify the trusted party generated said single list and whether a plurality of digital certificates are on any of said certificate revocation lists.

25. The method of claim 22, wherein said trusted party cryptographically manipulating the single list to form the single structure comprises:

deriving from said single list a tree having at least one root node; and

digitally signing the at least one root node.

26. The method of claim 25, wherein said trusted party forming comprises:

inserting markers at the beginning and end of each region of the single sorted list corresponding to each trusted data item issuer.

REPLACEMENT SHEET

11 APR 2000

9805705-2.

1105

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/32 — A1

(11) International Publication Number: WO 97/43842

(43) International Publication Date: 20 November 1997 (20.11.97)

(21) International Application Number: PCT/US97/08134

(22) International Filing Date: 13 May 1997 (13.05.97)

(30) Priority Data:
08/645,759 14 May 1996 (14.05.96) US

(71) Applicant: INTEGRIS SECURITY, INC. [US/US]; Suite 208, 1230 Oakmead Parkway, Sunnyvale, CA 94086 (US).

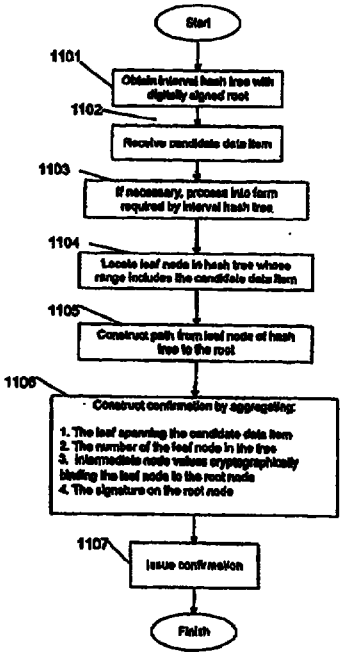
(72) Inventor: KOCHER, Paul, Carl; 2103 Menalto Avenue, Menlo Park, CA 94025 (US).

(74) Agent: LAURIE, Ronald, S.; McCutchen, Doyle, Brown & Enersen, Three Embarcadero Center, San Francisco, CA 94111 (US).

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

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: APPARATUS AND METHOD FOR DEMONSTRATING AND CONFIRMING THE STATUS OF DIGITAL CERTIFICATES AND OTHER DATA

(57) Abstract

Methods and apparatuses are disclosed for providing cryptographic assurance as to whether a particular data item is on a list. Typically the data item would be a digital certificate and the list would specify revoked certificates. The list of data items is typically converted into a set of ranges in which no data items exist, then the ranges are used as leaf nodes to construct a hash tree using a cryptographic hash function. The root node of the hash tree is digitally signed. A cryptographic confirmation whether a particular item is on the list consists of the range including the item in question, the hash tree nodes necessary to reconstruct the root from the range, and the digital signature on the root node. Verification of the confirmation involves testing the digital signature on the root node and checking that the root node can be constructed from a leaf node whose range includes the item in question.