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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 June 2008 (05.06.2008)

PCT

(10) International Publication Number
WO 2008/065344 A1

(51) International Patent Classification:
G06F 21/24 (2006.01)

(21) International Application Number:
PCT/GB2007/004427

(22) International Filing Date:
21 November 2007 (21.11.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

624061.8	1 December 2006 (01.12.2006)	GB
709764.5	22 May 2007 (22.05.2007)	GB

(71) Applicant and
(72) Inventor: **IRVINE, David** [GB/GB]; 82A Portland Street, Troon, Ayrshire KA10 6QU (GB).

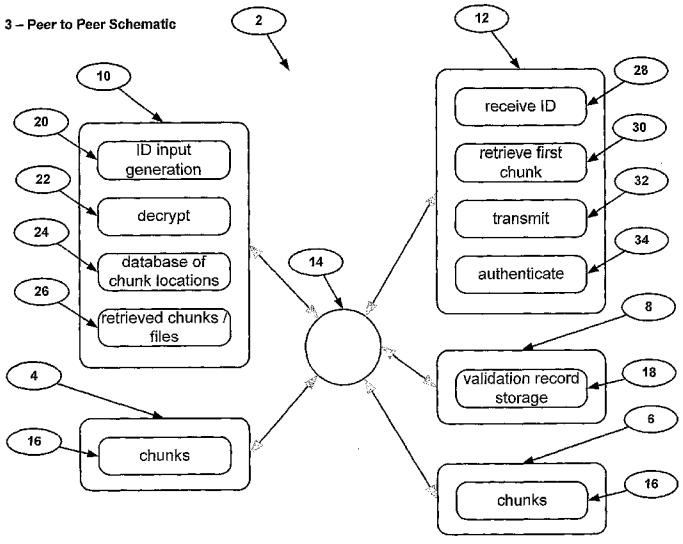
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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, LT, LU, LV, MC, MT, 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).

Declaration under Rule 4.17:
— *of inventorship (Rule 4.17(iv))*

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

(54) **Title:** ANONYMOUS AUTHENTICATION



(57) **Abstract:** This present invention relates to an authentication system for users requiring to be allowed access to resources stored on a distributed or peer-to-peer system. Information supplied during the authentication process by the user is extremely difficult to be compromised because these details are not stored on a central server and are not transmitted over the network during the process. In combination with this, information about the location of nodes is not stored together on the network. This obviates the opportunity for an unauthorised user to use that information to identify the location of resources in order to gain unauthorised access. It is the intention of this present invention to preserve the anonymity of the user and to provide a mechanism for secure access to private storage of data and preferably other resources on a distributed file system.

WO 2008/065344 A1

Anonymous Authentication

STATEMENT OF INVENTION:

Networks today rely on centralised servers or lists of account details and passwords; these are inherently insecure as they are a target in their own right for an attacker. Rather than strengthen security of such a target this invention removes the target altogether and provides a mechanism to self authenticate.

Another problem with today's authentication systems is that attackers target user IDs to obtain or test passwords against it. This invention removes this perception of a target by providing different user ID's on a frequent basis thereby giving the attacker (even if they possessed the password) no data element or clue to a data element that the password would be useful against.

Upon authentication this invention allows the user access to their resources as with traditional systems except in this case the user need not be known.

BACKGROUND:

AUTHENTICATION

Authentication servers are for user and data transaction authentication e.g. JP2005311545 which describe a system wherein the application of 'a digital seal' to electronic documents conforms to the Electronic Signature Act. This is similar to the case of signing paper documents but uses the application of an electronic signature through an electronic

seal authentication system. The system includes: client computers, to each of which a graphics tablet is connected; an electronic seal authentication server and a PKI authentication server, plus the electronic seal authentication server. US2004254894 discloses an automated system for the confirmed efficient authentication of an anonymous subscriber's profile data in this case.

JP2005339247 describes a server based one time ID system and uses a portable terminal. US2006136317 discloses bank drop down boxes and suggests stronger protection by not transmitting any passwords or IDs. Patent US2006126848 discloses a server centric and deals with a one time password or authentication phrase and is not for use on a distributed network. Patent US2002194484 discloses a distributed network where all chunks are not individually verified and where the manifest is only re-computed after updates to files and hashes are applied and are for validation only.

SELF-AUTHENTICATION

This is mostly used in biometric (WO2006069158). System for generating a patch file from an old version of data which consists of a series of elements and a new version of data which also consists of a series of elements US2006136514). Authentication servers (therefore not a distributed networking principle as per this invention) are commonly used (JP2006107316, US2005273603, EP1548979). However, server and client exchange valid certificates can be used (US2004255037). Instead of server, uses of information exchange system (semantic information) by participant for authentication can be used (JP2004355358), again this semantic information is stored and referenced unlike this present invention.

Concepts of identity-based cryptography and threshold secret sharing provides for a distributed key management and authentication. Without any assumption of pre-fixed trust relationship between nodes, the ad

hoc network works in a self-organizing way to provide the key generation and key management service, which effectively solves the problem of single point of failure in the traditional public key infrastructure (PKI)-supported system (US2006023887). Authenticating involves encryption keys for validation (WO2005055162) These are validated against known users unlike the present invention. Also, for authentication external housing are used (WO2005034009). All of these systems require a lost or (whether distributed or not) record of authorised users and pass phrases or certificates and therefore do not represent prior art.

Ranking, hashing for authentication can be implemented step-by-step and empirical authentication of devices upon digital authentication among a plurality of devices. Each of a plurality of authentication devices can unidirectionally generate a hash value of a low experience rank from a hash value of a high experience rank, and receive a set of high experience rank and hash value in accordance with an experience. In this way, the authentication devices authenticate each other's experience ranks (US2004019788). This is a system of hashing access against known identities and providing a mechanism of effort based access. This present invention does not rely or use such mechanisms.

QUICK ENCIPHERING

This is another method for authentication (JP2001308845). Self-verifying certificate for computer system, uses private and public keys – no chunking but for trusted hardware subsystems (US2002080973) this is a mec hanism of self signing certificates for authentication, again useful for effort based computing but not used in this present invention. Other authentication modes are, device for exchanging packets of information (JP2001186186), open key certificate management data (JP10285156), and certification for authentication (WO96139210). Authentication for Peer to Peer system is demonstrated by digital rights management (US2003120928). Digital rights management and CSC

(part of that patent s a DRM container) issues which are based on ability to use rather than gaining access to network or resources and therefore not prior art.

Known self-healing techniques are divided broadly into two classes. One is a centralized control system that provides overall rerouting control from the central location of a network. In this approach, the rerouting algorithm and the establishing of alarm collection times become increasingly complex as the number of failed channels increases, and a substantial amount of time will be taken to collect alarm signals and to transfer rerouting information should a large number of channels of a multiplexed transmission system fail. The other is a distributed approach in which the rerouting functions are provided by distributed points of the network. The following papers on distributed rerouting approach have been published: (these are all related to self healing but from a network pathway perspective and therefore are not prior art for this invention which deals with data or data chunks self healing mechanisms.

Document 1: W. D. Grover, "The Selfhealing Network", Proceedings of Grobecom '87, November 1987.

Document 2: H. C. Yang and S. Hasegawa, "Fitness: Failure Immunization Technology For Network Service Survivability", Proceedings of Globecom '88, December 1988.

Document 3: H. R. Amirazizi, "Controlling Synchronous Networks With Digital Cross-Connect Systems", Proceedings of Globecom '88, December 1988.

Document 1 is concerned with a restoration technique for failures in a single transmission system, and Document 2 relates to a "multiple-wave" approach in which route-finding packets are broadcast in multiple

wave fashion in search of a maximum bandwidth until alternate routes having the necessary bandwidth are established. One shortcoming of this multiple wave approach is that it takes a long recovery time. Document 3 also relates to fault recovery for single transmission systems and has a disadvantage in that route-finding packets tend to form a loop and hence a delay is likely to be encountered.

Summary of Invention

The main embodiments of this invention are as follows:

A system of anonymous authentication for data being stored or accessed within a distributed or peer to peer network.

An anonymous authentication product for data storage or access within a distributed or peer to peer network.

A system of anonymous authentication which has the functional elements of:

1. Validation
2. Provision of Key Pairs
3. Logon

A preferred system of claims 3, of anonymous authentication with functional linkages of:

1. Validation
    a. anonymity
    b. anonymous transaction
    c. peer ranking

2. Provision of Key Pairs
    a. provision of public ID
    b. document signing
    c. encryption/decryption

3. Logon

A product for anonymous authentication with functional linkages of;

a. Validation
b. Provision of Key Pairs
c. Logon

A product for anonymous authentication with functional linkages of;

a. Validation
    i. anonymity
    ii. anonymous transaction
    iii. peer ranking
b. Provision of Key Pairs
    i. provision of public ID
    ii. document signing
    iii. encryption/decryption
c. Logon

A method of above system and product of anonymous authentication for data storage or access in a distributed network or peer to peer network.

A method of above system and product of authenticating access to a distributed network comprising the steps of:

- creating a user identifier
- retrieving an encrypted validation record identified by the user

identifier

- decrypting the encrypted validation record so as to provide a decrypted result
- authenticating access to data in the distributed network using the decrypted result

A method of above wherein the steps of receiving, retrieving and authenticating are performed on a node in the distributed network separate from a node performing the step of decrypting.

A method of any of the above wherein the method further comprises the step of generating the user identifier using a hash of user input data.

A method of any previous claim wherein the user identifier is unique and suitable for identifying unique validation records.

A method of any previous claim wherein the step of authenticating access further comprises the step of digitally signing the user identifier.

A method of claim wherein the method further comprises the step of using the signed user identifier as a session passport to authenticate a plurality of accesses to the distributed network.

A method of any previous claim wherein the step of decrypting comprises decrypting an address in the distributed network of a first chunk of data and the step of authenticating access further comprises the step of determining the existence of the first chunk at the address.

A method of above where successful decryption of the ID chunk provides the user with a key pair to sign any requests and to consider the user as being authentic.

A method of above where successful decryption of the ID chunk provides a

data map of the data set of user's data and any keys associated with such.

A method of above wherein the method further comprises the step of using the content of the first chunk to obtain further chunks from the distributed network.

A method of any previous claim where the user ID is made unique using known changing data such as days since a time in history.

A system which provides a mechanism for self-authentication

A system which upon authentication allows the user access to their resources as with traditional systems except in this case the user need not be known.

A method of above where this variable information is merged or diluted into the user provided data prior to hashing or similar to produce a one time ID to further anonymity.

ESCRIPTION

Detailed Description:

(References to IDs used in descriptions of the system's functionality)

MID – this is the base ID and is mainly used to store and forget files. Each of these operations will require a signed request. Restoring may simply require a request with an ID attached.

PMID – This is the proxy mid which is used to manage the receiving of instructions to the node from any network node such as get/ put / forget etc. This is a key pair which is stored on the node – if stolen the key pair can be regenerated simply disabling the thief's stolen PMID – although there's not much can be done with a PMID key pair.

CID – Chunk Identifier, this is simply the chunkid.KID message on the net.

TMID – This is today's ID a one time ID as opposed to a one time password. This is to further disguise users and also ensure that their MID stays as secret as possible.

MPID – The maidsafe.net public ID. This is the ID to which users can add their own name and actual data if required. This is the ID for messenger, sharing, non anonymous voting and any other method that requires we know the user.

MAID – this is basically the hash of and actual public key of the MID. this ID is used to identify the user actions such as put / forget / get on the maidsafe.net network. This allows a distributed PKI infrastructure to exist and be automatically checked.

KID – Kademlia ID this can be randomly generated or derived from known and preferably anonymous information such as an anonymous public key hash as with the MAID.. In this case we use kademlia as the example overlay network although this can be almost any network environment at all.

MSID – maidsafe.net Share ID, an ID and key pair specifically created for each share to allow users to interact with shares using a unique key not related to their MID which should always be anonymous and separate.

Anonymous Authentication Description

Anonymous authentication relates to system authentication and, in particular, authentication of users for accessing resources stored on a distributed or peer-to-peer file system. Its aim is to preserve the anonymity of the users and to provide secure and private storage of data and shared resources for users on a distributed system. It is a method of authenticating access to a distributed system comprising the steps of;

- Receiving a user identifier;
- Retrieving an encrypted validation record identified by the user identifier;
- Decrypting the encrypted validation record so as to provide decrypted information; and ...
- Authenticating access to data in the distributed system using the decrypted information.

Receiving, retrieving and authenticating may be performed on a node in the distributed system preferably separate from a node performing the step of decrypting. The method further comprises the step of generating the user identifier using a hash. Therefore, the user identifier may be considered unique (and altered if a collision occurs) and suitable for

identifying unique validation records. The step of authenticating access may preferably further comprise the step of digitally signing the user identifier. This provides authentication that can be validated against trusted authorities. The method further comprises the step of using the signed user identifier as a session passport to authenticate a plurality of accesses to the distributed system. This allows persistence of the authentication for an extended session.

The step of decrypting preferably comprises decrypting an address in the distributed system of a first chunk of data and the step of authenticating access further comprises the step of determining the existence of the first chunk at the address, or providing the location and names of specific data elements in the network in the form of a data map as previously describe. This efficiently combines the tasks of authentication and starting to retrieve the data from the system. The method preferably further comprises the step of using the content of the first chunk to obtain further chunks from the distributed system. Additionally the decrypted data from the additional chunks may contain a key pair allowing the user at that stage to sign a packet sent to the network to validate them or additionally may preferable self sign their own id.

Therefore, there is no need to have a potentially vulnerable record of the file structure persisting in one place on the distributed system, as the user's node constructs its database of file locations after logging onto the system.

There is provided a distributed system comprising;

- a storage module adapted to store an encrypted validation record;
- a client node comprising a decryption module adapted to decrypt an encrypted validation record so as to provide decrypted information; and
- a verifying node comprising:

- a receiving module adapted to receive a user identifier;
- a retrieving module adapted to retrieve from the storage module an encrypted validation record identified by the user identifier;
- a transmitting module adapted to transmit the encrypted validation record to the client node; and
- an authentication module adapted to authenticate access to data in the distributed file system using the decrypted information from the client node.

The client node is further adapted to generate the user identifier using a hash. The authentication module is further adapted to authenticate access by digitally sign the user identifier. The signed user identifier is used as a session passport to authenticate a plurality of accesses by the client node to the distributed system. The decryption module is further adapted to decrypt an address in the distributed system of a first chunk of data from the validation record and the authentication module is further adapted to authenticate access by determining the existence of the first chunk at the address. The client node is further adapted to use the content of the first chunk to obtain further authentication chunks from the distributed system.

There is provided at least one computer program comprising program instructions for causing at least one computer to perform. One computer program is embodied on a recording medium or read-only memory, stored in at least one computer memory, or carried on an electrical carrier signal.

Additionally there is a check on the system to ensure the user is login into a valid node (software package). This will preferably include the ability of the system to check validity of the running maidsafe.net software by running content hashing or preferably certificate checking of the node and also the code itself.

Linked elements for Anonymous Authentication (Figure 1 – PT4)

The Anonymous Authentication invention consists of 3 key functional elements, with a further 6 functional elements being linked with.

The key functional elements are:
P12 – Logon
P13 – Provision of Key Pairs
P14 – Validation

The linked functional elements are:
P8 – Encryption / Decryption
P19 – Document Signing
P17 – Provision of Public ID
P1 – Peer Ranking
P24 – Anonymous Transactions
P25 – Anonymity

The anonymous authentication (PT4) itself is made up from linkage of elements, logon (P12) preferably provision of key pairs (P13) and validation (P14), to provide an anonymous authentication system for users requiring to be allowed access to resources stored on a distributed or peer-to-peer system and to preserve the anonymity of the user and to provide a mechanism for secure access to private storage of data and preferably other resources on a distributed file system. In addition, logon provision of key pairs element (P13) provides a sub-element provision of public ID (P17) which allows sub-element document signing (P19), and sub-element encryption/decryption (P8) where required; element validation is dependent on sub-element anonymity (P25) and provides sub-element anonymous transaction (P24) and is provisioned by sub-element peer ranking (P1).

Self Authentication Detail (Figure 2)

1. A computer program consisting of a user interface and a chunk server (a system to process anonymous chunks of data) should be running, if not they are started when user selects an icon or other means of starting the program.

2. A user will input some data known to them such as a userid (random ID) and PIN number in this case. These pieces of information may be concatenated together and hashed to create a unique (which may be confirmed via a search) identifier. In this case this is called the **MID** (maidsafe.net ID)

3. A TMID (Today's MID) is retrieved from the network, the TMID is then calculated as follows:

   The TMID is a single use or single day ID that is constantly changed. This allows maidsafe.net to calculate the hash based on the user ID pin and another known variable which is calculable. For this variable we use a day variable for now and this is the number of days since epoch (01/01/1970). This allows for a new ID daily, which assists in maintaining the anonymity of the user. This TMID will create a temporary key pair to sign the database chunks and accept a challenge response from the holder of these db chunks. After retrieval and generation of a new key pair the db is put again in new locations – rendering everything that was contained in the TMID chunk useless. The TMID CANNOT be signed by anyone (therefore hackers can't BAN an unsigned user from retrieving this – in a DOS attack)– it is a special chunk where the data hash does NOT match the name of the chunk (as the name is a random number calculated by hashing other information (i.e. its a hash of the TMID as described below)

- take dave as user ID and 1267 as pin.
- dave + (pin) 1267 = dave1267 Hash of this becomes MID
- day variable (say today is 13416 since epoch) = 13416
- so take pin, and for example add the number in where the pin states i.e.
- 613dav41e1267
- (6 at beginning is going round pin again)
- so this is done by taking 1st pin 1 - so put first day value at position 1
- then next pin number 2 - so day value 2 at position 2
- then next pin number 6 so day value 3 at position 6
- then next pin number 7 so day value 4 at position 7
- then next pin number is 1 so day value 5 at position 1 (again)
- so TMID is hash of 613dav41e1267 and the MID is simply a hash of dave1267

(This is an example algorithm and many more can be used to enforce further security.)

4. From the TMID chunk the map of the user's database (or list of files maps) is identified. The database is recovered from the net which includes the data maps for the user and any keys passwords etc.. The database chunks are stored in another location immediately and the old chunks forgotten. This can be done now as the MID key pair is also in the database and can now be used to manipulate user's data.

5. The maidsafe.net application can now authenticate itself as acting for this MID and put get or forget data chunks belonging to the user.

6. The watcher process and Chunk server always have access to the PMID key pair as they are stored on the machine itself, so can start and

receive and authenticate anonymous put / get / forget commands.

7. A DHT ID is required for a node in a DHT network this may be randomly generated or in fact we can use the hash of the PMID public key to identify the node.

8. When the users successfully logged in he can check his authentication validation records exist on the network. These may be as follows:

MAID (maidsafe.net anonymous ID)

1. This is a data element stored on net and preferably named with the hash of the MID public Key.

2. It contains the MID public key + any PMID public keys associated with this user.

3. This is digitally signed with the MID private key to prevent forgery.

4. Using this mechanism this allows validation of MID signatures by allowing any users access to this data element and checking the signature of it against any challenge response from any node pertaining to be this MID (as only the MID owner has the private key that signs this MID) Any crook could not create the private key to match to the public key to digitally sign so forgery is made impossible given today's computer resources.

5. This mechanism also allows a user to add or remove PMIDS (or chunk servers acting on their behalf like a proxy) at will and replace PMID's at any time in case of the PMID machine becoming compromised. Therefore this can be seen as the PMID authentication element.

PMID (Proxy MID)

1. This is a data element stored on the network and preferably named with the hash of the PMID public key.

2. It contains the PMID public key and the MID ID (i.e. the hash of the MID public key) and is signed by the MID private key (authenticated).

3. This allows a machine to act as a repository for anonymous chunks and supply resources to the net for a MID.

4. When answering challenge responses any other machine will confirm the PMID by seeking and checking the MIAD for the PMID and making sure the PMID is mentioned in the MAID bit – otherwise the PMID is considered rouge.

5. The key pair is stored on the machine itself and may be encoded or encrypted against a password that has to be entered upon start-up (optionally) in the case of a proxy provider who wishes to further enhance PMID security.

6. The design allows for recovery from attack and theft of the PMID key pair as the MAID data element can simply remove the PMID ID from the MAID rendering it unauthenticated.

Figure 3 illustrates, in schematic form, a peer-to-peer network in accordance with an embodiment of the invention; and

Figure 4 illustrates a flow chart of the authentication, in accordance with a preferred embodiment of the present invention.

With reference to Figure 3, a peer-to-peer network 2 is shown with nodes

4 to 12 connected by a communication network 14. The nodes may be Personal Computers (PCs) or any other device that can perform the processing, communication and/or storage operations required to operate the invention. The file system will typically have many more nodes of all types than shown in Figure 3 and a PC may act as one or many types of node described herein. Data nodes 4 and 6 store chunks 16 of files in the distributed system. The validation record node 8 has a storage module 18 for storing encrypted validation records identified by a user identifier.

The client node 10 has a module 20 for input and generation of user identifiers. It also has a decryption module 22 for decrypting an encrypted validation record so as to provide decrypted information, a database or data map of chunk locations 24 and storage 26 for retrieved chunks and files assembled from the retrieved chunks.

The verifying node 12 has a receiving module 28 for receiving a user identifier from the client node. The retrieving module 30 is configured to retrieve from the data node an encrypted validation record identified by the user identifier. Alternatively, in the preferred embodiment, the validation record node 8 is the same node as the verifying node 12, i.e. the storage module 18 is part of the verifying node 12 (not as shown in Figure 3). The transmitting module 32 sends the encrypted validation record to the client node. The authentication module 34 authenticates access to chunks of data distributed across the data nodes using the decrypted information.

With reference to Figure 4, a more detailed flow of the operation of the present invention is shown laid out on the diagram with the steps being performed at the User's PC (client node) on the left 40, those of the verifying PC (node) in the centre 42 and those of the data PC (node) on the right 44.

A login box is presented 46 that requires the user's name or other detail Preferably email address (the same one used in the client node software installation and registration process) or simply name (i.e. nickname) and the user's unique number, preferably PIN number. If the user is a 'main user' then some details may already be stored on the PC. If the user is a visitor, then the login box appears.

A content hashed number such as SHA (Secure Hash Algorithm), Preferably 160 bits in length, is created 48 from these two items of data. This 'hash' is now known as the 'User ID Key' (MID), which at this point is classed as 'unverified' within the system. This is stored on the network as the MAID and is simply the hash of the public key containing an unencrypted version of the public key for later validation by any other node. This obviates the requirement for a validation authority

The software on the user's PC then combines this MID with a standard 'hello' code element 50, to create 52 a 'hello.packet'. This hello.packet is then transmitted with a timed validity on the Internet.

The hello.packet will be picked up by the first node (for this description, now called the 'verifying node') that recognises 54 the User ID Key element of the hello.packet as matching a stored, encrypted validation record file 56 that it has in its storage area. A login attempt monitoring system ensures a maximum of three responses. Upon to many attempts, the verifying PC creates a 'black list' for transmission to peers. Optionally, an alert is returned to the user if a 'black list' entry is found and the user may be asked to proceed or perform a virus check.

The verifying node then returns this encrypted validation record file to the user via the internet. The user's pass phrase 58 is requested by a dialog box 60, which then will allow decryption of this validation record file.

When the validation record file is decrypted 62, the first data chunk

details, including a 'decrypted address', are extracted 64 and the user PC sends back a request 66 to the verifying node for it to initiate a query for the first 'file-chunk ID' at the 'decrypted address' that it has extracted from the decrypted validation record file, or preferably the data map of the database chunks to recreate the database and provide access to the key pair associated with this MID.

The verifying node then acts as a 'relay node' and initiates a 'notify only' query for this 'file-chunk ID' at the 'decrypted address'.

Given that some other node (for this embodiment, called the 'data node') has recognised 68 this request and has sent back a valid 'notification only' message 70 that a 'file-chunk ID' corresponding to the request sent by the verifying node does indeed exist, the verifying node then digitally signs 72 the initial User ID Key, which is then sent back to the user.

On reception by the user 74, this verified User ID Key is used as the user's session passport. The user's PC proceeds to construct 76 the database of the file system as backed up by the user onto the network. This database describes the location of all chunks that make up the user's file system. Preferably the ID Key will contain irrefutable evidence such as a public/private key pair to allow signing onto the network as authorised users, preferably this is a case of self signing his or her own ID – in which case the ID Key is decrypted and user is valid – self validating.

Further details of the embodiment will now be described. A 'proxy-controlled' handshake routine is employed through an encrypted point-to-point channel, to ensure only authorised access by the legal owner to the system, then to the user's file storage database, then to the files therein. The handshaking check is initiated from the PC that a user logs on to (the 'User PC'), by generating the 'unverified encrypted hash' known as the 'User ID Key', this preferably being created from the user's

information preferably email address and their PIN number. This 'hash' is transmitted as a `hello.packet' on the Internet, to be picked up by any system that recognises the User ID as being associated with specific data that it holds. This PC then becomes the 'verifying PC' and will initially act as the User PC's 'gateway' into the system during the authentication process. The encrypted item of data held by the verifying PC will temporarily be used as a 'validation record', it being directly associated with the user's identity and holding the specific address of a number of data chunks belonging to the user and which are located elsewhere in the peer-to-peer distributed file system. This 'validation record' is returned to the User PC for decryption, with the expectation that only the legal user can supply the specific information that will allow its accurate decryption.

Preferably this data may be a signed response being given back to the validating node which is possible as the id chunk when decrypted (preferably symmetrically) contains the users public and private keys allowing non refutable signing of data packets.

Preferably after successful decryption of the TMID packet (as described above) the machine will now have access to the data map of the database and public/private key pair allowing unfettered access to the system.

It should be noted that in this embodiment, preferably no communication is carried out via any nodes without an encrypted channel such as TLS (Transport Layer Security) or SSL (Secure Sockets Layer) being set up first. A peer talks to another peer via an encrypted channel and the other peer (proxy) requests the information (e.g. for some space to save information on or for the retrieval of a file). An encrypted link is formed between all peers at each end of communications and also through the proxy during the authentication process. This effectively bans snoopers from detecting who is talking to whom and also what is being sent or

retrieved. The initial handshake for self authentication is also over an encrypted link.

Secure connection is provided via certificate passing nodes, in a manner that does not require intervention, with each node being validated by another, where any invalid event or data, for whatever reason (fraud detection, snooping from node or any invalid algorithms that catch the node) will invalidate the chain created by the node. This is all transparent to the user.

Further modifications and improvements may be added without departing from the scope of the invention herein described.

Figure 5 illustrates a flow chart of data assurance event sequence in accordance with first embodiment of this invention

Figure 6 illustrates a flow chart of file chunking event sequence in accordance with second embodiment of this invention

Figure 7 illustrates a schematic diagram of file chunking example

Figure 8 illustrates a flow chart of self healing event sequence

Figure 9 illustrates a flow chart of peer ranking event sequence

Figure 10 illustrates a flow chart of duplicate removal event sequence

With reference to Figure 5, guaranteed accessibility to user data by data assurance is demonstrated by flow chart. The data is copied to at least three disparate locations at step (10). The disparate locations store data with an appendix pointing to the other two locations by step (20) and is renamed with hash of contents. Preferably this action is managed by another node i.e. super node acting as an intermediary by step (30).

Each local copy at user's PC is checked for validity by integrity test by step (40) and in addition validity checks by integrity test are made that the other 2 copies are also still ok by step (50).

Any single node failure initiates a replacement copy of equivalent leaf node being made in another disparate location by step (60) and the other remaining copies are updated to reflect this change to reflect the newly added replacement leaf node by step (70).

The steps of storing and retrieving are carried out via other network nodes to mask the initiator (30).

The method further comprises the step of renaming all files with a hash of their contents.

Therefore, each file can be checked for validity or tampering by running a content hashing algorithm such as (for example) MD5 or an SHA variant, the result of this being compared with the name of the file.

With reference to Figure 6, provides a methodology to manageable sized data elements and to enable a complimentary data structure for and compression and encryption and the step is to file chunking. By user's pre-selection the nominated data elements (files are passed to chunking process. Each data element (file) is split into small chunks by step (80) and the data chunks are encrypted by step (90) to provide security for the data. The data chunks are stored locally at step (100) ready for network transfer of copies. Only the person or the group, to whom the overall data belongs, will know the location of these (100) or the other related but dissimilar chunks of data. All operations are conducted within the user's local system. No data is presented externally.

Each of the above chunks does not contain location information for any

other dissimilar chunks. This provides for, security of data content, a basis for integrity checking and redundancy.

The method further comprises the step of only allowing the person (or group) to whom the data belongs, to have access to it, preferably via a shared encryption technique. This allows persistence of data.

The checking of data or chunks of data between machines is carried out via any presence type protocol such as a distributed hash table network.

On the occasion when all data chunks have been relocated (i.e. the user has not logged on for a while,) a redirection record is created and stored in the super node network, (a three copy process – similar to data) therefore when a user requests a check, the redirection record is given to the user to update their database.

This efficiently allows data resilience in cases where network churn is a problem as in peer to peer or distributed networks.

With reference to Figure 7 which illustrates flow chart example of file chunking. User's normal file has 5Mb document, which is chunked into smaller variable sizes e.g. 135kb, 512kb, 768kb in any order. All chunks may be compressed and encrypted by using Pass phrase. Next step is to individually hash chunks and given hashes as names. Then database record as a file is made from names of hashed chunks brought together e.g. in empty version of original file (C1########,t1,t2,t3: C2########,t1,t2,t3 etc), this file is then sent to transmission queue in storage space allocated to client application.

With reference to Figure 8 provides a self healing event sequence methodology. Self healing is required to guarantee availability of accurate data. As data or chunks become invalid by failing integrity test by step (110). The location of failing data chunks is assessed as unreliable and

further data from the leaf node is ignored from that location by step (120). A 'Good Copy' from the 'known good' data chunk is recreated in a new and equivalent leaf node. Data or chunks are recreated in a new and safer location by step (130). The leaf node with failing data chunks is marked as unreliable and the data therein as 'dirty' by step (140). Peer leaf nodes become aware of this unreliable leaf node and add its location to watch list by step (150). All operations conducted within the user's local system. No data is presented externally.

Therefore, the introduction of viruses, worms etc. will be prevented and faulty machines/ equipment identified automatically.

The network will use SSL or TLS type encryption to prevent unauthorised access or snooping.

With reference to Figure 9, Peer Ranking id required to ensure consistent response and performance for the level of guaranteed interaction recorded for the user. For Peer Ranking each node (leaf node) monitors its own peer node's resources and availability in a scaleable manner, each leaf node is constantly monitored.

Each data store (whether a network service, physical drive etc.) is monitored for availability. A qualified availability ranking is appended to the (leaf) storage node address by consensus of a monitoring super node group by step (160). A ranking figure will be appended by step (160) and signed by the supply of a key from the monitoring super node; this would preferably be agreed by more super nodes to establish a consensus for altering the ranking of the node. The new rank will preferably be appended to the node address or by a similar mechanism to allow the node to be managed preferably in terms of what is stored there and how many copies there has to be of the data for it to be seen as perpetual.

Each piece of data is checked via a content hashing mechanism for data

integrity, which is carried out by the storage node itself by step (170) or by its partner nodes via super nodes by step (180) or by instigating node via super nodes by step (190) by retrieval and running the hashing algorithm against that piece of data. The data checking cycle repeats itself.

As a peer (whether an instigating node or a partner peer (i.e. one that has same chunk)) checks the data, the super node querying the storage peer will respond with the result of the integrity check and update this status on the storage peer. The instigating node or partner peer will decide to forget this data and will replicate it in a more suitable location.

If data fails the integrity check the node itself will be marked as 'dirty' by step (200) and 'dirty' status appended to leaf node address to mark it as requiring further checks on the integrity of the data it holds by step (210). Additional checks are carried out on data stored on the leaf node marked as 'dirty' by step (220). If pre-determined percentage of data found to be 'dirty' node is removed from the network except for message traffic by step (230). A certain percentage of dirty data being established may conclude that this node is compromised or otherwise damaged and the network would be informed of this. At that point the node will be removed from the network except for the purpose of sending it warning messages by step (230).

This allows either having data stored on nodes of equivalent availability and efficiency or dictating the number of copies of data required to maintain reliability.

Further modifications and improvements may be added without departing from the scope of the invention herein described.

With reference to Figure 10, duplicate data is removed to maximise the efficient use of the disk space. Prior to the initiation of the data backup

process by step (240), internally generated content hash may be checked for a match against hashes stored on the internet by step (250) or a list of previously backed up data (250). This will allow only one backed up copy of data to be kept. This reduces the network wide requirement to backup data which has the exact same contents. Notification of shared key existence is passed back to instigating node by step (260) to access authority check requested, which has to pass for signed result is to be passed back to storage node. The storage node passes shared key and database back to instigating node by step (270)

Such data is backed up via a shared key which after proof of the file existing (260) on the instigating node, the shared key (270) is shared with this instigating node. The location of the data is then passed to the node for later retrieval if required.

This maintains copyright as people can only backup what they prove to have on their systems and not publicly share copyright infringed data openly on the network.

This data may be marked as protected or not protected by step (280) which has check carried out for protected or non-protected data content. The protected data ignores sharing process.

CLAIMS

1. A system that allows anonymous authentication access to a distributed system of; receiving a user identifier; retrieving an encrypted validation record identified by the user identifier; decrypting the encrypted validation record so as to provide decrypted information; and authenticating access to data in the distributed file system using the decrypted information to provide anonymous authentication, this system comprises of combination of following steps:
   a. Validation
   b. Provision of Key Pairs
   c. Logon

   the above combination provides a unique system with cumulative and synergistic benefits to allow anonymous authentication of users

2. A preferred system of claims 1, that allows anonymous authentication access to a distributed system of; receiving a user identifier; retrieving an encrypted validation record identified by the user identifier; decrypting the encrypted validation record so as to provide decrypted information; and authenticating access to data in the distributed file system using the decrypted information to provide anonymous authentication, this system comprises of combination of following steps;
   a. Validation, which further comprises of anonymity, anonymous transaction and peer ranking
   b. Provision of Key Pairs, which further comprises of provision of public ID, document signing and encryption/decryption
   c. Logon

   the above combination provides a unique system with cumulative and synergistic benefits to allow anonymous authentication of users

3. A product that allows anonymous authentication access to a distributed system of; receiving a user identifier; retrieving an encrypted validation record identified by the user identifier; decrypting the encrypted

validation record so as to provide decrypted information; and authenticating access to data in the distributed file system using the decrypted information to provide anonymous authentication, this product comprises of combination of following steps:

a. Validation
b. Provision of Key Pairs
c. Logon

the above combination provides a unique product with cumulative and synergistic benefits to allow anonymous authentication of users

4. A preferred product of claims 3, that allows anonymous authentication access to a distributed system of; receiving a user identifier; retrieving an encrypted validation record identified by the user identifier; decrypting the encrypted validation record so as to provide decrypted information; and authenticating access to data in the distributed file system using the decrypted information to provide anonymous authentication, this product comprises of combination of following steps;

d. Validation, which further comprises of anonymity, anonymous transaction and peer ranking
e. Provision of Key Pairs, which further comprises of provision of public ID, document signing and encryption/decryption
f. Logon

the above combination provides a unique product with cumulative and synergistic benefits to allow anonymous authentication of users

5. A method of claims 1 to 4, for authenticating access to a distributed network comprising the steps of;

a. creating a user identifier;
b. retrieving an encrypted validation record identified by the user identifier;
c. decrypting the encrypted validation record to provide a decrypted result;
d. authenticating access to data in the distributed network using the

decrypted result.

6. A method of claim 5 wherein the steps of receiving, retrieving and authenticating are performed on a node in the distributed network separate from a node performing the step of decrypting;

7. A method of any of claims 5, 6 wherein the method further comprises the step of generating the user identifier using a hash of user input data;

8. A method of any previous claims wherein the user identifier is unique and suitable for identifying unique validation records;

9. A method of any previous claims wherein the step of authenticating access further comprises the step of digitally signing the user identifier;

10. A method of claim 9 wherein the method further comprises the step of using the signed user identifier as a session passport to authenticate a plurality of accesses to the distributed network;

11. A method of any previous claims wherein the step of decrypting comprises decrypting an address in the distributed network of a first chunk of data and the step of authenticating access further comprises the step of determining the existence of the first chunk at the address;

12. A method of claims 1-2 where successful decryption of the ID chunk provides the user with a key pair to sign any requests and to consider the user as being authentic;

13. A method of claim 12 where successful decryption of the ID chunk provides a data map of the data set of user's data and preferably any keys associated with such;

14. A method of claim 13 wherein the method further comprises the step of

using the content of the first chunk to obtain further chunks from the distributed network;

15. A method of claim 13 which provides mechanism for self-authentication

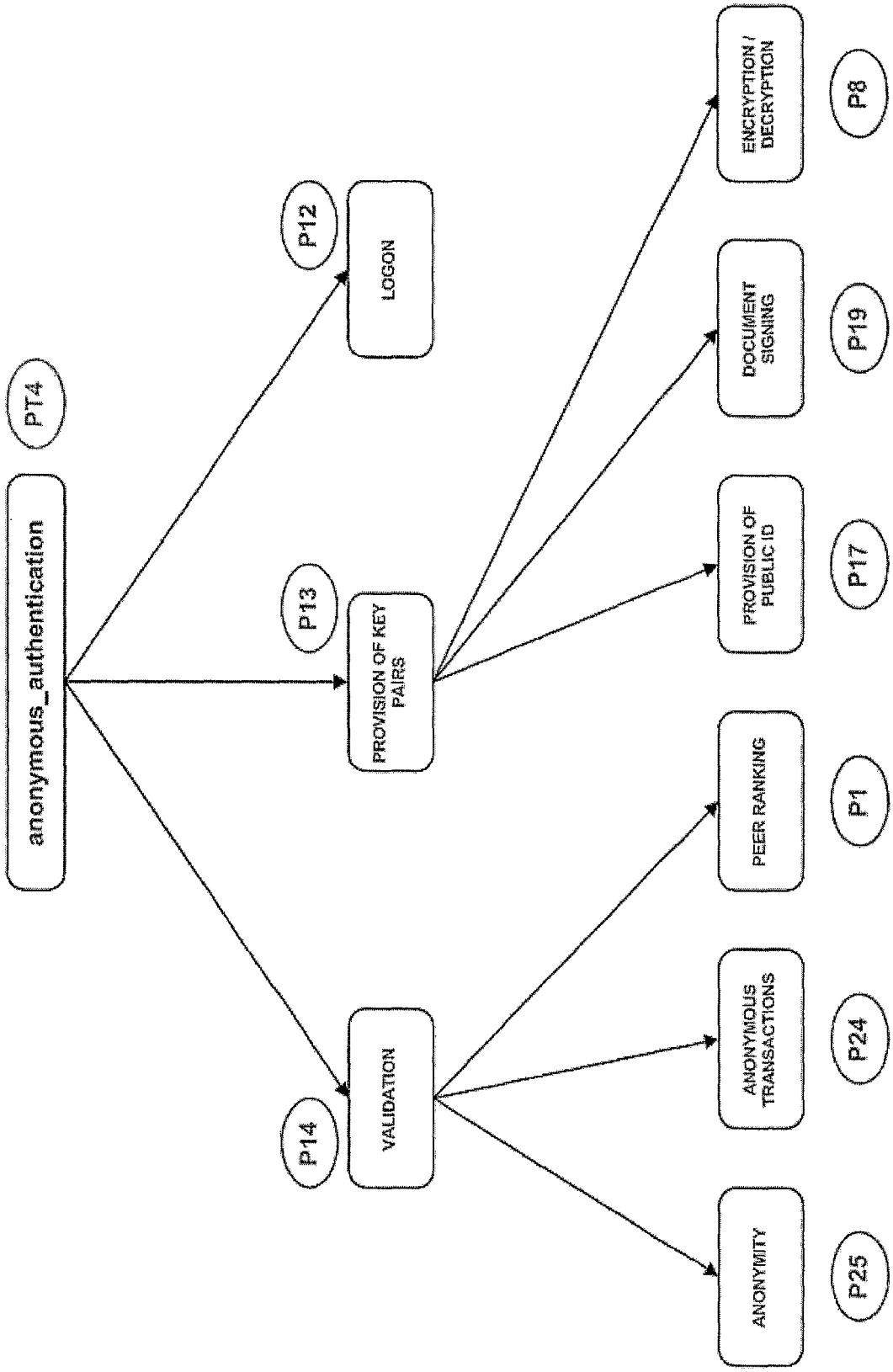
16. A method of any previous claims where the user ID is made unique using known calculable data such as days since a time in history or data retrievable from calculable location;

17. A method of claim 16 where this variable information is merged or diluted into the user provided data prior to hashing or similar to produce a one time ID to further enhance anonymity.

18. A method of claim 17 which upon authentication allows the user access to their resources as with traditional systems except in this case the user remains anonymous.

DRAWINGS:

Figure 1 – Anonymous Authentication associations



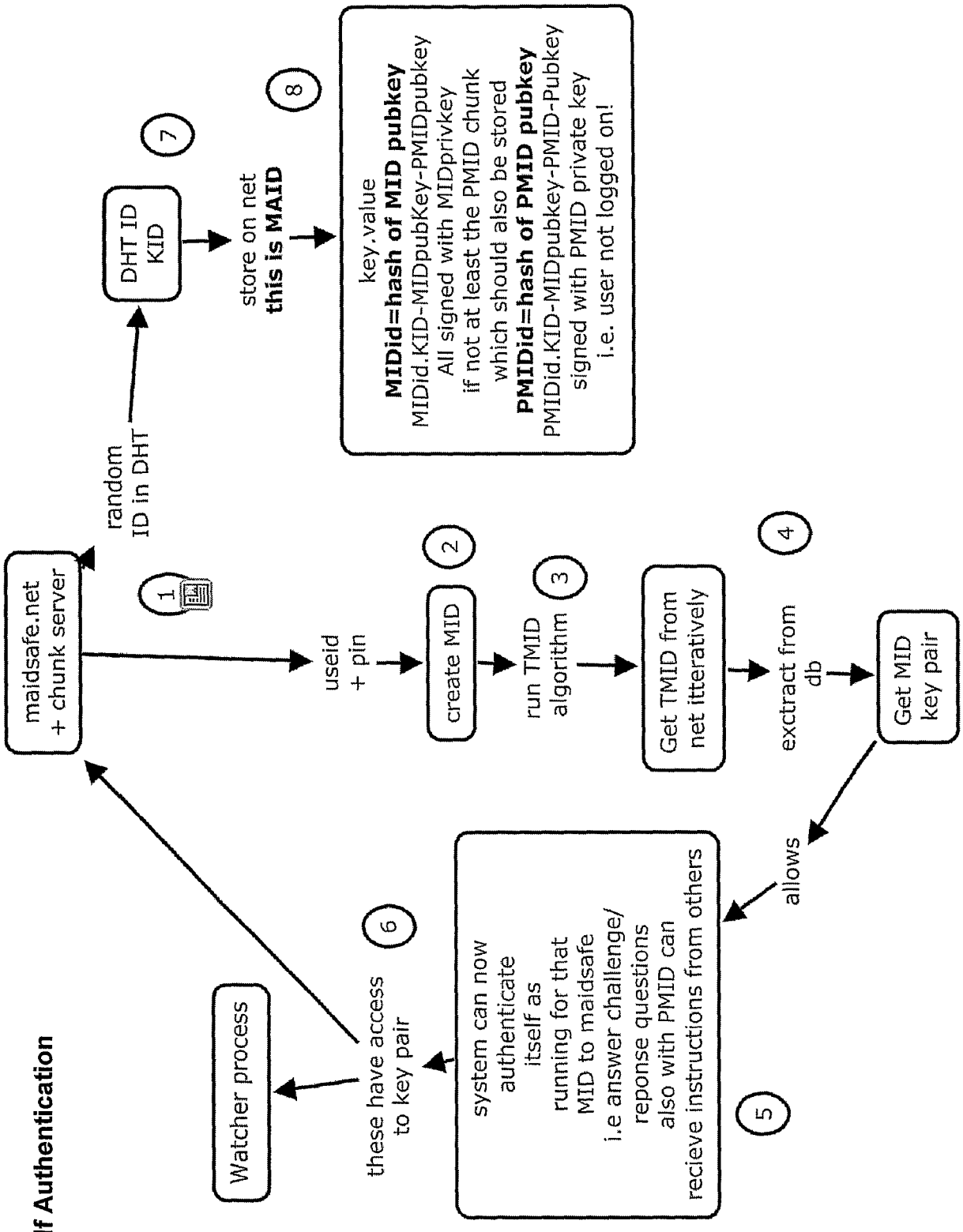


Figure 2 – Self Authentication

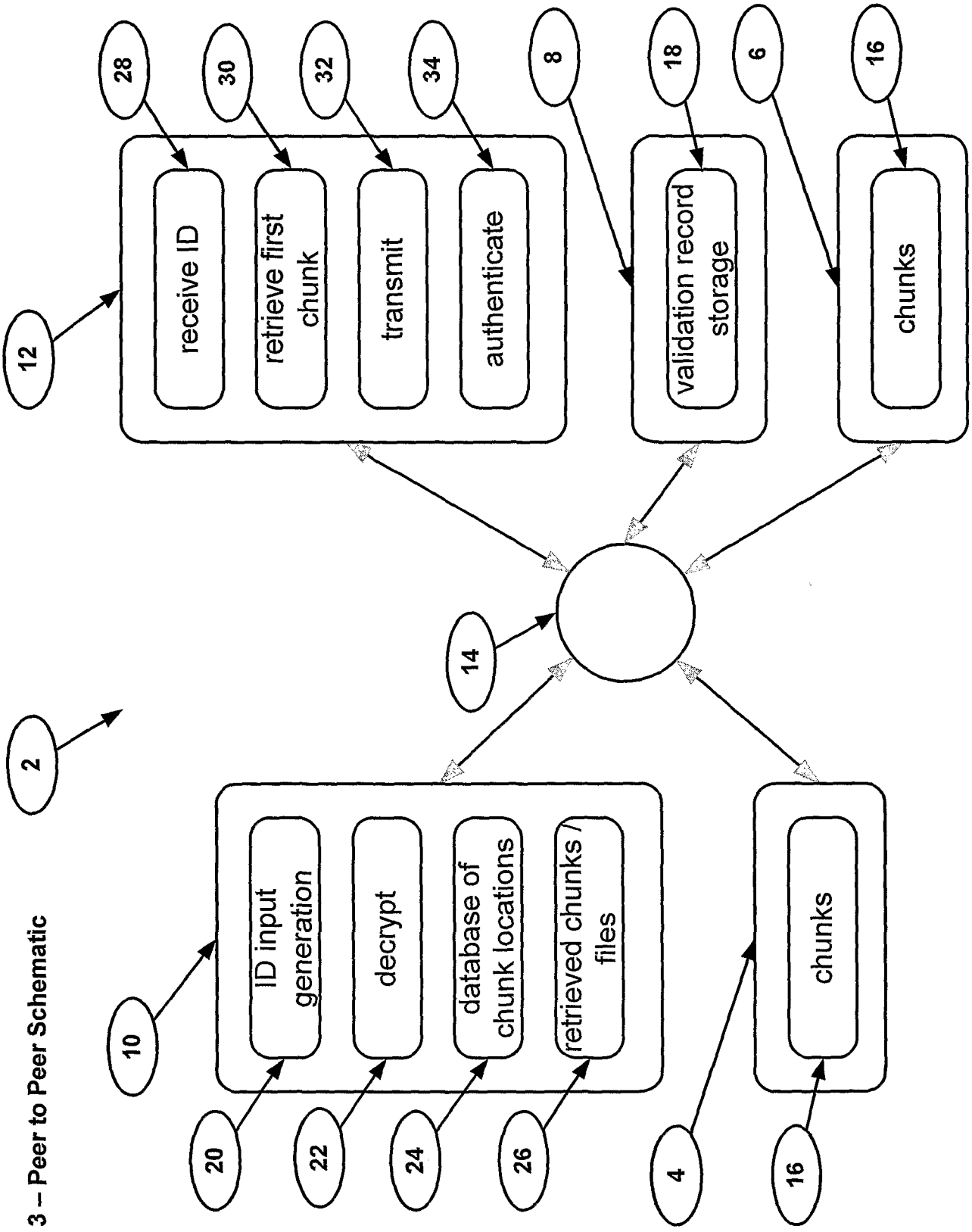


Figure 3 – Peer to Peer Schematic

Figure 4 – Authentication Flowchart

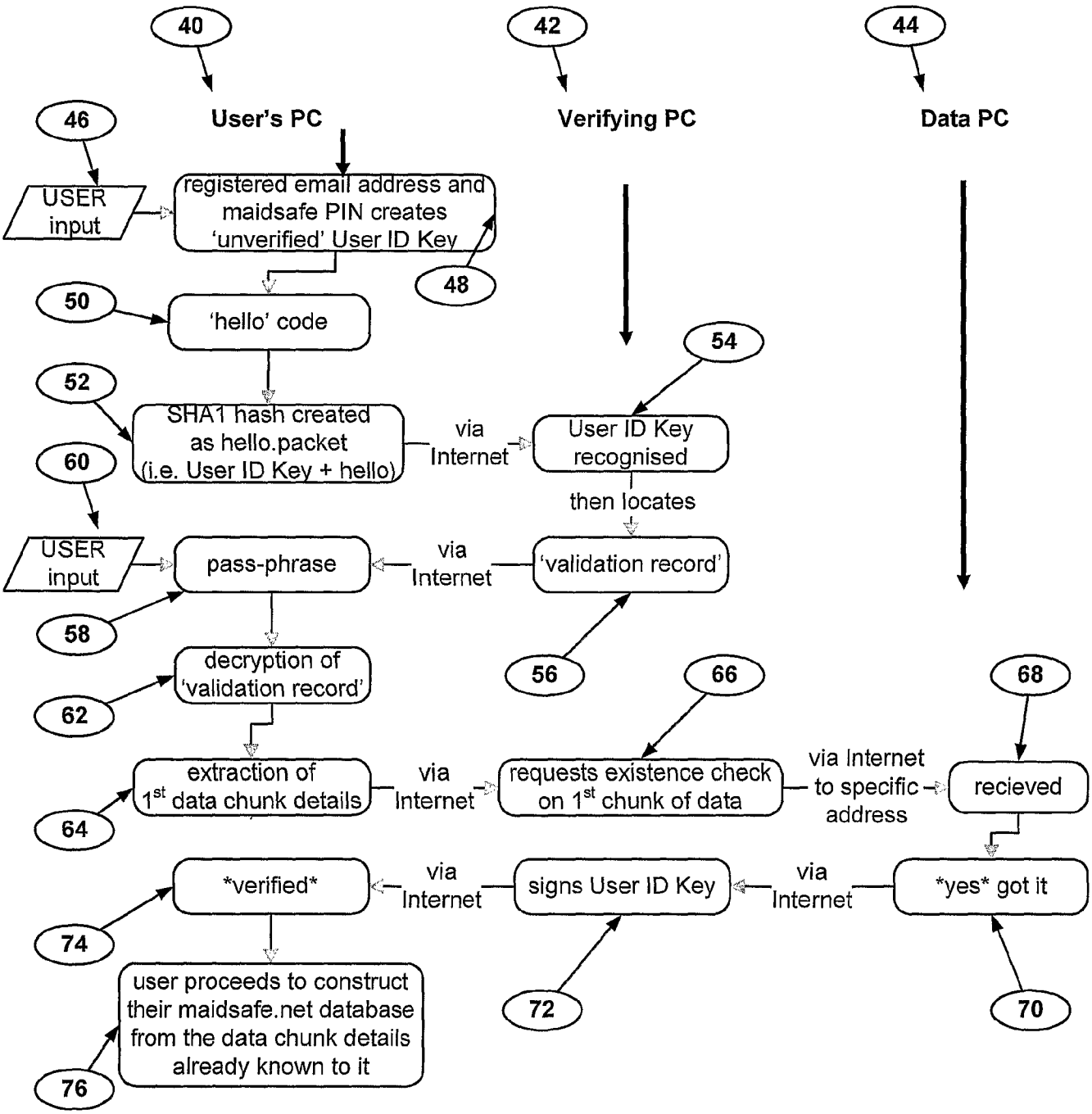


Figure 5 – Data Assurance Event Sequence

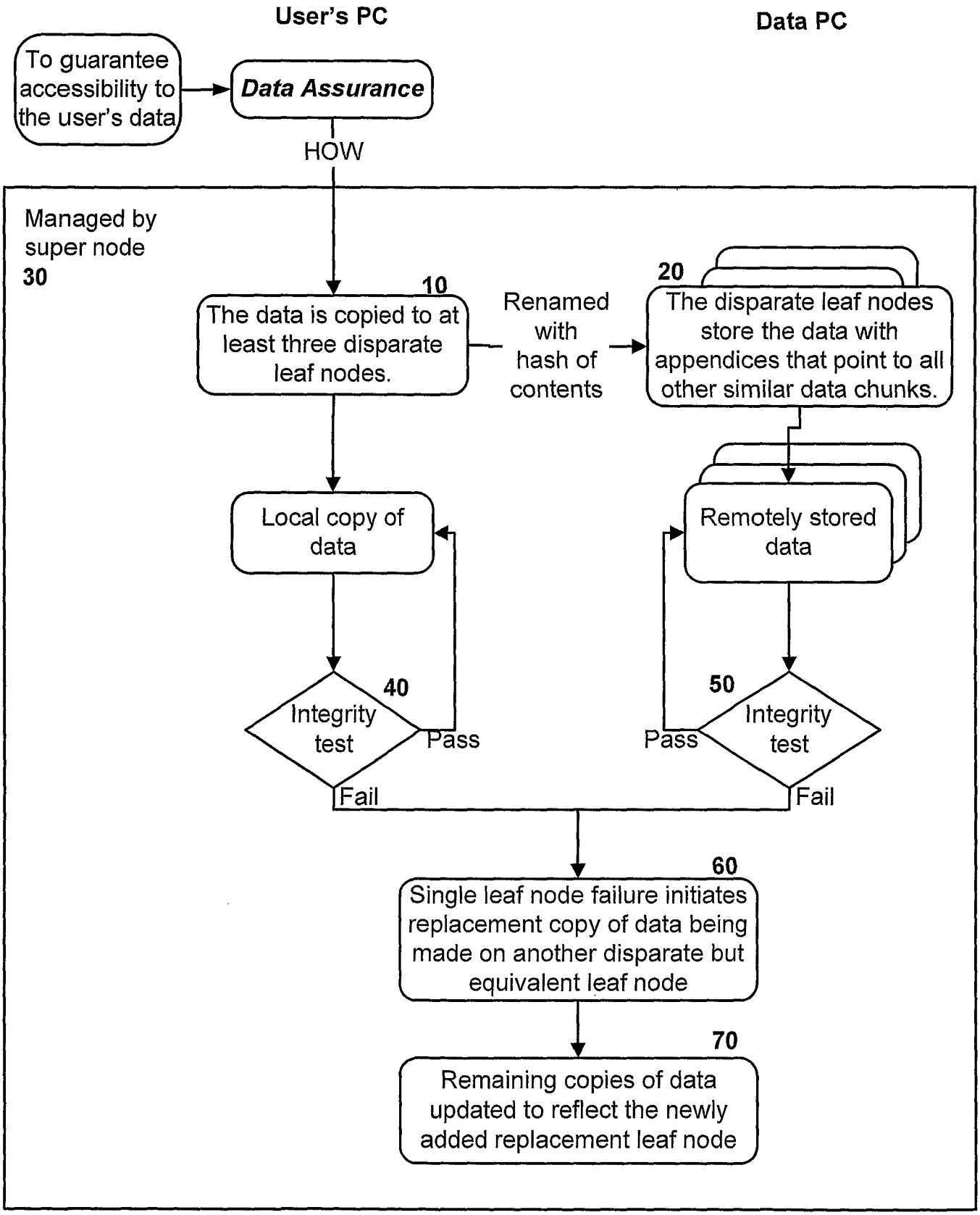
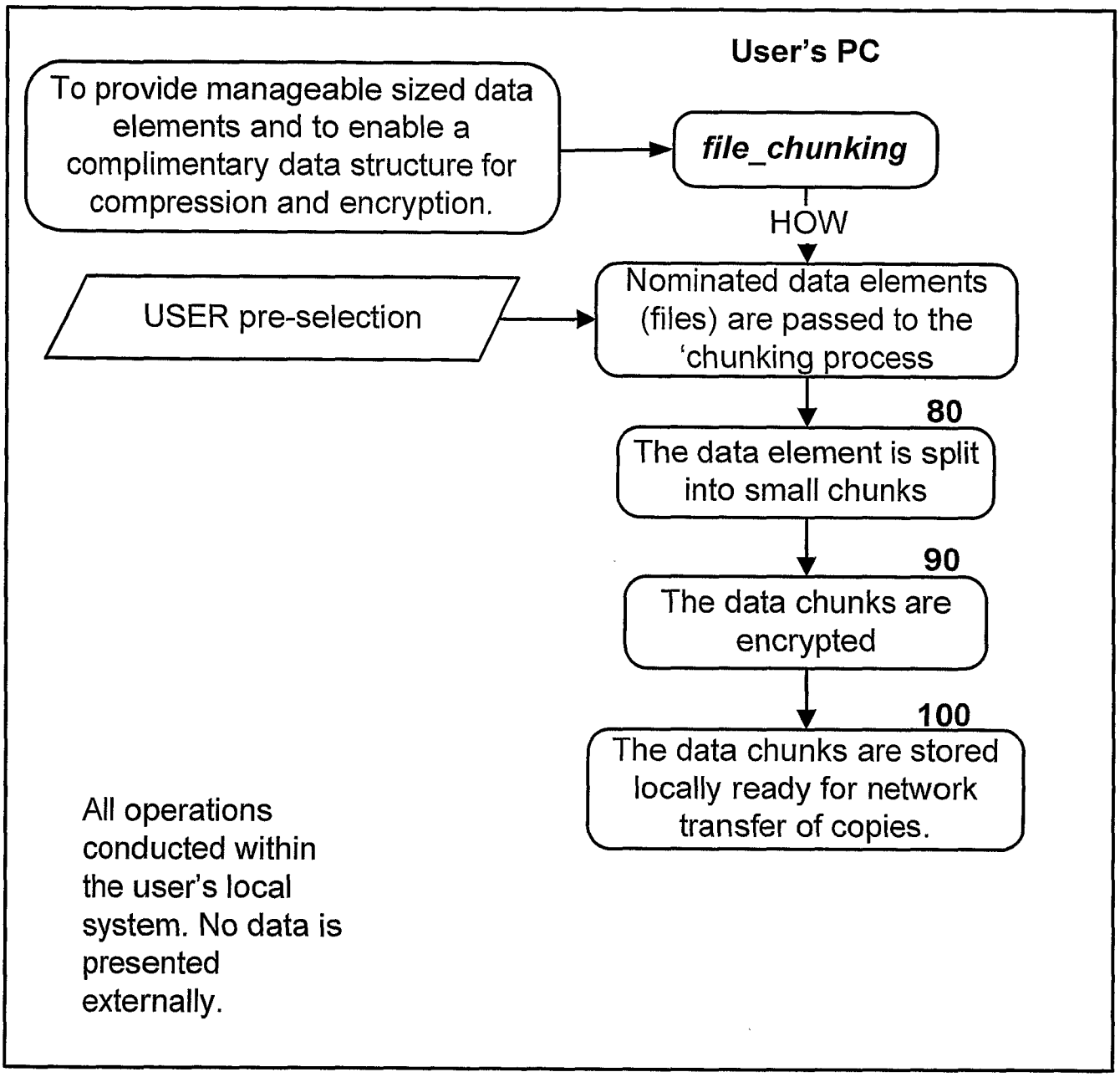


Figure 6 – Chunking Event Sequence



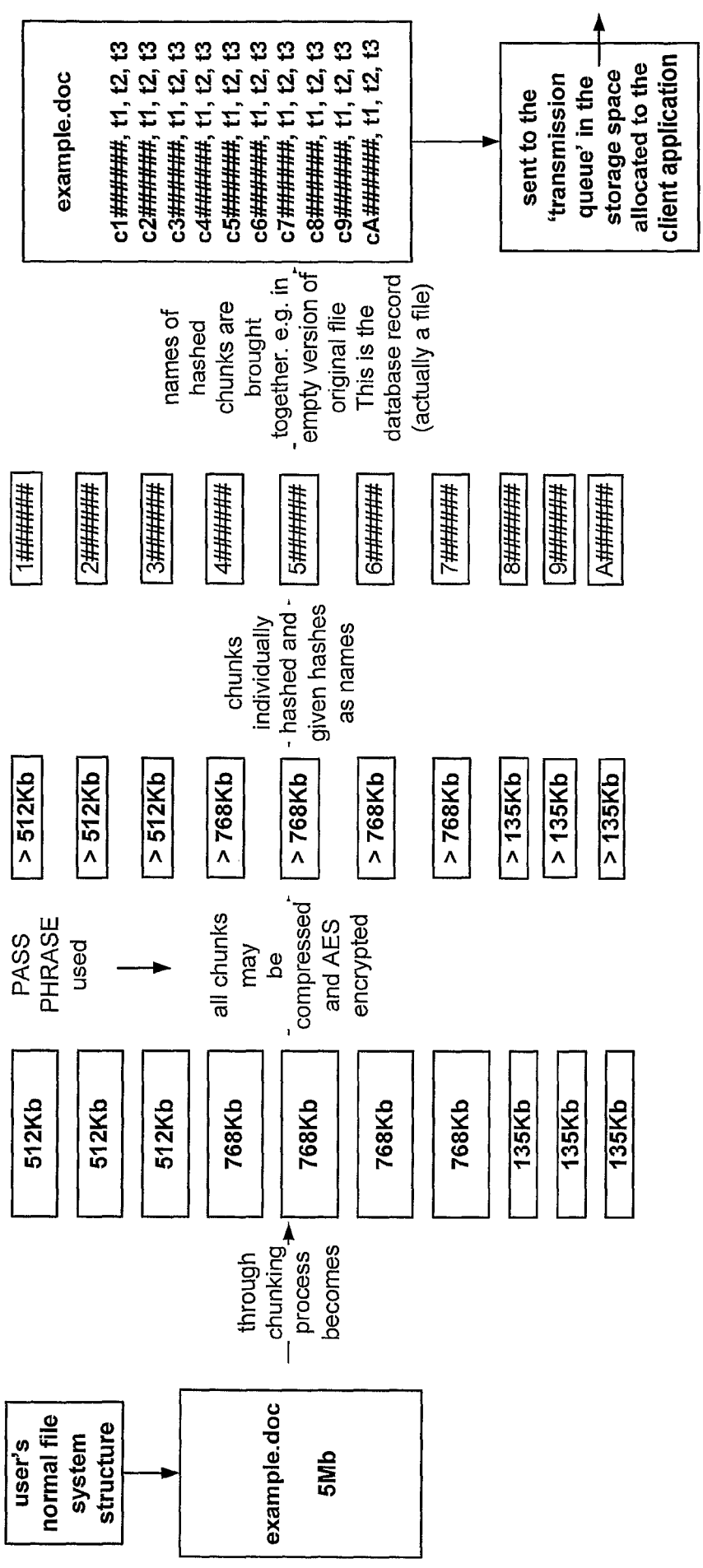
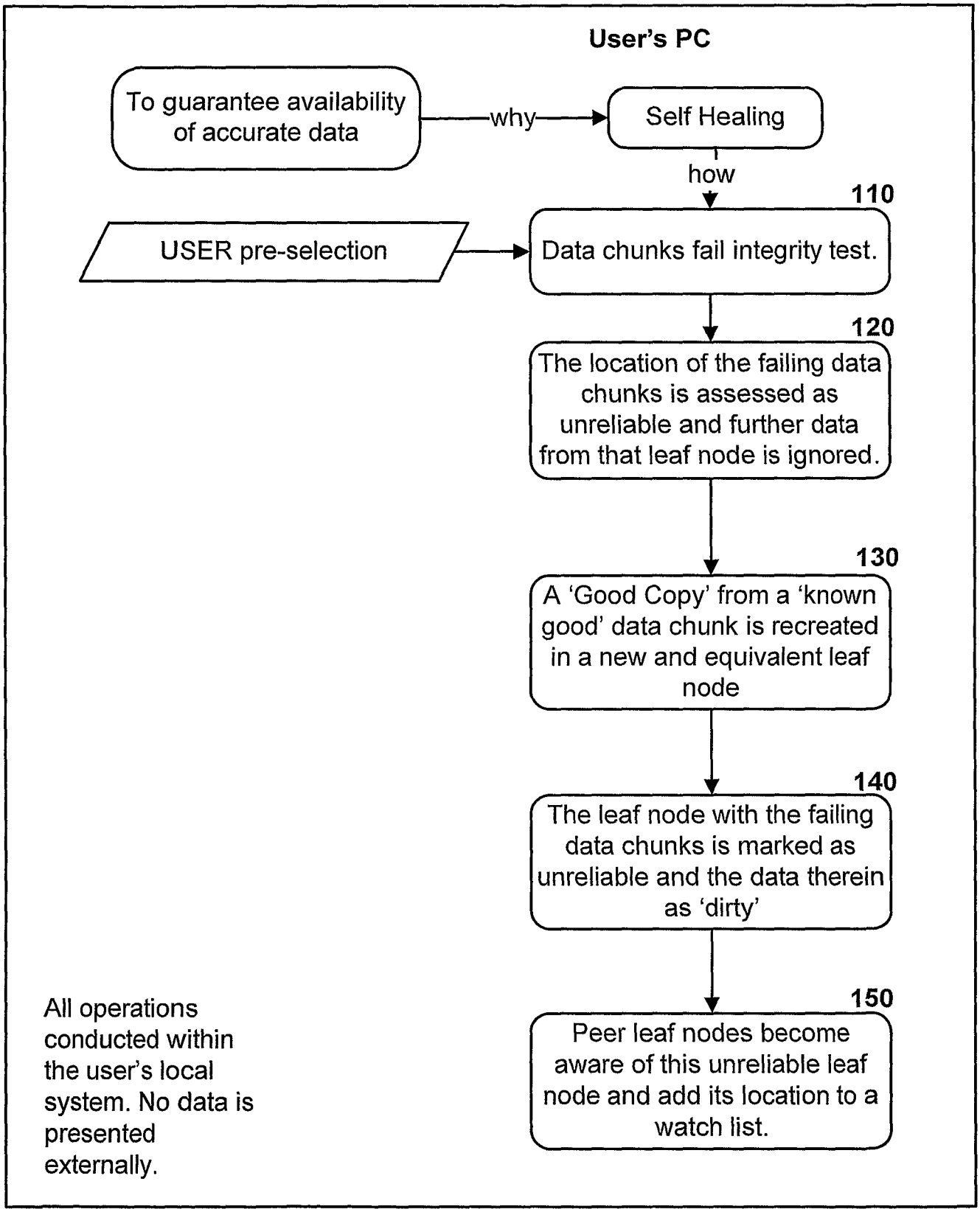


Figure 7 – Chunking Example

Figure 8 – Self Healing Event Sequence



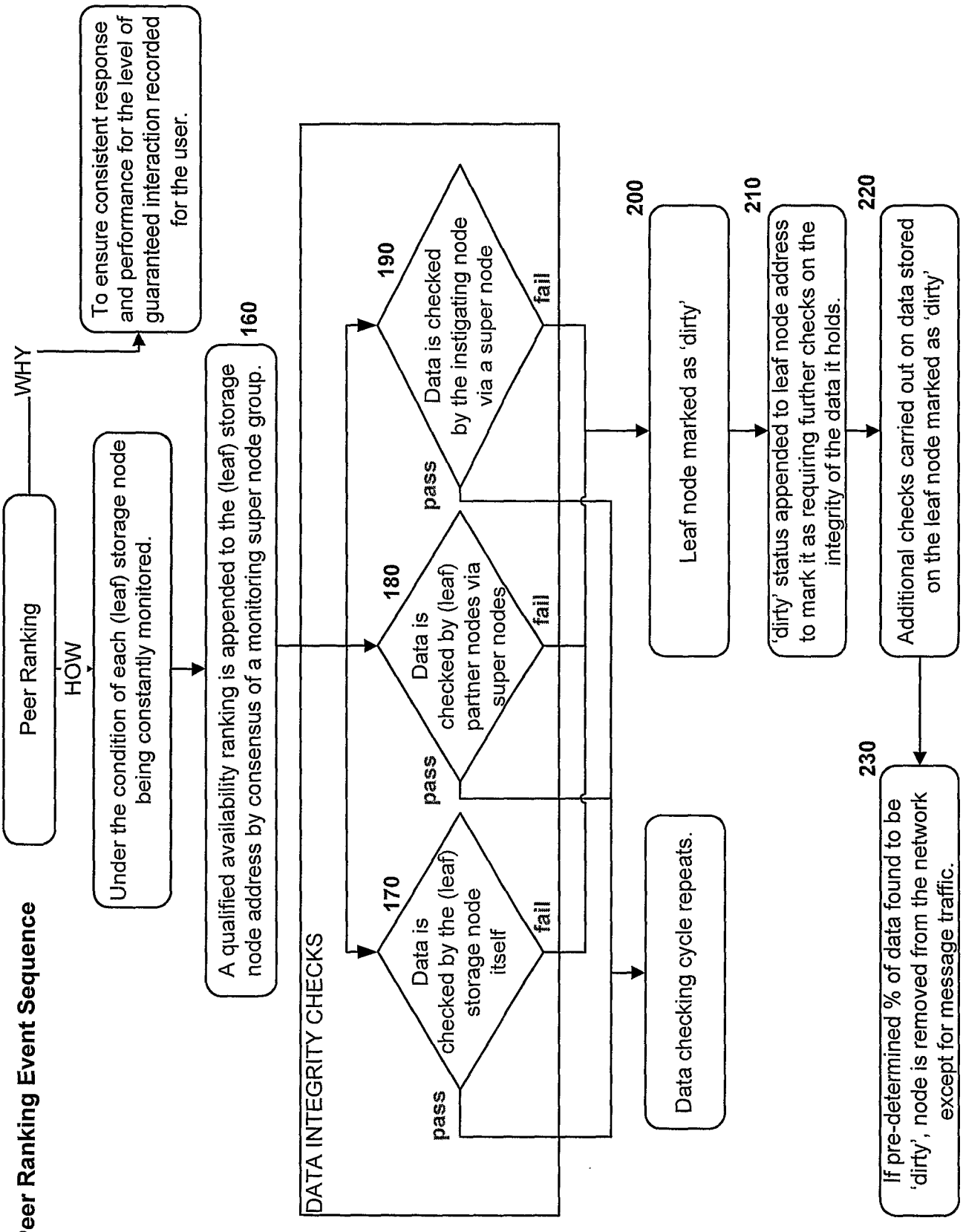


Figure 9 – Peer Ranking Event Sequence

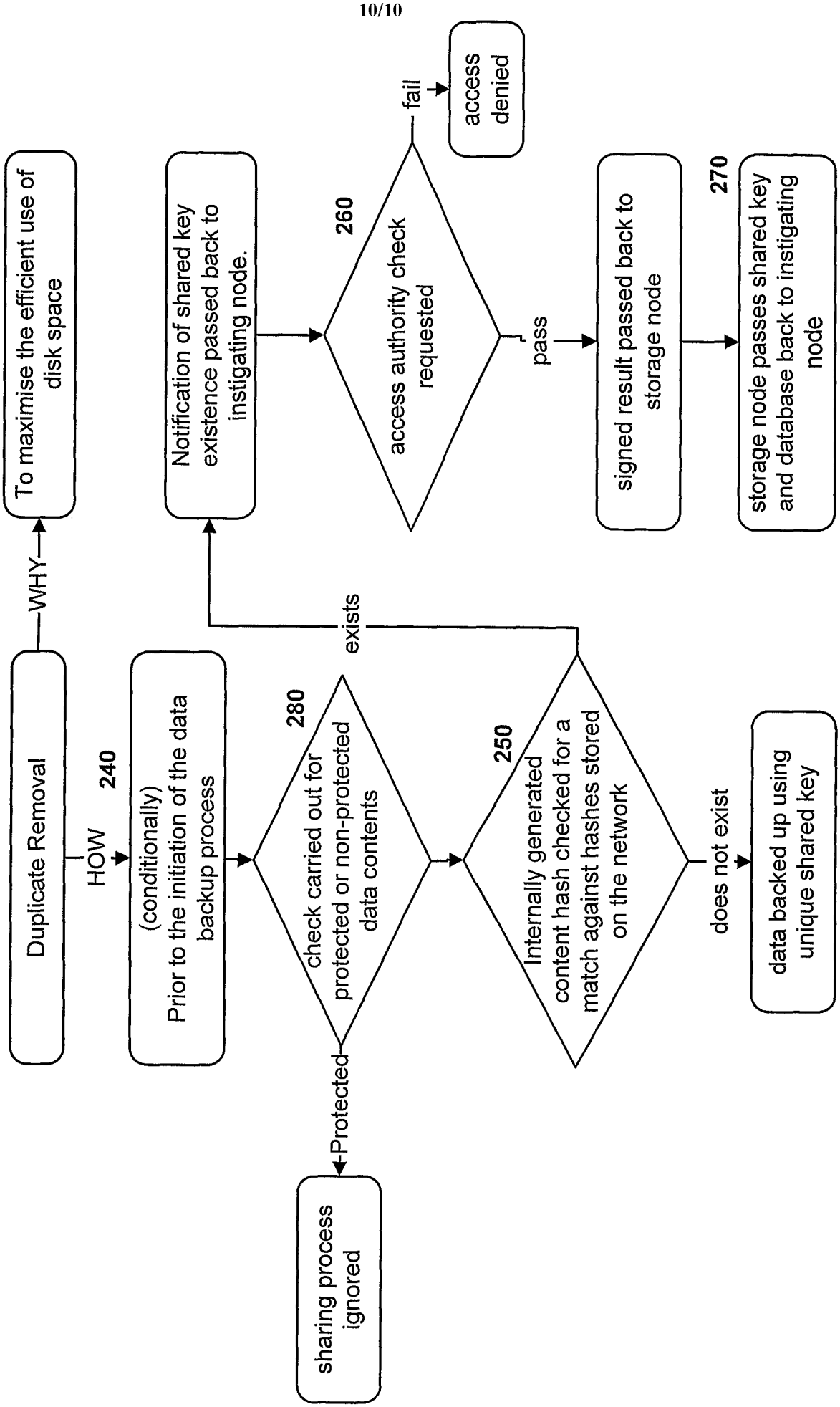


Figure 10 – Duplicate Removal Event Sequence

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2007/004427

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2002/038296 A1 (MARGOLUS NORMAN H [US] ET AL) 28 March 2002 (2002-03-28) abstract paragraphs [0001], [0007] paragraphs [0011] - [0016] paragraphs [0024] - [0029] paragraphs [0044] - [0047] paragraphs [0057] - [0071] paragraphs [0083] - [0097] paragraphs [0105] - [0118] paragraphs [0123] - [0129] paragraphs [0146], [0147] figures 1-11	1-15
Y		16-18

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 April 2008	07/05/2008

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Bichler, Marc

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2007/004427

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	LEACH MICROSOFT M MEALLING VERISIGN P ET AL: "A UUID URN Namespace; draft-mealling-uuid-urn-02.txt;" IETF STANDARD-WORKING-DRAFT, INTERNET ENGINEERING TASK FORCE, IETF, CH, no. 2, 28 January 2004 (2004-01-28), XP015032344 ISSN: 0000-0004 page 1, abstract page 14, section "Node IDs that do not identify the host"	16-18
A	WO 99/37054 A (INST OF SYSTEMS SCIENCE [SG]; HU JIAN [SG]; BAO FENG [SG]; DENG HUIJE) 22 July 1999 (1999-07-22) abstract pages 2-3 pages 7-9 claims 1-10	1-18
A	WO 03/012666 A (DIGITAL DOORS INC [US]) 13 February 2003 (2003-02-13) page 4, line 22 - page 13, line 19 figures 1A,1B,3	1-18

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2007/004427

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2002038296 A1	28-03-2002	US 2004139303 A1	15-07-2004
		US 2004139098 A1	15-07-2004
		US 2004162808 A1	19-08-2004
		US 2004143578 A1	22-07-2004
		US 2004143743 A1	22-07-2004
		US 2004143744 A1	22-07-2004
		US 2004143745 A1	22-07-2004
		US 2004255140 A1	16-12-2004
		US 2005131903 A1	16-06-2005
		US 2005131904 A1	16-06-2005
		US 2005131961 A1	16-06-2005
		US 2005131905 A1	16-06-2005
WO 9937054 A	22-07-1999	AU 6236498 A	02-08-1999
		GB 2349964 A	15-11-2000
WO 03012666 A	13-02-2003	CA 2454439 A1	13-02-2003
		EP 1412868 A1	28-04-2004