(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 2008/065351 A1

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

(21) International Application Number:
PCT/GB2007/004440

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

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0624058.4 1 December 2006 (01.12.2006) GB
0709761.1 22 May 2007 (22.05.2007) GB

(71) Applicant and
(72) Inventor: **IRVINE, David** [GB/GB]; 82A Portland Street, Troon, Ayrshire KA10 2QU (GB).

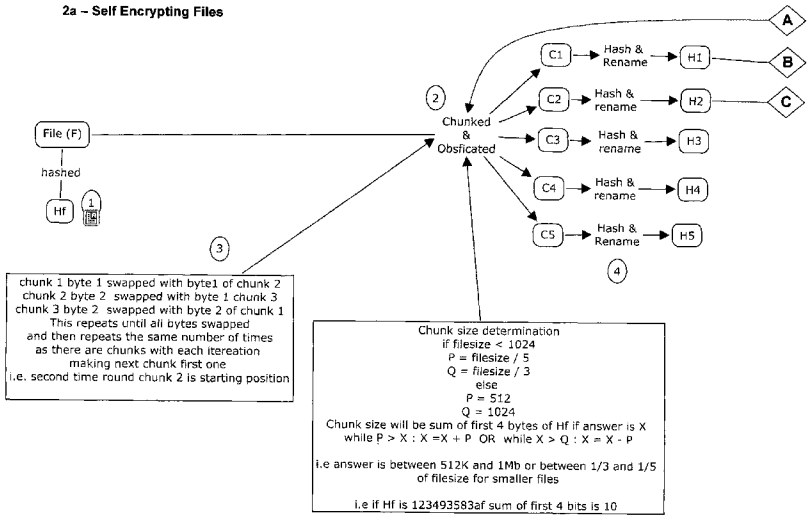
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, 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:** SELF ENCRYPTION



(57) **Abstract:** This present invention provides a method for data to be obfuscated in several ways preferably including self encryption and decryption. The data is preferably chunked, renamed, byte or bit swapped, encrypted and compressed through algorithms seeded by elements preferably derived from the data itself so that data holds the key to reversing the processes used and preferably these keys may be recorded for later use.

WO 2008/065351 A1

Self Encryption

STATEMENT OF INVENTION:

An issue with today's encryption techniques is that a user's key, biometric data or passphrase is used to encrypt every data element, thereby exposing the key on every data element encrypted. Another issue is that eventually all encryption is broken given enough resources, so it is therefore safe to assume that today's strong encryption methods will not suffice in years to come. This implies that storing encrypted data now, will not necessarily protect against that data being unencrypted through some discovered process in the future.

This present invention overcomes these issues by first obfuscating the data, by splitting it into smaller elements, then swapping parts of that data around in a manner to make every element useless on its own, and preferably using known information from the preferably smaller elements or chunks as encryption data that will allow the other elements to be encrypted. This allows data to be hidden and encrypted in such a way, that any attacker would require to obtain all data elements and know the manner in which they connect together and also then crack the encryption used. Even if the data chunks were not encrypted and their encryption was broken, they are useless on their own.

BACKGROUND:

Self-encryption is only possible with combination of number of elements. Described below is prior art for each element.

ENCRYPTION

WO2005093582 discloses method of encryption where data is secured in the receiving node via private tag for anonymous network browsing. However, other numerous encryption methods are also available such as (i) implantation of Reed Solomon algorithm (WO02052787), which ensures data is coded in parabolic fashion for self-repairing and storage, (ii) storage involves incremental backup (WO02052787), (ii) uses stenographic (US2006177094), (iv) use cipher keys (CN1620005), encryption for non text (US2006107048) and US2005108240 discloses user keys and randomly generated leaf node keys. The present invention uses none of these methods of encryption and in particular ensures all chunks are unique and do not point to another for security (an issue with Reed Solomon and N + K implementations of parabolic coding)

SELF-ENCRYPTION

Attempts to moving towards attaining some limited aspects of self-encryption are demonstrated by:

(a) US2003053053625 discloses limitation of asymmetrical and symmetrical encryption algorithms, and particularly not requiring generation of a key stream from symmetric keys, nor requiring any time synchronizing, with minimal computational complexity and capable of operating at high speed. A serial data stream to be securely transmitted is first demultiplexed into a plurality N of encryptor input data stream. The input data slices are created which have a cascade of stages, include mapping & delay functions to generate output slices. These are transmitted though a transmission channel. Decryptor applies inverse step of cascade of stages, equalizing delay function and mapping to generate output data slices. The output data streams are multiplexed. The encryptor and decryptor require no synchronizing or timing and operate in simple stream fashion. N:N mapping does not require expensive arithmetic and implemented in table lookup. This provides

robust security and efficiency. A significant difference between this approach and prior cipher method is that the session key is used to derive processing parameters (tables and delays) of the encryptor and decryptor in advance of data transmission. Instead of being used to generate a key stream at real-time rates. Algorithm for generating parameters from a session key is disclosed. This is a data communications network and not related to current invention.

(b) US2002184485 addresses secure communication, by encryption of message (SSDO-self signing document objects), such that only known recipient in possession of a secret key can read the message and verification of message, such that text and origin of message can be verified. Both capabilities are built into message that can be transmitted over internet and decrypted or verified by computer implementing a document representation language that supports dynamic content e.g. any standard web browser, such that elaborate procedures to ensure transmitting and receiving computers have same software are no longer necessary. Encrypted message or one encoded for verification can carry within itself all information needed to specify the algorithm needed for decryption.

Summary of Invention

The main embodiments of this invention are as follows:

A system of self encryption which has the functional elements of:

1. Duplicate Removal
2. Storing Files
3. Chunking
4. Encryption / Decryption

... with the additionally linked functional elements of:

1. Identify Chunks
2. Self Healing
3. Storage and Retrieval
4. Security Availability
5. Provision of Key Pairs

A system of self-encryption of data in a distributed and peer to peer network

A product for self-encryption of data in a distributed and peer to peer network

A system to provide self-encryption in a distributed network which is made of inter linkage all or some of the following elements;

a. encryption / decryption
b. chunking
c. duplicate removal
d. storing files

A system to provide self-encryption in a distributed network which is made of inter linkage all or some of the following elements and sub-elements;

a. encryption / decryption
    i. key pair
    ii. security
b. chunking
    i. identify chunking
c. duplicate removal
    i. identify chunking
    ii. storage & retrieval
    iii. self healing
d. storing files
    i. identify chunking
    ii. storage & retrieval
    iii. self healing

A product for self-encryption in a distributed network which is made of inter linkage all or some of the following elements;

a. encryption / decryption
b. chunking
c. duplicate removal
d. storing files

A product for self-encryption in a distributed network which is made of inter linkage all or some of the following elements and sub-elements;

a. encryption / decryption
    i. key pair
    ii. security
b. chunking

i. identify chunking

c. duplicate removal
   i. identify chunking
   ii. storage & retrieval
   iii. self healing

d. storing files
   i. identify chunking
   ii. storage & retrieval
   iii. self healing

A method of system and product for self-encryption of data in a distributed and peer to peer network

A method of above of securely protecting data in a distributed network, suitable for a self repairing process by chunking the data into many pieces.

A method of above where data privacy by byte or bit exchange and encryption is based on content derived from the data itself.

A method of above where data reconstitution capability is provided only for individuals who know of and/or have the original data elements.

A method of maximising disk space in a worldwide network by aiding the removal of duplicate files, as each data element will always produce the exact same chunks and names regardless of the actual file name itself.

A method of data encryption using only calculable elements from the file contents and not user keys or user passwords.

A method of above where the actual file is first passed though a content swapping (such as byte swapping)algorithm to completely dilute the

contents across the data element(s), thereby rendering each chunk useless even if the encryption key is known.

DESCRIPTION

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.

Linked elements for Self Encryption (Figure 1 – PT2)

The Self Encryption invention consists of 4 key functional elements, with a further 5 functional elements being linked with.

The key functional elements are:
P5 – Duplicate Removal
P6 – Storing Files
P7 – Chunking
P8 – Encryption / Decryption

The linked functional elements are:
P9 – Identify Chunks
P2 – Self Healing
P4 – Storage and Retrieval
P3 – Security Availability
P13 – Provision of Key Pairs

The self-encryption (PT2) itself is made up from linkage of elements, storing file (P6), duplicate removal (P5), chunking (P7) and encryption / decryption (P8) which allows a self-encryption process to provide security

and global duplicate data removal. In addition, storing file element (P6) is preferably dependent upon sub-elements storage and retrieval (P4) and sub-element identify chunks (P9) and generate sub-element self-healing (P2), duplicate removal element (P5) is preferably dependent on sub-element identify chunks (P9), chunking element (P7) generate sub-element identify chunks (P9) and encryption / decryption element (P8) can be provided by sub-element provision of keys (P13) to ensure validity of generating or requesting nodes anonymous identity (e.g. we don't know who it is but we know it was the node that put the chunk there) thereby ensuring security availability (P3).

Chunking (Figure 1 – P7)

According to a related aspect of this invention, files are split preferably using an algorithm to work out the chunk size into several component parts. The size of the parts is preferably worked out from known information about the file as a whole, preferably the hash of the complete file. This information is run through an algorithm such as adding together the first x bits of the known information and using modulo division to give a chunk size that allows the file to preferably split into at least three parts.

Preferably known information from each chunk is used as an encryption key. This is preferably done by taking a hash of each chunk and using this as the input to an encryption algorithm to encrypt another chunk in the file. Preferably this is a symmetrical algorithm such as AES256.

Preferably this key is input into a password creating algorithm such as pbkdf and an initial vector and key calculated from that. Preferably the iteration count for the pbkdf is calculated from another piece of known information, preferably the sum of bits of another chunk or similar.

Preferably each initial chunk hash and the final hash after encryption are stored somewhere for later decryption.

Self Encrypting Files (Figure 2a/b)

1. Take a content hash of a file or data element

2. Chunk a file with preferably a random calculable size i.e. based on an algorithm of the content hash (to allow recovery of file). Also obfuscate the file such as in 3

3. Obfuscate the chunks to ensure safety even if encryption is eventually broken (as with all encryption if given enough processing power and time)

   a. chunk 1 byte 1 swapped with byte1 of chunk 2
   b. chunk 2 byte 2 swapped with byte 1 chunk 3
   c. chunk 3 byte 2 swapped with byte 2 of chunk 1
   d. This repeats until all bytes swapped and then repeats the same number of times as there are chunks with each iteration making next chunk first one
   e. - i.e. second time round chunk 2 is starting position

4. Take hash of each chunk and rename chunk with its hash.

5. Take h2 and first x bytes of h3 (6 in our example case) and either use modulo division or similar to get a random number between 2 fixed parameter (in our case 1000) to get a variable number. Use the above random number and h2 as the encryption key to encrypt h1 or use h2 and the random number as inputs to another algorithm (pdbfk2 in our case) to create a key and iv.(initialisation vector)

6. This process may be repeated multiple times to dilute any keys throughout a series of chunks.

7. Chunk name i.e. h1 (unencrypted) and h1c (and likewise for each chunk) written to a location for later recovery of the data. Added to this we can simply update such a location with new chunks if a file has been altered, thereby creating a revision control system where each file can be rebuilt to any previous state.

8. The existence of the chunk will be checked on the net to ensure it is not already backed up. All chunks may be checked at this time.

9. If a chunk exists all chunks must be checked for existence.

10. The chunk is saved

11. The file is marked as backed up.

12. If a collision is detected the process is redone altering the original size algorithm (2) to create a new chunk set, each system will be aware of this technique and will do the exact same process till a series of chunks do not collide. There will be a back off period here to ensure the chunks are not completed due to the fact another system is backing up the same file. The original chunk set will be checked frequently in case there are false chunks or ones that have been forgotten. If the original names become available the file is reworked using these parameters.

Duplicate Removal (Figure 1 – P5)

According to a related aspect of this invention, data chunked and ready for storing can be stored on a distributed network but a search should preferably be carried out for the existence of all associated

chunks created. Preferably the locations of the chunks have the same ranking (From earlier ranking system) as user or better, otherwise the existing chunks on the net are promoted to a location of equivalent rank at least. If all chunks exist then the file is considered as already backed up. If less than all chunks exist then this will preferably be considered as a collision (after a time period) and the file will be re chunked using the secondary algorithms (preferably just adjusted file sizes). This allows duplicate files on any 2 or more machines to be only backed up once, although through perpetual data several copies will exist of each file, this is limited to an amount that will maintain perpetual data.

Encrypt – Decrypt (Figure 1 – P8)

According to a related aspect of this invention, the actual encrypting and decrypting is carried out via knowledge of the file's content and this is somehow maintained (see next). Keys will be generated and preferably stored for decrypting. Actually encrypting the file will preferably include a compression process and further obfuscation methods. Preferably the chunk will be stored with a known hash preferably based on the contents of that chunk.

Decrypting the file will preferably require the collation of all chunks and rebuilding of the file itself. The file may preferably have its content mixed up by an obfuscation technique rendering each chunk useless on its own.

Preferably every file will go through a process of byte (or preferably bit) swapping between its chunks to ensure the original file is rendered useless without all chunks.

This process will preferably involve running an algorithm which preferably takes the chunk size and then distributes the bytes in a

pseudo random manner preferably taking the number of chunks and using this as an iteration count for the process. This will preferably protect data even in event of somebody getting hold of the encryption keys – as the chunks data is rendered useless even if transmitted in the open without encryption.

This defends against somebody copying all data and storing for many years until decryption of today's algorithms is possible, although this is many years away.

This also defends against somebody; instead of attempting to decrypt a chunk by creating the enormous amount of keys possible, (in the region of 2^54) rather instead creating the keys and presenting chunks to all keys – if this were possible (which is unlikely) a chunk would decrypt. The process defined here makes this attempt useless.

All data will now be considered to be diluted throughout the original chunks and preferably additions to this algorithm will only strengthen the process.

Security (Figure 1 – P3)

According to a related aspect of this invention, each file is split into small chunks and encrypted to provide security for the data. Only the person or the group, to whom the overall data belongs, will know the location of the other related but dissimilar chunks of data.

Preferably, each of the above chunks does not contain location information for any other dissimilar chunks; which provides for security of data content, a basis for integrity checking and redundancy.

Preferably, 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 which allows persistence of data.

Preferably, 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.

Preferably, 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, which provides efficiency that in turn allows data resilience in cases where network churn is a problem as in peer to peer or distributed networks. This system message can be preferably passed via the messenger system described herein.

Preferably the system may simply allow a user to search for his chunks and through a challenge response mechanism, locate and authenticate himself to have authority to get/forget this chunk.

Further users can decide on various modes of operation preferably such as maintain a local copy of all files on their local machine, unencrypted or chunked or chunk and encrypt even local files to secure machine (preferably referred to as off line mode operation) or indeed users may decide to remove all local data and rely completely on preferably maidsafe.net or similar system to secure their data.

CLAIMS

1. A system to provide self-encryption in a distributed network which allows the data to be chunked, renamed, byte or bit swapped, encrypted and compressed through algorithms seeded by elements derived from the data itself so that data holds the key to reversing the processes used and these are recorded for later use and aids security and duplicate removal on a network wide basis this system comprises of combination of following steps;
   a. encryption / decryption
   b. chunking
   c. duplicate removal
   d. storing files

   the above combination provides a unique system with cumulative and synergistic benefits to allow people to secure communications and secure data.

2. A system of claim 1 to provide self-encryption in a distributed network which allows the data to be chunked, renamed, byte or bit swapped, encrypted and compressed through algorithms seeded by elements derived from the data itself so that data holds the key to reversing the processes used and these are recorded for later use and aids security and duplicate removal on a network wide basis this system comprises of combination of following steps;
   a. encryption / decryption, which further comprises of key pair and security,
   b. chunking, which further comprises of identify chunking,
   c. duplicate removal, which further comprises of identify chunking,
   d. storing files, which further comprises of identify chunking, storage & retrieval and self healing

   the above combination provides a unique system with cumulative and synergistic benefits to allow people to secure communications and secure data.

3. A product to provide self-encryption in a distributed network which allows the data to be chunked, renamed, byte or bit swapped, encrypted and compressed through algorithms seeded by elements derived from the data itself so that data holds the key to reversing the processes used and these are recorded for later use and aids security and duplicate removal on a network wide basis this product comprises of combination of following steps;
   e. encryption / decryption
   f. chunking
   g. duplicate removal
   h. storing files
   the above combination provides a unique product with cumulative and synergistic benefits to allow people to secure communications and secure data.

4. A product of claim 3 to provide self-encryption in a distributed network which allows the data to be chunked, renamed, byte or bit swapped, encrypted and compressed through algorithms seeded by elements derived from the data itself so that data holds the key to reversing the processes used and these are recorded for later use and aids security and duplicate removal on a network wide basis this product comprises of combination of following steps;
   a. encryption / decryption, which further comprises of key pair and security,
   b. chunking, which further comprises of identify chunking,
   c. duplicate removal, which further comprises of identify chunking,
   d. storing files, which further comprises of identify chunking, storage & retrieval and self healing
   the above combination provides a unique system with cumulative and synergistic benefits to allow people to secure communications and secure data

5. A method of claim 1-4 where it is to identify data elements using a data map with only a sequence of content hashes for each chunk of data before and after encryption;

6. A method of claims 1-5 storing and retrieving these maps on an insecure network;

7. A method of claim 5 where each, new iteration of a data element is appended to the data map to create a strong revision control system;

8. A method of claim 5 where data elements are obfuscated by encryption or other obfuscation technique, or similar, can be reconstructed in conjunction with the data map;

9. A method of claim 5 where the maps can be stored in private or public locations and/or biometrically accessed;

10. A system of claims 1-2 which allows data to have multiple locations, revisions and encryption or other obfuscation techniques and for the pointer to the data to be a very small file containing the basic information to reconstitute a complete data element at any time from any location on the network;

11. A system of claims 1-2 which allows the identification of which chunks to make up which files;

12. A system of claims 1-2 which allows data maps which preferably become discreet data chunks on the network, just like any other associated data element and are therefore undetectable as data maps;

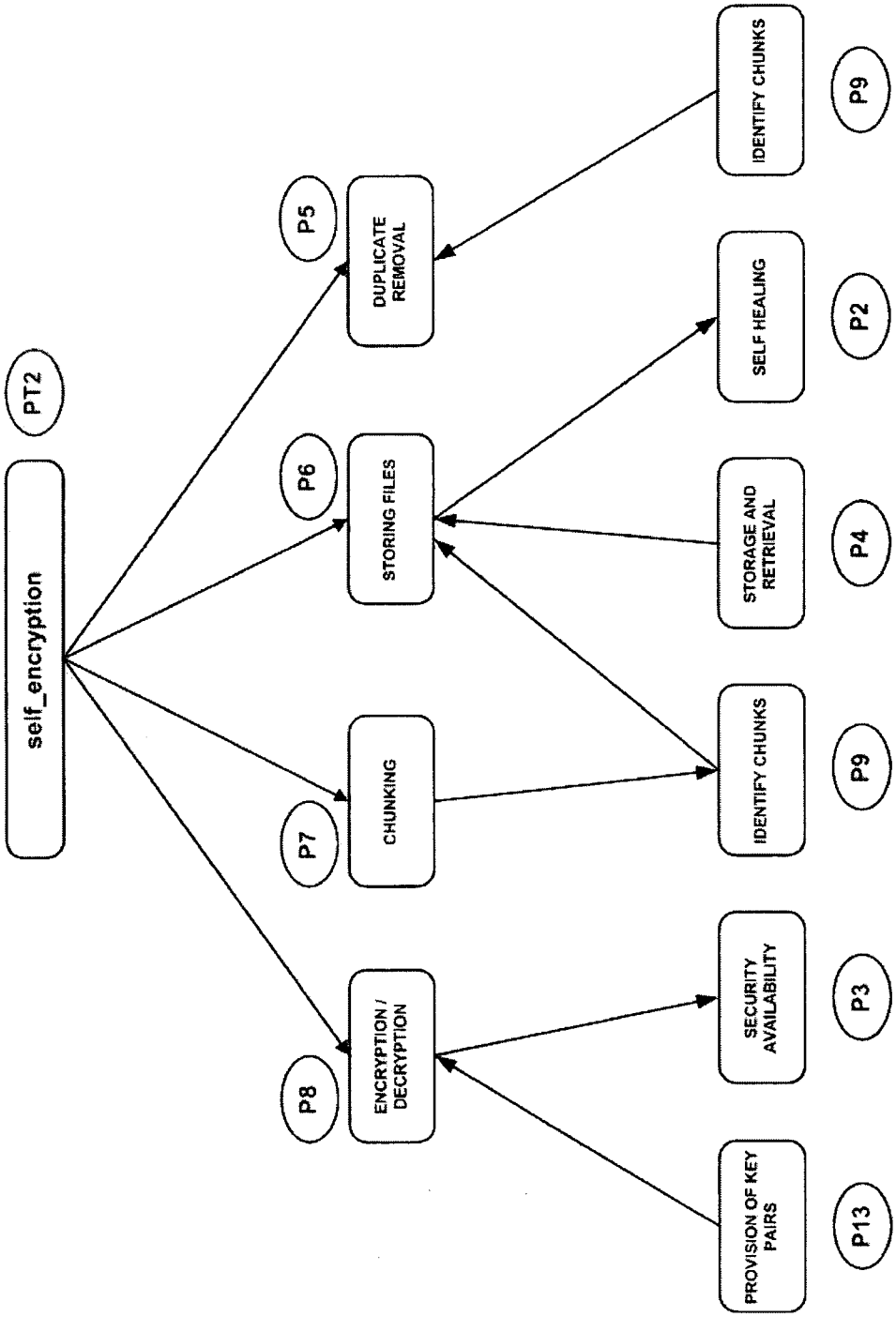


Figure 1 – Self Encryption associations

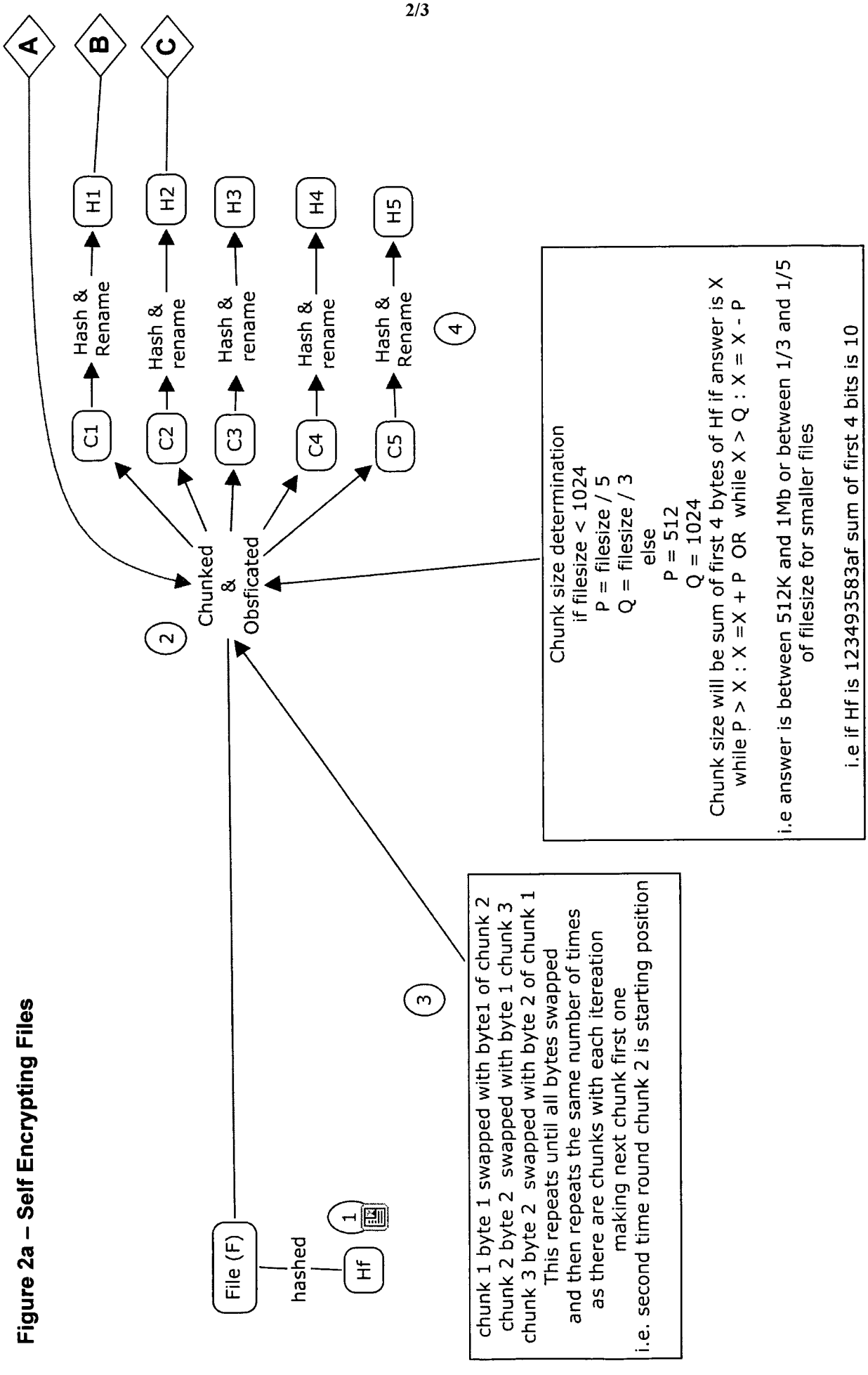


Figure 2a – Self Encrypting Files

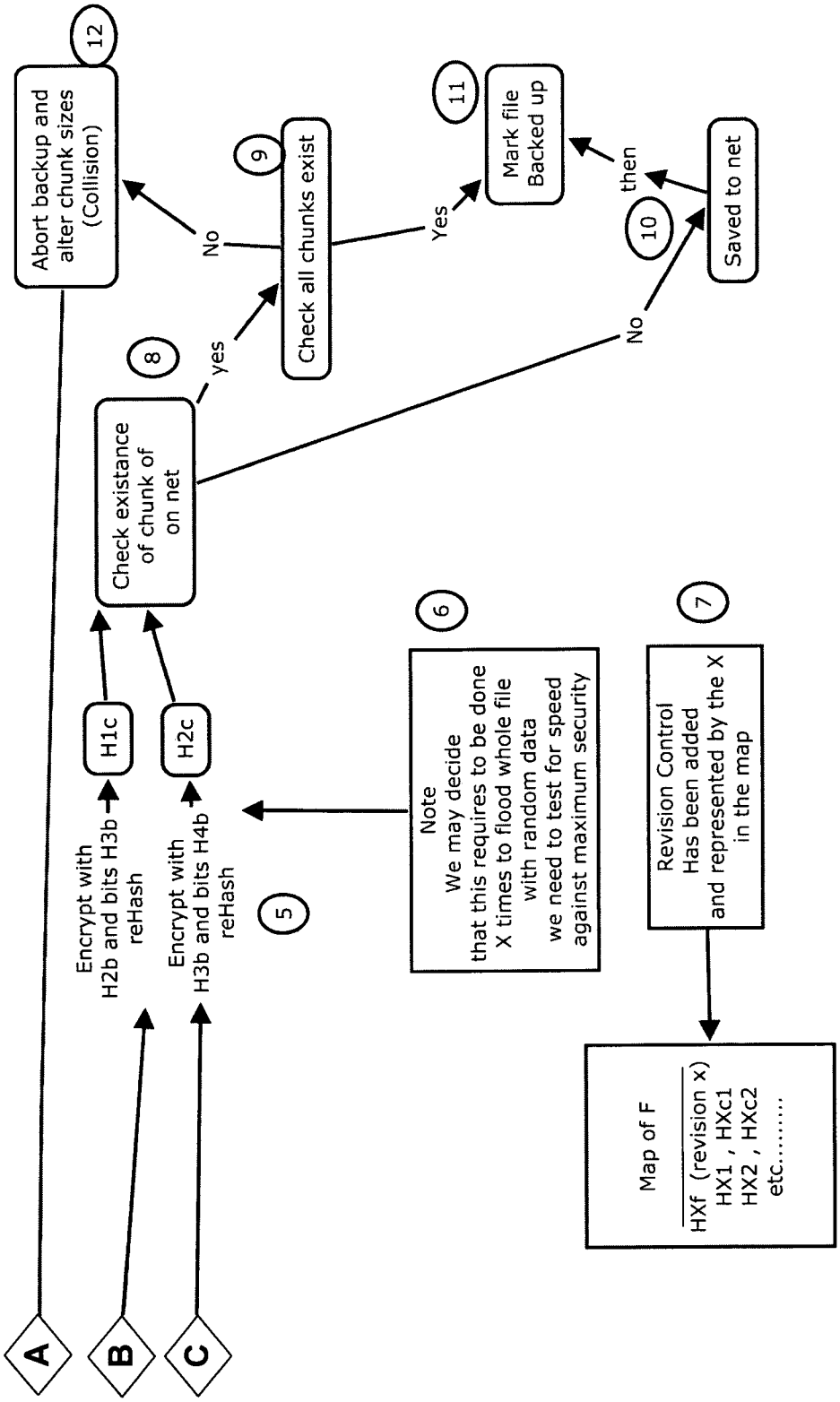


Figure 2b – Self Encrypting Files

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2007/004440

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 [0013] - [0016] paragraphs [0024] - [0029] paragraph [0047] paragraphs [0057] - [0060] paragraph [0068] paragraphs [0105] - [0118] figures 1-11	1-12
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	1-12

-/--

[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
7 March 2008	28/03/2008

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2007/004440

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
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	1-12
A	US 5 727 062 A (RITTER TERRY F [US]) 10 March 1998 (1998-03-10) abstract column 6, line 40 - column 10, line 61	1-12

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2007/004440

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
US 5727062 A	10-03-1998	NONE	