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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 October 2013 (24.10.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/156758 A1

(51) International Patent Classification:
H04L 9/06 (2006.01)

(21) International Application Number:
PCT/GB2013/050936

(22) International Filing Date:
11 April 2013 (11.04.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
1206636.1 16 April 2012 (16.04.2012) GB

(71) Applicant: **MAIDSAFE,NET LIMITED** [GB/GB]; 72 Templehill, Troon, Ayrshire, KA10 6BE (GB).

(72) Inventor: **IRVINE, David**; 82A Portland Street, Troon, Troon, Ayrshire, KA10 6QU (GB).

(74) Agent: **HARRISON GODDARD FOOTE**; Delta House, 50 West Nile Street, Glasgow, G1 2NP (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

WO 2013/156758 A1

(54) Title: METHOD OF ENCRYPTING DATA

(57) Abstract: A method of encrypting data comprising the steps of: creating a one time pad; and encrypting the data using the one time pad to produce output data, wherein the one time pad is generated using the data.

Method of Encrypting Data

The present invention relates to methods of encrypting and decrypting data. In particular, but not exclusively, the invention relates to improved methods which have, or come closer to having, perfect secrecy.

A perfectly secure cryptosystem is secure even when an adversary has unlimited computing power. It uses an encryption algorithm that does not depend for its effectiveness on unproven assumptions about computational hardness. The algorithm is not vulnerable to future developments, such as quantum computing.

In cryptography, there are two types of encryption: symmetric key cryptography and asymmetric key (also known as public-key) cryptography. With the former type, trivially related or identical cryptographic keys are used for both encryption of plaintext and decryption of ciphertext. With the latter, two different but mathematically related keys are used: a public key and a private key. The calculation of the private key is intended to be 'computationally infeasible' from the public key, even though they are related.

Conventional symmetric encryption involves complex substitution and transposition of data. At present, and despite their prevalence, it is not known whether there can be a cryptanalytic procedure which can reverse these transformations without knowing the key used during encryption. Symmetric ciphers have been susceptible to various forms of attacks, and it does appear that there is ongoing progress towards developing such a cryptanalytic procedure.

For instance, one example of a popular symmetric algorithm is AES. Until May 2009, the only successful published attacks against the full AES were side-channel attacks on some specific implementations. In December 2009 an attack on some hardware implementations was published that used differential fault

analysis. In November 2010, a published paper described a practical approach to a "near real time" recovery of secret keys from AES-128 without the need for either cipher text or plaintext. The first key-recovery attacks on full AES were published in 2011.

Another significant disadvantage of symmetric encryption is the key management required to use it securely. Each distinct pair of communicating parties must, ideally, share a different key, and usually each ciphertext exchanged as well. The number of keys required therefore increases in relation to the square of the number of network members.

Asymmetric encryption relies on mathematical problems that are thought to be difficult to solve, such as integer factorization or discrete logarithms. However there is no proof that a mathematical breakthrough could not occur which would make existing systems vulnerable to attack. Known asymmetric encryption methods are also computationally costly and slower compared with most symmetric key algorithms of equivalent security.

There are therefore disadvantages with both types of cryptography, and most practical encryption systems are therefore hybrid systems. A shared secret key, or session key, is generated by one party and this much shorter session key is then encrypted by each recipient's public key. Each recipient uses the corresponding private key to decrypt the session key. Once all parties have obtained the session key they can use a much faster symmetric encryption algorithm to encrypt and decrypt messages.

It is desirable to provide an improved method of encrypting data which is, or is closer to being, perfectly secure.

The conventional encryption of data involves encrypting data as a whole. This reduces the potential set of possible inputs. For instance, if an individual's bank

statement is encrypted, the output will be approximately the same size as the original bank statement. Furthermore, the security of a whole piece of data encrypted using a single algorithm depends upon that single algorithm not getting broken. One possible solution is to encrypt bits of files. However, this would require many passwords or algorithms.

Among symmetric key encryption algorithms, only the "one-time pad" has been proven to be secure, indeed perfectly secure, no matter how much computing power is available. In a one-time pad (OTP), each bit or character from the plaintext is encrypted by a modular addition with a bit or character from a secret random key of the same length as the plaintext, resulting in the ciphertext.

It has been proven that, if the key is truly random, as large as or greater than the plaintext, never reused in whole or part, and kept secret, the ciphertext will be impossible to decrypt or break without knowing the key. The method can be implemented as a software program, using data files as input (plaintext), output (ciphertext) and key data (the required random sequence). The XOR operation is often used to combine the plaintext and the key elements, since it is usually a native machine instruction and is therefore very fast.

However, practical problems have prevented one-time pads from being widely used. There must be secure generation and exchange of the key, which must be at least as long as the message. Also, importantly, sufficiently random numbers are difficult to generate using a computer. The random number generators in most programming languages are not suitable for cryptographic use. Even those generators that are suitable for normal cryptographic use involve cryptographic functions whose security is unproven.

It is desirable to provide an improved method of encrypting data which utilises the concept of the one-time pad but which overcomes one or more of the limitations of existing implementations.

According to the present invention there is provided a method of encrypting data comprising the steps of:

creating a one time pad;

encrypting the data using the one time pad to produce output data,

wherein the one time pad is generated using the data.

The method may include splitting the data into a plurality of data portions. The method may include taking a hash of each data portion.

The method may include obfuscating the data. The method may include obfuscating each data portion. The method may include obfuscating each data portion by concatenating the hashes of one or more other data portions.

The method may include encrypting the obfuscated data using the one time pad.

The one time pad may comprise key data which is generated by encrypting the data. The encryption process used to generate the key data may include one or more encryption parameters derived from the data. The one or more encryption parameters may be derived from one or more data portions. The encryption parameter may comprise an encryption key. The encryption parameter may comprise an initialisation vector.

The key data may be at least the same length as the data.

The encrypted data may be named using a hash of the encrypted data and then stored.

The method may include generating a data map for decrypting the output data. The data map may comprise the one or more encryption parameters.

The method may include generating a data atlas from a plurality of data maps. The data atlas may comprise a plurality of concatenated data maps.

The method may include removing duplicate information. The method may include at least reducing the number of multiple versions of identical data portions.

Embodiments of the present invention will now be described, by way of example only.

The present invention can provide a system of encryption that requires no user intervention or passwords. The resultant data item then has to be saved or stored somewhere as in all conventional methods. The encryption method of the invention relates to creating cipher-text (encrypted) objects that are extremely strong and closer to perfect in terms of reversibility, as opposed to known encryption ciphers. The method is based on symmetric encryption, and enhances this approach to produce highly secure data.

Within this specification, the following notation will be used:

H = Hash function such as SHA, MD5 or the like;
Symm = Symmetrical encryption such as AES, 3DES or the like;
PBKDF2 = Password-Based Key Derivation Function or similar;
f_c = file content;
f_m = file metadata;
fh = $H(f_c)$ or fh = $H(H(C_1)+H(C_2)+\ldots H(C_{n-1})$, where C_n is a data chunk;

The embodiment below will use AES as an example of a symmetric encryption algorithm and therefore will use a key and initialisation vector and plain-text input data.

Difficult to guess and uncompress-able output equates to random results based on random input data and random, unrelated algorithm inputs (plain text, key and iv in the case of modern symmetric ciphers).

The ideal cryptographic hash function has four main or significant properties. It is easy (but not necessarily quick) to compute the hash value for any given message; it is infeasible to generate a message that has a given hash; it is infeasible to modify a message without changing the hash; and it is infeasible to find two different messages with the same hash.

A cryptographically secure hash which is a one way function will create output that has a uniform distribution and can be computed in polynomial time. The output should be in fact random, although can be affected by size of input. Given a sufficiently large input the output will be random (within limits). The size of input required is dependent on the strength of the hash functions employed. In essence output can be considered evenly distributed and random. In cryptographically secure hashing, the data is analysed and a fixed length key called the hash of the data is produced. The hash cannot reveal the original data.

A hash function can be thought of as a unique digital fingerprint. However, it is possible to have two pieces of data with the same hash result. This is referred to as a collision and reduces the security of the hash algorithm. The more secure the algorithm, then the likelihood of a collision is reduced.

Early hash algorithms such as MD4, MD5 and even early SHA are considered broken, in the sense that they simply allow too many collisions to occur. Hence larger descriptors (keylengths) and more efficient algorithms are almost always required.

The following is one approach for carrying out the encryption method of the invention.

The data is split into a number of data portions or chunks (C_n). A hash of each chunk is taken (H_{cn}). In the case of AES or a similar cipher, [keysize] (C_{n-1}) is used as the key, and [next bytes iv size](C_{n-1}) is used as the IV (for AES 0 to 32 == key and 32 to 48 == iv).

Next, an obfuscation chunk ($OBFC_n$) is created by concatenating the hashes of other chunks ([unused part of](C_{n-1})(C_{n-2})(C_n).

An encryption cipher or similar reversible method is then run on (C_n), to produce random data (C_{random}).

The data can now be considered to be randomised and of the same length as the input data. The obfuscation chunk ($OBFC_n$) is also random output, but of a length less than the input data.

Next, the operation ($OBFC_n$)(repeated) XOR (C_{random}) is taken to produce the output data. Each of the output data is renamed with the hash of the new content and these hashes and saved.

A One Time Pad as defined by Shannon is regarded as the only cryptosystem with theoretically perfect secrecy. It presupposes the following: pads cannot be reused; for a Shannon implementation (as opposed to earlier cyclic pads) the pad must be as long as the message to be encrypted (i.e. a pad must be non-repeating); and the pad must contain only random data.

As the Shannon system suggests, a one time random pad which is longer than the data to be encrypted is required for a true one time pad. In this specification, a symmetric encryption cypher (AES as example, with CFB) is used to introduce

what can be described as randomness to the data itself. If this is truly random then it's the perfect pad in it's own right. Furthermore, an obfuscation pad is used, which almost creates a pad that is usable as a one time pad, however the pad is not as long as the message to be encrypted (it repeats as it is shorter than the data to be encrypted).

However, the data itself can be considered to be the pad and the obfuscation chunk is now repeating data (which is allowed by the definition of the Shannon Pad). Although this is a rather large amount of repeating data, it is also repeating random data. This can be considered as a form of one time pad. In addition, the actions taken on the data to include randomness as well as pad randomness result in increased security.

File Chunking

The size of the file (f.size()) is taken and the number (n) of chunks calculated. The number of chunks depends on the desired implementation, for instance a maximum number of chunks or a maximum chunk size may be desired.

Chunks of 256KB (settable) in length are created and then hashed. A hash of each chunk is taken, these are then hashed, and a structure is created which will be referred to as a data map.

The chunks are created with a fixed size to ensure that the set required to recreate the file is almost as large as the number of available chunks in any data store. This data map is mapped to the file metadata using fh.

Encryption Step

In the encryption stage, two separate non deterministic pieces of data are required: the encryption key (or password) and the Initialisation Vector (IV). To

ensure all data encrypts to the same end result, the IV is determined from what can be considered non deterministic data, that being the hash of one of the chunks.

Data is encrypted with the Key and IV ($Enc_{[key][IV]}(data)$). It is assumed that the Key and the IV for chunk n are derived from separate portions of the hash of chunk n-1. In the case of AES for instance, the first 32 bytes of this hash are the Key and the next 16 bytes are the IV ($Enc[H(C_{n-1\ [first\ 32\ bytes]})][H(C_{n-1\ [32\ to\ 48\ bytes]})][C_{Xn}) = C_{Xen}$).

Therefore, these items are selected from random data, although the randomness can be deterministic (if the output of an algorithm such as AES can be guessed, by guessing the input parameters, i.e. brute force) in the case of a one way function such as a cryptographic hash (as discussed).

The data is now represented as chunks of highly obfuscated chunks. The hash of each chunk is then taken again $H(C_{xen})$ and each chunk is renamed with the hash of its content.

Obfuscation Step

In the obfuscation step, each chunk is polluted with data from other chunks. For C_n, an identically-sized data chunk is created by repeatedly rehashing the hash of chunk n+2 and appending the result ($H(C_{n-2})+H(H(C_{n+2}))+H(H(H(C_{n+2}))) +\ldots$). This is called the XOR chunk n (CXORn) and is XOR'ed with chunk n. Although XOR has been used to obfuscate the data, this is not restrictive in any way and may be replaced by other obfuscation methods.

Data Map

Data maps are used to reverse the above process to retrieve the plain-text from the cipher-text chunks.

The encryption process can be reversed using data from the following steps that were described above: splitting the data into a number of chunks (C_n); [keysize] (C_{n-1}) as the key and [next bytes iv size](C_{n-1}) as the IV; and the obfuscation chunk ($OBFC_n$). This data is stored in a structure referred to as a data map. This is described in the following table.

$fh = H(H(C_1)+H(C_2)+\ldots H(C_{n-1})$	
$H(C_1)$	$H(C_{xe1})$
$H(C_2)$	$H(C_{xe2})$
…	…
$H(C_n)$	$H(C_{xen})$

In the above case, the hash of the concatenated pre-encryption hashes is used as the file hash. This is efficient in terms of processing time. However, the full file hash may be used.

With the above structure, the names of all the chunks are in the right hand column and all passwords and IV's (which are derived from the original chunk hashes) are stored in the left hand column. The file hash in the top row identifies the data element and acts as the unique key for this file.

Reversing the process is now straightforward. The chunks listed in right hand column are retrieved and each XOR chunk is created again. The obfuscation stage is reversed and each result decrypted. The results are concatenated.

This is the complete encrypt / decrypt process for each file.

The data maps (dm) from multiple files can be concatenated into a new structure referred to as the data atlas (da). Therefore, $dm_1 + dm_2 + \ldots = da$. This data atlas is itself now a large piece of data and is fed into the self-encryption process once more. This produces a single data map and more chunks. These chunks can be stored and the single remaining data map is the key to all the data.

The present invention allows for multiple data elements to be encrypted in a powerful fashion. All data is encrypted using no user information or input. This means that if the container for all the chunks is a single container then duplicate files will produce the exact same chunks and the storage system can automatically remove duplicate information. It is estimated the savings in data storage for such a system would be greater than 95%. Data compression could also be used during the hash/encryption of each chunk. This would further improve efficiency, particularly with regard to improving data de-duplication results.

Also, any break in an encryption cipher will not reveal any data to an attacker.

Whilst specific embodiments of the present invention have been described above, it will be appreciated that departures from the described embodiments may still fall within the scope of the present invention.

Claims

1. A method of encrypting data comprising the steps of:
creating a one time pad; and
encrypting the data using the one time pad to produce output data,
wherein the one time pad is generated using the data.

2. A method as claimed in claim 1, including splitting the data into a plurality of data portions.

3. A method as claimed in claim 2, including taking a hash of each data portion.

4. A method as claimed in any preceding claim, including obfuscating the data.

5. A method as claimed in claim 4 when dependent on claim 2 or 3, including obfuscating each data portion.

6. A method as claimed in claim 5, including obfuscating each data portion by concatenating the hashes of one or more other data portions.

7. A method as claimed in any of claims 4 to 6, including encrypting the obfuscated data using the one time pad.

8. A method as claimed in any preceding claim, wherein the one time pad comprises key data which is generated by encrypting the data.

9. A method as claimed in claim 8, wherein the encryption process used to generate the key data includes one or more encryption parameters derived from the data.

10. A method as claimed in claim 9 when dependent on claim 2 or 3, wherein the one or more encryption parameters are derived from one or more data portions.

11. A method as claimed in claim 9 or 10, wherein the encryption parameter comprises an encryption key.

12. A method as claimed in any of claims 9 to 11, wherein the encryption parameter comprises an initialisation vector.

13. A method as claimed in any of claims 8 to 12, wherein the key data is at least the same length as the data.

14. A method as claimed in any preceding claim, wherein the encrypted data is named using a hash of the encrypted data and then stored.

15. A method as claimed in any preceding claim, including generating a data map for decrypting the output data.

16. A method as claimed in claim 15 when dependent on any of claims 9 to 12, wherein the data map comprises the one or more encryption parameters.

17. A method as claimed in claim 15 or 16, including generating a data atlas from a plurality of data maps.

18. A method as claimed in claim 17, wherein the data atlas comprises a plurality of concatenated data maps.

19. A method as claimed in any preceding claim, including removing duplicate information.

20. A method as claimed in claim 19, including at least reducing the number of multiple versions of identical data portions.

21. A device for encrypting data comprising:

a processor which is configured to create a one time pad and to encrypt the data using the one time pad to produce output data,

wherein the processor is configured to generate the one time pad using the data.

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2013/050936

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/06
ADD.

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)
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 practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Alfred J. Menezes ET AL: "Handbook of applied cryptography, block ciphers" In: "Handbook of Applied Cryptography; [CRC PRESS SERIES ON DISCRETE MATHEMATICS AND ITS APPLICATIONS]", 1 January 1997 (1997-01-01), CRC Press, XP055032480, pages 233-282, figure 7.1 c -----	1,2,4,5, 7-10,12, 13,21
X	US 5 995 623 A (KAWANO KENJI [JP] ET AL) 30 November 1999 (1999-11-30) column 3, lines 8-34 column 12, line 11 - column 13, line 17; figures 6b,6c column 15, line 55 - column 16, line 15; figures 11D,11E ----- -/--	1,2,4,5, 7-10,12, 13,21

[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 application or patent 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
17 July 2013	25/07/2013

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, Fax: (+31-70) 340-3016	Manet, Pascal

INTERNATIONAL SEARCH REPORT

International application No.
PCT/GB2013/050936

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☒ Claims Nos.: 3, 6, 11, 14-20
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
see FURTHER INFORMATION sheet PCT/ISA/210

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2013/050936

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2010/086855 A2 (FORTRESS APPLIC LTD [IL]; GRESSEL CARMI DAVID [IL]; COURTOIS NICOLAS T) 5 August 2010 (2010-08-05) page 24, lines 4-11; figure 2c -----	1,2,4,5, 7-10,12, 13,21

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2013/050936

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5995623 A	30-11-1999	JP 3747520 B2	22-02-2006
		JP H09270785 A	14-10-1997
		US 5995623 A	30-11-1999
WO 2010086855 A2	05-08-2010	CN 102317904 A	11-01-2012
		EP 2382536 A2	02-11-2011
		US 2011286596 A1	24-11-2011
		WO 2010086855 A2	05-08-2010

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

Continuation of Box II.2

Claims Nos.: 3, 6, 11, 14-20

The application does not meet the requirements of Article 6 PCT, because claims 3, 6, 11, 14 to 20 are not clear. It is also found that claims 6, 11 and 15 to 20 lack of disclosure in the corresponding parts of the description (Article 5 PCT), no clear instructions are given in the description to carry out the invention.

Claim 3 indicates that a hash od
each data portion is taken but without using is, thus having no related technical effect.

Claims 3 to 6 apply operations to data portions
without indicating wether it is for use when the data is to be encrypted or when it is used to generate the one-time pad. Therefore, both can be understood.

Concatenating the hashes of data portion (claim 6) is not an
obfuscation as the data are not recoverable, the meaning of the claim is therefore so unclear that no opinion can be given.

Claim 11 adds that
encryption parameter comprises an encryption key. From the description, it is understood that this key is taken as the initial bytes of the previous plaintext, but for the first chunk no information is given, therefore leaving the reader in doubt regarding both the way to carry out the invention and also the technical effect. Page 10, lines 18-19 seems to indicate that the key in the top hash, but this can only be calculated with all the plaintexts so the recipient can't possibly compute it unless it is sent with the ciphertext in which case anyone can decrypt the whole data.

Claim 14 indicates that the encrypted data is named using a hash
of the encrypted data. A name is not a technical feature and it is also not clear what is meant. The description does not add any information on this aspect.

Claims 15 to 20 relate to a data map and data atlas, the
use of which is not clear for decryption. If the table given page 10 gives all password and IVs as indicated lines 16-19, then the whole scheme is not secure because anyone getting this map would decrypt the data.

The applicant's attention is drawn to the fact that claims relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure. If the application proceeds into the regional phase before the EPO, the applicant is reminded that a search may be carried out during examination before the EPO (see EPO Guidelines C-IV, 7.2), should the problems which led to the Article 17(2) declaration be overcome.