

(19)



(11)

EP 3 776 289 B9

(12)

CORRECTED EUROPEAN PATENT SPECIFICATION

(15) Correction information:

Corrected version no 1 (W1 B1)
Corrections, see
Description Paragraph(s) 18, 19

(51) International Patent Classification (IPC):

G06F 21/14 ^(2013.01) **G06F 17/16** ^(2006.01)
G06F 21/62 ^(2013.01) **G06F 21/32** ^(2013.01)
H04L 9/08 ^(2006.01)

(48) Corrigendum issued on:

17.01.2024 Bulletin 2024/03

(52) Cooperative Patent Classification (CPC):

G06F 21/14; G06F 21/32; G06F 21/6254;
H04L 9/085; H04L 9/0866; H04L 2209/16

(45) Date of publication and mention
of the grant of the patent:

18.10.2023 Bulletin 2023/42

(86) International application number:

PCT/GB2019/050858

(21) Application number: **19716488.2**

(87) International publication number:

WO 2019/186140 (03.10.2019 Gazette 2019/40)

(22) Date of filing: **26.03.2019**

(54) METHOD AND APPARATUS FOR DATA OBFUSCATION

VERFAHREN UND VORRICHTUNG ZUR DATENVERSCHLEIERUNG

PROCÉDÉ ET DISPOSITIF D'OBSCURCISSEMENT DE DONNÉES

(84) Designated Contracting States:

AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO
PL PT RO RS SE SI SK SM TR

(72) Inventors:

- **MAWDSLEY, Gary**
Manchester M3 3EB (GB)
- **TISHKOVSKY, Dmitry**
Manchester M3 3EB (GB)

(30) Priority: **26.03.2018 GB 201804818**
07.03.2019 GB 201903063

(74) Representative: **HGF**

HGF Limited
1 City Walk
Leeds LS11 9DX (GB)

(43) Date of publication of application:

17.02.2021 Bulletin 2021/07

(73) Proprietor: **Lockular Limited**

Kendal LA9 7RL (GB)

(56) References cited:

EP-A1- 3 015 988 WO-A1-2009/087764
WO-A1-2015/057854 JP-A- 2000 101 826
US-A1- 2016 253 514

EP 3 776 289 B9

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

DescriptionBackground

5 **[0001]** Increasing amounts of data is being stored. The stored data may relate to entities such as people and business. Particularly for such data privacy is a major concern. Regulations are also placing requirements on where data can be stored and in what form the data may be stored. Encryption is one solution to data privacy. However, encryption may not be as strong as desired. It is therefore often desired to obfuscate data.

10 **[0002]** WO 2015/057854 discloses a data collection procedure which can be performed for each subject of a study. The procedure transforms the data matrix X (of the participants' data) to AXB, where matrix A is a row operator that transforms data records (cases) in X and matrix B is a column operator that transforms data attributes (variables) in X, and the keys to generate these random operators are held separately by different parties.

15 **[0003]** EP 3 015 988 discloses systems and methods for transmitting data to at least one storage system. A request is received to store a data set in a storage location. In response, a first plurality of shares is generated, each containing a distribution of data from the data set, and at least one share is stored in a local memory configured for backup in at least one remote storage system. In some embodiments, the system comprises multiple geographically separated independent data storage systems.

[0004] It is an object of embodiments of the invention to at least mitigate one or more of the problems of the prior art.

Summary of Invention

[0005] According to aspects of the present invention there is provided methods and apparatus as set forth in the appended claims.

25 **[0006]** According to one aspect of the invention there is a computer-implemented method of enhancing data privacy, comprising determining one or more tensors of numeric data, transforming each of the one or more tensors of numeric data into at least two obfuscated fragments of data, and storing, separately, each of the at least two fragments of data.

[0007] Optionally, the method further comprises retrieving the at least two fragments of data, and performing a reverse transform on the at least two fragments of data.

30 **[0008]** According to another aspect of the invention, there is a computer-implemented method of pre-processing data comprising receiving non-normalised input data, translating the input data to at least one numeric data set and a dictionary, and converting the at least one numeric data set to column oriented form. Advantageously, numeric tensors can be extracted from the column oriented form for use in the method of enhancing data privacy. Optionally, the input data is in a table format. In further embodiments the input data is a document or image. The input data may be received in the form of a stream. The stream optionally comprises a series of data values.

35 **[0009]** Optionally, the method of enhancing data privacy comprises encrypting input data or data based on the input data to produce the one or more tensors.

[0010] Optionally, the transforming comprises applying a wavelet transform to each of the one or more tensors of numeric data to generate the at least two fragments of data. In another embodiment the transforming comprises applying a linear transform to a part of the numeric data. The reverse transform may in this case be the inverse linear transform.

40 **[0011]** In a further embodiment of the invention, the transforming may comprise both a wavelet transform and a linear transform.

[0012] Prior to transforming, the method may comprise balancing at least some of the one or more tensors of numeric data.

45 **[0013]** Optionally, the storing, separately, each of the at least two fragments comprises storing a first fragment of data at a first computer system and a second fragment of data at a second computer system.

[0014] According to an embodiment of the invention there is provided computer software which, when executed by a computer, is arranged to perform any of the embodied methods. The computer software may optionally be stored on a computer-readable medium. The computer-readable medium may be non-transitory.

50 **[0015]** According to one aspect of the invention there is a computer-implemented method, comprising receiving input data, determining one or more tensors of numeric data in dependence on the input data, transforming each of the one or more tensors of numeric data into at least two obfuscated fragments of data, and storing, separately, each of the at least two fragments of data. The method may comprise processing the input data to form the one or more tensors of numeric data.

55 **[0016]** According to one aspect of the present invention there is provided a computer-implemented method of enhancing data privacy, comprising determining, at a processor, one or more tensors of numeric data in dependence on input data; determining, at a processor, a transform in dependence on user-associated data; transforming, at a processor, each of the one or more tensors of numeric data into at least two fragments of data by applying the transform, wherein each of the fragments of data obfuscates the numeric data; and storing, separately, each of the at least two fragments of data

at a respective geographically separated storage system.

[0017] Advantageously, determining the transform in dependence on user-associated data allows each of a plurality of users to be associated with different data, and consequently a different transform. Each user being associated with a transform improves security and provides a decentralised system, wherein the user-associated data, and thus the user, is required to enable obfuscation and subsequent retrieval of data.

[0018] Optionally, the user-associated data comprises data indicative of an input received from a user via a user interface. Optionally, the user-associated data comprises biometric data associated with a user. For example, the biometric data may be indicative of one or more of: a fingerprint of the user, a heart rhythm of the user, an iris scan of the user, a facial structure of a user and DNA derived from the user.

[0019] According to one aspect of the present invention, there is provided a computing system, comprising one or more processors for operatively executing computer readable instructions; computer-readable data storage medium accessible to the one or more processors storing computer-readable instructions which, when executed by the one or more processors, perform a method comprising steps of: determining one or more tensors of numeric data in dependence on input data; determining a transform in dependence on user-associated data; transforming each of the one or more tensors of numeric data into at least two fragments of data by applying the transform, wherein each of the fragments of data obfuscates the numeric data; and storing, separately, each of the at least two fragments of data at a respective geographically separated storage system. In some embodiments, the computing system may comprise one computing apparatus. In other embodiments, the computing system may comprise two or more computing apparatuses, and the one or more processors may be distributed amongst the computing apparatuses. Advantageously, at least one of the computing apparatuses may be a user device such as a mobile phone. Thus, at least part of the computing system may be under physical user control. Obfuscation and subsequent reconstruction of the input data may then be decentralised, and data security improved.

Brief Description of the Drawings

[0020] Embodiments of the invention will now be described by way of example only, with reference to the accompanying figures, in which:

Figure 1 shows a system according to an embodiment of the invention;

Figure 2 shows a schematic illustration according to an embodiment of the invention;

Figure 3 shows a method according to an embodiment of the invention;

Figure 4 shows a method according to an embodiment of the invention;

Figure 5 shows a method according to an embodiment of the invention; and

Figure 6 shows a method of determining a transform according to an embodiment of the invention.

Detailed Description of Embodiments of the Invention

[0021] Figure 1 illustrates a system, generally denoted as 100, according to an embodiment of the invention. The system 100 is a system for obfuscating data to enhance privacy of the data, as will be explained.

[0022] The system 100 comprises a computer 130 on which a privacy engine 200 operatively executes. An embodiment of the privacy engine 200 is schematically illustrated in Figure 2. The computer 130 may be a server computer 130, although in other embodiments the computer 130 may be a computer operated by a user. The computer 130 will hereinafter be referred to as a Privacy Engine Computer (PEC) 130.

[0023] In some embodiments, the PEC 130 is communicable with a client computer 110. The client computer 110 operably provides data to the PEC 130 which it is desired to store in a privacy-enhanced form. By computer it is understood that one or both of the client computer 110 or computer 130 may be a mobile computer such as a mobile computing device e.g. a telephone, tablet or laptop computer in some embodiments.

[0024] In some embodiments, elements of the privacy engine 200 may be distributed between the PEC 130 and the client computer 110. Implementing elements of the privacy engine 200 within the client computer 110 can be beneficial for data security, as securely stored data may then only be reconstructed on the client computer 110 when accessed by a user of the client computer 110. Securely stored data then does not need to exist in a constructed form on the client computer 110 or the PEC 130 when not being directly accessed. Advantageously storing and reconstructing data in this manner changes the nature of how secure data is treated, in addition to changing the relationship between the client

computer 110 and the PEC 130 in comparison to conventional client relationships with a cloud-based data storage system. Fully reconstructed data is only resident when it is accessed by a user of the client computer 110, as will be explained further below, particularly with reference to method 400.

[0025] Each of the client computer 110 and PEC 130 may comprise one or more processors arranged to operably execute computer software thereon, where the computer software is stored in a computer-readable medium accessible to the one or more processors. The computer-readable medium may be one or more memory devices, where the memory may also store data for use by the software.

[0026] The client computer 110 and PEC 130 are communicably coupled by a computer network 120. The computer network 120 may comprise one or more networks such as Local Area Networks (LANs) and the Internet. The PEC 130 may provide an interface to receive data from the client computer 110. The data may be data which the client computer 110 wishes to store in the privacy-enhanced form as will be explained. The client computer 110 may execute a software application which is arranged to communicate with counterpart software executing on the PEC 130.

[0027] The system 100 further comprises a plurality of data storage systems. In the embodiments shown in Figure 1 the system comprises first and second data storage systems 140, 150. One or both of the first and second data storage systems 140, 150 may be geographically separated from the PEC 130. The PEC 130 and one or both of the first and second data storage systems 140, 150 may be communicably coupled over one or more networks such as the internet. The first and second data storage systems 140, 150 are separately operable to store data therein. Each of the data storage systems 140, 150 may comprise one or more data storage devices such as one or more of magnetic, optical or solid-state data storage devices. The PEC 130 is arranged to communicate data to each of the first and second data storage systems 140, 150 for storage therein. The data communicated to the first data storage system 140 by the PEC 130 is different to the data communicated to the second data storage system 150.

[0028] The first and second data storage systems 140, 150 may be located at separate geographic locations i.e. first and second data storage systems 140, 150 may be geographically separated. In particular, the first data storage system 140 may be located in a first territory and the second data storage system 140 located in a second territory. The first and second territories may have different rules or laws associated with the storage of data, particularly data relating to persons or entities, hereinafter personal data. Embodiments of the invention may allow data to be communicated from the PEC 130 to one or both of the first and second data storage systems 140, 150 without divulging or disclosing the personal data due to obfuscation of the data as will be explained. Whilst embodiments of the invention are described with reference to the first and second data storage systems 140, 150 for convenience, it will be appreciated that embodiments of the invention are not limited in this respect and that the system 100 may comprise more than two data storage systems 140, 150.

[0029] As noted above, Figure 2 schematically illustrates the privacy engine 200. The privacy engine 200 may be distributed among one or more processors across the PEC 130 and the client computer 110. The privacy engine 200 has operatively executing thereon an obfuscation module 210 and a data storage module 220. In some embodiments, the privacy engine 200 has operatively executing thereon one or both of a pre-processing module 230 and an encryption module 240. The modules 210 to 240 may be distributed across the PEC 130 and the client computer 110. For example, in some embodiments, all the modules 210 to 240 operatively execute on the PEC 130. In other embodiments, one or more of the modules, for example the pre-processing module 230 and the obfuscation module 210, may at least in part operatively execute on the client computer 110.

[0030] The obfuscation module 210 is arranged to operatively transform first data into a plurality of fragments of data, wherein each of the fragments of data obfuscates the first data. By obfuscate it is meant that the first data cannot be obtained with certainty from a subset of the fragments of data. That is, both knowledge of the obfuscation process and possession of all of the fragments of data is necessary to re-obtain the first data with certainty that the correct first data has been re-obtained, as will be explained.

[0031] A fragment of data may be referred to as a frade, and the process of producing the fragments may be referred to as frading.

[0032] The data storage module 220 is arranged to receive, from the obfuscation module 210, the plurality of fragments of data and to determine which of the plurality of data storage systems 140, 150 will store each respective fragment of data. The data storage module 220 is arranged to distribute storage of the plurality of fragments of data amongst the plurality of data storage systems such that the plurality of fragments are not all stored on the same data storage system 140, 150. In some embodiments, each of the plurality of fragments of data are each stored on a respective one of the plurality of data storage systems 140, 150. For example, a first fragment of data is stored on the first data storage system 140 whilst a second fragment of data is stored on a second data storage system 150. Thus, in some embodiments, the data storage module 220 receives and distributes storage of the fragments of data amongst the plurality of data storage systems 140, 150.

[0033] As noted above, in some embodiments the system comprises one or both of the pre-processing module 230 and the encryption module 240. The pre-processing module 230 is arranged to perform one or more operations on data prior to the data being provided to the obfuscation module 210. In particular, the pre-processing module 230 may convert

data into a format suitable for being operated on by the obfuscation module 210. The encryption module 240 may encrypt the data prior to the data being provided to the obfuscation module 210. The encryption may be performed according to a predetermined encryption algorithm or one of a plurality of encryption algorithms selected by a user.

[0034] Figure 3 illustrates a method 300 according to an embodiment of the invention. The method 300 is a computer-implemented method of enhancing data privacy. The method 300 may be performed by the PEC 130, or a combination of the PEC 130 and the client computer 110. In some embodiments, at least some of steps forming the method 300 may be performed by the obfuscation module 210 and the data storage module 220.

[0035] The method 300 comprises a step 310 of receiving data. The data may be received at the privacy engine 200 from the client computer 110. In some embodiments the data may be a file representative of, for example, an image, video data, a document such as a Microsoft Word (RTM) or PDF file (although embodiments are not limited to the type of document). In other embodiments the data may be at least a portion of structured data for storing a data set. For example, the data may be at least a portion of a relational data set as will be explained below particularly with reference to further explanation of the pre-processing module 230. For ease of understanding, an embodiment of the invention will be explained where the data received in step 310 is numeric data with it being understood that embodiments of the invention are not limited in this respect. For example, the data may be alphanumeric data. In the illustrated embodiment the received numeric data comprises a series of numeric values.

[0036] The method 300 in some embodiments comprises a step 320 of processing the data received in step 310. As noted above, in the illustrated embodiment step 310 comprises receiving a series of numeric values. In step 320 the received numeric values may be processed to form one or more tensors of numeric data e.g. as a vector. The tensor comprises N numeric values where N is an integer greater than one. Thus step 320 may comprise dividing or allocating the received data amongst a plurality of tensors.

[0037] An example tensor where $N=3$ determined in step 310 is:

[1517270400 784 785]

[0038] It will be understood that the numeric values are merely an example.

[0039] Step 320 may also optionally comprise pre-encrypting the data set by the encryption module 240, according to an encryption algorithm. The encryption algorithm used may be an encryption algorithm known to the skilled person, for example 256-bit AES encryption, although other encryption algorithms and techniques may be used. It is appreciated that this pre-encryption can be performed on the resultant tensors, or on the original data received in step 310 prior to the other operations performed in step 320.

[0040] Further operations performed in step 320 may be undertaken particularly by one or both of the pre-processing module 230 and the encryption module 240 and thus the one or more operations performed in step 320 may be omitted in some embodiments.

[0041] The method 300 comprises a step 330 of transforming the data to form a plurality of fragments of data.

[0042] In some embodiments, an initial part of step 330 may comprise determining or selecting an appropriate obfuscation process. The determination of the obfuscation process may comprise receiving an indication from the client computer 110 selecting one of a plurality of obfuscation processes supported by the PEC 130. The determination of the obfuscation process may comprise determining a transform to be used in transforming the data, as will be explained. In example embodiments a first obfuscation process is a wavelet-based obfuscation process utilizing a wavelet transform and a second obfuscation process is based on alternative transform data, as will be explained. The indication may provide a selection of one of the first and second obfuscation processes. In other embodiments, the obfuscation process may be provided to the PEC 130, such as from the client computer 110. The obfuscation process may be provided in the form of a set of instructions for carrying out the obfuscation process i.e. to form an algorithm by which the obfuscation is carried out. The obfuscation process may be provided in the form of bytecode communicated to the privacy engine 200. In some embodiments a wavelet is communicated from the client computer 110 to the privacy engine 200. In other embodiments, data for use in the obfuscation process, such as the transform data, is communicated from the client computer 110 to the privacy engine 200.

[0043] According to some embodiments of the present invention, a transform for use in step 330 may be determined in dependence on user-associated data. In this way, each of a plurality of users may be associated with different user-associated data, and consequently, a different transform.

[0044] A method 600 for determining the transform in dependence on user-associated data is illustrated in Figure 6. The method 600 may be performed as part of step 330 on the privacy engine 200. In other embodiments, the method 600 may be performed independently of method 300. For example, the method 600 may be performed prior to the method 300.

[0045] The method 600 comprises a step 610 of receiving or retrieving user-associated data to be used for determining the transform. The user-associated data may either be received from a user during step 610 or may have been determined previously and stored, for example on client computer 110. The user associated data may comprise one or both of biometric data associated with a user, or data indicative of an input received from a user, i.e. user input data.

[0046] For example, the user input data may comprise data indicative of an input received from a user via a user

interface. The user interface may receive an indication of a physical, audible or gesture input from the user. The user input data may then comprise data indicative of the received physical, audible or gesture input. In some embodiments, the user input data may comprise data indicative of a passphrase, i.e. alphanumeric string, provided verbally or via touch input from the user.

[0047] Additionally or alternatively to input received from a user, the user-associated data may comprise biometric data associated with a user. For example, the biometric data may be indicative of one or more biometric measurements of a user. Biometric measurements utilised in embodiments of the present invention may include: a measurement of a fingerprint of the user, for example received from a fingerprint scanner; a measurement of a heart rhythm of the user for example from an ECG; a measurement taken from an iris scan or a retinal scan of the user, a measurement of facial structure of the user from a facial recognition system, or a measurement obtained from a DNA sequencing system utilising user DNA. Step 610 may comprise receiving the one or more biometric measurements from the user, for example by communicating with one or more biometric measurement systems, such as the fingerprint scanner, facial recognition system, ECG system or DNA sequencing system. Alternatively, the biometric measurement may have been taken independently of method 600 and stored in a memory, and step 620 may then comprise retrieving the biometric data from the memory. It will be appreciated that the biometric measurements and biometric measurement systems are not limited to this list, and any biometric measurement related to a measurable human characteristic may be used to derive the biometric data.

[0048] Method 600 may comprise a step 620 of determining a user key K from the user-associated data. The user key K is uniquely associated with the user, and may be used to generate transforms for obfuscation and retrieval of data, as will be explained.

[0049] In an illustrated example of step 620, the user key K is generated from biometric data associated with the user, although it will be appreciated that in some embodiments step 620 may comprise generating the user key K from user input data such as a string indicative of a passcode.

[0050] As discussed, the biometric data received in step 610 may be associated with one or more biometric measurements. Each biometric measurement can be represented as one or more samples of a feature vector $X = (x_1, \dots, x_N)$. The feature vector X comprises a plurality of features x_i , which may each be an extracted parameter or principal component of the biometric measurement. For example, if the biometric measurement is an ECG measurement the biometric data may comprise a plurality of features associated with the ECG measurement. In some embodiments the features x_i comprise extracted parameters such as RQ-amplitude, QS-duration, RS-amplitude, ST-amplitude, QT-duration or any other known parameter of the ECG data. It will be appreciated that other features may also be extracted. Analogously, known features may be extracted from other biometric measurements such as a fingerprint scan, DNA sequencing, facial recognition etc. The skilled user will appreciate that for any biometric measurement a variety of parameters may be selected as features x_i , for example one or more principal components derived from a training set.

[0051] In some embodiments, step 620 may comprise serialising the feature vector X into an array of bytes to determine the user key K. The user key K may be determined analogously in step 620 in embodiments where other user-associated data is used. For example, if the user-associated data is a string indicative of a passcode, step 620 may comprise serialising the string into an array of bytes to determine the user key K.

[0052] It may be desired to determine the user key K such that utilizing two biometric measurements from the same individual taken at different time points will each result in determining the same user key K with a sufficiently high probability, i.e. to produce a stable biometric user key K. A stable biometric user key K may be generated from a plurality of samples of each feature x_i of the biometric measurement. In some embodiments of the present invention, step 620 comprises providing such a stable biometric user key K, as follows.

[0053] For a given biometric measurement, predetermined population data may be accessible by the privacy engine 200 during step 620. The predetermined population data may comprise information indicative of the distribution of each feature i over a predetermined population, for example by indicating one or more generalised attributes such as a mean

μ_i^g and a standard deviation σ_i^g estimate of each feature i over the predetermined population. In some embodiments, the predetermined population data may comprise a large enough number of training samples of each feature i such that the distribution of each feature i over the population may be derived with a reasonable degree of accuracy. Step 620

may then comprise determining a mean μ_i^g and a standard deviation σ_i^g estimate of each feature i over the predetermined population data.

[0054] Each biometric measurement received in step 610 may comprise M samples of the feature vector X. Step 620 may then comprise deriving a mean μ_i and a standard deviation σ_i estimate of each feature i for the M samples received from the user. For example an ECG measurement may comprise a plurality, such as 5, heartbeats ($M = 5$) from which each of the features i outlined above may be extracted.

[0055] Step 620 may comprise defining, for each feature i , a parameter k_i^g indicative of an interval of values such that the probability of any measurement of the feature i within the population to fall outside the interval is sufficiently

small. For example, each k_i^g may be defined such that the probability of any measurement of the feature i within the population to fall outside the interval $(\mu_i^g - k_i^g \sigma_i^g, \mu_i^g + k_i^g \sigma_i^g)$ is lower than a predetermined threshold. For

example, if the distribution of the feature i over the population is close to normal, the parameters k_i^g may be set at 4 or 5, although it will be appreciated that other values can be assigned.

[0056] Step 620 may comprise defining, for each feature i , a parameter k_i indicative of the distinguishability of each feature i for the M samples received from the user. Each k_i may be selected and adjusted for the user in advance. In some embodiments, each k_i is selected such that an appropriate function $F(k)$ is maximised, wherein $F(k)$ increases with entropy for generated user keys K and decreases with the number of false negative outcomes on the feature i .

[0057] Step 620 may then comprise determining the user key K in dependence on the parameters k_i , k_i^g , μ_i^g , σ_i^g , μ_i , and σ_i . That is, step 620 may comprise determining the user key K in dependence on the distribution of each feature i for the user's biometric data in comparison to the population distribution of each feature i . Advantageously, determining the user key K in dependence on these parameters allows generation of the same user key K from two biometric measurements from the same user with sufficiently high probability. This advantageously allows decentralisation of data access and improved data security.

[0058] For example, in one embodiment of the invention, the user key K is defined as a bit string:

$$K = c_1 d_1 c_2 d_2 \dots c_N,$$

$$c_i = \begin{cases} 2^{n_i}(n_i + j), & \mu_i - j \cdot (k_i \sigma_i) > \mu_i^g, j = 1, \dots, n_i; \\ 2^{n_i}(n_i - j + 1), & \mu_i - j \cdot (k_i \sigma_i) < \mu_i^g, j = 1, \dots, n_i; \\ \text{empty}, & \text{otherwise.} \end{cases}$$

$$n_i = \left\lceil \frac{k_i^g \sigma_i^g}{2(k_i \sigma_i)} \right\rceil$$

$$d_i (i = 1, 2, \dots, N - 1) = 2^{n_i} (2^{l(2n_i)-1})$$

[0059] Wherein for an integer a , $l(a)$ is defined as:

$$l(a) = \lceil \log_2(a + 1) \rceil + 1$$

[0060] And for integers a and b , b^a denotes the number a written as a bit string of length $l(b)$ bits representing a in binary.

[0061] It will be appreciated that the above illustrates only one possible user key K according to embodiments of the

present invention, and alternate keys utilising the parameters k_i , k_i^g , μ_i^g , σ_i^g , μ_i , and σ_i can also be envisaged.

[0062] Method 600 comprises a step 630 of determining the transform in dependence on the user key K . The determined transform may be utilised within the method 300 to transform the data in step 330.

[0063] Advantageously utilising a transform determined by method 600 to transform the data improves security of the data transformed and stored according to method 300. As the transform determined by method 600 is based on user-associated data, the user-associated data is required to determine either the transform or the corresponding reverse transform required to reconstruct the data according to a method 400. The user-associated data does not need to be stored within the system 100, and can be produced by the user each time the user desires to store or access data according to method 300 or 400. For example, the user input or biometric measurement may be provided by the user

each time the data is accessed or stored, ensuring the data cannot be reconstructed without the user providing said data. Such embodiments of the present invention provide a decentralised aspect to data, thus improving security and user agency.

[0064] Step 630 will be described in conjunction with step 330 with reference to illustrated embodiments, wherein the determined transform may be a wavelet transform, or a linear transform. In embodiments wherein the determined transform is a wavelet transform, the tensor of numeric data is transformed into the plurality of fragments which obfuscate the numeric data by application of a Wavelet to the tensor. In embodiments wherein the determined transform is a linear transform, the tensor of numeric data is transformed into the plurality of fragments obfuscating the numeric data by a linear transformation.

[0065] Referring first to the embodiments comprising application of the wavelet to the tensor, a mother wavelet, i.e. a wavelet transform, is selected. Step 630 may comprise determining the mother wavelet in dependence on the user key K. For example, the user key K may be used to generate a function $f(x)$ to be used as the mother wavelet.

[0066] In certain embodiments the mother wavelet is a discretely sampled wavelet, though continuous wavelets may also be envisaged. It will be understood that a wavelet is any small wave which in itself does not repeat. Application of the wavelet to the tensor comprises fitting the wavelet to the tensor. The amplitude and frequency of a sampled part of the wavelet are adjusted such that the resultant part of the wavelet is a good fit to a part of the tensor, and the adjustments to the frequency and amplitude are stored. Advantageously, the original tensor can be reconstructed from the mother wavelet and the adjustments to the frequency and amplitude.

[0067] In an illustrated embodiment the mother wavelet is described as a Haar wavelet with it being appreciated that embodiments are not limited in this respect. For instance, the mother wavelet may be a Daubechies wavelet, a Symlet wavelet, a Coiflet wavelet or a Shannon wavelet in other embodiments. This is a non-exhaustive list of possible mother wavelets, as will be appreciated.

[0068] For some wavelets an input tensor is required to be a certain length. For example, the input tensor may be required to be even (comprise an even number of values) i.e. for N to be an even value i.e. 2, 4, 6 etc. Thus step 330 may comprise balancing at least some of the one or more tensors of numeric data prior to performing the transforming. Balancing is understood to mean padding the tensor of numeric data to comprise a larger number of values. For example, balancing may comprise padding N from an odd value to an even value. For example, the example tensor above is balanced to:

[1517270400 784 785 0]

[0069] The balancing may be performed by inserting a numeric value into the tensor. The inserted numeric value may be zero, as illustrated above, although other values may be inserted.

[0070] In some embodiments, the wavelet processing comprises at least one level. In an illustrated example, the wavelet comprises five levels although it will be appreciated that other numbers of levels may be used. Applying the wavelet to the tensor, as above, yields two sets of values, a first set of wavelet coefficients and a second set of ordering values. The sets of values form the fragments of data. A first fragment comprises the plurality of wavelet coefficients. A second fragment comprises the plurality of ordering values. Example fragments are shown below where the values are scaled for convenience, in this case by $1.0E+9$. The fragments below are of equal length, however the invention is not limited in this respect and the length of each fragment will depend on the mother wavelet chosen.

$$\text{Fragment1} = [2.1457 \ 0 \ 0 \ 0 \ 07586 \ 1.0729 \ 0.0000]$$

$$\text{Fragment2} = [1 \ 1 \ 1 \ 1 \ 1 \ 2 \ 4]$$

[0071] As shown above, neither fragment contains directly data from the original tensor. Thus the fragments each obfuscate the original tensor data.

[0072] In a further illustrative embodiment of the invention the original tensor comprises 8 values:

$$\text{Original} = [2, 2, 0, 2, 3, 5, 4, 4]$$

[0073] The mother wavelet is a Haar wavelet. At each level of the at least one level, the wavelet processing comprises computing an array of pairwise averages of the original tensor:

$$\text{Averages1} = [2, 1, 4, 4]$$

[0074] The wavelet processing further comprises, at each of the at least one level, storing detail coefficients. The detail coefficients may, for example, be the differences of the second of each pair from the pairwise average:

$$5 \quad \text{Detail1} = [0, -1, -1, 0]$$

[0075] The process is repeated on the array of pairwise averages at each subsequent level of the plurality of levels. For example, if the process is repeated for the maximum number of levels for this tensor:

$$10 \quad \text{Averages2} = [3/2, 4]$$

$$15 \quad \text{Detail2} = [1/2, 0]$$

$$\text{Averages3} = [5/4]$$

$$20 \quad \text{Detail3} = [-5/4]$$

[0076] Averages3 may be used as the first fragment, and a vector comprising the Detail coefficients may be used as the second fragment.

[0077] In other embodiments, the wavelet processing comprises a plurality of scaling linear transformations. The scaling linear transformations are applied to portions of the wavelet curve, in order to fit it to the tensor. The scale of each linear transformation will match the amplitude of the curve. The fragments will then comprise information indicative of the scaling linear transformations performed.

[0078] In other embodiments of the invention, step 330 comprises transforming the tensor into a plurality of fragments which obfuscate the original tensor data by a linear transformation.

[0079] In these embodiments, step 630 may comprise determining a linear transform in dependence on the user key K. The linear transform may be representable as an invertible MxM matrix. In some embodiments the user key K may be utilised to determine the MxM matrix unambiguously, i.e. such that utilising the same user key K always results in the same MxM matrix. This can be achieved in a variety of ways as will be appreciated by the skilled person, and embodiments of the present invention are not limited to any particular technique.

[0080] As one example, the user key K may be utilised to generate a hash H of bytes of a fixed length, and H may be used as an initialisation parameter for a pseudo-random number generator G which may be used to generate an MxM matrix. As a second example, the user key K may be used in a hashing algorithm to produce a hash H of length 8Mx8M, and the hash H may then be rewritten as an MxM matrix of 8-byte numbers. Any hashing algorithm may be used to generate the hash H.

[0081] The linear transform may then be used to transform the data in step 330. In an illustrated example, the tensor of numeric data is a rank 1 tensor. As described above, the tensor may be balanced. The tensor may be balanced such that N is divisible by the number of desired fragments. For example, if two fragments are desired, the tensor is balanced to an even length, and if three fragments are desired, the tensor is balanced to a length divisible by three. Different numbers of desired fragments, and thus balancing N to alternate values, can also be envisaged. In the present example, the tensor is balanced such that N=4:

[1517270400 784 785 0]

[0082] The value of N may be selected to be divisible by a number of fragments without remainder. In the described embodiment the number of desired resultant fragments is two although other numbers of fragments may be chosen.

[0083] Step 330 comprises dividing the tensor into a plurality of parts. The length of the parts is selected in dependence on how many resultant fragments are desired from the transformation. For example, if two resultant fragments are desired, the tensor is split into first and second parts, each of length two:

$$55 \quad \text{Part1} = [1517270400 \ 784]$$

$$\text{Part2} = [785 \ 0]$$

[0084] It is appreciated that the tensor is not restricted to being divided into two parts, but rather the tensor may be split into more than two parts, with the parts all having equal dimension. Each part of the tensor may have the same number of values as there are desired fragments. In the illustrated embodiment, two fragments are desired, and so each part of the tensor has length 2.

[0085] Embodiments of the invention utilise transform data. The transform data is an invertible tensor transformation which can be numerically represented. For example, the transform data may be an invertible $M \times M$ rank 2 tensor, with matrix rank M . In some embodiments, the transform data is determined from the user key K in step 630, as has been explained. In some embodiments, M is chosen to be equal to the number of desired fragments, which is also the length of the data parts. For example, in the above case the transform data is selected to be an invertible 2×2 rank 2 tensor:

$$\text{Transform} = \begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix}$$

[0086] It is appreciated that any other invertible 2×2 rank 2 tensor may alternatively be chosen or determined as the transform data, for example by performing method 600.

[0087] The first part of the tensor (Part1) is transformed into a first transformed tensor in dependence on the transform data.

[0088] The transformation of the first part of the tensor into a first transformed tensor may be achieved by applying an operation on the part of the tensor and the transform data. For example, the operation may comprise multiplying the transform data by the first part of the tensor. It is appreciated that other operations aside from multiplication may be used. In the case of multiplication, transforming the first part of the tensor above in dependence on the transform data yields:

$$\begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix} * [1517270400 \quad 784] = [7586353568 \quad 3034541584] = \text{Trans1}$$

[0089] The second part of the tensor is transformed into a second transformed tensor in dependence on the transform data. As with the first part of the tensor, the transform may comprise multiplying the second part of the tensor by the transform data. In the illustrated example, the transformation of the second part of the tensor yields:

$$\begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix} * [785 \quad 0] = [3925 \quad 1570] = \text{Trans2}$$

[0090] The transformed tensors from the illustrated example are shown below.

$$\text{Trans1} = [7586353568 \quad 3034541584]$$

$$\text{Trans2} = [3925 \quad 1570]$$

[0091] The fragments of data are constructed in dependence on the two transformed tensors. In some embodiments, the first fragment of data is constructed by creating a vector from the first entry in each transformed tensor. The second fragment of data may be constructed by creating a vector from the second entry in each transformed tensor. In further embodiments, each further fragment of data may be constructed in an analogous way, by constructing a vector from a further entry of each transformed tensor. For example, two fragments of data can be formed from the two transformed tensors:

$$\text{Fragment1} = [7586353568 \quad 3925]$$

$$\text{Fragment2} = [3034541584 \quad 1570]$$

[0092] As shown above, neither fragment contains directly data from the original tensor. Thus, the fragments each

obfuscate the original tensor data.

[0093] In alternate embodiments wherein the input data has been divided into more than two parts, each further part of the tensor may be transformed in dependence on the transform data to yield a further fragment.

[0094] It is appreciated that the transform data may be other dimensions than a 2x2 rank 2 tensor. For example, the transform data may be a 3x3 or 4x4 rank 2 tensor, although embodiments are not limited in this respect.

[0095] In some embodiments of the invention the method 300 comprises a step 340 of storing the obfuscated data. The obfuscated data is the output of transforming step 330, and may comprise two or more fragments. In some embodiments, step 340 is performed by the data storage module 220. Data storage module 220 receives the two or more fragments, such as from the obfuscation module, as created in step 330. Step 340 comprises distributing the fragments amongst a plurality of data storage systems, for example data storage systems 140 and 150, such that all fragments of data are not stored on the same data storage system. In some embodiments the plurality of fragment are distributed each to a respective data storage system 140, 150. The first fragment (Fragment1) may be provided to the first data storage system 140 for storage therein and the second fragment (Fragment2) may be provided to the second data storage system 150 for storage therein.

[0096] In some embodiments, step 340 also comprises storing spurious, or fake, fragments of data as well as the obfuscated data. A spurious fragment is understood to be a fragment not derived from the original data, but created to mimic the characteristics of a real fragment. Spurious fragments may be used to improve obfuscation of the original data.

[0097] In some embodiments, a spurious fragment may be stored for every real fragment stored, although different numbers of spurious fragments may also be envisaged. Each spurious fragment may be generated with equal dimension and order of magnitude to a real fragment. The spurious fragments may be entirely or in part generated by, for example, random or pseudo-random number generation. The spurious fragments may be generated at the data storage module 220.

[0098] Each spurious fragment may be stored in the same data storage system as a real fragment, or may be stored in a separate data storage system. For instance, 2 spurious fragments (Spurious 1 and Spurious2) may be generated. The first data fragment and first spurious fragment (Fragment1 and Spurious1) may be provided to the first data storage system 140 for storage therein. The second data fragment and second spurious fragment (Fragment2 and Spurious2) may be provided to the second data storage system 150 for storage therein. In another embodiment a third data storage system (not shown) may store one or more spurious fragments alone i.e. without a data fragment also being stored on the same data storage system.

[0099] In some embodiments, step 340 also comprises a step of storing metadata. The metadata may comprise a schema for each input data set, which contains information about the input data. For instance, the schema may define the columns and data types of an input relational database. The metadata may further contain information about the locations of the real and spurious data fragments i.e. which data storage system stores the fragments. Step 340 may comprise a step of obfuscating the metadata before it is stored. The step of obfuscating the metadata may comprise any of the methods so far discussed, or the metadata may be obfuscated in an alternate way.

[0100] Figure 4 illustrates a method 400 according to an embodiment of the invention. The method 400 is a computer-implemented method of reconstructing data from obfuscated fragments of data. By reconstructing it is meant that a plurality of fragments of data are used in a reverse obfuscation process to obtain an original tensor of numeric data. The method 400 may be performed, in some embodiments, by the PEC 130 or in other embodiments by the client computer 110. In some embodiments, at least some of steps forming the method 400 may be performed by the obfuscation module 210 and the data storage module 220 as will be appreciated.

[0101] The method 400 comprises a step 410 of retrieving obfuscated data. The obfuscated data comprises a plurality of fragments created by method 300. As described above in connection with Figure 3, the plurality of fragments are distributed amongst the plurality of data storage systems, for example data storage system 140 and data storage system 150. Each fragment of data is retrieved from the respective data storage system 140, 150 at step 410. The metadata may also be retrieved at step 410, or the metadata may already be accessible by the PEC 130 or client computer 110.

[0102] In step 420 it is determined whether all of the fragments corresponding to an original tensor of data have been retrieved. If not, the method returns to step 410 wherein a further fragment of data is retrieved before another check is performed in a further iteration of step 420. Once it is determined in step 420 that all fragments have been retrieved the method moves to step 430. Determining that all fragments have been retrieved may comprise checking the retrieved fragments against information contained in the metadata.

[0103] The method 400 further comprises a step 430 of performing a reverse transform on at least one of the fragments of data.

[0104] An initial part of step 430 may, in some embodiments, comprise selecting an appropriate reverse obfuscation process, in particular selecting an appropriate reverse transform operation. Where the obfuscation process used to generate the fragments of data comprised application of a wavelet to a numeric tensor at step 330, a consistent reverse obfuscation process is selected. Alternatively where the obfuscation process used to generate the fragments of data comprised application of a linear transformation to a numeric tensor at step 330 a consistent reverse obfuscation process

is selected.

[0105] In the case of the data fragments being created by the application of Wavelet transform, the reverse transform step 430 comprises applying an inverse Wavelet transform. The inverse Wavelet transform is dependent on the wavelet used in step 330, which may for example be a Haar wavelet, although as will be appreciated, other wavelets may be used. The inverse wavelet transform is defined to reverse the process of the wavelet transform used in step 330.

[0106] In the case of the data fragment being created by the application of a linear transformation, step 430 comprises reconstructing the transformed tensors from the fragments of data and performing the reverse transform on the transformed tensors in dependence on reverse transform data.

$$\text{Fragment1} = [7586353568 \quad 3925]$$

$$\text{Fragment2} = [3034541584 \quad 1570]$$

[0107] The transformed tensors may be reconstructed from the data fragments by reversing the construction of the data fragments performed in step 330. This may comprise reconstructing a first transformed tensor from the first element of each data fragment, and reconstructing a second transformed tensor from the second element of each data fragment. This produces the transformed tensors:

$$\text{Trans1} = [7586353568 \quad 3034541584]$$

$$\text{Trans2} = [3925 \quad 1570]$$

[0108] In some embodiments, the reverse transform data is defined in dependence on the transform data used in step 330. According to an illustrated embodiment, the transform data is an invertible 2x2 rank 2 tensor. The reverse transform data may be determined by computing the matrix inverse of the transform data, which for the illustrated example yields the rank 2 tensor below.

$$\text{Reverse transform data} = \begin{bmatrix} 1 & -2 \\ -2 & 5 \end{bmatrix}$$

[0109] In embodiments where step 330 comprised multiplying a tensor part by transform data to obtain a transformed tensor, then the reverse transform comprises multiplying the reverse transform data by the transformed tensor. This operation retrieves the data part from which the data fragment was transformed. In the illustrated example, step 420 comprises multiplying the reverse transform data by the transformed tensor, as illustrated below.

$$\begin{bmatrix} 1 & -2 \\ -2 & 5 \end{bmatrix} * [7586353568 \quad 3034541584] = [1517270400 \quad 784] = \text{Part1}$$

[0110] In this example the reverse transform is applied to one transformed tensor, although it is appreciated that step 420 may comprise applying a reverse transform to a plurality of transformed tensors. That is, the reverse transform is applied to each of the transformed tensors.

[0111] Advantageously, if one or more modules of the privacy engine 200 are incorporated into the client computer 110, step 420 may be performed locally on the client computer 110 only when a user requires access to the original data fragments. In this way the reconstructed data may only exist at a time at which it is accessed by a user of the client computer 110, and so the reconstructed data will not exist outside the client computer 110, for example on PEC 130. In these embodiments, the reconstructed data will only exist at a time at which a user of the client computer 110 is directly accessing it. Thus, the nature of cloud-based data access is made significantly more secure.

[0112] Figure 5 illustrates a method according to an embodiment of the invention. The method may be performed as part of step 320 in some embodiments of the invention.

[0113] In some embodiments, the data received in step 310 will not be a series of numeric values. Figure 5 illustrates further operations performed on the data received in order to form the one or more tensors. For example, the data received in step 3210 may be a portion of a relational data set. The relational data set may contain a mixture of numeric,

string and date datatypes, although the possible datatypes are not limited in this respect. An example relational data set according to an embodiment of the invention is shown below.

Date	Name	Value
15/03/2018	ABC	432
15/03/2018	DEF	658
15/03/2018	GHI	5
16/03/2018	DEF	7654

[0114] Step 3240 comprises a step of converting non-numeric entries in the relational data set to numeric entries. In the case where there are date entries, this may comprise converting the date entries into number of seconds since 1970 epoch, although any other numeric unit could alternately be used. If the relational data set contains string values, the string values are converted to corresponding numeric values and a conversion relationship between the string values and numeric values may be stored in a dictionary table. The relational data set has consequently been decomposed into a numeric data set and a dictionary table. In a case where the relational data set already comprises numeric values, step 3240 may comprise performing an analogous operation of converting the numeric value to a corresponding converted numeric value, and a conversion relationship between the numeric values and converted numeric values may be stored in a dictionary table. Step 3240 may advantageously improve pseudonymisation of the relational data set.

[0115] Step 3250 comprises applying one or more normalisation techniques to the data set. This may comprise applying linear algebra QR decomposition to the data set of columns, in order to determine dependencies between columns, although it is appreciated that other normalisation techniques can be used. Step 3250 may further comprise dividing the data set into a plurality of smaller data sets. When reference is made to performing subsequent operations on the relational data set, this will be understood to mean either the relational data set received in 3210 or a data set resulting from the division of the relational data set in step 3250. The dictionary table may also be treated as a data set.

[0116] Step 3260 comprises converting the relational data set to column oriented form. Each row of the relational data set is assigned a numeric identifier, which may be a unique numeric identifier (UNI). The UNI for each row of the data set may be stored in a new column of data. An example relational data set and dictionary table are illustrated below.

UNI	Date	Name	Value
100	1521072000	1	532
101	1521072000	2	658
102	1521072000	3	5
103	1521158400	2	7654

Name	Corresponding String
1	ABC
2	DEF
3	GHI

[0117] The relational data set may subsequently be separated into a set of columns, and for each column a set of distinct column values is compiled. For each distinct column value, UNIs are identified that have this column value as an attribute. A data structure is subsequently formed that represents the relationship of the distinct column values to an array of UNIs.

[0118] A data structure may be produced for each column in the relational data set. For example, the 'Date' column in the above relational data set yields the following data structure.

Column Value	UNI Array
1521072000	[100, 101, 102]

(continued)

Column Value	UNI Array
1521158400	[103]

[0119] Each row of the data structure can be expressed as a rank 1 tensor, i.e. a vector:

[1521072000 100 101 102]

[0120] Thus, embodiments of the method illustrated in Figure 5 provide a tensor which may be provided to a method according to an embodiment of the invention such as illustrated in Figure 3 for obfuscation.

[0121] It will be appreciated that embodiments of the present invention can be realised in the form of hardware, software or a combination of hardware and software. Any such software may be stored in the form of volatile or non-volatile storage such as, for example, a storage device like a ROM, whether erasable or rewritable or not, or in the form of memory such as, for example, RAM, memory chips, device or integrated circuits or on an optically or magnetically readable medium such as, for example, a CD, DVD, magnetic disk or magnetic tape. It will be appreciated that the storage devices and storage media are embodiments of machine-readable storage that are suitable for storing a program or programs that, when executed, implement embodiments of the present invention. Accordingly, embodiments provide a program comprising code for implementing a system or method as claimed in any preceding claim and a machine readable storage storing such a program. Still further, embodiments of the present invention may be conveyed electronically via any medium such as a communication signal carried over a wired or wireless connection and embodiments suitably encompass the same.

[0122] All of the features disclosed in this specification (including any accompanying claims, abstract and drawings), and/or all of the steps of any method or process so disclosed, may be combined in any combination, except combinations where at least some of such features and/or steps are mutually exclusive.

[0123] Each feature disclosed in this specification (including any accompanying claims, abstract and drawings), may be replaced by alternative features serving the same, equivalent or similar purpose, unless expressly stated otherwise. Thus, unless expressly stated otherwise, each feature disclosed is one example only of a generic series of equivalent or similar features.

[0124] The invention is not restricted to the details of any foregoing embodiments. The invention extends to any novel one, or any novel combination, of the features disclosed in this specification (including any accompanying claims, abstract and drawings), or to any novel one, or any novel combination, of the steps of any method or process so disclosed. The claims should not be construed to cover merely the foregoing embodiments, but also any embodiments which fall within the scope of the claims.

Claims

1. A computer-implemented method of enhancing data privacy, comprising:

Determining (320), at a processor (200), one or more tensors of numeric data in dependence on input data; determining (600), at a processor (200), a transform in dependence on user-associated data; transforming (330), at a processor (200), each of the one or more tensors of numeric data into at least two fragments of data by applying the transform, wherein each of the fragments of data obfuscates the numeric data; and

storing (340), separately, each of the at least two fragments of data at a respective geographically separated storage system (140, 150), wherein the transforming comprises:

dividing each tensor of numeric data into one or more parts; transforming each part of the tensor of numeric data into a transformed part in dependence on the transform; and determining the at least two fragments of data in dependence on each of the transformed parts; wherein the transform is an invertible tensor transformation.

2. The method of claim 1, comprising:

retrieving (410) the at least two fragments of data; determining a reverse transform in dependence on the user-associated data; and

performing (430) the reverse transform on the at least two fragments of data, wherein the reverse transform provides the respective tensor corresponding to the at least two fragments of data.

5 3. The method of any preceding claim, wherein the user-associated data comprises data indicative of an input received from a user via a user interface.

10 4. The method of any of claims 1 or 2, wherein the user-associated data comprises biometric data associated with a user, wherein the biometric data is indicative of one or more of: a fingerprint of the user, a heart rhythm of the user, an iris scan of the user, a facial structure of a user and DNA derived from the user.

5 5. The method of any preceding claim, wherein the determining one or more tensors of numeric data comprises encrypting the input data or data based on the input data.

15 6. The method of any preceding claim, comprising balancing at least some of the one or more tensors of numeric data prior to performing the transforming.

7. The method of any preceding claim, wherein the invertible tensor transformation is representable as an $N \times N$ tensor, wherein N is at least 2.

20 8. The method of any preceding claim, wherein the transforming each part of the tensor of numeric data comprises multiplying each part of the tensor of numeric data by the transform.

25 9. The method of any preceding claim, wherein each transformed part comprises at least a first value and a second value, and wherein:

a first fragment of the at least two fragments of data is determined in dependence on the first value of each of the transformed parts; and

a second fragment of the at least two fragments of data is determined in dependence on the second value of each of the transformed parts.

30 10. The method of claim 2, wherein the reverse transform is performed on reverse transform data.

35 11. The method of any preceding claim, where the storing, separately, each of the at least two fragments at a respective geographically separated storage system (140, 150) comprises:

storing, at a first computer system (140) in a first geographic location, a first fragment of data corresponding to a first tensor of numeric data; and

storing, at a second computer system (150) in a second geographic location, a second fragment of data corresponding to the first tensor of numeric data.

40 12. The method of any preceding claim, wherein the input data is in a table format and the determining one or more tensors of numeric data comprises factoring out (3240) one or more tables of data and a dictionary, wherein each of the one or more tensors of numeric data corresponds to a portion of one or more of the one or more tables of data and the dictionary.

45 13. Computer software which, when executed by a computer, is arranged to perform a method according to any preceding claim, wherein the computer software is stored on a computer-readable medium.

50 14. A computing system (100), comprising:

one or more processors (200) for operatively executing computer readable instructions;
computer-readable data storage medium accessible to the one or more processors (200) storing computer-readable instructions which, when executed by the one or more processors, perform a method comprising steps of:

determining one or more tensors of numeric data in dependence on input data;

determining a transform in dependence on user-associated data;

transforming each of the one or more tensors of numeric data into at least two fragments of data by applying

the transform, wherein each of the fragments of data obfuscates the numeric data; and storing, separately, each of the at least two fragments of data at a respective geographically separated storage system (140, 150), wherein the transforming comprises:

dividing each tensor of numeric data into one or more parts;
transforming each part of the tensor of numeric data into a transformed part in dependence on the transform; and
determining the at least two fragments of data in dependence on each of the transformed parts;
wherein the transform is an invertible tensor transformation.

15. The computing apparatus (100) of claim 14, comprising a communication module for receiving the input data from a client computing apparatus (110).

Patentansprüche

1. Computerimplementiertes Verfahren zum Verbessern des Datenschutzes, umfassend:

Ermitteln (320), in einem Prozessor (200), eines oder mehrerer Tensoren numerischer Daten in Abhängigkeit von Eingabedaten;
Ermitteln (600), in einem Prozessor (200), einer Transformation in Abhängigkeit von benutzerverbundenen Daten;
Transformieren (330), in einem Prozessor (200), jedes des einen oder der mehreren Tensoren numerischer Daten in mindestens zwei Datenfragmente durch Anwenden der Transformation, wobei jedes der Datenfragmente die numerischen Daten verschleiert; und
separates Speichern (340) jedes der mindestens zwei Datenfragmente in einem jeweiligen geografisch getrennten Speichersystem (140, 150),
wobei das Transformieren umfasst:

Aufteilen jedes Tensors numerischer Daten in einen oder mehrere Teile;
Transformieren jedes Teils des Tensors numerischer Daten in einen transformierten Teil in Abhängigkeit von der Transformation; und
Ermitteln der mindestens zwei Datenfragmente in Abhängigkeit von jedem der transformierten Teile;
wobei die Transformation eine invertierbare Tensortransformation ist.

2. Verfahren nach Anspruch 1, umfassend:

Abrufen (410) der mindestens zwei Datenfragmente;
Ermitteln einer Rücktransformation in Abhängigkeit von den benutzerverbundenen Daten; und
Durchführen (430) der Rücktransformation an den mindestens zwei Datenfragmenten, wobei die Rücktransformation den jeweiligen Tensor bereitstellt, der den mindestens zwei Datenfragmenten entspricht.

3. Verfahren nach einem der vorhergehenden Ansprüche, wobei die benutzerverbundenen Daten Daten umfassen, die für eine Eingabe bezeichnend sind, die von einem Benutzer über eine Benutzerschnittstelle empfangen wurde.

4. Verfahren nach einem der Ansprüche 1 oder 2, wobei die benutzerverbundenen Daten biometrische Daten umfassen, die einem Benutzer zugeordnet sind, wobei die biometrischen Daten bezeichnend sind für eines oder mehrere von: einem Fingerabdruck des Benutzers, einem Herzrhythmus des Benutzers, einem Iris-Scan des Benutzers, einer Gesichtsstruktur eines Benutzers und vom Benutzer abgeleiteten DNA.

5. Verfahren nach einem der vorhergehenden Ansprüche, wobei das Bestimmen eines oder mehrerer Tensoren numerischer Daten das Verschlüsseln der Eingabedaten oder der auf den Eingabedaten basierenden Daten umfasst.

6. Verfahren nach einem der vorangehenden Ansprüche, umfassend den Ausgleich mindestens einiger des einen oder der mehreren Tensoren numerischer Daten vor dem Durchführen des Transformierens.

7. Verfahren nach einem der vorhergehenden Ansprüche, wobei die invertierbare Tensortransformation als NxN-

Tensor darstellbar ist, wobei N mindestens 2 ist.

8. Verfahren nach einem der vorhergehenden Ansprüche, wobei das Transformieren jedes Teils des Tensors numerischer Daten das Multiplizieren jedes Teils des Tensors numerischer Daten mit der Transformation umfasst.

9. Verfahren nach einem der vorhergehenden Ansprüche, wobei jeder transformierte Teil mindestens einen ersten Wert und einen zweiten Wert umfasst und wobei:

ein erstes Fragment der mindestens zwei Datenfragmente in Abhängigkeit vom ersten Wert jedes der transformierten Teile ermittelt wird; und
ein zweites Fragment der mindestens zwei Datenfragmente in Abhängigkeit vom zweiten Wert jedes der transformierten Teile ermittelt wird.

10. Verfahren nach Anspruch 2, wobei die Rücktransformation an Rücktransformationsdaten durchgeführt wird.

11. Verfahren nach einem der vorhergehenden Ansprüche, wobei das separate Speichern jedes der mindestens zwei Fragmente in einem jeweiligen geografisch getrennten Speichersystem (140, 150) umfasst:

Speichern, in einem ersten Computersystem (140) an einem ersten geografischen Standort, eines ersten Datenfragments, das einem ersten Tensor numerischer Daten entspricht; und
Speichern, in einem zweiten Computersystem (150) an einem zweiten geografischen Standort, eines zweiten Datenfragments, das dem ersten Tensor numerischer Daten entspricht.

12. Verfahren nach einem der vorhergehenden Ansprüche, wobei die Eingabedaten in einem Tabellenformat vorliegen und das Ermitteln eines oder mehrerer Tensoren numerischer Daten das Herausfaktorisieren (3240) einer oder mehrerer Datentabellen und eines Wörterbuchs umfasst, wobei jeder vom dem einen oder den mehreren Tensoren numerischer Daten einem Teil von einer oder mehreren der einen oder mehreren Datentabellen und des Wörterbuchs entspricht.

13. Computersoftware, die bei Ausführung durch einen Computer dazu ausgelegt ist, ein Verfahren nach einem der vorhergehenden Ansprüche durchzuführen, wobei die Computersoftware auf einem computerlesbaren Medium gespeichert ist.

14. Computersystem (100), umfassend:

einen oder mehrere Prozessoren (200) zum betriebsmäßigen Ausführen computerlesbarer Anweisungen;
ein computerlesbares Datenspeichermedium, auf das der eine oder die mehreren Prozessoren (200) zugreifen können und das computerlesbare Anweisungen speichert, die, wenn sie von dem einen oder den mehreren Prozessoren ausgeführt werden, ein Verfahren durchführen, das die Schritte umfasst:

Ermitteln eines oder mehrerer Tensoren numerischer Daten in Abhängigkeit von Eingabedaten;
Ermitteln einer Transformation in Abhängigkeit von benutzerverbundenen Daten;
Transformieren jedes des einen oder der mehreren Tensoren numerischer Daten in mindestens zwei Datenfragmente durch Anwenden der Transformation, wobei jedes der Datenfragmente die numerischen Daten verschleiert; und
separates Speichern jedes der mindestens zwei Datenfragmente in einem jeweiligen geografisch getrennten Speichersystem (140, 150),
wobei das Transformieren umfasst:

Aufteilen jedes Tensors numerischer Daten in einen oder mehrere Teile;
Transformieren jedes Teils des Tensors numerischer Daten in einen transformierten Teil in Abhängigkeit von der Transformation; und
Ermitteln der mindestens zwei Datenfragmente in Abhängigkeit von jedem der transformierten Teile;
wobei die Transformation eine invertierbare Tensortransformation ist.

15. Computergerät (100) nach Anspruch 14, umfassend ein Kommunikationsmodul zum Empfangen der Eingabedaten von einem Client-Computergerät (110).

Revendications

1. Procédé mis en oeuvre par ordinateur pour améliorer la confidentialité des données, comprenant :

la détermination (320), au niveau d'un processeur (200), d'un ou plusieurs tenseurs de données numériques en fonction de données d'entrée ;
la détermination (600), au niveau d'un processeur (200), d'une transformation en fonction de données associées à l'utilisateur ;
la transformation (330), au niveau d'un processeur (200), de chacun de l'un ou plusieurs tenseurs de données numériques en au moins deux fragments de données en appliquant la transformation, dans lequel chacun des fragments de données brouille les données numériques ; et
le stockage (340), séparément, de chacun des au moins deux fragments de données au niveau d'un système de stockage respectif géographiquement séparé (140, 150), dans lequel la transformation comprend :

la division de chaque tenseur de données numériques en une ou plusieurs parties ;
la transformation de chaque partie du tenseur de données numériques en une partie transformée en fonction de la transformation ; et
la détermination des au moins deux fragments de données en fonction de chacune des parties transformées ;
dans lequel la transformation est une transformation tensorielle inversible.

2. Procédé selon la revendication 1, comprenant :

la récupération (410) des au moins deux fragments de données ;
la détermination d'une transformation inverse en fonction des données associées à l'utilisateur ; et
l'exécution (430) de la transformation inverse sur les au moins deux fragments de données, dans lequel la transformation inverse fournit le tenseur respectif correspondant aux au moins deux fragments de données.

3. Procédé selon l'une quelconque des revendications précédentes, dans lequel les données associées à l'utilisateur comprennent des données indicatives d'une entrée reçue d'un utilisateur via une interface utilisateur.

4. Procédé selon l'une quelconque des revendications 1 ou 2, dans lequel les données associées à l'utilisateur comprennent des données biométriques associées à un utilisateur, dans lequel les données biométriques sont indicatives d'un ou plusieurs parmi : une empreinte digitale de l'utilisateur, un rythme cardiaque de l'utilisateur, un scan de l'iris de l'utilisateur, une structure faciale d'un utilisateur et un ADN dérivé de l'utilisateur.

5. Procédé selon l'une quelconque des revendications précédentes, dans lequel la détermination d'un ou plusieurs tenseurs de données numériques comprend le chiffrement des données d'entrée ou de données basées sur les données d'entrée.

6. Procédé selon l'une quelconque des revendications précédentes, comprenant l'équilibrage d'au moins certains de l'un ou plusieurs tenseurs de données numériques avant d'effectuer la transformation.

7. Procédé selon l'une quelconque des revendications précédentes, dans lequel la transformation tensorielle inversible est représentable sous la forme d'un tenseur $N \times N$, dans lequel N vaut au moins 2.

8. Procédé selon l'une quelconque des revendications précédentes, dans lequel la transformation de chaque partie du tenseur de données numériques comprend la multiplication de chaque partie du tenseur de données numériques par la transformation.

9. Procédé selon l'une quelconque des revendications précédentes, dans lequel chaque partie transformée comprend au moins une première valeur et une seconde valeur, et dans lequel :

un premier fragment desdits au moins deux fragments de données est déterminé en fonction de la première valeur de chacune des parties transformées ; et
un deuxième fragment desdits au moins deux fragments de données est déterminé en fonction de la deuxième valeur de chacune des parties transformées.

10. Procédé selon la revendication 2, dans lequel la transformation inverse est effectuée sur des données de transformation inverse.

11. Procédé selon l'une quelconque des revendications précédentes, dans lequel le stockage, séparément, de chacun des au moins deux fragments au niveau d'un système de stockage respectif géographiquement séparé (140, 150), comprend :

le stockage, au niveau d'un premier système informatique (140) dans un premier emplacement géographique, d'un premier fragment de données correspondant à un premier tenseur de données numériques ; et
le stockage, au niveau d'un deuxième système informatique (150) dans un deuxième emplacement géographique, d'un deuxième fragment de données correspondant au premier tenseur de données numériques.

12. Procédé selon l'une quelconque des revendications précédentes, dans lequel les données d'entrée sont dans un format de table et la détermination d'un ou plusieurs tenseurs de données numériques comprend la factorisation (3240) d'une ou plusieurs tables de données et d'un dictionnaire, dans lequel chacun de l'un ou plusieurs tenseurs de données numériques correspond à une partie d'un ou plusieurs de l'une ou plusieurs tables de données et du dictionnaire.

13. Logiciel informatique qui, lorsqu'il est exécuté par un ordinateur, est conçu pour exécuter un procédé selon l'une quelconque des revendications précédentes, dans lequel le logiciel informatique est stocké sur un support lisible par ordinateur.

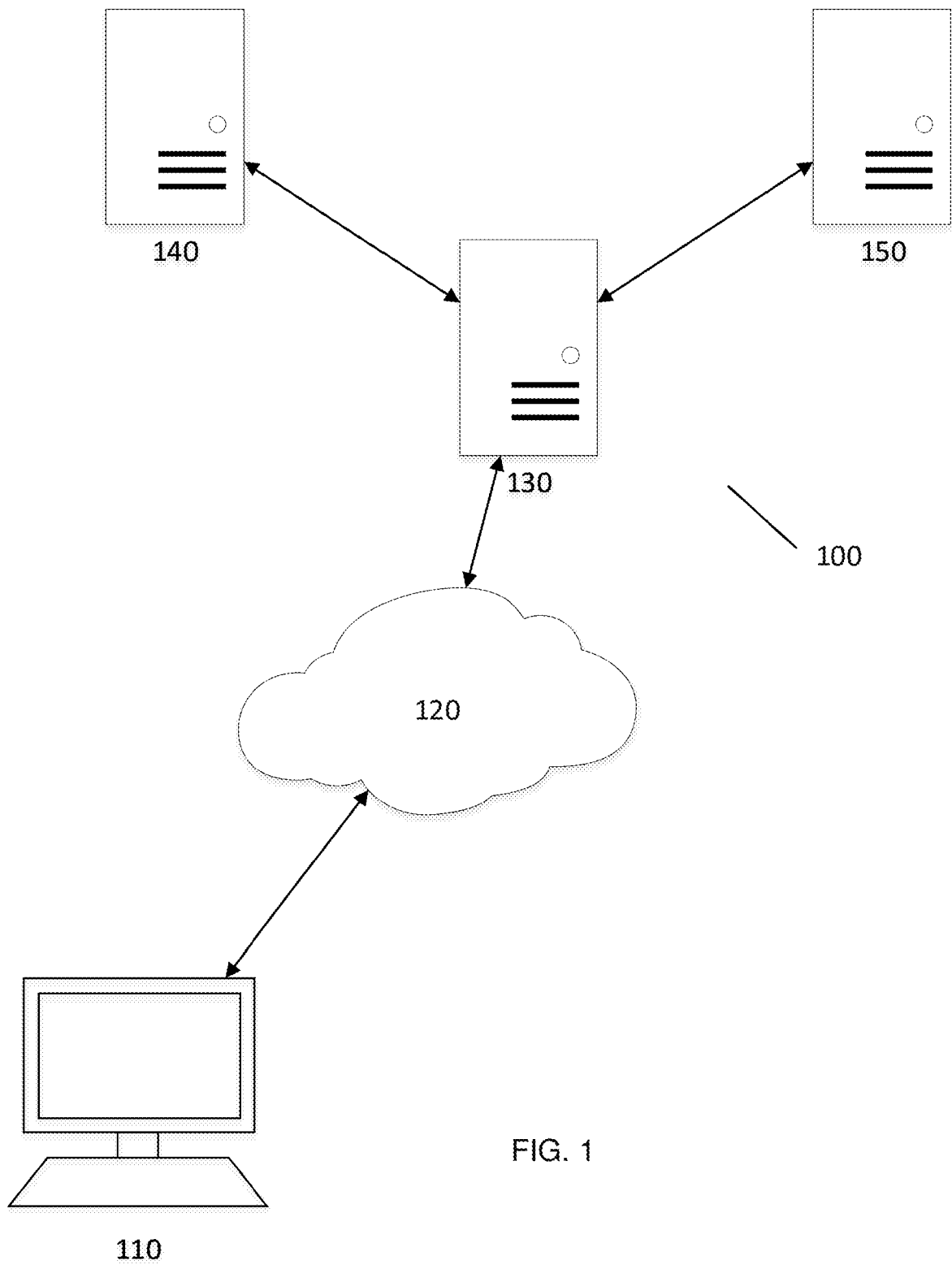
14. Système informatique (100) comprenant :

un ou plusieurs processeurs (200) pour exécuter de manière opérationnelle des instructions lisibles par ordinateur ;
un support de stockage de données lisible par ordinateur accessible à l'un ou plusieurs processeurs (200) stockant des instructions lisibles par ordinateur qui, lorsqu'elles sont exécutées par l'un ou plusieurs processeurs, exécutent un procédé comprenant les étapes de :

détermination d'un ou plusieurs tenseurs de données numériques en fonction de données d'entrée ;
détermination d'une transformation en fonction de données associées à l'utilisateur ;
transformation de chacun de l'un ou plusieurs tenseurs de données numériques en au moins deux fragments de données en appliquant la transformation, dans lequel chacun des fragments de données brouille les données numériques ; et
stockage, séparément, de chacun des au moins deux fragments de données au niveau d'un système de stockage respectif géographiquement séparé (140, 150), dans lequel la transformation comprend :

la division de chaque tenseur de données numériques en une ou plusieurs parties ;
la transformation de chaque partie du tenseur de données numériques en une partie transformée en fonction de la transformation ; et
la détermination des au moins deux fragments de données en fonction de chacune des parties transformées ;
dans lequel la transformation est une transformation tensorielle inversible.

15. Appareil informatique (100) selon la revendication 14, comprenant un module de communication pour recevoir les données d'entrée provenant d'un appareil informatique client (110).



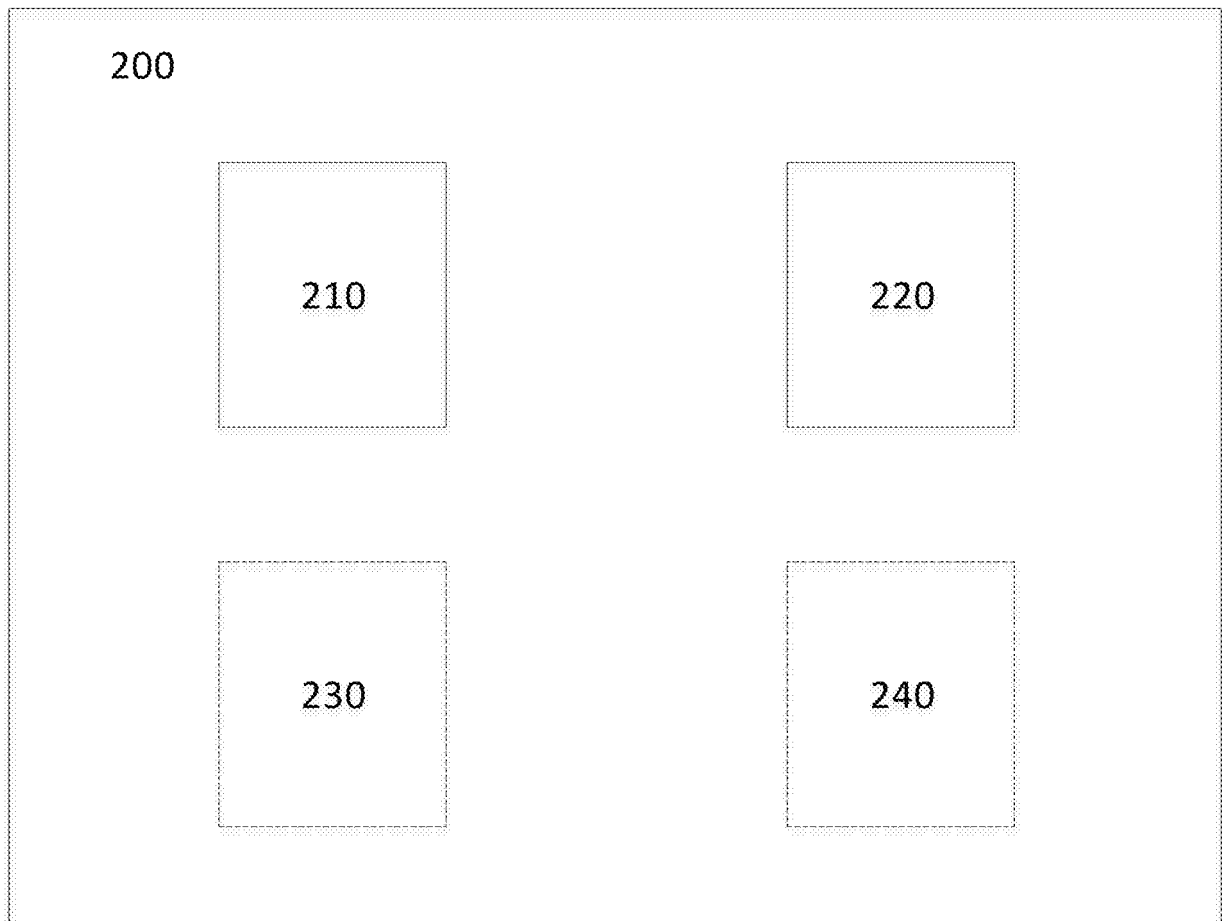


FIG. 2

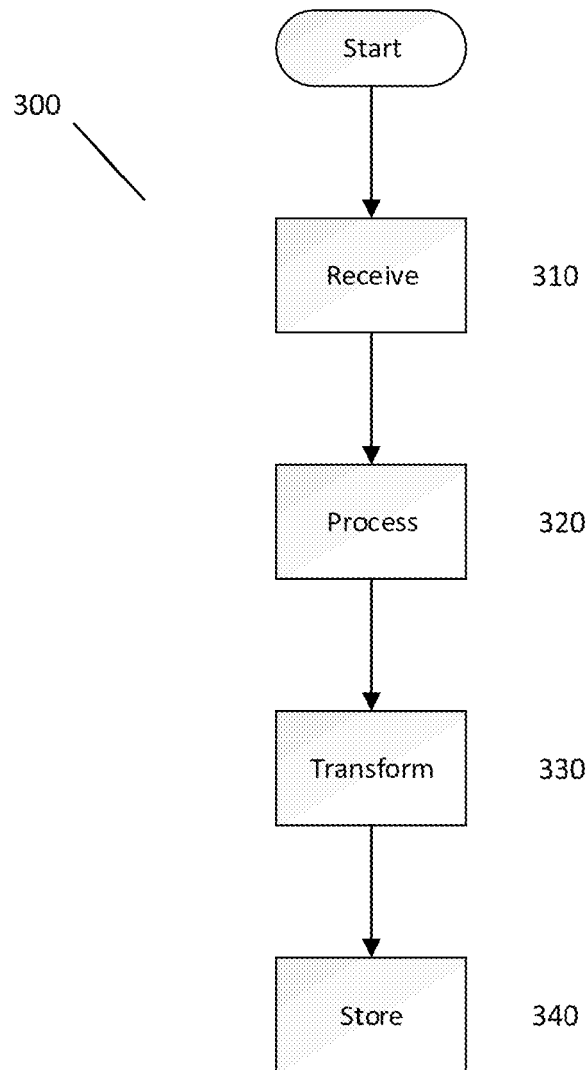


FIG. 3

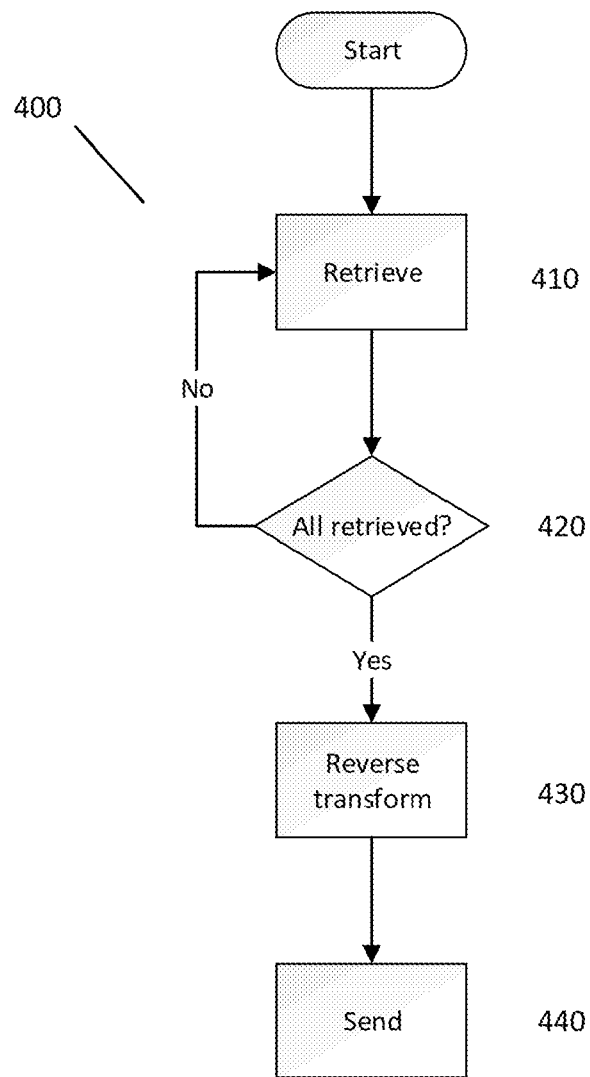


FIG. 4

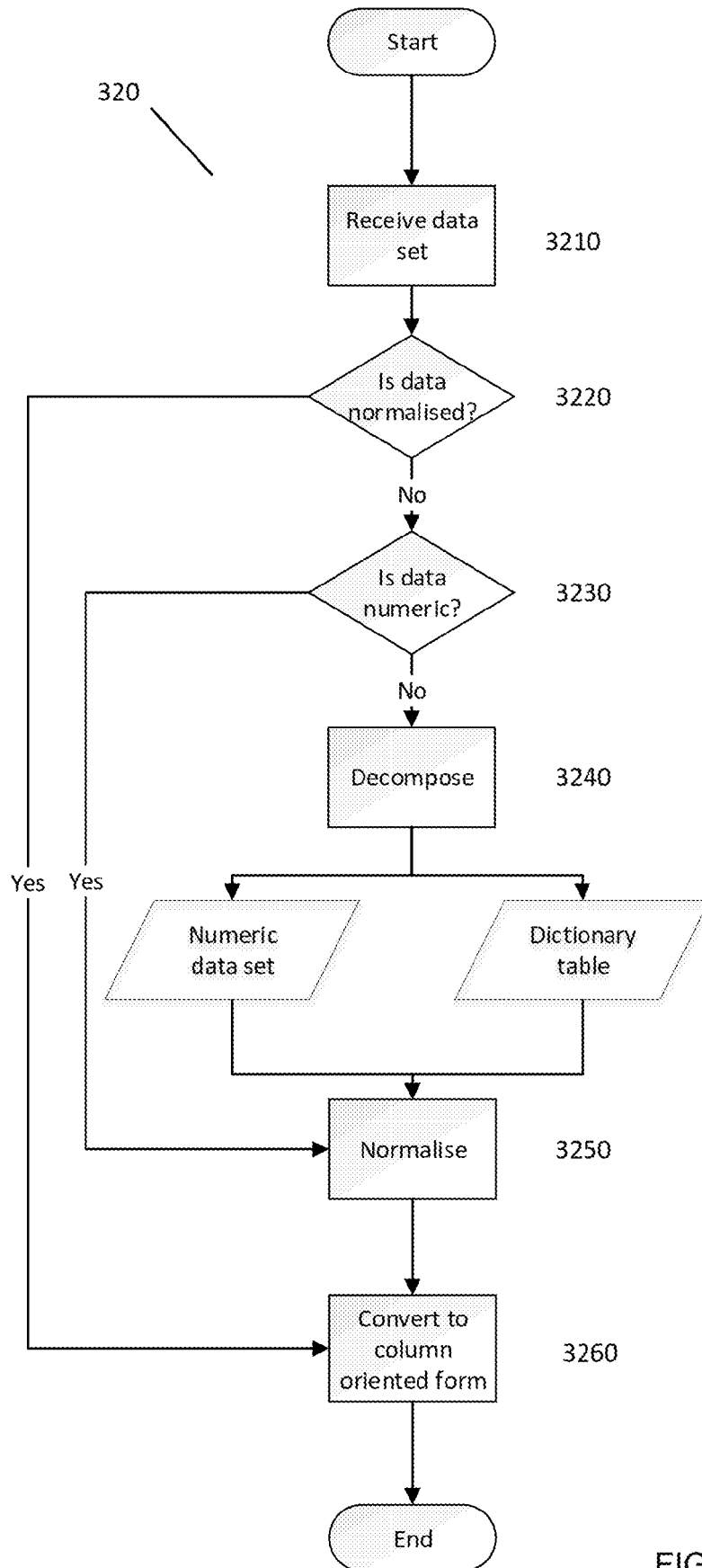


FIG. 5

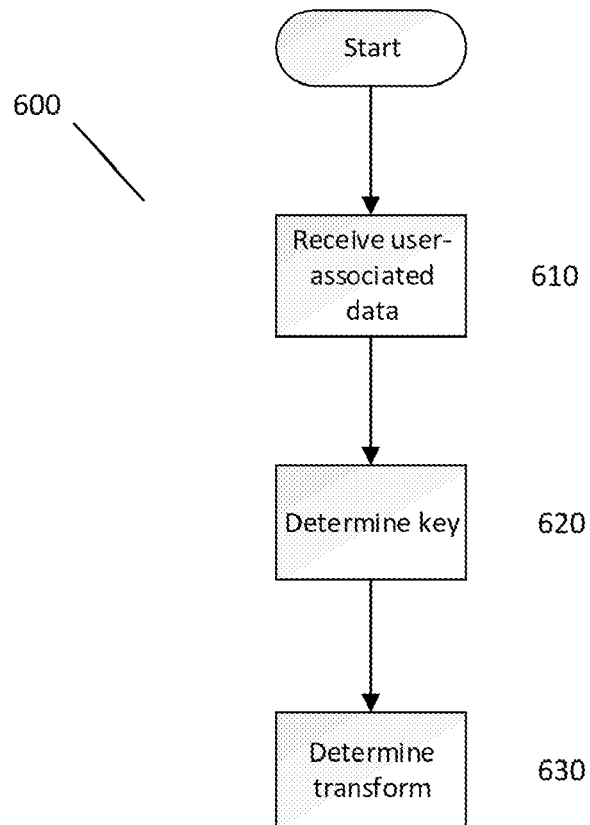


FIG. 6

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- WO 2015057854 A [0002]
- EP 3015988 A [0003]