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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 April 2022 (14.04.2022)

WIPO | PCT

(10) International Publication Number
WO 2022/073624 A1

(51) International Patent Classification:
G06F 21/62 (2013.01)

(21) International Application Number:
PCT/EP2020/078524

(22) International Filing Date:
11 October 2020 (11.10.2020)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HUAWEI TECHNOLOGIES CO., LTD.** [CN/CN]; Huawei Administration Building Bantian Longgang District, Shenzhen, Guangdong 518129 (CN).

(72) Inventor; and

(71) Applicant *(for US only)*: **NATANZON, Assaf** [IL/DE]; c/o Huawei Technologies Duesseldorf GmbH, Riesstr. 25, 80992 Munich (DE).

(74) Agent: **KREUZ, Georg**; Huawei Technologies Duesseldorf GmbH, Riesstr. 25, 80992 Munich (DE).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, 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, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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, ST, 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, KM, ML, MR, NE, SN, TD, TG).

(54) **Title:** DATA STORAGE ARRANGEMENT AND METHOD EMPLOYING LOCAL MEMORY AND DATA STORAGE

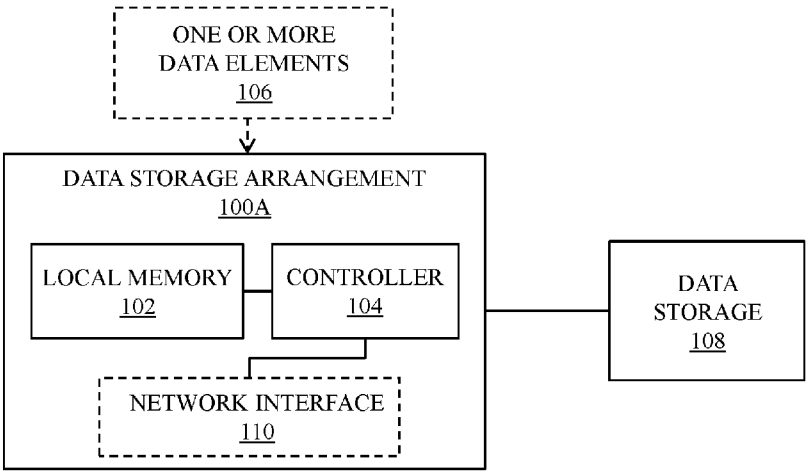


FIG. 1A

(57) **Abstract:** A data storage arrangement includes a local memory and a controller. The local memory is configured to store one or more data elements. The data storage arrangement is connected to a data storage and controller is configured to store at least one of the one or more data elements. The controller is further configured to receive data element to be stored, generate a copy of data portions to be anonymized, encrypt the copy of data portions to be anonymized utilizing an encryption key. The controller is further configured to mask data portions to be anonymized by replacing data portions with anonymous content thereby generating modified data element. The controller is further configured to store the modified data element and the encrypted copy of data portions to be anonymized in the data storage. The controller is further configured to store the encryption key in the local memory.

WO 2022/073624 A1

[Continued on next page]

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

DATA STORAGE ARRANGEMENT AND METHOD EMPLOYING LOCAL MEMORY AND DATA STORAGE

TECHNICAL FIELD

The present disclosure relates generally to the field of data storage systems; and more specifically, to data storage arrangements and methods for data storage arrangements that employ local memory and data storage for securely and efficiently storing sensitive data.

BACKGROUND

Nowadays, a vast amount of data is being produced and stored globally, on a regular basis. For example, social networks, internet of things, scientific experiments, commercial services, industrial services, banking services, businesses, and the like, play a significant role in generating this data. Presently, data storage systems (for example, secondary storage systems, remote data storage systems, and the like) are being employed to store this data. Proper management and storage of data in the data storage systems is therefore important, in order to improve and maximize efficiency of the data storage systems.

However, existing techniques for managing and storing the data in the data storage systems are associated with several limitations. Firstly, the existing techniques are inefficient in terms of identifying sensitivity of the data. As a result, sensitive data stored at the data storage systems stays unprotected and is accessible even to unauthorized persons. For example, a developer of a software application may require real world data, stored at the data storage systems, for developing and optimizing the software application. However, when sensitive data in the real-world data is not identified and suitable protected, the developer is able to access the sensitive data. This is undesirable from a perspective of privacy protection and data security. Secondly, the existing techniques do not account for sensitivity of the data when storing the data. As a result, the stored data is bulky and typically non-compliant with regulations for privacy protection and data security.

Therefore, in light of the foregoing discussion, there exists a need to overcome the aforementioned drawbacks associated with the existing techniques for managing and storing data in data storage systems.

SUMMARY

The present disclosure seeks to provide a data storage arrangement and a method for the data storage arrangement that employ a local memory and a data storage for securely and efficiently storing sensitive data. The present disclosure seeks to provide a solution to the existing problems of high storage requirements for storing bulky data, privacy leakage of sensitive data, lack of data security, inefficient compliance adherence that are associated with the existing techniques for managing and storing the data in data storage systems. An aim of the present disclosure is to provide a solution that overcomes at least partially the problems encountered in prior art and provide a data storage arrangement and a method that provide improved compliance and efficiency in deduplication when storing data in data storage to enable data privacy protection, data security, data compliance, and reduction of redundant data at a time of storing data.

The object of the present disclosure is achieved by the solutions provided in the enclosed independent claims. Advantageous implementations of the present disclosure are further defined in the dependent claims.

In an aspect, the present disclosure provides a data storage arrangement comprising a local memory and a controller. The local memory is configured to store one or more data elements. The data storage arrangement is connected to a data storage and the controller is configured to store at least one of the one or more data elements. In the data storage arrangement, the controller is further configured to: receive a data element to be stored; generate a copy of one or more data portions to be anonymized; encrypt the copy of the one or more data portions to be anonymized utilizing an encryption key; mask the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with an anonymous content thereby generating a modified data element to be stored; store the modified data element to be stored in the data storage; store the encrypted copy of the one or more data portions to be anonymized in the data storage; and store the encryption key in the local memory.

The data storage arrangement enables efficient and secure data storage by storing the modified data element having the anonymized content and the encrypted copy in the data storage, whilst storing the encryption key in the local memory. In this way, the data storage arrangement efficiently employs the local memory and the data storage to separately store data without putting security and privacy of entities associated with such data at risk. As the encryption key is not accessible in the data storage, and only the modified data element (which is selectively anonymized to hide sensitive data) and the encrypted copy (which is secured for confidentiality protection by encryption) are stored in the data storage, even when an unauthorized entity gains access to the data storage, the unauthorized entity is unable to gain access to the original sensitive data. In this way, the data storage arrangement provides data security and privacy protection, whilst also facilitating compliance adherence. The data storage arrangement efficiently utilizes storage spaces of both the local memory and the data storage to avoid any wastage of space. The data storage arrangement can be implemented and used easily.

In an implementation form, the anonymous content is an identifier of the encrypted copy of the corresponding data portion.

By virtue of the anonymous content being the identifier of the encrypted copy of the corresponding data portion, the anonymous content provides an efficient and secure anonymization effect in the modified data element as the identifier facilitates in establishing an association between the anonymous content and the encrypted copy, whilst providing privacy protection for sensitive data portions.

In an implementation form, the anonymous content is the encrypted copy of the corresponding data portion.

By virtue of the anonymous content being the encrypted copy of the corresponding data portion, the anonymous content provides a secure selective anonymization effect in the modified data element as the encrypted copy already protects the confidentiality of the corresponding data portion in the encrypted form.

In an implementation form, the anonymous content is a masking content.

By virtue of the anonymous content being the masking content, the one or more data portions to be anonymized are effectively hidden (namely, obfuscated) by the masking content to protect sensitive data in the one or more data portions to be anonymized.

In an implementation form, the controller is further configured to receive an indication on what data is to be anonymized, wherein the indication relates to one or more of local legislative requirements, personal information and/or faces.

The indication enables the controller to accurately identify the data that is to be anonymized, in order to effectively enable selective anonymization for only requisite data (that includes sensitive data) in the data element.

In an implementation form, the controller is further configured to parse for the data to be anonymized.

By virtue of parsing, there is facilitated accurate identification of the one or more data portions in the data element that comprise the data to be anonymized.

In an implementation form, the indication on what data is to be anonymized further relates to one or more locations of the data in the data element.

The indication relating to one or more locations of the data to be anonymized enables the controller to accurately identify the the data in the data element, for reliably implementing anonymization for this data.

In an implementation form, the controller is further configured to mask the one or more data portions to be anonymized and generate the modified data element to be stored by utilizing a write-with-mask command indicating location of data to be anonymized and the anonymous content to replace the data to be anonymized.

The write-with-mask command enables in accurately generating the modified data element by explicitly indicating both the location of the data to be anonymized and the anonymous content that is to replace such data.

In an implementation form, the write-with-mask command comprises an indicator to whether the data to be anonymized is present, and the controller is configured to determine if the data to be anonymized is present prior to masking the one or more data portions to be anonymized.

By virtue of the indicator, the process of anonymization performed by the controller is improved, as processing resources for anonymization will be spent by the controller only when the data to be anonymized is present prior to masking.

In an implementation form, the controller is further configured to generate a copy of the modified data element to be stored utilizing differential compression; and store the copy of the modified data element to be stored as the modified data element to be stored in the data storage.

The differential compression enables efficient utilization of storage space of the data storage by efficiently compressing and storing, at the data storage, the copy of the modified data element as the modified data element. Differential compression allows storage of only non-identical (or different) parts of data portions at the data storage and use of pointer for referring to similar pre-stored data portions at the data storage, in order to efficiently utilize the storage space of the data storage.

In an implementation form, the controller is further configured to generate a copy of the modified data element to be stored utilizing deduplication; and store the copy of the modified data element to be stored as the modified data element to be stored in the data storage.

The deduplication enables in efficiently reducing required storage capacity at the data storage for storing the modified data element, since only unique data is stored at the data storage and duplicate data is not stored at the data storage.

In another aspect, the present disclosure provides a method for a data storage arrangement. The data storage arrangement comprises a local memory being configured to store one or more data elements. The data storage arrangement is connected to a data storage for storing at least one of the one or more data elements. The method comprises: receiving a data element to be stored; generating a copy of one or more data portions to be anonymized; encrypting the copy of the one or more data portions to be anonymized utilizing an encryption key; masking the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with anonymous content thereby generating a modified data element to be stored; storing the modified data element to be stored in the data storage; storing the encrypted copy of the one or more data portions to be anonymized in the data storage; and storing the encryption key in the local memory.

The method enables efficient and secure data storage by storing the modified data element having the anonymized content and the encrypted copy in the data storage, whilst storing the encryption key in the local memory. Herein, generating the modified data element efficiently employs selective anonymization to anonymize only sensitive data in the data element. Further, encrypting the one or more data portions of the data element comprising the sensitive data

provides security and protects privacy of a given entity to which the sensitive data pertains. As the modified data element and the encrypted copy are stored at the data storage which is separate from the local memory at which the encryption key is stored, there is provided secure and reliable protection against unauthorized access to sensitive data as only authorized entities will have access to both the data storage and the local memory.

In another aspect, the present disclosure provides a computer-readable medium carrying computer instructions that when loaded into and executed by a controller of a data storage arrangement enables the data storage arrangement to implement the method.

The computer-readable medium carrying computer instructions of this aspect achieves all the advantages and effects of the data storage arrangement, or the method.

In another aspect, the present disclosure provides a data storage arrangement comprising a local memory being configured to store one or more data elements. The data storage arrangement further comprises a storing software module for storing at least one of the one or more data elements. The data storage arrangement further comprises: a receiving software module for receiving the data element to be stored; a copying software module for generating a copy of one or more data portions to be anonymized; an encrypting software module for encrypting the copy of the one or more data portions to be anonymized utilizing an encryption key; a masking software module for masking the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with anonymous content thereby generating a modified data element to be stored; and a software module for storing the modified data element to be stored in a data storage and for storing the encrypted copy of the one or more data portions to be anonymized in the data storage and for storing the encryption key in the local memory.

The data storage arrangement of this aspect employs dedicated software modules to perform dedicated processing tasks concerning securely storing the data element using anonymization and encryption. These software modules perform their processing tasks time-efficiently and reliably. The data storage arrangement of this aspect achieves all the advantages and effects of the data storage arrangement of an earlier aspect, or the method.

It has to be noted that all devices, elements, circuitry, units and means described in the present application could be implemented in the software or hardware elements or any kind of combination thereof. All steps which are performed by the various entities described in the present application as well as the functionalities described to be performed by the various

entities are intended to mean that the respective entity is adapted to or configured to perform the respective steps and functionalities. Even if, in the following description of specific embodiments, a specific functionality or step to be performed by external entities is not reflected in the description of a specific detailed element of that entity which performs that specific step or functionality, it should be clear for a skilled person that these methods and functionalities can be implemented in respective software or hardware elements, or any kind of combination thereof. It will be appreciated that features of the present disclosure are susceptible to being combined in various combinations without departing from the scope of the present disclosure as defined by the appended claims.

Additional aspects, advantages, features and objects of the present disclosure would be made apparent from the drawings and the detailed description of the illustrative implementations construed in conjunction with the appended claims that follow.

BRIEF DESCRIPTION OF THE DRAWINGS

The summary above, as well as the following detailed description of illustrative embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the present disclosure, exemplary constructions of the disclosure are shown in the drawings. However, the present disclosure is not limited to specific methods and instrumentalities disclosed herein. Moreover, those in the art will understand that the drawings are not to scale. Wherever possible, like elements have been indicated by identical numbers.

Embodiments of the present disclosure will now be described, by way of example only, with reference to the following diagrams wherein:

FIG. 1A illustrates a block diagram of a data storage arrangement, in accordance with an embodiment of the present disclosure;

FIG. 1B illustrates a block diagram of a data storage arrangement, in accordance with another embodiment of the present disclosure;

FIG. 2 illustrates an environment in which a data storage arrangement is used, in accordance with an embodiment of the present disclosure; and

FIGs. 3A and 3B illustrate steps of a method for a data storage arrangement, in accordance with an embodiment of the present disclosure.

In the accompanying drawings, an underlined number is employed to represent an item over which the underlined number is positioned or an item to which the underlined number is adjacent. A non-underlined number relates to an item identified by a line linking the non-underlined number to the item. When a number is non-underlined and accompanied by an associated arrow, the non-underlined number is used to identify a general item at which the arrow is pointing.

DETAILED DESCRIPTION OF EMBODIMENTS

The following detailed description illustrates embodiments of the present disclosure and ways in which they can be implemented. Although some modes of carrying out the present disclosure have been disclosed, those skilled in the art would recognize that other embodiments for carrying out or practicing the present disclosure are also possible.

FIG. 1A illustrates a block diagram of a data storage arrangement, in accordance with an embodiment of the present disclosure. With reference to FIG. 1A, there is shown a data storage arrangement **100A**. The data storage arrangement **100A** comprises a local memory **102** and a controller **104**. The local memory **102** is configured to store one or more data elements **106**. The data storage arrangement **100A** is connected to a data storage **108** and the controller **104** is configured to store at least one of the one or more data elements **106**. The data storage arrangement **100A** further comprises a network interface **110**. Typically, for example, the data storage arrangement **100A** is a local storage array (i.e. an array of local storage devices), and the data storage **108** is a remote data storage such as a cloud-based data storage.

In the data storage arrangement **100A**, the controller **104** is further configured to: receive a data element to be stored; generate a copy of one or more data portions to be anonymized; encrypt the copy of the one or more data portions to be anonymized utilizing an encryption key; mask the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with an anonymous content thereby generating a modified data element to be stored; store the modified data element to be stored in the data storage **108**; store the encrypted copy of the one or more data portions to be anonymized in the data storage **108**; and store the

encryption key in the local memory **102**. As an example, the modified data element and the encrypted copy of the one or more data portions to be anonymized may be stored in the cloud-based data storage.

Throughout the present disclosure, the term "*data storage arrangement*" refers to an arrangement of hardware, software, firmware, or a combination of these, that stores data by utilizing efficient data anonymization and deduplication. The term "*local memory*" refers to a hardware storage of the data storage arrangement **100A** that is used to store data locally (namely, on-site) with respect to a location of the data storage arrangement **100A**. The local memory **102** stores the one or more data elements **106**. The local memory **102** includes suitable logic, circuitry, or interfaces that are configured to store the one or more data elements **106**, pointers and other data based on instructions received from the controller **104**. Examples of the local memory **102** include, but are not limited to, Electrically Erasable Programmable Read-Only Memory (EEPROM), Random Access Memory (RAM), Read Only Memory (ROM), Hard Disk Drive (HDD), Flash memory, Solid-State Drive (SSD), and/or CPU cache memory. The local memory **102** may store an operating system and/or other program products (including one or more operation algorithms) to operate the data storage arrangement **100A**.

In an implementation, the controller **104** is configured to execute instructions stored in the local memory **102**. In an example, the controller **104** may be a general-purpose processor. Examples of the controller **104** may include, but are not limited to, a microprocessor, a microcontroller, a complex instruction set computing (CISC) processor, an application-specific integrated circuit (ASIC) processor, a reduced instruction set (RISC) processor, a very long instruction word (VLIW) processor, a central processing unit (CPU), a state machine, a data processing unit, and other processors or control circuitry. Moreover, the controller **104** may refer to one or more individual processors, processing devices, a processing unit that is part of a machine, such as the data storage arrangement **100A**. The controller **104** is configured to store at least one of the one or more data elements **106** by utilizing anonymization and anonymization-aware encryption. Anonymization enables the controller **104** to provide security and protect privacy of data when storing at the data storage **108**. Anonymization-aware encryption of only the anonymized data enables the controller **104** to provide, in a space-efficient manner, additional security of the anonymized data.

The network interface **110** includes suitable logic, circuitry, and/or interfaces that facilitate communication between with two or more devices, such as controllers, user devices, data

storages or servers. Examples of the network interface **110** include, but are not limited to, a network interface card (NIC), an antenna, a radio frequency (RF) transceiver, or network ports.

Herein, the term "*data storage*" refers to another hardware storage associated with the data storage arrangement **100A**. The controller **104** is coupled in communication with the data storage **108**, via a communication network. Notably, the data storage **108** is different from the local memory **102**. In an embodiment, the network interface **110** facilitates data communication between the controller **104** and the data storage **108**, over the communication network. The communication network may be wired, wireless, or any combination thereof. The communication network is one or more of: a peer-to-peer network, a hybrid peer-to-peer network, local area networks (LANs), radio access networks (RANs), metropolitan area networks (MANS), wide area networks (WANs), all or a portion of a public network such as the global computer network known as the Internet, a Wireless Fidelity (Wi-Fi) network, a wireless personal area network (WPAN), a private network, a cellular network and any other communication network or networks at one or more locations.

In an embodiment, the data storage **108** is a remote data storage. The remote data storage is an off-site storage that is arranged remotely from the location of the data storage arrangement **100A**. As an example, the remote data storage may be a cloud-based data storage, wherein data is stored at a cloud server (namely, a virtual server). In the cloud-based data storage, the data to be stored is transferred via the internet to the cloud server and can be retrieved as per requirement via an interface (for example, a web-based interface).

In another embodiment, the data storage **108** is a local data storage. The local data storage is an on-site storage that is arranged at the location of the data storage arrangement **100A**.

Herein, the term "*data element*" refers to a unit of data or information. The one or more data elements **106** refers to incoming information or data stream that arrive at the data storage arrangement **100A**. For example, the one or more data elements **106** may arrive as an input/output (I/O) request during deduplication process (i.e. when backup is performed from a primary storage system (e.g. a host server) to a secondary storage system, such as the data storage arrangement **100A**. In an embodiment, the one or more data elements **106** potentially include personal data of entities, such as personal identifiable data or data that may be subject to data privacy and security under various regulations or data protection laws, such as Data Protection Act (DPA). Examples of a given data element may include, but are not limited to a

document (such as a text document, a spreadsheet, a table, a form, a certificate, electronic mail, a presentation, and the like), an image, a video, an audio, or other forms of data elements.

In an embodiment, the one or more data elements **106** have a structured form. In such a case, each data element has well-defined and organized data portions in which data is present in a systematic manner. For example, the one or more data elements **106** may be a text document including a string of characters as "Dani, 33, 0541111111, 4580800080001999, A street" representing fields name, age, phone number, credit card number, and address fields in a sequence. In another embodiment, the one or more data elements **106** have an unstructured form. In such a case, data portions within each data element are not well-organized, and data may be present at any data portion.

In operation, the controller **104** is configured to receive the data element to be stored. In an embodiment, the data element to be stored is stored, upon receiving, at the local memory **102**. The data element to be stored may be received by the controller **104** from one or more external devices that is/are communicatively coupled to the data storage arrangement **100A** via a communication network. The data storage arrangement **100A** may be a secondary storage for storing backup of data of user devices or primary storage system(s). In such a case, the one or more external devices is/are the user devices or the primary storage system(s). In an example, the data storage arrangement **100A** is configured to store data of a plurality of user devices or host server in an organization. In an example, the data element may be received at the time of backup to perform deduplication. The data element corresponds to the one or more data elements **106**.

In an example, the data element (to be stored) that is received by the controller **104** may be Table 1 as given below

Name	Age	Phone Number	Credit Card Number	Address
Dani	33	0541111111	4580800080001999	A street
Josef	55	0541222222	4580123434001999	B street

Table 1

wherein name is of 64 characters, age is of 3 characters, phone number is of 10 characters, credit card number is of 16 characters, and address is of 64 characters.

In another example, the data element to be stored that is received by the controller **104** is a text document having an alphanumeric character string: Dani, 33, 0541111111, 4580800080001999, A street; Josef, 55, 0541222222, 4580123434001999, B street.

In an embodiment, the controller **104** is further configured to receive an indication on what data is to be anonymized, wherein the indication relates to one or more of local legislative requirements, personal information and/or faces. The indication facilitates in efficiently anonymizing only requisite data portions having sensitive data in the data elements, instead of other data portions of the data element that do not have sensitive data. The anonymization of data refers to a process of information sanitization, executed by the controller **104**, to enable protection of privacy. It is the process of either encrypting or removing personally identifiable information from one or more data elements **106**, so that people associated with the data remain anonymous. The data that is to be anonymized is typically "*sensitive data*", whereas other data is typically "*non-sensitive data*". Sensitive data is to be anonymized to safeguard security and privacy of a given entity (for example, such as an individual or an organization). Examples of the sensitive data include, but are not limited to, personally identifiable information (such as, social security numbers, credit card numbers, biometric information, medical data, and the like), business information (such as, trade secrets, supplier information, customer data, employee data, and the like), and classified information (such as, government top secret information, government confidential information, and the like). Non-sensitive data may include inconsequential data of the given entity or data that may already be a matter of public record or knowledge. Examples of non-sensitive data include, but are not limited to, zip code of the given entity, a contact number of the given entity.

The term "*local legislative requirement*" refers to a legal requirement that all people and organizations, in a particular geographical area, are required to comply with. These local legislative requirements explicitly lay down nature of data that is to be anonymized for privacy protection of entities. In an embodiment, the local legislative requirements vary according to the geographical area. In such a case, different geographical areas may have different local legislative requirements. In an example, a local legislative requirement may be General Data Protection Regulation (GDPR), which is a regulation in European Union (EU) law on data protection and privacy in the EU and European Economic Area (EEA). It addresses transfer of personal data outside the EU and the EEA areas, aims to give control to data subjects over their personal data, and simplify regulatory environment for international business. For example, GDPR put limitations on locations where the personal data of the EU and EEA data subjects

can be stored. This is an important consideration when the data storage **108** is the remote data storage, as there are limitations on geographical location for storing the sensitive data due to the local legislative requirements such as GDPR. For example, one such requirement may be that financial information of a given entity cannot be stored in a foreign country. In such a case, if the financial information is backed up to the remote data storage in the foreign country, access to said data may not be allowed. Therefore, such a requirement may indicate that the financial information is to be anonymized.

In an embodiment, the personal information refers to any information that is associated with a particular entity rather than anyone else. Examples of the personal information may include, but are not limited to a name, a face, a social security number, a date of birth, a place of birth, financial information, biometric information, medical information, educational information, employment information, and business information. In an example, at certain geographical locations, storing images and/or videos of children is forbidden. In such a case, the indication relates to a requirement for anonymizing faces of the children in images and/or videos.

In an embodiment, the indication of what data is to be anonymized is provided by at least one of: a user, a standard setting, the data element. In an example, the controller **104** may be configured to receive a user input that is an indication of which setting or configuration to select for identifying sensitive data. For example, a user may configure which anonymization is likely applied to the data, such as General Data Protection Regulation (GDPR) anonymization, anonymization of a portion, such as credit card numbers only, and the like, as per need. Such settings and configurations which are selected or configured may alternatively be prestored in the local memory **102**. In another example, a standard setting may indicate that a specific portion of an audio (i.e. the data element) includes sensitive data that is to be anonymized. In another example, the data element may itself include an indication of sensitivity of its data, for example, as a sensitivity score.

In an embodiment, the controller **104** is further configured to parse for the data to be anonymized. In such a case, the controller **104** parses the data element to be stored to find the data to be anonymized. This parsing operation is optionally performed based on the received indication of what data is to be anonymized. The data to be anonymized may belong to one or more data portions of the data element.

According to an embodiment, the controller **104** is further configured to parse the data element to be stored to find the one or more data portions to be anonymized by finding data portions in data fields being specified as fields to be anonymized. Data fields herein refer to broad categories of the data where similar data can be stored in a structured form under a given category. It will be appreciated that data fields pertaining to sensitive data may be specified as the fields to be anonymized and upon parsing, the data portions in said data fields are identified to be anonymized. In an example, the data element may be the Table 1, and the column-wise data fields may be name, age, phone number, credit card number, and address. Here, the data field 'credit card number' may be identified (or indicated) as the field to be anonymized as it includes personal information, and therefore, data under the data field 'credit card number' is parsed and identified to be anonymized.

According to an embodiment, the controller **104** is further configured to parse the data element to be stored to find one or more data portions to be anonymized by finding data portions having data formats matching anonymous formats. Data formats herein refer to a defined format for the data elements such as number of characters or number of alphabets or number of numeric values used, or certain patterns. In an embodiment, the controller **104** is configured to optionally receive the anonymous formats from at least one of: the user, the standard setting. In an example, parsing the data element by finding data portions having data formats matching anonymous formats is executed by understanding the format of the data elements if the data that is received is structured, not structured, or has some unique patterns. In an example, data formats of data such as credit card number may have a unique format of numbers only of specified length, such as 16 digits or age has a format of 1 or 2 digits. The credit card numbers have very special format and checksum built in which helps in finding of data formats of credit card numbers. Like credit card numbers, unique patterns may be identified for data that is to be anonymized to segregate such data portion(s).

In an embodiment, the indication on what data is to be anonymized further relates to one or more locations of the data in the data element. The data element may include one or more data portions in which the data is stored, wherein different data portions have different locations in the data element. In an embodiment, when the data element is in the structured form, different data portions may correspond to different data fields. Therefore, an indication of location of a given data field whose data is to be anonymized facilitates in accurately determining which data within the data element is to be anonymized. As an example, when the data element is the Table 1, the indication on what data is to be anonymized may indicate that data in column 4 (i.e. the

column corresponding to data field 'credit card number') is to be anonymized. Indicating the one or more locations of the data to be anonymized enables the controller **104** to accurately identify said data and subsequently perform anonymization in a reliable manner.

In an embodiment, the user specifies the one or more locations of the data to indicate what data is to be anonymized using SQL or NoSQL statements. For example, the user may specify row(s) and/or column(s) of a table that include the data that is to be anonymized.

In another embodiment, the controller **104** specifies the one or more locations of the data (that is to be anonymized) in the data element. The controller **104** may optionally determine the one or more locations of the data when parsing the data element for the data to be anonymized.

In yet another embodiment, a database wherefrom the one or more data elements **106** are sent to the controller **104** employs a tool to automatically analyze the database and identify the one or more locations of the data that is to be anonymized. This database is implemented at the one or more external devices. Such a tool could be implemented as a sensitive data discovery tool that identifies data fields in the database, and determines locations of data fields that include the sensitive data. In an example, the sensitive data discovery tool may identify "credit card number" field as sensitive, and therefore data stored under said field is to be anonymized. It will be appreciated that the database uses a generic Application Programming Interface (API) approach to notify the controller **104** of the locations of the data that is to be anonymized in the data element, irrespective of which database is employed.

The controller **104** is further configured to generate a copy of the one or more data portions to be anonymized. The copy of the one or more data portions to be anonymized is generated to enable the controller **104** to separate the data portions to be anonymized from the rest of the data element received by the controller **104**. In an example, the copy of the one or more data portions to be anonymized is generated based on the location of the data portions. For example, when the data element received by the controller **104** is the Table 1, and it is determined that the data to be anonymized is located in the column 4 of the Table 1, then a copy of the column 4 that is to be anonymized may be generated.

The controller **104** is further configured to encrypt the copy of the one or more data portions to be anonymized utilizing an encryption key. This process of encryption is performed for ensuring security and confidentiality of the copy of the one or more data portions to be anonymized. By encryption, the copy of the one or more data portions to be anonymized is

converted into another format that is unreadable without the encryption key. It will be appreciated that the encryption performed by the controller **104** could be a symmetric encryption or an asymmetric encryption. In case of the symmetric encryption, the encryption key that is employed for encryption is also employed for decrypting the encrypted copy of the one or more data portions to be anonymized. In case of the asymmetric encryption, a decryption key (i.e. Private key) that is employed for decrypting the encrypted copy of the one or more data portions to be anonymized is different from the encryption key (i.e. Public key).

In an embodiment, the controller **104** is configured to employ an encryption algorithm to encrypt the copy of the one or more data portions to be anonymized. The encryption algorithm employed for encryption may be known in the art. Examples of the encryption algorithm may include, but are not limited to, a Triple Data Encryption Algorithm (TDEA), a Blowfish encryption algorithm, a Twofish encryption algorithm, an Advanced Encryption Standard (AES) algorithm, an International Data Encryption Algorithm (IDEA), a Rivest-Shamir-Adleman (RSA) algorithm.

The controller **104** is further configured to mask the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with an anonymous content thereby generating a modified data element to be stored. Optionally, the controller employs a masking algorithm for performing the aforesaid masking operation on the one or more data portions to be anonymized. The masking algorithm may be well known in the art. By replacing the one or more data portions to be anonymized with the anonymous content, there is performed selective anonymization on the data element to generate the modified data element. The modified data element is different from the data element received by the controller **104** for storage. The modified data element includes the anonymous content and one or more data portions that were not identified to be anonymized. These one or more data portions that were not identified to be anonymized are in their original form as in the data element which was received. The process of masking that is performed by the controller **104** is basically selective anonymization of the data element, wherein only specific data in the data element is anonymized while other data in the data element remains unchanged.

In an embodiment, the anonymous content is an identifier of the encrypted copy of the corresponding data portion. Both the anonymous content and the encrypted copy of the one or more data portions to be anonymized correspond to the same data portion(s) of the data element. The anonymous content may, for example, be the identifier of the encrypted copy of the

corresponding data portion, and therefore facilitate in establishing an association between the anonymous content and the encrypted copy. Using the identifier of the encrypted copy of the corresponding data portion as the anonymous content provides an efficient and secure anonymization effect in the modified data element.

In an embodiment, the identifier of the encrypted copy of the corresponding portion is a string of one or more of: alphabets, numerals, special characters, symbols, or a combination of these. The identifier uniquely refers to the encrypted copy of the corresponding data portion. As an example, the identifier of the encrypted copy may be 'EC001'.

In an embodiment, the anonymous content is the encrypted copy of the corresponding data portion. In such a case, the one or more data portions to be anonymized are replaced with the encrypted copy of the corresponding data portion. As a result, in the modified data element, the one or more data portions to be anonymized are in their encrypted form, instead of their original form. As the encrypted copy already protects the confidentiality of the corresponding data portion in the encrypted form, replacing the corresponding data portion with the encrypted copy provides a secure selective anonymization effect in the modified data element.

In an embodiment, the anonymous content is a masking content. In this regard, the one or more data portions to be anonymized are hidden (namely, obfuscated) by the masking content (upon replacement with the anonymous content) to protect the sensitive data in the one or more data portions to be anonymized. The "*masking content*" is content that is used for replacing content in the one or more data portions to be anonymized, thereby effectively masking the content in the one or more data portions. In an embodiment, the masking content is a string of numbers, characters, and/or alphabets. For example, when the data portion to be anonymized includes an original credit card number "4580123434001999", a masking content employed as the anonymous content may be "1111222233334444". In such a case, the modified data element would include the anonymous content "1111222233334444" instead of the original credit card number. In another embodiment, the masking content is represented as NIL (namely, empty). The one or more data portions to be anonymized are replaced with NIL. According to an embodiment, NIL corresponds to bit '0'. When such a masking content is employed by the controller **104** for selective anonymization, the one or more data portions to be anonymized are replaced with the bit '0' while the other portions which are not to be anonymized remain unchanged. Beneficially, masking the data to be anonymized by replacing with the masking

content which is NIL also enables in reducing a data storage space required for storing the modified data element.

As an example, the data element received by the controller **104** may be the Table 1. Upon replacing a given data portion (for example, column 4) of the Table 1 with the anonymous content, there may be generated a modified data element that is a Table 2 (given below). Here, the masking content employed as the anonymous content may be "1111222233334444".

Name	Age	Phone Number	Credit Card Number	Address
Dani	33	0541111111	1111222233334444	A street
Josef	55	0541222222	1111222233334444	B street

Table 2

In another example, the data element (before anonymization) that is received by controller **104** may have the alphanumeric character string**:** Dani, 33, 0541111111, 4580800080001999, A street; Josef, 55, 0541222222, 4580123434001999, B street. Further, upon replacing a given data portion of this string with the anonymous content, the modified data element (generated upon anonymization) may have the alphanumeric character string**:** Dani, 33, 0541111111, 1111222233334444, A street, Josef, 55, 0541222222, 1111222233334444, B street.

In an embodiment, the controller **104** is further configured to mask the one or more data portions to be anonymized and generate the modified data element to be stored by utilizing a write-with-mask command indicating location of data to be anonymized and the anonymous content to replace the data to be anonymized. The write-with-mask command is an input/output write command that hints the data storage arrangement **100A** on areas or portions of the data element which may be modified during data anonymization. The location of data to be anonymized and the anonymous content are provided as an input while utilizing the write-with-mask command, and the modified data element is provided as an output after the write-with-mask command is executed. In an embodiment, the controller **104** executes the write-with-mask command.

In an embodiment, the write-with-mask command is utilized by writing it to a file. In an embodiment, the write-with-mask command is utilized by writing it to a block device. Herein, the term "*block device*" refers to a computer data storage device that supports reading and optionally, writing data in fixed-size blocks, sectors, or clusters. A size of the block device is generally 512 bytes or a multiple thereof. Generally, the data stored in the block devices is

stored in a certain file system format. Optionally, the write-with-mask command is written to the local memory **102**. Alternatively, optionally, the write-with-mask command is written to the data storage **108**.

In an embodiment, the write-with-mask command is represented by a function (1) that is given below:

WriteWithMask (Data, Device, offset, size, dataMask, bool isSensitive) (1)

wherein

'*Data*' refers to the data to be written (i.e. the anonymous content), it can optionally be expressed as (char *), wherein '*char**' refers to a pointer that holds address (namely, location) of the data;

'*Device*' refers to the device where the data is written, it can be a file, or a Logical Unit (LU);

'*offset*' refers to the offset where the data is written (i.e. in the file or the LU);

'*size*' refers to the size of the data, which is typically a multiple of 512 bytes for block devices;

'*dataMask*' refers to a data structure describing which parts of the data may be changed during anonymization; and

'*bool isSensitive*' refers to a Boolean indicator as to whether or the data to be anonymized is present.

In this regard, the dataMask is used for any change an application can cause to the data that the controller **104** hints to the local memory **102**. The dataMask can be a bitmap, where each bit indicates a byte in the data, or it may be kept in a different format.

In an embodiment, the write-with-mask command comprises an indicator to whether the data to be anonymized is present, and the controller **104** is configured to determine if the data to be anonymized is present prior to masking the one or more data portions to be anonymized. In the function (1), '*bool isSensitive*' is the indicator to whether the data to be anonymized is present. A value of this indicator indicates whether the data element has the one or more data portions which comprise the data (namely, the sensitive data) to be anonymized. It will be appreciated that the indication is provided by a Boolean variable (represented as the term "*bool*" in the function (1)), whose value is either "*true*" (namely, 1) to indicate that the data to be anonymized

is present, or "*false*" (namely, 0) to indicate that the data to be anonymized is not present. As an example, if the data element already includes anonymized data, then the value of '*bool isSensitive*' is false. This indicator facilitates in improving the process of anonymization performed by the controller **104**, as processing resources for anonymization will be spent by the controller **104** only when the data to be anonymized is present prior to masking.

It will be also appreciated that sensitive bit (namely, "*bool isSensitive*") utilized in the write-with-mask command can optionally also be used for encryption of the copy of the one or more data portions to be anonymized. For example, when the sensitive bit is true, the copy of the one or more data portions to be anonymized would be stored in the encrypted form in a manner that the sensitive data in the data portions would remain encrypted and would not be accessed without the encryption key. However, the non-sensitive data would be stored and accessed in its usual form (i.e. without encrypting).

In an embodiment, when the controller **104** is configured to mask the one or more data portions to be anonymized, a database writing to the local memory **102** can leverage the write-with-mask command to notify the local memory **102**, which is a lower level storage, regarding the one or more data portions of the data element which may be changed during anonymization.

In an example, the data element may be the Table 1 given earlier, and the database may serialize the Table 1 when writing it to the local memory **102**. In such a case, the database can understand the location where the data to be anonymized is written and may mark this data in the mask. For example, data in the first row in the Table 1 may be serialized as "Dani330541111458080008000 1999Astreet" and a corresponding data mask for this data may be "00000000000000001111111111111111000000". Herein, "1" indicates that data in that location is to be anonymized and the data comprises sensitive data. In this case, the locations where there is "1" are indicating locations of the credit card number "4580800080001999". Moreover, the database may also mark whether the data to be anonymized comprises sensitive data. For example, the data corresponding to the "*Credit card number*" column would be marked as sensitive = "*true*" as the data is in its original form.

The controller **104** is further configured to store the modified data element to be stored in the data storage **108**. In such a case, the modified data element is sent by the controller **104** to the data storage **108**, for storing at the data storage **108**. As an example, the modified data element may be sent over a communication network such as the Internet to the remote data storage.

The controller **104** is further configured to store the encrypted copy of the one or more data portions to be anonymized in the data storage **108**. In such a case, the encrypted copy is sent by the controller **104** to the data storage **108**, for storing at the data storage **108**. Moreover, as the encrypted copy only corresponds to sensitive data and not an entirety of data in the data element, the encrypted copy is small in size, which enables the controller **104** in efficiently utilizing storage space of the data storage **108**. As an example, the encrypted copy may be sent over a communication network such as the Internet to the remote data storage.

The controller **104** is further configured to store the encryption key in the local memory **102**. The encryption key is available only on-site (or on-premises) of the data storage arrangement **100A**, and the encryption key cannot be accessed in the data storage **108**. By storing the encryption key at the local memory **102**, away from the data storage **108**, there is ensured security of the encrypted copy stored at the data storage since unauthorized entities that may gain access to the data storage **108** would still be unable to decrypt the encrypted copy to access sensitive data without the encryption key.

It will be appreciated that the modified data element and the encrypted copy of the one or more data portions to be anonymized are stored separately from the encryption key. The modified data element is selectively anonymized and the encrypted copy is secured for confidentiality protection by encryption, therefore both of these can be effectively stored at the data storage **108** (which may be the local storage, or potentially the remote data storage) without any data security or privacy issues such as data leakage or non-compliance with data confidentiality regulations (such as the local legislative requirements), as the encryption key is stored at a different storage (i.e. the local memory **102**). Only authorized entities would have access to both the local memory **102** and the data storage **108**, and therefore be able to have complete access to the modified data element, the encrypted copy, and the encryption key. Such authorized entities would in turn, have access to sensitive data. However, even if unauthorized entities gain access to the data storage **108**, the sensitive data remains secure and inaccessible to the unauthorized entities as it is available only in anonymized form and encrypted form at the data storage **108**. Moreover, when the data storage **108** is subject to the data confidentiality regulations, such a manner of storing ensures proper compliance with these regulations.

In an embodiment, the controller **104** is further configured to:

- generate a copy of the modified data element to be stored utilizing deduplication; and

- store the copy of the modified data element to be stored as the modified data element to be stored in the data storage **108**.

Herein, the term "*deduplication*" refers to a process for eliminating duplicate (namely, redundant) data. Deduplication enables the controller **104** to prevent storing of duplicate data in the data storage **108**. As a result, a large amount of storage space is saved which further improves system performance of the data storage **108**.

In an embodiment, in the deduplication process, the copy of the modified data element undergoes variable length deduplication. The copy of the modified data element is chunked into different variable sized chunks (i.e. variable data chunks) based on a chunking algorithm. In an embodiment, each chunk comprises at least one data portion of the copy of the modified data element. In another embodiment, in the deduplication process, the copy of the modified data element undergoes fixed size deduplication. Herein, the copy of the modified data element is chunked into fixed size chunks. Typically, in the fixed size deduplication, size of each chunk is 8 Kilobytes (KB). According to an embodiment, the controller **106** is configured to employ a chunking algorithm to chunk the given data element. The chunking algorithm may be well known in the art.

Further, in the deduplication process, the controller **104** is configured to hash each chunk using a hashing algorithm. Thus, each chunk has a hash signature (or simple, a hash) which is generated using the hashing algorithm. In an example, content-based hashing algorithm, such as a Rabin hash, may be used to segregate the copy of the modified data element on locations based on the content of copy of the modified data element. Then, in the deduplication process, the controller **104** is configured to compare the hash signature of each chunk of the copy of the modified data element with hash signatures of chunks previously stored in the data storage **108**. If a hash signature of a given chunk of the copy of the modified data element is identical to the hash signature of any chunk amongst the chunks previously stored in the data storage **108**, then the given chunk of the copy of the modified data element is identified as a duplicate chunk and only a pointer to the previously-stored chunk having the identical hash signature with the given is stored (identical chunks are not stored) in the data storage **108**, which significantly reduces the storage space as compared to conventional systems and methods. In a case where the hash signature of the given chunk is not identical to the hash signatures of any of the chunks previously stored in the data storage **108**, then the given chunk is stored in the data storage **108** and the hash signature of the given chunk is added to a table of the previous hash signatures of

the chunks stored at the data storage **108**. It will be appreciated that compression ratios are high due to the similarity of the chunks. Therefore, the deduplication leads to an efficient utilization of storage space at the data storage **108**.

As an example, five chunks A1, B1, C, D, and E may already be stored at the data storage **108**. Hash signatures (or hash values) of the five chunks A1, B1, C, D, and E may be h_A, h_B, h_C, h_D, and h_E. The copy of the modified data element to be stored may be chunked into 3 chunks A2, B2, and F having hash signatures h_A, h_B, and h_F. Notably, the hash signatures of the chunks A2 and B2 of the copy of the modified data element are identical to the hash signatures of the chunks A1 and B1 previously stored in the data storage **108**. In such a case, when storing the copy of the modified data element as the modified data element to be stored in the data storage **108**, pointers to the previously-stored chunks A1 and B1 may be stored in the data storage **108**, and the chunk F of the copy of the modified data element may be stored in the data storage **108**.

In an embodiment, the controller **104** is further configured to:

- generate a copy of the modified data element to be stored utilizing differential compression; and
- store the copy of the modified data element to be stored as the modified data element to be stored in the data storage **108**.

Differential compression refers to finding similar data portions between the copy of the modified data element and modified data elements that are already stored at the data storage **108**, and compressing a similar data portion in the copy of the modified data element using another data portion that is already stored as reference. Differential compression enables much better compression as the similar data portion is already present in the modified data elements stored at the data storage **108**. In an embodiment, when utilizing differential compression, a hash for a first data portion of the copy of the modified data element is generated and compared to a stored hash for a second data portion that is already stored in the data storage **108**, and if the hashes match, the first data portion is compressed with reference to the second data portion. Notably, differential compression enables the controller **104** to store data portions of the copy of the modified data element that have matching hashes in a compressed form to save storage space at the data storage **108**.

The controller **104** is optionally configured to generate the hash (may also be referred to as similarity hash digest) for the first data portion of the copy of the modified data element using a hashing algorithm. Moreover, the hashing algorithm is used for all data portions of the modified data elements previously stored at the data storage **108** to enable detection of duplicate data portions of any modified data element that is to be stored in the data storage **108**. In an example, a content-based hashing algorithm, such as a Rabin hash, may be used to segregate the modified data element on locations based on the content of modified data element. Further, if the hash of the first data portion of the copy of the modified data element is identical to the stored hash of the second data portion previously stored in the data storage **108**, then the first data portion of the copy of the modified data element is identified as a duplicate data portion and then this first data portion is compressed with reference to the second data portion. As a result, when the copy of the modified data element is stored as the modified data element in the data storage **108**, the required storage space at the data storage **108** is significantly reduced compared to conventional systems and methods. In an example, the compression of the first data portion with reference to second data portion may be executed using compression algorithms known in the art.

In an embodiment, the controller **104** may calculate hash digest and similarity hash digests. Hash digest simply refers to hash values of all data portions of the copy of the modified data element and similarity hash digests refers, for example, to hash value of the first data portion after masking out the data that is already anonymized in the first data portion. The similarity hash digest can be calculated for any data portion of the copy of the modified data element, with or without removing the anonymized data. In an embodiment, the hash digest and the similarity hash digest is generated using hashing algorithms known in the art. In such an example, if the hash digest already exists in the data storage **108**, then the controller **104** identifies that an identical data portion of the copy of the modified data element already exists at the data storage **108** and keeps a pointer instead. If the controller **104** identifies another data portion (such as second data portion) at the data storage **108** with the same similarity hash digests, the controller **104** may compress the first data portion with the second data portion as a reference. In other words, data other than data that is already anonymized, is same for both the first data portion and the second data portion. As a result, a large amount of storage space is saved at the data storage **108** which further improves system performance of the data storage **108**. When two data portions have same similarity hash digests (or simple, similarity hash), the two data portions are considered to be similar as a range of data in a data portion is identical to

a range of data in another data portion. Thus, part of one data portion is identical to part of another data portion and overall, the two data portions are similar but not identical. Differential compression allows storage of only non-identical (or different) parts of data portions at the data storage **108** and use of the pointer for referring to similar pre-stored data portions at the data storage **108**, in order to efficiently utilize the storage space of the data storage **108**.

As an example, the copy of the modified data element may include the serialized data represented as a first sequence as "Dani330541111111111222233334444Astreet", where data "1111222233334444" is the anonymous content. Here, the similarity hash digest is calculated for "Dani330541111111Astreet". The similarity hash digest for the non-anonymized data in the first sequence is compared with a similarity hash digest for another sequence such as a second sequence that is already stored at the data storage **108**. The second sequence may, for example, be "Dani330541111111000000000000000Astreet", wherein "0000000000000000" is the anonymous content. The similarity hash digest for the second sequence is calculated for "Dani330541111111Astreet", and is therefore identical with the similarity hash digest for the first sequence. In such a case, the portion of the first sequence that is similar to a corresponding portion of the second sequence is compressed using the second sequence in the modified data elements that are already stored at the data storage **108**, in order for storage space savings at the data storage **108**.

FIG. 1B is illustrates a block diagram of a data storage arrangement, in accordance with another embodiment of the present disclosure. With reference to FIG. 1B, there is shown a data storage arrangement **100B**. The data storage arrangement **100B** comprises a local memory **102** that is configured to store one or more data elements **106**. The data storage arrangement **100B** further comprises a storing software module **112** for storing at least one of the one or more data elements **106**. The data storage arrangement **100B** further comprises: a receiving software module **114** for receiving the data element to be stored; a copying software module **116** for generating a copy of one or more data portions to be anonymized; an encrypting software module **118** for encrypting the copy of the one or more data portions to be anonymized utilizing an encryption key; a masking software module **120** for masking the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with anonymous content thereby generating a modified data element to be stored; and a software module **122** for storing the modified data element to be stored in a data storage **108** and for storing the encrypted copy of the one or more data portions to be anonymized in the data storage

108 and for storing the encryption key in the local memory **102**. The data storage arrangement **100B** is also shown to include the controller **104** and the network interface **110**.

The storing software module **112**, when executed by the controller **104**, is configured for storing at least one of the one or more data elements **106**. The storing software module **112** utilizes anonymization and anonymization-aware encryption when storing at least one of the one or more data elements **106**. Anonymization is executed on the one or more data portions to be anonymized in the data element. Anonymization-aware encryption is executed on the copy of one or more data portions to be anonymized.

In an exemplary implementation, the storing software module **112** may include at least one of: the receiving software module **114**, the copying software module **116**, the encrypting software module **118**, the masking software module **120**, the software module **122**. In an implementation, the storing software module **112** is potentially implemented as at least one circuit in the data storage arrangement **100B**. In another exemplary implementation, the receiving software module **114**, the copying software module **116**, the encrypting software module **118**, the masking software module **120**, and the software module **122** are implemented as separate circuitry that is operationally communicatively coupled to each other and are executed by the controller **104** of the data storage arrangement **100B** to perform their respective operations.

The receiving software module **114**, which when executed by the controller **104**, receives the data element to be stored. The data element to be stored is received from one or more external devices that is/are communicatively coupled to the data storage arrangement **100B** via a communication network. The one or more external devices is/are the user devices or the primary storage system(s).

The copying software module **116**, which when executed by the controller **104**, generates a copy of one or more data portions to be anonymized. The copy of the one or more data portions to be anonymized is generated to separate the data portions to be anonymized from the rest of the data element received by the receiving software module **114**.

The encrypting software module **118**, which when executed by the controller **104**, encrypts the copy of the one or more data portions to be anonymized utilizing an encryption key. This process of encryption is performed for ensuring security and confidentiality of the copy of the one or more data portions to be anonymized that was generated by the copying software module **116**.

The masking software module **120**, which when executed by the controller **104**, masks the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with anonymous content thereby generating a modified data element to be stored. The masking software module **120** performs selective anonymization on the data element by replacing only the one or more data portions (which include sensitive data) to be anonymized with the anonymous content to generate the modified data element. The modified data element provides privacy protection as the anonymous content enables in maintaining confidentiality of the sensitive data.

The software module **122**, which when executed by the controller **104**, stores the modified data element to be stored in the data storage **108**, stores the encrypted copy of the one or more data portions to be anonymized in the data storage **108**, and stores the encryption key in the local memory **102**. Storing the modified data element and the encrypted copy of the one or more data portions to be anonymized separately from the encryption key enables the data storage arrangement **100B** to provide data security and privacy protection. Only authorized entities would have access to both the local memory **102** and the data storage **108**, and therefore be able to access all data. Even if unauthorized entities gain access to the data storage **108**, sensitive data remains secure as it is available only in anonymized form and encrypted form at the data storage **108**. In an embodiment, the software module **122** is a part of the storing software module **112**.

FIG. 2 illustrates an environment in which a data storage arrangement is used, in accordance with an embodiment of the present disclosure. With reference to FIG. 2, there is shown an environment **200** in which a data storage arrangement **202** is used, in accordance with an embodiment of the present disclosure. Herein, a data element **204** is shown to comprise 9 data portions (depicted as hatched portions), wherein 7 data portions (depicted as horizontal stripes hatch) comprise non-sensitive data and 2 data portions (depicted as vertical stripes hatch) comprise sensitive data. The 2 data portions corresponding to the sensitive data are to be anonymized. Herein, the data element **204** is received and then stored in a local memory (not shown) of the data storage arrangement **202**. A modified data element **206** is generated by a controller (not shown) of the data storage arrangement **202**. The modified data element **206** comprises 9 data portions, wherein 7 data portions (depicted as horizontal stripes hatch) comprise the non-sensitive data and 2 data portions (depicted as diagonal brick hatch) comprise anonymous content. The modified data element **206** is then stored in a data storage **208** (depicted, as a remote data storage such as a cloud-based data storage). An encrypted copy **210**

(depicted as zig-zag hatch) of the 2 data portions to be anonymized is also stored in the data storage **208**.

FIGs. 3A and 3B illustrate steps of a method for a data storage arrangement, in accordance with an embodiment of the present disclosure. The method **300** is executed at a data storage arrangement (e.g. the data storage arrangement **100A** or **100B** or **202**) described, for example, in FIGs. 1A, 1B and 2. The method **300** includes the steps **302** to **314**.

The method **300** for a data storage arrangement (e.g. the data storage arrangement **100A** or **100B** or **202**) comprising a local memory **102** being configured to store one or more data elements **106**, the data storage arrangement **100A**, **100B**, **202** being connected to a data storage (e.g. the data storage **108** or **208**) for storing at least one of the one or more data elements **106**, wherein the method **300** comprises receiving a data element to be stored; generating a copy of one or more data portions to be anonymized; encrypting the copy of the one or more data portions to be anonymized utilizing an encryption key; masking the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with anonymous content thereby generating a modified data element to be stored; storing the modified data element to be stored in the data storage **108**, **208**; storing the encrypted copy of the one or more data portions to be anonymized in the data storage **108**, **208**; and storing the encryption key in the local memory **102**.

At step **302**, the method **300** comprises receiving a data element to be stored. The data element to be stored is received from one or more external devices that is/are communicatively coupled to the data storage arrangement **100A**, **100B**, **202**. The one or more external devices is/are the user devices or the primary storage system(s).

At step **304**, the method **300** further comprises generating a copy of one or more data portions to be anonymized. The copy of the one or more data portions to be anonymized is generated to separate the data portions to be anonymized from the rest of the data element that was received at the step **302**.

At step **306**, the method **300** further comprises encrypting the copy of the one or more data portions to be anonymized utilizing an encryption key. This process of encryption is performed for ensuring security and confidentiality of the copy of the one or more data portions to be anonymized that was generated at the step **304**.

At step **308**, the method **300** further comprises masking the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with anonymous content thereby generating a modified data element to be stored. There is performed selective anonymization on the data element by replacing only the one or more data portions (which include sensitive data) to be anonymized with the anonymous content to generate the modified data element. The modified data element provides privacy protection as the anonymous content enables in maintaining confidentiality of the sensitive data.

At step **310**, the method **300** further comprises storing the modified data element to be stored in the data storage **108**, **208**. The modified data element that provides the privacy protection for the sensitive data can be safely stored at the data storage **108**, **208** without risk to privacy as the sensitive data is replaced by the anonymous content in the modified data element.

At step **312**, the method **300** further comprises storing the encrypted copy of the one or more data portions to be anonymized in the data storage **108**, **208**. The encrypted copy generated at the step 306 is already secure owing to encryption, and therefore can be safely stored at the data storage **108**, **208** without risk of data leakage. Moreover, as the encrypted copy only corresponds to sensitive data and not an entirety of data in the data element, the encrypted copy can be stored in a space-efficient manner at the data storage **108**, **208**.

At step **314**, the method **300** further comprises storing the encryption key in the local memory **102**. As the encryption key is stored away from the encrypted copy, there is provided protection against data leakage or privacy issues. Moreover, when the data storage **108**, **208** is subject to data confidentiality regulations, a manner of storing of the steps **310**, **312**, and **314** ensures proper compliance with these regulations.

The steps **302** to **314** are only illustrative and other alternatives can also be provided where one or more steps are added, one or more steps are removed, or one or more steps are provided in a different sequence without departing from the scope of the claims herein.

In one aspect, there is provided a computer-readable medium carrying computer instructions that when loaded into and executed by a controller (e.g. the controller **104**) of a data storage arrangement (e.g. the data storage arrangement **100A** or **100B** or **202**) enables the data storage arrangement to implement the method **300**. The computer readable medium may include, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable

combination of the foregoing. Examples of implementation of the computer-readable medium include, but are not limited to, Electrically Erasable Programmable Read-Only Memory (EEPROM), Random Access Memory (RAM), Read Only Memory (ROM), Hard Disk Drive (HDD), Flash memory, a Secure Digital (SD) card, Solid-State Drive (SSD), a computer readable storage medium, and/or CPU cache memory.

Modifications to embodiments of the present disclosure described in the foregoing are possible without departing from the scope of the present disclosure as defined by the accompanying claims. Expressions such as "including", "comprising", "incorporating", "have", "is" used to describe and claim the present disclosure are intended to be construed in a non-exclusive manner, namely allowing for items, components or elements not explicitly described also to be present. Reference to the singular is also to be construed to relate to the plural. The word "exemplary" is used herein to mean "serving as an example, instance or illustration". Any embodiment described as “exemplary” is not necessarily to be construed as preferred or advantageous over other embodiments and/or to exclude the incorporation of features from other embodiments. The word "optionally" is used herein to mean "is provided in some embodiments and not provided in other embodiments". It is appreciated that certain features of the present disclosure, which are, for clarity, described in the context of separate embodiments, may also be provided in combination in a single embodiment. Conversely, various features of the invention, which are, for brevity, described in the context of a single embodiment, may also be provided separately or in any suitable combination or as suitable in any other described embodiment of the disclosure.

CLAIMS

What is claimed is:

1. A data storage arrangement (100A, 100B, 202) comprising a local memory (102) and a controller (104), the local memory (102) being configured to store one or more data elements (106), the data storage arrangement (100A, 100B, 202) being connected to a data storage (108, 208) and the controller (104) being configured to store at least one of the one or more data elements (106), wherein the data storage arrangement (100A, 100B, 202) is characterized in that the controller (104) is further configured to:

- receive a data element to be stored;
- generate a copy of one or more data portions to be anonymized;
- encrypt the copy of the one or more data portions to be anonymized utilizing an encryption key;
- mask the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with an anonymous content thereby generating a modified data element to be stored;
- store the modified data element to be stored in the data storage (108, 208);
- store the encrypted copy of the one or more data portions to be anonymized in the data storage (108, 208); and
- store the encryption key in the local memory (102).

2. The data storage arrangement (100A, 100B, 202) according to claim 1, wherein the anonymous content is an identifier of the encrypted copy of the corresponding data portion.

3. The data storage arrangement (100A, 100B, 202) according to claim 1, wherein the anonymous content is the encrypted copy of the corresponding data portion.

4. The data storage arrangement (100A, 100B, 202) according to claim 1, wherein the anonymous content is a masking content.

5. The data storage arrangement (100A, 100B, 202) according to any previous claim, wherein the controller (104) is further configured to receive an indication on what data is to be anonymized, wherein the indication relates to one or more of local legislative requirements, personal information and/or faces.

6. The data storage arrangement (100A, 100B, 202) according to claim 5, wherein the controller (104) is further configured to parse for the data to be anonymized.

7. The data storage arrangement (100A, 100B, 202) according to claim 5 or 6, wherein the indication on what data is to be anonymized further relates to one or more locations of the data in the data element.

8. The data storage arrangement (100A, 100B, 202) according to claim 7, wherein the controller (104) is further configured to mask the one or more data portions to be anonymized and generate the modified data element to be stored by utilizing a write-with-mask command indicating location of data to be anonymized and the anonymous content to replace the data to be anonymized.

9. The data storage arrangement (100A, 100B, 202) according to claim 8, wherein the write-with-mask command comprises an indicator to whether the data to be anonymized is present, and the controller (104) is configured to determine if the data to be anonymized is present prior to masking the one or more data portions to be anonymized.

10. The data storage arrangement (100A, 100B, 202) according to any preceding claim, wherein the controller (104) is further configured to:

- generate a copy of the modified data element to be stored utilizing differential compression; and
- store the copy of the modified data element to be stored as the modified data element to be stored in the data storage (108, 208).

11. The data storage arrangement (100A, 100B, 202) according to any of claims 1 to 10, wherein the controller (104) is further configured to:

- generate a copy of the modified data element to be stored utilizing deduplication; and
- store the copy of the modified data element to be stored as the modified data element to be stored in the data storage (108, 208).

12. A method (300) for a data storage arrangement (100A, 100B, 202) comprising a local memory (102) being configured to store one or more data elements (106), the data storage arrangement (100A, 100B, 202) being connected to a data storage (108, 208) for storing at least one of the one or more data elements (106), wherein the method (300) is characterized in that the method (300) comprises:

- receiving a data element to be stored;
- generating a copy of one or more data portions to be anonymized;
- encrypting the copy of the one or more data portions to be anonymized utilizing an encryption key;
- masking the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with anonymous content thereby generating a modified data element to be stored;
- storing the modified data element to be stored in the data storage (108, 208);
- storing the encrypted copy of the one or more data portions to be anonymized in the data storage (108, 208); and
- storing the encryption key in the local memory (102).

13. A computer-readable medium carrying computer instructions that when loaded into and executed by a controller (104) of a data storage arrangement (100A, 100B, 202) enables the data storage arrangement (100A, 100B, 202) to implement the method (300) according to claim 12.

14. A data storage arrangement (100A, 100B, 202) comprising a local memory (102) being configured to store one or more data elements (106), and the data storage arrangement (100A, 100B, 202) further comprising a storing software module (112) for storing at least one of the one or more data elements (106), wherein the data storage arrangement (100A, 100B, 202) is characterized in that the data storage arrangement (100A, 100B, 202) further comprises:

- a receiving software module (114) for receiving the data element to be stored;
- a copying software module (116) for generating a copy of one or more data portions to be anonymized;
- an encrypting software module (118) for encrypting the copy of the one or more data portions to be anonymized utilizing an encryption key;
- a masking software module (120) for masking the one or more data portions to be anonymized, by replacing the one or more data portions to be anonymized with anonymous content thereby generating a modified data element to be stored; and
- a software module (122) for storing the modified data element to be stored in a data storage (108, 208) and for storing the encrypted copy of the one or more data portions to be anonymized in the data storage (108, 208) and for storing the encryption key in the local memory (102).

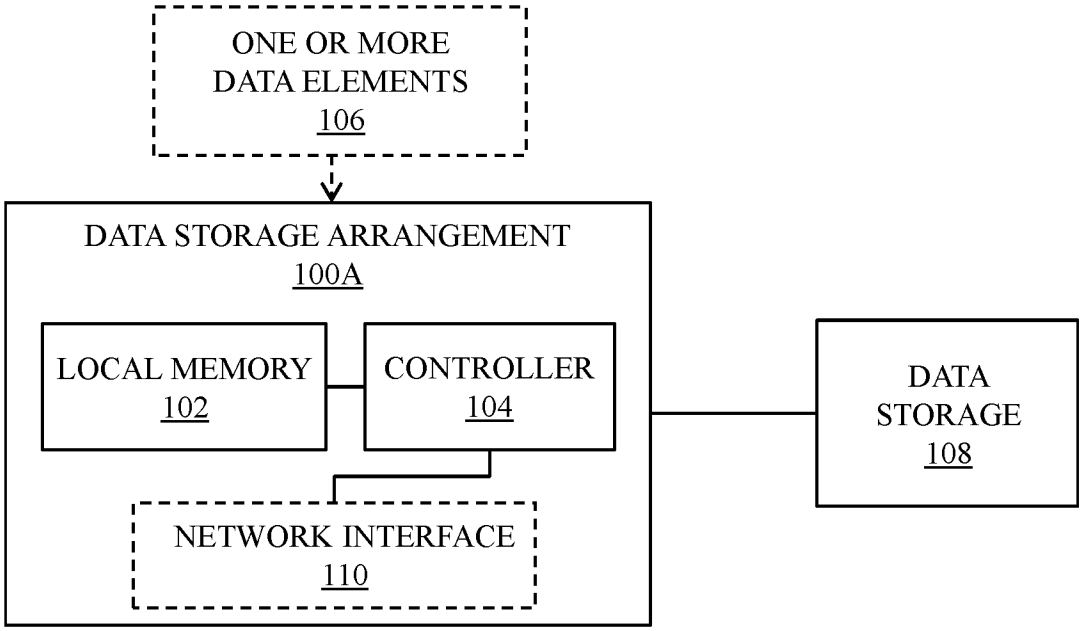

ONE OR MORE
DATA ELEMENTS
106
DATA STORAGE ARRANGEMENT
100A
LOCAL MEMORY
102
CONTROLLER
104
NETWORK INTERFACE
110
DATA
STORAGE
108

FIG. 1A

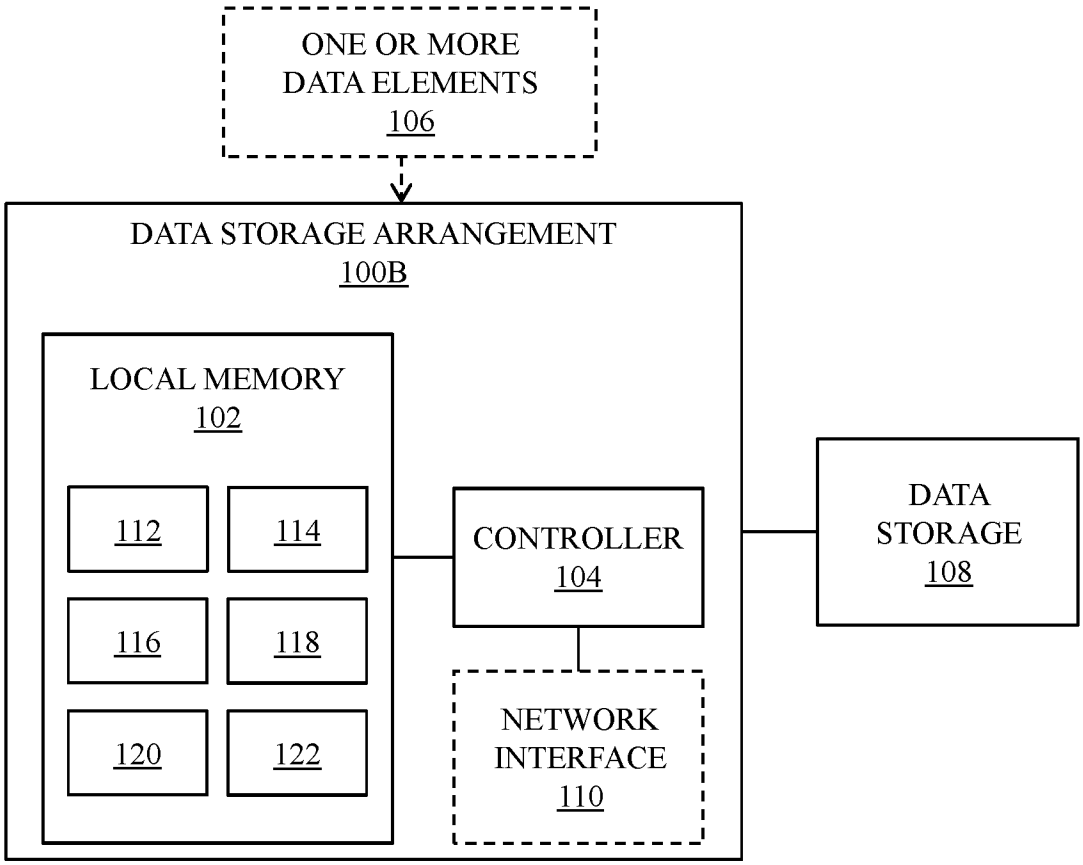

ONE OR MORE
DATA ELEMENTS
106
DATA STORAGE ARRANGEMENT
100B
LOCAL MEMORY
102
112
114
116
118
120
122
CONTROLLER
104
NETWORK
INTERFACE
110
DATA
STORAGE
108

FIG. 1B

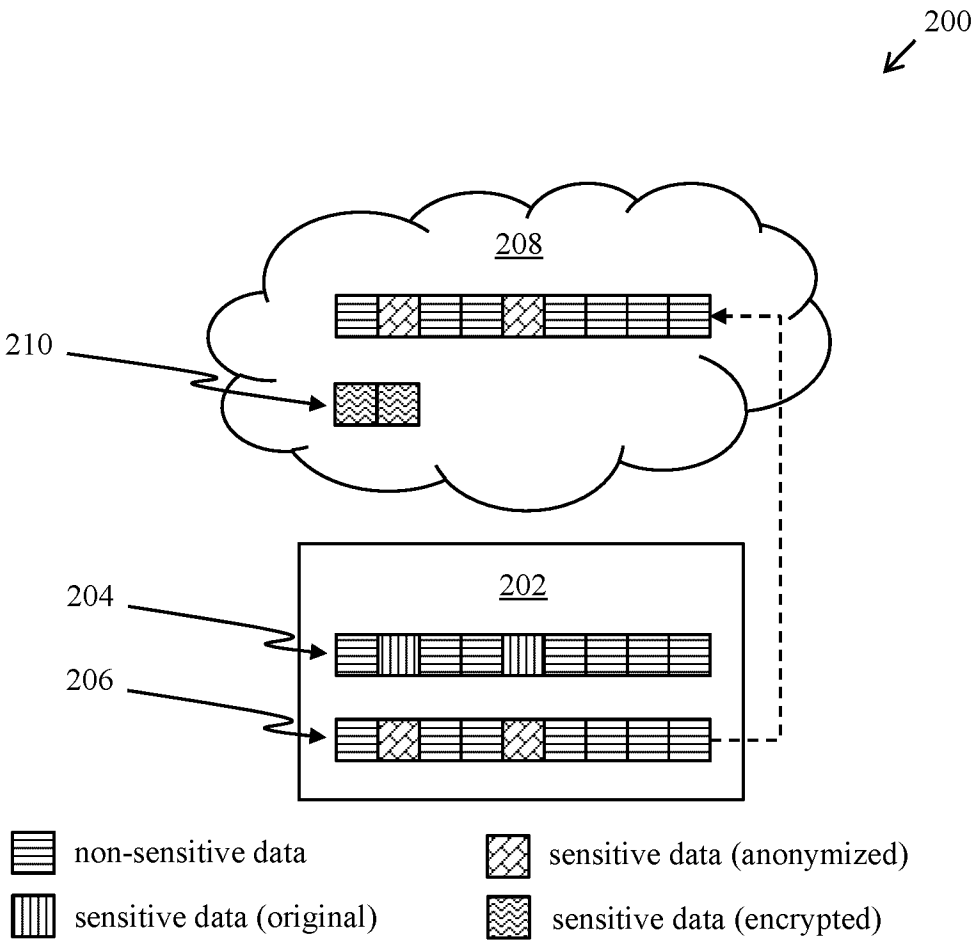

200
208
210
202
204
206
non-sensitive data
sensitive data (anonymized)
sensitive data (original)
sensitive data (encrypted)

FIG. 2

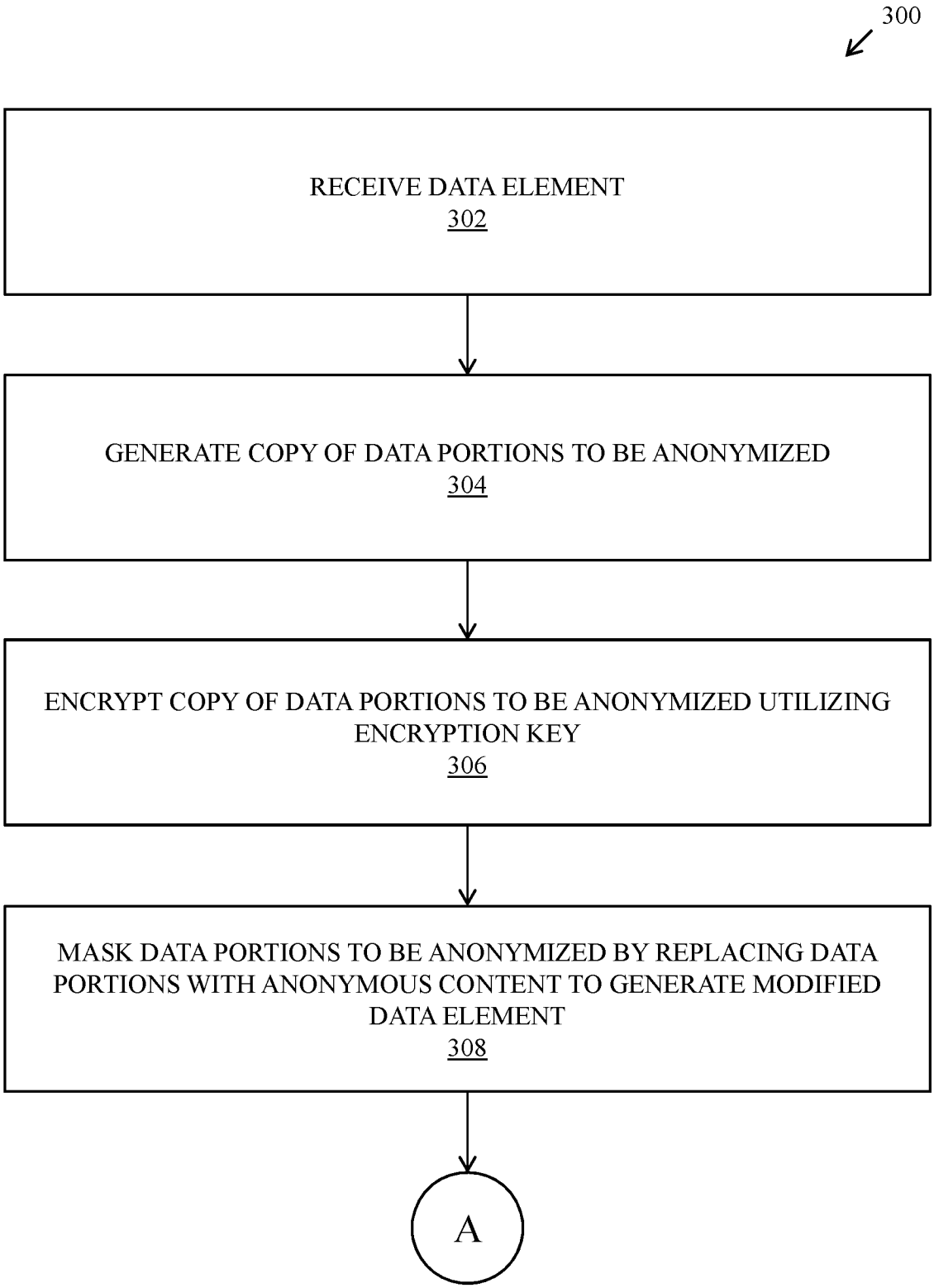

300
RECEIVE DATA ELEMENT
302
GENERATE COPY OF DATA PORTIONS TO BE ANONYMIZED
304
ENCRYPT COPY OF DATA PORTIONS TO BE ANONYMIZED UTILIZING ENCRYPTION KEY
306
MASK DATA PORTIONS TO BE ANONYMIZED BY REPLACING DATA PORTIONS WITH ANONYMOUS CONTENT TO GENERATE MODIFIED DATA ELEMENT
308
A

FIG. 3A

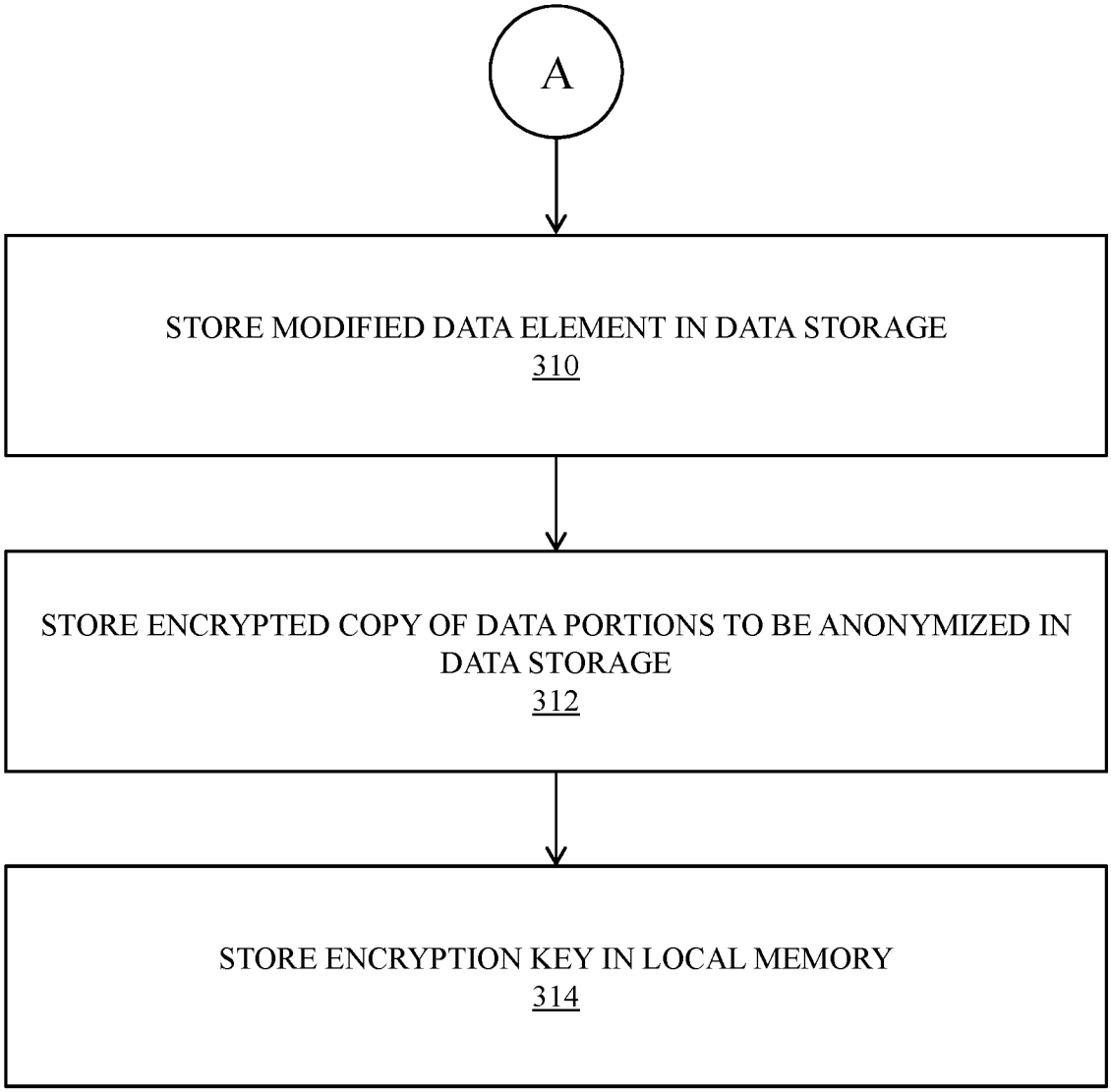

A
STORE MODIFIED DATA ELEMENT IN DATA STORAGE
310
STORE ENCRYPTED COPY OF DATA PORTIONS TO BE ANONYMIZED IN DATA STORAGE
312
STORE ENCRYPTION KEY IN LOCAL MEMORY
314

FIG. 3B

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2020/078524

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/62
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)
G06F

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 760 711 B1 (GILLETT DON M [US] ET AL) 6 July 2004 (2004-07-06)	1-7, 12-14
A	abstract column 7, paragraph 11 - column 8, paragraph 13	8-11

X	US 2015/169886 A1 (BHAGWAN VARUN [US] ET AL) 18 June 2015 (2015-06-18)	1-7, 12-14
A	abstract paragraph [0029] - paragraph [0032]	8-11

X	EP 3 156 932 A1 (DEUTSCHE TELEKOM AG [DE]) 19 April 2017 (2017-04-19)	1-7, 12-14
A	abstract paragraph [0029] - paragraph [0044]	8-11

	-/--	

[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
16 June 2021	24/06/2021

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	Mezödi, Stephan

1

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2020/078524

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	US 2005/165623 A1 (LANDI WILLIAM A [US] ET AL) 28 July 2005 (2005-07-28) abstract claim 1 paragraph [0021] - paragraph [0025] -----	1-7, 12-14 8-11

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2020/078524

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6760711 B1	06-07-2004	NONE	
US 2015169886 A1	18-06-2015	CN 104717199 A	17-06-2015
		US 2015169886 A1	18-06-2015
EP 3156932 A1	19-04-2017	NONE	
US 2005165623 A1	28-07-2005	NONE	