METHOD AND SYSTEM FOR QUERYING AND SEARCHING PROTECTED CLOUD DATA

Field of invention

This invention relates to a method and a system of managing data that contains private and public information stored on a cloud storage. More particularly, this invention relates to a method and a system of allowing a data owner to store data on a cloud storage in a secured manner while allowing a third party data service provider to perform queries and analytics on behalf of the data owner.

Background

Storing data on the cloud is convenient and useful as the data can be accessed anywhere and can be duplicated and easily backed up for added protection against loss and corruption. However, as the cloud storage is publicly accessible, it is important to ensure the privacy of the data. One common measure is to encrypt the data. However, encrypting the data causes difficulties in searching and analysing the data. Large amount of data contains patterns, trends and statistics that offer the data owner useful value when properly analysed. However, very often the data owner does not own the infrastructure or software services that perform analytics on the data.

Thus, those skilled in the art are constantly striving to provide a system that ensures the privacy of the data stored on cloud storage and at the same time, allows third party data service provider to perform queries and analytics on behalf of the data owner.

Summary of Invention

The above and other problems are solved and an advance in the art is provided by a method and a system of managing data that contains private and public information stored on a cloud storage in accordance with this invention. A first advantage of a system and a method in accordance with embodiments of this

invention is allowing a third party data service provider to perform queries and analytics on behalf of the data owner without compromising the privacy of data. A second advantage of a system and a method in accordance with embodiments of this invention is data owner need not own a large storage medium to store data. Hence, the data owner is able to free up facilities for housing storage medium for other uses. A third advantage of a system and a method in accordance with embodiments of this invention is that the data owned by the data owner can be accessed by third party who has the necessary resources to assist the data owner in analysing the data stored on the cloud storage.

In accordance with embodiments of this invention, a system and a method for storing and managing private and public data on a cloud storage is provided in the following manner. First, a raw data is received by the data owner server. Textual information from the raw data is extracted and subsequently categorised into public data and private data based on a policy file. A hash value data is then generated for the private data. The raw data is encrypted. Then, the method generates a package including the encrypted raw data, the public data and the hash value data. This package is transmitted to and stored on the cloud storage.

In accordance with embodiments of this invention, the step of categorising extracted textual information into public data and private data based on the policy file is provided in the following manner. The policy file stored on a memory of the processing system of the data owner server is retrieved and read. The extracted textual information is then compared with texts in the policy file. Extracted textual information is categorised in the private data if the extracted textual information matches any one of the texts in the policy file. The rest of the extracted textual information are categorised in the public data.

In accordance with embodiments of this invention, the step of extracting textual information is provided in the following manner. The raw data is read and portion of the raw data that contains textual information is determined. The extracted

textual information is then generated using the portion of the raw data being determined to be textual information.

In accordance with embodiments of this invention, the step of generating the hash value data for the private data may be provided in the following manner. A hash value for each of the texts in the private data is generated. A hash value data is then generated by appending the hash value for each of the texts in the private data to the hash value data.

In accordance with embodiments of this invention, the step of generating the hash value data for the private data may also be provided in the following manner. The value of k-consecutive text to be generated is determined. A hash value for each of the texts in the private data is generated. A subsequent hash value for each of k-consecutive text in the private data is generated. The hash value data is then generated by appending the hash value for each of the texts in the private data and the subsequent hash value for each of k-consecutive text in the private data to the hash value data, wherein k is less than the total number of text in the private data. Preferably, k is predetermined and can be retrieved from the policy file. Alternatively, k is equal to the total number of text in the private data.

In accordance with embodiments of this invention, the method further comprises steps being performed by the data owner server to generate query string in the following manner. A query text string received from a data consumer is read. The query text string is then categorised into the public data and the private data based on the policy file. A hash value data is then generated for the private data. A query string is subsequently generated based on the hash value data for the private data and the public data. The query string is then transmitted to the cloud storage.

In accordance with embodiments of this invention, the method further comprises steps being performed by the cloud storage to retrieve data in the following manner. Upon receiving the query string, the cloud storage determines the

type of the query string. If the query string is a request for encrypted raw data, cloud storage retrieves an encrypted raw data associated with the query string. The encrypted raw data is then transmitted to the data owner server. If the query string is a request for public data, cloud storage retrieves public data associated to the query string. The public data is then transmitted to the data owner server.

In accordance with embodiments of this invention, the step of retrieving public data associated with the query string may be provided in the following manner. First, the query string is read. The texts in the query string are matched with texts in package data stored in the cloud storage. Cloud storage then retrieves every package data that includes text that matches the text in the query string. Public data is extracted from each of the package data. The public data is then ranked according to number of texts that match the package data associated to the public data.

In accordance with embodiments of this invention, the system further includes a device for providing the raw data, wherein the processing unit, for executing the method is communicatively connected to the device to receive the raw data. Preferably, the device is a wearable medical device for tracking personal health status.

Brief description of drawings

The above and other features and advantages of a method and a system in accordance with this invention are described in the following detailed description and are shown in the following drawings:

Figure 1 illustrating systems communicatively connected to a network to provide or perform a system and/or method in accordance with an embodiment of this invention;

Figure 2 illustrating a representative processing system in one or more of the systems shown in figure 1 in accordance with embodiments of this invention;

Figure 3 illustrating a flow diagram of a process for processing and storing raw data in accordance with an embodiment of this invention;

Figure 4 illustrating a flow diagram of a process for generating hash value in accordance with an embodiment this invention;

Figure 5 illustrating a flow diagram of a process for generating a query string based on a query text string received from a data consumer in accordance with an embodiment of this invention;

Figure 6 illustrating a flow diagram of a process performed by the cloud storage to retrieve the required data in accordance with an embodiment of this invention;

Figure 7 illustrating an example of a raw data of a first patient;

Figure 8 illustrating another example of a raw data of a second patient;

Figure 9 illustrating another example of a raw data of a third patient; and

Figure 10 illustrating an exemplary example on a use of the system in accordance with this invention.

Detailed description

This invention relates to a method and a system of managing data that contains private and public information stored on a cloud storage. More particularly, this invention relates to a method and a system of allowing a data owner to store data on a cloud storage in a secured manner while allowing a third party data service provider to perform queries and analytics on behalf of the data owner.

It is envisioned that a system and/or method in accordance with embodiments of this invention may be used to allow a data owner to store data on a cloud storage in a secured manner while allowing a third party data service provider to perform queries and analytics on behalf of the data owner. In particular, data that are considered private are prevented from being made known to the third party data service provider when assessing the data from the cloud storage. This system and/or method enable a data owner to be a data aggregator for multiple data consumers and allow data consumers to perform useful operation on the data on behalf of the data owner without revealing any of the private data.

Figure 1 illustrates a system 100 that performs processes for providing the systems and methods in accordance with embodiments of this invention. The system 100 includes a data owner server 115 managed by a data owner 110, computer 125 accessible by a data consumer 120, and cloud storage server 130.

Data owner server 115 is a processing system that connects to cloud storage server 130 through wide area network 140 via a conventional wire connection, wireless connection or any other method. Data owner server 115 is being managed by data owner 110. Data owner server 115 manages data that are provided by participants such as clients, patients, or any users. Data that are provided by participants are transmitted to the data owner server 115. In turn, data owner server 115 executes application to process the data and transmits the processed data to cloud storage server 130. Data owner server 115 is also communicatively connectable to computer 125 through wide area network 140 via a conventional wire connection, wireless connection or any other method. Data owner server 115 executes application to process query received from data consumer 120 via computer 125 to retrieve certain data from the cloud storage server 130. Further details of processing of the data and query will be described in detail below.

Computer 125 is a processing system that connects to data owner server 115 and cloud storage server 130 through wide area network 140 via a conventional wire connection, wireless connection or any other method. Computer 125 is a processing system such as a desktop computer, laptop computer, or other computer terminals. Data consumer 120 uses computer 125 to perform queries and analytics on behalf of the data owner 110. Further details of search and query to retrieve required data from the cloud storage server 130 will be described in detail below.

Cloud storage server 130 is a processing system that provides resources to data owner server 115 and is connected to a wide area network 140 via a conventional wire connection, wireless connection or any other method. Cloud storage server 130 is connected to multiple storages 131-134 via a conventional wire

connection, wireless connection or any other method. Storages 131-134 are storage media for storing of digital content. One skilled in the art will recognize that other resources may include but are not limited to processing time, and applications.

Figure 2 illustrates exemplary processing system 200 that represents the processing systems in data owner server 115, computer 125 and cloud storage server 130 that execute instructions to perform the processes described below in accordance with this invention. One skilled in the art will recognize that the instructions may be stored and/or performed as hardware, firmware, or software without departing from this invention. One skilled in the art will recognize that the exact configuration of each processing system may be different and the exact configuration executing processes in accordance with this invention may vary and processing system 200 shown in figure 2 is provided by way of example only.

Processing system 200 includes Central Processing Unit (CPU) 205. CPU 205 is a processor, microprocessor, or any combination of processors and microprocessors that execute instructions to perform the processes in accordance with the present invention. CPU 205 connects to memory bus 210 and Input/Output (I/O) bus 215. Memory bus 210 connects CPU 205 to memories 220 and 225 to transmit data and instructions between the memories and CPU 205. I/O bus 215 connects CPU 205 to peripheral devices to transmit and receive data between CPU 205 and the peripheral devices. One skilled in the art will recognize that I/O bus 215 and memory bus 210 may be combined into one bus or subdivided into many other busses and the exact configuration is left to those skilled in the art.

A non-volatile memory 220, such as a Read Only Memory (ROM), is connected to memory bus 210. Non-volatile memory 220 stores instructions and data needed to operate various sub-systems of processing system 200 and to boot the system at start-up. One skilled in the art will recognize that any number of types of memory may be used to perform this function.

A volatile memory 225, such as Random Access Memory (RAM), is also connected to memory bus 210. Volatile memory 225 stores the instructions and data needed by CPU 205 to perform software instructions for processes such as the processes for providing a system in accordance with this invention. One skilled in the art will recognize that any number of types of memory may be used to provide volatile memory and the exact type used is left as a design choice to those skilled in the art.

I/O device 230, keyboard 235, display 240, memory 245, network device 250 and any number of other peripheral devices connect to I/O bus 215 to exchange data with CPU 205 for use in applications being executed by CPU 205. I/O device 230 is any device that transmits and/or receives data from CPU 205. Keyboard 235 is a specific type of I/O that receives user input and transmits the input to CPU 205. Display 240 receives display data from CPU 205 and display images on a screen for a user to see. Memory 245 is a device that transmits and receives data to and from CPU 205 for storing data to a media. Alternatively, in the case of the cloud storage server 130, memory 245 may be multiple storages such as storages 131-134 that transmit and receive data to and from CPU 205 for storing data. Network device 250 connects CPU 205 to a network for transmission of data to and from other processing systems.

A system and method for managing the data in a secured manner in accordance with an embodiment of the invention is described. Figures 3 and 4 illustrate how the data are processed on the data owner server 110 and stored on cloud storage 130. Figure 5 illustrates a process of converting a query string for retrieving data stored on the cloud storage. Figure 6 illustrates a process executed on the cloud storage for retrieving data. The remaining figures illustrate raw data and the corresponding processed package data in accordance with embodiments of the invention.

Figure 3 illustrates a flow diagram of process 300 performed by a processing system in data owner server 115 for storing of data on cloud storage 130. As will be

described below with reference to figure 10, process 300 may also be performed by a processing system in devices 116-118 for storing of data on cloud storage server 130. The process is provided by processes stored in software, firmware, or hardware. Process 300 begins in step 305 by receiving raw data. Raw data may be provided by various participants such as patient via a processing system such as a desktop computer, laptop computer, or other computer terminals that connect to data owner server 115 via a conventional wire connection, wireless connection or any other method. In the instance of storing data on cloud storage server 130 from the devices 116-118, raw data from the devices 116-118 is processed by a processing system that is provided in the device 116-118 or a processing system that is communicatively connected to the devices 116-118.

After receiving the raw data, process 300 extracts textual information in step 310. It is assume that raw data may contain image files and textual information. For purposed of this discussion, textual information is any information that is presented using words, characters, alphanumeric, etc. As only textual information is searchable, only textual information is extracted from the raw data for further processing. The textual information can be extracted in the following manner. The raw data is read to determine portion or portions which contain textual information. A textual information data is then generated in response to portion that is determined to be textual information.

In step 315, the extracted textual information is categorised into two categories, namely, public data and private data. The categorising of the extracted textual information is based on a policy file stored on the memory of processing system of data owner server 115 of devices 116-118. The policy file contains information for categorising data which may include words, phrases, numerals or other information that is considered private and/or sensitive that has to be remained private and out of reach to anyone other than the data owner who has authority to access the raw data via data owner server 110. The extracted textual information may be categorised in the following manner. First, the policy file is retrieved from the

memory of the processing system. The information in the policy file is then compared with each of the textual information. If any of the textual information matches the information in the policy file, the matched textual information will be categorised in the private data. Otherwise, the textual information will be categorised in the public data. For example, if the extracted textual information is {**A**n**B**n**C**n**D**n**E**} and the policy file includes "**A**", "**B**", "**C**" and "**D**" which are considered as private information, then the extracted textual information will be categorized into {**A B C D**} as private data and {n n n n **E**} as public data.

In step 320, process 300 generates hash value data for k-gram (shingles) for private textual data (i.e. k-shingles) using secret key. Further to the example above where the private data is {**A B C D**}, process 300 generates hash value for each private text as follows, $\{H_A\ H_B\ H_C\ H_D\}$. In order to preserve the relationship between adjacent words, process 300 also generates hash value for adjacent words as follows $\{H_{AB}\ H_{BC}\ H_{CD}\}$. Depending on the requirement, process 300 may further generate hash value for 3 words to preserve the relationship between 3 joining words as follows $\{H_{ABC}\ H_{BCD}\}$. Further details of step 320 will be described with reference to figure 4 below.

In step 325, the raw data is being encrypted with a secret key stored on the memory of processing system of data owner server 110. The secret key may be stored together with the policy file. The secret key is any key that is used for encrypting and decrypting raw data. A secret key may be a random sequence of bits and the number of bits used depends on the desired strength of the encryption. Essentially, the more bits used, the better the security. Some possible number of bits are 128, 192, 256, 512 and 1024. Nevertheless, the number of bits used is left to one skilled in the art.

In step 330, process 300 packages the encrypted raw data, hash values and clear public data. The package data is then transmitted to and stored on the cloud storage in step 335. Process 300 ends after step 335.

Figure 4 illustrates a flow diagram of process 400 performed by a processing system in data owner server 115 or devices 116-118 for processing step 320 in process 300. The process is provided by processes stored in software, firmware, or hardware. Process 400 begins in step 405 by determining number of texts, max(k). Max(k) is the desired number of consecutive texts that will have its order relationship preserved. Further to the example above where the private data is {**A B C D**}, a max(k) value of 2 ensures that we will be able to match 2 adjacent text correctly. The highest possible value of max(k) in this example is 4 since the private data contains 4 texts, i.e. "A", "B", "C" and "D". Max(k) may be determined based on the number of texts in the private data. Alternatively, max(k) may be predetermined by a user by storing the max(k) value in the policy file.

In step 410, process 400 generates a hash value for each of the k-consecutive text using a secret key. This secret key may be the same or different key used for encrypting the raw data in step 325 in process 300. The choice of the secret key is left to one skilled in the art. Further to the example above where the private data is {**A B C D**}, the hash value generated via step 410 is $\{H_A H_B H_C H_D\}$ for k=1.

In step 415, process 400 determines if there is any duplication. The use of hash value is such that a different hash value will be generated for each different text. Hence, no same text will have same hash value. If there is any duplication, process 400 proceeds to step 420 to discard the duplicate hash value and subsequently proceeds to step 425 to save the remaining hash value in memory of processing system of data owner server 115. Otherwise, process 400 proceeds directly to step 425 and save the hash value in memory of processing system of data owner server 115 or devices 116-118.

In step 430, a counter increment k. If k is more than max(k), this means that we have generated enough hash values to preserve relationship for k-consecutive

texts and process 400 ends. If k is less than max(k), then process 400 repeats from step 410 and generate a hash value for each of the k-consecutive text using the secret key. Essentially, the number of times step 410 is repeated depends on the value of max(k). Further to the example above where the private data is {**A B C D**}, the hash value generated via step 410 is $\{H_{AB}\ H_{BC}\ H_{CD}\}$ for k=2, $\{H_{ABC}\ H_{BCD}\}$ for k=3, and $\{H_{ABCD}\}$ for k=4.

Further to the example above where the private data is {**A B C D**}, the hash value data is $\{H_A\ H_B\ H_C\ H_D\ H_{AB}\ H_{BC}\ H_{CD}\ H_{ABC}\ H_{BCD}\}$, if max(k)=3 assuming the requirement is to preserve the relationship of at least 3 consecutive words. If the requirement is to preserve the relationship of two adjacent words only, then the hash value data will be $\{H_A\ H_B\ H_C\ H_D\ H_{AB}\ H_{BC}\ H_{CD}\}$. If the requirement is to preserve the relationship of four adjacent words, then the hash value data will be $\{H_A\ H_B\ H_C\ H_D\ H_{AB}\ H_{BC}\ H_{CD}\ H_{ABC}\ H_{BCD}\ H_{ABCD}\}$. Process 400 ends after step 435.

Figure 5 illustrates a flow diagram of process 500 performed by a processing system in data owner server 115 upon receipt of a query from a data consumer. The process is provided by processes stored in software, firmware, or hardware. Process 500 begins in step 505 by reading the query text string received from data consumer.

In step 510, query text string is categorised into two categories, namely, public data and private data. The categorising of the query text is based on the policy file stored on the memory of processing system of data owner server 115. This is similar to step 315 in process 300.

In step 512, process 600 determines if the query text string contains any private data. If there is no private data in the query text string, process 500 proceeds to step 513 to create a query string request for public data based on the public data to the cloud storage server 130. If the query text string contains private data, process 500 proceeds to step 515.

In step 515, process 500 generates hash value data for k-gram (shingles) for private textual data (i.e. k-shingles) using secret key. This is similar to step 320 in process 300 and further described with reference to figure 4.

In step 520, process 500 creates a query string to request for public data based on the hash value data and public data and transmits the query string to the cloud storage. Process 500 ends after step 520.

Figure 6 illustrates a flow diagram of process 600 performed by a processing system in cloud storage server 130 upon receipt of a query string from data owner. The process is provided by processes stored in software, firmware, or hardware. Process 600 begins in step 605 by reading the query string received from data owner.

In step 608, process 600 determines whether the query string is a request for public data. If the query string is a request for public data, process 600 proceeds to step 610. Otherwise, process 600 proceeds to step 630.

In step 630, process 600 retrieve encrypted data based on the information in the query string. In step 635, the retrieved encrypted data is transmitted to the data owner server 115.

In step 610, process 600 performs a matching of text in query string with text in all the package data stored on memory in processing system in cloud storage. Process 600 retrieves all the package data that matches the text in query string in step 615. Process 600 then extract public data from each of the retrieved package data in step 620. Each of the public data is then ranked according to the number of matches in step 622. For example, if the query string includes two texts, then the retrieved package data that contains both texts will have higher ranking than one that only contains one text. The ranked public data is then transmitted to the data owner server in step 625. Alternatively, the cloud storage server 130 may also transmit the

ranked public data directly to the data consumer without departing from the invention. One skilled in the art will recognise that step 622 for ranking data may be performed by the cloud storage server 130 if the cloud storage server 130 has the capability to allow such ranking of data. If the cloud storage server 130 does not allow such ranking of data, all data may be transmitted to the data owner server 115 after step 620 and ranking of the data may be performed by the data owner server 115 instead without departing from the invention.

An example on the use of the system and/or method will now be described as follows with reference to figures 7, 8, 9 and 10. This example illustrates the system and/or method being implemented in a hospital where some of patients' information is considered private and should not be made available to anyone other than one who has authority to access patients' information such as doctors.

Figure 10 illustrates an exemplary system 1000 that performs processes for providing the systems and methods in accordance with embodiments of this invention being implemented in a hospital. System 1000 includes cloud storage server 130, storages 131-134, data owner 110, devices 116-118, data consumer 120.

Data owner 110 in this exemplary system is the hospital administrator. Cloud storage server 130 may be any third party cloud storage service providers such as Amazon, IBM, Google, Hewlett-Packard, etc. Data consumer 120 is any third party data service provider to perform queries and analytics on behalf of the data owner.

Devices 116-118 are devices such as weighing machine, blood test kit, blood pressure device, or any wearable medical devices for tracking personal health status. Devices 116-118 are connectable to cloud storage server 130 through wide area network 140 via a conventional wire connection, wireless connection or any other method. One common method of connecting to cloud storage server 130 is via connection with a mobile device that has connectivity to the wide area network 140. Devices 116-118 each has a processing system 200 as described above to execute

processes required to process data and transmit data to the cloud storage server 130. The processing system may be provided in the devices 116-118 or as an attachment that is communicatively connected to the device 116-118. Policy file, encryption file and processes required to process data and transmit data to the cloud storage server 130 are stored in memory of each of devices 116-118 by an administrator of the hospital. This ensures that the policy file required to encrypt the data or hash private data is not made available to the public.

Arrows 150, 155, 160, 170 and 180 shows the flow of data being transmitted or received from various components of system 1000. Devices 116-118 only transmit packaged data to cloud storage server 130 as shown by one way arrow 180. Data owner 110 using data owner server 115 transmit and receive data from cloud storage server 130 as shown by two way arrow 160. Data consumer 120 via computer 125 transmits a search query request to data owner as shown by one way arrow 155. Depending on the processes, data retrieved from storages 131-134 may be transmitted from cloud storage server 130 to data consumer via data owner through arrows 160 and 150, or directly via one way arrow 170.

Figure 7 shows a first raw data 710 of a first patient “Tan Ah Ming”, figure 8 shows a second raw data 810 of a second patient “Tan Xiao Ming” and figure 9 shows a third raw data 910 of a third patient “Toh Ah Ming”. Raw data 710, 810 and 910 may be obtained from the device 116-118 or directly from data owner 110. Assuming the policy file contains “Name”, “NRIC”, and “Status”, the first, second and third raw data 710, 810 and 910 will be packaged according to process 300 as follows.

For the first raw data 710, the private data is {Tan Ah Ming S1234567A Healthy} while the rest of the data is public data 740. The process then generates hash value data for the private data. In this example, assuming only two adjacent words are required to be preserved, then max(k) value is 2 and the packed hash value data is {H_{Tan} H_{Ah} H_{Ming} $H_{S1234567A}$ $H_{Healthy}$ H_{TanAh} H_{AhMing} $H_{MingS1234567A}$

$H_{S1234567AHealthy}$}. Hence, the packaged data which include encrypted data 720, hash value data 730 and public data 740 will be transmitted to cloud storage server 130 and stored on the storages 131-134.

For the second raw data 810, the private data is {Tan Xiao Ming S7654321A Healthy} while the rest of the data is public data 840. The process then generates hash value data for the private data. For a max(k) value of 2, the packed hash value data is {H_{Tan} H_{Xiao} H_{Ming} $H_{S7654321A}$ $H_{Healthy}$ $H_{TanXiao}$ $H_{XiaoMing}$ $H_{MingS7654321A}$ $H_{S7654321AHealthy}$}. Hence, the packaged data which include encrypted data 820, hash value data 830 and public data 840 will be transmitted to cloud storage server 130 and stored on the storages 131-134.

For the third raw data 910, the private data is {Toh Ah Ming S1234568A Unknown} while the rest of the data is public data 940. The process then generates hash value for the private data. For a max(k) value of 2, the packed hash value data is {H_{Toh} H_{Ah} H_{Ming} $H_{S1234568A}$ $H_{Unknown}$ H_{TohAh} H_{AhMing} $H_{MingS1234568A}$ $H_{S1234568AUnknown}$}. Hence, the packaged data which include encrypted data 920, hash value data 930 and public data 940 will be transmitted to cloud storage server 130 and stored on the storages 131-134.

If a doctor or the administrator of the hospital wishes to access the records of his patient, “Tan Xiao Ming”, via the data owner server, he or she can simply retrieve the encrypted data 820 from the cloud storage server 130 as data owner server has the required secret key to decrypt the encrypted data 820. The choice of the decryption and encryption algorithm used is left to one skilled in the art. In particular, he or she can transmit a query string with the identification number “S1234567A” to the cloud storage server 130. Since the query string is not for a public data, process 600 will be routed to step 630 to retrieve the encrypted data associated to the identification number, NRIC, “S1234567A” and subsequently transmit the retrieved encrypted data to the data owner server 115.

If a data consumer 120 wishes to perform a query, he or she transmits a request with a query text string to the data owner server 115. The data owner server 115 then executes process 500 to create a query string and transmit to cloud storage server 130 to retrieve the required data. Assuming the query text string is "Status: Healthy", the data owner server 115 will execute process 500 to generate the following query string $\{H_{Healthy}\}$ to request for public data. At the cloud storage server 130, upon receipt of the query string, process 600 will retrieve public data 740 and public data 840 since both patients, "Tan Ah Ming" and "Tan Xiao Ming" are "Healthy".

If a data consumer wishes to perform a query on the health status of all the patients with the surname "Tan", he or she transmits a request with a query text string, "Name: Tan" to the data owner server 115. The data owner server 115 then executes process 500 to create the following query string $\{H_{Tan}\}$ to request for public data. At the storage cloud, upon receipt of the query string, process 600 will retrieve public data 740 and public data 840 since both patients', "Tan Ah Ming" and "Tan Xiao Ming", surname is "Tan".

If a data consumer wishes to perform a query on the health status of all the patients with blood type "O+", he or she transmits a request with a query text string, "Blood type: O+" to the data owner server 115. The data owner server 115 then executes process 500 to create the following query string {O+} to request for public data. At the cloud storage server 130, upon receipt of the query string, process 600 will retrieve public data 740 and public data 940 since both patients; "Tan Ah Ming" and "Toh Ah Ming", blood type is "O+".

If a data consumer wishes to perform a query on patients who bear the name "Tan Ah Ming", he or she transmits a request with a query text string, "Name: Tan Ah Ming" to the data owner server 115. The data owner server 115 then executes process 500 to create the following query string $\{H_{Tan}\ H_{Ah}\ H_{Ming}\ H_{TanAh}\ H_{AhMing}\}$, since

max(k) is 2. At the cloud storage server 130, upon receipt of the query string, process 600 will retrieve the following public data:

1. Public data 740 as there is a match of $\{H_{Tan}\ H_{Ah}\ H_{Ming}\ H_{TanAh}\ H_{AhMing}\}$;
2. Public data 940 as there is a match of $\{H_{Ah}\ H_{Ming}\ H_{AhMing}\}$; and
3. Public data 840 as there is a match of $\{H_{Tan}\ H_{Ming}\}$; and

As will be noted from the above, the packaged data of public data 740 has 5 matches, packaged data of public data 940 has 3 matches and packaged data of public data 840 has 2 matches. Hence, public data 740 is ranked as first, public data 940 is ranked second and public data 840 is ranked third. The ranked public data 740, 940 and 840 are then transmitted to the data consumer 120 via the data owner 110. Alternatively, the ranked public data 740, 940 and 840 may be transmitted directly to the data consumer 120.

The above is a description of embodiments of a system in accordance with the invention as set forth below. It is envisioned that those skilled in the art can and will design alternative embodiments of this invention based upon this invention that infringe on this invention as set forth in the following claims.

Claims

1. A system for storing private and public data on a cloud storage comprising:

a processing system unit having a processor and a memory communicatively connected to said processor, said memory having instructions executed by said processor to:

receiving a raw data;

extracting textual information;

categorising extracted textual information into public data and private data based on a policy file;

generating a hash value data for said private data, said generating said hash value data for said private data comprises:

determining number of k-consecutive text to be generated;

generating a hash value for each of a plurality of texts in said private data;

generating a subsequent hash value for each of k-consecutive text in said private data;

generating said hash value data by appending said hash value for each of said plurality of texts in said private data and said subsequent hash value for each of k-consecutive text in said private data to said hash value data, wherein k is less than the total number of text in said private data;

encrypting said raw data;

generating a package data including said encrypted raw data, said public data and said hash value data; and

storing said package data to said cloud storage.

2. The system according to claim 1, wherein said instructions to categorising extracted textual information into public data and private data based on said policy file comprises instructions to:

retrieving said policy file from said memory;

comparing said extracted textual information with a plurality of texts in said policy file; and

categorising said extracted textual information in said private data in response to said extracted textual information matches one of said plurality of texts in said policy file.

3. The system according to claim 2, wherein said instructions to categorising extracted textual information into public data and private data based on said policy file comprises instructions to:

categorising said extracted textual information in said public data in response to said extracted textual information that does not match any one of said plurality of texts in said policy file.

4. The system according to claim 1, wherein said instructions to extracting textual information comprises instructions to:

reading said raw data;

determining portion in said raw data that contains textual information;

generating said extracted textual information with said portion of said raw data being determined to be textual information.

5. The system according to claim 1, wherein k is equal to said total number of text in said private data.

6. The system according to claim 1, wherein k is predetermined and can be retrieved from said policy file.

7. The system according to claim 1, wherein said memory further comprises instructions to:

reading a query text string received;

categorising said query text string into said public data and said private data based on said policy file;

generating a hash value data for said private data;

generating a query string based on said hash value data for said private data and said public data;

transmitting said query string to said cloud storage.

8. The system according to claim 7, wherein said cloud storage comprises:

a processor and a memory communicatively connected to said processor, said memory having instructions executed by said processor to:

determining the type of query string;

retrieving an encrypted raw data associated with said query string in response to determining said type of query string is a request for encrypted raw data; and

transmitting said encrypted raw data to said processing system.

9. The system according to claim 7, wherein said cloud storage comprises:

a processor and a memory communicatively connected to said processor, said memory having instructions executed by said processor to:

determining the type of query string;

retrieving public data associated to said query string in response to determining said type of query string is a request for public data; and

transmitting said public data to said processing system.

10. The system according to claim 9, wherein said instructions to retrieving public data associated with said query string comprises instructions to:

reading said query string;

performing matching of text in said query string with text in package data stored in said memory in said cloud storage; and

retrieving every package data that includes text that matches said text in said query string;

extracting public data from each of said package data; and

ranking each of said public data according to number of texts that match said package data associated to said public data.

11. The system according to claim 1 further comprising:

a device for providing said raw data, wherein said processing unit is communicatively connected to said device to receive said raw data.

12. The system according to claim 11 wherein said device is a wearable medical device for tracking personal health status.

13. A method for storing private and public data on a cloud storage comprising:

receiving a raw data;

extracting textual information;

categorising extracted textual information into public data and private data based on a policy file;

generating a hash value data for said private data, said generating said hash value data for said private data comprises:

determining number of k-consecutive text to be generated;

generating a hash value for each of a plurality of texts in said private data;

generating a subsequent hash value for each of k-consecutive text in said private data;

generating said hash value data by appending said hash value for each of said plurality of texts in said private data and said subsequent hash value for each of k-consecutive text in said private data to said hash value data, wherein k is less than the total number of text in said private data;

encrypting said raw data;

generating a package including said encrypted raw data, said public data and said hash value data; and

storing said package to said cloud storage.

14. The method according to claim 13, wherein said step of categorising extracted textual information into public data and private data based on said policy file comprises:

retrieving said policy file from said memory;

comparing said extracted textual information with a plurality of texts in said policy file; and

categorising said extracted textual information in said private data in response to said extracted textual information matches one of said plurality of texts in said policy file.

15. The method according to claim 14, wherein said step of categorising extracted textual information into public data and private data based on said policy file comprises:

categorising said extracted textual information in said public data in response to said extracted textual information that does not match any one of said plurality of texts in said policy file.

16. The method according to claim 13, wherein said step of extracting textual information comprises:

reading said raw data;

determining portion in said raw data that contains textual information;

generating said extracted textual information with said portion of said raw data being determined to be textual information.

17. The method according to claim 13, wherein k is equal to said total number of text in said private data.

18. The method according to claim 13, wherein k is predetermined and can be retrieved from said policy file.

19. The method according to claim 13, further comprising:

reading a query text string received;

categorising said query text string into said public data and said private data based on said policy file;

generating a hash value data for said private data;

generating a query string based on said hash value data for said private data and said public data;

transmitting said query string to said cloud storage.

20. The method according to claim 19, wherein said cloud storage upon receiving said query string performs the following steps:

determining the type of query string;

retrieving an encrypted raw data associated with said query string in response to determining said type of query string is a request for encrypted raw data; and

transmitting said encrypted raw data to said processing system.

21. The method according to claim 19, wherein said cloud storage upon receiving said query string performs the following steps:

determining the type of query string;

retrieving public data associated to said query string in response to determining said type of query string is a request for public data; and

transmitting said public data to said processing system.

22. The method according to claim 21, wherein said step of retrieving public data associated with said query string comprises:

reading said query string;

performing matching of text in said query string with text in package data stored in said memory in said cloud storage; and

retrieving every package data that includes text that matches said text in said query string;

extracting public data from each of said package data; and

ranking each of said public data according to number of texts that match said package data associated to said public data.

5