FORM 2

The Patents Act 1970
(39 of 1970)
&
The Patent Rules 2003

PROVISIONAL SPECIFICATION

(See Section 10 and rule 13)

TITLE

A SYSTEM AND METHOD FOR DISTRIBUTED AND ENCRYPTED DATA STORAGE IN MULTIPLE CLOUD SYSTEMS

APPLICANT

RUNCY OOMMEN

#402, Coral Waters Apartment, Kodichikanahalli, Bannerghatta Road, Bangalore - 560076, Karnataka, India

PREAMBLE OF THE DESCRIPTION:

THE FOLLOWING SPECIFICATION PARTICULARLY DESCRIBES THE INVENTION

A SYSTEM AND METHOD FOR DISTRIBUTED AND ENCRYPTED DATA STORAGE IN MULTIPLE CLOUD SYSTEMS

TECHNICAL FIELD

[0001] The present disclosure relates to a data storage. Particularly, the present disclosure relates to a computing data, stored in one or more cloud platforms. More particularly, the present disclosure relates to a distributing and storing encrypted data in multiple cloud platforms.

BACKGROUND

[0002] Cloud storage is usually defined as “the storage of data online, in a cloud”, where a user’s data is stored in and accessible from multiple distributed, heterogeneous, and connected resources that comprise the cloud. The operations performed on the data stored in the cloud and managing the cloud itself is referred as cloud computing. Cloud computing has revolutionized the landscape of information technology (IT) infrastructure. It is based on the premise that both hardware and software resources previously maintained by the user’s own data center or local network can be made available through a network of cloud servers hosted on the Internet by third parties. Cloud computing therefore efficiently alleviates the need for the users to own and manage their own elaborate infrastructures and data centers.

[0003] However, on the individual user’s side, the scenario is quite different. When a user uploads a file to one of the cloud storage providers, the file/data is stored in a single entity. This may be vulnerable to attacks from unknown third-parties. Further, this allows the user to only use a specified amount of storage space allocated and has to pay for the extra storage when the allocated amount of storage is filled. Further, the conventional cloud storage systems do not have the ability to connect various cloud storage systems and increase the storage capacity.

[0004] This disconnected approach is cumbersome and restricts end users to efficiently create, locate, and configure data stored in multiple cloud storage systems. Further, conventional systems fail to distribute the content efficiently.

[0005] In order to overcome the drawbacks associated with the existing industrial automation translation tools, there was felt a need for a holistic, efficient and secure platform to store the user's data in a plurality of cloud storage systems. Further, there was felt a need for a platform for distributing and encrypting the data stored in the plurality of cloud storage systems.

OBJECTS

[0006] An object of the present disclosure is to provide a system and method for providing unified data storage platform for storing data efficiently across multiple cloud storage systems.

[0007] One more object of the present disclosure is to connect multiple cloud storage systems of a single user.

[0008] Another object of the present disclosure is to provide a system and method to split the user upload data, encrypt the uploaded data and distribute the same across multiple cloud storage systems.

[0009] Another object of the present disclosure is to decrypt the uploaded data, join the content which is split across multiple cloud storage systems when the user needs to retrieve the uploaded data.

[0010] Another object of the present disclosure is to provide a seamless and uniform interface to share, access and collaborate on data irrespective of the underlying cloud storage provider.

SUMMARY

[0011] The present disclosure envisages a system for distributing and storing data in multiple cloud storage systems. The application server is referred as a unified platform for distributing and storing data in multiple cloud storage systems. In accordance with the present disclosure, the term cloud storage system refers to storage providers who allow a user to store the data in a remote device and is accessible on any compatible device through a communication network.

[0012] Typically, the user uploads data to a unified platform by linking a plurality of cloud storage accounts. The application server receives the data uploaded by the user, splits the data into a plurality of chunks, encrypts the data and distributes the encrypted data in a plurality of linked cloud storage accounts. Further, when the user has to retrieve the uploaded file, the unified cloud platform decrypts the data from the multiple cloud storage accounts, joins the data and displays the same to the user. The application server extracts metadata from the plurality of cloud storage accounts for the splitting and distributing the data in multiple cloud storage accounts. In accordance with the present disclosure, the metadata includes the cloud storage status metadata which is used by an algorithm to factor in the split file size before distributing to multiple cloud servers. The examples of the metadata include, but are not limited to total space, used space, and maximum file size, and the like.

BRIEF DESCRIPTION OF ACCOMPANYING DRAWINGS

[0013] These and the other features, aspects, and advantages envisaged by the present disclosure will become apparent when the following detailed description is analyzed in conjunction with the accompanying drawings in which like characters represent like parts throughout, and wherein:

[0014] FIG.1 is a block diagram illustrating the environment in which the data is stored in multiple cloud storage systems, in accordance with the present disclosure;

[0015] FIG. 2 is a block diagram illustrating a file split process, in accordance with the present disclosure;

[0016] FIG. 3 is a system diagram illustrating the encryption of data stored in multiple cloud storage systems, in accordance with the present disclosure;

[0017] FIG. 4 is a system diagram illustrating the decryption of the data retrieved data from the multiple cloud storage systems, in accordance with the present disclosure;

[0018] FIG. 5 is a block diagram illustrating file joining process, in accordance with the present disclosure;

[0019] FIG. 6 is a flowchart illustrating the process of uploading a file to multiple cloud storage devices using a unified platform, in accordance with the present disclosure; and

[0020] FIG. 7 is a flowchart illustrating the process of retrieving the file from multiple cloud storage devices using a unified platform, in accordance with the present disclosure.

DETAILED DESCRIPTION

[0021] The present disclosure envisages a system that provides an application server for storing data in multiple cloud storage systems. The application server is referred as a unified platform for distributing and storing data in multiple cloud storage systems. In accordance with the present disclosure, the term cloud storage system refers to storage providers who allow a user to store the data in a remote device and is accessible on any compatible device through a communication network.

[0022] Typically, the user uploads data to a unified platform by linking a plurality of cloud storage accounts. The application server receives the data uploaded by the user, splits the data into a plurality of chunks, encrypts the data and distributes the encrypted data in a plurality of linked cloud storage accounts. Further, when the user has to retrieve the uploaded file, the unified cloud platform decrypts the data from the multiple cloud storage accounts, joins the data and displays the same to the user. The application server extracts metadata

from the plurality of cloud storage accounts for the splitting and distributing the data in multiple cloud storage accounts.

[0023] FIG.1 is a block diagram illustrating the environment in which the data is stored in multiple cloud storage systems, in accordance with the present disclosure. The block diagram includes an application server 102, a list of linked cloud storage accounts 104, and a database 120. Typically, the application server 102 includes a user interface 106, a processor 108, and memory module 110, along with a plurality of other components.

[0024] In accordance with the present disclosure, the user interface 106 forms a front-end system of the application server and the other components form the backend system for distributing and sharing the files in the plurality of cloud storage system. The user interface 106 receives inputs from the user and displays the output to the user. The inputs received from the user include but are not limited to registration of the user by providing required login credentials, details of a plurality of cloud storage accounts, contact email address, and the like. The user interface 106 is presented on a client device such as a laptop, desktop, smartphone, and the like. The details received from the user is stored in the database 120. The database 120 is communicatively coupled with the application server 102. The processor 108 is a hardware circuitry that responds and processes instructions for enabling the distribution and storage of data in multiple cloud storage systems. The examples of the cloud storage systems include, but are not limited to Google Drive™, Dropbox™, OneDrive™, Box™, Hive™, iDrive™, Amazon S3™, iCloud™, and the like.

[0025] Further, the memory 110 is a hardware device used for storing instructions for enabling the processor 108 to process the instructions. The memory 110 includes both temporary memory and permanent memory. In accordance with the present disclosure, the application server 102 includes a plurality of other modules such as an auxiliary module, feedback module, a machine learning module, an artificial intelligence module, a query module, an encryption module, a decryption module and the like for facilitating the efficient data distribution and storage in multiple cloud storage systems. Further, the database 120 is a repository to store the details of the user, a list of linked cloud storage accounts related to the user, application server-related data, metadata and the like. The database 120 also stores a

plurality of keys and checksum hash values that are used for encrypting, decrypting, and validating the data uploaded to the application server 102.

[0026] Referring to FIG. 1, the user uploads the data through the user interface 106. The uploaded data is split into multiple chunks and are encrypted using an encryption algorithm and are further stored in the cloud storage-A 112, the cloud storage-B 114, the cloud storage-C 116, and the like. For example, when the user uploads the data which is the form of a media file such as video and audio through the user interface 106, the application server 102 computes the size of the document and also computes the storage space available by linking all the cloud storage accounts. Further, the application server 102 computes the number of linked cloud storage accounts in which the data is split and distributed after encryption. When the user has to retrieve that textual document, the data is decrypted from multiple cloud storage accounts and are joined to provide as a single document.

[0027] FIG. 2 is a block diagram illustrating a file split process, in accordance with the present disclosure. The block diagram includes a client device 202, the application server 102, the database 120, the plurality of cloud accounts 204, and a plurality of encrypted files 206. The client device 202 is a device through which the user accesses the application server 102 for distributing the files in multiple cloud storage systems. The examples of the client device include, but are not limited to the laptop, desktop, smartphone, and the like.

[0028] The application server 102 receives one or more files from the client device 202 and also accesses the data related to the user, which includes the plurality of linked cloud storage accounts, user data, metadata, and the like. The data related to the user and the accounts are retrieved from the database 120. Further, once the data is related to the user and the linked accounts are retrieved, the application server 102 selects a plurality of accounts 204 to split the user uploaded the file. For example, when there are six accounts linked to the user, the application server 102 may select all the six accounts for storing the split files or may select four accounts for storing the split files. In accordance with the present disclosure, the number of accounts to be selected is determined by the application server 102 dynamically. In addition, the application server 102 computes the used space, free space and the available space for each of the user. In accordance with the present disclosure, when a user has linked

four cloud storage accounts, the application server 102 computes the total space available by combining the cloud storage accounts, free space available in each of the cloud storage, consumable space available for the application server 102 and the like for storing the split encrypted files. Once the number of accounts is selected and the available space is computed, the application server 102 generates a plurality of split files 204 and subsequently split encrypted filed 206. Further, the client device 202 also stores a hash value to the database 120 for future references. Furthermore, the respective hash values of the split files 204 is also stored in the database for future references.

[0029] FIG. 3 is a system diagram illustrating the encryption of data stored in multiple cloud storage systems, in accordance with the present disclosure. Typically, the encryption is done by the application server 102 using an encryption module (not shown in the figure). In accordance with the present disclosure, the encryption is carried out after the data into multiple chunks. Initially, a key-1A module 302 which is stored in the application server specific to one of the linked cloud storage accounts is XOR'd (exclusive OR operation) with a dynamic time stamp module 304 that provides the time stamp of the system. In accordance with the present disclosure, the dynamic time stamp module 304 derives an immediate prime number of an EPOCH module. The result of the XOR operation between the Key-1A module 302 and the dynamic time stamp module 304 results in an intermediate key A which is subsequently stored in the intermediate key module-A 308. Further, another XOR operation is performed with a key-2A module 306 which stores a key that is program specific to the linking cloud storage account. The result of the XOR operation between the intermediate key A and the key 2A results in a final key-A which is stored in the final key module 310. The final key-A module generates a split file A which is stored in the split file-A module 204a. Further, the split file-A generates a split encrypted file-A which is stored in the split encrypted file-A module 206a. The split encrypted files are stored in the linked cloud storage accounts. For example, the split encrypted file-A is stored in the cloud storage account-A 112. The split files are encrypted using a plurality of standard encryption techniques. One such example of the encryption performed on the split file is GNU Privacy Guard (GNUPG) encryption. The encryption is mostly a conventional symmetrical key cryptography, typically by using the final key to encrypt a session key which is only used once. In accordance with

the present disclosure, the further to the encryption, the splits are named uniquely and are stored in the respective cloud storage accounts. Further, the generated hash value for each of the split encrypted files is stored in the database 120. Similarly, for encryption is performed for each of the splits computed by the application server 102.

[0030] Referring to FIG. 4, there is shown a system diagram illustrating the decryption of the data retrieved from the multiple cloud storage systems. In accordance with the present disclosure, the decryption process is processed by a decryption module (not shown in the figure) and is processed when the user needs to retrieve the data uploaded on the application server. Typically, the decryption involves individual decryption of the encrypted file splits. In accordance with the present disclosure, the split encrypted file chunks is retrieved from the plurality of linked cloud storage accounts. The hash values are checked by the application server before generating the split files. An error message is communicated to the application server when the hash values does not match.

[0031] In an example, initially, when the data has to be retrieved from the split encrypted file chunks 206a, a key-1A module 302 containing a key that is specific to the cloud storage account-A 112 is XOR'd with the dynamic timestamp module 304. In accordance with the present disclosure, the dynamic timestamp is derived from an immediate prime number of the EPOCH dynamically. The first XOR operation between the values of key 1A module 302 and the dynamic timestamp module 304 generates an intermediate key-A which is stored in an intermediate key-A module 308. Further, a second XOR operation is performed between the intermediate key-A and the key-2A value which is stored in a program specific to the cloud storage account-A 112 to generate a final key value. The final key generated is stored in the final key-A module 310. Further, the same operation is performed for the cloud storage account-B 114 to generate a final key for that account. In accordance with the present disclosure, the XOR operations between the key values and dynamic time stamps to generate an intermediate key and subsequently generating a final key is practiced till split file for each of the cloud storage is created. Further, once the final key is generated, the hash value is checked before generating the spilt file. The metadata of the generated split file is communicated with the database 120.

[0032] Once the final key values are obtained, a decryption is performed on the split files to generate the decrypted split files. The example of one such decryption algorithm is GNUPG decryption. The obtained decrypted files are then joined to present a complete data file to the user through the application server.

[0033] FIG. 5 is a block diagram illustrating file joining process, in accordance with the present disclosure. FIG. 5 includes a plurality of split files 402 which are decrypted. The application server 102 joins the decrypted split files and joins them using a plurality of joining process. Further, the application server checks and matches the hash values before allowing the user to download the complete files. In accordance with the present disclosure, when the checksum hash values do not match, an error message is displayed to the user. In accordance with the present disclosure, when the checksum has value for each of the encrypted and decrypted file matches, the files are joined. The joined files are displayed to the user through the user interface of the client device 202. The database 120 communicates with both the client device 202 and the application server 102 for receiving the update of the file joining process.

[0034] FIG. 6 is a flowchart illustrating the process of uploading a file to multiple cloud storage devices using a unified platform, in accordance with the present disclosure. At first, the process is activated when a user access the application server through the user interface provided on the client device (Step 602). The user provides a plurality of information including, but not limited to a name, a phone number, an e-mail id, a password, a security question, a recovery e-mail, details of the linked cloud storage accounts, and the like. Once the initial registration details are provided, the user uploads a plurality of files to the application server. In accordance with the present disclosure, the user directly accesses the user interface of the application server when he is already registered to the unified cloud platform.

[0035] Further, the application server receives the uploaded data from the user’s client devices (Step 604). The uploaded data includes a plurality of formats such as textual data, image data, voice data, multimedia data, different file types, metadata, and the like. Further,

the application server extracts a list of all the linked accounts in which the uploaded details may be split and distributed (Step 606).

[0036] Once the list of all the details of the linked cloud storage accounts is extracted, the application server computes the storage size of the uploaded data and the storage space available by combining all the linked accounts (Step 608). Further, application server computes the size of the uploaded data and available storage space. In addition, the application server computes a maximum file size computation per account. The computation of the storage spaces is based on the standard protocols known to the one skilled in the art.

[0037] A comparison of the size of the uploaded file and the available storage space by combining a plurality of linked accounts is made to check the possibility of splitting and distributing the files between a plurality of cloud storage accounts (Storage 610). When the uploaded file size exceeds the amount of available storage by linking the plurality of cloud storage accounts, an error message is displayed to the user on the user interface of the client device (Step 620).

[0038] However, when the size of the uploaded file is not greater than the available storage space by combining the plurality of cloud storage accounts, the files are split into a plurality of shreds (Step 612). The splitting of the file is done using a plurality of standard techniques known to the person skilled in the art.

[0039] Further, the split data is encrypted by generating a final key which includes two XOR functions. (Step 614). The split data is encrypted by generating a hash value which includes two XOR functions. The first XOR function is between a first key which is specific to the cloud storage account and the dynamic timestamp. In accordance with the present disclosure, the dynamic timestamp is derived from the immediate prime number of an EPOCH time stamp. The first XOR function generates an intermediate key which is further XOR’d with a second key which is specific to the other cloud storage account.

[0040] The encrypted splits are now distributed to a plurality of linked cloud storage accounts (Step 616). The distribution is done based on the plurality of standard protocols

known to the person skilled in the art. Once the encrypted splits are distributed in the plurality of cloud storage accounts, the session is temporarily stopped (Step 618).

[0041] FIG. 7 is a flowchart illustrating the process of retrieving the file from multiple cloud storage devices using a unified platform, in accordance with the present disclosure. The decryption process is initiated when the user wants to retrieve the uploaded file (Step 702). The encrypted files are first downloaded from the plurality of linked cloud storage accounts by the application server (Step 704). Then, the files are decrypted for allowing the application server to join the files (Step 706). In accordance with the present disclosure, the decryption is done using two XOR operations. The first XOR operation is performed between the key-1 which is stored in the application server and is specific to the cloud storage account and the dynamic timestamp. In accordance with the present disclosure, the dynamic timestamp is derived from an immediate prime number of the EPOCH timestamp dynamically. The first XOR operation generates an intermediate key which is again XOR'd with a key-2 which is program specific to the linked cloud storage account to generate a final key.

[0042] The plurality of decrypted splits obtained is joined to obtain a final file (Step 708). The joining of the plurality of shreds is based on the standard joining techniques known to the person skilled in the art. Further, a checksum of the file splits during the encryption process and the decryption process is matched to ensure that the file is not used by an unauthorized third-party (Step 710). When the checksum matches, the file is presented to the user for downloading (Step 712). However, when the checksum does not match, the user is presented with an appropriate error message and requests the user to initiate the process again (Step 716). Once the user is presented with a whole of the document or presented with an appropriate error message, the session is temporarily terminated (Step 714).

TECHNICAL ADVANTAGES

[0043] The present disclosure envisages a system and method to provide a unified platform for storing data efficiently across multiple cloud storage systems. The unified platform lets the user upload data content and distribute the same across multiple cloud storage systems. Further, the unified platform allows the content to be encrypted and then distribute to multiple cloud storage systems. Further, this system to provide a seamless and uniform interface to share, access and collaborate on data irrespective of the underlying cloud storage provider. Further, the proposed system envisages the end-user to connect multiple cloud storage accounts, and also avoid third-party hackers to steal the content without the user's permission.

ABSTRACT

A SYSTEM AND METHOD FOR DISTRIBUTING AND STORING DATA IN MULTIPLE CLOUD STORAGE SYSTEMS

A system and method for distributing and storing data in multiple cloud storage systems. Typically, the user uploads data to a unified platform by linking a plurality of cloud storage accounts. The application server receives the data uploaded by the user, splits the data into a plurality of chunks, encrypts the data and distributes the encrypted data in a plurality of linked cloud storage accounts. Further, when the user has to retrieve the uploaded file, the unified cloud platform decrypts the data from the multiple cloud storage accounts, joins the data and displays the same to the user.

Date: 08-Sep-2016 **Rakesh Prabhu**

Place: Bangalore **Counsel for the Applicant**