Improvement and Analysis of Encryption and Decryption Performance in Cloud Document

Yonglong Zhuang, Xiaolan Weng, and Yuwu Wang
Huaiyin Normal University, Huai’an 223300, PR China
Email: zhuangyongl@yeah.net, wxl@hytc.edu.cn, wyw@hytc.edu.cn

Haiyang Zhuang
Department of Civil and Environmental Engineering, University of Southern California, CA, USA
Email: OCEAN_JOE@hotmail.com

Abstract—Cloud computing is the hot topic in recent years, and the cloud-based collaborative service is the emerging service. No matter where the users are or which computers the users use, it enables the cloud users to place their data into the cloud. As long as it connects to the Internet, it can store and get the data placed in the cloud. In the collaborative service situation, it can allow other cloud users in the group to share with the private information, such as the editing service in the on-line common file can make many collaborators jointly write a file. It encrypted and decrypted the whole file in the past in order to protect the data privacy, but it was rather time consuming in the multi-person collaboration. The paper analyzes the text editing in the collaborative service and introduces the rbTree-Doc framework in order to reduce the encrypted data number. Although it increases the cost of establishing and maintaining the rbTree-Doc, the experimental results show that the rbTree-Doc framework can make collaborators conduct the text editing function. When it conducts the insert, remove operations and the encrypted/decrypted algorithms, it adopts 3DES encryption algorithm to improve its 31.04% efficiency, and adopting AES encryption algorithm to improve its 23.94% efficiency.

Index Terms—Cloud Computing, Data Privacy, Collaborative Service, Red-Black Tree, Encryption Algorithm.

I. INTRODUCTION

Nowadays many manufactures have launched the cloud text editing service, such as Google Docs, Microsoft Office Live, Zoho and Adobe Buzz Word. These cloud services not only have the intact function, but also can be provided to users for free. On the one hand, the cloud computing can reach a file portability in the application, and then users can place the data into the cloud. Therefore, no matter in the office or the outdoor, or which computer or cell phone is used, it stores and gets the data placed in the cloud as long as it connects to the Internet. On the other hand, personal computer’s equipment function does not need to be very strong with the use of the cloud to deal with computing and storage assignments. It does not carry the complex operation system and application procedure any more, and it does not need the computer equipment’s management maintenance personnel. In the future, the development of computer equipment will be simplified and cheap.

The online Office Suite based on Web application procedure is becoming very popular. Users not only easily use the network browsers to collaborate a document with ease. The cloud-based collaborative service is the emerging service. It not only allows other cloud users in the group to share with the private information, such as online common text editing and calendar sharing, but also offers its availability and the common stored/shared state. The collaborative service deployed in the cloud platform has many advantages, such as the global accessibility, high availability, high tolerance, flexible resource distribution and extension. However, these advantages should be established in the cloud service provider which should be completely trusted. In fact, the information concentration makes the cloud service provider become the attack object. It does not only possibly to cause the malicious intrusion behavior, but also produce the risk doubts, that is, the privacy information is leaked by the servers[1]. On the other hand, users do not really possess the information. It can deal with the data through the cloud. Therefore, users are worried to leak or lose the data, and the problem may be an important barrier to the cloud service development.

The premise of the paper is that the cloud users cannot completely believe the cloud service providers, while the usual data privacy preserving method adopts the modern cryptography technology to encrypt user’s sensitive data, and then places them into the cloud. If it does not possess the decryption key, it cannot get the contents of the sensitive data[2-4]. If users use the text editing collaborative service, the shared document can be a new blank document. It will commonly create an intact document from scratch, but the owner and the co-author of the document has the encryption and decryption key. Users adopt the encryption key to encrypt the whole document in the editing process, and then upload the cipher-text to the cloud in order update and store the cipher text. In order to ensure that all documents possessed by the collaborators are always the latest state, the cloud service will proactively notice and simultaneously open the collaborators of its document.

© 2013 ACADEMY PUBLISHER
doi:10.4304/jnw.8.9.2071-2077
After downloading them into the host, it restores the whole document with the use of the decryption key.

The consumed time in the encryption/decryption process is proportional to the size of the data. When the document is becoming mature and intact, its occupied space will also be increasing. Although it just modifies a small part of the content, it also encrypts the whole document. All collaborators must download the whole cipher-text and then decrypt it to get the latest document. In the multi-people collaborative situation, the frequent modified documents equal to produce many encryption/decryption computations. It needs to consume much encryption/decryption time with the increasing of the data size in the document. The paper analyzes the text editing in the collaborative service, it divides the text inside the document into many blocks, and then maintains the corresponding position in all blocks with the use of red-black tree. When the text inside a certain block is altered, it just needs to encrypt and update its block. Finally, it makes the users who are doing the text editing function, such as insert/remove operations and encryption/decryption computations, locate in the highly efficient and no inductive state.

II. RELATED WORK

The chapter firstly explains the cloud computing definition, and then discusses the related papers which store the cloud data privacy. Finally, it introduces the red-black technology.

A. Cloud Computing

The popularization of the network and the mature of the virtual technology offer the cloud computing a good environment. Cloud computing represents a new information framework. It automatically divides the huge computing procedure into many small sub-procedures through the distributed computing. It deals with the computing distributed in the mass computers, and then passes the result back to the client. The server provider offers procedure computing, software application and data storage ability so that users can ubiquitously store and get them. Finally, data and server storage is distributed on the large data center established by the server provider. It can get the needed information and service by connecting user's equipment to the network.

According to the American national standards and technology association (NIST), it defines the present cloud computing standard as the following: 5 basic features, 4 deployed models and 3 server models.

5 basic features are as the follows:
(1) On-demand self-service
(2) Ubiquitous network service access with any devices
(3) Rapid elasticity of deploying
(4) Location independent resource pooling
(5) Measured Service

4 deployed models are as follows:
(1) Private Cloud: Private cloud is established by itself or is managed by the third party, and especially offers service. If it requires high data confidentiality, it can use the private cloud.

(2) Community Cloud: Community Cloud is established by multi groups or is managed by the third party, and especially offers service. It is suitable for academic unit required to share with the research data to establish it.

(3) Public Cloud: Public Cloud is possessed by a certain cloud server provider, and is open to the public or enterprise groups to offer cloud service. It is mostly suitable for the start-up business or small-medium enterprise whose data confidentiality is low.

(4) Hybrid Cloud: Hybrid Cloud is mixedly used by the above two or more clouds. It should ensure the data and application procedure's portability in different platforms on the basis of standards or with the use of the new technology.

3 service models are as follows:
(1) Platform as a service (PaaS): PaaS offers computing, storage, network and other resource's renting service, but users cannot contact with the hardware in essence. IaaS offers multi virtual machines into a certain host inside the computer room, and it can change the CPU number and storage space in terms of user's request.

(2) Software as a service (SaaS): SaaS provides software for users to use, and it is unnecessary for users to download or install any procedures. Users can directly adopt SaaS provider's application procedures by browsers or other specific tools connecting with the network. The Software-on-demand and Application-on-demand allows users to rent server provider's required application software's through the internet in terms of the real request.

B. Cloud Information Privacy

In order to solve the cloud information privacy problem, [2-4] are mainly focus on the policy protocol, and offers the cloud server provider a ideal execution policy. [6] in the experimental result shows that it can withstand Denial of Service (DoS) attack. The above situations are still on the premise of trusting cloud server provider. It can increase the existing cloud server's usage limitation when it executes these safe policies.

Different from formulating the safe policy, [2-4] assumes that users adopts the modern cryptology to encrypt sensitive data without trusting the cloud server provider completely, and then place the data into the cloud. If it does not possess the decryption key, it cannot get the contents of the sensitive data. [4] [7] regards Google Docs cloud server as the experimental scale, it designs a plug-in and then install it in the browsers. After downloading Google Docs document into the local computer, it will decrypt the document through the plug-in and then provide the decrypted document for users to check and edit. During the editing process, it also encrypts the document through the plug-in and then
uploads the Google Docs cloud for updating and storing. Although the accomplishment of the method is simple, how do other users shared with the document get the key will be a problem. Therefore, the concept of the attribute-based encryption (ABE) is taken seriously year by year [5] [8] [10]. ABE is similar to the traditional public key infrastructure (PKI), but their difference is that ABE adopts the randomly produced public key. Entity will adopt unique word string which is like the e-mail address as itself public key. ABE mechanism possesses the following 4 features: 1. It is necessary for resource providers to encrypt sensitive data in terms of attributes and unnecessary for them to be informed of group's identity and number so that it can reduce the expense of the data encryption and protect user's privacy. 2. As long as the groups accord with the required cipher-text, it can decrypt data and guarantee the data privacy. 3. User's private key is related to random polynomial or number. Different users' private key cannot be combined so that it can prevent users' collusion attack. 4. ABE possess the flexible access control strategy. It is very suitable for us to apply ABE to the cloud collaborative sharing service for ABE has the above advantages. [9-12] emphasize on improving ABE and introduce the idea of the fine-grained access control, while [7-9] introduce the generation of the key and the method of the distribution.

C. Wang etc, in [13-16] thinks that users can save the host hardware space and maintenance cost by storing the data into the cloud, but cloud service provider for all users have the storage of data integrity protection is a challenge, and the reality of the user's data or the hidden damage is not found. Therefore, the literature propose that the third auditory(TPA) should be responsible for helping user check dada's integrity, but it does not hope that the third auditory(TPA) is informed of the data contents stored in the cloud for it can protect user's data privacy.

C. Red-Black Tree

Red-black tree is a data structure of a binary search tree, which was first published by R. Bayer [17] in 1972. At that time, it was called symmetric binary B-tree. The present usual name was derived from L. J. Guibas and R. Sedgewick [18] in 1978. Red-black tree is an approximate balance tree, and its method is to color the node for ensuring that no maximum path is over twice as much as any other path. Red-black tree has the following 5 properties [19]:

1. Each node's color must be red or black.
2. Each leaf node's color must be black.
3. Root's color must be black.
4. If its parent node's color is red, the two children node's color must be black.
5. Its passed black node numbers must be the same, including any path from the root node to the leaf node.

Due to the property (4), it causes Red-black tree cannot have two red nodes connected. Therefore, the shortest possible path has the black node, and the longest possible path is the alternative red and black node. According to the property (5), all paths from the root node to the leaf node has the same number of the black node. It also shows that no path can surpass the twice length of other paths, ad the result is that the tree is generally balanced. There is complex algorithm[20][21]to support red-black tree so that it can guarantee it has good worse situation operating time in the process of search, insert and delete operation. Table 1 lists the time complexity when red-black tree deal with the search, insert and delete operation, n represents the number of the red-black tree node. Owing to its high efficiency, it is usually used in the real-time processing application.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Time complexity in worse case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>O(log n)</td>
</tr>
<tr>
<td>Insert</td>
<td>O(log n)</td>
</tr>
<tr>
<td>Delete</td>
<td>O(log n)</td>
</tr>
</tbody>
</table>

III. PROPOSED SCHEME

It develops the present text editing collaborative service mostly based on Asynchronous JavaScript and XML. AJAX [22] is a web development technology widely applied in the browsers, and it can transmit and update the data without updating the whole web. Meanwhile, it can avoid the server sending other unchanged information. It largely reduces the required transmitted data for it is unnecessary to reload the whole website, and then it makes user cannot think there is the process of transmitting the data or refreshing the screen. When client-site executes the link, form, text and other operations, it can produce JavaScript [23] triggering event, and then it transmits these events to the server-side for being solved. When servers respond, the client-side would simultaneously deal with it with the use of JavaScript, and then respond it and update the screen.

The paper introduces rbTree-Doc structure. The client-side needs to generate and maintain rbTree-Doc document. For example, users can choose the desired editing document and then download the necessary information to the host-side. Later, it decrypts the document, computes and reestablishes rbTree-Doc. Finally, the contents of the document is shown in the browsers. In the editing process, it synchronously updates rbTree-Doc, and then offers the corresponding node a tag to represent the contents have been changed when the document is executed the insert, remove operations. It just needs to deal with the node text possessing the tag without requiring the whole document, and then encrypts and computes the text. Later, it uploads the text into the cloud and the update it. The next chapter explains the rbTree-Doc data processing procedure in detail.

A. System Design

The system design is as shown in the figure 1. The cake drawing represents the cloud and the cloud drawing represents the cloud collaborative software. The right icon represents database function. The Rebuilding function means that it can accord with rbTree-Doc form when users choose to upload the existing document. Firstly, the cloud collaborative software (App) will take out user’s desired editing document file from the database,
and then record/store all users of its document. When users start up the document, it must execute the Building rbTree-Doc and then reestablish rbTree-Doc. In the meantime, it decrypts the document. In the editing process, it synchronously updates rbTree-Doc through Update_rbTree-Doc. It just encrypts and computes the contents of the changed tag node, and then it uploads and updates them in the cloud. The cloud collaborative software (App) also notices and transmits the necessary information to other collaborators, except updating/storing document. In the process of updating the document, it just needs to update the node and decrypts the updating text contents for users have established rbTree-Doc by themselves. Finally, it restores the document by operating Recovering function.

![Figure 1. The system flow chart in the paper](image)

1. Document Archive Form

Users can easily execute the document editing for network browsers can be the same as the traditional desktop text editing software. The process is mainly owing to the HTML editor of (What You See Is What You Get; WYSIWYG). The HTML editor of WYSIWYG can make the effect of the text, figure and other contents in the editing process be the same, such as the effect of the picture displayed on the screen is the same as the effect of the picture printed on the paper. It can offer the intuitional imaging interface (GUI), and can initiatively transform the contents into HTML so that users can easily edit the document without imputing any HTML tags.

When users upload the existing document, it must transform the user-define form document archive into the HTML form document archive. Rebuilding function transforms the original HTML form document archive into the content and style sheet; while Recovering function combines the content and style sheet, and then retrieves the original HTML form document. The coding method of the content is UTF16 for it can avoid producing the messy code and the problem of not unscrambling it correctly. The UTF16 coding method is the variable length method. For example, English words occupy 2 bytes, the usual Chinese words occupy 2 bytes, and the frequent used Chinese words occupy 4 bytes. After getting the content during the process of executing Rebuilding, it divides it into many blocks in terms of the pre-setting block length, and then encrypts/computes each block text respectively. Finally, it saves a XML form archive, as shown in the figure 2. In addition, the XML document archive and form archive will be jointly uploaded and stored in the cloud.

If block length is set as 5 bytes, it shows that each block just store 5 texts at most. If fill factor is set as 100, it shows that each block will fill 100% texts at most. Generally speaking, it is just set between 75-90 so that each block can leave some space for the future new texts to be stored in its block directly. It does not only reduce the computing number, but also add the number of the block. Encryption algorithm (encrypt-Alg) shows that it can use any encryption algorithms, such as Advanced Encryption Standard (AES) is an usual encryption algorithm. Element_id is the first text position (index) in the whole document in each block, while <data> is the encrypted cipher in each block text. The most important information is element_id and <data> when it establishes rbTree-Doc.

2. Establishing Red-black tree

When users firstly opens the document, it can receive the two archives from the cloud collaborative software (App): one is XML document archive, the other is form archive. In order to show the original document contents, it uses the red-black tree and element_id value. The feature of the red-black tree is the auto-balanced binary search tree and can be sorted in terms of the key in the key-value pair. Element_id is the first text position (index) in the whole document in each block, while it also shows that element_id is not only the unique value, but also possesses the sequence so that it is very suitable to be the red-black tree key. Take the figure 2 as an example the reestablished rbTree-Doc has two nodes, the key set is \{0,5\}, and the node content(value) is the decrypted <data> whose element_id is located in the block.

B. Data Processing Flow

After reestablishing the rbTree-Doc, users may do the following operations to the document, such as insert and remove operation so that it may change partial or the whole data. If the document contents are changed; it can maintain rbTree-Doc or upload partial/the whole nodes. In the process, it must require high efficient and user’s no inductive state, except the document contents should be completely corresponded to rbTree-Doc.

1. Position

It is crucial to get document’s precise position which influences the content before realizing the highly efficient document data handling. When the document contents are changed, it can get the mouse cursor position in the document through the program writing. The next step is to find the corresponding rbTree-Doc node in the
document changed contents. It is unnecessary to use traversal method to search the key set for rbTree-Doc regard the first text's position in the block document as the key. It as better efficiency if it directly uses the search key set. After getting the corresponding nodes, it should transform the mouse cursor position in the document into the node position in the text block of offset-position- node ID. Position is the mouse cursor position in the document, and node ID is the node key. Therefore, wherever edits it in the document, rbTree-Doc can rapidly find out the corresponding node and its text position, and then do the corresponding operation handling.

2. Insert Operation

When the document inserts texts, other nodes are unnecessary to make any variations, except the corresponding nodes in the rbTree-Doc will be influenced. When it happens insert events, it can get the present mouse cursor position in the document, the inserted words, the corresponding node's node ID, its word position (offset), and then update key set. When it happens the insert events, it also increases partial words' position for the key is the first document position in the node block. Therefore, it must give the correct new key = key - word length, that is, delete the position after the words. At this time, it should change the key, and the rbTree-Doc framework is not changed.

The insert word is usually a word, also can be a more word, such as copying/pasting a section of words. If the length of the node block plus the word length is not more than the setup block length, it can directly insert the word into the node's word block. If the remaining space of the node block is not enough to be placed into word, it will cut the offset into before and after partial words. At this time, it will fill the before word into a block length. The final insert node and key is still the first word document position, and its content is the remaining word and after. If the length of word is surpassing to a block length, it will divide the word in terms of the block length and then insert the node to store it. If the remaining word length and the after length are surpass to a block length, it should insert a node to store word and after respectively.

3. Remove Operation

When it happens the remove events, it can get the present mouse cursor document position and then delete the word length. Firstly, it can get the corresponding first/last influenced node's node ID. If it is the same node ID, it can directly remove the word; otherwise, all related words between the two nodes will be deleted. If the word can not be stores in the node, it will immediately remove its node. Finally, when it happens the remove events, partial word position will be reduces so that it needs to give the correct key new key = key - word Length, that is, delete the position after the words. In addition, the update operation in the word actually is a remove operation plus an insert operation. Therefore, the paper will not describe it in detail.

IV. EXPERIMENTAL ENVIRONMENT AND DATA ANALYSIS

The paper uses the usual encryption algorithms. It adopts 3DES and AES to do the experiment. The key length of Triple DES (3DES) can be 56, 112 or 168 bits, while the key length of AES can be 128, 192 or 256 bits. Generally speaking, if the key length is longer, the encryption safety is higher, but the encryption time is correspondingly increasing. The paper adopts the highest safety key length, and all encryption algorithms adopt CBC working model and PKCS5Padding filling method. In the process of the experiment, all tests are operated in the Virtual Machine (VM) which has 2.33GHz Core 2 CPU, 2GB memory and Windows XP 32bit. The procedure is written by JAVA (JDK 1.6). The paper does the experiment aiming at the 4 different sizes of the document archives: 1MB, 4MB, 8MB and 12MB. The document content is finished by using the filling method in the JAVA explanation document.

<table>
<thead>
<tr>
<th>Encryption algorithm</th>
<th>1MB</th>
<th>4MB</th>
<th>8MB</th>
<th>12MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DES Encrypt(ms)</td>
<td>187.4</td>
<td>618.6</td>
<td>1131.4</td>
<td>1721.8</td>
</tr>
<tr>
<td>3DES Decrypt(ms)</td>
<td>146.8</td>
<td>581.2</td>
<td>1100</td>
<td>1709.2</td>
</tr>
<tr>
<td>AES Encrypt(ms)</td>
<td>112.8</td>
<td>327.8</td>
<td>628</td>
<td>915.8</td>
</tr>
<tr>
<td>AES Decrypt(ms)</td>
<td>97.2</td>
<td>325</td>
<td>643.6</td>
<td>952.8</td>
</tr>
</tbody>
</table>

Table 2 represents the consumed average time when it encrypts/decrypts different sizes of document archives with the use of the encryption algorithm. In the multi-people collaborative situation, it must upload the user's modified document contents into the cloud collaborative service at intervals and then notice other collaborators to update their documents. Take Google Docs as an example, it sets an AJAX as the updating trigger, it will initiatively uploads the changed contents every 30 seconds [24]. If just a small part of the contents are modified, it also can encrypt the whole document. In order to get the latest
document, all collaborators still upload the whole encrypted document and then decrypt it. If it is the frequent modified document contents, it equals to produce mass encryption/decryption computing. User’s consumed time in computing the encryption/decryption is still considerable long.

According to the idea, the proposed rbTree-Doc in the paper will do the experiment when the word content in the 1 node, 5 nodes, 25 nodes and 125 nodes is changed. In the experimental process, it does not directly insert and delete a word in the document random position until it reaches the setup changed node numbers. Figure 3 is the consumed time in the encryption computing when the block length is set as 1000 bytes. Compared with the table 2, the encryption computing to the whole document has considerable high efficiency and the required transmitting data number in not very large. The result is that collaborators can have more efficiency when updating the document contents.

V. CONCLUSION

The cloud-based collaborative service is the emerging service. In the past, encrypting/decrypting the whole document method for protecting the data privacy in the multi people collaborative situation were rather time-consuming. The paper analyzes the word editing in the collaborative service and introduces rbTree-Doc framework for reducing the required encrypted data number. Although it increases the cost of establishing and maintaining the rbTree-Doc, the experimental results show that the rbTree-Doc framework can make collaborators conduct the text editing function. When it conducts the insert/remove operations and the encrypted/decrypted computing, it adopts 3DES encrypting algorithm to improve its 31.04% efficiency, and adopting AES encrypting algorithm to improve its 23.94% efficiency. In the future, it hopes that it can reduce the consumed time in analyzing, dismantling, and restoring the word form, and it can reduce the waste of the space in the rbTree-Doc insert/remove mechanism. Merging the word blocks can reduce the node numbers and improve the operating efficiency in the Red-black tree.

ACKNOWLEDGEMENT

This study was supported in part by the Education Department of Jiangsu province industrialization project JHB2012-53 and Science & Technology Support (industrial) Project HAG2012059 in Huainan city.

REFERENCES


