ATLANTIS PRESS

Advances in Computer Science Research, volume 79

2nd International Conference on Sensor Network and Computer Engineering (ICSNCE 2018)

A Searchable Re-encryption Storage Method in Cloud Environment

Wang Hui
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
e-mail: 277019826@qq.com

Hong Bo, Tang Junyong
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China

***Abstract*—Traditional cloud storage systems do not adapt well to different application environment and does not guarantee the integrity and confidentiality of cloud data. In order to solve the problems brought by the hysteretic and density in the cloud storage system,A Searchable Re-encryption Storage Method (SReCSM) is proposed in this paper. The central idea of the SReCSM is to generate a re-encryption key and decrypte while the keyword matched. Simulation results show that SReCSM introduced in this paper has better security and reliability.**

Keywords-Cloud Storage; Re-encryption; Decrypt; Reliability

I. INTRODUCTION

In order to meet the various storage demands, cloud storage is designed to store data in cloud and is widely used in the Internet. Compared with traditional data storage, it greatly improves the efficiency of the mass data storage and utilization of network resource. It is essential to reduce user download wait time for a requested file from a network to enhance client quality experience. Therefore, cloud storage faces serious security and efficient problems [1].

Traditional cloud storage systems do not adapt well to different application environment and does not guarantee the integrity and confidentiality of cloud data. Therefore, it's very dangerous for sensitive data to be stored directly in the cloud. The reliability of the mobile cloud storage depends on the extent of the impact on system storage efficiency while the storage solution fails [2]. Therefore, storing sensitive data on un-trusted server is a challenging issue [3]. Simple encryption techniques have key management issues and which can't support complex requirements such as query, parallel modification, and fine-grained authorization. To guarantee confidentiality and proper access control of sensitive data, classical encryption is used [4].To solve the problems brought by the hysteretic and density of traditional storage methods in the cloud storage system, in this paper, SReCSM, Searchable Re-encryption Storage Method is proposed.

II. RELATED WORK

Efforts have been taken by researchers, developers, practitioners, and educators to identify and discuss the technical challenges and recent advances related to cloud storage. Han et al. proposed [5] multi-path data prefetching in mobile cloud storage. Wang et al. [6] presented an Optimized Replica Distribution Method (ORDM) in cloud storage system. Chen et al. [7] proposed a new metric called joint response time, which not only considers the waiting time when the requested data are unavailable but also the queuing delay and service time when data become available.

Tysowski et al. proposed several useful schemes in [8]. One scheme is a Manager based Re-encryption Scheme (MReS), which acomplished the security through the proxy re-encryption scheme. And the other cryptographic scheme proposed in [8] is Cloud-based Re-encryption Scheme (CReS). Although MReS and CReS make up for the limitations of the existing schemes, these two schemes are more complex and need more time to re-encrypt. [9] propose proxy re-encryption algorithm to confirm the security of the datain thecloud, which can alleviate the client's burden, and enhance the confidentiality of cloud data. However, there are two major disadvantages with these techniques. First, for re-encryotion, the data owner must obtain user's public key before uploading. Second, because the same plaintext is used with different keys gerenated by proxy, therefore, the storage overhead becomes excessive.

III. SEARCHABLE RE-ENCRYPTION METHOD

The central idea of the searchable re-encryption is to generate a re-encryption key and decrypte while the keyword matched. By using searchable re-encryption method, SReCSM may increase the storage requirements because all the encrypted data must be stored. The objective of the proposed searchable re-encryption method is to improve the storage requirements, bring down the security risks, minimize the scheduling time, overhead ratio and the storage cost. This technique will not provide the effective data utilization. The new method SReCSM is easier and faster to implement in the cloud while sorting the matrix by using the searcherable keyword. SReCSM would reduce the cost and the time complexity will be reduced.

A. Symbol Definitions

Symbols in the Searchable Re-encryption method are defined in the in Table I.

TABLE I. SYMBOL DEFINITION

Symbol	Definition
PR(Key)	Private key of the user
PU(Key)	Public key of the user
Key(Wd)	Keyword
DB	Database
ED(Key)	Editing Keyword
EN(Dat)	Encrypted Data
PR(Key)Re	Re-encrypted Private key
PU(Key)Re	Re-encrypted Public key
ED(Dat) Re	Re-encrypted Encrypted Data
DE(Dat)	Decrypted Data
Key(U)	Keyword got through the public key
Key(R)	Keyword got through the private key
Sen	Sender
Rec	Receiver
Ser	Cloud Server

B. Keyword Editing

The keyword can be edited in three cases while searching. We can edit the keyword while inserting, deleting or substituting. Let us suppose that n is this notation and the keyword edited is $ED(Key)$.

Case 1 If $ED(Key) \sim= n(i)$

Inserting the character into the first place
For example:
Character string (old) = ‘‘BOK’’
If $i=3$; $ED(Key) \sim= n(3)$
In third place have to insert new character
Character String (new) = ‘‘BOOK’’

Case 2 If $ED(Key) < n(i)$

Deleting the character into the first place
For example:
Character string (old) = ‘‘BOK’’
If $i=3$; $ED(Key) < n(3)$
In third place have to delete the letter
Character String (new) = ‘‘BO’’

Case 3 If $ED(Key) > n(i)$

Substitute the character into the first place
For example:
Character string (old) = ‘‘BOK’’
If $i=3$; $ED(Key) > n(3)$
The third position in the keyword is replaced with a new character
Character String (new) = ‘‘BOK’’

These three different cases can help us edit the keywords easily. The user handles some of the wrong situations by using the keyword. And we can edit the keyword to recover errors through these three different situations.

C. Searchable Re-encryotion

The core of the searchable re-encryption method is to generate a re-encryption key only when the keyword matches to the character. Re-encryption operates over two groups G_1 and G_2 of prime order q with a bilinear map e: $G_1 \times G_1 \to G_2$. The parameters of the cloud system are random generators $g \in G_1$ and $Z = e(g, g) \in G_2$. The Searchable Re-encryption can be defined in the following algorithms.

Next, the algorithm 2 is described in detail. In algorithm 2, the keywords should be searched first. The file would be decrypted with this keyword only when the keyword matches. And then the data can be received, stored, and accessed in the cloud.

Algorithm 2: Searching Keyword
Input: Keyword for $PR(Key)$
Output: Keyword Matching
If $PR(Key) = Key(Wd)$
  Goto $DB(i)$
Else If $ED(Key) > n(i)$ \\ case-3
  Goto $DB(i)$
  Keyword Not Matching
  Else If $ED(Key) < n(i)$ \\ case-2
  Keyword Matching"
End If

The keywords are included in the private key and will be searched in the database. Keyword contained in the private key should be validated in the cloud storage database. In the cloud system, the data will be transmited to the servers and then be stored in the database. Data can be accessed from the database through this keyword. The mode of "Keyword Editing " can be used to insert, delete, and replace when the keyword does not match. On the contrary, the data will be received from the cloud server once the keyword matches.

Algorithm 3: Sharing Data
Generate $PU(Key)$ and $PR(Key)$
$PU(Key) = \{EN(Dat), Key(U)\}$
$PR(Key) = \{Key(R)\}$
Sen shares $PU(Key)$ to Ser
Sen shares $PR(Key)$ to Rec
Send $EN(Dat)$ to Ser
Rec sends $Key(R)$ to Ser
Ser verify $Key(R)$
If $Key(U)$ = $Key(R)$
  Ser sends $\{DE(Dat), Key(R)\}$ to Rec
End If

The data may be shared between the owners and the users through the cloud server, which is introduced in detail in algorithm 3. First, the public key and the private key will be created by the data owners. Next, the public key is shared to the cloud server and the private key is shared to the data receiver. There is the keyword in each private key and public key, and the data users can retrieve data through this

keyword. Finally, the data will be transfered by the cloud server only when the keyword of the public key and the keyword of the private key is matched.

Algorithm 4: Encrypting Data

Define two prime numbers as s and t

Assign $r = s*t$, where r will be used for the module of the private key and the public key

Assign Euler's function as $E(r) = (s-1)(t-1)$

Assign i as integer and $1 < i < E(r)$ for all $\{i, E(r) = 1\}$

In which, i and $E(r)$ are co-prime

Assign $D = \{f_1, f_2, \cdots, f_n\}$; f as file, D as data and n as the number of the files.

$$D(r) = \{f_1(r), f_2(r), \cdots, f_n(r)\}$$

Encrypted data,

$$EN(Dat) = \{D(r) \ mod \ E(r); Key(Wd), PU(Key)\}$$

Algorithm 4 introdeces the scheme of data encryption, in which the data will be encrypted through the RSA algorithm. First, the two prime numbers are multiplied together, and then compute the product. Each data may be encrypted with Euler's function of $E(r)$. Keyword of the public key and private key are contained in encrypted data. And each data is modeled by using function of $E(r)$.

Algorithm 5: Re-encrypting Data

Generate $PR(Key)_{Re}$, $PU(Key)_{Re}$

Assign $D = \{f_1, f_2, \cdots, f_n\}$, $g \in G_1$; f as files, D as the data and n as the number of the files.

$$D(r) = \{f_1(r), f_2(r), \cdots, f_n(r)\}$$

Re-Encrypted data,

$$ED(C_M)_{\mathrm{Re}} = PU(PU(Key), \quad ED(C_M))_{Re}$$

Algorithm 5 introdeces the scheme of data re-encryption, in which the data will be encrypted through the AES algorithm. The re-encryption keys for authorized user's list will be generated by the user 'M'. The public key and private key of the users' can be used by 'V'.

$$PU(Key)_{\mathrm{Re}} = PR(Key)_{\mathrm{Re}} = (g^{x_i})^{\frac{1}{x_M}}, \forall v_i \in V$$

The CSReSM generates the $v_I \in Z_q^*$ and encrypts the data through the following steps randomly:

$$Z_{new}^{v_I} = \frac{Z^{v_I}}{PU(Key)}$$

$$(EN(Dat) \cdot Z_{new}^{v_I}) = (Z^{v_I} \cdot M)$$

$$EN(C) = Z^{v_I} \cdot M, \quad EN(CM) = g^{v_I x_M}$$

The encrypted data '$EN(C)$' will be uploaded through CSReSM and '$EN(C_M)$' represents the mobile user 'M'. The CSReSM transforms '$EN(C_M)$' into '$EN(C_M)_{Re}$' using the re-encryption key $PU(Key)_{\mathrm{Re}}$ and $EN(C_M)$ as shown in the following equation:

$$EN(C_M)_{\mathrm{Re}} = e(PU(Key), \quad EN(C_M))$$

$$= e(g^{\frac{x_i}{x_M}}, \ g^{x_M v_I})$$

$$= e(g, \ g)^{x_i v_I}$$

Algorithm 6: Decrypting Data

Consider D key = $\{PR(Key), Key(Wd)\}$

For $f = 1$ to N //Data files

$$DE(Dat) = \{EN(Dat) \ mod \ PR(Key)\}$$

End for f

loop

Goto $DE(Dat)$

Algorithm 6 introdeces the scheme of data decryption, in which the data will be decrypted through the steps described above. First, the decryption key will be used by the user, and the decryption key contains the keyword of the private key. After selecting the number of files, the data will be decrypted through modulo function.

D. Security analysis

Assuming that the reliability of the cloud storage system is identified by symbol A. The time of encryption through different encryption algorithms is A_t, the encryption time A_t is reversed first, and after the normalization processing, A_j is got from A_t.The storage cost of the different node is normalized to be thevalue A_k. The number of the storage states for the cloud storage is the value n the reliability model of the system is shown in formula (1).

$$A = [1-(1-A_j)^n][1-(1-A_k)^n] \quad (1)$$

It can be concluded from the analysis of the reliability model. When the vale t tends to be infinite, which means $t \to \infty$, the storage cost of the node in the cloud system also tends to a certain stable value, and the state of storage strategy tends to be stable. When the value of A_j and A_k are more closer to 1, and the value of n is more large, the cloud storage system will be more reliable and with higher security.

IV. EXPERIMENT

The proposed scheme SReCSM Searchable Re-encryption Storage Method was developed and verified through Python. The real-time ability, security and reliability of SReCSM was evaluated through a series experiments. In this part, we analyzed the prediction, the searching time, searching efficiency and storage space while performing moving, copying, encryption, re-encryption, storing and decryption. The final experimental results are comparative analyzed with the existing technologies including CReS and

MReS, and the results show that SReCSM has better security and reliability.

A. Uploading and Downloading of SReCSM

The data response tests are performed on file upload, file copy and file movement, for large files and small files respectively.It can be known that 94 percent of transactions in a mobile cloud storage system can be implemented quickly within two seconds. The experimental results show that the system responds fast. Table Ⅱ and Table III are the test result.

TABLE II. THE PERFORMANCE OF THE MOBILE END UPLOAD THE DATA

type of test	utilization rate of Mobile CPU （%）	transmission rate （Mbps）	utilization rate of Mobile /transmission rate
Raw data	16.436	3.57020	4.603663
After the encryption	38.432	2.36015	16.283711

The ratio of CPU occupancy to upload speed is shown in table Ⅱ, which the data on the mobile side are tested before the encryption and after the encryption respectively.

TABLE III. THE PERFORMANCE OF THE MOBILE END DOWNLOAD THE DATA

type of test	utilization rate of Mobile CPU （%）	transmission rate （Mbps）	utilization rate of Mobile /transmission rate
Raw data	14.681	3.90135	3.763056
After the encryption	35.221	2.76147	12.754439

The ratio of CPU occupancy to download speed is shown in table III, which the data on the mobile side are tested before the decryption and after the decryption respectively. It can be known from the table Ⅱ and the table III, if the encryption and decryption mechanism are used for transmission, then the CPU utilization will be increased by an average of 22% ~ 25% and the overall file transfer rate will be reduced by 30% ~ 35%. As we can see，when the encryption and the decryption mechanism are used, more than three times the performance loss can be caused on the mobile end side.

B. Encryption and Re-Encryption SReCSM

The keyword can be set and searched in proposed method SReCSM. Only if the value of the keyword just matches, data can be received. Suppose that the number of the users in proposed method SReCSM is 600 users, and the available number of the searching keywords range from 600 to 6000. The cloud server store the keywords and all encrypted data through the database.

There are two questions to be considered: One is the impact of encryption and decryption on file speed.The other is the impact of encryption and decryption on the performance of the client host. The experimental data are listed in Table IV, which includes the time spent on encrypting the different sizes or different type files by using SReCSM and the time spent on transmitting the file.

TABLE IV. TIME COMPARISON ON ENCRYPTION AND DECRYPTION BY USING SRECSM

File size （M）	File type	SReCSM encryption （ms）	HDS upload （ms）	SReCSMdecryption （ms）	HDS download （ms）
3.07	pdf	1050	2685	370	2800
3.22	MP3	1178	2600	478	2830
23.8	mkv	3238	5930	2648	6290
25.8	doc	3140	5260	2163	6400
166.518	rmvb	23830	46400	16500	42460

It can be concluded from the above test data, the time spent on encryption or decryption by using SReCSM is regardless of the file type.

The same size files are encrypted by different CReS, MReS, or SReCSM algorithms and then the re-encryption time is different, as shown in table V and Figure 1. The 167.58 MB file in table V is the test case.

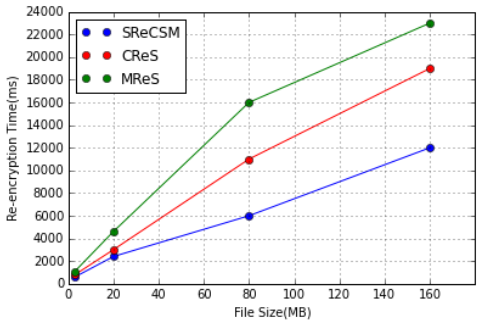

Figure 1. Comparison of Re-encryption time

In the SReCSM proposed in this paper，the time of encryption or decryption is relatively short. It may take a relatively long time to encrypt files by using the CReS, which cause a significant additional time overhead for HDFS. However, the encryption time that MReS encrypting the file was not significantly increased compared to SReCSM. Besides the impact on overall transmission rates, the impact of encryption and decryption on mobile performance is also important.

TABLE V. TIME COMPARISON FOR DIFFERENT ALGORITHM ENCRYPTION

File size （M）	MReS Re-encrypt （ms）	CReSRe-encrypt （ms）	SReCSMRe-encrypt （ms）
3.04	1048	770	720
23.15	3230	2901	2600
80.35	12010	10230	8560
167.58	23820	23612	23598

The comparison of the storage space required in different schemes is shown in Figure 5a. As the number of keywords increase, the storage space required by the cloud system also becomes larger. More storage space have been required by the existing schemes such as CReS and MReS. However, we

can see in figure 2a that SReCSM utilize the data transfer effectively and moreover reduce the storage space requirements in the system.

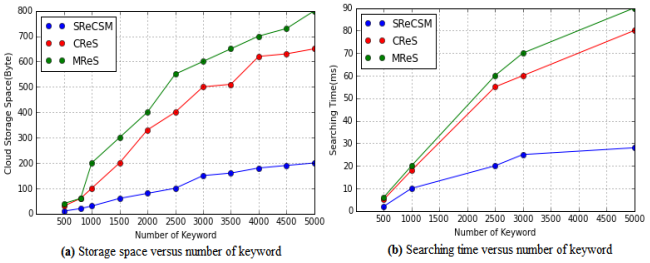

Figure 2. Comparison while increasing the number of keywords

It can be observed from the trends presented in Figure 2b that the searching time varies according to the number of keywords. The available number of the searching keywords range from 600 to 6000. As the number of keywords increases, searching time required by the system also increase. More searching time have been need by CReS and MReS. And we can see in Figure 2b that lesser time was need to search using SReCSM. If the searching keyword does not match, SReCSM will immediately use the edit function.

MReS, CReS, and SReCSM offload the re-encryption operations in cloud system. And then in the following experiment, we examined the energy consumption and turnaround time while the re-encryption operation were performed. The experimental results are shown in Fig. 3.

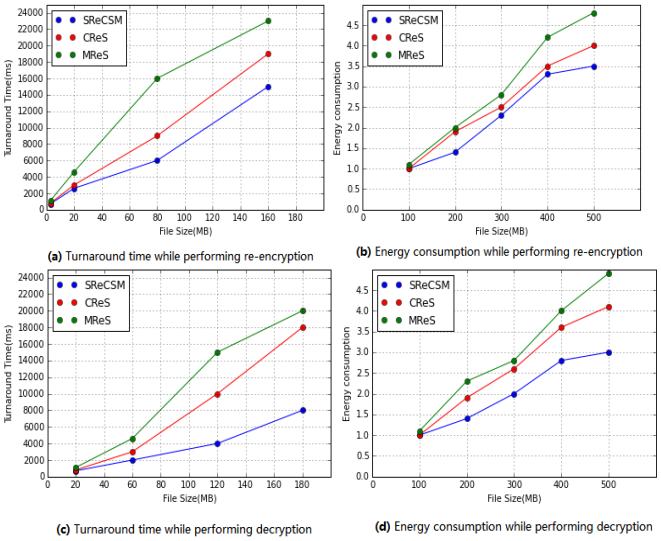

Figure 3. Comparison while re-encryption and decryption

We can see from Figure 3a and 3b, while re-encryption operations were performed in mobile terminal, the turnaround time and energy consumption will increase according to the file size increases.In the same way, Figure 3c and 3d show that, while decryption operations were performed in mobile terminal, the turnaround time and energy consumption will increase according to the file size increases.

Using the reliability model formula (1) of the cloud storage system proposed in 3.4, combine the time required for processing the same size of file in table III, when a different algorithm CReS，MReS and SReCSM is used, the encryption time required for encrypting file, after that the encryption time is reversed, A_j then be got after the encryption time is normalized.In the same way, after normalizing, storage cost A_k is got. If both A_j and A_k are closer to 1, and the number of storage state in the cloud storage system is larger, then the reliability of the system is higher. According to the above analysis，the data in one hour is sampled continuously, combined with the data in table III and table IV, the reliability contrast diagram for SReCSM is shown in figure 4.

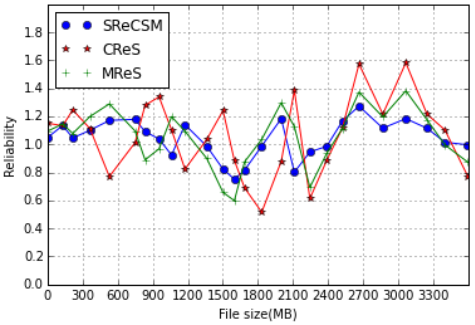

Figure 4. The reliability contrast diagram

It can be know the reliability of the system through different algorithm by comparing the data in figure 4.The reliability of the system is relatively high. That is， it has little impact on file transfer and user experience by using CreS re-encryption. It may take a relatively long time to re-encrypt files by using the MReS.However, the re-encryption time that MReS combined with CReS for re-encrypting the file was not significantly increased compared withCReS. The value of the reliability is consistent with the use of CReS, which can be maintained around one. It is concluded that the system is relatively reliable by using SReCSM encryption.

Through these simulation experiments, it is verified that SReCSM has a good user experience. It is also verified that the mechanism of SReCSM can effectively improve the efficiency of the cloud storage.When a mobile terminal makes a request, the optimal node is selected and then the time can be saved effectively. In the SReCSM presented in this paper, the re-encryption and decryption has the following characteristics:transport security and storage security of the user data are guaranteed.

V. CONCLUSIONS AND FUTURE WORK

The existing schemes such ascloud-based re-encryption scheme CReS and manager-based re-encryption scheme MReS re-encrypts the keyword to transmit safety. But these two schemes are more complex and need more time to re-

encrypt. In order to reduce the computational complexity, enhance the secure transmission,a new scheme SReCSM is proposed.SReCSM has high reliability proved through a series of simulation experiments. Turn around time and energy consumption of different size of files are compared and analyzed through these experiments. When the file size increases, proposed SReCSM can achieve accurate predictions, reduce the storage space requirement and the re-encrypting time. Finally, it is concluded that the SReCSM proposed in this paper has better security and reliability.

ACKNOWLEDGMENT

Foundation item: The Industrial research project of Science and Technology Department of Shaanxi Province(Grant No. 2016KTZDGY4-09); Laboratory fund of Xi'an Technological University (GSYSJ2017007); Research project on teaching reform of education in Shaanxi province (Grant No. 17JY015); Characteristic disciplines in Education department of Shaanxi province (Grant No. 080901); Research project on teaching reform of Xi 'an Technological University (Grant No. 17JGZ10); The principal fund of Xi'an Technological University (Grant No. XGYXJJ－0528).

REFERENCES

[1] Jung, Kye-Dong, Moon, Seok-Jae, Kim, Jin-Mook: 'Data access control method for multimedia content data sharing and security based onXMDR-DAI in mobile cloud storage'.Multimedia Tools and Applications, v 76, n 19, October 1, 2017, pp 19983-19999

[2] Chekam, T.T., Zhai, E., Li, Z., Cui, Y., Ren, K.: 'On the synchronization bottleneck of OpenStack Swift-like cloud storage systems'. In: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1–9

[3] Li, L., Li, D., Su, Z., Jin, L., Huang, G.: 'Performance analysis and framework optimization of open source cloud storage system'. China Commun. 13(6), 2016, pp. 110–122

[4] Iliadis, I., Sotnikov, D., Ta-Shma, P., Venkatesan, V.: 'Reliability of geo-replicated Cloud storage systems'. In: 2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing, 2014, pp. 169–179

[5] Han, Lin; Huang, Hao; Xie, Chang-Sheng: 'Multi-path data prefetching in mobile cloud storage'. In: Proceedings - 2014 International Conference on Cloud Computing and Big Data, CCBD 2014, March17, 2014, pp. 16-19

[6] Wang, Yan; Wang, Jinkuan: 'An Optimized Replica Distribution Method in Cloud Storage'. Journal of Control Science and Engineering, v 2017

[7] Chen, Ming-Hung; Tung, Yu-Chih; Hung, Shih-Hao; Lin, Kate Ching-Ju; Chou, Cheng-Fu: Availability Is Not Enough: Minimizing Joint Response Time in Peer-Assisted CloudStorage Systems. IEEE Systems Journal, v 10, n 4, December 2016, pp. 1424-1434

[8] Tysowski, P.K., Hasan, M.A.: 'Re-encryption-based keymanagement towards secure and scalable mobile applica-tions in clouds'. IACR Cryptology e Print Archive 668, 2011

[9] Purushothama, B.R; Shrinath, B.; Amberker, B.B. : 'Secure cloud storage service and limited proxy re-encryption for enforcing accesscontrol in public cloud'. International Journal of Information and Communication Technology, v5, n2, 2013, pp.167-186