Secure group collaboration using AES-256 with audit report to avoid repudiation In Google drive

Shivani S. Kale.
Department of Computer Science and Engineering
KIT's College of Engineering, Kolhapur, Maharashtra, India.

Abstract - With the advent of Web 2.0, end users generate and share more and more content. One such service in this context is the collaborative edition of online documents. This service is commonly provided through Cloud Computing as Software as a Service. However, the Cloud paradigm still requires users to place their trust in Cloud providers with regard to privacy. This is the case of Google Docs, a very popular service without privacy support for the documents stored on its servers. So here we discuss the issues and propose the add-on utility. The add-on utility guarantees privacy of shared documents in Google Docs as google docs is used as collaboration tool. It also generates the audit report to avoid repudiation while group collaboration.

Keywords-Repudiation, cloud, collaboration, Google Docs, Time key.

I.INTRODUCTION

Cloud solutions are scalable and ubiquitous, and follow a pay per use approach at all levels. One of the main barriers to the adoption of Cloud Computing is security. User data are stored on provider servers and there is no guarantee that this information will not be accessible to a third party. This can contravene legal requirements when the stored data are sensitive, as occurs in health care or banking environments. GDocs resource sharing service gives no guarantee that the documents will be safe and secure at the server side. Here we present a solution for secure online document sharing and secure resource sharing in group.

II.LITERATURE SURVEY

A number of different methods have been proposed to support secure sharing of GDocs. The recent scheme [3] provides solution to share and edit documents in the Cloud thus improving Google Docs. This offers the possibility of working with personal or shared documents using a public Cloud service, preventing access to third parties. But the scheme used to do the encryption or decryption of the documents while storing or sharing needs the password to be shared. Again in the scheme [3] the owner of the document is also allowed to do changes in the document. The available schemes explained above do not provide any support to reliable group communication as the GDocs is popularly used as collaboration tool. These schemes also bother to keep "n" number of password to share "n" number of documents with "n" users. As data is stored or shared with users, data is exposed to third party. So some of the researchers have used the following strategies to enhance the real time services.

Lilian Adkinson-Orellana et al. [4] have stated that in the new shared index, document must contain data related to the encryption of the shared document. This data will be created as a hidden file. When an owner shares a document, the new shared index willcontain the information associated with the encryption of the document copied from the general index. The content of this sharedindex is also ciphered using AES with a 128- bit key. The password to encrypt the index will be the shared key, which will preferably be different from the master key. Accordingly, when an owner shares a document he will simply have to give the shared key to the rest of the users. So that the master key will remain private and safe as shown in Fig1.

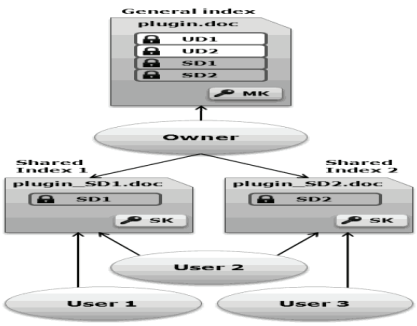


Figure 1: Private and shared indices of the owner of the documents

But the scheme can further be improved by removing the necessity of keeping both general index and shared index synchronized. Gabriele D'Angelo et al. [5] have proposed that in Content cloaking (CoClo) content is protected from unauthorized accesses, service providers and third parties. The transmission of clear text content is avoided when the web application does not support secure transmission protocols .With CoClo server has access to encrypted data only. This scheme supports AJAX based browser, some global add-on required to support the content cloaking. Daniel A. Rodríguez-Silva1 et al. [3] have presented a new security mechanism for SaaS applications of Google Docs service to have an additional privacy layer to protect their documents. This scheme needs the user"s password to encrypt the documents. So if the user forgets the password the information cannot be recovered. The present add-on application provides the possibility to share encrypted documents with other users. The only condition is that all users should have installed the Firefox add-on and knows the shared password.

III. LIMITATIONS

From the above survey we can identify limitations in GDocs are as mentioned below:

1. Password is shared.
2. Other than Owner writer is also allowed to modify the document
3. Operations are not performed on client side.
4. Group communication is unreliable.
5. N number of passwords is required for sharing "n" documents.
6. Service provider is able to see the data.

IV.PROPOSED WORK

In the proposed work, the user in the access control lists (acl) is allowed to share the document via an external storage service, with a desired group of other users. The proposal guarantees that only users in the specified group will be able to access the resources. The resources will remain confidential to all the other parties, including the service itself. The key used to protect a resource can be derived from a secret held by each user .A variation of Diffie-Hellman key agreement method and public tokens are used. The service offered is realized by users desiring to exchange confidential resources compared to existing applications. This approach offers stronger guarantee in terms of protection of resource confidentiality. The approach is fully compatible with the design of cloud storage applications as shown in figure (2). Figure 3 shows core components of the system

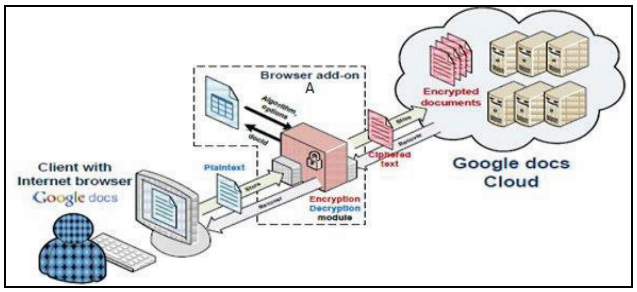


Figure 2: Secure sharing of documents in Google Docs using Add-on Utility

The proposed system (A) is designed in the following way:

1. Key Agreement Mechanism

In the key agreement method slight variation of Diffie-Hellman (DH) key Exchange agreement method is used. In this the two involved parties do not directly interact for computing the common secret but they interact with the external service.

Let (G, ・) be a public algebraic cyclic group of prime order q =| G | and ・ be the internal operation of the group with multiplicative notation. We assume that G is generated by an element $g \in Z_p$ (with p = 2q + 1 and p, q are two prime integers) in such that q=| G | and G= $\{g^e \bmod p: 0 \le e \le q-1\}$.

Each user u ∈ U chooses a secret integer parameter $e_u \in [0, q-1]$, computes the value $g^{e}{}_u \in G$, and inserts $g^{e}{}_u$ in a public catalog managed by the external service.

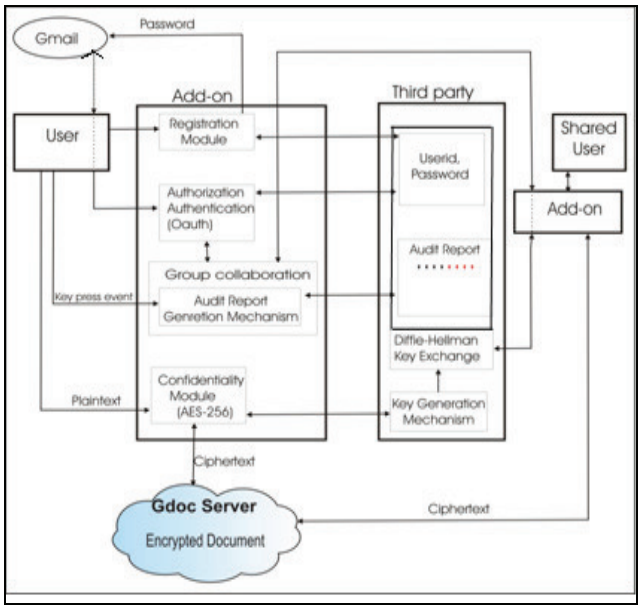


Figure 3: Core components of the system

2. Key agreement function

Whenever user *u* needs to share a common secret with user u_i, user *u* can efficiently compute such a secret by querying the public catalog to retrieve the public parameters g^{e}_{ui} and *q*, and by applying key agreement function. Derived keys are being used for secure sharing. Using key derivation method we will be computing key starting from the value of another key and a publicly available piece of information, called token.

Given a set K of keys and $k_i, k_j \in K$, a token $t_{i,j}$ between them is defined as $t_{i,j} = E_{ki}(k_j)$ as shown in fig 4(a) where E is a symmetric encryption function. Once the keys are derived, it will be used to encrypt documents using key assignment function from fig 4(b) that determines the keys used for encrypting document.

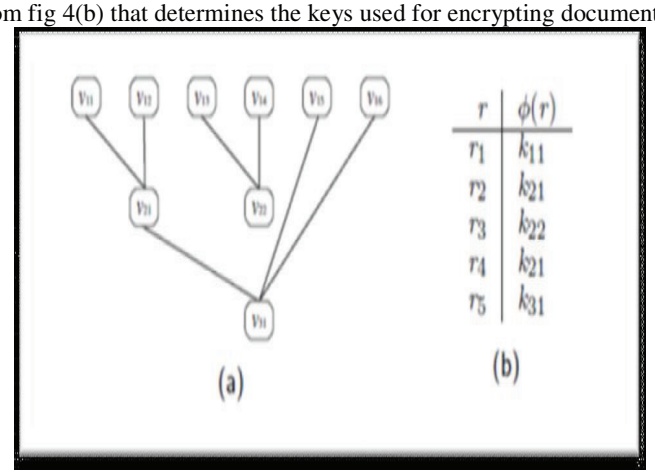


Figure 4: (a) An example of key and token graph (b) and key assignment function

V. AUTHENTICATION AND AUTHORIZATION ALGORITHM

The authentication functionality that allows data owners to compute the digest, signs (DSA), and correctly encrypts their document. In addition to this the functionality allows encrypted documents to be given to the third party service for their management. Each user is only able to first create and then use token chains whose starting points are the root vertices corresponding to keys that user can compute through modified version of Diffie-

Hellman computations. Therefore, whenever user needs to share a document with other users, the user must first encrypt document with a new key and then must add the appropriate tokens that other users in acl(r) can use to derive the new key. The authorization functionality will be implemented to allow users to retrieve the documents that are authorized. In particular, every time an authorized user *u* needs to access a resource *r*, the third party service has to deliver the encrypted resource to *u* along with a token chain ending to the vertex representing acl(r), which the user follows to derive the decryption key. User *u* can then decrypt the document and use the public Diffie-Hellman parameter of owner(r) for decryption of shared document.

VI. GENERATION OF TIME KEY AND AUDIT REPORT.

Repudiation is possible while online group collaboration in Google docs. This is avoided by keeping the records of the changes made by individual users to the gdocs document during collaboration as shown in figure 5.

During group collaboration users perform operations such as changing of data, deletion of data of the shared document. These operations on the document generates the key press, key down and key up events of java script which is sent by add-on as ajax request to Third Party (TP).The changes made to the document by users are shown with red color text in the audit report with the email-id of the user who made the changes to the document.

Audit reports are produced by TP by capturing the ajax request. The audit report lists the previous and changed information along with the name of the person. Any changes to the document can be tracked with the help of audit Report. Hence using audit report repudiation can be traced.

In this module the group and time key will be generated by server for secure group communication as we are using GDocs as the collaboration tool .This method is used basically in the concept of Broker architecture for Publish Subscribe scenario and we are using this for secure group communication for GoogleDocs to share documents securely. The following two server services are implemented to generate Group key and Time key.

1. Group key

The group key is generated according to the groups as shown in figure (6).

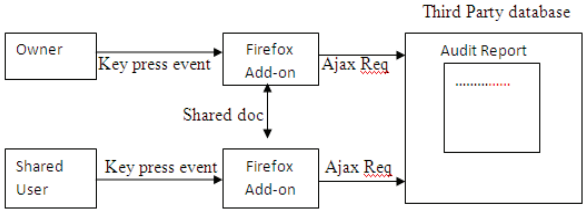


Figure 5: Audit Report Functionality

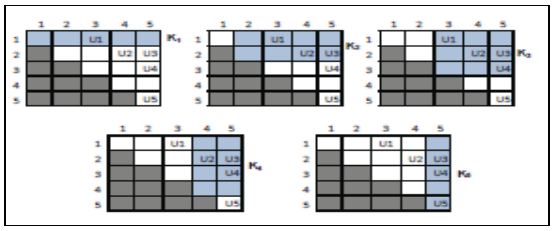


Figure 6: Example of Key group

2. Time Key:

While sharing the documents to unregistered users, unregistered users get automatically registered by add-on through TP. TP registers the user by sending the credentials of add-on to unregistered user's mailbox and one time password on his mobile. In case secure time constraint group collaboration session, OTP expiry can be set by owner, after that expiry no document will be shared with that user by owner. Generation of One time password is shown in figure 7.

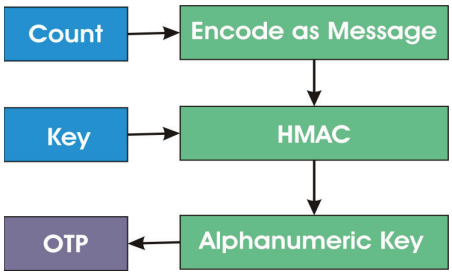


Figure 7: OTP generation mechanism

VII. EXPERIMENTAL RESULTS

The successful implementation of the Add-on utility's rigorous testing has given good results such as authentication, authorization, confidentiality, group collaboration and audit report.

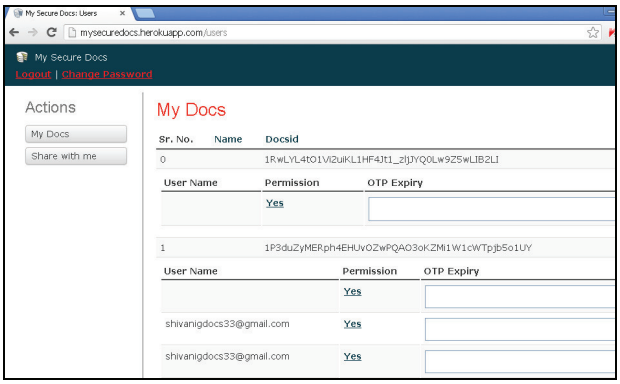


Figure 8: Third Party service.

As shown in figure 8 third party (http://mysecuredocs.herokuapp.com) keeps the record of documents shared by owner with users in "My Docs" however "Share with me" Keeps the record of documents shared by users with owner.

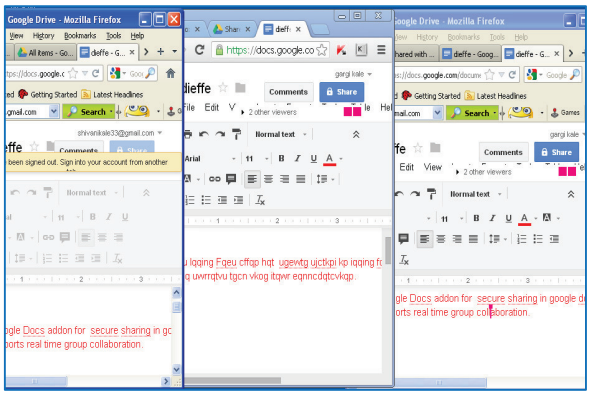


Figure 9: Secure sharing using Secure Google Docs add-on

Figure 9 depicts three windows. Window1 is of Firefox Browser with add-on installed. Window2 is of Chrome browser without add-on. The document of GDocs opened in window1 is in plain text which is shared with user in window2 where the contents are seen encrypted. Window 3 depicts the decrypted contents of shared user in Firefox browser.

VIII. TESTING OF POSSIBLE ATTACKS

1) Security provided

- Registration with Gmail (Authorization for Gdoc)
- Registration with Firefox Add-ons(Authorization for Add-on)
- Sending one time password on mobile phone.(Authentication of user)

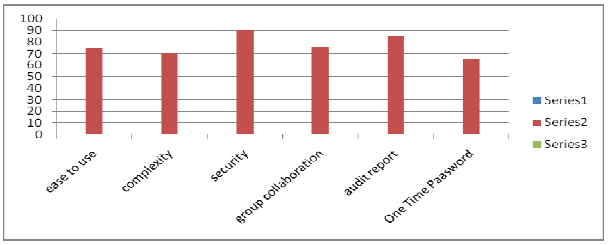
2) Resistance to various attacks like

The implemented add-on is resistant to various attacks as follows

- Anonymously login attack
- unauthorized access to document attack
- man in the middle attack
- Timing attack

3) Analysis on googleproductforums

The add-on implemented is posted on this site to use by users. Users have tried this add-on and posted their replies on the forum. The replies are summarized in the table shown in table1. Based on these replies graph is plotted as shown in graph 1.



Graph 1: Add-on analysis graph via Googleproductforums

IX. CONCLUSION

In today's world major use of Gdocs computing is for storage, online collaboration and data sharing. Storing data on Gdocs servers is not safe as they are readable by vendors. Attacker may get access to storage service of Gdocs and read the data. Attacker may steal the data while sharing data or while group collaboration. Vendors may compromise the confidential data to business rivals. These all reasons raise the question of confidentiality of the data shared/stored over the Gdocs.

The current work is a step taken towards answering these questions. The Add-on provided for secure data sharing has given good results. The testing of Add-on for secure data sharing, secure group collaboration with audit report functionality has been done successfully. The results obtained by storing data, sharing data on Gdocs and sharing with group of users are done by maintaining high confidentiality. By using the developed add-on, Gdocs user is assured about data privacy, confidentiality and secure group collaboration. Data is not only secured from other users of Gdocs but it is also protected from vendor of Gdocs.

Owner of document can get the audit report of the session to get the details. The audit report functionality helps to have data repudiation which may result while online group collaboration. The one time password feature provided by add-on gives time bounded group collaboration session to users. The sharing mechanism allows user to share documents in much secured way. So only owner of data and the shared users can access and use shared data. This gives assurance to sharing party about the confidentiality of data.

REFERENCES

[1] De Capitani di Vimercati, S. DTI,Univ. degli Studi di Milano, Crema, Italy Foresti, S.; Jajodia, S.; Paraboschi, S. Pelosi, G. ; Samarati, P. "Encryption based Policy Enforcement for Cloud Storage" in Distributed Computing Systems Workshops (ICDCSW), 2010 IEEE 30th International Conference on 21-25 June 2010, PP: 42 – 51.

[2] Tien-Dung Nguyen Dept. of Comput. Eng., Internet Comput. & Security Lab., Suwon, South Korea Eui- Nam Huh"An Efficient key MANAGEMENT FOR SECURE MULTICAST IN SENSOR CLOUD "Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on 23-25 May 2011 PP: 3 – 9.

[3] Lilian Adkinson-Orellana1, Daniel A. Rodríguez-Silva1, Felipe Gil-Castiñeira2, Juan C. Burguillo- Rial2. "Privacy for Google Docs: Implementing a Transparent Encryption Layer" www- gti.det.uvigo.es/~darguez/pub/2010_CloudViews_GoogleDocs.

[4] Lilian Adkinson-Orellana, Daniel A. Rodriguez-Silva, Francisco J. "Sharing Secure Documents in the Cloud. A Secure Layer for GoogleDocs" Proceedings of 1st International Conference on Cloud Computing and Services ... www-gti.det.uvigo.es/~darguez/publications1.html.

[5] Gabriele D'Angelo Fabio Vitali University Bologna Italy, "content cloaking: preserving privacy with Google Docs and other web applications" proceedings of the 2010 ACM symposium on Applied Computing .PP: 826-830, ACM New York.