Privacy for Google Docs: Implementing a Transparent Encryption Layer

Lilian Adkinson-Orellana[1], Daniel A. Rodríguez-Silva[1], Felipe Gil-Castiñeira[2], Juan C. Burguillo-Rial[2],

[1] GRADIANT, R&D Centre in Advanced Telecommunications,
Lagoas-Marcosende s/n, 36310, Vigo, Spain
{ladkinson, darguez}@gradiant.org

[2] Engineering Telematics Department, Universidade de Vigo,
C/ Maxwell, s/n, Campus Universitario de Vigo, 36310, Vigo, Spain
{xil, jrial}@det.uvigo.es

Abstract. Cloud Computing is emerging as a mainstream technology thanks to the provided cost savings in deployment, installation, configuration and maintenance. But not all is positive in this new scenario, user's (or company's) confidential information is now stored in servers possibly located in foreign countries and under the control of other companies acting as infrastructure providers; so its security and privacy can be compromised. This fact discourages companies and users to adopt new solutions implemented following the Cloud Computing paradigm. In this paper we propose a solution for this problem. We have conceived a new transparent user layer for Google Docs, and implemented it as a Firefox add-on, which encrypts the information before storing it on Google servers; making virtually impossible to get access to the information without the right password.

Keywords: cloud computing, google docs, security, privacy, firefox add-on.

1 Introduction

The continuous evolution of Information Technologies (IT) and the lower cost of servers and desktop PCs (which are becoming more and more powerful) is promoting the emerging of new IT services. Among them stands out the Cloud Computing (or simply "Cloud") paradigm. We can say that Cloud Computing has been born as the evolution and combination of several technologies, mainly: distributed computing [1], distributed storage [2] and virtualization [3]. Cloud Computing implies a change in the traditional paradigms basically because the infrastructure is completely hidden to the final user. In Cloud Computing we can find 3 levels or layers as shown in Fig. 1.

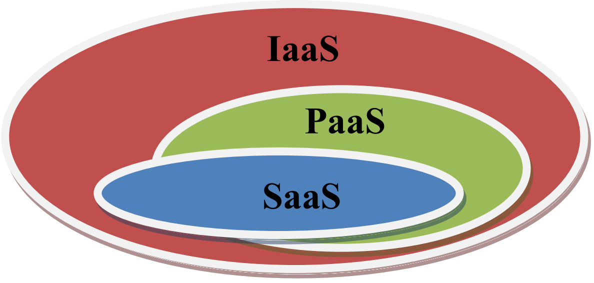


Fig. 1. Cloud Computing levels: Infrastructure, Platform and Software as a Service.

- IaaS (Infrastructure as a Service) is the lower level and includes infrastructure services, i.e. (virtual) machines used to run applications. We can find examples of IaaS in Amazon EC2, GoGrid or RackSpace.

- PaaS (Platform as a Service) is the next abstraction level. Here we can find a platform that allows developers to build applications following a specific API. Examples of PaaS are Google App Engine, Microsoft Azure or Sales Force.

- SaaS (Software as a Service) is the highest level and involves applications offered as a service that are executed in the Cloud (over a PaaS or a Iaas). Examples of SaaS are Google Apps, Salesforce or Zoho.

Some of the specific advantages of using cloud services are scalability, ubiquity, pay-per-use and no hardware/maintenance investment, however there exist some problems related with integration with current systems and especially with the security and the reliability on the service [4].

As well known examples of Cloud Computing applications (belonging SaaS level) we may cite Google's Gmail email client service or Google Docs, a web based editor for text documents and spreadsheets that offers its service to the users who have a Google account. This paper explains the development of an add-on for the Firefox browser [5], which allows users of Google Docs service to use a security layer to protect their documents in a transparent way.

This paper is organized as follows. Section 2 describes in a more deeply way the problematic of privacy in Cloud Computing, and also the services that offer possibilities for editing documents on the Cloud, giving particular emphasis on Google Docs. Section 3 describes the functionality and the internal structure of the presented add-on, as well as an example of the requirements and the behavior for users. Finally, section 4 gives some conclusions and raises possible future enhancements to extend the add-on functionality, explaining the current difficulties to implement them.

2 Privacy in Cloud Computing

2.1 Cloud Privacy

Concerning security in Cloud Computing paradigm, it does not only include aspects of confidentiality or privacy of the information, but it also could affect to the loss of data, although it is out of the scope of this paper. Since the processing of applications is moved to the cloud servers, sensitive data of users is exposed to the infrastructure provider. This means that users must trust in providers, nevertheless this is not always feasible, so some security mechanisms are required to solve the problem.

Particularly, it is especially critical the case when users store sensitive data remotely, because in the case the cloud servers, containing that information, suffered an attack; user's data would be compromised. For example in [6] it is explored information leakage in third-party clouds (Amazon EC2) and it is described how it is possible, under specific circumstances, to access the information of a cloud server (a virtual machine) from another different virtual machine both in the same physical server.

In the case of a service like Google Docs, the documents of the user are simply protected by the password associated to his Google account. If the session was not properly closed or his password was stolen, all the documents that were kept using this service would be exposed.

2.2 Document editing cloud solutions (SaaS)

Actually, there are many SaaS that offer the possibility of editing documents on the Cloud. In Table 1 some known solutions are compared taking into account their main features.

The table shows a representative set of solutions, most of them free, but there are much more web based editors that can be found with similar characteristics to the described in the table, mainly because Cloud software solutions are becoming more and more common. For example, OpenOffice offers an online version too, which is very similar to the desktop application, but it is still a beta version. Many people use this kind of software for free and this means that they have to be careful with the sensitive information they are storing in the Cloud.

Table 1. Comparison among different Cloud office applications.

	Maximum document size	Maximum storage	Price	Real time collaboration	Edit uploaded documents	Type documents
Google Docs	500K	1 GB	free	Yes	Yes	Text Spreadsheets Presentations
Zoho	-	1 GB	free	No	Yes	Text Spreadsheets Presentations
Microsoft Office Live	25MB	5 GB	free	No	No	Text Spreadsheets Presentations
ThinkFree	10 MB	1 GB	30 days trial	Yes	-	Text Spreadsheets Presentations
Feng Office	-	300MB	30 days trial		Yes	Text Spreadsheets Presentations
Adobe BuzzWord	10 MB	-	free	Yes	No	Text

The reason why we have chosen Google Docs is that it is a very popular and free service, with a complete and well-documented API. The use of this API simplifies the development of possible extensions. In addition it has an easy interface that allows users to made changes at real time on shared documents.

3 Security layer to protect Google Docs documents

3.1 Firefox add-on to protect Google Docs documents

The security layer we have implemented to add privacy to Google Docs documents relies on a Firefox add-on based on JavaScript [7] and XUL [8], a language similar to XML used to create Firefox extensions.

The add-on uses two hidden documents created using the Google Docs API, which will contain all the information needed to encrypt and decrypt user's information. One of the documents contains the data about the user's ciphered documents (algorithm used, password and encryption options if it was necessary). The other one maintains the same information, but only about the documents that are currently being shared.

When the add-on is enabled, it starts an asynchronous communication with Google Docs servers using the API, sending AJAX requests to authenticate the user and get data about all the documents owned by the user, their sharing permissions or the content of the documents. In this way, the add-on also gets access to the hidden documents described before, which will act as indices of the ciphered documents, whether there are shared or not.

Furthermore, as we can see in Fig. 2, when the process starts, it creates two channel listeners to capture all the data that it is sent to and received from the servers. When for example, a document is saved, the message with the data is intercepted, encrypting only the user's content and leaving the rest unmodified. A password chosen by the user is required in the encryption process, so nobody else could access the information of the document. Afterwards, the plaintext is replaced with the ciphered text, and the message is released, so it continues its way to the server. With this method, the user's information received by the server is indecipherable, but the server will not notice any difference because only the document's content is modified. It is also remarkable that every time a document is encrypted, the information related to the process is stored in the indices. If any condition of the ciphering changed, such as the document's password or the algorithm used, the indices would be automatically updated with the updated information.

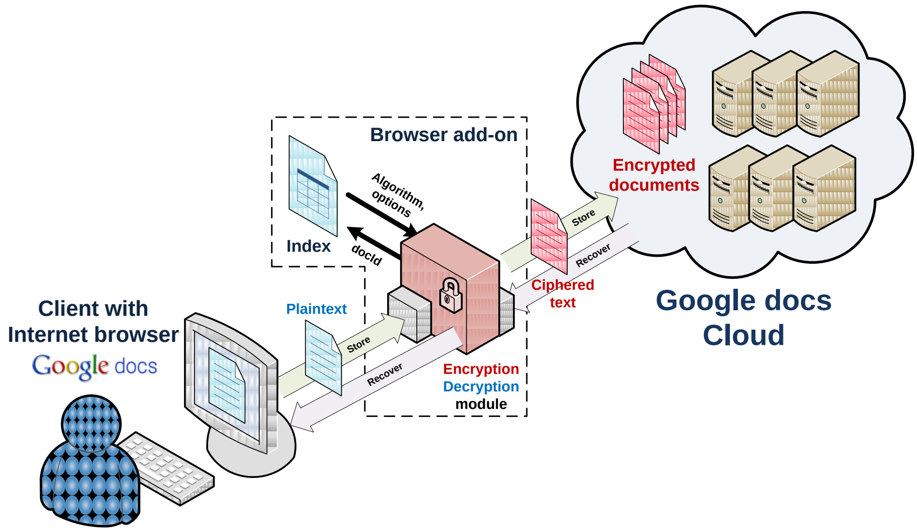


Fig. 2. Behavior of the add-on: components and actions involved in a secure editing.

When an encrypted document is requested, the same process is executed, but in the opposite way. After identifying the document, the hidden index is read and the associated information to the encrypted document is obtained. Then the ciphered data of the incoming message is accessed, decrypted with the information recovered from the index (algorithm, key size, mode…) and finally replaced with the plaintext. When the document is finally shown to the user, it is completely readable, and he can work with it as it was a normal one.

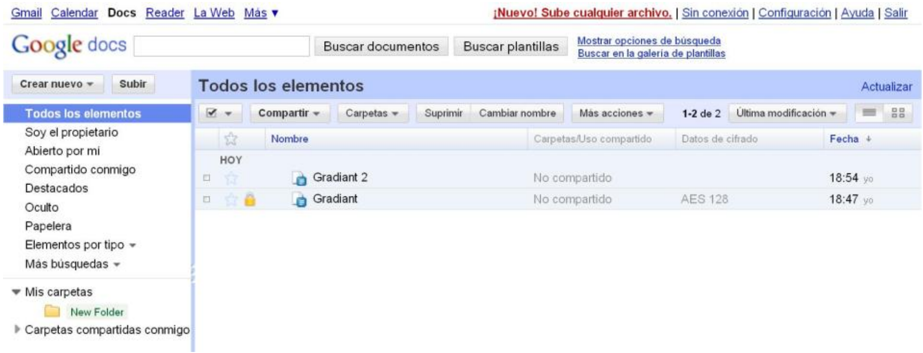
3.2 Using the secure Google Docs

In this section we will describe the functionality of the add-on, from the user's perspective.

The first step to be able to use the add-on is to install the .xpi file, which is a compressed file that follows the typical structure for a Firefox add-on. Once the add-on has been installed and the user accesses to Google Docs with his Google account (http://docs.google.com), it is necessary to activate the add-on by pressing a new button with a lock image that appears at the status bar or alternatively through the browser's tools menu.

Once it has been enabled, the main difference the user can find with respect to the normal use of Google Docs is that the index table with his/her documents contains more information; indicating which ones have been previously ciphered and which algorithms had been used in each case (see Fig. 3).

.



	Nombre	Carpetas/Uso compartido	Datos de cifrado	Fecha
HOY				
□ ☆	Gradiant 2	No compartido		18:54 yo
□ ☆ 🔒	Gradiant	No compartido	AES 128	18:47 yo

Fig. 3. Google Docs interface showing the index table of available documents

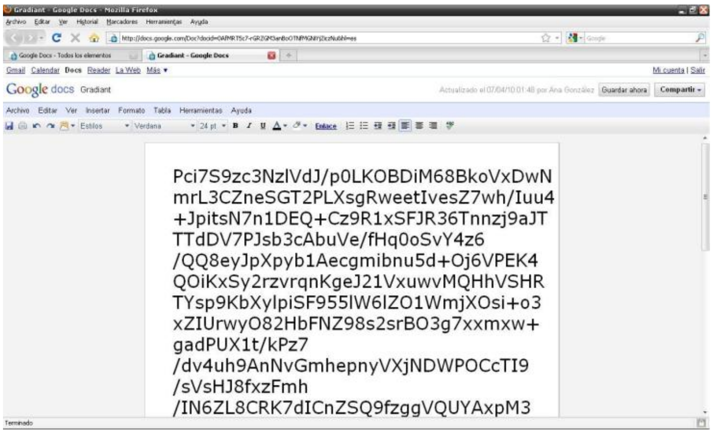
Supported algorithms are shown in Table 2 with their main properties. The user can choose any of them, and his/her election will influence on the security level and the speed of the encryption process [9]. The choice of the user's password is very important too, since the most secure environment could be compromised by the use of a weak password.

If the add-on is enabled, when the user is about to save the changes in a new document, or in one that had not been ciphered till that moment, a new popup window appears, asking the user for the password to cipher the information, and the possible encryption algorithms with their corresponding options (as the key size, or the mode if it was the case).

Table 2. List of supported encryption algorithms and its main features.

	Name	Block size	Key size	Security	Speed	Speed depends on key size?
AES	Advanced Encryption Standard	128 bits	128, 192, 256 bits	Secure	Fast	Yes
DES	Data Encryption Standard	64 bits	56 bits	Insecure	Slow	-
Triple DES	Triple Data Encryption Algorithm	64 bits	56-168 bits	Moderately secure	Very Slow	No
Blowfish	-	64 bits	32-448 bits	Moderately secure	Fast	No
RC4	Rivest Cipher 4	64 bits	8-2048 bits	Insecure	Very fast	No
TEA	Tiny Encryption Algorithm	64 bits	128 bits	Insecure	Fast	No
xxTEA	Corrected Block TEA	arbitrary, (min 64 bits)	128 bits	Moderately secure	Fast	No

After this step, the user will be able to work with the data as usual, being completely transparent the process of ciphering the data. If the add-on is disabled, and the user tries to access to his ciphered documents, the result will be unintelligible to him, as it can be observed in Fig. 4.


Gradiant - Google Docs - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
Gradiant - Google Docs
Gmail Calendar Docs Reader La Web Más ▾
Google docs Gradiant
Actualizado el 07/04/10 01:48 por Ana González
Guardar ahora
Compartir ▾
Mi cuenta | Salir
Archivo Editar Ver Insertar Formato Tabla Herramientas Ayuda
Estilos
Verdana
24 pt
Enlace
Terminado

Pci7S9zc3NzlVdJ/p0LKOBDiM68BkoVxDwN
mrL3CZneSGT2PLXsgRweetIvesZ7wh/Iuu4
+JpitsN7n1DEQ+Cz9R1xSFJR36Tnnzj9aJT
TTdDV7PJsb3cAbuVe/fHq0oSvY4z6
/QQ8eyJpXpyb1Aecgmibnu5d+Oj6VPEK4
QOiKxSy2rzvrqnKgeJ21VxuwvMQHhVSHR
TYsp9KbXylpiSF955lW6lZO1WmjXOsi+o3
xZIUrwyO82HbFNZ98s2srBO3g7xxmxw+
gadPUX1t/kPz7
/dv4uh9AnNvGmhepnyVXjNDWPOCcTI9
/sVsHJ8fxzFmh
/IN6ZL8CRK7dICnZSQ9fzggVQUYAxpM3

Fig. 4. User’s ciphered document opened without using the add-on

Once the parameters of the encryption have been set for a document, it is also possible to modify them. For example, changing the password or the algorithm that was used the last time the document was saved. If the user wants to eliminate the ciphering of a concrete document (deleting its password) he can do it as well.

4 Conclusion and future work

In this paper we have presented a new security mechanism for SaaS applications that brings the possibility to the users of Google Docs service to have an additional privacy layer to protect their documents on the cloud server side, with a very simple interface. Without the user's password used to encrypt the documents, the information cannot be recovered, even by the person concerned; so if the user forgets it, the data would not be readable.

This application is currently being improved with the possibility to share encrypted documents with other users, with the only condition that all users have installed the Firefox add-on and know the shared password.

As a future enhancement of the service, we are working in the usage of the same solution to manage the spreadsheets, but in this case some interesting problems arise: it is possible to encrypt the data of the spreadsheet, but the operations usually involved in this type of documents are not performed on client side, instead they are carried out on Cloud servers. So, it would be required a specific platform to allow processing encrypted operations with encrypted data [10,11], and this feature depends exclusively on the provider.

5 References

1. Mei-Ling Liu , "Computacion Distribuida. Fundamentos Y Aplicaciones", Ed. Pearson Educación, 2004
2. F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, R. E. Gruber, "Bigtable: A Distributed Storage System for Structured Data", Proceedings on 7th Symposium on Operating Systems Design and Implementation (OSDI'06), November 6-8, 2006, Seattle, WA (USA)
3. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. 2003. Xen and the art of virtualization. SIGOPS Oper. Syst. Rev. 37, 5 (Dec. 2003), 164-177
4. Tim O'Reilly, "The Fuss About Gmail and Privacy: Nine Reasons Why It's Bogus", http://oreillynet.com/pub/wlg/4707
5. Firefox add-ons website: https://addons.mozilla.org
6. T.Ristenpart,E.Tromer,H.Shacham,andS.Savage."Hey, you, get off of mycloud: exploring information leakage in third-party compute clouds, "InCCS'09:proceedings of, the 16th ACM conf. on Computer and comm. security, pages 199–212, NewYork, NY, USA, 2009
7. T. Negrino, D. Smith, "Javascript", Pearson – Prentice Hall, 5th ed., Madrid, 2005
8. Mozilla Development Center: XUL [Online]. Available: https://developer.mozilla.org/en/XUL. [Accessed: April 9, 2010]
9. A. A. Tamimi, "Performance Analysis of Data Encryption Algorithms". Available: www.cs.wustl.edu/~jain/cse567-06/ftp/encryption_perf.pdf [Accessed May 11, 2010]
10. Ernest F. Brickell, Yacov Yacobi. "On Privacy Homomorphisms (Extended Abstract)", Advances in Cryptology – EUROCRYPT'87, LNCS, Springer-Verlag 1987, pp. 117-125
11. Juan Ramón Troncoso-Pastoriza, Stefan Katzenbeisser, and Mehmet Celik. "Privacy preserving error resilient DNA searching through oblivious automata". In 14th ACM Conference on Computer and Communications Security, pages 519-528, Alexandria, Virginia, USA, October 29-November 2 2007. ACM Press.