Security requirements for data storage services on public clouds

Vitor Hugo Galhardo Moia and Marco Aurélio Amaral Henriques

***Abstract*— Nowadays, cloud computing is a consolidated technology and users are taking all their data to the cloud. Security and privacy are strongly required by users who are concerned about their data being exposed. Cryptography is one of the solutions to avoid unauthorized access to data, but with so many cloud service providers, can users trust them the safety of their data? How safe are they? In this paper we present an analysis of security features of some cloud service providers and point out some problems. Moreover, we propose a set of requirements for a privacy oriented cloud service and make recommendations to improve the security and privacy offered by new and existing providers.**

***Keywords*— Cloud Computing, Cloud Storage, Privacy, Security, Key management**

I. Introduction

In an era where privacy is becoming the main concern to many users, services involving cloud storage of sensitive data must be as secure as possible. Gradually the news reveal the increasing amount of threats that surround us every day on the Internet and end users are realizing the risks of storing their data on public clouds without a minimal level of protection. Techniques that can minimize the potential damages caused by unwilling disclosure of sensitive data are necessary and cryptography is being used to this end. Some cloud service providers (CSP) visualized a business opportunity in this field and started offering cloud storage solutions to minimize user's concern.

However, providing cryptography services is not trivial. There are obstacles to overcome and sometimes they are just put away by providers who prefer easy solutions or prefer to focus on usability over security. For users, the main obstacle in this environment might be their lack of expertise on cryptography. They do not know exactly what it is, how this technique can protect them and the importance of managing the cryptographic keys.

In this scenario, we present an analysis of some features offered by CSPs that could be harmful to users privacy and discuss the requirements that should be met by CSPs to address such problems. Finally, we make some recommendations that a CSP should adopt in order to improve the security and privacy offered to its users.

The remainder of this paper is structured as follows: Section II presents an analysis of security mechanisms of some cloud services. Section III describes the requirements that should be adopted for every CSP in order to improve their security and the user's privacy. And Section IV gives the conclusions.

Vitor Hugo Galhardo Moia and Marco Aurélio Amaral Henriques, Faculty of Electrical and Computer Engineering, University of Campinas, Campinas-SP, Brazil, E-mails: vhgmoia@dca.fee.unicamp.br, marco@dca.fee.unicamp.br. This work was partially supported by CNPq (153392/2014-2).

II. Analysis of cloud security mechanisms

In this section, we will present an analysis of cloud security mechanisms available nowadays.

Onedrive (Microsoft) [1] and Google Drive [2] are cloud service providers that offer data storage service for its customers for free, charging only for customized services. Both providers use TLS (Transport Layer Security) to protect the communication between the servers and user's device, so all the data sent to the servers are protected during transmission. Users can recover their password whenever necessary (there is a specific process for that) and can share files.

ownCloud [3] is a free and open source project that allows users to build their own private cloud. It has an application called Encryption App that handles all the cryptographic services. If users enable this app, all their data is encrypted on the server side. ownCloud allows users to choose if they want to enable the recovery password process or not. If so, all user's data is also encrypted with the administrator's keys, and in case the password is lost, the administrator can recover all user's data and create a new password for him. It is also possible to use a secure communication channel (TLS) and to share data with other users. The ownCloud service does not allow data deduplication on its servers [4].

SpiderOak [5] is a CSP that offers cryptographic services which are performed on client's side. Using a application provided to them, users can encrypt their data before sending to the cloud. There is a master key to wrap all the keys used to encrypt user's data. The master key is encrypted using the result of a PBKDF2 (Password Based Key Derivation Function) [6] applied to the user's password. The data's name (filename) is also encrypted. Users can share their files (paid version), but they are not allowed to recover their password when they lose it. If this happen, they lose access to all their data. A data deduplication process runs on the servers, but only on each user's account (single-user).

Wuala [7] [8] is a cloud service provider similar to SpiderOak. However, Wuala uses asymmetric cryptography to encrypt user's file keys and the deduplication process takes place in the servers on all user accounts (cross-user). Wuala does not have a free version.

Cyphertite [9] also uses similar cryptographic schemes to those of SpiderOak. The main differences between them are: Cyphertite does not encrypt filenames, it is an open source project and users can share files.

Credeon [10] and BoxCryptor [11] [12] are cryptographic modules intended to be used with a CSP, like Google Drive, Microsoft OneDrive and so on. All user's data is encrypted before it is sent to the cloud. While Credeon uses only symmetric encryption, does not allow password recovery, has no filename encryption and is not open source, BoxCryptor has the opposite characteristics: it uses asymmetric encryption, allows password recovery and filenames encryption and is an open source project. However, both do not enable deduplication and use PBKDF2 to derive keys from user's password.

Despite all security features adopted by these cloud services, some problems still remain. In the next section, we will present some requirements that these and other CSPs should adopt in order to improve the security and privacy of their users. Each CSP will be analysed with respect to each requirement.

III. Security requirements for data storage services

In this section we propose a small set of security requirements for data storage services on public clouds. For each requirement we present its characteristics and discuss how to make good use of them from a security point of view. We choose these requirements because we believe they are the most important as they have direct impact on security. Also, they can mitigate common attacks available nowadays, like those related to unauthorized disclosure of user's data, insecure applications, insider attacks, account hijacking, among others.

There are other important security requirements for cloud, like usability, and homomorphic encryption. However, we will not discuss them here due to lack of space and because we believe they require a deep investigation, which will be provided in a future work.

A. Cryptographic key life cycle control

There are some recommendations provided by NIST (National Institute of Standards and Technology) which all cryptographic systems should adopt. These recommendations are related to the management of cryptography keys, cryptoperiods and states of cryptography keys.

According to NIST [13], cryptoperiod is the time which a key is allowed for use, considering some factors like the estimated effective lifetime of the key algorithm, type and purpose of a key. The cryptographic keys go through some states during their life. There are about 6 possible states during the key life time: pre-activation state, active state, deactivated state, destroyed state, compromised state and destroyed compromised state. The key state changes in some specific events, such as the expiration of a cryptoperiod or the detection of a compromised key. More details about the states and all the possible transitions can be seen in ref. [13].

None of the CSPs presented in section II respect the cryptography key life cycle. They do not control the states of the keys as recommended by NIST and there is no cryptoperiods clearly defined for the keys. This fact may not seem so important now, but in the long term, it could compromise the security of the user's data and even the security of the whole system. We recommend that some kind of key life cycle control is implemented in a more privacy conscious system, even if this control is as simple as the one provided by a traditional public key infrastructure (PKI).

B. Security with Deduplication

According to Harnik et al. [14], deduplication is a technique that stores only a copy of a redundant data. Instead of making other copies, it creates links of that data in order to save disk space. There are two basic approaches to perform this technique: target-based and source-based. In the target-based approach, the client is unaware of the deduplication, because this process occurs on the CSP after users send their data. The CSP handles it and the main objective is to save disk space.

In source-based approach, the client application is responsible for the deduplication process and it takes place before the data is sent to the cloud. The objective of this approach is to save disk space and bandwidth. The application communicates with the cloud provider to verify the existence of an equal data on the cloud storage infrastructure. If so, the CSP just create a link of this data on the client's account and the process is concluded. On the order hand, if there is no equal data stored, the application sends the data to the cloud and stores it.

An important characteristic of deduplication is the search space when looking for equal data. If the CSP receives a request to verify if a certain data is already stored and look up only in the user's space, it is a single-user deduplication. But instead, if the CSP look up in all user spaces for an equal data, it is a cross-user deduplication. The latter method can save more disk space since a group of users could have the same file (like a famous music) but it is more vulnerable to an attack as explained below.

Harnik et al. [14] points out a vulnerability in the deduplication process: The identifying files attack. All providers that use sourced-based approach with cross-user deduplication mode are vulnerable. The attacker has a certain file and wants to know if someone else has the same one. He sends it to the cloud and keep monitoring the network to see if his file is uploaded or not to the cloud. If there is no equal file in the cloud, it will be uploaded. Otherwise, the file will not be sent and the attacker can conclude that someone had already uploaded the same file. One possible situation where this attack can be useful is the case where an authority wishes to verify if someone is storing a particular file in a provider. If so, this authority can demand the file owner identity from the CSP.

The CSPs that use deduplication are SpiderOak, Cyphertite and Wuala. The first two make use of single-user approach while the latter cross-user. We could not determine if the process executed by these three CSPs were target-based or source-based due to a lack of information. There is no information about deduplication on OneDrive, so we could not analyse it.

However, in order to avoid attacks as the one described and to obtain some benefits from deduplication, we recommend that this technique should be used on target-based approach. This will save disk space and protect the user's privacy.

C. Separation of Passwords

A common feature on Internet service that requires an authentication process is the password recovery. If users forget their passwords they can go through this process and get back their passwords or create new ones.

However, this might not be a good solution for CSPs with cryptographic services. Usually user's files are encrypted using keys based on their passwords. If an user forgets his password, it will not be possible to decrypt the keys and the user will lose access to all his data. But, if the CSP can recover user's password, it has access to user's keys somehow. In this case, the user is not the only one capable of accessing his files and he is indirectly sharing those files with the CSPs. This might not sound good for those users very concerned about privacy and they might not adhere to this service.

Separation of passwords is related to the separation of the authentication password from the one used to encrypt user's keys. In this way, users will have two passwords, one used to authenticate them with the CSPs, and another, unknown by the CSPs, used to encrypt user's keys, which is not stored anywhere and is used only on user's device.

Nevertheless, there might be cases where users work for companies which own their data. It could be interesting for those companies to be able to recover a key in cases where their staff is not available to give them access to the data stored securely.

The CSPs that allow password recovery are OneDrive, Drive, ownCloud and Boxcryptor (paid version). Boxcryptor has a customized version for companies that covers the example mentioned above.

The password recovery could be a good solution to release the users from the burden of losing a password but, on the other hand, it gives someone else the ability to access the data stored. Some users would rather have this option than losing all their data, while others not, specially those more concerned about their privacy. We think that the right answer could be simply letting the user choose to enable or not this feature and this is the approach we recommend with respect to password recovery.

D. High level of secrecy

The levels of secrecy are classified according to the amount of privacy they provide.

- **Level 1 - Communication channel encryption (CCE):** This level is related to the CSPs that use no cryptography beyond the encryption of the communication channel, using technologies like SSL/TLS (Secure Sockets Layer/Transport Layer Security). The client's application and the CSP create a tunnel where all the information/data that go through this tunnel is encrypted. The communication is also protected against tampering and sniffing attacks. It is important to clarify that this level only provides secrecy in the communication channel. The data will be sent to the CSP in encrypted form but it will be decrypted and stored in plain text when it arrives there. The CSP will have access to the user's data and it can use this ability to improve search mechanisms or advertising. Besides, it will be easier to do deduplication. Sending and receiving data in this case require four cryptographic operations (client encryption, server decryption, server encryption and client decryption). However, as there is no cryptography of data at rest, there is no overhead for users related to key management and the entire process is transparent for users.

- **Level 2 - CCE and server-side encryption:** This level corresponds to the application of cryptographic protocols by the CSPs to protect user's data while in transit and at rest. After receiving the data, the CSPs encrypt and store it in their servers. Figure 1 illustrates the basic scenario where data is sent to the cloud through a encrypted channel. The CSP generates a secret key used to encrypt user's data before storage. The secret key is wrapped with a public key and saved with the corresponding data. Figure 2 covers the recovery process, when the private key is used to decrypt the secret key that is used to open the user's data. Then the data is sent back to the user through a encrypted channel.

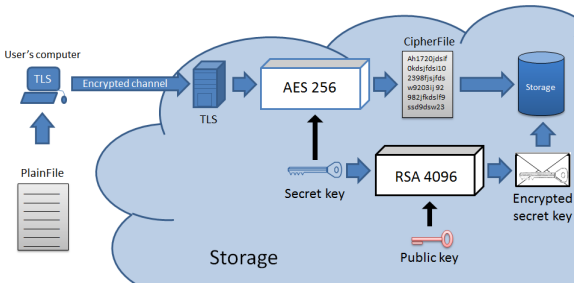

Fig. 1. Level 2 - Storage

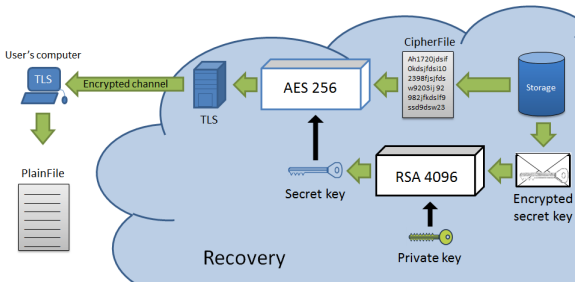

Fig. 2. Level 2 - Recovery

Most of the security stands at the CSPs. They are responsible for managing the cryptographic keys. As a result, they are able to access user's data as in Level 1, and, among other things, they can apply deduplication to the plaintext files. In this level, sending and receiving data require six cryptographic operations: Four, as in Level 1, plus two required in this level for encrypting when data arrives at CSP and decrypting when it leaves.

Some CSPs encrypt the secret key and/or the private key using user's password. This may be a good way to

keep all the keys safe, but it will make the recovery process harder. Level 2 also does not require that users worry about key management, as the CSP assumes this overhead.

- **Level 3 - Client-side encryption:** The most privacy friendly level is the one where data encryption is done in user's machine, before it leaves. Some CSPs provide applications for users to perform all the cryptographic operations locally. However, some users prefer to outsource these applications to trusted third parties. Then they send (and receive) only encrypted data to (from) the cloud. In this level, the process of sending data to the CSP does not require a secure communication channel as in Level 2 because the data is already encrypted. If an attacker can somehow get the data during the transmission phase, he won't be able to read it. In this level the CSPs are not capable to do any processing in the data. The deduplication process is not performed easily anymore. The user needs to take some specific measures, like encrypting the data using its own hash as a key, for example. In this case identical files will remain identical after encryption. The user's password can also be used to encrypt his keys, but it is safer to use a different password (encryption password) from the one used in the authentication process at the CSP (login password, which could be known somehow by the CSP - by storing it plain text for instance). However, with a encryption password unknown by the CSP, the user will lose access to all his data if he forgets this password, since key recovery process is not possible anymore. This level demands less cryptographic operations when sending and receiving data, requiring only two operations: data encryption on client when uploading and decryption on client when downloading. However, as a drawback, we point out the key management overhead that will be placed on users. This is an important concern because all the security stands on the cryptographic keys, and it could be a burden for some users to handle them. Figure 3 and 4 illustrate a basic scenario of storing/recovering files using a Level 3 architecture. Every file is encrypted using a different secret key that is wrapped with the owner's public key. The file and secret key (both encrypted) are stored in the cloud and the recovery process requires the download of both. First, the secret key is decrypted with the user's private key and then it is used to decrypt the file. The user must be aware that the data is unprotected while in his computer. This ends our explanation about Level 3.

All the CSPs discussed so far can be classified according to these levels. Microsoft OneDrive and Google Drive are Level 1 CSPs. ownCoud (with Encryption App) fits in Level 2. The rest of the CSPs (SpiderOak, Cyphertite, Wuala, Credeon and BoxCryptor) belongs to Level 3.

Analyzing these levels we can conclude that the most recommended from the privacy point of view is Level 3. We can justify this affirmative with the fact that the user is the only one who has access to his data. In terms of usability, Level 2 is the recommended as the CSP handles the cryptographic key management for the users. We believe that a better solution would be to use Level 3 with an application that would manage user's keys (only at client side) in an easy and almost transparent way, reducing the overhead on the user.

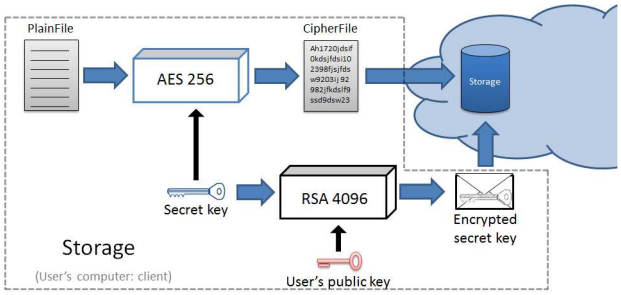

Fig. 3. Level 3 - Storage

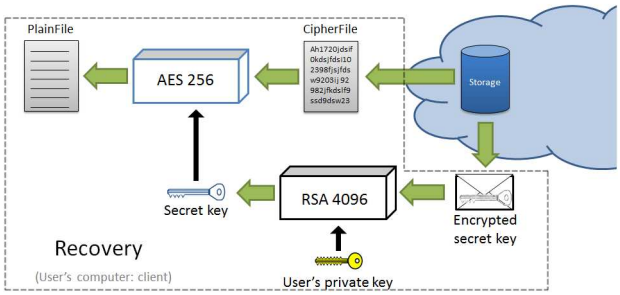

Fig. 4. Level 3 - Recovery

E. File attributes encryption

Another important security service that CSP should provide is file attributes encryption. Some attributes, especially filenames, are usually related to the file content, so, in plaintext form, they could help an attacker making a decision on which files are worth stealing. If cryptography is applied to the attributes, an attacker will have a hard time figuring out which file is interesting to him.

Just a few providers (SpiderOak, Wuala and Boxcryptor) offer this kind of service and they usually charge for it. The reason might be because this service requires a more complex management. However we recommend it for those requiring an extra level of privacy.

F. Source and executable code signatures

Normally the CSPs do not have their source codes open to the community for review. This fact may decrease the trust of users on the CSP because they have to believe that the codes are really secure. It's not enough for the community to have the CSPs word; it has to see the codes and how they really works. Besides, this could also be good for the CSPs, once the community could help them improve their codes, pointing out real and potential vulnerabilities.

However, having access to the source code may not be enough. It is necessary to have proofs that the running code is really the one generated by the open source code. In order to

give such guarantee, providers could use techniques like code signing.

The CSPs that have their source codes available for the community are: Cyphertite, ownCloud and Boxcryptor. All of them store their codes on the GitHub platform. However, none of them use techniques like code signing. We recommend the publication of source code and a strict process for both source and executable code signing and verification in order to provide a more secure cloud service.

G. Multi-factor authentication

Multi-factor authentication is the process that combines two or more authentication methods to verify someone's identity. Usually these methods use something the user know (password), something he has (cryptographic device) or even something he is (fingerprints, iris, voice etc.). Nowadays it is easy to find two factor authentication using login/password and mobile phones [15]. The login/password is the most common way of authentication and does not require extra equipment. Besides that, people are already familiar with it. However, as mobile phones became popular, they are being used as a second factor to improve the authentication process.

The same idea could be adopted to cloud storage. When cryptography is used to protect user's data, all the security could fall down if someone else get the user password. Some CSPs use user's password to derive a symmetric key to encrypt/decrypt user's private key. In this kind of solution, user credentials are the weakest link on the whole system. If an attacker could get these credentials, all user's data are compromised. Indeed this attack is not so hard these days, when users have to manage multiple passwords every day. There is a lot of services on the Internet that require an identity, which translates in a new login/password. With so many accounts, users tend to create passwords easier to remember, and also easier to attack. There are also other ways to get user's password, like using malwares or social engineering, but this subject is beyond the scope of this work.

Some of the CSPs addressed here do not offer better ways to authenticate users, relying on only the login/password method. Onedrive and Google Drive are the only ones that offer the possibility of a two-factor authentication, using user's mobile phone as an extra method.

In order to improve authentication and prevent an attacker from getting user's password and access all his data, we recommend to integrate at least two-factor authentication into the CSP architecture.

IV. Conclusion

In this paper we presented an analysis of some security features offered by CSPs and showed that they could be insufficient to guarantee user's privacy. We also presented some requirements that CSPs should adopt in order to improve security and privacy in their services, as for example, a classification level for CSPs according to where and how key management is done.

We found that the most popular CSPs have a lot of space for improvements with respect to the user's privacy point of view. Moreover, we found that key management may be a overhead too heavy for end users and this is probably one of the main reasons for some CSPs to take the burden of such management. However, none of the CSPs studied give the user a chance to choose which key management to use, and just a few use better ways to authenticate their users besides login and password. Moreover, none meet all the requirements proposed in this paper to improve security and privacy. There are still some requirements that are not fulfilled by anyone, like the key life cycle. Some of the providers have their codes available, but do not offer ways to prove that the application running in their servers is indeed the result of a compilation of the code made available.

As future works we point out an analysis of the implementation of two-factor authentication in some CSPs and the proposal of a protocol to simplify the management of key life cycle, allowing more end users to gain full control over their keys and, consequently, over their data stored on clouds. There is also a need to create an easier way to audit applications and prove their authentic relation comparing with some available code. Furthermore, we plan to study deeply other important security requirements as, for example, the usability, which is frequently left out from more secure systems, and homomorphic encryption, which can provide a completely new class of security services.

References

[1] Microsotf OneDrive [Accessed 2015 may 15] Available: https://onedrive.live.com/about/pt-br/

[2] Google Drive: Visão geral das conexões SSL [Accessed 2015 may 15] Available: https://support.google.com/a/answer/100181?hl=pt-BR

[3] B. Schießle, "Owncloud: Introduction to the new ownCloud Encryption App" [Accessed 2015 may 15] Available: http://blog.schiessle.org/2013/05/28/introduction-to-the-new-owncloud-encryption-app/

[4] Deduplication on Owncloud: Frequently Asked Questions [Accessed 2015 may 15] Available: https://owncloud.org/faq/#deduplication

[5] SpiderOak: Engineering. The Details Behind What We Do. [Accessed 2015 may 15] Available: https://spideroak.com/engineering_matters

[6] Password-Based Cryptography Specification: Version 2.0 [Accessed 2015 may 15] Available: https://www.ietf.org/rfc/rfc2898.txt

[7] D. Grolimund, L. Meisser, S. Schmid and R. Wattenhofer, "Cryptree: A Folder Tree Structure for Cryptographic File System", [Accessed 2015 may 15] Available: http://dcg.ethz.ch/publications/srds06.pdf

[8] L. Meisser, "Wuala Blog. Wuala's Encryption For Dummies", [Accessed 2015 may 15] Available: https://www.wuala.com/blog/2011/04/wualas-encryption-for-dummies.html

[9] Cyphertite: Cryptography [Accessed 2015 may 15] Available: https://www.cyphertite.com/papers/WP_Crypto.pdf

[10] Credeon Cloud Data Protection [Accessed 2015 may 15] Available: http://psg.hitachi-solutions.com/credeon/cloud-data-protection-overview

[11] Boxcryptor: Technical Overview, [Accessed 2015 may 15] Available: https://www.boxcryptor.com/en/technical-overview

[12] Boxcryptor: Issues on Server 2012 Deduplicated File System, [Accessed 2015 may 15] Available: https://forums.boxcryptor.com/topic/issues-on-server-2012-deduplicated-file-system

[13] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for key management - part 1: General (revised/).", in National Institute of Standards and Technology (NIST) special publication, [Online] Marc 2007, [Accessed 2015 may 15] Available: http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf

[14] D. Harnik, B. Pinkas and A. Shulman-Peleg, "Side Channels in Cloud Services: Deduplication in Cloud Storage", IEEE Computer and reliability societies, nov/dec 2010

[15] S. Lee, I. Ong, H.T. Lim, H.J. Lee, "Two factor authentication for cloud computing", International Journal of KIMICS, vol 8, pp. 427-432, aug 2010.