Development of Zero-Knowledge Cloud Encryption Gateway

Aram, Jivanyan
Skycryptor
Yerevan, Armenia
aram@skycryptor.com

Roland, Yeghiazaryan
Russian Armenian University
Yerevan, Armenia
roland.yeghiazaryan@gmail.com

ABSTRACT

Public cloud storages such as Box, Dropbox, Google Drive or OneDrive are used now in ubiquitous way for both personal and business purposes. They provide easy access to user’s files from anywhere and anytime, provide great features for collaboration and at last give the users a free storage. But on the other hand cloud storage providers gain full access to the users data. This is a very serious security issue and is an obstacle which discourages many individuals and businesses from using these services. The design of cloud encryption gateway which will allow the users to secure their data in clouds without compromising the clouds usability and convenience is a hard cryptographic and technical problem. In this paper we will review the existing solutions and will briefly introduce our own solution called Skycryptor which provides a perfect secrecy for users without compromising other advantages offered by cloud storage providers.

Keywords

cryptography, secure function evaluation, white-box, oblivious transfer

1. INTRODUCTION

Dropbox, Google Drive or any other public cloud storage provider take and can read all information the users store there. This is a very serious security issue [1] and for all individuals and organizations caring about their data security but still wanting to benefit from public cloud storages, the only solution is using some encryption tool which will help to encrypt user information before uploading it to the cloud. The design of advanced security solution for public cloud storages and for distributed file storages in general is a hard scientific and technical problem [2][3][4][5][6]. On the other hand, there is a tradeoff between security and usability, as encryption eliminates the easy access to data via search and also makes the sharing/ collaboration harder. Various special cloud encryption gateways had been emerged in recent years aiming to secure the users data in public cloud storages without compromising the sharing and collaboration features provided by the storage providers. In this paper we will review the main solutions existing in this domain showing what level of security is provided by each of them and their main advantages and disadvantages from both the security and usability points of view. Next we present our own cloud encryption gateway called Skycryptor which provides highest-level security for users and has certain competitive advantages over other solutions.

2. EXISTING CLOUD ENCRYPTION GATEWAYS

There are dozens of cloud encryption tools operating in the market. In this chapter we review the most widely used solutions.

2.1. Sookasa

Sookasa [7] is a new emerged cloud encryption gateway specially designed to help the companies from regulated industries and facilitate compliance with six federal standards such as HIPAA, FERPA, PCI DSS, GLBA, FINRA, SOX. It helps such organizations to store their data in cloud in encrypted form and also have full visibility on how the data was used or shared. However Sookasa does not provide a perfect secrecy as it handles all user secret key management on behalf of the users.

Sookasa secures the users files in Dropbox in the following manner:

1. Sookasa creates a special folder in user's Dropbox.
2. The user put sensitive files in that folder which are seamlessly encrypted with AES-256 encryption with unique file key randomly generated for that file.
3. Sookasa encrypts the file encryption key with the Sookasa's public key. The encrypted file key is stored at the beginning of the file.
4. The encrypted files are synced among all devices
5. When Alice shares some file with Bob and Bob wants to access Alice's encrypted file, Sookasa's server takes the file key encrypted with the server's public key, decrypts it and sends the file key to Bob.

As can be seen Sookasa owns all encryption keys used for securing the users files. This is a serious security drawback as the powerful adversary or Sookasa itself can always access the users sensitive files Such solution may satisfy specific companies but it cannot be a reliable security solution for companies which want to fully exclude the chance of their data appearing into third-parties hands.

2.2. nCryptedCloud

nCryptedCloud [8] is another cloud encryption gateway working with most cloud storage providers such as Dropbox, Google Drive, OneDrive and Egnyte. It provides a rich functionality of file/folder sharing and unlike Sookasa allows securing any file in any folder. However from the security point of view there is still a little difference between Sookasa and nCryptedCoud. The later provide perfect security for individual files meaning the user does not need to share the file encryption keys with nCryptedCloud as far as the file should not be shared with others users. But for securely collaborating on cloud files, the user again needs to share the file encryption keys with nCryptedCloud's server. The following examples highlight the main file storing and sharing functionality.

File encryption works as follows:

1. Alice creates a secure unique password for her file.
2. Alice encrypts the plaintext data using AES-256 Zip encryption by using the generated password.
3. Alice encrypts the file password with the her public key and stores the encrypted password in the encrypted Zip file among with the encrypted file.
4. When she wants to share the file with Bob, she encrypts the file password with nCryptedCloud’s server’s public key and sends it to server as well.
5. When Alice wants to open the encrypted file, she just takes the encrypted password from the zipped file and decrypts it with her private key. Next he decrypts the AES encrypted file with that password.
6. Bob receives the shared file and when he needs to access the files on her machine, nCryptedCloud verifies that Bob has access to the file key and distributes it to him. Bob stores the received key on his local key store, so for further accesses he does not need the nCryptedCoud to distribute him the file key.

Again the main drawback of nCryptedCloud is the fact that it can learn the secret keys and/or passwords used for file encryption. Although they claim that they never can access the cloud encrypted files, theoretically they can do it having the cloud storage access token for each user as well as the secret keys generated by user for securing data.

2.3. BoxCryptor

Boxcryptor is the only cloud encryption gateway among the existing solutions providing a zero-knowledge service to users. Its secure key management is based on asymmetric RSA cryptosystem and all files are encrypted with AES-256 block cipher. Each user has own private and public keys. The file encryption procedure works as follows

1. Create a secure random file key.
2. Encrypt the file using the file key.
3. Encrypt the file key with the user's public key.
4. Store the encrypted file key next to the encrypted data in the encrypted file.
5. If file is shared among many users, encrypt the file key with each user’s public key and append it to the encrypted file.

The main drawback of Boxcryptor is the fact, that if multiple users have access to a file, the file key is encrypted multiple times with different user public keys and each result is stored in the encrypted file. This forces the user to re-encrypt each file with different file key every time the group of people having access to the file is changed. Also the file size is growing linearly with number of people having access to it as for each new user having access to that file a new ciphertext should be stored at the beginning of the file.

3. SKYCRYPTOR

Skycryptor is a novel cloud encryption gateway which goal is to provide zero-knowledge security to users by preserving the main advantages that cloud storage providers have to offer. Its key management technique is based on so called proxy re-encryption scheme[10]0[11][12], in which context the Skycryptor service acts as a semi-trusted proxy server responsible for keeping proxy re-encryption keys and re-encrypting the file encryption keys upon authorized access request. Each user has public/private key pair specific to the proxy public key encryption algorithm. The file encryption works as follows:

1. For each file Alice generates a random file encryption key and encrypts the file with AES-256 CBC mode encryption.
2. Alice encrypts the file encryption key with her public key using proxy public key encryption algorithm. The encrypted file key is appended to the encrypted file and is stored in the cloud.
3. When Alice wants to share the encrypted file with Bob, he creates a special proxy re-encryption key and stores it in skycryptor servers. Alice also creates permission for Bob allowing him to access the file.
4. When Bob wants to decrypt the Alice’s shared file, he takes the file key encrypted with Alice’s public key from the cloud-stored file and sends it to Skycryptor service. Skycryptor checks the permission and re-encrypts the ciphertext with help of the proxy re-encryption key so the result is already the file key encrypted with Bob’s public key. The result is sent to Bob.
5. Bob receives the encrypted file key, decrypts it with own private key and reveals the key, which can be used finally to decrypt the encrypted file.

As can be seen, skycryptor server never learns the file keys. The proxy re-encryption allows re-encrypting the ciphertext without decrypting them. This powerful technique allows building an efficient and privacy-preserving cloud encryption gateway.

The next fundamental advantage of skycryptor is that it provides search functionality over encrypted data based on proprietary searchable encryption algorithm [13][14][15]. The Skycryptor's client application, which is responsible for file encryption/decryption functionality, also builds a secure (encrypted) index on users' files and uploads it to Skycryptor's server. The user can search in his encrypted files with web or mobile interfaces. The user's query is encrypted again before it is sent to server.

4. CONCLUSION

Public cloud storages thoroughly changed the way we work and collaborate together on digital data. But this should not be done at the price of compromised personal or corporate privacy. In this paper we have shown what solutions can be applied to ensure privacy in insecure clouds by representing also our solution Skycryptor which is specially designed to transform public cloud storages to privately secure environments without compromising their usability and convenience.

REFERENCES

[1] Mather, T., Kumaraswamy, S., Latif, S. (2009). Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance. O'Reilly media.

[2] K. E. Fu, Group sharing and random access in cryptographic storage file systems, Master's thesis, MIT, 1999.

[3] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh, SiRiUS: Securing Remote Untrusted Storage, in Proceedings of NDSS, no.0121481, ISOC. Geneva.

[4] Harrington, A., Jensen, C. Cryptographic access control in a distributed file system.In Proceedings of 8th ACM Symposium on Access Control Models and Technologies.

[5] Fu, K. Integrity and access control in untrusted content distribution networks. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.2005.

[6] Kallahalla, M., Riedel, E., Swaminathah, R., Wang, Q., AND Fu, K.Plutus: scalable secure file sharing on untrusted storage. USENIX,2003.

[7] http://www.sookasa.com

[8] https://www.ncryptedcloud.com

[9] http://www.boxcryptor.com

[10] K. B. Giuseppe Ateniese and S. Hohenberger. Key-private proxy re-encryption. In CT-RSA, 2009.

[11] M. Green and G. Ateniese. Identity-based proxy re-encryption. In J. Katz and M. Yung, editors, ACNS, volume 4521 of Lecture Notes in Computer Science, 2007

[12] Matt Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. InProceedings of Eurocrypt '98, volume 1403, pages 127–144, 1998.

[13] S. Kamara, C. Papamanthou, and T. Roeder. Dynamic searchable symmetric encryption. In T. Yu, G. Danezis, and V. D. Gligor, editors, ACM CCS 12, pages 965–976, Raleigh, NC, USA,

[14] S. Kamara and C. Papamanthou. Parallel and dynamic searchable symmetric encryption. In A.-R. Sadeghi, editor, FC 2013, volume 7859 of LNCS, pages 258–274, Okinawa, Japan, Apr. 1–5, 2013. Springer, Berlin, Germany.

[15] P. van Liesdonk, S. Sedghi, J. Doumen, P. H. Hartel, and W. Jonker. Computationally efficient searchable symmetric encryption. In Proc. Workshop on Secure Data Management (SDM), pages 87–100, 2010