Mathematical Problems of Computer Science 44, 101--108, 2015.

# Implementation Aspects of Search Functionality Over Encrypted Cloud Data

Aram H. Jivanyan[1] and Mihran M. Hovsepyan[2]

[1]Onecryptor CJSC
[2]Russian-Armenian(Slavonic) University of Armenia
e-mail: aram@skycryptor.com, hovsepyan.mihran@gmail.com

**Abstract**

Searchable encryption allows the user to store his data in untrusted environment such as public cloud storages in encrypted form but still be able to access the data via search. Meantime preventing the storage provider to learn either the data or even the search queries. The importance of such functionality raised with the wide adoption of public cloud storages such as Dropbox or Google Drive and this discipline gained high attention from research community. However, there is no practical application of searchable encryption functionality in industry. In this paper we introduce a novel cloud encryption gateway the goal of which is to protect users data in Dropbox and Google Drive without compromising the usability of those services and particularly providing search functionality over the encrypted data.

**Keywords:** Searchable encryption, Public cloud storage, Cloud encryption gateway, Skycryptor.

## 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]--[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 and 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. It is important that none of the existing cloud encryption gateways provides search functionality over encrypted data. Next we present our own cloud encryption gateway called Skycryptor which provides a highest-level security for users and implements search functionality over encrypted data.

## 2. 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 puts sensitive files in that folder which are seamlessly encrypted with AES-256 encryption with a 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 in the 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 provides 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 other 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 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 she 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 in the non-corporate case 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 a different file key every time the group of people having access to the file is changed. Also the file size is growing linearly with a 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.

## 2.4. Skycryptor

Skycryptor is a novel cloud encryption gateway the goal of which 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 a 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 the 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, she 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 the 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 his 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[13],[14],[15] based on proprietary searchable encryption algorithm.

## 3. Searchable Encryption Implementation Aspects

Providing a zero-knowledge search functionality over the encrypted data means that the host server which keeps the user's data and/or the searchable index, should not learn any information about the file content at any moment of its operation. This means that the files, searchable indices or even search queries should never appear on servers in the plaintext form. Among these security requirments the following issues should be considered while implementing a secure indexing and search functionality for large-scale application.

- The user can add files from different devices.
- The user should be able to search from any of his devices or web application.
- The user should have access only to his files.

The searchable encryption algorithm used by Skycryptor requires several AES-256 keys to be employed during the secure index construction. Each user has a specific searchable encryption master key and all the other keys required for index construction or search are generated from the master searcable encryption key. As the user should be able to make search from both web service and client applications, the master key should be accessible on all these platforms. But

from the other hand it should never be revealed at server side. The master key generation and recovery is done as follows:

1. When the user first logins via some client application, he generates the searchable encryption master key among the other encryption keys .
2. The user encrypts the searchable encryption master key with his password-key. The password-key is an AES-256 key generated from the user's password with the help of key stretching algorithm PBKDF2, which operates with HMAC-256 and 50000 iterations and the username is taken as a salt value.

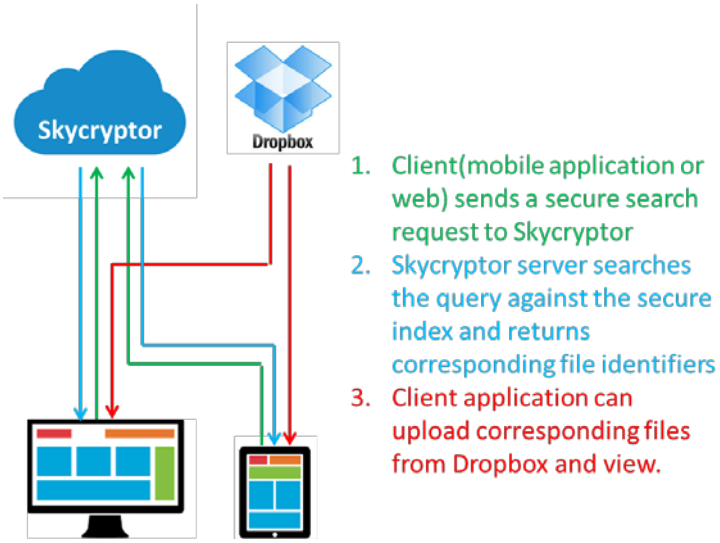


Fig. 1. Secure Search Request.

3. The encrypted searchable encryption master key is stored in Skycryptor servers.
4. When the user logins to web portal, his password-key is generated at client site in browser. When the user logins via client application, the password-key is generated at user device with the client application.
5. After succesfull authentication the user gets his encrypted searchable encryption key which can be decrypted with the help of the password-key. All the other keys necessary for secure searching are then generated from the master key.
6. Having the searchable encryption keys recovered already the user can post search queries to the Skycryptor servers. The search request flow is depicted in Fig. 1.

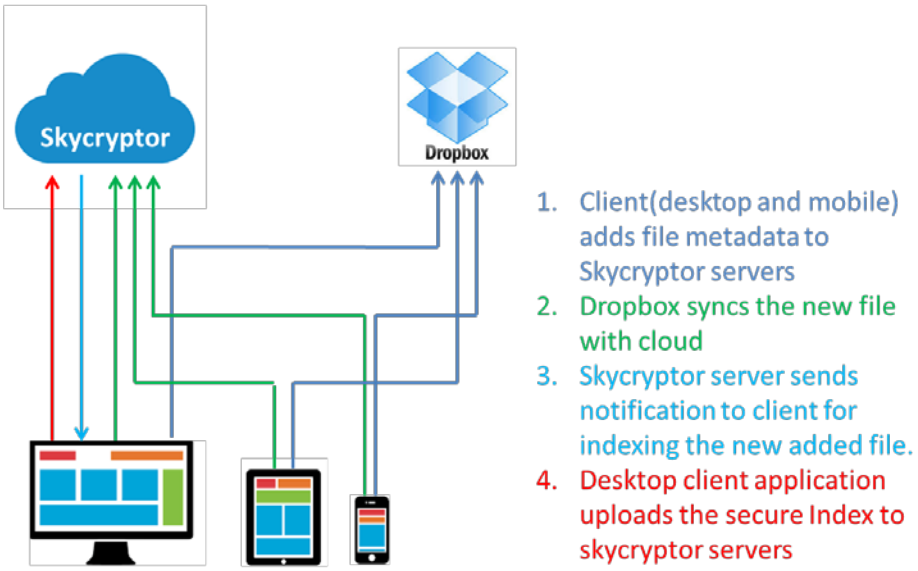


Fig. 2. Secure Index Generation.

The Skycryptor's client application, which is responsible for file encryption/decryption functionality, also allows to build a secure (encrypted) index on users' files and which will be uploaded to Skycryptor's server. The index generation flow is exposed in Fig. 2 and is comprised of the following steps:

1. When the user encrypts a new file, the file metadata such as file name and path is posted to Skycryptor server and the encrypted file is synced with the cloud service in parallel. Note that the user can add new files from both desktop and mobile devices.
2. The skycryptor server generates a new unique ID for each new added file and creates a special notification object which can prompt the user that the file should be indexed. The notification contains the file's local path as well as its unique ID.
3. The index file notifications can be processed only by the user desktop client applications. The desktop application time-by-time checks for new notificaitons and discovering a new notification starts processing it in the following way.
    a. The application decrypts and tokenizes the file content getting all unique words contained in the file.
    b. The file unique ID contained in the notification acts as the file unique identifier in the secure index. The client application indexes the new added file according to the specified secure searchable indexing algorithm. Secure index is an encrypted index comprised of the encrypted keywords and encrypted file identifier in a specific form.
    c. The secure index is uploaded to Skycryptor servers.

## 4. Conclusion

In this paper we covered the details of our cloud encryption gateway called Skycryptor and discussed the implementation aspects of secure search functionality over the encrypted files. To

the best of our knowledge this is the first attempt to apply searchable encryption functionality in real industry.

## References


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

[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, ISOC. Geneva, no.0121481, pp. 74--86, 2003.

[4] Harrington, C. Jensen, "Cryptographic access control in a distributed file system", In Proceedings of 8th ACM Symposium on Access Control Models and Technologies, pp. 158--165, 2003.

[5] K. Fu, *Integrity and Access Control in Untrusted Content Distribution Networks*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.2005.

[6] M. Kallahalla, E. Riedel, R. Swaminathah, Q. Wang and K. Fu, "Plutus: scalable secure file sharing on untrusted storage", In Proceedings of the 2nd USENIX Conference on File and Storage Technologies, pp. 29-42, 2003.

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

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

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

[10] K. B. Giuseppe Ateniese and S. Hohenberger, "Key-private proxy re-encryption", In CT-RSA '09 Proceedings of the Cryptographers' Track at the RSA Conference, pp. 279-294, 2009.

[11] M. Green and G. Ateniese, "Identity-based proxy re-encryption", ACNS, Proceedings of the 5th international conference on Applied Cryptography and Network Security" of Lecture Notes in Computer Science, vol. 4521, pp. 288-306, 2007.

[12] M. Blaze, G. Bleumer and M. Strauss, "Divertible protocols and atomic proxy cryptography", In Proceedings of Eurocrypt '98, vol. 1403, pp. 127–144, 1998.

[13] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption", *ACM CCS 12*, Raleigh, NC, USA, pp. 965–976, 2012.

[14] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption", *FC* Okinawa, Japan, *2013, LNCS, Springer, Berlin, Germany*, vol. 7859, pp. 258–274, 2013.

[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)*, pp. 87–100, 2010.


**Submitted 30.07.2015, accepted 23.11.2015**

# Ամպային պահոցներում պահվող գաղտնագրված տվյալներում որոնման գործողության իրականացման մանրամասների մասին

Ա. Ջիվանյան և Մ. Հովսեփյան

**Ամփոփում**

Որոնվող գաղտնագրման մեթոդները թույլ են տալիս օգտագործողին պահել իր տվյալները գաղտնագրված կերպով անվստահելի միջավայրում, ինչպիսիք են՝ ամպային բաց պահոցները, բայց միևնույն ժամանակ հնարավորություն են տալիս օգտագործողին կատարել անվտանգ որոնման գործողություն այդ գաղտնագրված տվյալներում: Ընդ որում, ամպային պահոցների սերվերները չեն կարող կարդալ ո՛չ օգտագործողի տվյալները, ո՛չ նույնիսկ որոնվող հարցումները: Նման ֆունկցիոնալության կարևորությունը աճել է ամպային պահոցների մասշտաբային կիրառմանը զուգահեռ, և գիտության այս ճյուղն արժանացել է մեծ ուշադրության հետազոտողների կողմից: Սակայն դեռևս չկա որոնվող գաղտնագրության որևէ ինդուստրիալ կիրառություն: Այս հոդվածում մենք ներկայացնում ենք ամպային պահոցների գաղտնագրման նոր համակարգ, որի նպատակն է պաշտպանել օգտագործողի տվյալները Dropbox-ում կամ Google Drive-ում առանց այդ ծառայությունների կիրառելիությունը բարդացնելու և, մասնավորապես, ապահովելու որոնման գործառույթը գաղտնագրված տվյալներում:

# О деталях имплементации поиска в зашифрованных данных в облачных хранилищах

А. Дживанян и М. Овсепян.

**Аннотация**

Шифрование допускающее поиск данных позволяет пользователю хранить свои данные в ненадежной среде, например, в общедоступном облачном хранилище, в зашифрованном виде, тем самым не ограничивая возможности поиска этих данных. При этом провайдер хранилища не имеет возможности читать данные, результаты поисковых запросов и даже сами запросы. Значимость такой функциональности растет в связи с ростом популярности облачных хранилищ, таких, как Dropbox и Google Drive, и параллельно с этим среди научного сообщества этой дисциплины растет интерес к изучению этого вопроса, однако на данный момент в индустрии не существует практического применения этой функциональности. В данной статье мы представляем новую систему шифрования пользовательских данных в облачных хранилищах Dropbox и Google Drive, сохраняя при этом все удобства использования этих служб, в частности возможность поиска по данным.