

BIG DATA PRIVACY PROTECTION USING CRYPTOGRAPHIC CLOUD STORAGE

Vidhya R¹, MadhuKiranPathange S², NikhilGrandhi³

¹ Assistant professor, Department of Computer Science, SRM University, Tamilnadu, India

^{2,3} Student, B Tech IV year, Department of Computer Science, SRM University, Tamilnadu, India
²pskiran86@gmail.com

Abstract: The world today generates a huge amount of data every day. It includes clickstream data, browsing history, social media activity, opinion polls etc. This massive amount of data, Big Data, is collected regularly by companies and governments. Big data now exists in almost every application relating to human behaviour and interactions and as such it becomes too vital to ensure its security and privacy. In the past, large amounts of data were hosted on proprietary infrastructure and private networks. Today, it is possible for any organization of any size to cheaply store and access Big Data via public cloud infrastructure.

Keywords: Big Data, Cloud storage, Searchable encryption, Data confidentiality

1. Introduction

With the development in cloud computing and storage technologies, users are increasingly storing and accessing their data on cheap public cloud servers. The benefits of public cloud storage is that users need not foot the costs of building and maintaining a private cloud infrastructure and can instead pay a cloud service provider an amount corresponding to its storage needs. The availability and reliability of the data stored at a public cloud server come at the cost of significant security and confidentiality-concerns. These concerns are the biggest obstacle to the adoption of cloud storage [1]. The solution to the above problem would be to develop a virtual private storage service which has the security and assurance of a private cloud and the availability and reliability of a public cloud. This can be achieved via the latest achievements in cryptographic techniques namely, searchable encryption [add references]. Through searchable encryption one can encrypt data at the client side,

send it to the server and then search over the encrypted data without revealing anything about the plaintext data. Searchable encryption is better than traditional encryption as it doesn't require - downloading of all the data and decrypting it in order to use it. It only retrieves the specific data which we require through a keyword search on the encrypted data. The keyword in question would also have to be encrypted in order to prevent leakage of information to the server. The purpose of this paper is to demonstrate a feasible solution to the problem of Big Data privacy on the cloud and to discuss a possible implementation of symmetric searchable encryption. In the following sections we will discuss the architecture of the cryptographic cloud followed by the working of the symmetric searchable encryption scheme.

2. Architecture of Cryptographic Cloud

The architecture would consist of three parties: Party A that stores its data on the cloud, Party B which needs to access the cloud data and the cloud service provider. Now consider the scenario where Party A stores and retrieves its data from the cloud. To do this it first generates a cryptographic key which is stored on A's local system. This key is kept secret and never revealed to the cloud service provider or Party B or anyone else. To upload data to the cloud, A will add appropriate metadata, encrypt the data and metadata with the searchable encryption scheme and finally upload the encrypted data to the cloud. Whenever A wants to download data it first generates a token. This token is created with the help of a search query and the cryptographic key stored in A. The token is then sent to the cloud service provider where it is used to search for the appropriate encrypted files and send them to the user. A then uses the cryptographic key to decrypt the files. Thus,

data is stored and retrieved without revealing anything about the data to the cloud. We can observe this scenario in Fig 1.

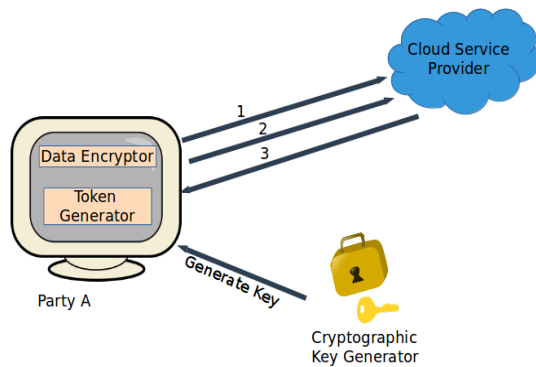


Figure 1. (1) Process the data, encrypt it and send it to the cloud, (2) Token is generated based on search query and sent to the cloud service provider, (3) Cloud searches for the appropriate encrypted files and sends them back to Party A.

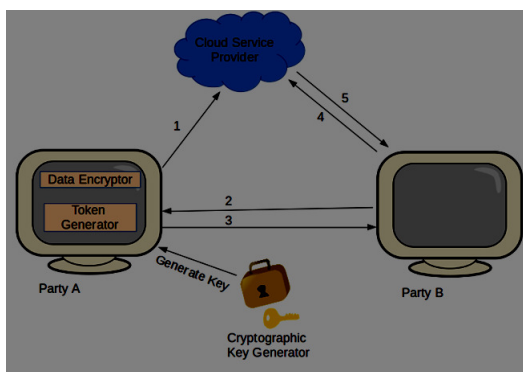


Figure 2. (1) Process the data, encrypt it and send it to the cloud, (2) Party B requests Party A to search for a keyword (3) A token and credential are generated and sent to Party B (4) Party B sent the token to the cloud (5) Cloud searches for the appropriate encrypted files and sends them back to Party B.

Now consider the case where Party B wants to access the data stored by A on the cloud. When Party B requests data, Party A generates a token and a credential for B. The token and the credential are then sent to Party B. Party B transfers the token to the cloud, which uses it to retrieve the encrypted files and sends them to B. These files are then decrypted by B using its credential. Thus, data is

anything about the data to the confidentially shared between the two parties through the cloud. We can observe this scenario in Fig. [2].

3. Symmetric Searchable Encryption

Symmetric Searchable Encryption (SSE) is an encryption scheme which enables one to search over encrypted data using tokens over an encrypted search index. This helps us in creating our cryptographic cloud as it is only through the ability to search that we are freed from having to download and decrypt the entire cloud data in order to obtain any particular file.

Consider a search index for a collection of files which maps keywords to the files which contain them. In a searchable encryption scheme, the search index is encrypted such that one can make use of it only with the help of tokens or trapdoors. A token generated for a keyword retrieves from the search index all the encrypted files which contain that keyword. Moreover, one can create a token only with the secret key which was used to create the search index. This ensures that only users with the knowledge of the secret key can access the search index and retrieve files.

Despite all the safety guarantees the searchable encryption scheme provides information may be leaked to the cloud provider. This can happen if the cloud provider analyses the search and retrieval pattern of the user. The cloud provider could then make a guess at what keywords are being used and what all documents have those keywords in common. But this leakage of information is because of the retrieval of files and not because the failure of the encryption scheme and is inevitable. One way to prevent this would be to introduce false positives and then sort them out at the user's machine. However this leads to more complexity. SSE schemes were first introduced in [3]. They were subsequently improved and given new security definitions in [2, 4, and 5].

The Symmetric Searchable Encryption Scheme consists of the following steps:-

Key Generation

Key word Index creation

Encryption

Search

Decryption

Adding Documents

- 1) **Key Generation:** Two symmetric keys K_1 and K_2 are generated and stored at the local machine of the user. They are used in encrypting the search index contents and the documents respectively. As they are symmetric keys, the same key is used for both encryption and decryption.

Table 1. Document Content IDs

Document	Content ID
D1	1
D2	2
D3	3
D4	4
D5	5
D6	6

2) **Keyword Index Creation:** In this step the Keyword to Document index is created. Let D be the set of all documents we wish to upload to the cloud. First, assign each document in D a unique content id. After that decide upon the keywords to be used and build an index which maps the documents' content ids to the corresponding keywords they contain as in Table 2. Using this index we create an inverted index which maps the keywords to the corresponding content ids of the files. The inverted index is shown in Table 3.

Table 2. Keyword index

Content ID	Keyword
1	w1,w3,w5
2	w2,w3,w7
3	w1,w4,w9
4	w6,w8,w9
5	w8,w10
6	w10

3) **Encryption:** The keyword index and the documents are encrypted using the keys generated in the Key Generation step. The key K_1 is used to construct the encrypted keyword index. The keywords are encrypted as $ENC_{K1}(w_i || n)$, where

ENC_{K1} represents encryption with key K_1 , represents concatenation, w_i is the i^{th} keyword in the keyword index created in Keyword Index Creation step, and n is the n^{th} time of occurrence of the keyword w_i in D . These encrypted keywords are then mapped to their corresponding content id of the document the keyword is located in. This is shown in Table 3. The key K_2 is used to encrypt the documents in D . The encrypted documents are renamed to their corresponding content id. The encrypted documents and index are then sent to the cloud.

4) **Search:** In this step we create a search token to retrieve the files from the cloud. It takes a keyword as input and outputs the token as a list of $ENC_{K1}(w_i || i)$, where ENC_{K1} represents encryption with key K_1 , w_i is the keyword, $||$ is concatenation and $0 < i \leq n$, and n is the total number of documents. This token is then sent to the cloud where it uses this list against the encrypted index to retrieve all the corresponding encrypted files. These files are then sent to the user who requested them.

5) **Decryption:** The encrypted files that the user receives are decrypted using the key K_2 . This is represented as $DEC_{K2}(D_i)$ which represents decryption of document D_i with key K_2 .

Table 3 Inverted Index

Keyword	Content ID
W1	1,3
W2	2
W3	1,2
W4	3
W5	1
W6	4
W7	2
W8	4,5
W9	3,4
W10	5,6

6) **Adding documents:** With regards to adding more documents to D , instead of adding them directly to the existing index we add them as a separate group say document group G . We perform on G the same

operations as we did on D and add its index to the cloud. We can then search both D and G on the cloud.

4. Conclusion

This paper shows how the SSE scheme can be used to guarantee privacy on a public cloud. We have discussed the architecture of the cryptographic cloud and also the working of SSE. This can be used to fulfil the security of many a cloud based service and assure the users of the confidentiality of their data. Such a scheme enables users to build a secure storage over the infrastructure of the cloud provider.

Cryptographic cloud storage ensures that the storage provider learns nothing about the data, does not tamper with the data, keeps it available at all times to any authorized device, provides a public cloud service's advantage of backing up data, retrieves the data efficiently and enables sharing of the data. As technology keeps growing so does the amount of data used by it. Storage of this data in a private cloud is secure but infeasible, storage in a public cloud is feasible but insecure. Cryptographic cloud provides a happy marriage of the best features of both private and public clouds.

5. Future Works

The construction of the SSE discussed in the paper deals only with text documents. It can be enhanced to deal with other formats such as spreadsheets, images, videos, etc. This would require extracting meaningful keywords from these files in order to index them. In this manner the cloud can be enabled to handle multiple formats and thus boost its usability.

The cloud service provider despite not being able to read the encrypted data may attempt to tamper it or add files of its own to the user's data. This causes loss or corruption of the user's data which the cloud service provider can easily escape blame by accusing the fault to lie with the user. This can be prevented with the help of integrity checks. Integrity checks allow users to detect whether changes have been made to the data. This can be done without having to retrieve the data.

6. References

- [1] S. Kamara and K. Lauter, "Cryptographic cloud storage," Financial Cryptography and Data Security, 2010.
- [2] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In A. Juels, R. Wright, and S. De Capitani di Vimercati, editors, ACM Conference on Computer and Communications Security (CCS '06), pages 79–88. ACM, 2006.
- [3] D. Song, D. Wagner, and A. Perrig. Practical techniques for searching on encrypted data. In IEEE Symposium on Research in Security and Privacy, pages 44–55. IEEE Computer Society, 2000.
- [4] E-J. Goh. Secure indexes. Technical Report 2003/216, IACR ePrint Cryptography Archive, 2003. See <http://eprint.iacr.org/2003/216>.
- [5] Y. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In J. Ioannidis, A. Keromytis, and M. Yung, editors, Applied Cryptography and Network Security (ACNS '05), volume 3531 of Lecture Notes in Computer Science, pages 442–455. Springer, 2005.
- [6] Jayasri .J and Logeshwari . N, ARM Based Electronic Notice Board all the way through Zigbee with Room Lights Control via PIR Sensor, International Innovative Research Journal of Engineering and Technology, pp.14-17, 2016.
- [7] MdIftekhar Salam , Wei-Chuen Yau, Ji-Jian Chin , Swee-Huay Heng, Huo-Chong Ling, Raphael C-W Phan , Geong Sen Poh, Syh-Yuan Tan and Wun-She Yap. Implementation of searchable symmetric encryption for privacy-preserving keyword search on cloud storage. Human-centric Computing and Information Sciences 5(1), 1–16 (2015)
- [8] Ateniese, Giuseppe, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, and Dawn Song. "Provable data possession at untrusted stores." In Proceedings of the 14th ACM conference on Computer and communications security, pp. 598-609. AcM, 2007.

