# Securing Shared Data in Cloud Computing by using Cryptographic Schemes

Ibtissam Ennajjar
LIROSA Laboratory
Abdelmalek Essaadi University
Tetuan, Morocco

Youness Tabii
LIROSA Laboratory
Abdelmalek Essaadi University
Tetuan, Morocco

Abdelhamid Benkaddour
LIROSA Laboratory
Abdelmalek Essaadi University
Tetuan, Morocco

## ABSTRACT
Sharing data using cloud computing can be a good solution to profit from reduced costs, scalability, flexibility and more advantages that user can get from the cloud infrastructure. But when the object shared consists of sensitive or personnel data, the security concerns increase. Problems related to the lack of data confidentiality and integrity represent the number one problem encountered in cloud infrastructure and that restrains users and a number of organizations to benefit from the cloud's services. In this paper we discuss the requirements needed to secure data in cloud environment and we propose a new approach to enhance the security of the data shared in cloud storage. Our method is based on cryptographic models. It is confidential, flexible and it reduces time of computation by adopting simple and efficient key management process and encryption schemes.

## General Terms
Security, Cloud Computing, Shared Data.

## Keywords
Cloud Computing, Security, Data Sharing, Cryptography, Cloud Storage.

## 1. INTRODUCTION
Nowadays, it is hardly possible to companies or particulars to imagine their lives far from the Internet. We live in the era of the big data, the internet of things and the vision of anything as a service that have the purpose to facilitate more and more our life. At the technical level, the origin of this new generation is a cloud computing architecture. Cloud computing brings many benefits for organizations what makes it a good opportunity to improve IT productivity and to allow business revolution. By using cloud services, companies can benefit from good maintenance, reduced costs, efficient infrastructure, and a number of software products and many services to do work correctly. Nevertheless, they must absolutely care about their data. Using any device connected to the Internet and sharing data through it means that we lose control over our data and other parties can manipulate it and use it for their benefit. This can be more critical if this data is confidential and needs to be secure. As we cannot trust the cloud provider, thinking about methods to secure data seems to be crucial. In this paper we will concentrate on the security of data shared using cloud storage; the situation when we have some data that we need to share with a group using cloud services. This utility is very used by enterprises and also by individuals due to the high storage capacity of the cloud and the performance of its material. So, how can we be sure that our shared data would not be viewed by unauthorized parties including the cloud provider? One trivial approach to solve the problem is encryption. But, is it really a good method? Let us first see what happened in traditional public encryption scenario. And see if it fits with cloud's client needs. As we know in public key encryption we need a couple of public key and private key. The first one is used for the encryption process and the second for the decryption one. In case of a group of users as in cloud, data owner must encrypt its data many times for each user. So he needs multiple couples of public and private keys corresponding to each authorized user. This method has many downsides for example: the hardness of keys management; the number of computation because that data will be encrypted many times as much of users; also if the data owner wants to change in data, a decryption of all copies of users is required and an encryption will be done again with the modification, what makes traditional encryption a big obstacle in cloud environment [3]. As it can be seen, handling the sharing of data with several users is difficult and needs more flexibility and efficiency in terms of handling access control, key management, the encryption mechanism and the decryption process contrary to two-party communication or when data are related to one user[2].

In light of these remarks, we look for a method to share data in cloud in purpose to be accessed by a group of authorized users excepting the cloud provider and malicious users. This method should guaranty confidentiality and having a powerful access control system without complexity in key management strategy or in computations. Such a method would make allowance for the following facts: first we concur that the method has two main pillars the confidentiality of data and the efficiency of the access control model that will help to assure this confidentiality. At the confidentiality level, we need to think about the conception of encryption/decryption mechanisms in the way to minimize computations and complexity whilst ensuring an efficient security and key management design. For the access control model, it must be scalable, fine grained, achieve the user accountability and handle the user revocation, furthermore it has to be a collusion resistant. The model should give the data owner the possibility to indicate a group of users that are allowed to view his or her data; no-one, other than the data owner and the members of the group, should gain access to the data, including the cloud service provider. The data owner should be able to add new users to the group and the system must be scalable as cloud it is. Furthermore, the data owner should be able to revoke access rights against any member of the group over his or her shared data. No member of the group should be allowed to revoke rights or join new users to the group [1]. The work we done consists of a new approach where we include most of recommendations needed to secure data shared in cloud.

Our preeminent contributions, as reported in this paper, are as follows:

- The suggested methodology provides the confidentiality of the data by adopting symmetric encryption and key partitioning distribution.
- The secure data sharing over the cloud among the group of users is assured without the elliptic curve or bilinear Diffie-Hellman problem cryptographic re-encryption.
- The proposed approach gives a secure and flexible personal data sharing scheme in cloud computing. The approach maintains the flexibility of encrypting data using a specified access policy and a user set which is a list of selected authorized users. Thus, only the user who is in the list can access the data.
- The framework achieves efficient user revocation by dropping user from the user list without moving to re-encrypting data or changing keys.

The remainder of this paper is organized as follows: Section II discusses related works; Section III introduces the system model, security requirements and the details of our scheme; and Section IV concludes the paper.

## 2. RELATED WORK

The existing works having the aim to secure the data sharing in untrusted environments like cloud computing, concentrate on securing the access to the data by providing a scalable and efficient access control model. The fundamental insight of those productions is to give the data holder the right to encrypt its data and outsource it to the cloud and give the access to it by constructing a key management following an access strategy [12].

Ciphertext policy attribute based encryption (CP-ABE) has become a widespread method studied to be incorporated in cloud area to secure data sharing. It is based on the principle of verifying the access to data by ensuring that user's attributes correspond to the access policy proposed by data owner. This policy is included in the ciphertext before to be sent to the cloud storage and the attributes are enclosed in the private key given to the user. In this way, ciphertext can't be deciphered unless the user's attributes fit the policy [7]. Although the CP-ABE is considered as an improved version of Attribute Based Encryption scheme that seems suitable to be applied in cloud environment, many drawbacks and limits hinder its full utilization like user revocation problem; time computation cost; difficulties in the use of bilinear pairing and so on. A number of researchers had been drilling down into the application of CP-ABE in cloud and they had tried to improve its design in the purpose to reduce its limits.

Jing-yi et al [3] proposed to use CP-ABE mechanism but they combine it with a broadcast encryption (BE) scheme to handle the user revocation problem encountered in CPABE. By this combination, they handle access control and user revocation by putting indexes set where each authorized user is affected to an index. That way, only users who are indexed in the set can access data and when a user quit a dropping of the index solve the problem without generating new keys or re-encrypting data. They also tried to minimize the time consuming of decryption algorithm at the receiver side by partial decryption. As the policy of CPABE is exposed while merging with ciphertext, it can represent a threat if a party know conditions to decipher data, so they suggest obfuscating the access policy of ciphertext and client's attributes. This approach is an amelioration to access based broadcast encryption proposed in [4].

Song Lingwei et al [6] suggested a method to solve the attribute revocation problem present in CP-ABE by allowing distinctly the cancellation of fine grained attributes related to a revoked user, while maintaining the collusion attack feature of CPABE and other characteristics like: flexibility, scalability, and confidentiality of the access control system in untrusted cloud. They used a combination of three techniques CPABE scheme, linear secret sharing schemes and counter mode encryption (CTR). In this scheme the data owner divides his data into several pieces and encrypts them using CTR symmetric encryption, then he applies CP-ABE principle on the content keys of his encrypted data and according to the user's attributes data can be decrypted or not. Besides of the application of CP-ABE reasoning, the user can't decrypt data if its attributes reside in the revocation list proposed by the authors.

S. Kumar [8] et al proposed to use a hash function associated with the asymmetric encryption when using attribute based encryption algorithm to make this algorithm more useful in practice and in real time systems.

X. Dong et al [11] introduced an effective, scalable and privacy-preserving data sharing service for cloud computing environment. This service is based on the combination of ciphertext policy attribute based encryption and identity based encryption (IBE) techniques. The method applies fine grained access control, full collusion resistance and backward secrecy to protect the cloud data from being accessed by inner intruder, including the cloud and from external attackers and unauthorized outer users. The scheme is based on a bilinear mapping as in CPABE where each file is described by a set of attributes and an access tree converted to a linear secret sharing scheme matrix. This method is also used by A. Balu et al in [13]. But the particularity in [11] is that authorized users are recognized by a user list containing users' IDs what includes the utility of IBE technique in this approach. The scheme reposes on four algorithms: system initialization; encryption; key generation and decryption. The scheme does not disclose any attribute of users to the cloud what keeps the privacy of the users away from the cloud.

M. Ali et al [2] proposed a method to secure data sharing avoiding the use of data re-encryption in the cloud side. The methodology consists of a process to design the file's upload; download and update mechanisms as well as the access control's procedure. They propose to use a symmetric key encryption by applying AES algorithm in contrary to the use of bilinear pairing or el Gamal cryptosystem where there is more computation and can be complex in practice. They also use a hash function to generate a key randomly. And a hash based message authentication code (HMAC) to guaranty the file integrity. The key management is handled by keeping the key hidden by partitioning it and a secure overwriting is adopted to delete the key from storage. They suggest using access control lists (ACL) to manage authorized users and to control user leaving or arrival. The mechanisms used in this approach ensure time consumption reduction. However, the proposed methodology is based on the trust on a third party what reminds the threats caused by malicious insiders.

D.Tiwari et al present in [9], a framework and a methodology to share data in cloud using proxy re-encryption based on elliptic curve discrete logarithm problem (ECDLP). By this proposition, they ensure the integrity and the protection of data privacy by cancelling the reliance of data security management on the cloud provider. They introduce trusted proxy agent that acts as a trusted intermediary between the contributing parties and conserves the privacy of the data

holder. The use of this proxy reduces computation cost and facilitates the security management. The protocol is built around many phases: the initialization of the system by generating public and private keys pairs of participating parties using an elliptic curve cryptography; the encryption of the data by the data holder ; the generation of a metadata that includes error detection code and message identification code before storing the data in cloud server; the request of data by user to the trusted proxy agent; the re-encryption of data by the trusted proxy agent to be understandable by user and the decryption of data by user. All those operations are cared out under an authentication protocol based on signatures given by the data originator.

G.Wei et al proposed in [10] a protocol for sharing data in cloud that gives flexibility, data confidentiality and data sharer's anonymity without requiring any fully trusted party. They proposed a proxy re-encryption approach free of pairing, anonymous and unidirectional. Their system is composed of three parties: data holder; data sharer and cloud provider. The re-encryption process is done at the cloud side after the distribution of keys by the data holder and before the retrieval of the data by the cloud sharer. They first encrypt the data using symmetric key encryption and then they encrypt the keys of symmetric key encryption by proxy re-encryption. That way, the data originator can have a flexible sharing of data with other clients via semi-trusted cloud servers. The proposed proxy re-encryption scheme is accompanied by anonymity characteristic and doesn't require the time consuming operation like bilinear pairing used in most works to secure data sharing in cloud.

# 3. OUR PROPOSED FRAMEWORK TO SECURE DATA SHARING IN CLOUD

In this section, we present the system model of our approach. We give an overview of the entities that make up our design. Then we detail how the model works and we describe its design.

## 3.1 Entities

Our system model as shown in figure 1 requires four entities:

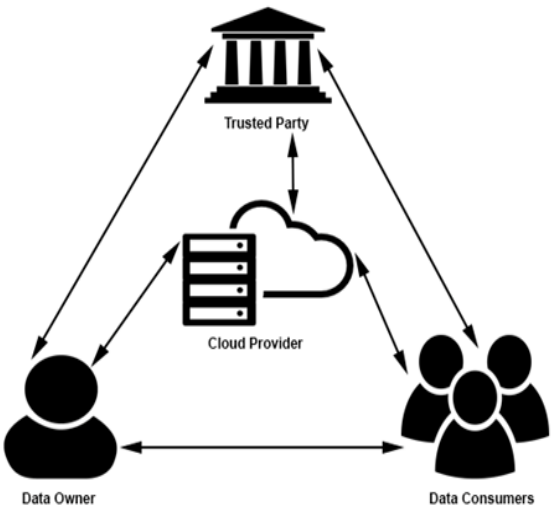


**Fig 1: The system model of our approach**

Data Holder (DH): is the owner of the data who needs to store it in cloud; his aim is to share this data with specific users by exploiting the beneficial aspects of cloud infrastructure like maintenance and low cost. He can be an enterprise or an individual customer who stores his data in a set of cloud servers. He interacts with cloud provider to have the right to access the cloud servers. He asks of maintaining confidentiality and access control over its data once moved from his border. Also he must indicate authorized users of its data by specifying an access control strategy.

Cloud Provider (CP): our study is done over a cloud environment. The cloud provider is a main entity; he offers to the DH a storage service to share its data with extensive space and high computation power. CP mustn't see the data in plain-text form or manipulate it. He just involves in its upload and downloads operations. The CP servers have to be always online to give access to DH and users to data.

Trusted Party (TP): is a trusted third party who is responsible of the security management. It helps the data owner to maintain the security of data. It is just charged by distributing public keys to the data owner also it produces and delivers private keys to the data consumers. This party is also vulnerable and data must be kept hidden from it.

Data Consumer (DC): symbolizes each user of cloud asking to use the shared data. DC must be known by the DH or has a manner to characterize him in purpose to create an access policy that distinguishes authorized users. The DH has the right to give access to DC or to deny it and according to the decision of the DH, DC can decrypt data using his private keys and use it.

## 3.2 System Design

In this section, we introduce the design of our suggested approach. We propose various cryptographic key processes that allow our methodology to attain the confidentiality and security target.

Before uploading its data, the DH should classify it according to the level of security it needs because uploaded data does not always demands high security, and encrypting it using complex schemes and processes will just be a waste of costs and computations. So the DH has to choose first the type of the security he needs: High level, Medium level or Low level. The first one is appropriate to the data the owner wants to be highly secure like: financial transactions, medical documents, secret files of organizations, etc. the second level is for the data with medium sensitive degree like personal files, videos, pictures, documents, etc. and the latest is for general data like photos, videos, etc; data that can be shown in public and everyone can see and to which there are no highly restricted access except authentication; data that we don't consider very interesting to secure immensely. In accordance to the choice of level given by the DH, a set of operations will be executed.

We consider that there is a good coverage of classical network security protocols and secure communication channel, which restrict eavesdropping or producing of any communication by our model adversaries. Building upon the concrete realization of communication between partners, such protocols could include Virtual Private Networks (VPN) between parties, Transport layer Security (TLS), Secure Socket Layer (SSL), Internet Protocol Security (IPSec) channels and HTTPS for Web-based information exchange.

In order to supply a verification process for data integrity and authenticity, we propose the use of different cryptographic Hash-based Message Codes (HMACs) signatures on every encrypted file.

Any client program or service should be authenticated in purpose to prevent unauthorized third parties from taking part in the system simply by adopting a false identity.

In our design we concentrate on handling the storing of sensitive data that needs more confidentiality.

### *3.2.1 Data Upload:*

The upload process consists of two operations: data encryption and data storing.

When the DH needs to share sensitive data with a specific group through cloud storage, he must interact with a TP to make this data confidential. This relation is required because of the complexity of keys generation, management and computation when using cryptographic schemes. The DH sends a request of encryption to the TP. The demand is attached with the file (O) to be stored and a list of users that are authorized to see the original data. Users can only read data or have the right to read and write. The list sent to the TP is used to produce the access control list (ACL) for the file by the TP. When DH sends a new file, a new list of group should be attached to it. If the group already exists, the DH sends just a group index. After receiving the file and the list of group, the TP develops the ACL and generates the group of users. Each ACL has an index, an owner id, a list of authorized users ids and other descriptive attributes.

Later, the TP creates a key K for encryption scheme. K is a random secret produced by TP for each file. The length of K is as recommended for symmetric key cryptosystem is 256 bits. We choose to generate a random number of 256 bits and apply a hash algorithm to it, what generates K to be used for symmetric encryption to secure the data. Next, we propose to divide this key and generate a pair of sub-keys corresponding to each user. One portion will be sent to the user and the second still at the TP center relied to the user index. So, the decryption of the data needs the collation of the two keys which correspond to the secret key K. we give the relation between the two portions as: $K=k1 \oplus k2$.

Using the method above TP encrypts the data O utilizing K. Then he produces two partitions of K; k1 the TP portion and k2 the DC portion, then he removes K. There is a different couple of k1 and k2 for every user. The k1 and k2 for each user is inserted into the ACL for decryption use. Subsequently the TP sends the encrypted file C, the group id and the appropriate k1 to the DH. And he sends the group id, the k2 to listed users in the ACL using public keys of the users. The data owner after receiving the encrypted file, he uploads it to the cloud provider. The key k is dropped from the TP after encryption.

When a new member joins the group a key generation process will be activated but just for the new user. No change will be done for other users' keys or for data.

### *3.2.2 Data download*

When a data customer needs to use the stored file in the cloud, he sends a request to the TP via the cloud. The cloud will verify the identity of the user via an authentication service and sends a request to the TP to verify its authorization referring to the ACL. If the user exists in the list of corresponding group of the requested file, the TP asks for the key portion k2. Then he calculates K by applying exclusive OR operation over k2 and the corresponding k1 from the ACL.

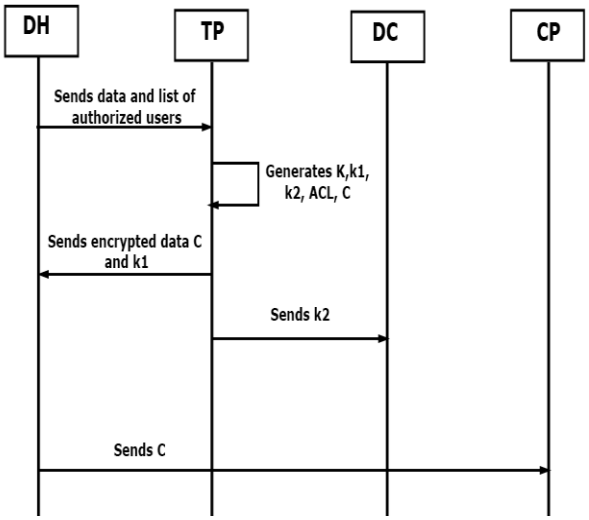


**Fig 2: The data upload model**

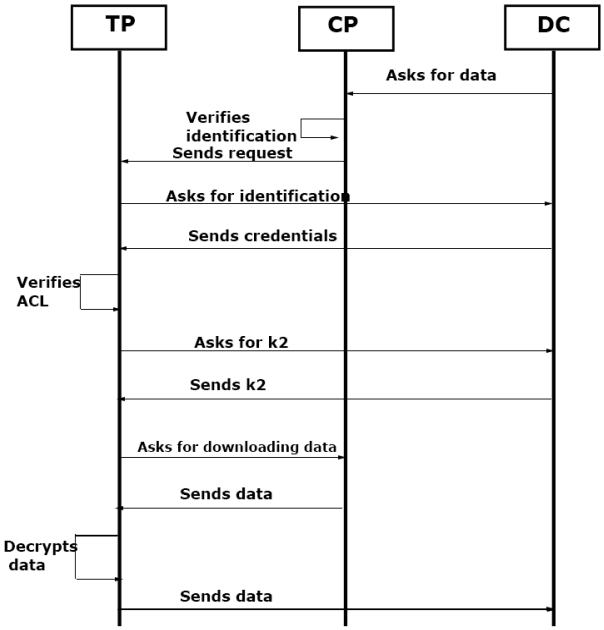


**Fig 3: The data download model**

Since each user has a specific pair of portions the identity management can be easily done. Thus, the TP downloads the file from the cloud and proceeds to the decryption process after verifying the data integrity using the HMAC signature. If the user has the corresponding k2 the file will be decrypted, otherwise, the decryption process will fail. After favorable decryption the file O will be sent to the user. And the key k generated will be deleted from the TP center.

### *3.2.3 Data update*

When a data customer or the data owner needs to update the file, the process to follow is similar to the upload method just here the process won't include the ACL generation or any other access control tool. After updating a data by a user or the owner, he needs to send a request of update to the TP containing the file id, the group id and k2 to encrypt the file and uploads it directly to the cloud. This is done after that TP verifies that the user has the right to write on file from the ACL corresponding to the updated file. If the user has the right to update the file, the TP encrypts it using k generated from k2 and k1, calculates the HMAC signature and sends the

encrypted file to the cloud. Also in this case the k must be deleted after encryption.

### *3.2.4 Adding a new user*

In the case when a new user needs data, only data owner who has the right to add users to a specific group. He sends a user id, the access rights appropriated to him to be added into the ACL, the ids of files he can access, and the group id to which the new user will belong. The TP after receiving that information, it updates the ACL related to each file the new user is authorized to access. The corresponding portion of key related to each file will be generated and shared with the new user mentioning the file id that is relying on it.

### *3.2.5 User revocation*

When a user quits the system, the TP must have knowledge of its departure to prevent a malicious use of data. So here the data owner should inform the TP about this member to remove him from the ACLs of different files to which he has access. The use of key partitioning helps the TP and the DH to keep data unchanged even after a member departure because even with the portion k2 in possession of the user without the second part he can't get the data decrypted from the TP.

## 4. CONCLUSION

This paper gives a proposition to a new framework to secure data sharing in cloud computing storage using symmetric key encryption schemes. Our framework ensures confidentiality and proposes an efficient and flexible access control system managed by the data owner and the third trusted party. We tried in this paper to give the overview of our cryptographic schemes and different methods we adopt for the realization of our framework with the target to give access to data just for authorized users excepting the cloud provider and malicious users. The approach proposes a method to guaranty confidentiality and access control system without complexity in key management strategy or in computations.

## 5. REFERENCES

[1] D. Thilakanathan, S. Chen, S. Nepal and R. A. Calvo, ''Secure data sharing in the cloud,'' Security, Privacy and Trust in Cloud Systems, chapter, Part 1, pp. 45-72, 2014

[2] M.Ali et al. ''SeDaSC: secure data sharing in clouds, '' IEEE System Journal, 2015

[3] F. Jing-yi , H. Qin-long, M. Zhao-feng, Y. Yi-xian, ''Secure personal data sharing in cloud computing using attribute based broadcast encryption,'' The Journal of China Universities of Posts and Telecommunications, Issue 6, pp -51,2014

[4] Z. Zhou , D. Huang, ''An efficient ciphertext-policy attribute based encryption and broadcast encryption'', Proceedings of the 17th ACM conference on Computer nd communications security pp 753-755, 2010.

[5] S. lingwei, Y. Fang, Z. Ru, N. Xinxin, ''Method of secure, scalable, and fine-grained data access control with efficient revocation in untrusted cloud'', The journal of China Universities of Posts and Telecommunications, Issue 2, pp 38-43, April 2015.

[6] J. Bethencourt, A. Sahai, B.Waters, ''Ciphertext-policy attribute based encryption'', Proceedings of the 2007 IEEE Symposium on Security and Privacy, pp 321-334.

[7] N. S. Kumar, G.V. Rajya Lakshmi, B. Balamurugan, ''enhanced attribute based encryption for cloud computing,'' , Procedia Computer Science, Volume 46, 2015, Pages 689–696

[8] D.Tiwari, G. Gangadharan, ''Secure sharing of data in cloud computing'', Security in Computing and Communications,pp 24-35, 2015.

[9] G.Wei, R.Lu, J.Shao, ''EFADS: efficient, flexible and anonymous data sharing protocol for cloud computing with proxy re-encryption'' Journal of Computer and System Sciences, Volume 80 Issue 8, December, 2014, Pages 1549-1562.

[10] X. Dong et al, ''Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing'', Computers & Security(2014), pp 1-14.

[11] B.Suzie, A. Reiter, F. Reimair, D. Venturi, B. Kubo, "Secure data sharing and processing in heterogeneous clouds," , Procedia Computer Science 68 (2015), pp 116-126.

[12] A. Balu , K. Kuppusamy, ''An expressive and provably secure Ciphertext-Policy attribute based encryption'', Information Sciences: an International Journal, Vol 276 issue C, August 2014, pp 354-362