A Survey of Security Issues for Data Sharing over Untrusted Cloud Providers

1 Ebtesam A. Alomari, 2 Muhammad M. Monowar
1,2 Faculty of Computing and Information Technology, King AbdulAziz University, Jeddah, Saudi Arabia

Abstract
Interest in cloud computing market is growing rapidly in the recent years. It provides on-demand, high scalability computing resources with high availability and reliability. However, security is considered as one of the biggest obstacles for cloud computing. In this paper, we focus on the main security and privacy issues, pertinent to store and share data on untrusted cloud storage. We have investigated several existing approaches focused on security concerns in cloud computing. We classify the discussed studies based on three categories, which are: encryption and key management approaches, searching methods over encrypted data and access control schemes. We provide a comparative study of the existing solutions from each category identifying their strengths and limitations. We emphasize that although several techniques were proposed to protect data from cloud provider or any unauthorized users, there are still limitations need to be tackled. Therefore, we identified the research challenges indicating the future research directions in this area.

Keywords: Cloud Computing, Entrusted Storage, Security, Encryption.

1. INTRODUCTION
Nowadays, cloud storage is becoming a good solution for organizations that suffer from resource limitation. The cloud service provider (CSP) offers to the users the opportunity to get unlimited storage and computation capacities, which is billed based on the actual amount of consumed resources [1]. Further, It offers high scalability, redundancy and disaster recovery capabilities. In addition, it gives an opportunity to improve security and privacy where cloud storage data are maintained within a cloud, which can be reliable and faster to restore data more the traditional data center [2].

In the other side, public cloud computing environment considers more complex than traditional data center, where cloud is a multi domain environment, in which each domain can use different security, privacy, and trust requirements. In addition, the rapid changes to cloud environments may affect enterprises ability to set consistent security standards. Moreover, the security level given by cloud provider is not sufficient where it is considered as untrusted side [3].

Furthermore, the owners of the data have only limited control over the IT infrastructure, on the other side, the cloud providers have the ability to change user’s sensitive data and IT systems since they have excessive privileges. This leads to low trust level when storing and sharing data on a cloud. Therefore, the cloud users might feel uncomfortable when the cloud provider force hem to give personal information. In addition, the CSP may loss the reputation and credibility when the sensitive information stored on the cloud platforms is disclosed [3].

This paper will focus on the security challenges of storing and sharing data on cloud storage. There are two main concerns in this area, which are unauthorized users who can get to the sensitive data where the control is not in the hand of the owner, and the other concern is the cloud providers themselves because the data is kept in their storages [4].

Therefore, to provide trusted data sharing through untrusted cloud providers, users need techniques that enforce security policies in addition to offering the following requirements [5]:

- Confidentiality: The cloud storage provider should not have the ability to expose the data confidentiality where data stored on the cloud should be kept private, and only authorized users are able to gain access to encrypted content [1].
- Access control: The access to data store on the cloud should be under data owner control. Therefor, only designated user can access to the data while the cloud provider should not get any right to access the data.
- Authorization: Users, who do not have the permission, should not be able to access the data.
- Integrity: The system should detect any unauthorized changes.
Moreover, the CSP must achieve the above requirements and should provide to the user the following issues [1]:

- Enable encrypting data at either the data owner’s machine or on third party who offered encryption/decryption service within the trusted environment.
- Provide efficient filter and search over encrypted data and retrieve results matching such queries.
- Guarantee that only authorize users are able to access the data.

In this paper, we present a survey of the state-of-the-art solutions, proposed considering the security issues for data sharing over untrusted cloud providers. Our main contributions include: i) classify the existing solutions into three categories: encryption and key management techniques, searching methods over encrypted data, and access control schemes. ii) Provide a comparative study of the existing solutions from each category identifying their related strengths and limitations. iii) Identify the possible research challenges indicating the future research directions in this area.

The remainder of this paper is organized as follows: Section 2 provides an overview of cloud computing, Section 3 presents cryptographic background, Section 4 discusses the existing solutions along with their comparative study, Section 5 discusses the challenges and future research issues, and Section 6 concludes the paper.

2. CLOUD COMPUTING OVERVIEW

Cloud computing is the next generation platform that provides dynamic resource. It consists of three components, which are [4]:

- Cloud service provider (CSP): Responsible for providing high computation power and preserving the clients’ data.
- Owner: Can be individual consumer or organizations that use cloud resources.
- User: Registered by the owner to use his data on the cloud.

Furthermore, as given by the NIST (U.S. National Institute of Standards and Technology) [5], and as shown in Figure 1 above, there are five essential characteristics, which are on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service.

Cloud-computing technologies can be implemented in different architectures, under several service and deployment models, as explained in the following sections.

2.1 Deployment Models

There are several cloud deployment models [6], [7]. One of them is a public cloud computing where the infrastructure and computational resources become available to the public consumers over the Internet. The cloud provider owns it and operates it to the public. Google Apps is considered as an example for this model. The other model is a private cloud. The computing environment in this model operates for a single organization, which also has the management responsibility. It gives the organization the ability to control and manage the infrastructure resources, and cloud consumer's more than public cloud.

The other two deployment models are community and hybrid clouds. In the community model, several organizations can share the cloud infrastructure with the same policy and compliance considerations. However, a hybrid cloud is considered as more complex than the other deployment models because it is a composition of two or more clouds (public, private...).

2.2 Service Models

The most used service models, as provided in several articles are the following four: [6], [7]:

![Fig 1: NIST model of cloud computing definition [5]](image-url)
Software-as-a-Service (SaaS) is a model, which delivers application and computational resource as a service. It decreases the total cost of software and hardware operations, improvement, and maintenance. The cloud provider becomes responsible for security provisions, so the cloud consumers do not manage or control the cloud infrastructure or individual applications, but they are able to choose preferences and set limited administrative applications.

Platform-as-a-Service (PaaS) provides computing platform as a service. It focuses on reducing cost and complexity of hardware and software components of the platform. The cloud consumer has control on applications platform, which mean that security provision is divided between cloud provider and consumer.

Infrastructure-as-a-Service (IaaS) provides the main computing infrastructure as a service to prevent the need of buying and managing hardware and software infrastructure components. The clients have the ability to decide which operating system and deployment environment are needed. This leads us to conclude that cloud consumer is responsible for security provision. Amazon’s Elastic Computing Cloud (EC2), considers as example of this model, it enables users running their software by rent computing power from Amazon.

3. CRYPTOGRAPHIC BACKGROUND

In general, when user wants to store data securely on an un-trusted storage, he needs a way to encrypt the data and then store keys on secure keystore as presented in figure 2. Further, there are a wide range of the encryption schemes and key management approaches that are applicable to a cloud environment. This section provides an overview about the main approaches that can be used to store data securely in the cloud.

3.1 Encryption Approach

Encryption is converting clear text into ciphertext. It ensures the confidentiality, and protects the information from unauthorized people during transiting over unreliable network, or during storing in untrusted storage. In this section, we give a brief overview over some cryptographic methods.

3.1.1 Symmetric and Asymmetric Algorithms

The common data encryption methods are symmetric and asymmetric algorithms. Symmetric cryptography uses the same key to perform encryption and decryption. On the other hand, asymmetric uses two different keys (a public key) for encryption operation, and (a private key) for the decryption operation, such as Elliptic Curve Cryptography (ECC).

Generally, both of them can be used for cloud cryptography, but symmetric algorithms are faster than asymmetric. So, it is more efficient and suitable for encrypting large volumes of data [9].

3.1.2 Attribute-Based Encryption

Another set of cryptographic techniques is Attribute-based encryption (ABE). It can be a suitable approach when we need each user to be able to decrypt only his authorized data. Therefore, the decryption will only work if there is a match between the attributes associated with the decryption key and the policy used to encrypt a message. This flexibility makes it an attractive choice for cloud computing [10]. It has developed into two branches, which are: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). KP-ABE [10]:

- KP-ABE: combines the access policy with keys corresponding to attributes.
- CP-ABE: allows the specification of a decryption policy to be associated with a ciphertext.

3.1.3 Proxy Re-Encryption

In this scheme a semi-trusted proxy is used to replace a ciphertext that can be decrypted by user A into another ciphertext that can be decrypted by user B, without recognizing the original information and user secret keys [10]. Therefore, this approach enables the data owner delegates a third party to perform some computational intensive tasks, such as, re-encryption while leaking little part of the information to them.

3.2 Key Store

It is important to manage the encryption keys and store them in trusted environment to provide secure storage approach, and prevent an attacker from getting the keys to decrypt the data. Keys are stored on a keystore that can be on one of the following [8]:

- Portable device: owned by an authorized user within the trusted environment.
- Specialized server: placed in somewhere, in a trusted environment.

4. THE EXISTING SECURITY SCHEMES FOR DATA SHARING

This section discusses the state of the art schemes for data sharing over untrusted cloud providers. We classify the schemes into three categories, which are encryption and key management techniques, efficient
searching methods over encrypted data, and data access control schemes.

4.1 Encryption and Keys Management

Today, the most famous cloud storage services such as Drop box and Sugarsync is storing file in encrypted format by using AES. However, the main limitation is that the encryption keys are managed by them [9], which makes them unsuitable choice from users who need high confidentiality level. Therefore, several researches conducted to find more secure solution.

Huang et al. [11], proposed service named SSTreasury+. It is responsible for encrypting user’s data before uploading it to the cloud. It consists of three components as shown in figure 3:

- SSGuard: responsible for encrypting file by random AES (symmetric key algorithm) before uploading and decrypting after downloading to get the original contents.
- SSCoffers: store the encrypted file.
- SSManger: responsible for storing keys. The keys are encrypted before stored in SSManger.

The owner is the only one that has the decryption key, which used to decrypt the keys in SSManger. In this approach, they concerned on providing portable and flexible decryption key method, so they supposed encoding the decryption keys into the QR Code to be store in Smartphone or flash drive. However, their method increases the overhead on the user because he needs to install desktop application that performs the encryption/decryption locally.

Fig 3: SSTreasury Architecture [11]

Saleh and Meinel [12] proposed HTTP proxy named HPISecure that responsible for encrypting or decrypting the data. It needs to be installed on the client machine. Further, it intercepts the HTTP request/response objects, and then encrypts the data before transmitting to the cloud, or decrypts the data received back from the server. They supported using public-key cryptography. Furthermore, to make it difficult for malicious users to uncover the private key of the user, they suggested giving each user a group of private keys, so every document may be encrypted using different keys. On the other side, for key management, they offer using the facilitator, which can be a third-party cloud provider or USB that store the keys and related information. However, the limitations in this work are that the user needs to install the application on each machine that he needs to use it. Also, they limited the sharing and collaboration of the documents between a group of users.

Hwang et al. [9] proposed a business model for cloud computing based on the concept of providing two separate service provider one for Encryption/Decryption and the other for storage. The storage system stores user data and keys, which have already been encrypted while Encryption/Decryption Service System performs the decryption/encryption operations, and then deletes the data. The main concept of Hwang et al. approach is dividing the process into more than one service provider in order to reduce operational risk of disclosing user data. However, there is no guarantee that the Encryption/Decryption Service System really deletes the data and do not store or use it.

Moreover, Puttaswamy et al. [13] proposed a tool called Silver line to achieve strong data confidentiality in cloud. Unlike the above approaches, they focused on the applications that are data as well as computation intensive. Their main goal is encrypting as much of the application data as possible without breaking the application’s functionality. Hence, while the cloud application cannot perform computation on any data it cannot access in plaintext, they suggested decrypting only the data that are not included in the computation.
Moreover, the encryption/decryption is performed locally on user machine while the decryption keys are stored in different service provider. They considered sharing data securely between group members. So, they suggested dividing the users into groups, and assign a single encryption key to this group. However, this makes the encrypted data vulnerable to a variety of attacks by the cloud. Table 1 summarizes the approaches with their related strengths and limitations.

### 4.2 Searching Methods Over Encrypted Data

In general, when the owner of the secret key needs to search some encrypted data that are stored in the cloud storage, he can perform one of the two following possible techniques [14]:

- Get all encrypted data from the cloud server, and then search after decrypting all the downloaded data. However, it is inconvenient method, and it considered as inefficient, especially if the encrypted data are huge or the user is a thin client like the mobile users.
- Send his key to the cloud provider, which becomes responsible for the decryption and search procedures. However, this way may lead to serious security risk where the cloud server obtains the secret key.

Therefore, there are several searchable encryption schemes were proposed to search over encrypted data stored in the cloud storage. We will present the main schemes in the rest of this section.

Koletka and Hutchison [1] described a solution that allows searching over encrypted data that are stored on a public cloud. Their architecture indicated that the owner is responsible for all the security operations on the data; include creating an Encrypted Keywords List, adding user or modifying user rights. In addition, they suggested that the server using symmetric searchable encryption scheme to determine the results of the query. When the user sends encrypted keyword, the cloud provider will perform cryptographic operations over the encrypted files in the Storage Service to find all files that contain the keywords. The limitation in this strategy is the huge performance overheads on the owner side.

Liu et al. [15] proposed a secure and privacy preserving keyword searching (SPKS) scheme, which allows the cloud service provider to participate in data decryption, and to return only files containing certain keywords specified by the users. They support using

<table>
<thead>
<tr>
<th>Approach name</th>
<th>Function</th>
<th>Strengths</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSTreasury+ [11]</td>
<td>- Offer desktop application to encrypt/decrypt users file before uploads it to the cloud.</td>
<td>The keys are stored in ciphertext in different service provider, and the decryption key is only with the owner.</td>
<td>- Increase overhead on user machine.</td>
</tr>
<tr>
<td></td>
<td>- The keys are stored in different service provider.</td>
<td></td>
<td>- Difficult portability: User needs to install the proposed encryption application in each machine that he wants to use it.</td>
</tr>
<tr>
<td>HPISecure [12]</td>
<td>- Propose HTTP proxy to be installed on the client machine. It is responsible for encrypting or decrypting the data.</td>
<td>Each user has a group of private keys so each document can be encrypted using different keys.</td>
<td>- This method limits the sharing and collaboration of encrypted documents between users.</td>
</tr>
<tr>
<td></td>
<td>- Offer storing the keys on third-party cloud provider or portable devices.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A business model for cloud computing [9]</td>
<td>- Suggest using third-party cloud provider for Encryption/Decryption.</td>
<td>The third-party cloud provider that responsible for the cryptography operations can be used from any machine, unlike the desktop applications that need installation.</td>
<td>- There is no guarantee that the Encryption/Decryption Service System delete the data after perform the operations.</td>
</tr>
<tr>
<td>Silver line [13]</td>
<td>- Perform encryption/decryption locally on user browser.</td>
<td>They focused on the applications that are data as well as computation intensive</td>
<td>- Overhead on user machine</td>
</tr>
<tr>
<td></td>
<td>- The keys are stored in the same organization that wants to securely deploy its application and data on the cloud.</td>
<td></td>
<td>- Not all data in the cloud are encrypted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- The proposed approach for sharing the encryption key between a group of user is not secure.</td>
</tr>
</tbody>
</table>
public key encryption to encrypt both file and keywords before upload them to the CSP. When the owner wants to retrieve a file containing a keyword, he need to run the Trapdoor algorithm to compute a short trapdoor for the keyword using his private key, and then sends it to the CSP. Hence, this increased the overhead on the owner. In addition, there approach is not suitable for one too many communications while they do not take care for fine-grained access policy.

To handle these drawbacks, Koo et al. [16] provided a searchable encryption schemes suitable for cloud storage systems with a huge amount of data. It is based on attribute-based encryption (ABE), where the keywords as a part of attributes can be served as index terms to be searched. This scheme is suitable for one-upload-many-download more than using public key encryption with keywords search (PEKS). In addition, this scheme supports simple comparisons between the searched entities instead of an exhaustive search with cryptographic calculations. Further, it improves the efficiency of the searching process by involving both cloud provider and owner. The data owner can define both of fine-grained access policy and the keyword list, which is needed to retrieve its file under the access policy. Moreover, unauthorized users who do not have enough attributes to fulfill the access policy, he will not be able to access the data in CSP. Also, the identity of the receiver or the sender is replaced with a set of attributes, so this ensures anonymity for both of them. Therefore, this scheme enhances the confidentiality for data in cloud storage.

**Table 2:** The schemes of searching over encrypted data in cloud storage

<table>
<thead>
<tr>
<th>Searchable Scheme</th>
<th>Reduce overhead on owner</th>
<th>Multiple keyword Searching</th>
<th>Ranked query result</th>
<th>Allow define access policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using symmetric searchable encryption scheme. [1]</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Privacy preserving keyword-searching (SPKS) scheme. [15]</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Searchable encryption schemes based on attribute-based encryption (ABE). [16]</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Using ranked searchable symmetric encryption scheme over encrypted cloud data. [17]</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Multi-keyword ranked search over encrypted data in cloud computing (MRSE). [18]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Even though the above approaches supported multiple keyword searching, they did not consider ranking the retrieved files based on the most relevant to user query. Therefore, Wang et al. [17] aimed to improve system usability by returning the matching files in a ranked order. They proposed using ranked searchable symmetric encryption scheme over encrypted cloud data. So, the matched files will be retrieved in a ranked order based on certain relevance criteria such as keyword frequency. The experimental results showed that it is effective and efficient since it ranks results instead of returning undifferentiated results. Further, security analysis indicated that their scheme enjoys a strong security guarantee compared to the symmetric searchable encryption schemes. However, they only supported single keyword search.

Cao et al. [18] also handled the above limitation. They proposed multi-keyword ranked search over encrypted data in cloud computing (MRSE). They selected the ranking criteria “coordinate matching” that use the number of query keywords appearing in a file and perform as many matches as possible, to get the relative file to the search query. Therefore, if one query’s keyword is found in most files, the importance of this keyword in the query will be less than other keywords, which appear in less number of files. On the other side, the file will be preferred more in case the frequencies of the keyword on this file is higher than the frequency of the same keyword on the other files. Table 2 above gives a brief comparison among searching schemes.

### 4.3 Fine Grained Data Access Control

There is a need to devise efficient techniques which will enable delegating the cloud providers to participate in executing the computational intensive tasks of data access control, without disclosing the underlying data contents to them. In addition, the technique should offer a secure way for revoking the access rights from users when they become unauthorized to access the encrypted data.

Zhao et al. [19] suggested using elliptic curve encryption scheme. While the data are stored in the cloud in encrypted form, users who need accessing the data should send a request to the owner. Then, owner sends a credential to the Cloud Storage Provider for the re-encryption of the data, and sends a credential for the user to decrypt the re-encrypted data as shown in figure 4 in steps (C) and (D). In this case, user can require the data from cloud provider and decrypt the re-encrypted data by...
his private key, which means that authorizes users only can access to data by issuing credentials.

Fig 4: Zhao et al. scheme [19]

However, the main limitations in their approach are that they suppose the data owner knows the private key of the cloud provider. In addition, their algorithm involves many multiplication calculations, which requires a large amount of computation. Further, this solution causes heavy computation overhead on the data owner because it is responsible for key distribution and fine-grained access control.

Yu et al. [20] investigated adding the security concerns by using proxy re-encryption (PRE) scheme. They focused on the problem of revoking user securely. When owner wants to revoke any user, he will send PRE keys to the CSP, and then delegate it to execute the re-encryption. Further, the same researchers (Yu et al.) [21], suggested another solution combining the techniques of proxy re-encryption with Key Policy Attribute Base encryption (KP-ABE) and delegating to the CSP the tasks of data file re-encryption and user secret key update. Moreover, they suggested assigning each data file with a group of attributes, and associating each user with an expressive access structure, which is defined usually as an access tree over data attributes. Therefore, the user will be able to decrypt the ciphertext only when his access structure satisfies the data attributes. However, even their technique ensures data confidentiality and considers user revocations by updating user secret keys for all the users except the user who will be revoked; it is still vulnerable to collusion attack using a revoked user and cloud server. Furthermore, Do et al. [22] considered this limitation and suggested scheme to prevent the collusion attack by having trusted authority, called privilege manager group who is responsible for generation of re-encryption. Also, they provide selectively delegate decryption right approach by using Type-based Proxy re-encryption scheme. They suggested dividing data file into the header and the body where header is sent to privilege manager group, and the body to the cloud service provider. Therefore, by using Type-based PRE, owner can grant weight to show the whole or part of the data to users according to his reliability towards this user. For example, if Alice is more reliable than Bob, the part of the message, which can read can be set to have a selective delegation. However, this method did not consider the performance and efficiency.

Wang et al. [10] investigated applying a hierarchical attribute-based encryption scheme (HABE) to provide fine-grained access control, full delegation and high performance mechanism. They proposed combining a hierarchical identity-based encryption (HIBE) scheme and a ciphertext policy attribute-based encryption (CP-ABE) scheme. Further, they suggested a scalable revocation approach, by using proxy re-encryption (PRE) and lazy re-encryption (LRE) to revoke access rights from users efficiently.

However, the main problems of the approaches [10], [20], [21] are that they require the data owner to be online to send the PRE keys to the cloud provider when user is revoked from the system. Hence, any delay of issuing PRE keys may lead to potential security risks.

On the other side, when owner delegate CSP to distribute the update keys to the remaining authorized users, the CSP should know the identities of these users, and then he can know the effective time of each one, which mean more information are leaked to the service provider. Therefore, other researchers suggested that the data owner should take responsibility for distributing the update keys, to avoid leaking additional information. However, as discussed before, this will lead to huge overhead on owner side.

To address this problem, Liu et al. [23] investigated scheme different from prior work, where it enables access right for each user to be effective for a predetermined time, in addition it enables the cloud service provider to re-encrypt ciphertexts automatically based on its own time. Therefore, through the process of user revocations, the data owner can be offline. Their proposed strategy was based on a time-based proxy re-encryption (TimePRE) scheme. It integrates the concept of time into the combination of ABE and PRE, where each data is connected with an attribute-based access structure and an access time. In addition, each user is recognized by a group of attributes and an eligible time periods, which declare the user’s access right period.

The strong points in this approach are that the data owner can be offline when the user is revoked. The cloud service provider can automatically update the access time of the data when it receives a data access request. Moreover, only the users whose attributes satisfy the access structure and whose access time are not finished, can get the re-encrypted ciphertext. Hence, this scheme is suitable when the period of validity of each user’s access right is determined previously. However, they did not support enabling the data owner to perform the revocation at any time. Furthermore, they did not take in consideration that the CSP may collude with malicious users by always update their access time to fake time.
Then, the revoked users can recover data, and the revocation mechanism will lose effectiveness.

On the other side, it is important to guarantee that the key investigated to each user cannot be shared; such key abuse problem exists in ABE-based access control schemes. Li et al. [24] focused on this problem and provided using tracing algorithm to prevent illegal key sharing by collude. The validity of this algorithm is based on the fact that the user cannot change his identity in his secret key inserted in the attribute private key. Hence, users able to access the file if and only if the user’s attributes satisfy the file’s access structure. The Table 3 gives a brief comparison among access control schemes.

<table>
<thead>
<tr>
<th>Cryptographic Techniques</th>
<th>Strength</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Elliptic curve to encrypt+ access to data by issuing credentials. [19]</td>
<td>Authorizes users only have access to data</td>
<td>Heavy computation overhead on the data owner</td>
</tr>
<tr>
<td>Integrate the technique of proxy re-encryption with CP-ABE. [20]</td>
<td>Supports attribute revocation.</td>
<td>-Data owner should be online to send the PRE keys to the CSP.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Delay in sending the keys may cause security risk.</td>
</tr>
<tr>
<td>Combine techniques of ABE, proxy re-encryption, and lazy re-encryption. [21]</td>
<td>Delegate most of the computation tasks involved in user revocation to the CSP.</td>
<td>-Data owner should be online to send the PRE keys to the CSP.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-It can violate data confidentiality through collusion of revoked user and cloud server.</td>
</tr>
<tr>
<td>- Fine grant access control scheme by combine a (HIBE), and (CP-ABE). - Revocation scheme by combine (PRE) and lazy re-encryption (LRE) to the HABE scheme. [10]</td>
<td>Achieve a scalable revocation mechanism in cloud computing</td>
<td>-Data owner should be online to send the PRE keys to the CSP.</td>
</tr>
<tr>
<td>Type-based Proxy re-encryption. [22]</td>
<td>Solve the weakness in [20] (secure against collusion attack)</td>
<td>This method did not consider the performance and efficiency.</td>
</tr>
<tr>
<td>TimePRE (integrates the concept of time into the combination of ABE and PRE). [23]</td>
<td>-Enable the CSP to re-encrypt ciphertexts automatically based on its own time. - The data owner can be offline in the process of user revocations.</td>
<td>Delay of issuing the PRE keys -Not suitable when data owner wants to revoke a user from the system at any time.</td>
</tr>
</tbody>
</table>

5. CHALLENGES AND FUTURE RESEARCH ISSUES

5.1 Encryption and Keys Management

In this paper, we discussed approaches for encryption and keys management. We notice that the approaches that offer a tool for encryption/decryption locally are secure because the cryptography operations are performed in user machine. However, these approaches may increase the overhead on the owner. In addition, it is not suitable for thin client like mobile users. Further, users have to install the tool in each machine they need to use, which conflict with the main advantages from using cloud that allowed access the data from anywhere at any time.

Hence, the future proposed solution need to consider the portability, which mean users should be able to encrypt/decrypt their data from any place. Thus, user should be able to use any computer to access his data securely, not only the machine that he used for the first time.

On the other side, even though the approaches that suggested using a third party service provider to perform encryption/ decryption operations is handle the portability problem. The main challenge in this case is how to provide restrictions to prevent encryption/ decryption provider from storing data or keys after the operation was performed, and how to guarantee that to the users to increase the trust.

Furthermore, there is a need to provide a suitable solution based on the type of data that are stored in the cloud. For documents-based application and for users that need to store the file in cloud storage such Google docs and Drop box, there is a need to offer a secure way for sharing files with other users. In addition to consider the collaboration and how users can work at the same time and edit on the same encrypted files. On the other side, the solution should also focus on the database-based applications that store structured data on the cloud. These data usually involved in computation to perform the application’s functions. So, there is a need to more secure and efficient solution to handle the challenge of encrypting the data without breaking the functionality of the application.
To sum up, the proposed solution should cover the following:

a. Take in consideration thin client’s machine that cannot perform huge operations.
b. Enable user to be easily access to his encrypted data on cloud from any machine.
c. Enable storing the keys safely while preventing cloud-based service and unauthorized parties from stealing the keys.
d. Offer secure and suitable key sharing between users that are allowed to access the data.
e. Ensure that the responsible for Encryption/decryption cannot store or forward the data, to unauthorized user, in plaintext after perform the operation.

5.2 Searching Methods Over Encrypted Data

The discussed approaches for keyword searching over encrypted data assumed that the owner is responsible for specifying the keywords that are attached with the file. This keyword list is stored with the file in ciphertext, in the cloud storage, as an index term to be searched to retrieve the file. However, they did not take in consideration that when user updates the file, the files’ keywords will be changed, so the attached keywords list needs to be re-identified. Therefore, the future researchers need to tackle this issue when developing a searchable encryption scheme.

The other challenge that needs to be considered is the high-level executive user, who is automatically qualified to search any value and not willing to reveal query to the owner. Therefore, there is a need to develop more efficient schemes to find out how to hide the high-level executive user query contents from the owner in addition to guarantee the data owner that the hidden contents are authorized.

To sum up, there is a need to find efficient search scheme that should satisfy the following requirements:

a. Allow the CSP to participate in the searching, to decrease computational overhead in owner side.
b. Increase the searching speed and retrieve the data efficiently without revealing the content of the query or the search results to the cloud provider.
c. Retrieve the most relevant searching results and rank them in efficient way.
d. Enforce strong access policy to ensure that neither the cloud providers nor unauthorized users can be able to search and decrypt the retrieved data.

5.3 Fine Grained Data Access Control

To enhance the security, only authorized users should have access to the data. Therefore, there is a need to implement strong access policy schema and also to support user revocation, without heavy computation overhead on the data owner. As proposed in [20], to enable data owner to be offline through the process of user revocations, the CSP could be granted to re-encrypt ciphertexts automatically based on specified time. However, this mechanism is not suitable when data owner wants to revoke a user from the system at any time. Therefore, future work on this area should cover this limitation. In addition, collude attack should be prevented in order to avoid violating data confidentiality through collusion of the revoked user and cloud provider.

To sum up, the efficient approach for fine grained access control should cover the following requirements:

a. Reduce the workload on the data owner to ensure that the approach is suitable for the data owners that cannot be often online.
b. Delegate most of the computation tasks involved in assigning access policy or revoking user to the CSP.
c. Enhance the performance where the delay especially in revoking users may cause a security risk.
d. Prevent collude attack. The data confidentiality should not violate by revoked user, and he should not be able to decrypt files even if he receives it.

6. CONCLUSION

Nowadays, cloud services are increasingly being used for every kind of computing. However, the security is considered as the main obstacles. This paper focuses on the security issues related to share and store data on untrusted cloud provider. Basically, to increase user trust in using cloud storage, CSP should provide a complete security guarantee to the data throughout the entire process from the owner to the cloud storage, and then to authorize users. Therefore, when the owner store data on cloud storage, the data have to stay private, and the owner should has the control on the data access and share.

Furthermore, there are several mechanisms and techniques are applied to shield the critical information from unauthorized parties. In this paper, we survey several existing techniques and classify them based on three categorizes, which are: encryption and key management approaches, searching over encrypted data and access control schemes.

We conclude that to enhance the security, and enjoy the benefit of cloud, there is a need to provide as strong-as-possible mechanisms, without heavy computation overhead on the data owner. Moreover, the solutions should take consideration of the performance and pay attention to the speed of searching and decrypting since the amount of data in the cloud is huge, whereas the technique will be inefficient if it takes too long time to retrieve data to users. Finally, we presented in this survey the limitations and challenges that requiring future researches to handle them.
REFERENCES


