An Efficient Compression and Encryption Based on Data Security for Cloud Computing Using LZ4 Algorithm

Dr. S. Rajeswari MCA., M.Phil(CS)., Ph.D.,

Department of Computer Application, Sree Saraswathi Thyagaraja College, Pollachi, India.

rajeswari@stc.ac.in

Mrs. R. NithyaDevi M.Sc (CS)., M.Phil Research Scholar, Sree Saraswathi Thyagaraja College, Pollachi, India. nithyaait@gmail.com

Abstract - Cloud computing, or something being in the cloud, is an expression used to describe a variety of different types of computing concepts that involve a large number of computers connected through a real-time communication network such as the internet. An alternative perspective on the topic of cloud security is that this is but another, although quite broad, case of "applied security" and that similar security principles that apply in shared multi-user mainframe security models apply with cloud security. Solutions to various cloud security issues vary, from cryptography, particularly public key infrastructure (PKI), to use of multiple cloud providers, standardization of APIs, and improving virtual machine support and legal support. Our proposed system is an IBE (Identity-based encryption) - based mechanism and it provides two-factor data encryption protection that contains the distinctive clients/individual users. The administrator transfers the information documents and data is compressed and encrypted format. LZ4 algorithms are used for compression and encryption. Where compression provides less storage space and increasing security. This encrypted data will be stored on cloud. The key will be generated. At the point when the client need information, he ask for the Edu-admin, Admin check the client is approved one or not, and if client is approved then key will be sent to client mail. We proposed safe information sharing plan, which can accomplish secure key circulation and information sharing this process is completely transparent to the sender and cloud server cannot decrypt any cipher text at any time.

Key words: cloud computing, cloud security, cryptography, data compression, two factor authentication, PKI.

I. INTRODUCTION

1.1 CLOUD COMPUTING

Cloud computing, or something being in the cloud, is an expression used to describe a variety of different types of computing concepts that involve a large number of computers connected through a real-time communication network such as the internet. In science, cloud computing is a synonym for distributed computing over a network and means the ability to run a program on many connected computers at the same time.

1.2 CLOUD COMPUTING SECURITY

An alternative perspective on the topic of cloud security is that this is but another, although quite broad, case of "applied security" and that similar security principles that apply in shared multi-user mainframe security models apply with cloud security. The relative security of cloud computing services is a contentious issue that may be delaying its adoption. Physical control of the Private Cloud equipment is more secure than having the equipment off site and under someone else's control. Physical control and the ability to visually inspect data links and access ports is required in order to ensure data links are not compromised. Issues barring the adoption of cloud computing are due in large part to the private and public sectors' unease surrounding the external management of security-based services. It is the very nature of cloud computing-based services, private or public, that promote external management of provided services. This delivers great incentive to cloud computing service providers to prioritize building and maintaining strong management of secure services. Security issues have been categorized into sensitive data access, data segregation, privacy, bug

DOI : 10.18231/2454-9150.2018.0983

exploitation, recovery, accountability, malicious insiders, management console security, account control, and multi-tenancy issues. Solutions to various cloud security issues vary, from cryptography, particularly public key infrastructure (PKI), to use of multiple cloud providers, standardization of APIs, and improving virtual machine support and legal support

1.3 GENERAL PROJECT DETAILS

CLOUD computing provides seemingly unlimited "virtualized" resources to users as services across the whole Internet, while hiding platform and implementation details. Today's cloud service providers offer both highly available storage and massively parallel computing resources at relatively low costs. As cloud computing becomes prevalent, an increasing amount of data is being stored in the cloud and shared by users with specified privileges, which define the access rights of the stored data. One critical challenge of cloud storage services is the management of the ever-increasing volume of data.

enhanced security protection for asymmetric encryption, there is a single secret key corresponding to a public key or an identity. The decryption of ciphertext only requires this key. The key is usually stored inside either a personal computer or a trusted server, and may be protected by a password. The security protection is sufficient if the computer/server is isolated from an opening network. Unfortunately, this is not what happens in the real life. When being connected with the world through the Internet, the computer/server may suffer from a potential risk that hackers may intrude into it to compromise the secret key without letting the key owner know. In the physical security aspect, the computer storing a user decryption key may be used by another user when the original computer user (i.e. the key owner) is away (e.g., when the user goes to toilet for a while without locking the machine). In an enterprise or college, the sharing usage of computers is also common. For example, in a college, a public computer in a copier room will be shared with all students staying at the same floor. In these cases, the secret key can be compromised by some attackers who can access the victim's personal data stored in the cloud system. Therefore, there exists a need to enhance the security protection

An analogy is e-banking security. Many e-banking applications require a user to use both a password and a security device (two factors) to login system for money transfer. The security device may display a one-time password to let the user type it into the system, or it may be needed to connect with the computer (e.g., through USB or NFC). The purpose of using two factors is to enhance the security protection for the access control. As cloud computing becomes more mature and there will be more applications and storage services provided by the cloud, it is easy to foresee that the security for data protection in the cloud should be further enhanced

1. Our system is an IBE (Identity-based encryption)-based mechanism. That is, the sender only needs to know the identity of the receiver in order to send an encrypted data (ciphertext) to him/her. No other information of the receiver (e.g., public key, certifi- cate etc.) is required. Then the sender sends the ciphertext to the cloud where the receiver can download it at anytime.
2. Our system provides two-factor data encryption protection. In order to decrypt the data stored in the cloud, the user needs to possess two things. First, the user needs to have his/her secret key which is stored in the computer. Second, the user needs to have a unique personal security device which will be used to connect to the computer (e.g., USB, Bluetooth and NFC). It is impossible to decrypt the ciphertext without either piece.
3. More importantly, our system, for the first time, provides security device (one of the factors) revocability. Once the security device is stolen or reported as lost, this device is revoked. That is, using this device can no longer decrypt any ciphertext (corresponding to the user) in any circumstance. The cloud will immediately execute some algorithms to change the existing ciphertext to beun-decryptableby this device. While the user needs to use his new/replacement device (together with his secret key) to decrypt

DOI : 10.18231/2454-9150.2018.0983

his/her ciphertext. This process is completely transparent to the sender. 4) The cloud server cannot decrypt any ciphertext at any time.

II. LITERATURE SURVEY

2.1 DUPLESS: SERVER-AIDED ENCRYPTION FOR DEDUPLICATED STORAGE

Cloud storage service providers such as Google Drive, Dropbox, Mozy, and others perform data deduplication to save space by only storing one copy of each data uploaded. Should users conventionally encrypt their datas, thus, savings are lost. Message-locked encryption (the most prominent manifestation of which is convergent encryption) determines this tension. Thus it is inherently subject to brute-force attacks that can recover datas falling into a known set. This proposed an architecture that provides secure deduplicated storage resisting brute-force attacks, and realizes it in a system called DupLESS. In DupLESS, users encrypt under message-based keys attained from a key-server via an unaware PRF protocol. It enables users to store encrypted data/content with an existing service, have the service perform data deduplication on their behalf, and yet achieves strong confidentiality guarantees.

2.2 PROOFS OF OWNERSHIP IN REMOTE STORAGE SYSTEMS

Cloud storage systems are becoming increasingly accepted. A promising knowledge that keeps their cost down is data deduplication, which stores only a single copy of repeating data/content. User-side data deduplication attempts to identify data deduplication opportunities already at the user and save the bandwidth of uploading copies of existing datas to the server. In this work This identify attacks that exploit user-side data deduplication, allowing an attacker to gain access to arbitrary-size datas of other users based on a very small hash signatures of these datas. More specifically, an attacker who knows the hash signature of a data can convince the storage service that it owns that data, hence the server lets the attacker download the entire data.

To overcome such attacks, This introduce the notion of proofs-of - ownership (PoWs), which lets a user efficiently prove to a server that that the user holds a data, rather than just some short information about it. This formalizes the theory of proof-of-ownership, under accurate security definitions, and rigorous efficiency requirements of Peta byte scale storage systems. This then present solutions based on Merkle trees and specific encodings, and analyze their security. This implemented one variant of the approach. These performance measurements indicate that the approach incurs only a small overhead compared to naive user-side data deduplication.

2.3 CLOUDEDUP: SECURE DATA DEDULICATION WITH ENCRYPTED DATA/CONTENT FOR CLOUD STORAGE

With the constant and exponential raise of the number of users and the size of their data/content, data/content deduplication becomes more and more a requirement for cloud storage providers. By storing a unique copy of duplicate data/content, cloud providers greatly reduce their storage and data/content transfer costs. The advantages of data/content deduplication unfortunately come with a high value in terms of new security and privacy challenges. This Proposed ClouDedup, a secure and efficient storage service which assures block-level data deduplication and data/content confidentiality at the same time. Even though based on convergent encryption, ClouDedup remains secure thanks to the definition of a component that implement an additional encryption operation and an access control mechanism. Furthermore, as the requirement for data deduplication at block-level lifts an problem with respect to key management, This suggest to include a new component in order to execute the keys management for each block together with the actual data deduplication operation.

This designed a system which achieves confidentiality and enables block-level data deduplication at the same time. This system is built on top of convergent encryption. This showed that it is worth performing block-level data deduplication instead of data level data deduplication since the gains in terms of storage space are not affected by the overhead of metadata management, which is minimal. Additional layers of encryption are added by the server and the optional HSM. Thanks to the features of these

DOI : 10.18231/2454-9150.2018.0983

components, secret keys can be generated in a hardware dependent way by the device itself and do not need to be shared with anyone else. As the additional encryption is symmetric, the impact on performance is negligible. This also showed that this design, in which no component is completely trusted, prevents any single component from compromising the security of the whole system. This solution also prevents curious cloud storage providers from inferring the original content of stored data/content by observing access patterns or accessing metadata. Furthermore, this showed that this solution can be easily implemented with existing and widespread technologies.

Finally, this solution is fully compatible with standard storage APIs and transparent for the cloud storage provider, which does not have to be aware of the running data deduplication system. Therefore, any potentially untrusted cloud storage provider such as Amazon, Dropbox and Google Drive, can play the role of storage provider. As part of future work, ClouDedup may be extended with more security features such as proofs of retrievability , data/content integrity checking and search over encrypted data/content In this paper This mainly focused on the definition of the two most important operations in cloud storage, that are storage and retrieval. These plans to define other typical operations such as edit and delete. After implementing a prototype of the system, this aim to provide a full performance analysis. Furthermore, this will work on finding possible optimizations in terms of bandwidth, storage space and computation.

III. EXISTING SYSTEM

Today's cloud service providers offer both highly available storage and massively parallel computing resources at relatively low costs. As cloud computing becomes prevalent, an increasing amount of data is being stored in the cloud and shared by users with specified privileges, which define the access rights of the stored data. One critical challenge of cloud storage services is the management of the ever-increasing volume of data. To make data management scalable in cloud computing, deduplication has been a well-known technique and has attracted more and more attention recently. Data deduplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. Instead of keeping multiple data copies with the same content, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy.

Although data deduplication brings a lot of benefits, security and privacy concerns arise as users' sensitive data are susceptible to both inside and outside attacks. Traditional encryption, while providing data confidentiality, is incompatible with data deduplication. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different ciphertexts, making deduplication impossible.

3.1 CRYPTOSYSTEMS WITH TWO SECRET KEYS

There are two kinds of cryptosystems that requires two secret keys for decryption. They are certificateless cryptosystem (CLC) and certificate-based cryptosystem. Certificateless cryptosystem was first introduced in and further improvements can be found . It combines the merits of identity-based cryptosystem (IBC) and the traditional public-key infrastructure (PKI). In a CLC, a user with an identity chooses his own user secret key and user public key. At the same time the authority (called the Key Generation Centre (KGC)) further generates a partial secret key according to his identity. Encryption or signature verification requires the knowledge of both the public key and the user identity. On the opposite, decryption or signature generation requires the knowledge of both the user secret key and the partial secret key given by the KGC. Different from the traditional PKI, there is no certificate required. Thus the costly certificate validation process can be eliminated. However, the encryptor or the signature verifier still needs to know the user public key. It is less convenient than IBC where only identity is required for encryption or signature verification. Similar to CLC, another primitive called certificate-based cryptosystem (CBC). Further variants may include . The concept is almost the same as CLC, except that the partial secret key given by the KGC (which is called the certificate) is a signature of the identity and the public key of the user by

DOI : 10.18231/2454-9150.2018.0983

the KGC. (Note that in CLC, the partial secret key given by the KGC is just the signature of the identity of the user.) Due to the similarities, CBC faces the same disadvantages as CLC mentioned above.

3.2 CRYPTOSYSTEMS WITH ONLINE AUTHORITY

Mediated cryptography was first introduced for the purpose of revocation of public keys. It requires an online mediator, referred to a SEcurity Mediator (SEM), for every transaction. The SEM also provides a control of security capabilities. If the SEM does not cooperate then no transactions with the public key are possible any longer. In other words, any revoked user cannot get the cooperation from the SEM. That means revoked users cannot decrypt any ciphertext successfully. Later on, this notion was further generalized as security mediated certificateless (SMC) cryptography. In a SMC system, a user has a secret key, public key and an identity. The user secret key and the SEM are required to decrypt a ciphertext or sign a message. On the opposite side, the user public key and the corresponding identity are needed for signature verification or encryption. Since the SEM is controlled by the revocation authority, the authority can refuse to provide any cooperation for revoked user so that no revoked user can generate signature or decrypt ciphertext. Note that SMC is different from our concept. The main purpose of SMC is to solve the revocation problem. Thus the SME is controlled by the authority and it has to be online for every signature signing and ciphertext decryption. Furthermore, it is not identity-based. The encryptor (or signature verifier) needs to know the corresponding

3.3 CRYPTOSYSTEM WITH SECURITY DEVICE

The paradigm of key-insulated cryptography was introduced and variants were proposed . There is a physically-secure but computationally-limited device in the system. A long-term key is stored in this device, while a short-term secret key is kept by users on a powerful but insecure device where cryptographic computations take place. Short term secrets are then refreshed at discrete time periods via interaction between the user and the base while the public key remains unchanged throughout the lifetime of the system. The user obtains a partial secret key from the device at the beginning of each time period. He then combines this partial secret key with the one from the previous period, in order to renew the secret key for the current time period. Different from our concept, key-insulated cryptosystem requires all users to update their key in every time period. It may require some costly time synchronization algorithms between users which may not be practical in many scenarios. The key update process requires the security device. Once the key has been updated, the signing or decryption algorithm does not require the device anymore within the same time period. While our concept does require the security device every time the user tries to decrypt the ciphertext. Furthermore, there is no key updating required in our system. Thus we do not require any synchronization within the whole system.

3.4 CRYPTOSYSTEM WITH REVOCABILITY

Since our system is an IBE-based mechanism, IBE-based systems supporting revocability. The first revocable IBE is proposed by in which a ciphertext is encrypted under an identity id and a time period T , and a non-revoked user is issued a private key skid;T by a PKG such that the user can access the data in T .proposed the security notion for revocable IBE. To achieve adaptive security, proposed a revocable IBE scheme based on the combination of attribute-based encryption and IBE. formalized a revised notion for revocable IBE. Since its introduction, there are many variants of revocable IBE,. The premise of a revocable IBE system is mainly related to a time period: next the decryption rights of the next time period relies on a secret token (for the next time period) issued by PKG and a current time period key. However, this premise yields inconvenience once the cur- rent time period key is lost.

3.5 DRAWBACKS

- However, previous deduplication systems cannot support differential authorization duplicate check, which is important in many applications.
- In existing data deduplication systems, the private cloud is involved as a proxy to allow data owner/users

DOI : 10.18231/2454-9150.2018.0983

to securely perform duplicate check with differential privileges.

- Existing prototype overhead is high in file upload operations.

IV. PROPOSED SYSTEM

In this project, aiming at efficiently solving the problem of Construction Roadmap. two different encryption technologies: one is IBE and the other is traditional Public Key Encryption (PKE). We first allow a user to generate a first level ciphertext under a receiver's identity. The firstlevel ciphertext will be further transformed into a second level ciphertext corresponding to a security device. The resulting ciphertext can be decrypted by a valid receiver with secret key and security device. Here, one might doubt that our construction is a trivial and straightforward combination of two different encryptions. Unfortunately, this is not true due to the fact that we need to further support security device revocability. A trivial combination of IBE and PKE cannot achieve our goal. To support revocability, we employ re-encryption technology such that the part of ciphertext for an old security device can be updated for a new device if the old device is revoked. Meanwhile, we need to generate a special key for the above ciphertext conversion. We also guarantee that the cloud server cannot achieve any knowledge of message by accessing the special key, the old ciphertext and the updated ciphertext. We further use hash-signature method to "sign" ciphertext such that once an component of ciphertext is tempered by adversary, the cloud and ciphertext receiver can tell. From the above presentations, we can see that our two-factor protection system with security device revocability cannot be obtained by trivially combining an IBE with a PKE.

SETUP PHASE: the setup phase generates all public parameters and master secret key used throughout the execution of system. The public parameters are shared with all parties participating into the system (including data sender/receiver, cloud server and a PKG), while the master secret key is given to the PKG.

KEY AND DEVICE ISSUED PHASE: A SDI and a PKG will respectively generate a security device and a secret key for a registered user IDi in secure channel such that the user can combine the security device with the secret key to recover message from its encrypted format.

FIRST-LEVEL CIPHERTEXT GENERATION PHASE: A data sender encrypts a data under the identity of a data receiver, and further sends the encrypted data to the cloud server. Knowing public parameters param, a data and a receiver's identity IDi , a data sender encrypts a data to a first level encryption

SECOND-LEVEL CIPHERTEXT PHASE: After receiving the first- level ciphertext of a data from the data sender, the cloud server generates the second-level ciphertext. Knowing public parameters param, a first level encryption for the user, and the information (IDi , tpki) stored in List, the cloud server encrypts.

4.1 STORING AND ACCESSING OWN DATA

Once user is authenticated to the Cloud Server, user can access the file storage and can upload any type document in the cloud storage.

Here the file is first encrypted before uploading and the same is decrypted at the time of downloading. Or user can simple store original format file in common folder which he/she wants to share with other authenticated user directly without worrying about key sharing mechanism

4.1.1 UPLOADING ENCRYPTED FILE

If user is authenticated then cloud server will load Emodule to clients end to perform encryption operation. Here client upload encrypted file on cloud server private folder using symmetric key encryption technique.

At the time downloading encrypted file user will ask to provide the decryption key if key is valid then only file will get downloaded at clients end. This encryption and decryption of data will be done at client side by making use of a symmetric key so it is not possible for CSP to gain access to key so even if the data stored is in encrypted format and the algorithm used to encrypt it is available to cloud, it is difficult to decrypt it. User is assured about security of data stored in cloud. This ensures data privacy of private compartment.

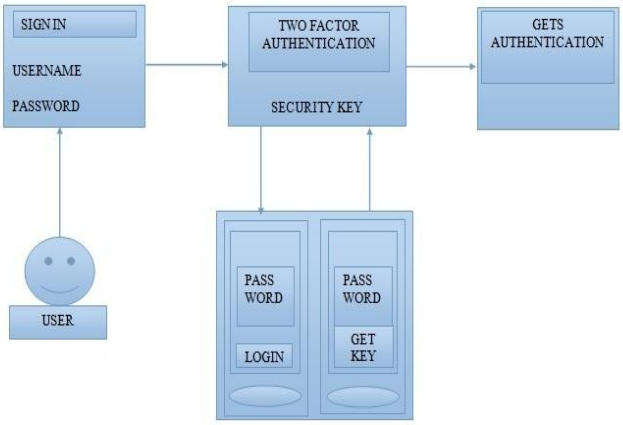
4.1.2 UPLOADING PLAINTEXT FILE

DOI : 10.18231/2454-9150.2018.0983

At the time of uploading plain text file user need not worry about encryption. Here cloud will load Emodule to clients end upon request and then user can select file to upload. User can store file to either common folder or private folder. At the time of downloading the file user can simply request file without worrying about decryption key.

4.2 LZ4 ALGORITHM

The greater part of the issues in the cloud is security, so we are taking one public cloud. We are taking one association that contains the distinctive clients/individual users. The administrator transfers the information documents and data is compressed and encrypted format**. LZ4 algorithms** are used for compression and encryption. We are using compression for less storage space and increasing security. This encrypted data will be stored on cloud. The key will be generated. At the point when the client need information, he ask for the Edu-admin, Admin check the client is approved one or not, and if client is approved then key will be sent to client mail. We proposed safe information sharing plan, which can accomplish secure key circulation and information sharing. The fundamental commitments of over plan are to protect path for key conveyance few secure similar channels. The clients get the keys safely from administrator, when the client is authorized one, then the client will receive the key. Our plan is to accomplish fine –grained to get control with the assistance of the gathering client list, any authorized client can utilize the sources in the cloud. The information measure like decreased by size of record, so that security increases



Architecture Diagram

V. LZ4 ALGORITHM

The LZ4 algorithm takes the given text data as a series of sequence .

Step 1: Each one starts with a one byte (8 bits) field that is separated by two half bit tokens. Step 2: The first field shows the number of literal bytes that are copied to the output.

Step 3: The second field shows that the number of bytes to copied to the already decoded output (with 0 represents the match length to 4 bytes).

Step 4: If the value of it field length is larger than 15 then add extra one byte of data to the length.

Step 5: Similarly, if the value of string length is larger than 255 then add extra one byte to the string length. If the value is less than 255 then copy same to the output.

ADVANTAGES

- Our system is designed to solve the differential privilege problem in secure deduplication.
- In our system, the S-CSP is honest but curious and will honestly perform the duplicate check upon receiving the duplicate request from users.
- In our system a higher level confidentiality is defined and achieved.
- Our authorized duplicate check scheme incurs minimal overhead compared

VI. RESULT AND DISCUSSION

Implementation is stage in the report where the theoretical design is turned into the working system. The most crucial stage is giving the users confidence that the new system will work effectively and efficiently. The performance of reliability of the system is tested and it gained acceptance.

The whole system falls into two subsystems under dynamic data sharing; the two are evaluated individually because of different evaluation criteria. For the security subsystem, we made a comparative study on authentication, accuracy, storage and response between our system and existing system. We presented the design and implementation of an automated dynamic authentication management system that achieves a good

DOI : 10.18231/2454-9150.2018.0983

balance between the user authentication and deduplication efficiency.

We discussed the key issues for proposed system implementation, including private cloud support (registration and authentication), dynamic multi public cloud management and optimization of data sharing with deduplication models. Evaluations are performed on a prototype system. We achieved some more features which described in terms of performance as follows,

MODULARITY AND EXTENSIBILITY: Modularity and extensibility are closely related to each other. In achieving tight cohesion as well as loose coupling, the roles embedded in our system have independent functions and can be integrated into most cloud computing environments as an independent subsystem. Besides, information generated or passed is designed in file format in order to support new resource types and interact with other components. However under these features our proposed work exhibited more security and utilization.

TRANSPARENCY: Cloud users do not need traversal of all the nodes or cloud expertise to get information. We design a uniform and friendly interface component for accessing the information authenticated and shared. Thus client query response time is reduced when compared the existing system with our proposed system.

DATA PROTECTION:

- Privileges and POW based access control lists to define the permissions attached to the data sharing.
- Storage encryption to protect against unauthorized access at the data center (especially by malicious access).
- Transport level encryption to protect data when it is transmitted.
- Hardening of the cloud multi public servers to protect against known, and unknown, vulnerabilities in the file request and maintenances.

High accuracy and efficiency is the primary design goal of the secure data sharing among dynamic subsystem. Thus goals are achieved by our proposed work when compared existing work. Data sharing, deduplication and collaboration in the cloud is still currently a strong focus of research today and in particular many works are focusing on solving the user revocation problem as well as ways to manage the sharing and collaboration of large data sizes. Our experimental results indicated that the efficiency and accuracy of our system meet the demand of online system for cloud data sharing and deduplication based storage management.

VII. CONCLUSION

In this work, the examination is based on the notion of authorized data deduplication was proposed to protect the data security by including differential privileges of users novel two-factor data security protection mechanism for cloud storage system, in which a data sender is allowed to encrypt the data with knowledge of the identity of a receiver only, while the receiver is required to use both his/her secret key and a security device to gain access to the data. Our solution not only enhances the confidentiality of the data, but also offers the revocability of the device so that once the device is revoked, the corresponding ciphertext will be updated automatically by the cloud server without any notice of the data owner . As proof of idea, we showed that our secure proposed system incurs to reduce the bandwidth, storage capacity, and also reduces amount of response time.

VIII. FUTURE WORK

Cloud computing promises to increase the velocity with which applications are deployed. Thus the proposed system in multi public cloud structure deals with reducing the storage space and providing privacy for users. The future research scheme provides privacy and complexity while handling the data sharing over cloud. So in future we reviewed current state-of-the-art literature in relation to secure and dynamic data sharing in the cloud and gave a brief overview on the future of data sharing in the cloud where the data owner could have more control over the usage of their data.

DOI : 10.18231/2454-9150.2018.0983

REFERENCE

[1] OpenSSL Project, (1998). [Online]. Available: http://www.openssl.org/

[2] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de-duplication," in Proc. 24th Int. Conf. Large Installation Syst. Admin., 2010, pp. 29–40.

[3] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: Serveraided encryption for deduplicated storage," in Proc. 22nd USENIX Conf. Sec. Symp., 2013, pp. 179–194.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in Proc. 32nd Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2013, pp. 296–312.

[5] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," J. Cryptol., vol. 22, no. 1, pp. 1–61, 2009.

[6] M. Bellare and A. Palacio, "Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks," in Proc. 22nd Annu. Int. Cryptol. Conf. Adv. Cryptol., 2002, pp. 162–177.

[7] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider, "Twin clouds: An architecture for secure cloud computing," in Proc. Workshop Cryptography Security Clouds, 2011, pp. 32–44.

[8] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in Proc. Int. Conf. Distrib. Comput. Syst., 2002, pp. 617–624.

[9] D. Ferraiolo and R. Kuhn, "Role-based access controls, " in Proc. 15th NIST-NCSC Nat. Comput. Security Conf., 1992, pp. 554–563.

[10] GNU Libmicrohttpd, (2012). [Online]. Available: http://www. gnu.org/software/libmicrohttpd/

[11] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in Proc. ACM Conf. Comput. Commun. Security, 2011, pp. 491–500.

[12] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," in Proc. IEEE Trans. Parallel Distrib. Syst., http:// doi.ieeecomputersociety.org/10.1109/TPDS.2013.284, 2013.

[13] libcurl, (1997). [Online]. Available: http://curl.haxx.se/libcurl/

[14] C. Ng and P. Lee, "Revdedup: A reverse deduplication storage system optimized for reads to latest backups," in Proc. 4th Asia- Pacific Workshop Syst., http://doi.acm.org/10.1145/2500727. 2500731, Apr. 2013.

[15] W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage," in Proc. 27th Annu. ACM Symp. Appl. Comput., 2012, pp. 441–446.

DOI : 10.18231/2454-9150.2018.0983