A Lightweight Secure Data Sharing Scheme for Mobile Computing

Contents

Acknowledgement

List of figures

Glossary

ABE – Attribute based encryption

CP – Chipertext policy

CSA – Cloud service provider

DO – Data Owner

DU – Data User

ESA – Encryption Service provider

LDSS – Lightweight Secure Data Sharing Scheme

SA - Security Analysis

TA – Trust Authority

Abstract

With the invention and implementation of cloud computing capabilities, it is now possible for personal and organizational mobile devices to perform storage and retrieval of data regardless of the physical location. However, it is consequent that the challenges that the cloud faces in terms of data security have developed and become worse, thereby preventing any further development and sophistication of the mobile computing. It is evident that there have been researches and studies that have focused on improving the security state of the cloud. However, it is clear that most of the researches and studies have not been successful due to the fact that mobile devices operate under limited resources and computing power. It is highly recommended that solutions which contain relatively low computational overhead be implemented. This study proposes a lightweight data sharing scheme (LDSS) to be applied in the mobile computing sector. The scheme adopts the CP-ABE technology that has been mostly used for normal cloud computing environments, only this time the access structures have been altered to favor mobile cloud computing environments. Through the use of the scheme, the CP-ABE is moved from the mobile devices to now reside in the proxy servers located externally. The scheme also proposes to introduce the lazy-revocation approach which is aimed at reducing the costs for user revocation. Through the experiments conducted and the results displayed, it is evident that LDSS can lead to a recognizable reduction in the overhead on the side of the mobile computing devices in scenarios where users share data through the cloud.

1. Introduction

In the recent past, the cloud computing platform together with the use of mobile devices have become the talk of the town as far as technological development is concerned. People and organizations have become accustomed to the new way of operation, often called the digital age where a huge percentage of data is stored in cloud based systems and access to the resources is made available through mobile devices. Due to the fact that mobile devices often have limited amounts of storage capacities and processing power, it is essential to complement this shortcoming by using the cloud service providers to perform basic storage processes and processing (Kousalya et al., 2018). Through the use of the cloud services, people and entities have the capability of sharing data files and other types of media. It is evident that through the demand and requirement for data security, cloud service providers often provide provisions where users have the options of choosing the security protocols to use when handling their information. However, the security provisions are not efficient enough due to the fact that users have limitation when it comes to the extent in which they interact with the mobile device platforms.

To address some of the issues stated regarding the use of mobile cloud computing devices, it is vital to exploit the option of encryption of data and information before it is availed to the cloud service providers. However, this approach creates a new problem and challenge where complexity involved in the exclusive exchange of cipher texts for decryption process and the need for users to be granted certain user privileges. Amidst the options given, it is often preferred to accord several assumptions regarding the cloud service providers. These assumptions include the trust that the service providers are honest, that data is encrypted before it is availed to the cloud, that the cloud access system employs effective authentication. These assumptions can be

summarized into several technical categories including cipher text access control, hierarchical, homomorphic and attribute-based access control (Dheepa et al., 2018).

The growth and development of the mobile computing technology calls for opinions, researches and proposals to address the ever lurking security concerns. In this study, we propose the LDSS approach to be used for the mobile computing platforms. There are several contributions that the LDSS approach brings on the table. First, the study designs and introduce the LDSS-CP-ABE method to accord access control with regard to the use of cipher texts for encryption (RESHMA & VEMULA, 2018). Secondly, the technique exploit the services of the proxy servers to be used in the encryption and decryption processes. Thirdly, we propose the introduction of the lazy re-encryption systems and decryption system to minimize on the revocation experienced. Finally, the study implements a prototype and framework of the LDSS operation.

2. Background Literature

This section bases on the various researches and works that relate to cipher text schemes that have a direct relation to the study's proposed scheme.

Access control is very essential in the operation of any information technology system to ensure data privacy and security so that only the legitimate and authorized users can access the protected resources (Dheepa et al., 2018). Over time, there have been several studies and research pertaining the access control policies in the mobile cloud computing platforms. On typical terms, the cloud service and platform is regarded to be honest and curious. It is thus vital to ensure that all the data files are encrypted before they are uploaded onto the cloud. User authorization, on the other hand, is often achieved through the distribution of various keys in the system. The entire research can be categorized into simple cipher text access control, hierarchical, homomorphic, and attribute-based encryption systems.

All the above works and research place their focus and attention towards the control of access to files and resources on the cloud. Most of the solutions and frameworks identified above are applied in normal cloud computing environments but not the mobile cloud computing environment. However, there have been researches that have deeply touched on the aspect of access control in the mobile computing environment. An example of such a research is that of MobiCloud where the traditionally existing MANETs network system is transformed into a new and better framework which is service oriented (Bhavani, 2018). In the operation of this new approach and framework by MobiCloud, each mobile device and entity in the system is specifically regarded to be a service node. However, one aspect about the use and implementation of MobiCloud is the fact that the users need to have complete trust to the cloud service providers on matters of privacy and security.

In summary of the related research articles and proposals, it is evident to note that most of the studies propose techniques and mechanisms of addressing access control in the normal cloud environments but not the mobile cloud environment. It is also evident that most of the proposals have not addressed the issue about privilege change, thereby leading to extremely high costs of revocation. This scenario is not directly applicable and relatable to that of the mobile devices due to their limited computational power and storage capabilities. For the cases of the studies that have touched on mobile cloud computing, they have not addressed the issue about the credibility of the servers and service providers due to the assumption of the honesty of the providers which is often not the case in reality. In this study, we propose the LDSS scheme which adopts the operation of CP-ABE technology to manage the access control policies on the cloud, making it suitable to be used by mobile devices. Application of LDSS approves of its safety and security in the mobile cloud computing environment and clearly demonstrates its efficiency and scalability when it comes to ABE schemes.

3. Existing Algorithm and techniques

This section of the study presents several technique preliminaries that relate to LDSS with further presentation of the entire system model with assumptions.

3.1 Bilinear Pairing

The bilinear function is defined using the function e:G0*G0=G1 where G0 and G1 originate from the prime order of p and can be multiplied. The e in the function possesses the properties of being bilinear, non-degeneracy, and computable. In most of the implementations, G0 is often taken as a set of points located on the elliptic curve, G1 as a subgroup and e as a Weil.

3.2 Attribute Based Encryption

Attribute Based Encryption derives from the Identity Based Encryption model that operates for a one-to-many scenario for sharing data on the cloud. ABE encryption is divided into Ciphertext ABE and Key Policy ABE. Users often choose the encryption model that bests suits their specifications and requirements for access control.

3.3 Secret Sharing Scheme

This scheme specifically functions to secure data and information that is deemed to be secret. Thereare some pre assumption related to securitywhich are:-

I. Semi-Trusted Server - In the operation of LDSS, it is evident to note that its operation is similar to that of cloud service providers where they providers are honest but in some way curious (Dheepa et al., 2018). This means that the providers need to execute the demands and requirements set by the users, but have the capability of peeping into the information. In the operation of LDSS, it is also vital to note the

introduction of two servers (proxy encryption and proxy decryption servers) which will aid in the process of decryption and encryption with the intention of minimizing overhead incidences.

II. Trusted Authority- In order to make the proposed LDSS system feasible, it is vital to introduce a trusted authority. The TA operates to generate and introduce public and private keys to the users of the system. Through this functionality, it is possible for the users to access cloud resources and share them without necessarily being aware of the encryption processes underway.

III. Lazy Re-encryption - There are cases where some privileges regarding user data are revoked. In normal operation, such cases demand for re-encryption of the user data which may end up causing delays and overhead in the operation of the encryption scheme. To address the issue, lazy re-encryption comes in handy to offer the functionality that data will only be re-encrypted when the users have updated their data.

4. Proposed Mechanism

This section presents the LDSS system design where the first sections gives the overview of the systems and presents the system's operations and then later discusses the LDSS in detail.

4.1 Overview

The proposed LDSS system is generally composed of six components. First, the Data Owner (DO) is responsible for uploading data to the mobile computing cloud platform and shares the data with other users (Sushmitha et al., 2018). The DO is also tasked with the responsibility of defining the access control demands and policies. Secondly, the Data User (DU) operates by retrieving data from the cloud platform. Thirdly, the Trust Authority (TA) generates and distributes public and private keys to the users. Fourthly, the Encryption Service Provider (ESP) makes the data encryption services and operations essential for the DO. Fifthly, the Decryption Service Provider (DSP) is responsible for the decryption services and processes for the DU. Lastly, Cloud Service Providers (CSP) provides the storage capabilities for the DO and performs operations on the data as required or executed by the DO.

Through the illustration in the figure below, the DO sends data and files to the mobile cloud platform. Due to the credibility issues of the cloud, it is vital for the data to be encrypted first before it is uploaded. Access control protocols and policies are defined by the user in the form of the tree where requirements and measures are put in place to determine the qualifications for the DU to retrieve the data. In the use of LDSS, the symmetric encryption method is used to encrypt all the files in the system, (Valarmathi, 2019). The access control policies for the users are embedded in the symmetric key's cipher text. Only the DU's that satisfy the requirements and qualifications demanded will have access to the system's resources. ESP and DSP are

functional to reduce overhead experiences. The LDSS-CP-ABE is modified to accord the necessary data privacy and security during the computational processes of ESP and DSP.

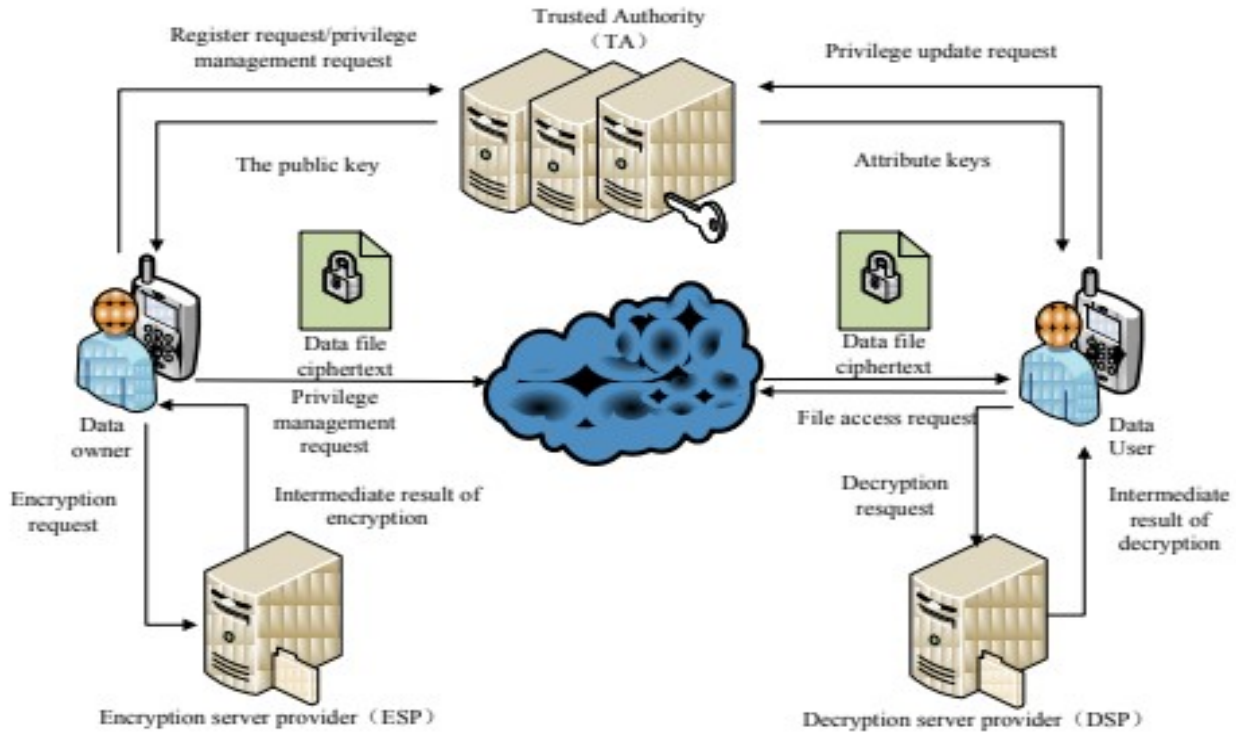

Fig 1 – Architecture of ncryption and decryption algorithms

4.2 The LDSS-CP-ABE Algorithm

To successfully conduct an illustration of the LDSS-CP-ABE algorithm, it is vital to perform basic definition of terms.

I. Attribute:

The attribute is responsible for defining the nature of access privilege that a [articular data file possesses. Attributes are given to data users through the specifications by data owners. In availing the data to the cloud, the data owners are required to set a certain set of attributes and

control policies to manage the access and retrieval of the data by data users.

II. Access Control Tree:

The access control tree caters for the specific representation and expression of the access policies defined by data owners (Jyothi & Sulthana, 2017). The tree is made up leaves and non-leaves nodes that represent attributes and relational operators respectively. Each of the nodes of the access control tree contains a secret which can be further divided and split up into several secrets to the nodes that lie below them.

III. Version attribute:

The version attribute appears in the implementation of the LDSS-CP-ABE algorithm to enhance the security of the algorithm. It operates as an addition to the access control tree that exists originally thereby forming a new root.

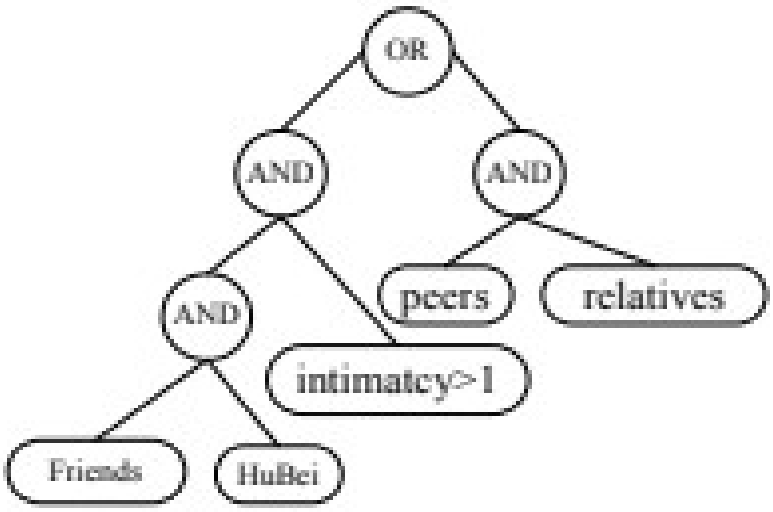

Fig 2 – Control Access tree

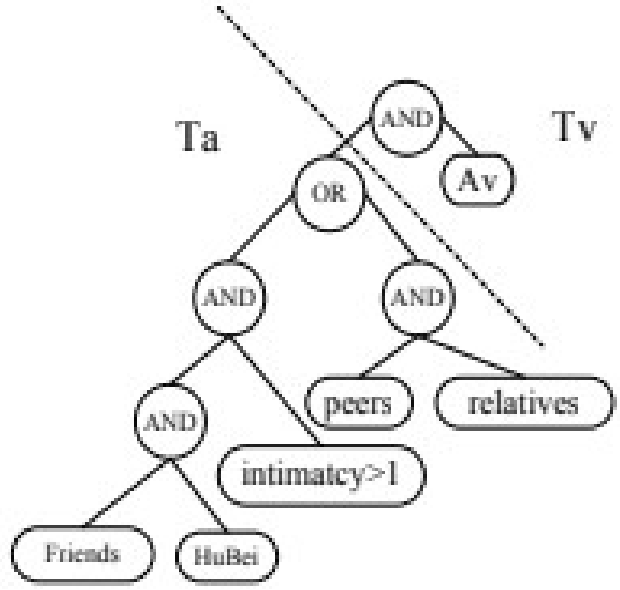

Fig 3 Access control tree with several version attributes

Through the definition of the above terms, the LDSS-CP-ABE algorithm operates with inclusion of the following basic sub-functions. First, the Setup (A,V) is responsible for generating the final master key MK, the algorithm's public key PK, and final version attribute V. It is essential to note that this sub-function is automatically called by the trusted authority to ensure successful encryption of the data files. Secondly, the KeyGen (Au, MK) is tasked with generating various attribute keys for use by the data users relative to the attribute and the master keys set. Thirdly, the Encryption sub-function (K, PK, T) operates by generating the functioning cipher text CT with basis on the algorithm's symmetric key, access control tree and the public key. Lastly, the Decryption sub-function (CT, T, SKu) conducts a decryption of the entire ciphertext through the use of the access control tree and the several attribute keys presented.

4.3 ADF in the operation of LDSS-CP-ABE

The decryption field is brought into the operation of the LDSS algorithm to spearhead the implementation and realization of a dynamic nature of user privilege management. It is the intention and goal of the field to ensure that the access control strategy is safe and secret against the entire cloud. To clearly understand the operation of the attribute description field, it is vital to have an understanding on the basic definitions on basic terms.

Attribute definition field: This is a string that is made up of binary bits. The string describes information regarding the structure and operation of the attributes managed by the DO, DU and the various data files in the cloud.

Attribute description bit: This takes to account each and every bit that exists in the description field with correspondence to each attribute.

With regard to the several string components for the attribute description field, there are three major kinds of ADFs, with each of the kinds either related to the data owners, data users, and/or the data files themselves. The ADF related to the data owners is generated by the third party trusted authority while that of the data users is generated both the trusted authority and the mobile cloud with the data owner being the supervisor of the entire process. For the case of the ADF for data files, most of the information regarding the description is stored by the data owners.

5. Algorithm Implementation

LDSS is specifically designed to accord the mobile computing cloud an effective data sharing functionality. The entire process included in the scheme takes to account system initialization, sharing of files, authorization protocols, and file access protocols (Bhavani, 2018). It is also vital for the scheme to support basic revocation processes and file updated processes.

5.1 System Initialization

The process of system initialization is realized through the initialization of Function 1 through a series of steps described below. First, when the DO performs a registration on the TA platform, the TA consequently executes the Setup() function which then generates a master key with an additional public key (Valarmathi, 2019). The public key is then sent to the specific data owner while the trusted authority keeps the master key to itself. Secondly, the data owner performs a definition of attribute sets where each of the attributes selected are assigned to their contacts. Information regarding this assigning process is then sent over to the trusted authority and the mobile cloud. Lastly, the trusted authority and the mobile cloud now have access to the information on attributes and definitions presented by the DO; they receive the information and store it accordingly.

For file sharing to be successful in the scheme, Function 3 needs to be incorporated to ensure effective encryption of the data files present. The file sharing process is as follows. First, the data owner performs their selection of the data files and content that is to be uploaded and then encrypts the files through symmetric encryption by making use of the symmetric key K and ending up generating cipher text C (Jyothi & Sulthana, 2017). Secondly, the data owner then assigns the access control policies to the files through the use of the Function 3 ending up generating a cipher version of K (CT).

5.2 User Authorization

The entire process regarding user authorization makes use of the Function 2 in generating attribute keys to be used by the DUs. The entire process flows as follows. First, the data users login into the cloud system and the request authorization to access the files. It is vital, at this stage, to note that some of the attribute keys (SK) are already in possession of the data users. Secondly, the trusted authority takes note of the request and performs a check to ascertain whether the data user has logged on the system before. Cases where the users have accessed the system before demand to go to the third step, otherwise jump to the fourth step. Thirdly, the trusted authority then summons the execution of Function 2 for the generation of attribute keys. Fourthly, the trusted authority performs a comparison of the ADF in the present attribute key with that in the system's database. If the ADFs match, then go to the fifth step, otherwise proceed to the sixth step. Fifthly, if the ADFs bit is revoked then the trusted authority does nothing but when the ADFs bit is revoked, the trusted authority then ends up generating attribute keys for the data users. Sixthly, the trusted authority performs a check on the various versions of attributes used by the data users. When the check results ascertain that the keys are not the same with the prevailing version, they keys are then automatically updated.

5.3 Access Files

When data users request to have access to certain sets of data files, Function 4 comes in handy to offer the decryption protocols for the access process. First, the data users need to send the access request to the mobile cloud. Secondly, the mobile cloud services through their systems check to determine whether the requesting data user meets the set requirements. If the data user fails to meet the requirements set for access control, the cloud services do not approve the request, otherwise the data user is availed with the ciphertext. Thirdly, the data user then receives the cipher text from the cloud

and proceeds to perform the decryption process through the use of Function 4. Lastly, the data user then accesses the decrypted files after using the symmetric key to the cipher text contents of the data files.

5.4 Privilege Revoked

It is evident and possible to acknowledge the fact that data owners can perform revocation from the data users. First, the data owners contact the trusted authority and informs them of one attribute clause which has been revoked by the data user. Secondly, the trusted authority together with the cloud services systems update the information into their databases. Thirdly, the data owners then mark the various selected bit of the attribute's ADF to the data files.

5.5 Documentation Updates

It is important to acknowledge the existence and implementation of the lazy re-encryption mechanism such that when a data owner has revoked an attribute from the side of the data user, the attribute is not aptly updated first. However, when the file is updated, possession of a single revoked file calls for the update of the attribute. First, it is the duty of the data owner to check and determine whether there is any bit ADF that is set to #. Secondly, the data owner then contacts the trusted authority and avails the details on the attributed to be updated. Thirdly, the trusted authority then performs the update process where the new specifications are attached to the related attributes as provided for by the data owners. Fourthly, the trusted authority then sends the new public key to the data owners who process the key and perform encryption on the files using the new public key. Lastly, the data owners set # as a bit representation of the ADF for the data files.

6. Security Analysis

The security assessment section is indicated with basis on the various assumption made in the third section where possibilities of malicious users exposing plaintext is not dealt with.

6.1 Security Analysis of LDSS-CP-ABE

LDSS-CP-ABE algorithm is specifically designed to operate on top of the ABE algorithm whose security has basis on the bilinear diffie-hellman set of assumptions (Jyothi & Sulthana, 2017). Due to the fact that the operation of LDSS-CP-ABE has a variation of the original BSW CP-ABE, the encryption and decryption processes of the scheme are thus safe and secure. The only difference that exists between the study's proposed scheme relative to the BSW CP-ABE scheme is that with the LDSS based, a version attribute is used which then alters the access control tree slightly.

6.2 Data Confidentiality against Conspiracy

Factors regarding the confidentiality of data in the proposed scheme can be broken down into two major aspects. First, it is evident that data files in the scheme are encrypted using the symmetric approach (Halevi, 2017). This implies that the data files are additionally protected under the additional security mechanisms in the symmetric encryption system. Secondly, the symmetric key is further encrypted through attribute encryption. However, with this second case, the security of the process is highly dependent on the entire process of encryption. In this section, the discussion surrounds the safety of the data files even with scenarios where malicious users and entities conspire to acquire the keys. Such conspiracy attacks revolve around various users of the cloud, DSP and ESP.

6.3 Confidentiality of Access Control Policy

The security requirements regarding the access control policies determine that none other than the data owners should be in possession of the specific content and details of the policy. The proposed scheme exploits the services provided by ADF so that in the use of access control policy, descriptions are made based on the ADF bit. This operates by only allowing the cloud service systems and ESP to have access to the various ADF bits but not access to the exact content. This thus protects the entire access control policy.

Conclusion

In the recent past, there have been myriads of research and studies pertaining access control specifically relating to attribute based encryption. However, there have been a constant challenge regarding the applicability of the algorithms to the mobile cloud computing environment due to the fact the mobile devices often possess limited computing power, storage and other key resources for computing devices. In this research, the LDSS scheme is proposed to try and address the issue. The approach starts by incorporating the capabilities of the CP-ABE algorithm which is responsible for transferring the overhead for computation from the mobile device systems to the proxy server systems. This move makes it possible for the devices to share the necessary resources on the cloud. In the future of the research, we will look into deeper matters that relate to data integrity to enhance the privacy of the data files on the cloud. Future works will also relate in the application of cipher text retrieval mechanisms in the existing sharing schemes to offer a more scalable approach in addressing the concerns on access control.

References

Bhavani, S. D. (2018). Achieving an Effective Lightweight Secure Data Sharing Scheme for Mobile Cloud Computing.

Dheepa, K., Saravanan, D., Parthiban, R., & Kumar, K. S. (2018). SECURE AND FLEXIBLE DATA SHARING SCHEME BASED ON HIR-CP-ABE FOR MOBILE CLOUD COMPUTING. *International Journal of Pure and Applied Mathematics*, *119*(14), 823-828.

Halevi, S. (2017). Homomorphic encryption. In *Tutorials on the Foundations of Cryptography* (pp. 219-276). Springer, Cham.

Jyothi, P., & Sulthana, S. A. FRIVOLOUS CLOSED PROTECTED DATA SHARING SCHEME IN MOBILE CLOUD COMPUTING.

Kousalya, D., Prasanna, G., & Parimala, A. J. IMPLEMENTING AN EFFICIENT METHOD FOR SHARING DATA IN MOBILE CLOUD COMPUTING.

RESHMA, M., & VEMULA, M. V. R. A LIGHTWEIGHT PRIVACYDATA DISTRIBUTED SCHEME FOR MOBILE CLOUD COMPUTING.

Sushmitha, B., Harsha, K. S., Srilatha, N., & Sneha, T. (2018). A Novel Approach Secure Data Sharing for Mobile Cloud Computing.

Valarmathi, J. A Lightweight Secure Data Sharing Scheme for Mobile Cloud Computing.