AN EFFICIENT FILE HIERARCHY ATTRIBUTE-BASED ENCRYPTION SCHEME IN CLOUD COMPUTING

C.Ramadas[1], Arun[2]
[1,2]Computer Science and Engineering, Indira Institute of Engineering and Technology, Chennai, India

ABSTRACT

Cloud computing is one of the most promising application platforms to solve the explosive expanding of data sharing. In cloud computing, to protect data from leaking, users need to encrypt their data before being shared. It enables customers with limited computational resources to outsource their large computation workloads to the cloud, and economically enjoy the massive computational power, bandwidth, storage, and even appropriate software that can be shared in a pay-per-use manner. Access control is paramount as it is the first line of defense that prevents unauthorized access to the shared data. On the one hand, the outsourced computation workloads often contain sensitive information, such as the business financial records, proprietary research data, or personally identifiable health information etc. To combat against unauthorized information leakage, sensitive data have to be encrypted before outsourcing so as to provide end to- end data confidentiality assurance in the cloud and beyond. Clients risk key exposure that is hardly noticed but inherently existed in previous research. Furthermore, enormous client decryption overhead limits the practical use of ABE. The proposed collaborative Mechanism effectively solves not only key escrow problem but also key exposure. Meanwhile, it helps markedly reduce client decryption overhead. However, ordinary data encryption techniques in essence prevent cloud from performing any meaningful operation of the underlying cipher text-policy, making the computation over encrypted data a very hard problem. The proposed scheme not only achieves scalability due to its hierarchical structure.

INTRODUCTION

In cloud computing, authority accepts the user enrollment and creates some parameters. Cloud service provider (CSP) is the manager of cloud servers and provides multiple services for client. Data owner encrypts and uploads the generated ciphertext to CSP. User downloads and decrypts the interested ciphertext from CSP. The shared files usually have hierarchical structure. That is, a department of files are divided into a number of hierarchy sub departments located at different access levels. If the files in the same hierarchical structure could be encrypted by an integrated access structure, the storage cost of ciphertext and time cost of encryption could be saved. Presently a day's more number of plans utilized encryption for control the information in Cloud. It empowers clients with restricted computational assets to outsource their expansive calculation workloads to the cloud, and monetarily appreciate the monstrous computational power, data transfer capacity, stockpiling, and even proper programming that can be partaken in a compensation for each utilization way. Distributed computing is a progressive registering worldview which empowers adaptable, on-request and minimal effort utilization of figuring assets. Those points of interest, unexpectedly, are the reasons for security and protection issues, which rise in light of the fact that the information claimed by various clients are put away in some cloud servers rather than under their own control. The security issue of distributed computing is yet to be settled. To manage security issues, different plans in light of the Attribute-Based Encryption have been utilized. From one perspective, the

outsourced figuring workloads often contain sensitive information, for instance, the business money related records, prohibitive research data, or eventually identifiable prosperity information et cetera. To fight against unapproved information spillage, sensitive data must be mixed before outsourcing so as to offer end to-end data protection affirmation in the cloud and past. Regardless, normal data encryption procedures by and large shield cloud from playing out any critical operation of the essential figure content game plan, making the count over encoded data a troublesome issue. The proposed plot not simply achieves flexibility due to its dynamic structure. We give the protection secure out in the open social distributed computing. In our venture we actualize progressive property base security the pecking orders are Cloud specialist, Domain expert and clients. Cloud expert can just have benefit to make or expel the domain (private cloud specialist) in cloud and they can keep up every one of the points of interest in general cloud Domain expert can make or evacuate the clients inside the area this clients are called private clients. Clients are two sorts private cloud client and open cloud client's Private cloud clients are depends the space Public clients under cloud specialist. Clients can transfer the documents in two ways: Public and Private. On the off chance that the private client transfer general society document, the record perceivability and availability is just inside area itself and same space clients can get to that document with no security validation If the general population client transfer people in general document, the record perceivability and openness is constantly open any cloud client can get to that document. For Private transfer If private client transfer the private document implies that record perceivability is just inside space yet document openness is who have the emit key (OTP) implies who have benefit to get to the record If general society client transfer the private document implies that document perceivability is open anybody can obvious the document yet who have a benefit (OTP) to get to they just can get to the document.

OBJECTIVE

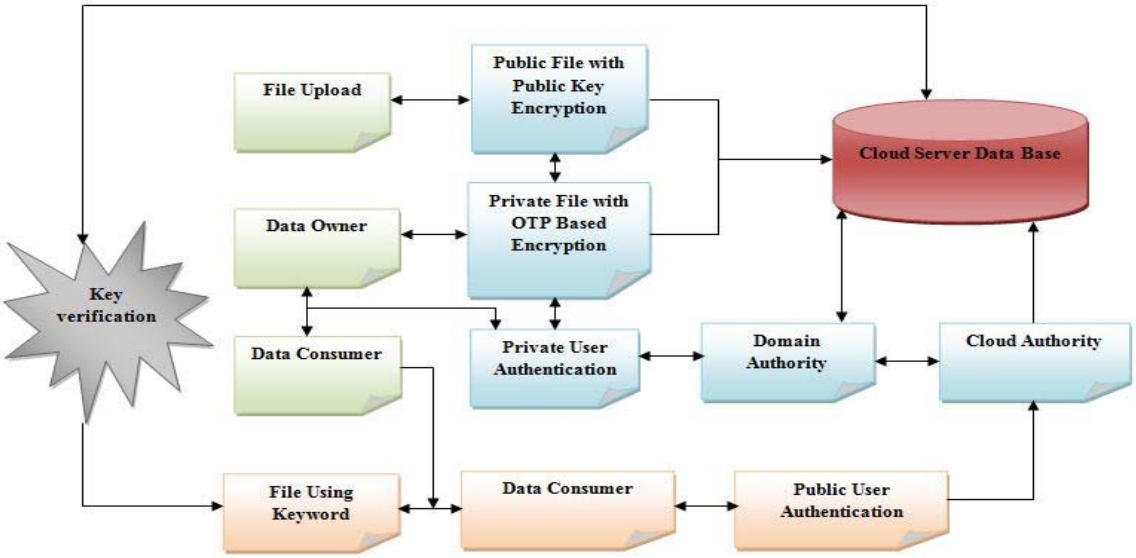
To realize scalable, flexible and fine-grained access control of outsourced data in cloud computing. The outsourced computation workloads contain sensitive information such as business financial records, proprietary research data or personally identifiable health records etc. Users may try to access the data files outside their privileges. Hence a hierarchy is proposed where a particular department of users trusts a domain authority. The domain authority in turn trusts the trusted authority.

SCOPE

We provide the privacy secure in public social cloud computing. In our project we implement hierarchical attribute base security the hierarchy are Cloud authority, Domain authority and users. Cloud authority can only have privilege to create or remove the domain(private cloud authority) in cloud and they can maintain all the details in overall cloud Domain authority can create or remove the users inside the domain this users are called private users . Users are two types private cloud user and public cloud user’s Private cloud users are depends the domain Public users under cloud authority.

SYSTEM DESIGN

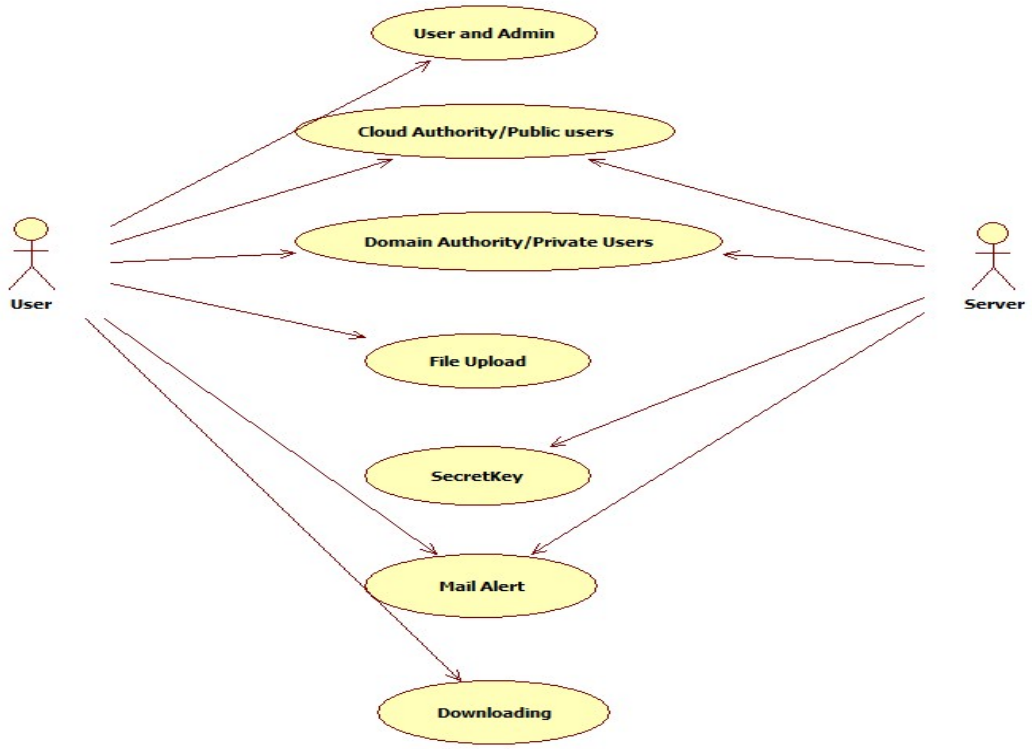
SYSTEM ARCHITECTURE



UML DIAGRAM

USE CASE DIAGRAM

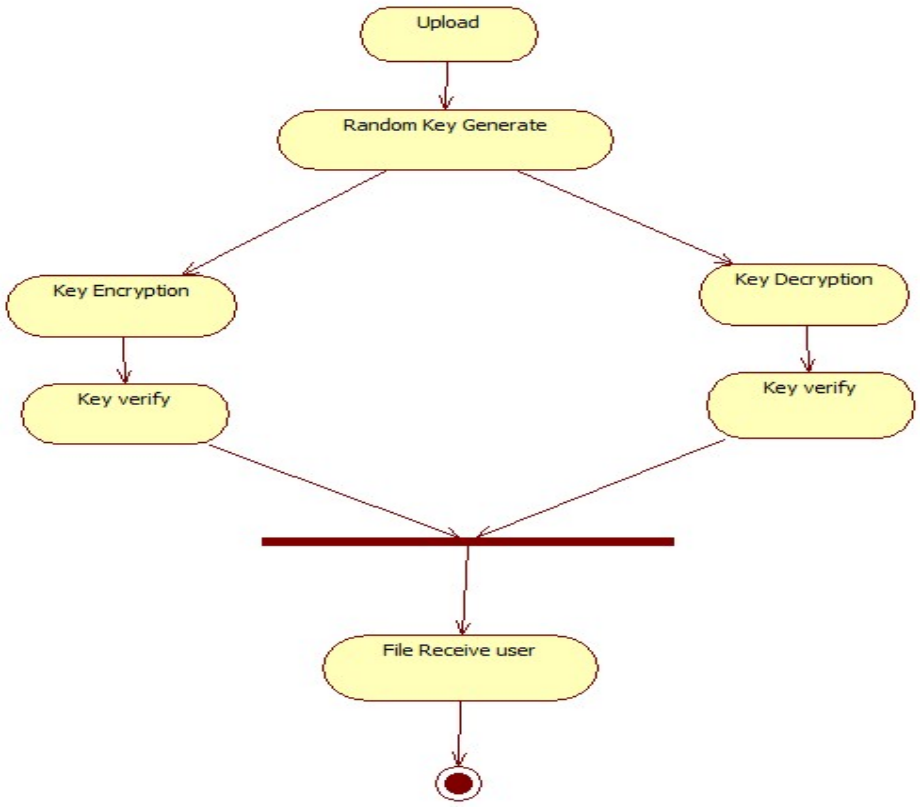
A use case illustrates a unit of functionality provided by the system. The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases. The use case has two actors: user and server. User gives the image as input and server performs the operation



ACTIVITY DIAGRAM

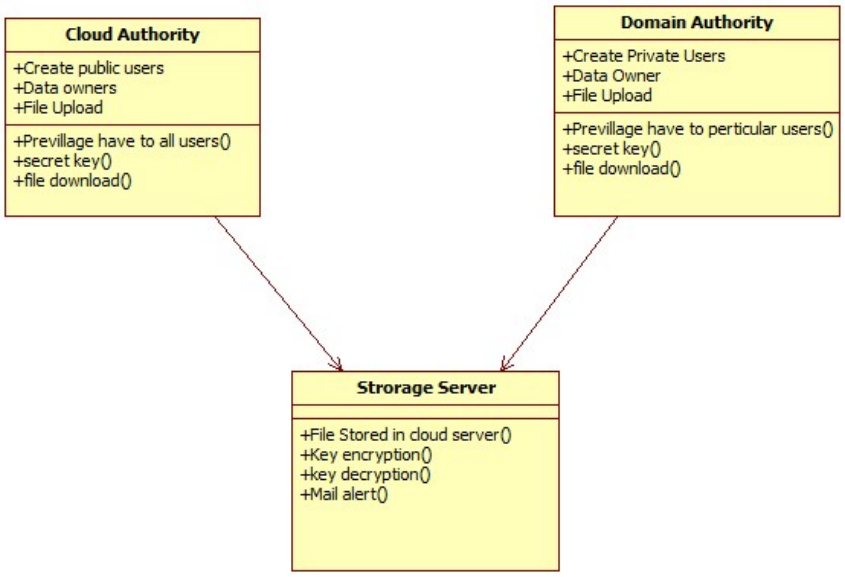
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial state) of the workflow;
  An encircled black circle represents the end (final state)



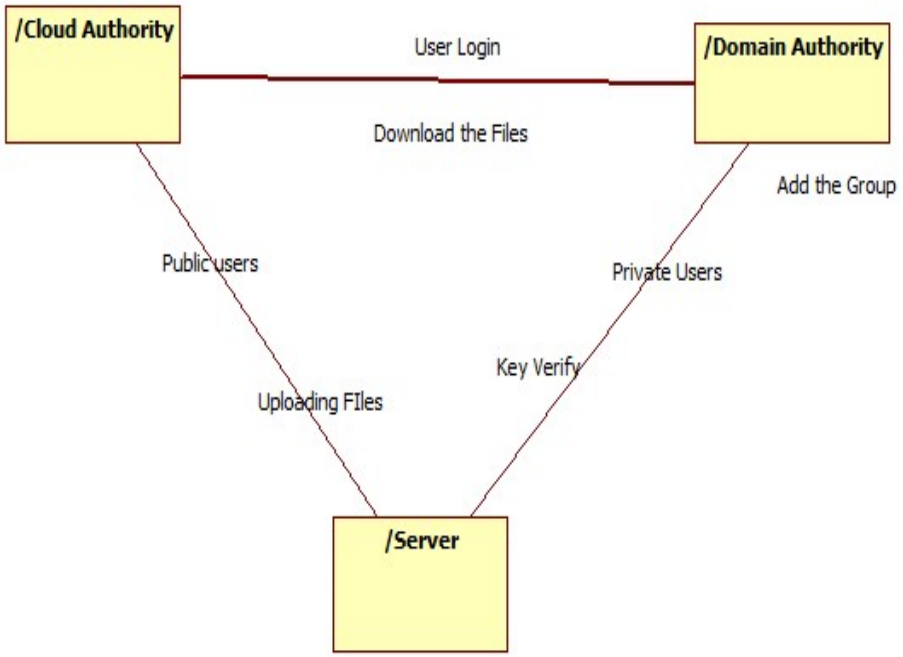
CLASS DIAGRAM

The class diagram shows how the different entities (people, things, and data) relate to each other; in other words, it shows the static structures of the system. A class diagram can be used to display logical classes. Class diagrams can also be used to show implementation classes, which are the things that programmers typically deal with. A class is depicted on the class diagram as a rectangle with three horizontal sections, as shown in above figure. The upper section shows the class's name; the middle section contains the class's attributes; and the lower section contains the class's operations (or "methods"). The diagram has five main classes which give the attributes and operations used in each class.



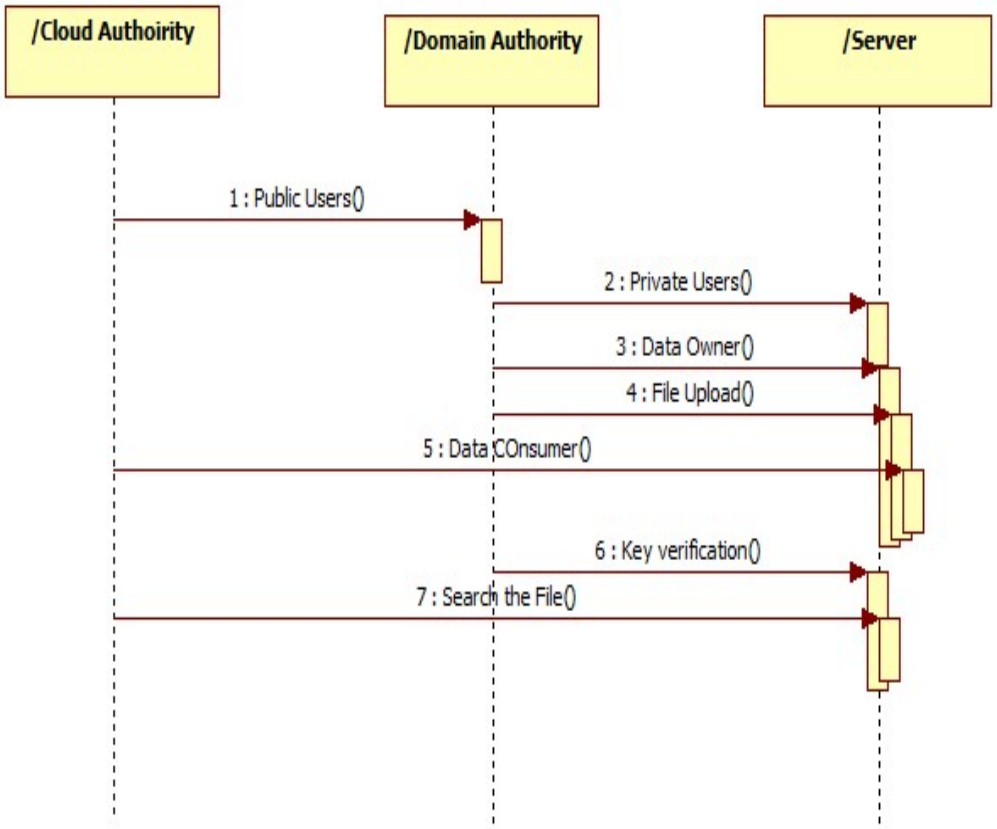
COLLABORATION DIAGRAM

Collaboration diagrams are a technique for defining external object behavior. They include the same information as Sequence Diagrams (or message trace diagrams) but are better able to show asynchronous message passing. Collaboration diagrams show how objects collaborate by representing objects by icons and their message passing as labeled arrows.



SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios**.**



DATA FLOW DIAGRAM

Data Flow Diagram / Use Case Diagram / Flow Diagram:

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system

The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

YSTEM IMPLEMENTATION

MODULES

- Data Owner
- Data Consumer
- Domain level Security
- Attribute based security
- Secret file accessing

MODULES DESCRIPTION

Data Owner

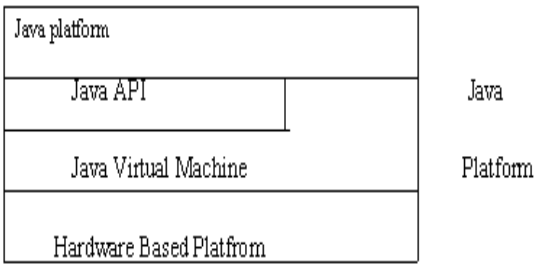
In this module, the data owner uploads their data in the cloud server. For the security purpose the data owner encrypts the data file and then store in the cloud. The data owner can change the policy over data files by updating the expiration time. The Data owner can have capable of manipulating the encrypted data file. The data owner can set the access privilege to the encrypted data file. Data owner to delegate most of the computational overhead to cloud servers. The use of KP-ABE provides fine-grained access control gracefully. Each file is encrypted with a symmetric data encryption key ( ), which is in turn encrypted by a public key corresponding to a set of attributes in KP-ABE, which is generated according to an access structure. The encrypted data file is stored.With the corresponding attributes and the encrypted. If the associated attributes of a file stored in the cloud satisfy the access structure of a user's key, then the user is able to decrypt the encrypted, which is used in turn to decrypt the file. Data owners encrypt their data files and store them in the cloud for sharing with data consumers. To access the shared data files, data consumers download encrypted data Files of their interest from the cloud and then decrypt them. Each data owner consumer is administrated by a domain authority. A domain authority is managed by its parent domain authority or the trusted authority. Data owners, data consumers, domain authorities, and the trusted authority are organized in a hierarchical manner.

Data Consumer:

The user can only access the data file with the encrypted key if the user has the privilege to access the file. For the user level, all the privileges are given by the Domain authority and the Datauser's are controlled by the Domain Authority only. Users may try to access data files either within or outside the scope of their access privileges, so malicious users may collude with each other to get sensitive files beyond their privileges. Data owners encrypt their data files and store them in the cloud for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them. Each data owner/consumer is administrated by a domain authority. A domain authority is managed by its parent domain authority or the trusted authority. Data owners, data consumers, domain authorities, and the trusted authority are organized in a hierarchical manner. data consumers will be always online. They come online only when necessary, while the cloud service provider, the trusted authority, and domain authorities are always online. The cloud is assumed to have abundant storage capacity and computation power. In addition, we assume that data consumers can access data files for reading only.

Data consumer create the account and then login to access the cloud storage information and data consumer entry level based on the hierarchical manner.

API is departmental into libraries **(packages)** of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.



JAVA Platform

As a platform-independent environment, Java can be a bit slower than native code. However, smart compliers, wheel-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening portability.

CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION

A semi-anonymous attribute-based privilege control scheme AnonyControl and a fully-anonymous attribute-based privilege control

scheme Anony Control-F to address the user privacy problem in a cloud storage server. Using multiple authorities in the cloud computing system, our proposed schemes achieve not only fine-grained privilege control but also identity anonymity while conducting privilege control based on users' identity information. More importantly, our system can tolerate up to $N - 2$ authority compromise, which is highly preferable especially in Internet-based cloud computing environment.

FUTURE ENHANCEMENTS

Future enhancement of this project is following schemes. A unified scheme for resource protection in automated trust negotiation. Automated trust negotiation using cryptographic credentials

REFERENCES

REFERENCES

[1]K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, T. V. X. Phuong, and Q. Xie, "A DFA-basedfunctional proxy re-encryptionscheme for secure public cloud data sharing," IEEE Transactions on Information Forensics and Security, vol. 9, no. 10, pp. 1667–1680, October 2014.

[2]T. H. Yuen, J. K. Liu, M. H. Au, X. Huang, W. Susilo, and J. Zhou, "ktimesattribute-basedanonymousaccess control for cloudcomputing," IEEE Transactions on Computers, vol. 64, no. 9, pp. 2595–2608, September 2015.

[3]F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweightdevices," IEEE Transactions on Information Forensics and Security, vol. 9, no 5, pp. 763–771, May 2014.

[4]C. Fan, S. Huang, and H. Rung, "Arbitrary-state attribute-basedencryption with dynamic membership," IEEE Transactions on Computers, vol. 63, no. 8, pp. 1951–1961, August 2014.

[5]T. Jung, X. Mao, X.-Y. Li, S.-J. Tang, W. Gong, and L. Zhang, "Privacypreservingcloud data accesswith multi-authorities," in Proc. IEEE INFOCOM, Apr. 2013, pp. 2634–2642.

[6] J. Hur, and D. K. Noh, "Attribute-basedaccess control with efficient revocation in data outsourcing systems," IEEE Trans. ParallelDistrib. Syst., vol. 22, no. 7, pp. 1214-1221, 2011.

[7] N. Oualha, and K. T. Nguyen, "Lightweightattribute-basedencryption for the Internet of Things," in Proc. ICCCN, 2016, pp. 1-6.

[8] S. Easwarmoorthy, F. Sophia, and A. Karrothu, "An efficient key management infrastructure for personalhealth records in cloud," in Proc. WiSPNET, 2016, pp. 1651-1657.

[9] D. Pletea, S. Sedghi, M. Veeningen, and M. Petkovic, "Secure distributed key generation in attributebasedencryptionsystems," in Proc. ICITST, 2015, pp. 103-107.

[10] G. Zhang, L. Liu, and Y. Liu, "An attribute-basedencryptionschemesecureagainstmalicious KGC," in Proc. TRUSTCOM, 2012, pp. 1376-1380.