IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Time Based Hybrid Secured Access Control System for Cloud Data

[1]Nagur Sagar Siddaram,[2]Dr C.S Jayasheela

[1]M. tech student, [2]Assistant Professor

[1]Depratment of ISE,

[1] Bangalore Institute of Technology, Bangalore, India

Abstract: The cloud-based data outsourcing paradigm is a two-edged sword. On the one hand, it relieves data owners of technical administration responsibilities and makes it easier for them to share their data with their intended consumers. On the other side, it raises additional concerns about privacy and security. Several works have been presented to offer fine-grained data access control in order to guarantee data confidentiality against the honest-but-curious cloud service provider. To support fine-grained data access control, a number of projects have been proposed. To date, no solution has been able to handle both fine-grained access control and sensitive data publication related to time factor. To do so the project titled Time-based hybrid secured access control system for cloud data has been proposed which uses Ciphertext-policy attribute-based encryption (CP-ABE) along with Residue Number System (RNS) encryption technique for data confidentiality, security and integrity of data owner data. Also, considering the factor of time every time when the user requests for the file a new secret code is generated and give to user to decrypt the file received from data owner. The proposed system works properly with no issues.

Index Terms **– Cloud, Data User, Data Owner, Central Authority (CA), RNS encryption technique.**

I. INTRODUCTION

Cloud storage offers significant benefits in terms of data sharing convenience and cost savings. Since the data stored within cloud is data outside organizational limits, causing consumers to lose control of their data and creates legitimate safety issues that hold down cloud computing adoption. Meanwhile posing additional issues in terms of data privacy. Data is no again in data owner's trusted domain. Hence, the data owner cannot rely on cloud server for undertaking safely access and control of data. As a result, secure access control is becoming difficult cloud storage issue.

There have been multiple efforts on privacy-preserving data sharing in the cloud-based upon several cryptographic techniques, where systems based upon Cipher text-policy attribute-based encryption (CP-ABE) has much interest as it might give data owners fine-grained as well as easy access control over their data. Nonetheless, such schemes assess a user's access permission solely depending upon his or her intrinsic characteristics, with no consideration for other essential factors such as time. In actuality, when dealing with sensitive data, the time element frequently plays a significant role. When storing sensitive data in cloud, data owner of the data might wish for various consumers for accessing material at assorted times.

i. The Central Authority (CA) is in charge of complete system's security protection. The Data Owners (Owners), The Data Users (Users) are registered and maintained by the Central Authority (CA). The Central Authority (CA) having authority to Add, Edit or Delete registered Owners after Owner's work is completed. It distributes security keys to each User and publishes system parameters. It also serves as a time agent, ensuring that the timed-release function is maintained.

ii. The Data Owner (Owner) is responsible for storing files in the cloud, which approved Data Users then access. The Data Owners are in charge of uploading all of the files in the system. The system will encrypt the file once uploaded using the Data Owners Encryption. The Data Owner must specify each file's Access Policy. Domain Attribute and Sub-Domain Attribute are used to create access policies.

iii. The Data Consumer (Users), when they register themselves, receives the Identity Key by email. The Data Consumer (User) receives their access key via email from the relevant data owner. You may download the files that you access through the access key, remember that the access control is established by the data owner. Suppose the users wish for downloading any file. First, the file must be selected from list, and system requires the access key. After the system receives an access key, the attribute set is separated from the key and access rights and time is checked. If user is accessible, encrypted file, that is decoded using the decryption key, may be downloaded and downloaded to the local consumer data system.

iv. Cloud Service Provider (Cloud) is responsible for storing the file uploaded by the Data Owner (Owner). If the Cloud Service Provider tries to look into the file, he will see the encrypted text.

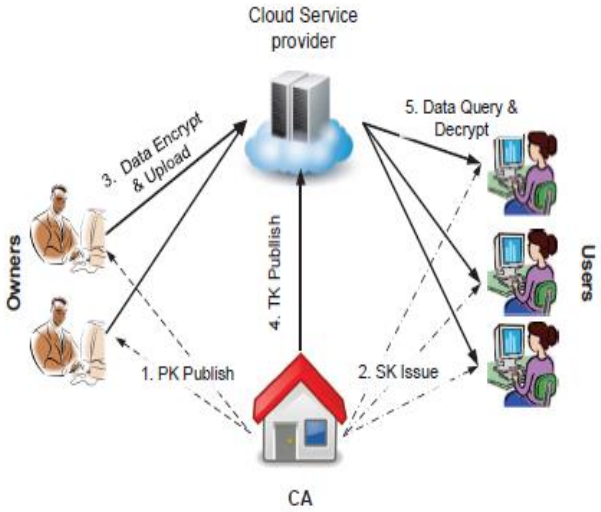

Figure1: System Model

II. LITERATURE SURVEY

AuthorsQin Liuyz,ChiuC.Tanz,JieWuz, withGuojunWangy [1] according to their research, the potential cost benefits by outsourcing to supplier of cloud services, cloud computing is becoming more popular (CSP). Data owner could encrypt outsourced data for safeguarding data from a potentially untrustworthy CSP. To offer fine-grained access control, flexible encryption techniques attribute-based encryption (ABE) is used.

ABE enables data encryption through the use of a multi-attribute access structure. Users are given attribute keys instead than unique decryption keys for specific files. To decrypt a file, Users should have necessary access structure characteristics. For example, an encrypted file that has access structure {(a1 ∧ a2) ∨ a3} implies which either user with a1 and a2 characteristics or a3 user may decrypt file. From this paper we gain the knowledge of how the attribute-based encryption (ABE) works. Users are issued attribute keys. Attribute keys are given to users. To decrypt a file, consumers should have appropriate attributes which comply with access structure.

AuthorsKanYang,XiaohuaJia,KuiRen, Bo Zhang,RuitaoXie [2] according to their research in the system they have proposed. This enables data owner to manage access rules more directly and does not need the data owner to provide keys, Cyphertext-Policy Attribute-based Encryption (CP-ABE) is considered one of most advanced methods for cloud storage data access control. There is an authority in the CP-ABE scheme that is in charge of attribute management and key distribution. Authority could be a university's registration office, company's human resource department, or something else entirely. Data owner sets access restrictions and encrypts data accordingly. Each user will be assigned a secret key based on their characteristics.Ciphertexts can only be decrypted if characteristics of user satisfy access rules. User might have characteristics granted by several authorities in cloud storage systems,owner might provide data with different agencies administered consumers.

John Bethencourt, AmitSahai, Brent Waters [3] according to their research they have developed a Ciphertext-Policy Attribute Based Encryption technology. Their approach enabling novel sort of encrypted access control in which a collection of attributes defines a user's private keys, and party data encryption may set a policy for these characteristics that specifies which users can decode. Their approach is robust to Collusion attack when an attacker gets many private keys and enable policies as to any monotonous tree access structure to be defined. Finally, they offeredsystem implementation, which incorporated many optimization approaches. Designed a new two-level random masking methodology to provide a revolutionary private key randomization strategy. The key to our safety tests in generic bilinear group model is the use of groups with efficient bilinear mappings. From this paper we learn how the use of multiple combination algorithm can protect the data in more secure manner.

Elisa Bertino, Gabriel Ghinita and Ashish Kamra [4] according to their research, Organizations are subject to security breaches resulting in data theft and unauthorized disclosures as they rely on, sometimes distributed, Operational, decisional and strategic information systems. Though several techniques exist to protect data when it is transmitted across sites, such as encryption and digital signatures, A fully complete data security strategy must also incorporate restricted access control systems relying on data contents, subject qualifications and characteristics or other relevant information, like time. Data semanticist should be considered in order to develop adequate access control rules, it is now widely accepted. To meet these demands, the database security research community has created a number of database-specific access control strategies and procedures throughout the years.

Zhiguo Wan, Jun'e Liu, and Robert H. Deng [5] according to their research and study, in recent years, Cloud computing is considered one of IT industry's most important concepts. Because this new computer technology requires customers to commit highly sensitive data to cloud providers, concern over outsourced data security and privacy is growing. Various methods for the management of access to outsource data in cloud computing were developed utilizing attribute-based encryption (ABE), but most are inflexible when it comes to creating sophisticated access control rules. In this study we present hierarchical HASBE, a hierarchical attribute-set-base encryption that extends the scalable, flexible and thorough access control of outsourced data into cloud computing to include ciphertext policy attribute/set-based encryption (ASBE) by Hierarchical user structure.

III. METHODOLOGY

System design is referred to as the process of defining system features like module, architectures, parts and interfaces and data for system based upon given requirements. It identifies, creates and designs systems in order to fulfill the particular goals and requirements of project. The two parameters time and attributes are combined in a cloud storage access control technique that can achieve fine granularity and timed release simultaneously.

Create a separate entity (central authority or CA) for handling timed-release function. CA needs to broadcast universal time-related token regularly for releasing access privilege, in addition to attribute-associated private keys. Such design requires only a little investment to deliver our desired access control strategy, which is both fair and valuable.

USE CASE DIAGRAMS

Use cases and actors are two fundamental components of a use case diagram. Use case is list of circumstances which describe how source as well as destination interacts. Connection among actors and use cases is shown in a case diagram. Use case diagram's notation is relatively simple, further it does not have as many symbols as other UML diagrams.

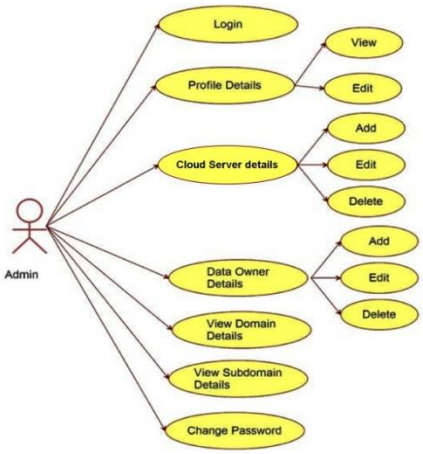

Figure 2: Use Case Diagram for the Central Authority (CA)

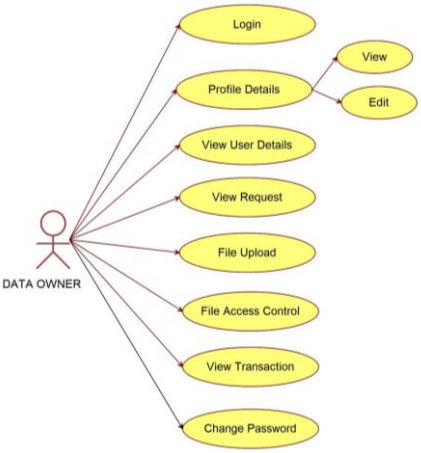

Figure 3: Use case Diagram for the data owner

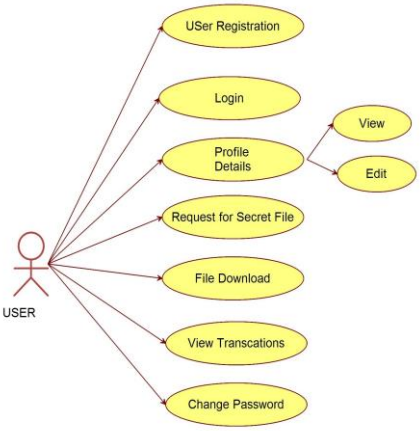

Figure 4: Use case Diagram for the data user

Figure 2 represents the use case diagram for the central authority. The admin (central authority) has these use cases that are login, profile details, cloud service details, data owner details, domain details, sub domain details, change password.

Login is through which the admin logs into the application. Profile details are where the admin looks after the data owner and users who have logged into the application and can edit their profiles. Cloud server details are the links to the cloud where the file is being stored, admin has the authority to add, edit, and delete the server details. Domain details and sub domain details are also maintained by admin.

Figure 3 represents the use case diagram for data owner. The data owner has these use cases that are login, profile details, user details, view request, file upload, file access control, view transaction, change password.

Login is through which the data owner logs into the application. Profile details is where the data owner can view and edit profile. View request is where requests from user are seen. File upload is where the data owner uploads the file. Access controls are given to the file in the file access control. Data owner can see how many transactions have taken place in view transaction. Data owner can change his password in change password use case.

Figure 4 represents the use case diagram for the data user. The data user has these use cases that are user registration, login, profile details, request for secret file, file download, view transaction, change password.

User registration is where the new users register into the application. login is through which the data owner logs into the application. Profile details is where the data user can view and edit profile. User requests for the file the request is stored in request for secret file use case. User can download the file from the file download use case. User can see transactions in view transaction. Data user can change his password in change password use case.

IV. IMPLEMENTATION

Residue numeral system (RNS) is number system expressing the integer module by its numeral values. This depiction is made possible by the Chinese theorem, which says that if N is product of modules, there is precisely one number with any collection of modular values in the interval of length N. Arithmetic of numerical residue system is also known as multimodular arithmetic.

Multi-modular arithmetic, particularly in linear algebra, is often used for computation with big numbers since it is quicker than conventional numeral systems even if conversion time is determined.

A collection of k integrators {m1, m2, m3,..., mk} termed modules is specified for a residual numerical system which are usually intended to be coprime pairs (which is, any 2 of them has greatest common divisor equal to 1). Residue number schemes for non-coprime module have been developed but are not widely utilized because of worse characteristics.

In the residual numeral system, integer x is depicted in the module set of its residues {x1, x2, x3,..., xk} under Euclidean division. xi= x mod mi, 0<= xi< mi for each i.

Let M be all mi's product. In the residual numeral system described by the mi's two entiers, whose difference is multiple M, are the same. More specifically, the remaining Chinese theorem states that each of M various residue sets constitutes precisely one residual modulo M class. Which is, every set of residue in the range 0,...., M-1 represents precisely one integer X. For number signed, dynamic range is -[M/2] <= X <= [M-1/2].

Pseudo code for RNS

Key Generation
Step 1: Start
Step 2: Generate two distinct prime number P1 & P2
Step 3: Let M=P1×P2
Step 4: Let A1=M/P1
Step 5: Let A2=M/P2
Step 6: Solve eqn ((A1×T) mod P1) == 1 and get value of T
Step 7: T1 = T
Step 8: Solve equation ((A2×T) mod P2) == 1 and get the value of T
Step 9: T2 = T
Step 10: Form Encryption Key (P1,P2)
Step 11: Form Decryption Key (A1,A2,T1,T2,M)
Step 12: Stop

Encryption Process

Step 1: Start
Step 2: Let F be File Input and (P1,P2) is encryption key
Step 3: Let N=Number of bytes in F
Step 4: Create file EF for storing encrypted data
Step 5: Let I=1
Step 6: Read Ith byte (B) of F
Step 7: Convert the B to ascii value V
Step 8: Let R1 = V % P1
Step 9: Let R2 = V % P2
Step 10: Append (R1, R2) in EF
Step 11: I=I+1
Step 12: if I <= N goes to Step 6
Step 13: Close the File EF
Step 14: Stop

Decryption Process

Step 1: Start
Step 2: Let EF is Encrypted File Input and (A1, A2, T1, T2, M) is decryption key
Step 3: Let N=Number of Pair values in EF
Step 4: Create file F1 to store decrypted data
Step 5: Let I=1
Step 6: Read Ith Pair value (R1, R2) from EF
Step 7: Let E=[(A1×T1×R1)+(A2×T2×R2)] mod M
Step 8: Convert E to Byte Value E1
Step 10: Append E1 in F1
Step 11: I=I+1
Step 12: if I <= N goes to Step 6
Step 13: Close the File F1
Step 14: Stop

V. RESULT

The result chapter, presents in detail the contribution of the project. Result chapter provides an extensive discussion about the results and evaluate our contribution to the project along with a detailed description, resolve the overall research goal.
Steps to be followed by data user to assess the file from data owner through public cloud (Drive HQ).
Step 1: Data Owner and Data User must register themselves within the Central Authority (CA).
Step 2: Data User must provide user id, password, e-mail also domain and sub-domain through which they need to access the file when registering into the application. Data Owner must provide id and password to register into application i.e., is Central Authority (CA). Figure 5 shows the Data User registration process.
Step 3: Data User when registered to the application receives identification key (id key) to the registered e-mail.
Step 4: To access the application the user must provide the identification key. Figure 6 shows the identification key received at Data User e-mail.
Step 5: If the identification key provided is correct the Data User can access the application and request the file from Data Owner. The Data User must properly mention the domain and sub domain of which Data Owner he needs the file from.
Step 6: The Data owner see the request send the secret key to the Data User email through which they can decrypt the file.
Step 7: The Data Owner now selects the file mentions the proper date and time at which the file must be released selects the domain and sub domain. Encrypts the file using RNS encryption technique and uploads the file to the public cloud. Figure 7 shows the file uploadation process by proper mentioniong of data and time as well as proper selection of domain and sub domain.
Figure 8 shows the file uploades successfully to the public cloud. Figure 9 shows encrypted format of file uploaded to the cloud.
Step 8: The Data User encrypts the file using the secret key sent by the Data Owner. Figure 10 shows the secret key received from the Data Owner.

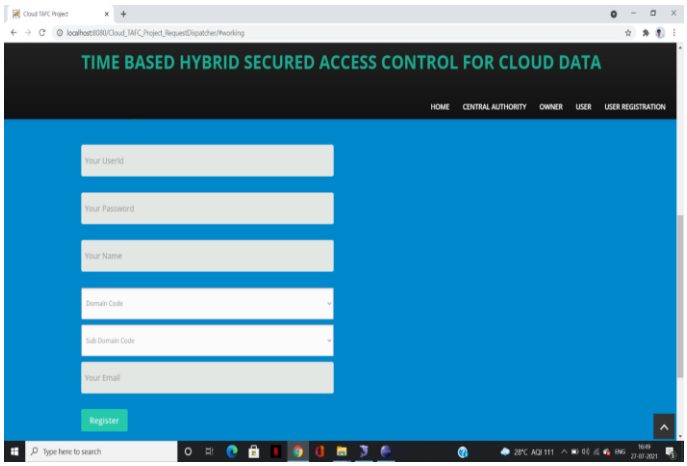

Figure 5: User registration

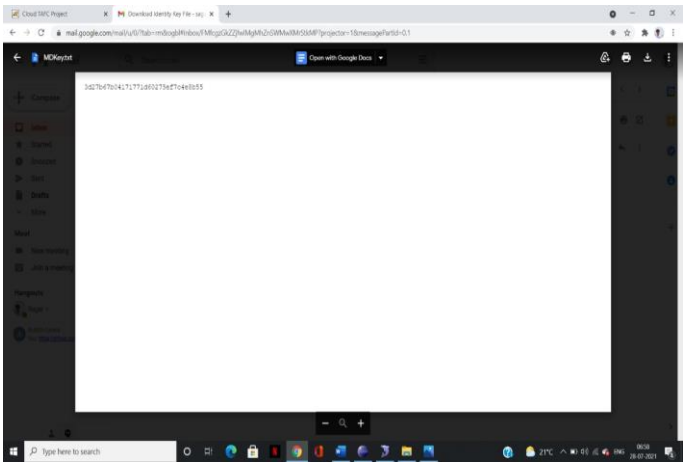

Figure 6: Identification key

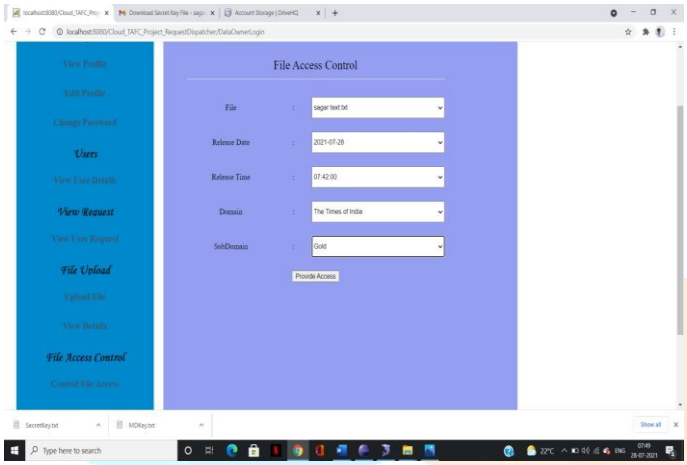

Figure 7: Owner giving access control to file

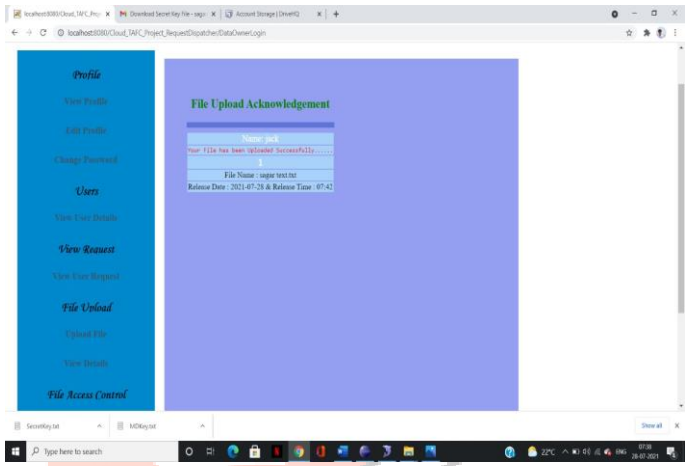

Figure 8: File uploaded to the cloud

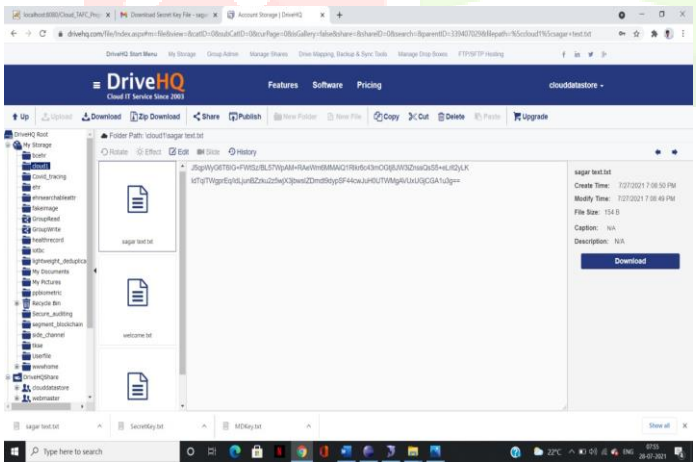

Figure 9: Encrypted file in public cloud

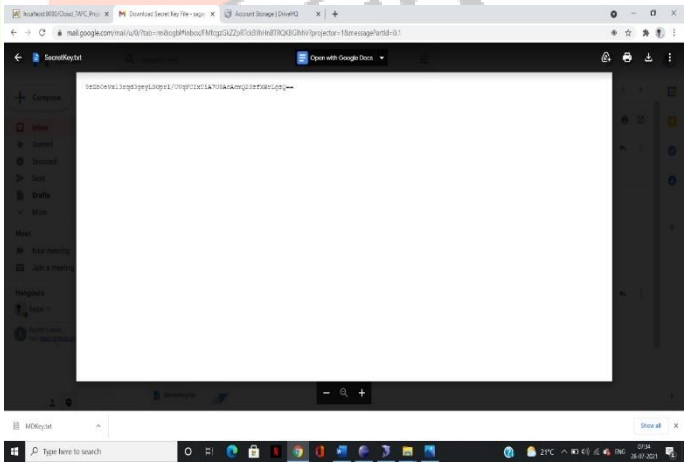

Figure 10: Secret Key used by user to decrypt file

VI. CONCLUSION AND FUTURE WORK

The application is developed on web technology using Model View Controller (MVC) architecture. The application uses a hybrid cloud concept that uses public storage servers and private encryption servers, giving more security to the application. The application was tested with sample files for uploading and downloading, and all experimental results show this application meets functional requirements.

For protecting data from cloud service provider, Owner encrypts file with two-layer encryption. The two algorithms used are RNS and DES. The encrypted file now stored is in the public cloud. If the cloud services provider tries to access the file, he may only see the encrypted text. If the user needs to access the file for information, he must first register within the application. To check if the user is genuine, he will receive an identification key to his E-mail, which is given during registration. If the user provides the proper identification key, then access to the application file is possible. To access the proper file, user must provide the proper domain and sub domain of the file. Now, the Owner provides feasible time and date only after the mentioned time and date are passed, the file is

accessible to the user. The Owner sees the request for the file, and he sends the secret key to the user's E-mail, which user has to provide so as to download the file. If the user appropriately provides the secret key, the file will be downloaded to the user's system.

Currently the application can encrypt the files in text and photo format. In future will try to use audio and video files because data can be in any format. Advancement would be to auto-delete the file from the cloud when the use of the files is completed. If there is an attack on the cloud, the files uploaded to the cloud would be deleted, and is not possible to access the file and also provide a storage backup of deleted file, in case need of deleted file may arise in some point in time.

REFERENCES

[1] Liu, Qin, et al. "Reliable re-encryption in unreliable clouds." *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*. IEEE, 2011.

[2] Yang, Kan, et al. "DAC-MACS: Effective data access control for multiauthority cloud storage systems." *IEEE Transactions on Information Forensics and Security* 8.11 (2013): 1790-1801.

[3] Bethencourt, John, Amit Sahai, and Brent Waters. "Ciphertext-policy attribute-based encryption." *2007 IEEE symposium on security and privacy (SP'07)*. IEEE, 2007.

[4] Bertino, Elisa, Gabriel Ghinita, and Ashish Kamra. *Access control for databases: Concepts and systems*. Now Publishers Inc, 2011.

[5] Wan, Zhiguo, and Robert H. Deng. "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing." *IEEE transactions on information forensics and security* 7.2 (2011): 743-754.