

SECURE AND RELIABLE DATA ACCESS CONTROL MECHANISM IN MULTI-CLOUD ENVIRONMENT WITH INTER-SERVER COMMUNICATION SECURITY

Nagesh Shenoy Havanje¹, Kodihalli Ramanna Anil Kumar², Suchitra Nagesh Shenoy³, Abhishek Suresh Rao^{4,*} and Rajgopal Kondalakadu Thimmappayya⁵

Received: February 02, 2021; Revised: June 21, 2021; Accepted: July 30, 2021

Abstract

Cloud system are dynamic and offer lucrative services to its users but often these systems are doubted for its honest but curious property. Data confidentiality, user privacy and integrity maintenance are the major tasks performed by the cloud servers to secure the infrastructure. CP-ABE guarantees fine-grained access control of data by legitimate users in cloud storage system but doesn't guarantee message interception by spying intruders who fake the legitimate access IDs. Multi cloud systems are highly vulnerable to these attacks as flow of sensitive information is much due to inter-cloud communication. To solve the above problems, we propose a practical CP-ABE based framework for multi-cloud system consisting mechanisms to secure inter-server communication, maintain data and user privacy, data integrity verification and rekeying to handle attribute revocation. We give performance analysis of our mechanisms under IEP and CDHP assumptions and prove it to be secure against eavesdropping and attacks. Results shows performance of proposed scheme in terms of efficiency and security.

Keywords: Cloud computing, ECDH, AES, CP-ABE, RSA, SHA3, Attribute revocation

Introduction

Background

Cloud computing is seen as a promising computing paradigm which provides on-demand online services such as software, platform and infrastructure. Enterprises and individuals prefer it due to its features of being economical, scalable, and remotely accessible, though the data security, user privacy and access control are urgent problems to

tackle in cloud environment. Once the data owner outsources the data, they do not have the control on their data. Servers, in general, are provisioned with Access Control Lists (ACL) to grant access to users based on its IDs list. Server may erroneously or deliberately provide access of data to unauthorized users. To address this problem, fine grained access

¹ Department of Computer Science and Engineering, Canara Engineering College, Mangalore. E-mail: h.nagesh.shenoy@gmail.com

² Department of MCA, DSCM & IT, Bangalore. E-mail: anilkumar@dayanandasagar.edu.

³ Department of Electronics and Communication Engineering, Canara Engineering College, Mangalore. E-mail: suchiprab_16@yahoo.co.in

⁴ Department of Information Science and Engineering, NMAM Institute of Technology, Nitte. E-mail: abhishekr@nitte.edu.in

⁵ Department of Computer Science and Engineering, Canara Engineering College, Mangalore. E-mail: rajgopalkt@gmail.com

* Corresponding author

control mechanisms (Goyal *et al.*, 2006; Bethencourt *et al.*, 2007) were introduced in the past. Attribute Based Encryption, commonly referred as ABE mechanism was initially proposed by Sahai and Waters in 2005 (Sahai and Waters, 2005). ABE is currently a universal mechanism to solve the access control problem. It basically associates attributes set with the private keys of the user. ABE is an extended version of primitive IBE scheme (Boneh and Franklin, 2003). ABE can be broadly classified as KP-ABE (Key Policy Attribute Based Encryption) and CP-ABE (Ciphertext Policy Attribute Based Encryption) (Sahai and Waters, 2005; Goyal *et al.*, 2006).

Conventional encryption techniques such as ID based encryption and public key encryption (Mishra *et al.*, 2020; Shankar *et al.*, 2020) are for direct ciphering of data belonging to single user. But these techniques alone are insufficient for one to many encryption scenarios. CP-ABE mechanism, presented by (Goyal *et al.*, 2006) for an adaptable one-to-many encryption that has now become a standard for fine-grained access control over encrypted data. In this technique, a user can define access policy over their data that gets encrypted and keys are issued by authority for privileged users to access data based on their attributes. Therefore, if receiver's attribute set satisfies the access policy defined by data owner then only they would be able to fetch the decrypted data. But still there exists a serious issue i.e., attribute revocation, that needs to be looked after. As user's ID attributes are not permanent and may constantly change so the new attribute may be added, replaced or old attributes may get revoked. Thus, in real time CP-ABE implementation there is always a need of flexible attribute management scheme (Premkamal *et al.*, 2020).

CP-ABE schemes are a standard these days for cloud systems but multi-clouds need elevated level of security due to their infrastructure characteristics. Multi-cloud storage systems reduce many risks involved with single cloud storage systems due to its characteristic of storing data in numerous servers and providing inter cloud data transmission (Wang *et al.*, 2020). A multi-cloud is sometimes called as inter-cloud or cloud-of-clouds as a phrase coined by (Cachin *et al.*, 2010; Vukolic *et al.*, 2010). stated that fundamental level in cloud is intra-cloud and advanced level is inter-cloud where multiple clouds interact with each other. Major reason of migration from single to multi-cloud was explained by (Bowers *et al.*, 2009) that more than 80% analysts and admins in organizations are worried of system failure, data misplacement and other risks while using cloud-based systems. Also, a noted researcher (Vukolic *et al.*, 2010) stated that users are migrating

to multi-cloud systems because of insufficient security and trust by many CSPs (Cloud Service Providers). But again, there exist a need to incorporate a mechanism through which it can be ascertained that message communication between servers on clouds are free from any hack or eavesdropper.

Our Contribution

In this paper, we address various issues involved in multi-cloud scenario and propose schemes such as ECDH (ECC+Diffie-Hellman) algorithm to shield message interception while servers make internal communication of sensitive message, CP-ABE for fine grained access control, rekeying mechanism at the time of user revocation, AES for message encryption, RSA for key encryption, and SHA-3 digital fingerprinting for integrity verification. Our architecture proves to be fault tolerant, provides better efficiency to the system and strongest shield against various attacks.

Paper Organization

The rest of this paper is organized as follows. Section 2 covers the related works, in section 3, we review some of the fundamentals such as grouping, difficulty assumptions and access structure. Section 4 describes the background about CP-ABE algorithm, RSA, ECC and ECDH. Section 5 demonstrates the proposed system model. The performance results are provided in Section 6. Section 7 concludes the paper.

Related Works

Access control mechanisms are fast and reliable in private cloud storage servers (patil *et al.*, 2014; Riad *et al.*, 2019) but it is still a hot topic of research in public cloud storage system (Li *et al.*, 2019; Riad *et al.*, 2019; Priyanka and Ramakrishna, 2020; Xiong *et al.*, 2020). ABE mechanism, being very popular in fine grained access control has some of the noted extension works. Yu *et al.* (Yu *et al.*, 2010) introduced Key Policy ABE (KP-ABE) (Goyal *et al.*, 2006) to perform fine grained access control in cloud systems. Subsequently, a range of diverse single-authority based ABE mechanisms were introduced by various researchers (Zarandioon *et al.*, 2012; Li *et al.*, 2014; Youn *et al.*, 2019). (Chase, 2007) introduced the first and fundamental multi-authority ABE mechanism that involved a universal Certificate Authority (CA). His work got much appreciation but CA was still considered as system efficiency bottleneck due to vulnerabilities present in it.

Lately, many research works have been carried out in the area of ABE supporting attribute revocation issue (Pirretti *et al.*, 2006; Bethencourt *et al.*, 2007; Boldyreva *et al.*, 2008) have introduced time bound rekeying technique in which every attribute is assigned with an expiration time limit. (Bethencourt *et al.*, 2007) tried to improvise the scheme by utilizing user's secret key and assigning it an expiration time. The keys were updating at the user's end to maintain efficiency in computation and limit computational overhead at authority end. (Boldyreva *et al.*, 2008) offered binary tree scheme to build data structure as a fast revocation mechanism for IBE.

Though, the above schemes were performing well but still were lacking support for immediate rekeying in the case of user move out or in and that is the reason that these schemes were termed as coarse-grained evocation mechanisms. Also, the two major performance issues in these schemes with respect to security are backward and forward security (Chase, 2007; Patil *et al.*, 2014; Youn *et al.*, 2019; Priyanka and Ramakrishna, 2020).

Security in cloud storage servers is of utmost importance and single cloud always remain vulnerable to security and other issues. Being easy for service providers to implement least APIs for single cloud interaction and easy operations, they generally opt for it. But the SLAs offered by these providers easily get wrecked due to dependency on single cloud and server crashes. It is an understood fact that single storage servers fail to prevent data loss at the time of crisis and thus lead to permanent damage and service availability.

Cachin *et al.* (2010) have researched on intercloud models with respect to security and integrity of data. They utilized a fault resilient technique with a secure admission control to make sure utmost data privacy and security in multi-cloud systems (Papaioannou *et al.*, 2012; Bessani *et al.*, 2013; Chen *et al.*, 2013). (Liu *et al.*, 2009) splits data into fixed sized chunks and use ECC algorithm to encrypt those chunks before storing those in various storage servers in a redundancy group. Also, it distributes them among the storage nodes in a redundancy group, which can be dynamically restored at the time of data recovery. It significantly decreases the storage space consumption than the mechanisms which stores replica into servers yet ensuring full data recovery in crisis. (Bermbach *et al.*, 2011) introduced Metastorage, that targets on reducing communication delay and maintain data consistency in multi-cloud systems. They put a priority queue on data access to minimize the communication latency among clouds and concluded that smaller sized files have better data consistency.

In this field security and privacy are the challenging task. The cloud computing has several advantages such as low cost infrastructure, scalability, flexibility, high computational capacity, scalability, global access and many more. However, the authors has pointed out that the data integrity, security and privacy are the major challenges in this field. Thus authors (Nagesh *et al.*, 2017) presented a survey on security and privacy management in cloud computing field. This survey shows that still there is much scope present in this field of security provisioning in cloud computing.

Similarly, another comparative analysis for data security while focusing on the cryptographic approaches to secure the cloud computing system is presented (Shenoy *et al.*, 2019). authors reported the significance of cryptography based schemes and presented a review study about symmetric, asymmetric algorithms and hashing mechanisms

Similar to this work, another review study is presented (Rajgopal *et al.*, 2018) and reported that load balancing plays important role in cloud computing systems for various real-time and offline applications. According to this, load balancing is a technique which helps to distribute the loads to other virtual machines in the considered environment which helps to improve the resource utilization and improves the task completion rate. This paper presents the comparative analysis between static and dynamic load balancing techniques.

While going through the various state-of-the-art works, we found that still there is a dire need of research in the area of multi cloud and data security. Servers internally need to have a strong shield against eavesdroppers to restrict unauthorized access to data while servers make internal communication. Our research reveals that none of the research has paid special attention towards this problem. Instead, majority of the research focuses on security for unauthorized user access that happens in single cloud but multi cloud needs special mechanism to restrict spying while inter cloud data exchange happens.

Preliminaries

Bilinear Map

Let G and G_T be multiplicative cyclic groups of prime order p . Then, a bilinear mapping $e : G * G \rightarrow G_T$ has the below mentioned properties.

Bilinearity: For all $a, b \in \mathbb{Z}_p$, and $g \in G$, their exists $e(g^a, g^b) = e(g, g)^{ab}$.

Non-Degeneracy: $e(g, g) \neq I$, where I is an identity element of G_T .

Computable: e can be efficiently computed.

Computationally Hard Problems

In this section, we consider the following two computational hard problems.

Integer factorization problem (IFP):

Let p and q be ρ -bit primes and $N = pq$ and Gen_F be a polynomial-time algorithm which inputs 1^ρ and outputs (N, p, q) . The factoring assumption relative to Gen_F states that given N , it is computationally infeasible (hard) problem to derive p and q , except with negligible probability in ρ . The formal definition of this problem follows that of (D. Hofheinz and E. Kiltz, 2009): for a probabilistic polynomial-time (PPT) algorithm A , the factoring advantage is defined by

$$Adv_{Gen_F, A}^{IFP}(\rho) = \Pr[(N, p, q) \leftarrow Gen_F(1^\rho) : A(N) = \{p, q\}].$$

The factoring assumption (with respect to Gen_F) states that $Adv_{Gen_F, A}^{IFP}(\rho)$ is negligible in ρ for every PPT A . We say that $(t_{IFP}, \epsilon_{IFP}) - IFP$ assumption holds if $Adv_{Gen_F, A}^{IFP}(\rho) \leq \epsilon_{IFP}(\rho)$, for any sufficiently small $\epsilon_{IFP}(\rho) > 0$, and its running time is at most t_{IFP} .

Computational Diffie-Hellman problem (CDHP):

The problem of breaking the Diffie-Hellman scheme with a RSA modulus $N = pq$ and base g is equivalent to the problem of computing a value of the following function [29]:

$$CDHP(N, g, X, Y) : (g)N \times (g)N \rightarrow (g)N,$$

which is defined by

$$CDHP(N, g, g^a, g^b) = g^{ab} \pmod{N}.$$

Here, $(g)N$ represents a cyclic subgroup of Z_N^* generated by g . The adversary A advantage in solving the CDHP is:

$$Adv_{Z_N, A}^{CDHP}(\rho) = \Pr[A(N, g, g^a, g^b) = g^{ab}].$$

We say that $(t_{CDHP}, \epsilon_{CDHP}) - CDH$ assumption holds if

$Adv_{Z_N, A}^{CDHP}(\rho) \leq \epsilon_{CDHP}$, for any sufficiently $\epsilon_{CDHP} > 0$, with its running time at most t_{CDHP} . As stated in (McCurley, 1988), any algorithm that will break the CDHP for a non-negligible proportion of the possible inputs can be used to factor N . This implies any algorithm that will break the CDHP for a given modulus N can also be used to break the original Diffie-Hellman scheme for the prime moduli that are factors of N .

Access Structure

We let $A_1, \dots, A_n$ represent a set of parties. A collection $B \subseteq 2\{A_1, \dots, A_n\}$ is monotone if $\forall C, D$: if $C \in B$ and $C \subseteq D$, then $D \in B$. A monotone access structure will be a collection of non-empty subsets of $A_1, \dots, A_n$. The sets in B are called authorized sets.

Background

In this section, we will provide overview of our work which involves CP-ABE, ECC, ECDH key sharing scheme and RSA.

CP-ABE

A Ciphertext-Policy Attribute-Based Encryption (CP-ABE) system consists of four principle algorithms:

Setup $(\lambda) \rightarrow (PP, MSK)$: The setup algorithm initiated by TA, takes security parameter (λ) as input, that runs in polynomial time and outputs the public parameter (PP) and a master secret key (MSK) . A user belonging to a set of users is identified by $u \in \{1, \dots, U\}$ and attribute universe as AU .

KeyGen $(PP, MSK, AS) \rightarrow PRK_{K, S}$: KeyGen algorithm is run by TA. This algorithm takes public parameter (PP) , the master secret key (MSK) , and an attribute set (AS) as an input, and returns a private key $PRK_{u, AS}$, which is allocated to a user u .

Encrypt $(Msg, PP, AP) \rightarrow CT$: This algorithm takes a message (Msg) , public parameter (PP) , and an access policy AP over attribute universe (AU) as an input, and returns a ciphertext (CT) , where only legitimate users whose attributes set (AS) satisfies AP can recover message (Msg) . AP is implicitly included in CT .

Decrypt $(CT, PP, PRK_{K, S}) \rightarrow Msg$ or $\perp$: The algorithm takes ciphertext (CT) , public parameter (PP) , and a private key $(PRK_{K, S})$ as an input. If attribute set (AS) is legitimate for file access, the algorithm returns original message (Msg) , otherwise it returns $\perp$ indicating the failure of decryption.

RSA Process

1. Choose $p = 5, q = 17$
2. $n = p * q = 5 * 17 = 85$
3. $\phi(n) = (p - 1) * (q - 1) = 4 * 16 = 64$
4. $\exists e : (1 < e < \phi(n)) \Rightarrow e \leftarrow 11, (e, n) \leftarrow \text{coprime}$
5. $\exists d : (d * e \pmod{\phi(n)} = 1) \Rightarrow d = 35, \because [(35 * 11) \pmod{64} = 1]$
6. $PublicKey = (e, n) = (11, 85)$
7. $PrivateKey = (d, n) = (35, 85)$

8. choose $m = 3$; $encrypt(m) = c = 3^{11} \bmod 85$
 $\rightarrow c = 7$
9. $decrypt(c) = m = 7^{35} \bmod 85$
 $\rightarrow 37881869226566478 \bmod 85$
 $\rightarrow m = 3$

Elliptic Curve Cryptography

During 1980s, Neal Koblitz and Victor Miller presented an algorithm called ECC i.e. Elliptic Curve Cryptography (Miller, 1985; Koblitz, 1987). Subsequently, ECC gained immense popularity and has grown into a standard public-key encryption system, which has been utilized in diverse fields and developed as a trendy area of research in the cryptography domain (Hema *et al.*, 2019).

ECC, as depicts, is based on the elliptic curve. It has many combination variants in combination with various encryption algorithms and it has maximum encryption intensity compared to the existing PKI algorithms. Compared to RSA, which is universal standard in public key cryptography, ECC provides the similar level of security with short key length.

A prime field F_p (contains integers modulo p) is a type of finite field or Galois Field $GF(p)$, where p is a prime digit. For the general EC (Elliptic Curve), select p in a way that it is a prime digit, and let F_p signify the prime field. An elliptic curve E is defined over F_p . Every point on this curve belong to the defined prime field. These points satisfy the equation given below. Here, the pair (x, y) denotes the cartesian coordinates of a point on the curve as shown in Figure 1.

$$y^2 = x^3 + Ax + B$$

A, B are also the elements of the prime field F_p . To make the curve non-singular, it must satisfy the condition.

$$4A^3 + 27B^2 \neq 0 \pmod{p}$$

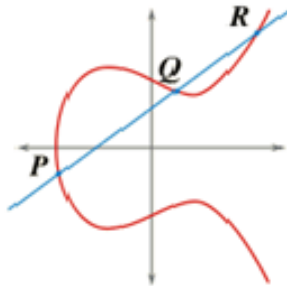


Figure 1. Elliptic Curve.

Key Generation in ECC

Let the defined elliptic curve be denoted by E . It is defined over a prime field F_p . Let T be a point on such an elliptic curve $E(F_p)$, and T must have a prime order n [1].

- 1: $KeyGen(p, E, G, n)$
- 2: $PriKey(d_A) \leftarrow R \mid R = random([1, n-1]) \in F_p$
- 3: Let $Bp(G) = \text{Base Point of Generator } G$
- 4: $PubKey(H_A) \leftarrow PointMult(PriKey(d_A) * Bp(G))$
- 5: return $PubKey, PriKey$
- 6: end

Point Multiplication is achieved by simple elliptic curve operation, point addition and point doubling.

ECDH Key Exchange

ECDH is a combination of the Diffie-Hellman algorithm with elliptic curves. Basically, it is considered as a key-agreement protocol than an encryption algorithm. This means that ECDH defines (to a certain level) the process of key generation and key exchange among parties. It is now up to the system that how it encrypts the data using those keys.

When there is a demand that two parties say, Alice and Bob intend to exchange information securely, and any third party who may listen to the information but should not decode them, then in that case this protocol works.

Proposed ECDH Key Exchange

Figure 2 shows the proposed ECDH key exchange scheme which contains two entities i.e. proxy server and storage server. As the interaction between both the entities are obvious due to frequent data exchange at the times of file uploading or retrieving or at the time of integrity check, there is

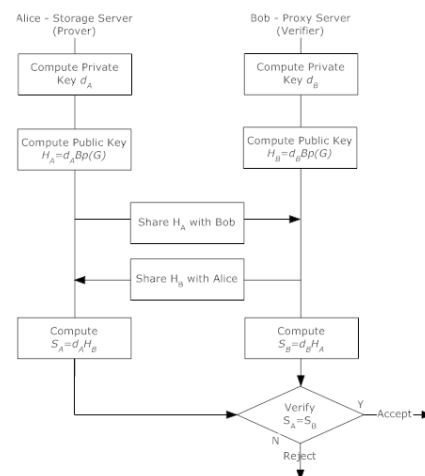


Figure 2. Proposed ECDH Key Exchange.

always a chance that eavesdroppers may listen to the communication and hack the data. We consider proxy server as Bob and storage server as Alice and make a key exchange before they initiate any data exchange. This way it is ensured that any third party trying to spy on the communication is isolated from the system.

Procedure:

1. Setup->An elliptic curve E is chosen by Alice and Bob over F_p with a common base point $Bp(G)$ which is an element of F_p .
2. Alice generates a private key d_A and keeps it confidential. Now, Alice performs point multiplication to generate public key i.e. $H_A = d_A Bp(G)$. Alice sends H_A to Bob.
3. Similarly, Bob generates a private key d_B and performs point multiplication to generate public key i.e. $H_B = d_B Bp(G)$. Bob sends H_B to Alice.
4. Public keys i.e. H_A and H_B are exchanged between Alice and Bob. The Man In the Middle (MITM) would intercept H_A and H_B , but won't be able to find out neither d_A nor d_B without solving the discrete logarithm problem.
5. Alice computes $S = d_A H_B$ using point multiplication of her private key and shared public key of Bob.
6. Bob computes $S = d_B H_A$ using point multiplication of his private key and shared public key of Alice.
7. S is common for both Alice and Bob as $S = d_A H_B = d_A (d_B Bp(G)) = d_B (d_A Bp(G)) = d_B H_A$

MITM only knows H_A and H_B (together with the other domain parameters) but would not be able to find out the shared secret S .

Implementation

System Model

The proposed architecture model consists of seven entities namely, Proxy Server, Certificate Authority, Attribute Authority, Trusted Authority, Data User, Data Owner and Storage Servers. The overall system architecture is illustrated in Figure 3 below:

Proxy Server

Proxy Server (PS) is a centralised entity which acts as a mediator between data users/owners and other cloud entities such as Trusted Authority (TA), Attribute Authority (AA) and Storage Servers (SS). It holds information such as file ID, location and user ID details, keys, hashes, etc. and perform jobs like ciphering & deciphering, fragmentation & merging, hash code generation and various storage server coordination and search request management tasks.

Certificate Authority

Certificate Authorities (CA) is a trusted entity which issues the public key certificate to entities such as data users and Attribute Authority. Approval of CA is must for registration to be part of this system. CA generates unique ID and Public Key for users and other entities.

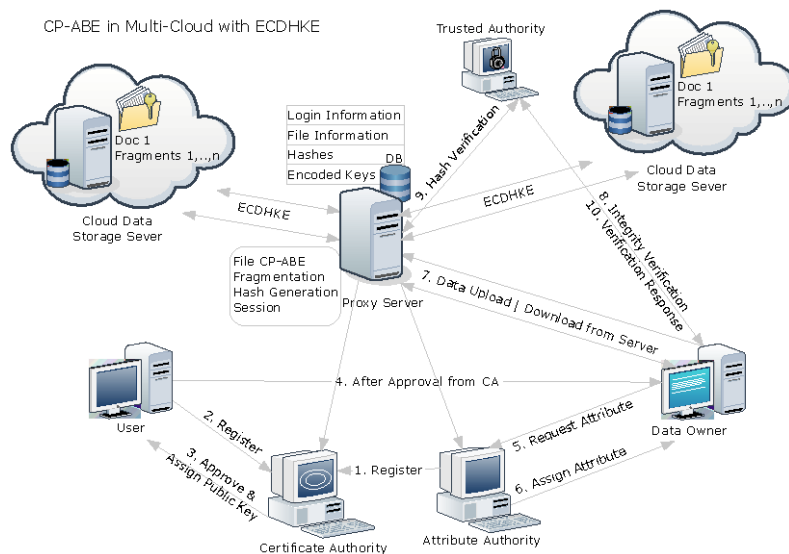


Figure 3. System Architecture.

Attribute Authority

Attributes define the access policy for data users. Attribute Authorities (AA) is an entity which issues attributes to the users and update or revoke it in certain cases. AA generates public attribute keys and issue secret keys to users enrolled with the system. Even at the time of attribute update, AA reassigns the attribute with updated keys and while revoking it takes care of re-keying the attributes. Files are encrypted with the attribute keys so that only legitimate users can decrypt and view files.

Trusted Authority

Trusted Authority (TA) is assumed to most reliable entity in cloud systems. In proposed model it takes care of encrypting secret keys with RSA algorithm so that encrypted keys can get stored in proxy server DB. Also, it generates secret keys for Attribute Authority so that those keys can be associated with the attributes and can be assigned to the registered users of the system. Trusted Authority acts as a judgement entity to validate the data integrity. As any audit request comes, it generates SHA-3 hash of the file fragments and compare it with old hashes to check whether fragment is intact.

Data User

Users of the system are eligible to register themselves and obtain unique registration IDs, certificate and privileged attribute. Data Users (DU) can access the data as per their access rights. A user may have multiple attributes to have a broader access rights. Attributes are assigned by the AA as per their role.

Data Owner

Data users can be a Data Owners (DO) by uploading and outsourcing their data to the system for remote and global access. Access policies on their files are defined by the Data Owners. Attributes are allocated to the files so that only the users who hold those attributes can access the files.

Storage Server

Basic function of Storage Server (SS) is to maintain DB of encrypted file fragments. PS coordinates with SS to store or retrieve the fragment. ECDH key exchange is must between SS and PS to make a secured communication channel. SS are quite secured as they do not store file private keys so they can't decrypt the files residing within them.

Proposed System Modelling

Proposed architecture consists of following stages

Table 1. Notations.

Notations	Meaning	Notations	Meaning
λ	Security Parameter	$PriK_{aid}$	Attribute Authority's Private Key
PSK	Primary Secret Key	$PubK_{aid}$	Attribute Authority's Public Key
UP	Universal Parameter	AVK_{xaid}	Attribute Version Key
$sigk_{CA}$	CA Signature Key	APK_{xaid}	Attribute's Public Key
$valk_{CA}$	CA Validation Key	$AS_{uid,aid}$	Attribute Set
U	User	$SK_{uid,aid}$	Secret Key
uid	User ID	AS	Attribute Set
AA	Attribute Authority	D	Data
CA	Certificate Authority	$EncD$	Encrypted Data
TA	Trusted Authority	C_A	Comprising Authority Set
Det_u	User Details	$DRID$	Deciphering Request ID
UPK_{uid}	Universal Public key of User	$MKUK$	Modifying Key to Update secret key
USK_{uid}	Universal Secret key of User	csk	Content Secret Key
Det_{AA}	Attribute Authority Details	$\widetilde{csk}$	New Content Secret Key

Stage 1: System Setup

We consider four basic operations while setting up the system:

$TAInit(1^\lambda) \rightarrow (valk_{TA}, sigk_{TA})$. Trusted Authority is assumed to be a reliable entity of the system. The trusted authority initialization process takes internal security token λ . A validation key $valk_{TA}$ and a signature key $sigk_{TA}$.

$CAInit(1^\lambda) \rightarrow (PSK, UP, (valk_{CA}, sigk_{CA}))$. The certificate authority initialization process takes internal security token λ . It yields the primary secret key PSK , the universal parameter UP , a validation key $valk_{CA}$ and a signature key $sigk_{CA}$.

$InitServers(PS, |SS|) \rightarrow$ Initialize Proxy Server PS and Storage Servers SS , where $|SS| = \{SS_1, SS_2, \dots, SS_n\}$

$UserSignUp(UP, sigk_{CA}, Det_u) \rightarrow (uid, UPK_{uid}, USK_{uid})$. Signup algorithm of user need certain inputs i.e. the universal parameter UP , $sigk_{CA}$ of Certificate Authority and the details provided by user while signup i.e. Det_u . Upon validation of user it generates a unique user ID uid and allots it to the user. Also, it assigns a universal public key UPK_{uid} and universal secret key USK_{uid} to the user.

$AA\text{SignUp}(Det_{AA}) \rightarrow (aid)$. Signup algorithm of the attribute authority AA need details Det_{AA} as input. Upon validation it generates aid and assigns it to the AA.

$AA\text{Init}(UP, aid) \rightarrow (PriK_{aid}, PubK_{aid}, \{AVK_{x_{aid}}, APK_{x_{aid}}\})$.

The AA initialization process inputs the universal parameter UP and the AA's identity aid and outputs a private key $PriK_{aid}$ and public key $PubK_{aid}$. Also, for every attribute x it generates $\{AVK_{x_{aid}}, APK_{x_{aid}}\}$ as a pair of attribute version key and attribute public key.

Stage 2: Secret KeyGen

$SecKeyGen(PriK_{aid}, UP, \{APK_{x_{aid}}\}, AS_{uid, aid})$

$SK_{uid, aid}$ In this step, to generate the secret key $SK_{uid, aid}$ for the user uid , inputs such as AA's private key $PriK_{aid}$, the universal parameter UP , attribute's public keys array $\{APK_{x_{aid}}\}$, array of attributes $AS_{uid, aid}$, and the user ID uid are provided to the system.

Stage 3: Cipherring

$Enc(D, AP, UP, \{APK_k\}_{k \in C_A}, \{APK_{x_k}\}_{x_k \in S_{A_k}}^{k \in C_A}) \rightarrow EncD$.

$EncD$. To convert data D to encrypted data $EncD$, this function inputs the data D , access policy AP for selected attributes, universal parameter UP , public key set $\{APK_k\}_{k \in C_A}$ by comprising authority set C_A , attributes' public key set $\{APK_{x_k}\}_{x_k \in S_{A_k}}^{k \in C_A}$. Initially, the data D is ciphered by using AES encryption algorithm with a content secret key csk that generates $EncD$. Then, it encrypts the content secret key csk through RSA algorithm keeping access policy AP .

$Verify(ECDHKE(PS, |SS|)) \rightarrow \text{Proxy Server}$ PS verifies array of Storage servers $|SS|$ using key exchange proposed mechanism $ECDHKE(PS, |SS|)$. If it verifies $|SS| = \text{Valid}$ then, $Connect(PS, |SS|)$.

$FragmentData(EncD, TF) \rightarrow \text{Encrypted}$

Data is divided into total no. of fragments TF . Fragment size FR_{size} is calculated as $FR_{size} = S(EncD)/TF$, where $S(EncD)$ is size of encrypted data. Set of fragments is calculated as $|FR| = \{FR_i, FR_{i+1}, \dots, FR_n\}$ Where $FR_i = F'_{FR_{size}}$.

$HashGen(|FR|) \rightarrow \text{SHA3 hash is generated for each fragment as,}$

$\forall FR \in |FR| \text{ do}$

$H_{FR_i} = \text{SHA3Process}(FR_i) \text{ end}$

$\text{StoreHash}(|H_{FR_i}|) \rightarrow \text{All the hashes and fragment IDs are stored by TA.}$

$\forall H_{FR} \in |H_{FR}| \text{ do}$

$TA_{save}(FR_{ID}, H_{FR_i}, FID, TimeStamp)$

end

$\text{StoreFragment}(|FR|) \rightarrow \text{All the fragments and fragment IDs are stored by set of Storage servers } |SS|.$

$\forall FR \in |FR| \text{ do}$

$\forall SS \in |SS| \text{ do}$

$SS_i \rightarrow \text{SaveFrag}(FR_i)$

$PS_{save}(FR_{ID}, H_{FR_i}, FID, TimeStamp)$

end

Stage 4: Deciphering

In this stage a Deciphering Request ID is generated by the cloud servers to initiate the decryption of data:

$\rightarrow DRIDGen(EncD, UPK_{uid}, \{SK_{uid, k}\}_{k \in C_A}) \rightarrow DRID$.

The deciphering request id generation function inputs the encrypted data $EncD$ having access policy AP , user's universal public key UPK_{uid} and a user's secret keys set $\{SK_{uid, k}\}_{k \in C_A}$. Legitimate users having their uid in AP can get $DRID$ for $EncD$.

$Verify(ECDHKE(PS, |SS|)) \rightarrow \text{Proxy Server}$

PS verifies array of Storage servers $|SS|$ using key exchange proposed mechanism $ECDHKE(PS, |SS|)$. If it verifies $|SS| = \text{Valid}$ then, $Connect(PS, |SS|)$.

$FetchData(|SS|)$

Proxy Server PS fetches all the encrypted data fragments from arbitrarily selected storage server SS by,

$$|FR| = PS_{Fetch}(SS_i FR_i, \dots, SS_n FR_n)$$

Fragments are merged together to form the single encrypted data chunk.

$$EncD = \sum_{i=1}^n FR_i$$

$Dec(EncD, DRID, USK_{uid}) \rightarrow D$. To convert encrypted data $EncD$ to plain data D , inputs such as $DRID$, universal secret key USK_{uid} of user need to be provided. It decrypts the content secret key csk through RSA algorithm. And then the resultant csk is used to decipher $EncD$ using AES algorithm to generate data D .

Stage 5: Attribute Revocation

In this stage, AA should update the modified keys, unrevoked user's key should be updated and encrypted data should be re-encrypted.

$\text{ModKeyGen}(PriK_{aid}, \tilde{R}_{aid}, AVK_{\tilde{x}_{aid}}) \rightarrow (\widetilde{AVK}_{\tilde{x}_{aid}}, MKUK_{\tilde{x}_{aid}}, \widetilde{csk}_{\tilde{x}_{aid}}).$

This function inputs the authority private key $PriK_{aid}$, revoking attribute $\tilde{R}_{aid}$ and the prior version key $AVK_{\tilde{x}_{aid}}$. Subsequently, it generates

new attribute version key $\widetilde{AVK}_{\tilde{x}_{aid}}$, key for user to modify secret key $MKUK_{\tilde{x}_{aid}}$ and the new content secret key $\widetilde{csk}_{\tilde{x}_{aid}}$.

$$ASKUpdate(SK_{uid,aid}, MKUK_{\tilde{x}_{aid}}) \rightarrow SK'_{uid,aid}.$$

This function inputs the present secret key $SK_{uid,aid}$ and its modified key to update the key $MKUK_{\tilde{x}_{aid}}$ for all the non-revoked users in the group. Subsequently, a new secret key is generated $SK'_{uid,aid}$ so that data access of the revoked users can be barred.

$$EncUpdate(EncD, csk, \widetilde{csk}_{\tilde{x}_{aid}}) \rightarrow EncD'.$$

Proxy server decrypts old csk using RSA and decrypts the encrypted data $EncD$ using csk . Subsequently, it re-encrypts the decrypted data using $\widetilde{csk}_{\tilde{x}_{aid}}$ to generate new encrypted data $EncD'$ and encrypts $\widetilde{csk}_{\tilde{x}_{aid}}$ using RSA.

Stage 6: Integrity Check

In this stage, TA performs the integrity check of fragments which are doubted for tempering.

$$HashCheck(SS_{ID}, FID) \rightarrow (VR).$$

This function inputs the Storage Server ID SS_{ID} , and fragment ID FID , that needs to be checked for consistency. TA sends request to PS to fetch fragment from particular SS with input ID using,

$$FR = PS_{Fetch}(SS_{ID}, FR_i)$$

Once FR is fetched PS sends FR to TA . TA generates new hash for the fragment as,

$$NH_{FR} = SHA3Process(FR).$$

TA then fetches old hash of the fragment as,

$$OH_{FR} = TA_{FetchHash}(FR_i)$$

New and old hashes are compared by TA as,

$$IntRes = comp(NH_{FR}, OH_{FR}).$$

If $IntRes = true$ then, fragment is considered as genuine in that storage server else it is considered as corrupted.

Results

We estimated the computation time for ECDH key exchange, data encryption, data decryption and SHA-3 hash generation. We do the simulation on a Windows-based system with an Intel I5-2410M CPU at 2.30GHz and 8.0GB RAM. The code uses the BouncyCastle library version v1.48 to simulate the ECDH schemes. We used different curves of ECDH to measure curve efficiency. All the programs were written in Java language running on JDK v1.7. For remote storage of data, 2 Apache servers with two MySQL v5.1 databases (acting as cloud storage servers) are used on local network with 100MBPS bandwidth. All the encryption, decryption and hash generation results are the conducted with different file sizes on an average of 10 trials.

Time Efficiency of Algorithms

In the below Figure 4, time efficiency comparison of various ECDH X9.62 curve variants are shown corresponding to time in milliseconds.

In the below Figure 5, time efficiency comparison of various ECDH SEC curve variants are shown corresponding to time in milliseconds.

In the below Figure 6, time efficiency comparison of various ECDH NIST curve variants are shown corresponding to time in milliseconds.

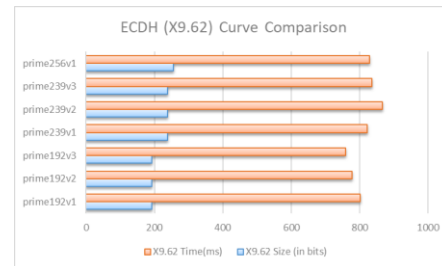


Figure 4. ECDH (X9.62) Curve Comparison.

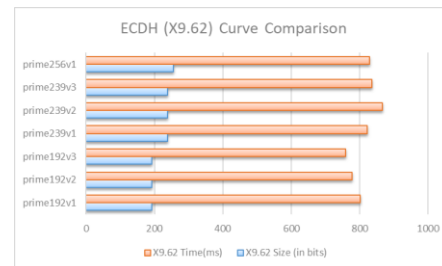


Figure 5. ECDH (SEC) Curve Comparison.

In the below Figure 7 encryption time efficiency of AES algorithm is computed on various file sizes and the time taken is measured in milliseconds.

In the below Figure 8 decryption time efficiency of AES algorithm is computed on various file sizes and the time taken is measured in milliseconds.

In the below Figure 9 hash generation time by SHA-3 algorithm is computed on various file sizes and the time taken is measured in milliseconds.

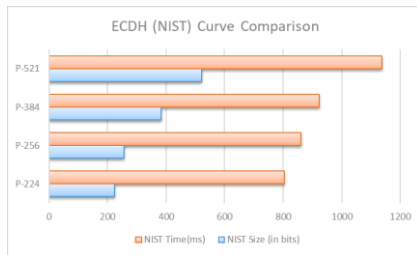


Figure 6. ECDH (NIST) Curve Comparison.

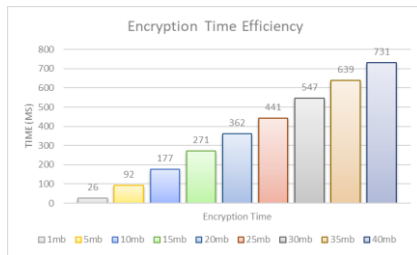


Figure 7. Encryption Time Efficiency.

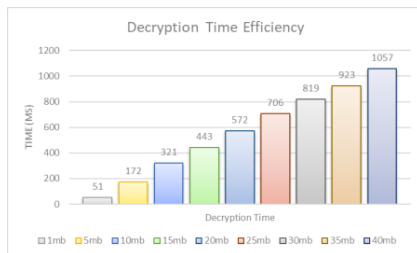


Figure 8. Decryption Time Efficiency.

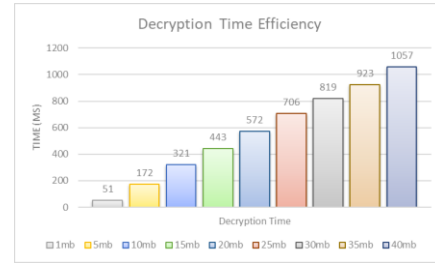


Figure 9. Hash Generation Time.

Functional Comparison

Below in table 2, we have compared functional characteristics of various schemes presented by various researchers with our scheme. Factors that affect security in cloud environment are considered here such as key-exchange mechanism, ABE, replication of data, integrity auditing and revocation procedure. We found with the comparisons that our scheme is quite novel as we don't have any other scheme that takes care of all these parameters in one single system. Primarily, there are very few works on multi-cloud and in that key-change between servers are not studied by any other researchers as per our knowledge. ABE is relatively common in cloud systems but integration of CP-ABE in multi-cloud scenario results in better security. We present the brief comparison among various relevant works in table 2.

Encryption key Security : Symmetric Key, which is used to encrypt the data is again encrypted using RSA public key and get stored on the server. This way, the secrecy of data on server is also maintained and decryption key is also not revealed.

Inter-Server Communication Security : In the proposed scheme, mutual authentication is must i.e., servers first need to authenticate their identity via ECDH key exchange before machining any communication. This way, eavesdroppers can't impersonate IDs of legitimate server and make communication and attacks such as MITM and Node Impersonating attack can be restricted.

Data Security : Data is always encrypted and stored on the server, so it maintains certain amount

Table 2. Functional Comparison

	Year	Multi-Cloud	Inter Server Key-Exchange	ABE Mechanism	Data Replica	Integrity Auditing	Revocation Operation
Zhuang <i>et al.</i>	[15] 2018	Yes	No	No	No	No	No
Li <i>et al.</i>	[16] 2017	No	No	Yes	No	No	Yes
Peng <i>et al.</i>	[17] 2017	No	No	No	Yes	No	No
Yang <i>et al.</i>	[18] 2016	Yes	No	Yes	No	No	No
Xue <i>et al.</i>	[19] 2016	No	No	Yes	No	No	No
Wang <i>et al.</i>	[20] 2020	Yes	No	No	Yes	No	No
Megouache <i>et al.</i>	[21] 2020	Yes	No	No	No	Yes	No
Chang <i>et al.</i>	[22] 2019	Yes	No	No	No	Yes	No
Proposed	2018	Yes	Yes	Yes	Yes	Yes	Yes

of security but after fragmentation of encrypted data it is impossible to retrieve the plain text out of any individual or group of fragments without clubbing all the fragments at one place.

TPA Integrity Verification: Third party auditor provides reliable integrity check on suspicious files. SHA-3 guarantees faster hash generation and ensures higher level of security among its peers such as SHA-1 and SHA-2. **Revocation Management:** At the time of attribute revocation, keys are again generated, data is decrypted using previous key and again encrypted using new key. This secret key is again encrypted and stored. This way users who are out of the group and still hold previous attribute won't get access to data.

Impersonation Attack: The suggested method never sends the user's ID or password over the public channel. Rather, identity and password are hidden using hashing mechanism and additional process is performed on it for security. For authenticating, the technique also employs a high-entropy one-time key that is provided to the user over a separate frequency channel. As a result, the suggested approach is resistant to impersonation attacks.

Phishing attack: The system performs a secure authentication mechanism between the client and the cloud server. Only a legitimate server would deliver accurate user identity information that the client would verify. As a result, the method is also resistant to phishing attacks.

Conclusion

In this paper, we presented an architecture for multi-cloud systems where inter-server communication needs sufficient security. We introduced a security model, which involves not only data security on storage servers and policies for fine grained access control but also a key exchange mechanism to make communications among servers more secured. Inter-server key exchange and use of PKI for symmetric key encryption makes overall scheme very reliable. Standard algorithms such as ECDH, AES, RSA, and SHA-3 adds strong shield against attacks and maintains high efficiency. Our analysis and results demonstrated that proposed scheme is efficient enough to be used in real time scenarios as an extension of CP-ABE in multi-cloud environment.

References

- Bermbach, D., Klems, M., Tai, S., and Menzel, M. (2011). *Metastorage: A federated cloud storage system to manage consistency-latency tradeoffs*. In: *IEEE International Conference on Cloud Computing (CLOUD)*. p. 452.
- Bethencourt, J., Sahai, A., and Waters, B. (2007). *Ciphertext-policy attribute-based encryption*. *Proceedings of the IEEE Symposium on Security and Privacy*, (SP'07):321-334.
- Bethencourt, John., Amit, Sahai., and Brent, Waters. (2007). *Ciphertext-policy attribute-based encryption*. *Security and Privacy, SP'07. IEEE Symposium on*. IEEE.
- Boldyreva, A., Goyal, V., and Kumar V. (2008). *Identity-based encryption with efficient revocation*. *Proceedings of the 15th ACM Conference on Computer and Communications Security*, (CCS'08):417.
- Boneh, D. and Franklin, M.K. (2003). *Identity-based encryption from the Weil pairing*. *SIAM J. Comput.*, 32(3):586-615.
- Bowers, K.D., Juels A., and Oprea A. (2009). *HAIL: A high-availability and integrity layer for cloud storage*. *CCS'09: Proc. 16th ACM Conf. on Computer and communications security*. p. 187-198.
- Cachin, C., Haas, R., and Vukolic, M. (2010). *Dependable storage in the Intercloud*. *IBM Research Report*, 3,783:1.
- Chang, Jinyong. (2020). *Efficient Identity-Based Provable Multi-Copy Data Possession in Multi-Cloud Storage. Revisited*. *IEEE Communications Letters*.
- Chase, M. (2007). *Multi-authority attribute based encryption*. *Proceedings of the 4th Theory of Cryptography Conference*. Springer., 515-534.
- Chen, H., Hu, Y., Lee, P., and Tang, Y. (2013). *Nccloud: A network-codingbased storage system in a cloud-of-clouds*. *IEEE Transation on Computers*.
- Goyal V., Pandey O., Sahai A., and Waters B. (2006). *Attribute-based encryption for fine-grained access control of encrypted data*. *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 89-98.
- Hema, V., Sri Vigna, and Ramesh Kesavan. (2019). *ECC Based Secure Sharing of Healthcare Data in the Health Cloud Environment*. *Wireless Personal Communications*, 108(2)1,021-1,035.
- Koblitz N. (1987). *Elliptic Curve Cryptosystems*. *Mathematics of Computation*.
- Li, J., Huang, X., Li J., Chen, X. and Xiang, Y. (2014). *Securely outsourcing attribute-based encryption with checkability*. *IEEE Transactions on Parallel and Distributed Systems*, 25(8):2,201-2,210.
- Li, Jiguo, Ningyu Chen, and Yichen Zhang. (2019). *Extended file hierarchy access control scheme with attribute based encryption in cloud*

- computing. *IEEE Transactions on Emerging Topics in Computing*.
- Li, W., Li, X., Wen, Q., Zhang, S. and Zhang, H. (2017). Flexible CP-ABE Based Access Control on Encrypted Data for Mobile Users in Hybrid Cloud System. *J. of Com. Sci. and Technol.*, 32(5):974-990.
- Li, W., Xue, K., Xue, Y. and Hong, J. (2016). TMACS: A Robust and Verifiable Threshold Multi-Authority Access Control System in Public Cloud Storage. *IEEE Transactions on Parallel and Distributed Systems.*, 27(5):1,484-1,496.
- Liu, C., Gu, Y., Sun, L., Yan, B., and Wang, D. (2009). RADMAD: High reliability provision for large-scale deduplication archival storage systems. In: *Proceedings of the 23rd International Conference on Supercomputing*, p. 370.
- M. Vukolic. (2010). The Byzantine empire in the intercloud. *ACM SIGACT News.*, 41:105-111.
- McCurley K.S. (1988). A key distribution system equivalent to factoring. *J. of Cryptology.*, 1(2):95-105.
- Megouache, Leila, Abdelhafid Zitouni, and Mahieddine Djoudi. (2020). Ensuring user authentication and data integrity in multi-cloud environment. *Human-centric Computing and Information Sci.*, 10:1-20.
- Miller V.S. (1985). Use of elliptic curves in cryptography. *Advances in Cryptography.*, 1985.
- Mishra, Pragya and Vandani Verma. (2020). Study of Identity-Based Encryption for Cloud Data Security. *Decision Analytics Applications in Industry*. Springer, Singapore. p. 401-408.
- Nagesh, S.H., Kumar, K.A. and Rajgopal, K.T. (2017). Cloud architectures encountering data security and privacy concerns-A review. In: *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. p. 1,729-1,735.
- Papaioannou, T.G., Bonvin, N., and Aberer, K. (2012). Scalia: an adaptive scheme for efficient multi-cloud storage. *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC'12)*. IEEE Computer Society Press., 20.
- Patil, S., Vhatkar P., and Gajwani J. (2014). Towards secure and dependable storage services in cloud computing. *Int. J. of Innovative Research in Advanced Eng.*, 1(9):57-64
- Peng, S., Zhou, F., Wang, Q., Xu, Z. and Xu, J. (2017). Identity-Based Public Multi-Replica Provable Data Possession. *IEEE Access*. v.5., 6,990-27,001.
- Pirretti, M., Traynor, P., McDaniel, P., and Waters, B. (2006). Secure attribute-based systems. In: *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06)*. p. 99-112.
- Premkamal, Praveen Kumar., Syam, Kumar Pasupuleti., and Alphonse P.J.A. (2020). Dynamic traceable CP-ABE with revocation for outsourced big data in cloud storage. *Int. J. of Communication Sys.*, 4,351.
- Priyanka, J. and Ramakrishna, M. (2020). Performance Analysis of Attribute based Encryption and Cloud Health data Security. *4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE.
- Rajgopal, K.T.K.R., Anil, Kumar, and Nagesh, Shenoy. (2018). Load balancing in cloud computing: A survey on popular techniques and comparative analysis. *Global J. of Computer Sci and Technol.*, 18(1)
- Riad, Khaled, Rafik Hamza, and Hongyang Yan. (2019). Sensitive and energetic IoT access control for managing cloud electronic health records. *IEEE Access.*, (7):86,384-86,393.
- Sahai, A. and Waters, B. (2005). Fuzzy identity-based encryption. *advances in cryptology-eurocrypt*. Springer Berlin Heidelberg., 3,494:457.
- Shankar, Achyut, Pandiaraja, P., Sumathi K., Thompson, Stephan, and Pavika, Sharma. (2020). Privacy preserving E-voting cloud system based on ID based encryption. *Peer-to-Peer Networking and Applications.*, 1-11.
- Shenoy, H., Anil Kumar, K.R. and Rajgopal, K.T. (2019). A Comparative Performance Evaluation Approach to Assess Data Security using Cryptographic Algorithms. *Indian Journal of Public Health Research & Development.*, 10(5).
- Wang, Liangmin., Zhendong Yang., and Xiangmei Song. (2020). SHAMC: A Secure and highly available database system in multi-cloud environment. *Future Generation Computer Systems.*, 105:873-883.
- Wang, Pengwei. (2020). Optimizing data placement for cost effective and high available multi-cloud storage. *Computing and Informatics*. 39(1-2):51-82.
- Xiong, Shuming., Qiang, Ni., Liangmin, Wang., and Qian, Wang. (2020). SEM-ACSIT: Secure and Efficient Multiauthority Access Control for IoT Cloud Storage. *IEEE Int. of Things J.*, 4:2,914-2,927.

- Yang, L., Humayed, A., and Li, F. (2016). A multi-cloud based privacy-preserving data publishing scheme for the internet of things. Proceedings of the 32nd Annual Conference on Computer Security Applications-ACSAC 16.
- Youn, Taek-Young. (2019). Authorized client-side deduplication using CP-ABE in cloud storage. Wireless Communications and Mobile Computing.
- Yu S., Wang C., Ren K., and Lou W. (2010). Achieving secure, scalable, and fine-grained data access control in cloud computing. Proceedings of the 29th IEEE International Conference on Computer Communications. IEEE., 1-9.
- Zarandioon, S., Yao, D., and Ganapathy, V. (2012). K2c: Cryptographic cloud storage with lazy revocation and anonymous access. Proceedings of the 8th International ICST Conference on Security and Privacy in Communication Networks. Springer. p. 59-76.
- Zhuang, H., Rahman R., Hui, P. and Aberer, K. (2018). Optimizing Information Leakage in Multicloud Storage Services. IEEE Transactions on Cloud Computing., 1-1.