Journal of Medical Systems (2019) 43: 258
https://doi.org/10.1007/s10916-019-1381-7

MOBILE & WIRELESS HEALTH

Data Access Control and Secured Data Sharing Approach for Health Care Data in Cloud Environment

A. Pugazhenthi [1] · D. Chitra [1]

Received: 4 April 2019 / Accepted: 7 June 2019 / Published online: 1 July 2019

Abstract

One of the vital hitches in today's world of research is ensuring the security of the Cloud. This security can be ensured by inventing new techniques that may possibly help in safeguarding and assuring the optimal status of information in the cloud. Contents that are stored in the Cloud are majorly affected by the attempts made by illegitimate sources that are trying to access it. The security level of the contents that are stored in the cloud can be guaranteed by focusing on a task that helps in both securing and verifying the data. In order to manage the security level, (SE-KAC) which is also known as Scalable and Enhanced Key-Aggregate Cryptosystem was initiated in current research method. But this method of securing and sharing secret keys cannot be used in the stage of decryption. However this problem can be resolved in proposed method since it introduces sharing of the data securely using a method called Improved Diffie Hellman Key Exchange Algorithm (IDHKE). By introducing the Improved Diffie Hellman Key Exchange Algorithm, securely sharing the secret keys to the receivers of the data has been achieved. The secret key details can be exchanged securely using this method. By this means it makes sure its affirmations. Here the key is safely generated using one random prime number, a master secret key and parameter value. Intended for the secured and consistent access control limitation, an encryption which is attribute-based is used. The proposed method thus ensures the protected data transmission with exact and trustworthy validation.

Keywords Cloud security · Access control · Encryption · Secured key sharing · Access policy limitation

Introduction

Sharing of data can be executed using a system that deployed with Cloud. Using this method is beneficial to both the user and organization in several ways [1]. In comparison to manual exchange of data this method is both time and cost efficient [2]. The reason behind this is due to the enormous contribution of data to the cloud from various users. Google Docs is one such tool [3]. This provides opportunity to a group of individuals to successfully connect and share documents. This method is found to be more effective than the previously used methods. The earlier used methods send only the updated version of a document to its members through email attachments [4]. Due to its effectiveness, people are looking forward to use this on phones, computers and laptop [5]. Users are fond of sharing this information worldwide [6]. Due to the feature to work together, the students are widely benefitted. This also increases the efficiency of their work [7].

On sharing data using cloud, privacy and privacy policies of data is questionable. The Cloud provides a variety of services to its users [8]. Thus for the users to adhere to these services a secured setting has to be ensured [9]. While storing the data in the Cloud and meanwhile of sharing the information, the security of the Cloud is more likely to be violated [10]. The following are the list of problems that a user could probably face while trying to store a sensitive information. They are: "Information Stealing, Key Handling Issues, Authentication Issues, and Revocation Handling and so on" [11].

Both the Cloud service providers and users often face the same difficulty. So for a secured storing and sharing of sensitive information, the above stated issues have to be sorted out

This article is part of the Topical Collection on *Mobile & Wireless Health*

✉ A. Pugazhenthi
pugal268@gmail.com

D. Chitra
chitrapacet@gmail.com

[1] Department of Computer Science and Engineering, P. A. College of Engineering and Technology, Pollachi, India

[12]. To attain an environment with complete cloud security, several researches have been conducted [13]. Unique techniques have been innovated by existing methods to overcome the security issue. Despite helping in avoiding the security issues, this method has its own drawbacks performance wise.

This research method majorly aims at introducing a new algorithm for transferring the key. Improved Diffie Hellman Key Exchange Algorithm (IDHKE) is used this method for securely sharing the data contents from cloud server to users. Secret key information can be securely exchanged through this method. Therefore here there is assurance that the required security level can be attained. The key generation process here is secure and it creates impediments such that hacking becomes improbable with the selection process of one random prime numbers in tandem with the master secret key as well as the parameter value. Suggested technique ensures that the data transmission is secure and has been authenticated in a manner that is both precise as well as reliable.

Following is the general outline of the research work. Here, the various aspects behind the cloud security and related issues have been examined. In second section, there are several techniques that have been deployed during the process of research which have been in detail reviewed, discussed and analyzed. In third section, the research methodology has been evaluated and a discussion related to the entailed information has been reviewed using apt examples and subsequently the relevant explanation has also been provided. In fourth section, with regard to the research process and experimental evaluation had been conducted and its inferences drawn have been clearly given along with the evaluated numerical outcome. Finally in fifth section, conclude the overall work.

Related review

Identity-based (ID-based) ring signature was introduced by Xinyi Huang et al. [14]. This method helps us get rid of certificate verification process. Security level of ring signature is amplified by using forward secure ID-based ring signature method. Data owners have to re-authenticate data when single user secret key has been impractical to compromised. Thus in this method, if there is compromising of any secret key of any user, the previously generated signatures of others remains incorporated and the user still remains valid.

An Efficient revocation (EABDS) in cloud computing was suggested by Huang Qinlong et al. [15] This is an attribute-based secure data sharing scheme. Suggested scheme here enables the process of encryption of data with the aid of a DEK or Data encryption key by deploying a process of symmetric encryption. This thereafter encrypts DEK on the basis of the Cipher text policy attribute-based encryption or the CP-ABE. Thus it assures data security and helps in attaining fine-grained access control.

Shared authority based privacy preserving authentication protocol (SAPA) was anticipated by Hong Liu et al. [16] This tackles the privacy issues for cloud storage whereas the present security solutions focuses primarily on authentication. Their protocol is hence attractive for multi-user collaborative cloud applications.

An effectual, flexible and scalable privacy-preserving data policy with semantic security was proposed Xin dong et al. [17] Two techniques are used in this. They are Ciphertext policy attribute-based encryption also known as CP-ABE and IBE or Identity based Encryption. Hence this results in engendering a cloud data sharing service that both reliable as well is secure. This facilitates a vibrant data access to users.

A searchable encryption namely MPSE also known as multi-party searchable encryption was introduced by Qiang Tang et al. [18]. This facilitates users to precisely select and allow one another to execute respective encrypted data. Here worst-case and average-case collusion inherent on account of the user status dynamics what has been taken into consideration and evaluation is a security based model. This new scheme with a proven and established security was suggested.

A vital test in data sharing systems is access policies implementation and policies updates support. This was proposed by Junbeom Hur et al. [19]. The process of using Ciphertext policy attribute-based encryption facilitates data owners to determine the working parameters of their own access policies to preside over attributes of existing users. This thus then results in effectively implementing data policies and data distribution subsequently.

A unique patient centric structure and means for data access control to Personal Health Records (PHRs) was stated by Ming Li et al. [20] These are stored up in moderately trusted servers. Attribute based encryption (ABE) approach is to encrypt patient's PHR file to acquire scalable data access control and fine-grained for PHRs. This strategy partitions users in PHR system to many security domains. Therefore, users and owners reduces key management complexity.

Searchable encryption termed MPSE was introduced by Tang et al. [18] This facilitates users to precisely select and allow to execute their respective encrypted data. Here worst-case and average-case collusion inherent on account of the user status dynamics what has been taken into consideration. This is on account of the user status dynamics. He also suggested a new scheme with attestable security.

Dynamic secure group sharing in public cloud computing environment was framed by Kaiping Xue [21] This structure merges improved Treebased Group Diffie-Hellman (TGDH), proxy re-encryption and proxy signature into protocol. Team leader offers certain privileges to group management. They can select one or more group members. This can be done using proxy signature technique. Session key are secured in digital envelopes whereas data sharing files are placed in Cloud Servers. TGDH scheme modifies group key pair with

dynamism, while in group. This facilitates exit group or joining group does not need entire group members to be in online entire time.

Secured data sharing on cloud environment

The key to this research method is secured sharing of data contents from the cloud server to the users by introducing the Improved Diffie Hellman Key Exchange Algorithm (IDHKE). Secret key information can be securely transferred using this method thereby ensuring the security level that has been promised. The key generation is done securely here and thus it becomes difficult for hacking. It is done by choosing one randomly selected prime numbers in tandem with the master secret key as well as the parameter value. The proposed method ensures the secured data transmission with accurate and reliable authentication. The following sub sections deals with the detailed explanation of proposed research methodology.

System setup

Essentially the framework consists of four main entities: Central Authority (CA), Data Owners (Dos) and Data Receivers (DRs), Cloud.

- **Central Authority (CA):** CA is essentially responsible for gathering and in charge of issuing the cryptographic key for each client as indicated by their trait set and after that divide into two sections (two-factor): former is Secret Part Key (SPK), is considered as potential-uncertain spot (e.g., PC). Later is Security Device Key (SDK) is considered as physically-secured however computationally restricted gadget (security gadget). Moreover, CA is likewise in charge of refreshing each client's security gadget (and the comparing SDK). Uniquely, in the SDK update stage, the CA creates another SDK determines security gadget and relating re-encryption key is transmitted to cloud.
- **Cloud:** As such the cloud may be considered as a semi-trust party which maintains all details related to encrypted shared data and additionally executes tabular maintainence as well. The tabulated form of information contains information regarding the users' universal identity also known as the UID as well as the corresponding re-encryption key. In the event where the DR requests a query related to the shared data, the corresponding cloud actually takes the role of a proxy and then performs re-encryption for the subsequent encrypted data that has been shared data through the deployment of the DR's re-encryption key that corresponds and thereafter which completes the process of re-encryption of data sharing to DR.
- **Data Owner (DO):** A DO is referred to the person who is a user and willingly wishes to share data and other related information with other users (DRs). The data and information that is shared undergoes the process of encryption through the deployment of CP-ABE in lieu of access policy.
- **Data Receiver (DR):** DR is potential or existing client who can get the mutual information from the cloud. At the point when a DR needs to recover the mutual information, the cloud if re-encryption is performed after that profits subsequent re-scrambled ciphertext. Reencrypted ciphertext is decoded utilizing DR's own SDK and SPK, if DR's quality set fulfills entrance approach of mutual information. SDK is never uncovered with security gadget while decoding, whilst an incomplete unscrambling procedure utilizing SDK is executed in security gadget. When security gadget is stolen or lost, DR can disavow and acquire another security gadget by collaborating with CA..

Secured sharing of data stored in cloud storages

Assuming that G1 is bilinear group and has a prime order p, and assuming that g assumed as G1, the generator. In addition, assuming $e : G1 \times G1 \rightarrow G2$ denote bilinear map. Security parameter, k, establish groups sizes. As well as infer Lagrange coefficient $\Delta_{i,\ S}$for $i \in Z_p$ and set S, of elements in Z_p: $\Delta_{i,S}(x) = \pi_{j\in s, j\neq s} \frac{x-j}{i-j}$. Relate attribute with elements in z_p^*. Construction follows.

CA setup

CA framework is modelled by running CA setup, which accepts security factor as information. CA initially picks two multiplicative gatherings G and G_T with similar prime request p and bilinear guide : $G \times G \rightarrow G\_T$. This calculation takes as info two framework parameters, in particular, (a) the most extreme tree profundity d, (b) greatest hub cardinality num. Calculation continues as below. Characterize the universe of genuine qualities $U = \{1,...,n\}$, and $(num - 1)$- estimated sham attributes $〚 u 〛 \hat{}* = \{n + 1,...,n + num\text{-}1\}$. Subsequently, characterize (d, num)- all inclusive access tree T. In continuation, d,num,u,u^*,T will all be accepted as understood contributions to every one of strategies. Presently, for every genuine characteristic j∈u, pick a lot of | T|numbers $\{ 〚 t\hat{}* 〛 \_(j,x)\} x \in \Phi\_T$ consistently at irregular from Z_p. Further, for each fake property j∈u, pick lot of $|\Phi\_T|$ numbers $\{ 〚 t\hat{}* 〛 \_(j,x)\} x \in \Phi\_T$ consistently at irregular from 〚 Z〛 _p. At long last, pick y consistently

at irregular in ⟦ Z⟧ _p. The open PK factors are:

$$\mathrm{GPP} = \mathrm{e}(\mathrm{g},\mathrm{g})^{\mathrm{y}}, \left\{\mathrm{T}_{\mathrm{j,x}} = \mathrm{g}^{\mathrm{t}_{\mathrm{j,x}}}\right\}_{\mathrm{j}\in\mathrm{u},\mathrm{x}\in\Psi_{\mathrm{T}}}, \left\{\mathrm{T}^{*}_{\mathrm{j,x}} = \mathrm{g}^{\mathrm{t}^{*}{}_{\mathrm{j,x}}}\right\}_{\mathrm{j}\in\mathrm{u},\mathrm{x}\in\Phi_{\mathrm{T}}}$$
$$\mathrm{MK\ is}: \mathrm{y}, \left\{\mathrm{t}_{\mathrm{j,x}}\right\}_{\mathrm{j}\in\mathrm{u},\mathrm{x}\in\Psi_{\mathrm{T}}}, \left\{\mathrm{t}^{*}_{\mathrm{j,x}}\right\}_{\mathrm{j}\in\mathrm{u},\mathrm{x}\in\Phi_{\mathrm{T}}} \tag{1}$$

The CA recognizes both AA Registration and User Registration.

User Registration: Each client should enlist to CA amid framework instatement. Client is a lawful client in the framework, the CA at that point doles out an internationally extraordinary client personality uid to this client. For every client uid, the CA initially produces two irregular numbers u_uid; u_uid^' ∈Z_p as its worldwide mystery keys. It at that point creates the client's worldwide open keys.

AA Registration: Every AA ought to likewise enroll itself to the CA amid the framework introduction. On the off chance that the AA is a legitimate specialist in the framework, the CA initially doles out a worldwide credit expert personality help to this AA. At that point, the CA transmits other worldwide open/mystery client (GPK_uid^';GSK_uid) to AA_aid. It likewise transmits check key to AAaid, which is utilized to confirm authentications of clients issued by CA.

AA setup

Let S_{aid} specifies attributes set managed by every attribute authority AA_{aid}. It chooses three randomnumbers α_{aid}, β_{aid}, $\gamma_{aid} \in Zp$ as the authority secret key

$$\mathrm{SK}_{\mathrm{aid}} = (\alpha_{\mathrm{aid}}, \beta_{\mathrm{aid}}, \gamma_{\mathrm{aid}}) \tag{2}$$

where α_{aid} is utilized for data encryption, β_{aid} is to differentiate attributes from diverse AAs and γ_{aid} is for attribute revocation. It constructs public key PK_{aid} as

$$\mathrm{PK}_{\mathrm{aid}} = \left(\mathrm{e}(\mathrm{g},\mathrm{g})^{\alpha_{\mathrm{aid}}}, \mathrm{g}^{\beta_{\mathrm{aid}}}, \mathrm{g}^{\frac{1}{\beta_{\mathrm{aid}}}}\right) \tag{3}$$

For each attribute $x_{aid} \in S_{aid}$ constructs public attribute key as

$$\mathrm{PK}_{\mathrm{x\,aid}} = \left(\mathrm{PK}_{1,\mathrm{x\,aid}} = \mathrm{H}(\mathrm{x\,aid})^{\mathrm{v}_{\mathrm{x\,aid}}}, \mathrm{PK}_{2,\mathrm{x\,aid}} = \mathrm{H}(\mathrm{x\,aid})^{\mathrm{v}_{\mathrm{x\,aid}}\gamma_{\mathrm{aid}}}\right) \tag{4}$$

By selecting an attribute key implicitly as $VK_{x_{aid}} = v_{x_{aid}}$. All public attribute keys $\{PK_{xaid}\}_{x_{aid}\in S_{aid}}$ are available on bulletin board of AA_{aid}, with public key PK_{aid} of the AA_{aid}.

Secret key generation (γ, MK)

Every client is essential to validate to AA_aid to be entitle few traits from AA_aid. Client presents endorsement Certificate(uid) to AA_aid. AA_aid verifies client by utilizing the check key issued by the CA. If it is legitimate client, AA_aid provides numerous traits S_(uid,aid) to client uid as per job or character in its organization space. Else, it prematurely ends. At that point, the AA_aid creates the client's mystery key SK_(uid,aid) by executing mystery key age calculation SKeyGen. Consider a client uid with a property set. The key age calculation yields private key D that empowers A to unscramble message encoded in a (d, num)- limited access tree T' iff $T'(\gamma) = 1$.

Algorithm here continues as it pursues. For every client, pick an irregular polynomial q_x for non-leaf hub x in widespread access tree T. Polynomials are picked in accompanying manner in top-down way, beginning from root hub r. For every x, set degree c_x of polynomial q_x not exactly limit esteem, i.e., c_x = num − 1. Presently, for root hub r, set qr.(0) = y and pick c_r different purposes of polynomial q_r haphazardly to characterize it totally. For some other non-leaf hub x, set q_x (0) = q_parent (x)(index(x)) and pick c_x different guides arbitrarily toward totally characterize q_x. When polynomials have been selected, give accompanying mystery esteems to client:

$$\left\{\mathrm{D}_{\mathrm{j,x}} = \mathrm{g}^{\frac{\mathrm{q_x(j)}}{\mathrm{t}_{\mathrm{j,x}}}}\right\}_{\mathrm{j}\in\mathrm{u},\mathrm{x}\in\Psi_{\mathrm{T}}}, \left\{\mathrm{D}^{*}{}_{\mathrm{j,x}} = \mathrm{g}^{\frac{\mathrm{q_x(j)}}{\mathrm{t}^{*}{}_{\mathrm{j,x}}}}\right\}_{\mathrm{j}\in\mathrm{u}^{*},\mathrm{x}\in\Phi_{\mathrm{T}}} \tag{5}$$

Above secret values is set as decryption key D.

Data encryption (M,PK,T′)

In order for encrypting a message $M \in GPP$, encrypter ϵ decides to select a (d, num)-bounded access tree T'. ϵ Thereafter he selects an assignment of real attributes that corresponds to leaf nodes in T'.

Presently, to have the option to encrypt message M with entrance tree T'., encrypter it to ordinary structure (whenever required). Subsequently ϵ characterizes guide among hubs in T'and all inclusive access tree T. At long last, for every non-leaf hub x in T'., ϵ picks self-assertive (num-k_x)- estimated set ω_x of sham tyke hubs of map(x) in T. Let f(j,x) be boolean capacity to an extent that f(j, x) = 1 if genuine trait j∈U is related with leaf offspring of hub x ∈ T'. what's more, 0 generally. Presently, pick an irregular esteem s∈Z_p and distribute ciphertext E as:

$$< T^{'}, E^{'} = \mathrm{m} > Y^{s}, \left\{ E_{j,x} = T^{s}_{j,map(x)} \right\} j \epsilon u, x \epsilon \Psi_{T'} : f(j,x) = 1, \left\{ E^{*}_{j,x} = T^{*s}_{j.map(x)} \right\} j = att(z) : z \epsilon w_x, x \epsilon \Phi_{T'} \quad (6)$$

Data decryption (E, D)

All clients who are legally users in the framework can openly question any intrigued encoded information. After getting the information from server, client runs unscrambling calculation Decrypt to decode ciphertext by utilizing mystery keys from various AAs. Client has fulfill entrance structure characterized in ciphertext E, client attain substance key. Characterize a recursive calculation Decrypt Node(E,D, x) considers information ciphertext E, private key D, and hub x in T0. It yields gathering component of G2 or ⊥. In initial place, consider situation when x is leaf hub. Let j = att(x) and w be parent of x. At that point, have:

$$\text{Decrypt Node}(\mathrm{E}, \mathrm{D}, \mathrm{x}) = \begin{cases} \mathrm{e}\left(\mathrm{D}_{\mathrm{j,map(w)}}, \mathrm{E}_{\mathrm{j,w}}\right) = \mathrm{e}\left(\mathrm{g}^{\frac{\mathrm{q_{map(w)(j)}}}{\mathrm{t_{j,map(w)}}}}, \mathrm{g}^{\mathrm{s.t_{j,map(w)}}}\right) & \text{if } \mathrm{j} \in \gamma \\ \perp \text{otherwise} \end{cases} \quad (7)$$

which decreases to e(g,g)^(⟦s.t_(j,map(w))⟧ ^) when j∈γ. Assume recursive situation when x is non-leaf hub in T'. Calculation continues as pursues: For hubs z are offspring of x, it needs DecryptNode(E,D, z) and preserves yield as F_z. Furthermore, for every fake hub z ∈ω_x (where ω_x is arrangement of sham hubs of map(x) in T picked by encrypter), it summons capacity DecryptDummy(E,D,z) characterized underneath, and preserves yield as Fz. Give j chance to be spurious trait related with z. At that point, have:

$$\text{Decrypt Node}(\mathrm{E}, \mathrm{D}, \mathrm{x}) = \mathrm{e}\left(\mathrm{D}^{*}_{\mathrm{j,map(w)}}, \mathrm{E}^{*}_{\mathrm{j,x}}\right) = \mathrm{e}\left(\mathrm{g}^{\frac{\mathrm{q_{map(w)(j)}}}{\mathrm{t^{*}_{j}map(w)}}}, \mathrm{g}^{\mathrm{s.t^{*}_{j,map(w)}}}\right) \quad (8)$$

which diminishes to e(g,g)^(⟦s.q_map(x)^ (j)⟧ ^). Let Ω_x be discretionary k_x-sized arrangement of tyke hubs z with the end goal that F_z≠⊥. Further, let S_x be the association of sets Ω_x and ω_x. In this manner have |S_x| = num. Let g^ = e(g,g). If no k_x-sized set x exists, at that point hub x was not fulfilled and capacity returns ⊥. Something else, calculate:

$$\mathrm{F_x} = \prod_{\mathrm{z} \in \mathrm{S_x}} \mathrm{F_z^{\Delta_t, s^{'}_x(0)}},$$

Where i = att(z) if z is leaf node, i = index (map (z)) otherwise $\mathrm{s^{'}_x} = \{\mathrm{i} : \mathrm{z} \in \mathrm{S_x}\}$

$$= \prod_{\mathrm{z} \in \Omega_x} \mathrm{F_z^{\Delta_t, s^{'}_x(0)}} \prod_{\mathrm{z} \in \omega_x} \mathrm{F_z^{\Delta_t, s^{'}_x(0)}} \quad (9)$$

$$= \begin{cases} \pi_{z \in \Omega_x} \left(\hat{g}^{s.q_{map(x)}(i)} \right)^{\Delta_t, s^{'}_x(0)} \text{if } x \in \Psi_{T'} \\ \pi_{z \in \omega_x} \left(\hat{g}^{s.q_{map(x)}(i)} \right)^{\Delta_t, s^{'}_x(0)} \text{if } x \in \Psi_{T'} \\ \pi_{z \in \Omega_x} \left(\hat{g}^{s.q_{map(z)}(0)} \right)^{\Delta_t, s^{'}_x(0)} \text{if } x \in \Psi_{T'} \\ \pi_{z \in \omega_x} \left(\hat{g}^{s.q_{map(x)}(i)} \right)^{\Delta_t, s^{'}_x(0)} \text{if } x \in \Psi_{T'} \\ \pi_{z \in S_x} \left(\hat{g}^{s.q_{map(x)}(i)} \right)^{\Delta_t, s^{'}_x(0)} \text{if } x \in \Psi_{T'} \\ \pi_{z \in \Omega_x} \left(\hat{g}^{s.q_{map(parent(z))}(index(map(z)))} \right)^{\Delta_t, s^{'}_x(0)} \text{else} \\ \pi_{z \in \omega_x} \left(\hat{g}^{s.q_{map(x)}(i)} \right)^{\Delta_t, s^{'}_x(0)} \text{else} \end{cases} \quad (10)$$

$= \prod_{z \in S_x} \hat{g}^{s.q_{map(x)}(0)} = e(g,g)^{s.q_{map(x)}(0)}$ (with polynomial interpolation) and return outcome.

Describe DecryptNode, decryption algorithm merely invokes it on root r′of T′. Sense DecryptNode(E,D,r′) = $e(g,g)^{sy}$ iff T′(γ) = 1 ($F_{r'} = e(g,g)^{s.q_{map(r')}(0)} = e(g,g)^{s \cdot q_r(0)} = e(g,g)^{sy}$, where r is root of universal tree T). As $E^{'} = M \cdot e(g,g)^{sy}$, decryption algorithm merely partitions $e(g,g)^{sy}$ and recovers M.

Secured key sharing method

Diffie-Hellman algorithm is a primary scheme anticipated for trading of keys essential in unbalanced encryption. The security of this calculation can't be undermined on the grounds that few security conventions and administrations rely on Diffie-Hellman key trade for solid correspondence. In this work, have utilized an arbitrary factor to make this calculation increasingly effective. Arbitrary factor creates shared keys for every message is traded among sender and beneficiary. Along these lines, distinctive ciphertext will be created each time notwithstanding for a similar message. In this way, frameworks utilizing this plan will turn out to be progressively tolerant to different assaults. The essential rendition of Diffie Hellman calculation begins with choosing prime number 'q' and crude root 'a' where a < q. Forces of 'a' produces whole numbers from 1 to q-1, for example a mod q, a2 mod q, a3 mod q,… … aq-1 mod q create entire whole numbers from 1 to p-1. Here, primary client chooses an irregular characteristic number 'I' as private key. Open key for a similar client is determined as ai mod q. Likewise next client chooses its

private key as 'j'. Second client's open is created similarly as above. The two clients at that point trade their open keys. Every client utilizes its own private key and other client's open key to register common mystery key. Utilizing this mystery key, encryption/decoding happens.

Algorithm:

1. Choose prime number 'q' and primitive root 'a' (1<= a <=q).
2. User 1:

- Choose pr(1)=i
- pu(1)= a^i mod q

3. User 2:

- Choose pr(2)=j
- pu(2)= a^j mod q

4. Both parties swap public keys.
5. User1:

- Shared secret key, K= $pu(2)^{pr(1)} = a^{ji}$ mod q

6. User 2:

- Shared secret key, K=$pu(1)^{pr(2)} = a^{ij}$ mod q

Just realize private keys I, j as they are not broadcasted and can't be captured. Therefore, imparting gatherings can figure mutual mystery key. Anyway there is no chance to validate parties as personality isn't connected to keys they share. Therefore, calculation is as yet inclined to man-in-the-middle attack. To maintain a strategic distance from such assaults, open key endorsements and advanced marks are utilized, as depicted in Authenticated Diffie Hellman Key Agreement Protocol. Likewise, mutual key stays consistent for session, similar message while scrambling numerous occasions will give the equivalent enigmatic content each time. In this way, much of the time happening designs in ciphertext is utilized to discover connections between ciphertext and plaintext. Along these lines, to maintain a strategic distance from known plaintext assaults, present an irregular factor in key generating protocol that an alternate ciphertext is created for equivalent plaintext every time it is encoded.

Improved Diffie Hellman algorithm

Here, choose prime number 'q' and crude root 'a', where a < q. Principal client at that point chooses an arbitrary normal number 'I' as private key. Open key is determined as ai mod q. Also, client chooses an irregular normal number 'j' as private key. Open key is provided as aj mod q. Open keys are traded amongst open channel. The two clients presently chooses an arbitrary number 't', 's', 0 < t,s < q. Irregular whole number 't,s' is camouflaged and traded amongst two clients. Other client can extricate estimation of irregular whole number from message as an information of private is an open keys. Mutual mystery key is determined by main client utilizing its private key, subsequent client's open key and irregular numbers 't,s', via exponentiation in limited field. Mystery key is utilized to encode and decode message.

Algorithm:

1. Choose prime number 'q' and primitive root as 'a' (1<= a <=q).
2. User 1:

- Choose pr(1)=i, 0<i<q
- pu(1)= a^i mod q

3. User 2:

- Choose pr(2)=j, 0<j<q
- pu(2)= a^j mod q

4. Both parties swap its public keys.
5. User 1:

- Choose random integer 't', 0<t<q
- Compute x= pu(1)pr(2) = a^{ij} mod q
- Transmit t.a^{ij} to User 2.

6. User 2:

- Choose random integer 's', 0<s<q
- Compute y= pu(1)pr(2) = a^{ij} mod q
- Transmit s.a^{ij} to User 1.
- Haul out 't' as t=(t.a^{ij})/y

7. User 1:

- Haul out 's' as s=(s.a^{ij})/x
- Shared secret key, K= $pu(2)^{t.s.pr(1)} = a^{ts.ji}$ mod q
- Encrypt plaintext as: C=E(M,K).

8. User 2:

- Shared secret key, K=$x^{t.s} = a^{t.s.ji}$ mod q
- Decrypt cipher text as: M=D(C,K)

For every message random factor differs and also has an therein has an inherent shared secret key. This results in improving Diffie Hellman algorithm security and helps address issues and other plaintext attacks. In order to generate a new ciphertext a similar block of text undergoes the process of

Table 1 Confidentiality comparison values

Delegation ratio (%)	SE-KAC	IDHKE
0.1	25.2	27
0.2	38	41
0.3	44	46.8
0.4	58	62.5
0.5	69	72
0.6	78	79.8
0.7	80	83.4
0.8	87	89.6
0.9	96	98

encryption by using a varying key every time the need arises. In this scenario presuming that an attacker attempts to intercept messages and tries to create impediments during the process of transmission. The attacker will be unable to map already determined sets of ciphertext and plaintext as both of these would differ and the attacker will fail in his attempt to intercept the transmission. Additionally deriving keys based on a message like the aforesaid would be impossible.

Results and discussion

Simulation outcomes are analyzed based on prevailing and proposed approach. Performance assessment is attained by matching this methodology with exisiting approached based on specific factors. Here an exisiting approach known as SE-KAC is public-key cryptosystems, which encrypts message utilizing public key and cipher text identifier termed as class [22]. Here, Improved Diffie Hellman Key Exchange Algorithm (IDHKE) is anticipated to offer efffectual healthcare information security. Performance is analyzed

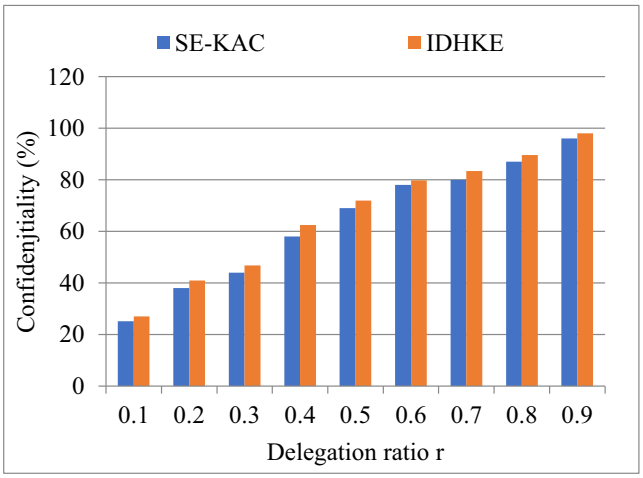

Fig. 1 Confidentiality

Table 2 Integrity comparison values

Integrity (%)		
Delegation ratio (%)	SE-KAC	IDHKE
0.1	21.4	23.5
0.2	27.5	29
0.3	38.4	39
0.4	45.9	47
0.5	54.9	58
0.6	68.3	69.8
0.7	78.3	79.2
0.8	88.3	89.5
0.9	95.8	97.2

based on integrity, confidentiality, resource utilization rate and user satisfaction degree.

Confidentiality comparison

Table 1 depicts comparison of prevailing SE-KAC with anticipated IDHKE model for confidentiality analysis. Here, delegation ratio is 0.9, confidentiality is 96% in IDHKE.

Figure 1 depicts comparison of previling SE-KAC approach with anticipated IDHKE model. X-axis measures delegation ratio. Y-axis measures confidentiality. Delegation ratio is depicted as delegated cipher text classes proportion to total classes. Anticipated approach is termed as IDHKE uses dual encryption with cipher text identifier termed as class. In proposed IDHKE technique, produces higher confidentiality rate of 98.00% for delegation rate (0.9), whereas the existing methods produces confidentiality rate of 96.00%. Experimental outcomes validate IDHKE technique with superior confidentiality to deal with SE-KAC approaches.

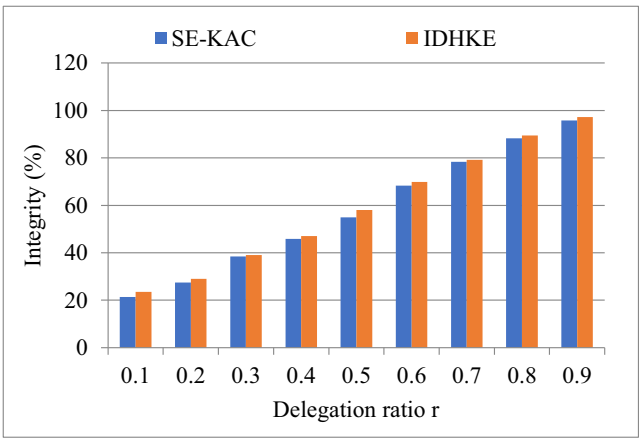

Fig. 2 Integrity

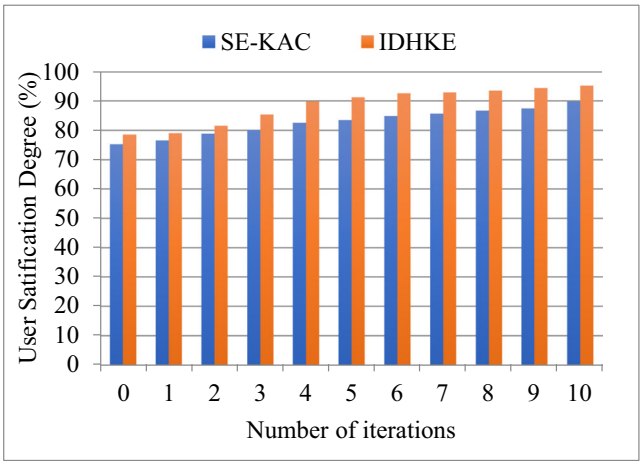

Fig. 3 User satisfaction degree as number of iterations

Integrity comparison

Table 2 depicts analysis of prevailing SE-KAC and anticipated IDHKE for integrity measurement. Here delegation ratio is 0.9, integrity is 95.8% in IDHKE.

Figure 2 shows analysis of prevailing SE-KAC approach with anticipated IDHKE for integrity measurement. X-axis determines delegation ratio. Y-axis determines integrity. Delegation ratio is defined as ratio of surrogated cipher text classes to total classes. Existing SE-KAC is public-key cryptosystems, that encrypts message with public key and cipher text identifier termed as class. Current IDHKE uses dual encryption method with cipher text identifier termed as class. IDHKE technique offers superior integrity rate of 97.2% for delegation rate (0.9), whereas the existing methods produces integrity of 95.8%. Experiment validates that IDHKE method acquires superior integrity while evaluating with KAC method.

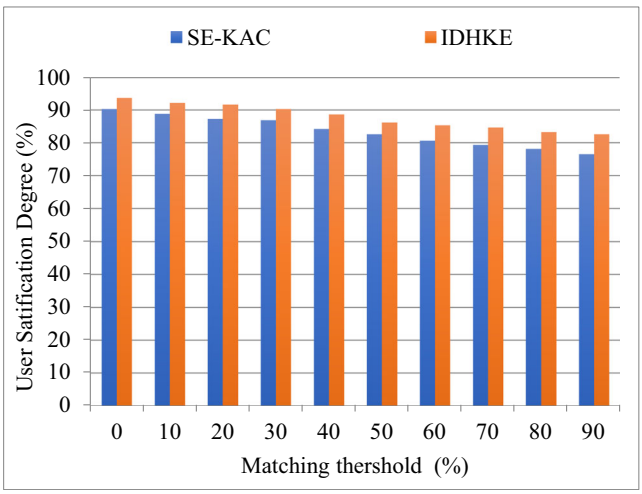

Fig. 4 User satisfaction degree as matching threshold varies

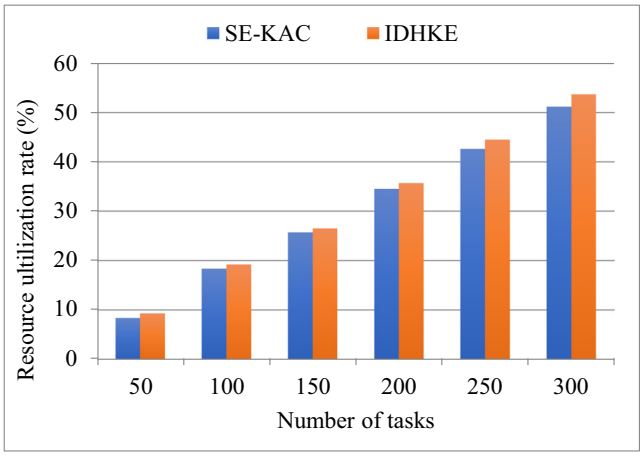

Fig. 5 Resource utilization rate

User satisfaction degree

User satisfaction degree with cloud resource is distinct as QoS provided by resource to user typically. There are two methods to analyze QoS: objective and subjective. Subjective demands feedback information from users till resource usage completion. Here, an objective approach is selected, where higher satisfaction degree specifies closer comparison amongst user's resource request and ability of resource selection.

It is observed from Fig. 3, IDHKE attains 80% user satisfaction degree after two iterations, and increases to 90% as iterations raises to 4. In IDHKE algorithm, user preference and implemented is allocated to every clusters sourced on users' resource requirements, which chooses resources sequence where final one is nearer to user's request than prior selection. Continuous feedback integration would facilitate cloud provider to incrementally elicit user's need from resource request. Investigation based on these demands human subject analysis and provided for future studies. In proposed IDHKE technique, produces higher user satisification degree of 95.18% for number of iterations (10), whereas the existing methods produces user satisification degree of 89.81%.

As depicted in Fig. 4, with matching threshold value that changes from 0 to 90%, user satisfaction degree offered by CUP reduced by 10%. In the proposed IDHKE technique, produces higher user satisification degree of 82.55% for matching thershold (90), whereas the existing methods produces user satisification degree of 76.52%.

Resource utilization rate

Resource utilization rate is defined differently in cloud computing environment. It is the ratio of total quantity of allocated resources to total amount of available resources when CUP initiates. It is a crucial performance indicator, openly associated with cloud provider profit called gain.

Here, comparison between SE-KAC and IDHKE depicts resource utilizations rate performacne. It is observed from Fig. 5 that resource utilization rate rises as number of users rises from 50 to 300. This is due to the fact that more resources are occupied for numerous tasks. Moreover, IDHKE resource utilization rate is extremely superior in contrast to prevailing approaches like SE-KAC. Therefore, IDHKE attains superior resource utilization rate by eliminating huge capability resource to user task with high priority and lower demand, providing high-capability resources to high-demanding users. In the proposed IDHKE technique, produces higher resource utilization rate of 53.68% for number of tasks (300), whereas the existing methods produces resource utilization rate of 51.21%.

Conclusion

In this work, secured sharing of the secret keys to the data receivers has been optimized by introducing the Improved Diffie Hellman Key Exchange Algorithm. This method be able to securely exchange the secret key information, thus the ensured security level can be guaranteed. Here the key generation is done securely by choosing one random prime numbers along with master secret key and parameter value. In this work attribute based encryption method is adapted for the secured and reliable access control limitation. The proposed method guarantees the secured data transmission with accurate and reliable authentication. The performance comparison results of the proposed and existing methods are measured in terms of the data confidentiality, data integrity, resource utilization rate, user satsfication degree. The overall implementation of the research method is done in the cloudSim environment from which it is proved that the proposed method leads to provide the optimal outcome than the existing research methods.

References

1. Lanc, D., Fan, L., Mackinnon, L., and Buchanan, B., U.S. Patent Application No. 15/383,540, 2017.
2. Syed, S. M. A., Mohammad, G. K. B. A., and Halgamuge, M. N., The much needed security and data reforms of cloud computing in medical data storage. In: *Applying Big Data Analytics in Bioinformatics and Medicine*, 2017, 99.
3. Mościcki, J. T., and Mascetti, L., Cloud storage services for file synchronization and sharing in science, education and research. *Futur. Gener. Comput. Syst.* 78:1052–1054, 2018.
4. Hampton, S. E., Jones, M. B., Wasser, L. A., Schildhauer, M. P., Supp, S. R., Brun, J. et al., Skills and knowledge for data-intensive environmental research. *Bioscience* 67(6):546–557, 2017.
5. Ghosh, R. K., Mobile distributed systems: Networking and data management. In: *Wireless Networking and Mobile Data Management*. Singapore: Springer, 2017, 3–20.
6. Veletsianos, G., and Shaw, A., Scholars in an increasingly open and digital world: Imagined audiences and their impact on scholars' online participation. *Learn. Media Technol.* 43:1–14, 2017.
7. Madhav, N., and Joseph, M. K., Cloud for engineering education: Learning networks for effective student engagement. In: *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*. IEEE, 2017, 1–4.
8. Sethi, S., and Sruti, S., Cloud security issues and challenges. In: *Resource Management and Efficiency in Cloud Computing Environments*. IGI Global, 2017, 89–104.
9. Patel, A. K., Cloud storage and its secure techniques. *Int. J. Eng. Sci.* 7:6603, 2017.
10. Wang, L., Ranjan, R., Chen, J., and Benatallah, B. (Eds), *Cloud computing: methodology, systems, and applications*. Boca Raton: CRC Press, 2017.
11. Nikkhah, H. R., and Sabherwal, R., A privacy-security model of mobile cloud computing applications, International Conference on Information Systems(ICIS). Security, 2017.
12. Mushtaq, M. O., Shahzad, F., Tariq, M. O., Riaz, M., and Majeed, B., An efficient framework for information security in cloud computing using auditing algorithm shell (AAS). arXiv preprint arXiv: 1702.07140, 2017.
13. Ali, M., Dhamotharan, R., Khan, E., Khan, S. U., Vasilakos, A. V., Li, K., and Zomaya, A. Y., SeDaSC: Secure data sharing in clouds. *IEEE Syst. J.* 11(2):395–404, 2017.
14. Huang, X., Liu, J. K., Shaohua Tang, I. E. E. E., Xiang, Y., Liang, K., Li, X., and Zhou, J., Cost-effective authentic and anonymous data sharing with forward security. *IEEE Trans. Comput.* 64(4): 971–983, 2015.
15. Huang, Q., Ma, Z., Yang, Y., Fu, J., and Niu, X., EABDS: Attribute-based secure data sharing with efficient revocation in cloud computing. *Chin. J. Electron.* 24(4):862–868, 2015.
16. Liu, H., Ning, H., Xiong, Q., and Yang, L. T., Shared authority based privacy-preserving authentication protocol in cloud computing. *IEEE Transactions on Parall and Distr* 26(1):241–251, 2015.
17. Dong, X., Yu, J., Luo, Y., Chen, Y., Xue, G., and Li, M., Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing. *Computers & Security* 42:151–e164, 2014. ScienceDirect journal homepage: www.elsevier.com/locate/cose. Elsevier Ltd 2013.
18. Tang, Q., Nothing is for free: Security in searching shared and encrypted data. *IEEE T INF. FOREN. SEC* 9(11):1943–1952, 2014.
19. Hur, J., Improving security and efficiency in attribute-based data sharing. *IEEE Trans. Knowl. Data Eng.* 25(10):2271–2282, 2013.
20. Li, M., Yu, S., Zheng, Y., Ren, K., and Lou, W., Scalable and secure sharing of personal Health Records in Cloud Computing using attribute-based encryption. *IEEE T. PARALL. DISTR* 12:743–754, 2012.
21. Xue, K., and Hong, P. A dynamic secure group sharing framework in public cloud computing. *IEEE T CLOUD. COMPUT* 2(4):459–470, 2014.
22. Pugazhenthi, A., and Chitra, D., Secured and memory overhead controlled data authentication mechanism in cloud computing. *Clust. Comput.*, 2018.