International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/54 Volume 11, Issue 6, November – December (2024)

RESEARCH ARTICLE

Enhancing Secured Data Sharing in Mobile Cloud Environments Using Self-Generative Schnorr Certificateless Signcryption

Sowmya V L
Department of Computer Science and Engineering, BMS Institute of Technology and Management, Bengaluru, India.
soumyavl@bmsit.in

Shankar R
Department of Computer Science and Engineering, BMS Institute of Technology and Management, Bengaluru, India.
shankar@bmsit.in

Anitha Premkumar
Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, Bengaluru, India.
p.anitha@manipal.edu

Umi Salma Basha
Computer Science and Engineering, Jazan University, Gizan Saudi Arabia.
ubasha@jazanu.edu.sa

Rajesh Natarajan
Information Technology Department, College of Computing and Information Sciences, University of Technology and Applied Sciences, Shinas, Oman.
✉ rajesh.natarajan@utas.edu.om

Received: 29 September 2024 / Revised: 27 November 2024 / Accepted: 10 December 2024 / Published: 30 December 2024

Abstract – Mobile cloud computing uses cloud infrastructure to provide apps and services to mobile devices. In this scenario, cloud computing promotes data sharing among authorised users by providing access to resources stored on Cloud Servers (CS). Data sharing is the process of distributing data to various users or applications; however, providing secure access and data protection remains a major concern. To solve this, a novel technique known as Random Self-Generative Schnorr Certificateless Signcryption based Secured Data Sharing (RSGSCS-SDS) is presented to improve data confidentiality in mobile cloud environments. The RSGSCS-SDS approach begins with mobile. In order to access different services, cloud users must register their data with CS. Upon registration, CS uses the Random Self-Generative Schnorr Certificateless Signcryption method to create a unique user's own set of public and private keys. Employing using particular policy attributes, when a user asks for access to data, the server verifies their authorisation. While the user is validated, the server returns the requested data in ciphertext with a corresponding digital signature. The user then checks the signature before decrypting and accessing the contents. The user cannot access the original data until the validity of the signature has been established, ensuring secure access. Outcome of RSGSCS-SDS approach providing protected data distribution. Proposed approach shows better performance of secure data sharing based on metrics of data confidentiality, data integrity, computing time, storage overhead and delay. It shows less delay and storage overhead in addition to improving data confidentiality and data integrity compared to the existing certificateless signcryption method.

Index Terms – Mobile Cloud Computing, Cloud Server, Data Secrecy, Random Self-Generated Schnorr Certificateless Encryption, Ciphertext, Signature.

1. INTRODUCTION

The development of mobile cloud computing increases businesses by storing their data in the cloud and securely sharing it with authorised staff with improved productivity. Mobile cloud computing is utilized for storing, processing, and accessing data. With the rapid expansion of cloud servers, a large amount of data is shared through cloud computing, enabling different users to share data under specific access control policies. The distribution of data among mobile devices causes challenges in protecting sensitive information from unauthorized access, interception, and corruption during data access. With maximum security, many cryptographic techniques have been developed. However, distributing sensitive data in cloud environments involves difficulties. It includes protection of user privacy when using mobile devices with limited resources and guaranteeing lightweight operations.

Data was accessed [1] by Multi-Authority Ciphertext Policy Attribute Based Encryption with Elliptic Curve Cryptography (MA-CPABE-ECC) with a lesser key size. However, it did not significantly improve data confidentiality and data privacy. The lightweight data sharing strategy (LDSS) enabling computing in the cloud using handheld gadgets has been set forward in [2]. In a cloud setting, LDSS used CP-ABE in combination with access control technology. A considerable degree of complex access control tree modification was employed by LDSS to route requests as of mobile tools to exterior proxy servers. However, the amount of storage overhead was not decreased by LDSS.

The Modified Group Key Protocol Version (MGPV) protocol was proposed in [3] to protect against all possible assaults. The group protocol effectively decreases computing complexity while ensuring data access. The data is accessible to authorised group members. Despite the decrease in computational complexity, the technique did not result in a reduction in time consumption.

Medi-Block, as described in [4], with the combination of tamper-proof and anonymous identity management mechanisms. The designed block is specifically used for the secure sharing of medical documents. The block architecture employs bilinear mapping techniques throughout the authentication step, completely eliminating the need for third-party trust. It fails to provide security for data. To address data security problems, a bio-computing approach was introduced in [5].

In order to improve security during data transmission and prevent resource duplication among communicating users, a privacy-preserving decision-making technique for information-sharing was developed [6]. A Peer-to-Peer Cloud System (P2PCS) was utilised in [7] to handle and analyse large amounts of data, part of an effective hybrid mobile cloud computing strategy built around the concept of cloudlets. However, the rate of data secrecy was not threatened by this hybrid design. Federated Learning was used in [8] to illustrate the FL2S protocol for IoT. This method created a hierarchical asynchronous federated learning approach that depends on responsive task decomposition to enable secure information exchange. Our FL2S method did not, however, succeed in reducing storage overhead as planned.

Ciphertext-Policy Attribute-Based Signcryption is the basis of the privacy-preserving information access management method that was presented [9] to allow fine-grained administration while protecting attribute privacy in multi-authority cloud storage systems. However, computational complexity failed to be minimised. In [10], a cryptographic strategy was created that involves client-side information encryption prior to entering the cloud, using a multifold symmetric-key cryptography approach based on DNA encryption. Nonetheless, this approach failed to minimise computing complexity. The research identifies drawbacks such as high computational complexity, increased storage overhead, longer computation times, decreased data secrecy, and reduced security. To solve these issues, a novel solution termed the Random Self-Generative Schnorr Certificateless Signcryption-based Secured Data Sharing (RSGSCS-SDS) Technique has been developed for usage in mobile cloud environments.

RSGSCS-SDS contribution has for improving data secrecy. In mobile cloud, secure information access obtained to various services, users must first register their information with the server and perform Schnorr Certificateless Signcryption.

- The main purpose of developing the RSGSCS-SDS technique is to improve data confidentiality levels in mobile cloud environments.
- To increase data confidentiality and integrity, a Random Self-Generative Schnorr Certificateless Signcryption is employed in the RSGSCS-SDS technique.
- Every registered user's private and public keys are generated by mobile CS using Random Self-Generative Schnorr Certificateless Signcryption. When the keys are generated, the user contacts the mobile cloud server with a request.
- The server in the cloud authenticates them using policy attributes. The user receives the requested data in ciphertext and a signature from the mobile cloud server following verification.
- Decrypting the information involves verifying the user's signature. Security of access ensured, authorized individuals can access innovative data when the signature is confirmed as authentic.

There are six sections in the paper. A summary of earlier studies on safe data sharing in cloud systems may be found in Section 2. The suggested RSGSCS-SDS method is fully explained in Section 3 as well as is accompanied by a concise demonstration. Three parameters are used in an experimental evaluation, as described in Section 4. In Section 5, outcomes are compiled, and the recommended strategy is contrasted with the most advanced techniques available today.

2. RELATED WORKS

A secure and effective entrance control paradigm for cloud sources attribute-based encryption-based technique was developed [11]. The encrypted ciphertext is shared over a DHT network. However, this model did not minimise storage complexity. In [12], a PEKS system was designed for string searching, allowing users. Group of cloud users access their data. The access control accesses encrypted files depending upon roles. However, it fails to address issues of space complexity.

PROUD, launched in [13] alongside the ABSC solution, was intended to offload the data decryption procedure and reduce the computing cost on the user part. It enabled end users to reduce decryption overhead while verifying partially decrypted data received from the edge server. However, PROUD did not shorten the overall computation time. In [14], end-to-end encryption (E2EE) is provided by hybrid cryptographic techniques. Cryptographic technique overcomes issues performance presented by multimedia cloud providers to achieve secure data access. Multimedia cloud computing improves data integrity and secrecy. This tactic boosted data confidentiality but did not reduce computing expenses.

In [15], Secure Data Access and Sharing Scheme explained. Users who entered right password and used biometric authentication could access cloud storage provider. This technique, however, did not reduce the amount of time spent transferring data. An anonymous attribute-based broadcast encryption (A2B2E) approach with a hidden access policy was first described in [16]. This strategy allowed data owners to share their information with multiple participants based on the access rules. However, the A2B2E algorithm did not lower computing complexity.

In [17], a comprehensive and useful approach was provided, wherein cloud computing enabled data centre-based information sharing between users and cloud service providers (CSPs). In a cryptographic system, data is accessed and protected by an authenticated cloud user. Here, the cloud service provider enhances data privacy. However, the method did not achieve higher integrity. Computational complexity was not addressed by this method, though. ASDS system was established [18] to improve user revocation flexibility and data access management in cloud environments. Although this strategy was effective in thwarting replay and collusion assaults, it did not lower storage overhead.

An authority-verified, privacy-preserving CP-ABE method through constant-size secret keys was devised [19]. Selective security has been achieved by employing the above approach towards the decisional n-BDHE issue. In [10], a novel method created by asymmetric key technology improves security in cloud environments by locking keys during file exchange. However, the system did not take time complexity into account. In [21], Secure Authentication and Data Sharing in the Cloud (SADS-Cloud) consist of three processes. However, the data confidentiality rate was not enhanced.

Rescue Chain has been presented in [22] for secure and efficient information-sharing for Unmanned aerial vehicles (UAVs)-assisted disaster rescue. Here, UAV data is processed and stored in a vehicular fog computing system with the aid of idle computing resources. The processing resources are identified as an optimal allocation strategy to achieve better data allocation. However, storage overhead was not minimized.

Policy hiding in MCC setting employed [23] by privacy-preserving access control method for supporting encryption and effective policy updates. However, time was higher. In [24], secure data-distribution system was designed. Encrypted data helps protect confidentiality However, storage complexity was not decreased. In [25], a proficient and secure data-sharing system was presented with higher safety. Next, cloud user verification is performed to share data by authorized users to reduce incorrect computation. At last, lightweight operation is provided to verify data by the owner and data requester sides. But Computational complexity was not reduced.

In [26], an efficient, Provably Secure Data Selection Sharing Scheme (EPSDSS) was developed. The trust system effectively manages centralized trust models to attain higher dependability and accountability of data. Here, data owners carry out secure access. Thus, it shares data and generates information with better cloud storage.

In [27], a safe and distributed IoT data storage approach was presented. For providing shorter messages from original messages by IoT devices, an ultra-lightweight secret sharing algorithm is carried out. Lastly, a balanced index structure is carried out to obtain improved IoT data retrieval efficiency based on blockchain information. However, time complexity was not decreased. SLFG-DSS was developed in [28]. In the sharing scheme, a resisting decryption key operation was used to access data in a secure manner. During data decryption, user data and data owners are verified with higher security level requirements. However, storage overhead was not minimized.

In [29], a privacy-preserving access control model was presented to perform secure data access in the mobile cloud. In addition, fully outsourced attribute-based signcryption (ABSC) was carried to access data by data owner. It helps achieve high security and less computation time. Mobile data security achieved by RPO-CFE-SMC was developed in [30].

With the purpose of AES chaotic fuzzy encryption and red panda optimization algorithm, user tasks are accessed by mobile devices with minimum energy consumption.

A proficient and secure data-sharing method was presented [31]. User's details confirmed to find official as well as illegal users to avoid unauthorized mobile data access. Thus, secure data access by the data owner and data requester is obtained through lightweight operations. However, it has a higher complexity during data sharing. MECC system portrayed [32]. The cryptosystem carries authentication, data compression, and safe data transfer approaches to secure data access by cloud servers. Though processing time was reduced, the data confidentiality rate was not efficient.

An authentication and authorization scheme were designed in [33] for distributing data among mobile services. The authentication phase accurately identifies mobile users who can access different services with minimized space complexity. Information distributed [34] with CP-ABE. It effectively shares access and stores mobile data with higher data security. However, the computational complexity of the algorithms was higher. MP-RAGBE was introduced [35] to increase data transmission efficiency and security in mobile cloud computing. However, it failed to reduce time.

Based on the above-mentioned issues in mobile cloud computing, proposed work to increase confidentiality with minimum delay has been developed.

2.1. Summarization Table of the Related Work

In this section, summarization tabulation is illustrated in Table 1 below.

Table 1 Summarization Table

Method	Contribution	Merits	Demerits
MA-CPABE-ECC [1]	Data encryption and Elliptic Curve Cryptography	Higher security and lesser computation time	Issues on data confidentiality and data privacy
LDSS [2]	CP-ABE combined with access control technology	Secure data access with minimized time	High storage overhead
MGPV [3]	Group protocol for data accessing	Computational complexity is reduced	Time consumption is high
Medi-Block [4]	Tamper-proof and anonymous identity management mechanism	Efficient user authentication	Failed to provide data security
Biocomputing approach [5]	Polymerase chain reaction (PCR) and primer production techniques	Improves security	Delay occurs
Privacy-preserving decision-making technique [6]	Prevent resource duplication and provide information sharing	Security during data transmission	Higher complexity
P2PCS [7]	Effective hybrid mobile cloud computing strategy	Data handling	The rate of data secrecy was not threatened
Federated Learning [8]	Hierarchical asynchronous federated learning approach	Secure information exchange	Reducing storage overhead is difficult
Ciphertext-Policy Attribute-Based Signcryption [9]	Privacy-preserving information access management method	Protect data attribute privacy	Decrease in computational complexity
Multifold symmetric-key cryptography approach [10]	Cryptographic strategy	Better data encryption	Increased storage overhead
Secure and effective entrance control paradigm [11]	ABE, DHT, and IDTRE carry out	Obtain cipher text for secure access	Fails to reduce memory utilization

PEKS [12]	RBAC for several-user PEKS	cloud users access	Issues on space complexity
PROUD [13]	Decryption procedure	Minimizes computation cost while decrypting original user data	PROUD did not shorten the computation time
Hybrid cryptographic techniques [14]	E2EE solution	Improves data secrecy	Did not reduce computing expenses
Secure Data Access and Sharing Scheme [15]	Perform authentication	Safe enhanced	High time utilization
A2B2E [16]	Hidden access policy to share data by data owners	Access multiple data information	Lower computing complexity is a failed
A comprehensive and useful approach [17]	User authentication and cryptographic system	Authorized user protects data information	Fails to achieve higher data integrity
ASDS [18]	User revocation flexibility and data access management	Efficient data access	Did not attain lower storage overhead
An authority-verified, privacy-preserving CP-ABE method [19]	Privacy-preserving approach	Higher data security	Failed to improve data confidentiality
Asymmetric key technology [20]	Develops asymmetric key technology for file exchange	Improves data security	Time complexity issue occurred
Secure Authentication and Data Sharing in the Cloud [21]	Handle large data outsourcing, allocation	Sharing with higher security	Failed to enhance privacy
Rescue Chain [22]	Secure and efficient information-sharing for UAVs	Optimal allocation strategy to achieve better data allocation	Storage overhead was not minimized
Privacy-preserving access control method [23]	ABSC as well as the policy modernize method	Access data with higher security	Failed to reduce time complexity
Secure data-sharing system [24]	Patients’ sensitive information in electronic medical records is encrypted into cipher data.	Enhance deduplication efficiency	Storage complexity was not decreased
Proficient and secure data-sharing system [25]	Cloud user verification and lightweight operation	Perceptive information distributed via authorized access	Computational complexity was not reduced
Efficient, Provably Secure Data Selection Sharing [26]	Trust organization System	Time was reduced	Failed to enhance privacy
Secure and distributed IoT data storage approach [27]	Ultra-lightweight secret sharing algorithm	Delivery of original messages with minimum storage	Time complexity was not decreased
SLFG-DSS [28]	Resisting decryption key operation	Access data by verified users with higher security level	Storage overhead was not minimized

Privacy-preserving access control model [29]	Fully outsourced attribute-based signcryption	Access data by data owner with high security and less computation time	Enhanced confidentiality rate was not achieved
RPO-CFE-SMC [30]	AES chaotic fuzzy encryption and red panda optimization algorithm	Mobile device access data with minimum energy consumption	Efficient data communication is difficult
Efficient and secure data sharing scheme [31]	User authentication and sharing	The mobile user avoids unauthorized mobile data access	Higher complexity during data sharing
MECC [32]	Cryptosystem along with data transfer approaches	Access data by cloud server	The confidentiality rate was not efficient
Authentication and authorization scheme [33]	Mobile user authentication phase	Identifies mobile users to access data on different services	Failed to increase secrecy
CP-ABE [34]	Secure and efficient cloud data sharing	Shares, access and store mobile data with higher data security	Computational complexity was higher
MP-RAGBE [35]	Data encryption	Increases data transmission efficiency	Time consumption of secure data transmission was high

2.2. Problem Definition

Mobile cloud computing presents flexible on-demand data services to mobile users over the internet. Due to numerous cloud users in computing resources, data security is the most significant aspect of cloud data privacy. Here, mobile user authentication is mainly perceptive protection in CC from unauthorized user access in cloud services. Traditional methods are introduced to verify and validate cloud user identity. Cloud data is accessed using the user ID and password. However, conventional authentication data confirmation performance failed to enhance. In the cloud, User authentication and data security are significant problems. Hence, the above issues are overcome by proposing three different proposed techniques.

3. RANDOM SELF-GENERATIVE SCHNORR CERTIFICATELESS SIGNCRYPTION BASED SECURED DATA SHARING (RSGSCS-SDS) TECHNIQUE

A public-key cryptography technique called Random Self-Generative Schnorr Certificateless Signcryption (RSGSCS) combines digital signature and encryption procedures. Data security in mobile clouds is improved by a more effective method called Schnorr Signcryption. Encryption and digital signature verification enhance data secrecy, which is important for transferring data in the cloud between servers and mobile users. The digital signature generation and verification algorithm used by the proposed RSGSCS-SDS technique is based on public key encryption, which enhances confidentiality.

Figure 1 indicates the RSGSCS-SDS Technique architecture diagram, which aims to improve data secrecy. Three primary components make up the cloud-based architecture: a Mobile Cloud Server (MCS), a Data Owner (DO) for safe data sharing, and numerous mobile cloud users (referred to as "mcu_1, mcu_2, mcu_3,... mcu_n"). Initially, the Mobile Cloud Server receives data uploads from the Data Owner. The cloud owner confirms the identification of mobile CU before granting access to this data. Mobile CS sends encrypted information and a signature to the mobile user after successful authentication. After that, the mobile CU confirms the signature and extracts the original data by decrypting it.

The registration phase is performed to register user details on MCS. Cloud users log in to the server and register their information. For each registered mobile user, a key pair is generated by the mobile cloud server. Based on successful registrations with one-time passwords, keys are generated to perform data encrypt or decrypting. With the aid of generated keys, data confidentiality and integrity rates are increased by secure data access from unauthorized users in cloud servers. The generation of private keys is used in the cloud to evade illegal data. By applying the Policy Attribute Schnorr Certificateless Signcryption Process, signcryption is carried out to perform data encryption and digital signatures. Original information is encrypted to encrypted text, and a signature is generated. Signature verification and data decryption are carried out during unsigncryption. Via receiver's private key, authorized users are only allowed for decrypting data. For accessing secure data on the cloud, information is sent to the server to attain secure data transmission. Hence, the performance of secure access improved for RSGSCS-SDS with minimum overhead.

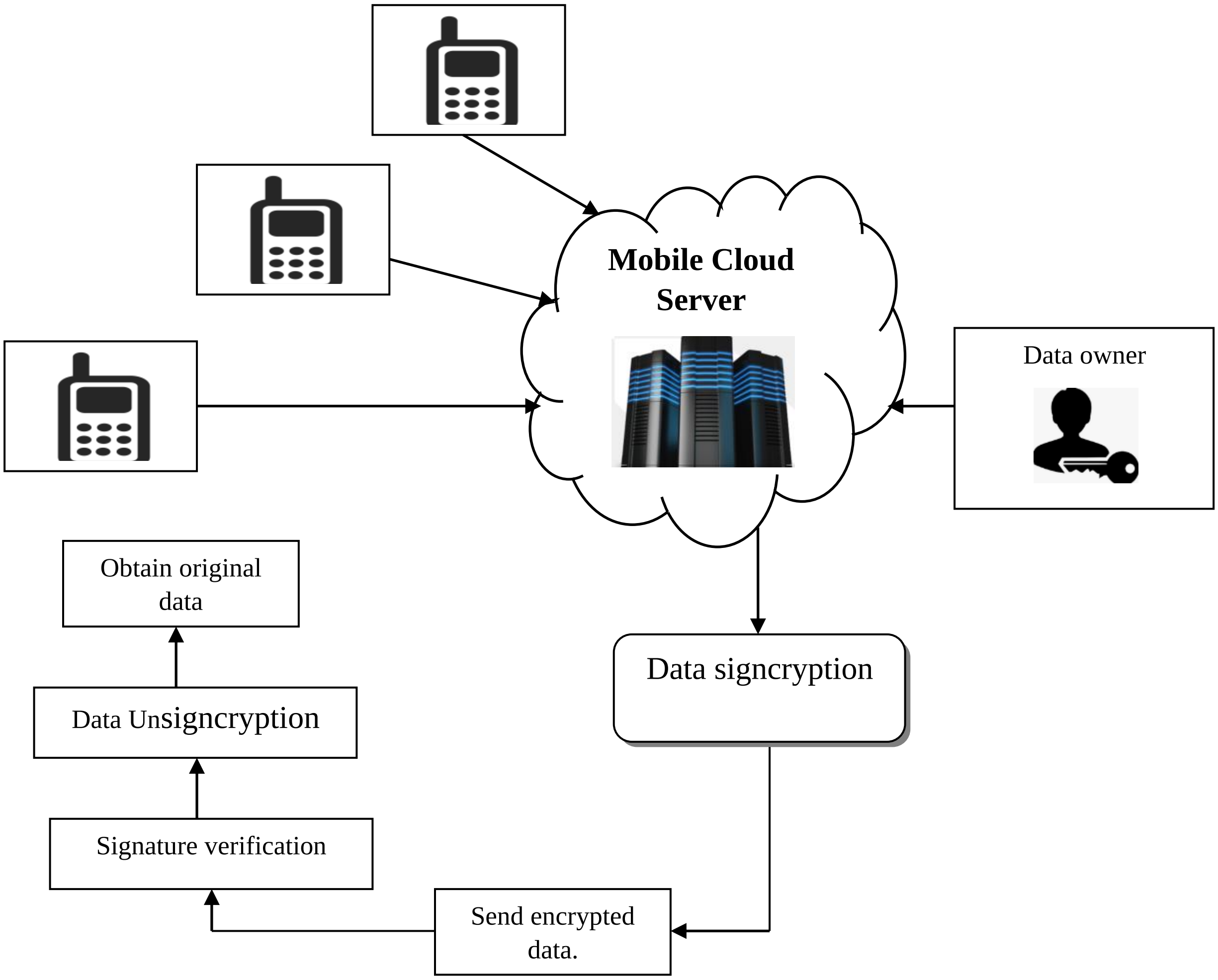

Figure 1 Architecture Diagram of RSGSCS-SDS

3.1. User Registration and Key Generation

RSGSCS-SDS approach consists of two main processes. User provides mobile cloud servers during the registration procedure with their information. After that, the user's data is accumulated in the server's database. As a user registers, the cloud service provider sends them an OTP (one-time password). There is a time limit for the user to enter this OTP. The OTP is only valid for a certain amount of time; therefore, if it is not entered within that window of time, the user will need to log in again and provide their information again. The mobile user receives a confirmation message from the cloud server showing successful registration as soon as the OTP is input correctly. The mobile cloud server generates a random self-generated public key and a random self-generated private key for each registered user following this registration process. Registration and key generation procedures for the suggested RSGSCS-SDS approach are shown in Figure 2.

$$mcu \xrightarrow{\boldsymbol{Details}} MCS \qquad (1)$$

As shown in equation (1), the mobile cloud user is represented by 'mcu' in (1), while the mobile cloud server is denoted by 'MCS'. The OTP is sent to the registered cellphone number by the mobile cloud server.

$$MCS \xrightarrow{\text{OTP}} mcu \qquad (2)$$

From equation (2), mobile CU needs to input OTP within the time range that mobile CS specifies. Mobile user receives a Successfully Registered Message (SRM) from the server after this is completed. User's data is now saved on server, which also creates the keys. For example, the Schnorr key generation technique produces a positive integer as the secret signature key (private key), 'PI'.

$$Pri_{Key} = PI \qquad (3)$$

In equation (3), private key is denoted by 'Pri$_{key}$', 'PI' is chosen in arbitrary. Public verification key can be acquired with private key generation.

$$Pub_{Key} = f(PI) \qquad (4)$$
$$f(PI) = PI + 1\,mod\,16 \qquad (5)$$

From equations (4) and (5), '$f(PI)$' represents the one-way function. 'Pub_{Key}' symbolizes the gathered public verification key from "PI" As the User_id, public verification keys are utilised. The registered user receives a user ID, which is regarded as the public key. Several policy attributes are generated by the mobile cloud server during the ID-generating procedure. In the context of the mobile cloud, the keys are given to registered mobile users. 'f(PI)' indicates the one-way function in equations (4) and (5), and 'Pub$_{key}$' is the public verification key that is generated from 'PI'. The User_id, which functions as a public key and is given to registered users, is created using these public verification keys. The mobile CS creates dissimilar policy attributes while the ID generation procedure is in progress. After that, the keys are given out to mobile users who have registered in the mobile cloud environment.

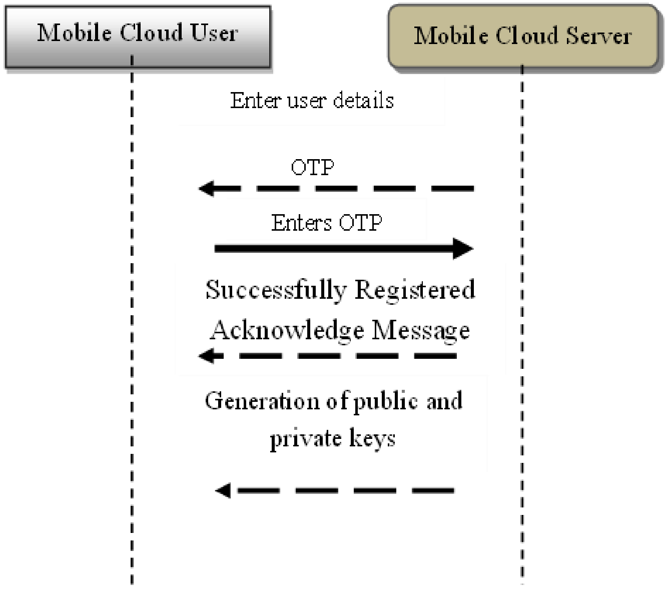
Mobile Cloud User
Mobile Cloud Server
Enter user details
OTP
Enters OTP
Successfully Registered Acknowledge Message
Generation of public and private keys

Figure 2 Diagrammatic Representations of Registration and Key Generation

3.2. Signcryption

Signcryption is used by the RSGSCS-SDS approach to facilitate effective data exchange with authorised organisations according to policy attributes. A public-key cryptosystem called signcryption combines data encryption and digital signatures. Encryption, as well as decryption, employs the receiver's public and private keys. Data encryption converts the original file or data into a different format that is only readable by those with the proper authorisation. A registered mobile user must log in to mobile CS to access resources. Subsequently, policy characteristics established during the registration process are verified by the mobile cloud server.

Figure 3 demonstrates Policy Attribute Schnorr Certificateless Signcryption Process. A registered mobile user enters their User_Id in the login window to access data on mobile CS. Next, mobile CS verifies that the User ID entered corresponds to the one that was registered and saved in the server's database. Policies that combine qualities determine the access permissions that are granted. 'If-then' rules are used in the Policy Attribute Schnorr Certificateless Signcryption scheme to achieve this. The user is considered authorised and given access if the ID they entered matches the one in the database. If not, access is refused since the user is deemed unauthorised. Following authentication, the server transmits encrypted information to the authorised user in the ciphertext. The architecture of the Signcryption Process is shown in Figure 4.

Let's assume the data is represented as $'d_1, d_2, d_3, .. d_n$ '. Ciphertext for data distributed through mobile CS is given as follows:

$$Cypher(d) \leftarrow Encryption\langle Pub_{Key}, d\rangle \qquad (6)$$

As shown in equation (6), Cipher (d) from (6) shows the ciphertext of data 'd'. With the recipient's public key (Pub_{Key}) Or user_Id, encryption is performed. Sender's private key used in digital signature. Legitimate digital signature provides evidence that the data was created by the identified sender (a mobile cloud server) and that no unauthorised parties have altered it. The Schnorr Certificateless Signcryption algorithm uses a private key to operate. The data $d = d_1, d_2, d_3, .. d_m$is (0, 1) is examined, and the formulation for the signature creation is as follows:

The ciphertext of the data 'd' is represented by Cipher (d) in (6). Through encryption, recipient's public key (Pub_{Key}) equivalent for user_Id. For establishing digital signature, sender's private key is then employed. This legitimate digital signature attests to the fact that the data was created by the mobile cloud server, which is the authenticated sender, and that no unauthorised parties have altered it. The Schnorr Certificateless Signcryption technique is employed using the private key. Assume that di$\in\{0,1\}$ and that the data are$d =$

$d_1, d_2, d_3, \ldots . d_m$. The following is a description of the process of creating a signature:

$$Signature_d = h(PI \| d) \qquad (7)$$

From equation (7), '$Signature_d$'denotes the signature of mobile cloud data 'd'.'$(PI\|)$' represents the concatenation. 'h' symbolizes the cryptographic hash function. 'PI' symbolizes the positive integer. After signature generation, the mobile cloud server transmits the ciphertext and signature.

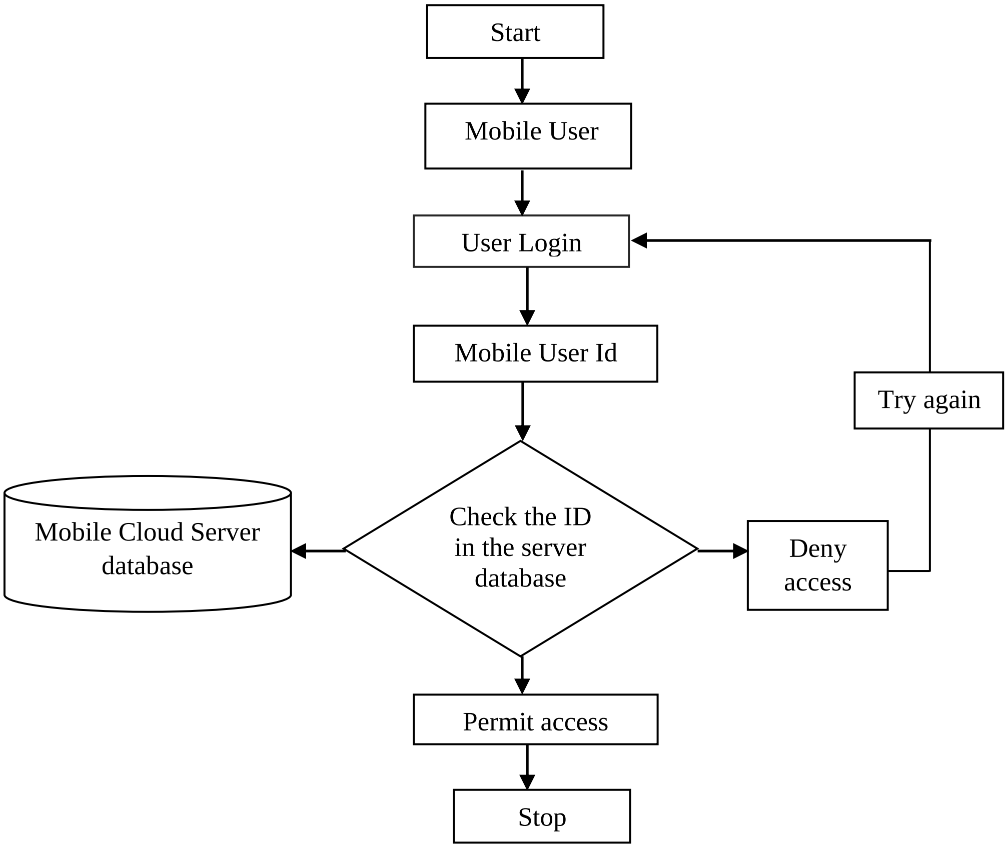

Figure 3 Policy Attribute Schnorr Certificateless Signcryption Process

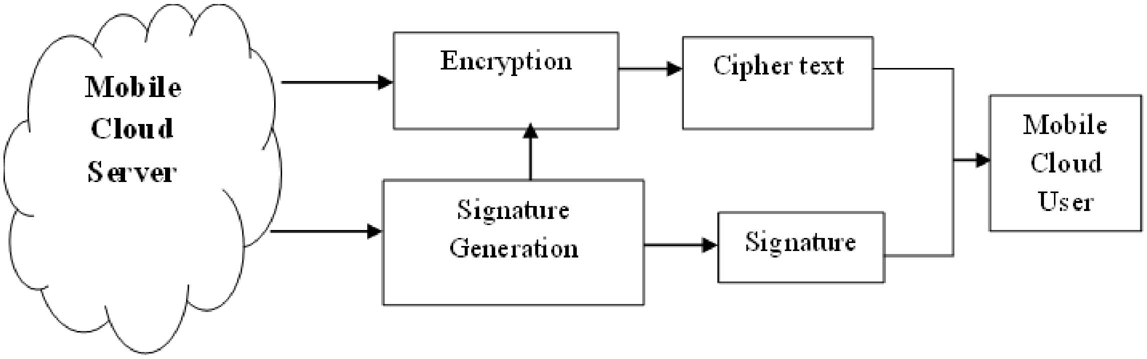

Figure 4 Signcryption Process

3.3. Unsigncryption

For recovering innovative information, RSGSCS-SDS approach does unsigncryption by signature verification as well as decryption. Schnorr Certificateless Signcryption is used to verify signatures.

Figure 5 depicts a block schematic of encryption and signature confirmation procedures that yield plain text. Signature is verified using public key. Formulation is as follows:

$$Signature_d'' = h\ (PI_v \| d) \qquad (8)$$

$$\begin{aligned} f\ (x) &= \\ if\ (Signature_d = Signature_d{}'')\ &;\ signature\ is\ valid \\ otherwise;\ &\quad signature\ is\ not\ valid \end{aligned} \tag{9}$$

From equations (8) and (9), signature created in receiver part is ‘$Signature_d{}''$. Signature one-way function is$f(x)$, ‘PI_v’ represents a positive integer, $and\ h$ is the cryptographic hash function. At last, ensure a created signature.$Signature_d{}''$ is confirmed through the public key ‘ Pub_{Key} ’. As both signatures are matched, it is legitimate as well as MCU decrypts ciphertext. Or else, signature is coordinated as well as unacceptable. Mobile cloud users do not decrypt ciphertext with higher information-sharing security. For getting original information, decryption employed via authorized user.

$Signature_d{}'$ denotes the signature produced at the receiving end, and $f(x)$ denotes the signature's one-way function, according to equations (8) and (9). The positive integer ‘PI_v’ is represented by h, the cryptographic hash function. Finally, duringPub_{Key}, $Signature_d{}''$, is checked and validated. MCU decrypts ciphertext once both signatures match, indicating that the transaction is legitimate. If not, the signature is invalid since it does not match. It not decrypted with MCU. Consequently, security information exchanged between two organisations enhanced. After decrypting the ciphertext, novel information attained by authorised user in equations (8) and (9), whereas one-way function of the signature is represented by f(x). ‘PI_v' is positive integer as well as ‘h’ is cryptographic hash function. Lastly, confirm that created signature, $Signature_d{}''$, is legitimate by using the public key, Pub_{Key}’. When the two signatures match, the mobile cloud user decrypts the ciphertext, indicating that it is genuine. If not, the signature does not match and is therefore invalid. The user of the mobile cloud does not crack ciphertext. As an outcome, the security of the information exchange between two organisations is improved. Once decrypts ciphertext, authorised user can access original information as,

$$d \leftarrow Decryption\ \langle Pri_{Key}, Cipher(d)\rangle \tag{10}$$

In equation (10), the original data is denoted by ‘d’, while the mobile user's private key is denoted by ‘ Pri_{Key} ’. After decryption, the original data is eventually recovered, and the output layer shows the outcome. Data from the mobile cloud server can be accessed by an authorised user thanks to the RSGSCS-SDS method. Consequently, data secrecy is improved by the RSGSCS-SDS approach.

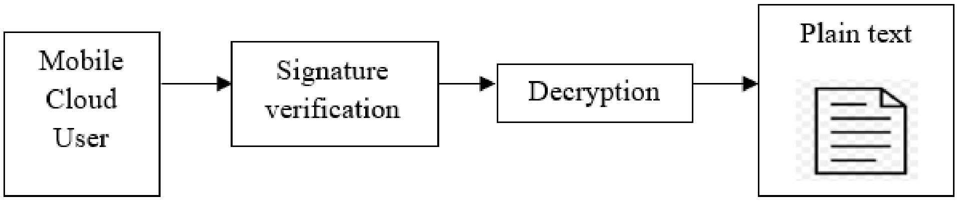

Figure 5 Unsigncryption Process

```
Input: Number of mobile cloud users$mcu_1, mcu_2, mcu_3, \dots . mcu_n$ , data $d_1, d_2, d_{3,\dots.} d_m$
Output: Increases data access security
Begin
Number of users $mcu_1, mcu_2, mcu_3, \dots . mcu_n$ taken as input at the input layer
// Registration and key generation
For every $mcu_i$
Register the details to the server.
Mobile cloud server sends $OTP$
User enters ‘$OTP$’ at specific time period$'t'$
If$mcu_i$ enter ‘$OTP$’
$MCS$ sends ‘$SRM$’ to $mcu_i$
End if
For each registered ‘$mcu_i$’
$MCS$ creates private and public key
End for
End for
// Signcryption
If MCU accesses data, then
Login to MCS with a valid '$Id$’
End if
If (Id matched with server database), then
An unauthorized user is defined as a mobile cloud user.
$MCS$ permits the access
else
MCU is unauthorized user
$MCS$ denied access
```

```
End if
Encrypt the data using a public key
Generate the digital signature
Send to the authorized user
// Unsigncryption
If (a signature is valid), then
Decrypt data employs private key
Achieve original data
End if
End
```

Algorithm 1 Random Self-Generative Schnorr Certificateless Signcryption-Based Secured Data Sharing

Random Self-Generative Schnorr Certificateless Signcryption portrayed in algorithm 1. Initially, the number of mobile users with various data is considered as input. For each mobile user, a registration and key generation process is performed. At server, user listed their information and keys pair produced. Based on generated keys, Random Self-Generative Schnorr Certificateless Signcryption is performed. Here, data encryption and decryption are performed. Mobile user is logged at signcryption with ID to access data. While ID match as well as stored to identify authorized user. If the ID is not matched, then it is considered an unauthorised user. Thus, authorised users are allowed to access data through a mobile cloud server. 'Pri_{Key}' achieves original content, and user's end validates signature with the public key. Using generated Pri_{Key}, unsigncryption is presented to decrypt original data. In the end, the mobile cloud environment's safe data access is improved by the RSGSCS-SDS approach.

4. EXPERIMENTAL EVALUATION

The designed RSGSCS-SDS approach is experimentally evaluated using Java and the CloudSim network simulator.

Amazon Access sample database employed to enable safe data sharing and access in mobile cloud context. It collected as of UCI machine learning repository https://archive.ics.uci.edu/ml/datasets/Amazon+Access+Samples. The dataset includes a large amount of data on cloud users, with varied registered and authorization data. It includes users as well as allocated access. In file, four attributes categories included in below table 2. Dataset describes users who can possibly have entrée and are labelled '1'. Otherwise, it will be 0. It includes examples of access given inside a corporation that has been anonymised. The dataset's main objective is to control authorised users' access by using past data. The details of the dataset are provided in the table 2 given below.

Table 2 Amazon Access Samples Dataset

S. No	Attributes	Description
1	PERSON_ATTRIBUTE	User who was given access
	PERSON_ID	ID of the user
	PERSON_MGR_ID	ID of the user's manager
	PERSON_ROLLUP_1	User grouping ID
	PERSON_ROLLUP_2	User grouping ID
	PERSON_ROLLUP_3	User grouping ID
	PERSON_DEPTNAME	Department description ID
	PERSON_LOCATION	Region ID
	PERSON_BUSINESS_TITLE	Title ID
	PERSON_BUSINESS_TITLE_DETAIL	Description ID
	PERSON_JOB_CODE	Job code ID
	PERSON_COMPANY	Company ID
	PERSON_JOB_FAMILY	Job family ID
2	RESOURCE_ID	Resources user can possibly have access to 1- access 2- no access
3	GROUP_ID	Groups that user can possibly have access to 1- access 2- no access
4	SYSTEM_SUPPORT_ID	System that a user can possibly be supporting 1- access 2- no access

The simulation analysis of RSGSCS-SDS technique is conducted by comparing existing algorithms, namely MA-CPABE-ECC [1], Lightweight Data Sharing Scheme (LDSS) [2], privacy-preserving access control model [29] and RPO-CFE-SMC [30]. The results are experimentally evaluated based on metrics such as data confidentiality, computational time, storage overhead, data integrity rate and delay.

4.1. Impact of Storage Overhead

It is memory used during key generation process for obtaining enhanced security. This is how the storage overhead is computed:

$$Storage_{Over} = Numberofdata * Memory[privatekey + publickey] \qquad (11)$$

From equation (11), ‘$Storage_{Over}$’ denotes the storage overhead. Kilobytes (KB) are used to express the storage overhead measurement. Table 3 displays the storage overhead for three different approaches according to the quantity of data points (25–250). In comparison to the other two ways, the suggested RSGSCS-SDS strategy has a reduced storage overhead, according to the results. For instance, the proposed RSGSCS-SDS technique results in a storage overhead of 300KB when the number of data points for sharing information as of mobile CS is 25 while existing [1], [2], [29] and [30] result in storage overheads of 385KB, 354KB, 342KB and 325 KB. Remaining results are also collected and shown in following graphical depiction.

Table 3 Tabulation of Storage Overhead

Number of data	Storage overhead (KB)				
	MA-CPABE-ECC	LDSS	Privacy-preserving access control model	RPO-CFE-SMC	RSGSCS-SDS Technique
25	385	354	342	325	300
50	398	365	353	341	315
75	415	379	362	350	328
100	426	391	370	361	340
125	437	412	384	376	355
150	452	435	405	396	380
175	473	458	439	418	400
200	496	479	468	443	421
225	507	495	476	457	445
250	519	512	495	465	459

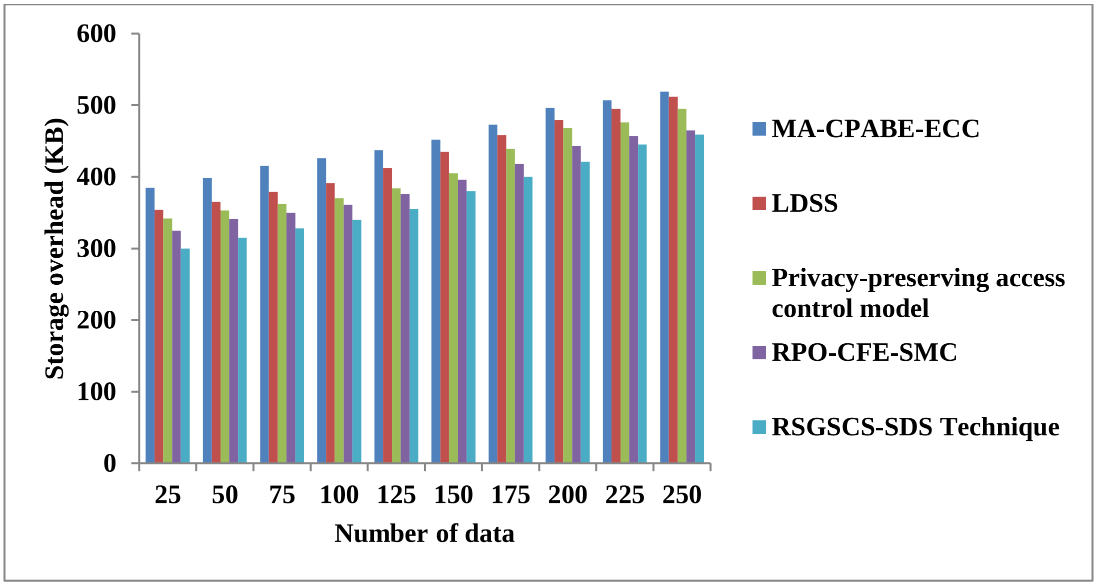

Figure 6 Measurement of Storage Overhead

The comparative findings of storage overhead with respect to the quantity of data points are exposed in Figure 6. The experiment is conducted using RSGSCS-SDS Technique with MA-CPABE-ECC [1], LDSS [2], privacy-preserving access control model [29] and RPO-CFE-SMC [30], respectively. The findings show that, in comparison to the current techniques, the suggested RSGSCS-SDS Technique achieves a lower storage overhead. The suggested technique's usage of Random Self-Generative Schnorr Certificateless Signcryption is what causes this reduction. For ensuring safe information sharing, mobile cloud user ID is checked when they ask for information about entering the cloud. Via information proprietor authentication is carried out. Owner asks CS to provide data when the mobile user ID on file matches the one they registered with. The suggested RSGSCS-SDS Technique successfully differentiates between authorised and unauthorised mobile users in this way. The comparison results show that, on average, the suggested RSGSCS-SDS

Technique reduces storage overhead by 17%, 13%, 9% and 5% than [1], [2], [29] and [30].

4.2. Computation Time ($Comp_{Time}$)

Time consumed for carry out safe information sharing between cloud servers and users referred as $Comp_{Time}$. Itis given as,

$$Comp_{Time} = [Numberofdata * Timeconsumedtoshareonedata] \quad (12)$$

From equation (12), '$Comp_{Time}$'denotes computation time. It calculated in milliseconds (ms).

Table 4 illustrates computation time. In table 4, proposed RSGSCS-SDS of time is less than that of the other two methods. Number of data 25 considered. By applying the proposed RSGSCS-SDS technique, the computation time obtained is 25ms, while the existing MA-CPABE-ECC [1], LDSS [2], privacy-preserving access control model [29] and RPO-CFE-SMC [30] consumed as 41ms, 34ms, 31ms and 28ms. Likewise, the remaining result outcomes are attained and depicted in the graphical representation Figure 7.

Table 4 Tabulation of $Comp_{Time}$

Number of data	Computation time (ms)				
	MA-CPABE-ECC	LDSS	Privacy-preserving access control model	RPO-CFE-SMC	RSGSCS-SDS Technique
25	41	34	31	28	25
50	43	37	34	31	27
75	45	39	36	34	29
100	47	41	39	35	30
125	50	43	40	38	31
150	52	45	42	40	33
175	55	47	45	42	34
200	57	49	48	44	37
225	59	51	49	46	39
250	61	53	51	49	41

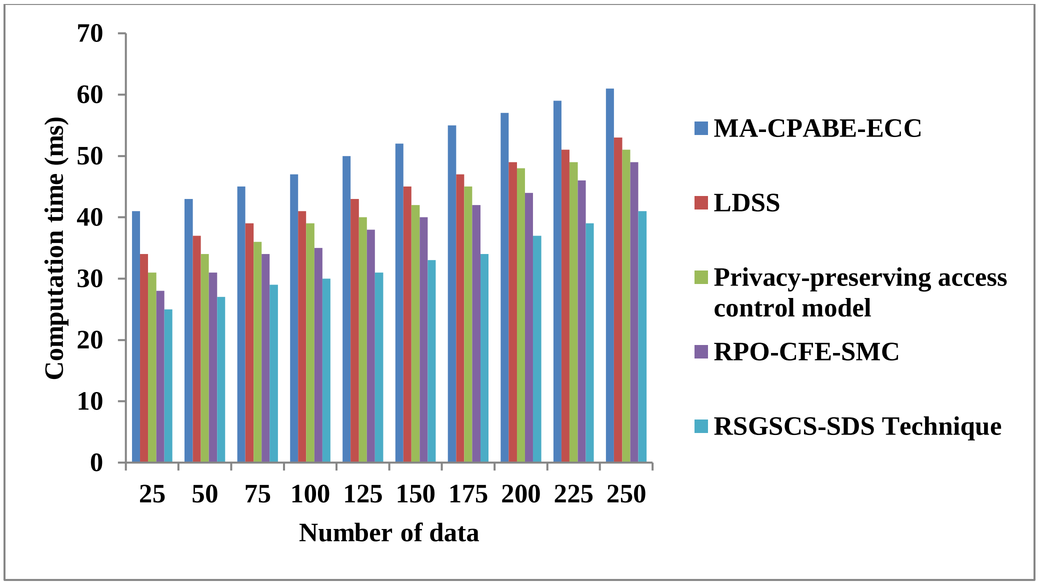

Figure 7 Measurement of $Comp_{Time}$

The comparison of the $Comp_{Time}$ results with a number of data are explained in Figure 7. RSGSCS-SDS reduces computing time than other approaches. This is due to the fact that random self-generative Schnorr certificateless signcryption was utilised in the proposed RSGSCS-SDS Technique. In order to share safe data, mobile cloud users must authenticate themselves when attempting to enter information in a mobile cloud environment. Owner handles authentication using user's identification. Users can request efficient data sharing from the cloud server when their mobile user ID precisely matches the ID that was stored at the time of registration. This makes the suggested RSGSCS-SDS Technique more accurate and time-efficient in identifying authorised or unauthorised mobile users. Comparing the suggested RSGSCS-SDS Technique to MA-CPABE-ECC [1], LDSS [2], privacy-preserving access control model [29] and RPO-CFE-SMC [30] reduces time consumption by 36%, 26%, 21% and 16%.

4.3. Data Confidentiality Rate

The ability to secure information through unauthorised manipulation on a server located in the cloud is referred to as the data confidentially ratio. The ratio of the total amount of cloud data to the quantity of information accessed by authorised users is used to calculate data confidentiality. The formula for the data secrecy rate is,

$$DataCon_{Rate} = \left(\frac{Numberofdataaccessedbytheauthorizeduser's}{Totalnumberofclouddata}\right) * 100 \qquad (13)$$

In equation (13), the data confidentially rate is represented by '$Data\ Con_{Rate}$'from (13). The rate of data confidentiality is calculated as a percentage (%).

The data confidentially rates for the three approaches are shown in Table 5 for a range of data amounts from 25 to 250. RSGSCS-SDS consistently produces a greater data secrecy rate, according to the results over four methods. For instance, the RSGSCS-SDS Method attains an 89% when 25 data points are shared from the mobile cloud server. In existing [1], [2], [29] and [30] attains 75%, 78%, 83% and 86% of confidentially rate. A graphical depiction displaying data confidentially rates are also provided and include the remaining nine outcomes.

Table 5 Tabulation of $Data\ Con_{Rate}$

Number of data	Data confidentially rate (%)				
	MA-CPABE-ECC	LDSS	Privacy-preserving access control model	RPO-CFE-SMC	RSGSCS-SDS Technique
25	75	78	83	86	89
50	77	79	85	87	90
75	78	81	86	89	91
100	81	83	88	90	92
125	82	85	89	91	93
150	84	86	91	93	94
175	85	87	89	91	95
200	88	90	92	93	94
225	89	91	91	92	96
250	90	93	93	94	97

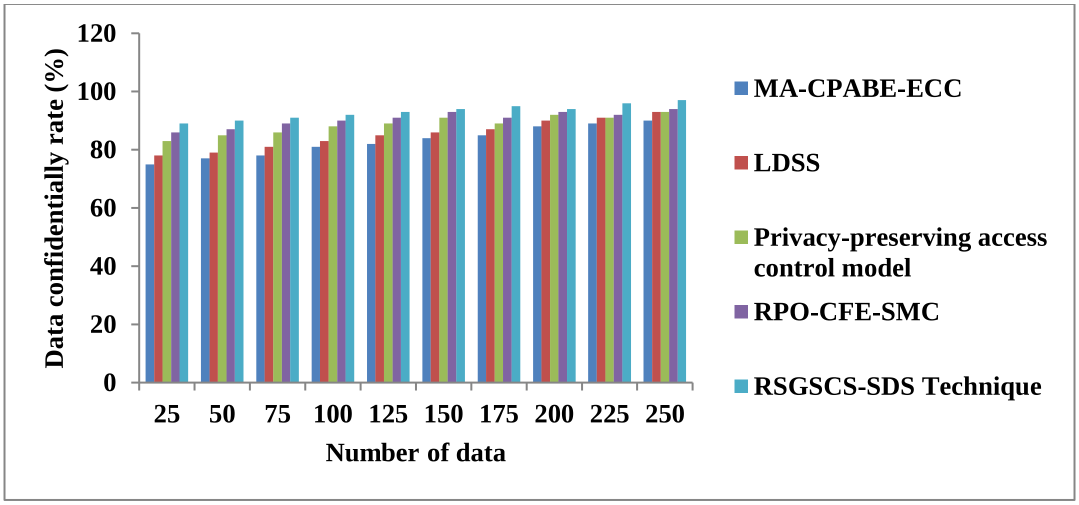

Figure 8 Measurement of $Data\ Con_{Rate}$

A comparison of $Data\,Con_{Rate}$ based amount of data is shown in Figure 8. The confidential rate using RSGSCS-SDS Technique, MA-CPABE-ECC [1], LDSS [2], privacy-preserving access control model [29] and RPO-CFE-SMC [30] are depicted in the above figure. Comparing the suggested RSGSCS-SDS Technique to the current approaches, the results show that it produces a greater data confidentially rate. The use of the random self-generative Schnorr certificateless Signcryption method is credited with this improvement. In order to ensure secure data sharing, mobile cloud users authenticate themselves when accessing data; data owners carry out authentication depending on the user's identification. When mobile user's ID matches ID, effective distributed is requested that was saved at the time of registration. This procedure keeps the data secrecy rate high while allowing the RSGSCS-SDS Technique to reliably differentiate between authorised and unauthorised mobile users. Comparatively speaking, the RSGSCS-SDS Technique outperforms better results when compared with MA-CPABE-ECC [1], LDSS [2], privacy-preserving access control model [29] and RPO-CFE-SMC [30] in terms of enhanced data confidentiality rate by 13%, 9%, 5% and 3%.

4.4. Data Integrity Rate

It determined as data unaltered or unchanged via some unlawful users.

$$Rate_{DI} = \left[\frac{Numberofdatanotaltered}{n}\right] * 100 \qquad (14)$$

From equation (14), $Rate_{DI}$ indicates a data integrity rate, number of data 'n' denoted as unit of percentage (%).

Table 6 Tabulation of Data Integrity Rate

Number of data	Data Integrity Rate (%)				
	MA-CPABE-ECC	LDSS	Privacy-preserving access control model	RPO-CFE-SMC	RSGSCS-SDS Technique
25	74	76	82	85	88
50	75	77	84	86	89
75	76	79	85	87	90
100	80	81	86	89	92
125	81	84	87	90	92
150	82	85	90	92	93
175	84	86	88	90	95
200	86	89	91	92	93
225	88	90	90	90	95
250	89	92	92	93	96

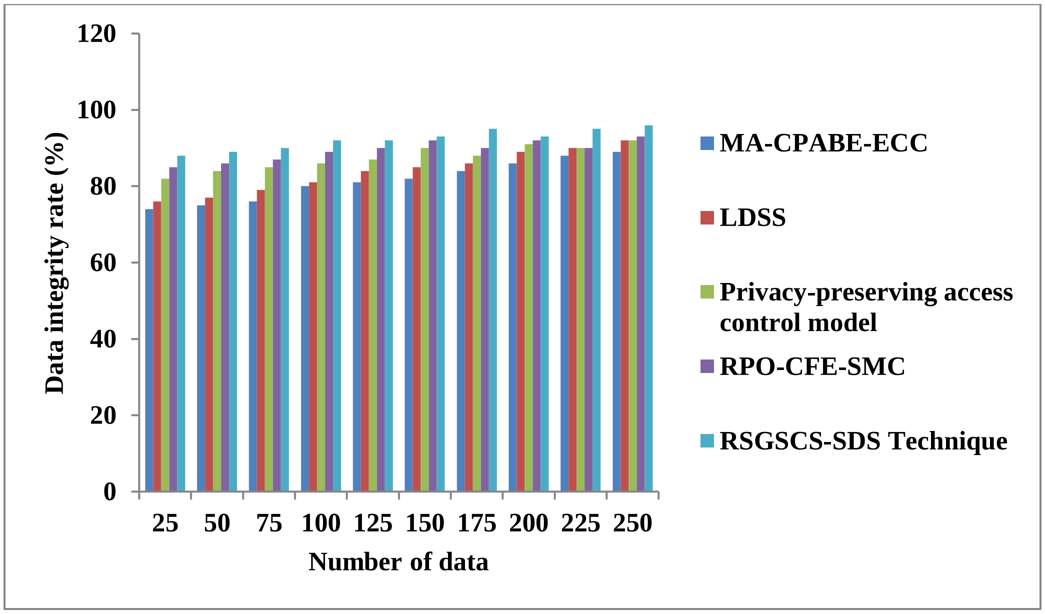

Figure 9 Measurement Data Integrity Rate

Table 6 and Figure 9 illustrate the performance investigation of data integrity rates using the proposed RSGSCS-SDS technique, existing MA-CPABE-ECC [1], LDSS [2], privacy-preserving access control model [29] and RPO-CFE-SMC [30]. Contrary to existing methods, integrity was enhanced by RSGSCS-SDS. However, in experiments conducted with 25 data, the integrity rate was observed to be 88% using RSGSCS-SDS Technique, and it was 74%, 76%, 82% and 85% using methods [1], [2], [29] and [30] respectively. Random Self-Generative Schnorr Certificateless Signcryption is used for session key pair generation in encryption as well as decryption. Consequently, unauthorized users were unable to alter or modify any data, enhancing data integrity during communication between the cloud user and server. Therefore, RSGSCS-SDS of integrity enhanced by 14%, 10%, 6% and 3% when compared to existing [1], [2], [29] and [30].

4.5. Impact of Delay

Its time consumed while sharing data packets between mobile users and data owners. It is estimated based on difference between actual arrival time and expected arrival time. The delay is expressed in milliseconds (ms) and formulated as given below.

$$Delay = Time_A - Time_{Ex} \tag{15}$$

From equation (15), delay is determined. Here, '$Time_A$' denotes the actual arrival time of data and '$Time_{Ex}$' specifies the expected arrival time to share data on mobile devices.

Table 7 Tabulation of Delay

Number of data	Delay (ms)				
	MA-CPABE-ECC	LDSS	Privacy-preserving access control model	RPO-CFE-SMC	RSGSCS-SDS Technique
25	15.8	16.8	14.3	12.5	11.2
50	17.3	19.3	15.4	14.8	13.1
75	18.3	20.6	16.7	15.1	14
100	20.1	22.7	18.3	16.7	15.3
125	23	25.3	21.5	18.4	17.2
150	26.2	28.1	24	22.3	20.7
175	30.1	32.4	27.3	25.4	22.3
200	32.5	34.2	29.1	27.5	24.4
225	34.6	37.2	31.1	29.4	26.2
250	36	38	33.4	31.2	28.5

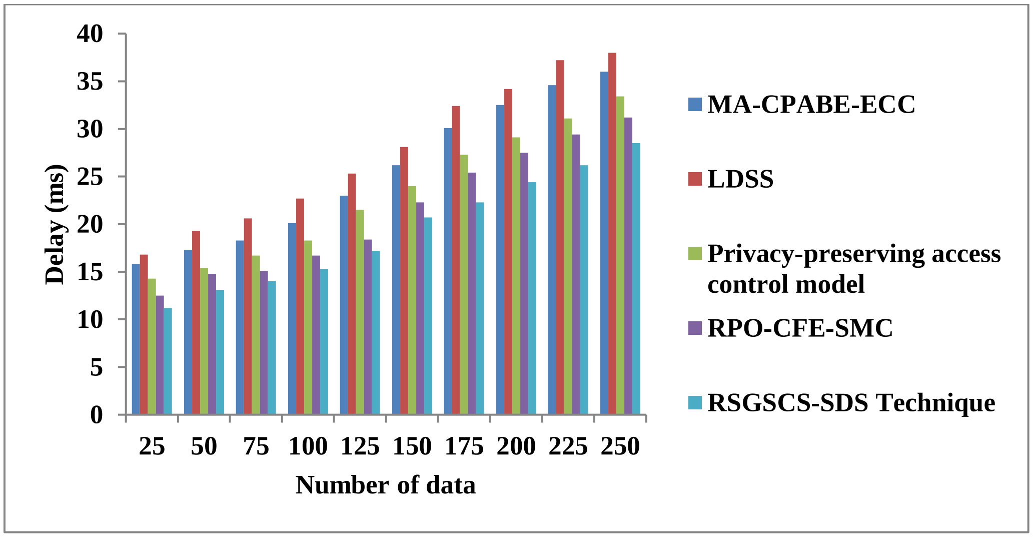

Figure10 Measurement Delay

Table 7 as well as Figure 10, illustrate performance outcomes of delay. RSGSCS-SDS delay is lower than that of the other methods [1], [2], [29] and [30]. For instance, when considering 25 data, the delay time consumption for secure data sharing using the RSGSCS-SDS Technique was found to be 11.2 ms, and conventional methods [1], [2], [29], and [30] were obtained as 15.8ms, 16.8ms, 14.3ms and 12.5ms, respectively. Different performance results were observed for each method with varying numbers of data. Through this performance analysis, it was determined that the delay using the RSGSCS-SDS Technique decreased by 24%, 30 %, 17% and 9% when compared to the MA-CPABE-ECC [1], LDSS [2], privacy-preserving access control model [29] and RPO-CFE-SMC [30]. This reduction is achieved by the application of the Random Self-Generative Schnorr Certificateless Signcryption, which enhances secure data encryption. The Signcryption accurately identifies authorized or unauthorized users before access, resulting in minimum time consumption.

5. CONCLUSION

For preserving confidentiality and attaining effective cryptographic solutions, data security and privacy are major concerns for mobile users. In this paper, an efficient and secure RSGSCS-SDS technique is developed to achieve secure sharing of a large amount of data in mobile devices that reduce memory consumption in a cloud computing environment. Registration executed in RSGSCS-SDS. Every MCU, key pair generated by key generation. Random Self-Generative Schnorr Certificateless Signcryption is carried out for secure data access. Encryption carries out by lesser space complexity. Authorized or unauthorized user determined via ID to enhance privacy. At last, information received via authorized person. At receiver end, data decryption is presented to obtain original information by verifying the digital signature. Implantation as well as results made in proposed technique. Compared to existing works, RSGSCS-SDS offers better in confidentiality as well as minimizes time and space complexity.

REFERENCES

[1] G. K. Sandhia and S. V. K. Raja, “Secure sharing of data in cloud using MA-CPABE with elliptic curve cryptography”, Journal of Ambient Intelligence and Humanized Computing, Springer, 2021, Pages 1-15Volume 13, pages 3893–3902.

[2] Ruixuan Li, Chenglin Shen, Heng He, XiwuGu, Zhiyong Xu and Cheng-Zhong Xu “A Lightweight Secure Data Sharing Scheme for Mobile Cloud Computing”, IEEE Transactions on Cloud Computing, Volume 6, Issue 2, April-June 2018, Pages 344 – 357.

[3] Pandi Vijayakumar, S. Milton Ganesh, Lazarus Jegatha Deborah, SK Hafizul Islam, Mohammad Mehedi Hassan, Abdulahmeed Alelaiwi and Giancarlo Fortino, “MGPV: A novel and efficient scheme for secure data sharing among mobile users in the public cloud”, Future Generation Computer Systems, Elsevier, Volume 95, June 2019, Pages 560-569

[4] Chaitanya Singh, Deepika Chauhan, Sushama A. Deshmukh, Swati Sudhakar Vishnu and Ranjan Walia,“Medi-Block record: Secure data sharing using block chain technology”, Informatics in Medicine, Elsevier, Volume 24, 2021, Pages 1-15.

[5] Sreeja Cherillath Sukumaran and Mohammed Misbahuddin, “PCR and Bio-signature for data confidentiality and integrity in mobile cloud computing”, Journal of King Saud University - Computer and Information Sciences, Elsevier, Volume 33, Issue 4, May 2021, Pages 426-435.

[6] Alaa Omran Almagrabi and A. K. Bashir, “A Classification-based Privacy-Preserving Decision-Making for Secure Data Sharing in Internet of Things Assisted Applications”, Digital Communications and Networks, Elsevier, Volume 8, Issue 4, August 2022, Pages 436-445.

[7] Loai A. Tawalbeh and Gokay Saldamli, “Reconsidering big data security and privacy in cloud and mobile cloud systems”, Journal of King Saud University-Computer and Information Sciences, Elsevier, Volume 33, Issue 7, September 2021, Pages 810-819.

[8] Qinyang Miao, Hui Lin, Xiaoding Wang, Mohammad Mehedi Hassan, “Federated deep reinforcement learning based secure data sharing for Internet of Things”, Computer Networks, Elsevier, Volume 197, October 2021, Pages 1-15.

[9] Qian Xu, Chengxiang Tan, Zhijie Fan, Wenye Zhu, Ya Xiao and Fujia Cheng, “Secure Multi-Authority Data Access Control Scheme in Cloud Storage System Based on Attribute-Based Signcryption”, IEEE Access, Volume 6, 2018, Pages 34051 – 34074.

[10] Manreet Sohal and Sandeep Sharma, “BDNA-A DNA inspired symmetric key cryptographic technique to secure cloud computing”, Journal of King Saud University - Computer and Information Sciences, Elsevier, Volume 34, Issue 1, January 2022, Pages 1417-1425.

[11] Suyel Namasudra, “An improved attribute-based encryption technique towards the data security in cloud computing”, Concurrency Computation Practice Experience, Wiley, Volume 31, Issue 3, 2019, Pages 1-15.

[12] K. Rajesh Rao, Indranil Ghosh Ray, Waqar Asif, Ashalatha Nayak and Muttukrishnan Rajarajan, “R-PEKS: RBAC Enabled PEKS for Secure Access of Cloud Data”, IEEE Access, Volume 7, 2019, Pages 133274 – 133289.

[13] Sana Belguith, Nesrine Kaaniche, Mohammad Hammoudeh, Tooska Dargahi, “PROUD: Verifiable Privacy-preserving Outsourced Attribute Based SignCryption supporting access policy Update for cloud assisted IoT applications”, Future Generation Computer Systems, Elsevier, Volume 111, October 2020, Pages 899-918.

[14] Shilpi Harnal and R.K. Chauhan, “Hybrid Cryptography based E2EE for Integrity & Confidentiality in Multimedia Cloud Computing”, International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume 8 Issue 10, 2019, Pages 918-924.

[15] Xiong Li, Saru Kumari, Jian Shen, Fan Wu, Caisen Chen, SK Hafizul Islam, “Secure Data Access and Sharing Scheme for Cloud Storage”, Wireless Personal Communications, Springer, Volume 96, Issue 4, 2017, Pages 5295–5314.

[16] Hu Xiong, Hao Zhang and Jianfei Sun “Attribute-Based Privacy-Preserving Data Sharing for Dynamic Groups in Cloud Computing”, IEEE Systems Journal, Volume 13, Issue 3, September 2019, Pages 2739 – 2750.

[17] Prerna Agarwal, Dr.S.P.Singh and Pranav Shrivastava, “A Safe and Resilient Cryptographic System for Dynamic Cloud Groups with Secure Data Sharing and Efficient User Revocation”, Turkish Journal of Computer and Mathematics Education, Volume 12, Issue 3, 2021, Pages 5164-5175.

[18] Nabeil Eltayieb, Ping Wang, Alzubair Hassan, Rashad Elhabob and Fagen Li “ASDS: Attribute-based secure data sharing scheme for reliable cloud environment”, Volume 2, Issue 2, March/April 2019, Pages 1-11.

[19] Leyou Zhang, Yilei Cui and Yi Mu, “Improving Security and Privacy Attribute Based Data Sharing in Cloud Computing”, IEEE Systems Journal, Volume 14, Issue 1, March 2020, Pages 387 – 397.

[20] K. V. Pradeep, V. Vijayakumar, and V. Subramaniyaswamy, “An Efficient Framework for Sharing a File in a Secure Manner Using

Asymmetric Key Distribution Management in Cloud Environment", Journal of Computer Networks and Communications, Hindawi Publishing Corporation, Volume 2019, 2019, Pages 1-15.

[21] Uma Narayanan, Varghese Paul Shelbi Joseph, "A novel system architecture for secure authentication and data sharing in cloud enabled Big Data Environment" Journal of King Saud University – Computer and Information Sciences Volume 34, Issue 6, June 2022, Pages 3121-3135.

[22] Yuntao Wang, Zhou Su, Qichao Xu, Ruidong Li, Tom H. Luan, and Pinghui Wang, "A Secure and Intelligent Data Sharing Scheme for UAV-Assisted Disaster Rescue"Volume: 31, Issue: 6, December 2023,Pages 2422 – 2438.

[23] Pattavee Sanchol and Somchart Fugkeaw , "A Fully Outsourced Attribute-Based Signcryption Scheme Supporting Privacy-Preserving Policy Update in Mobile Cloud Computing"Volume: 11,Pages 145915 – 145930.

[24] Zhiqiang Wang, Wenjing Gao, Ming Yang, Rong Hao, "Enabling Secure Data sharing with data deduplication and sensitive information hiding in cloud-assisted Electronic Medical Systems" Volume 26, 2023, Pages 3839–3854.

[25] Xiuqing Lu, Zhenkuan Panand Hequn Xian, "An efficient and secure data sharing scheme for mobile devices in cloud computing", Issue :9, Pages 1-13,

[26] S. Velmurugan, M. Prakash, S. Neelakandan and Arun Radhakrishnan, "Provably secure data selective sharing scheme with cloud-based decentralized trust management systems" Volume 13, Issue :1, Pages 1-20.

[27] Na Wang, Junsong Fu, Shancheng Zhang, Zheng Zhang, Jiawen Qiao, Jianwei Liu, and Bharat K. Bhargava, Life Fellow, "Secure and Distributed IoT Data Storage in Clouds Based on Secret Sharing and Collaborative Blockchain" Volume: 31, Issue: 4, 2023, Pages 1550 – 1565.

[28] Haifeng Li, Caihui Lan, Xingbing Fu, Caifen Wang, Fagen Li and He Guo, "A Secure and Lightweight Fine-Grained Data Sharing Scheme for Mobile Cloud Computing", Volume 20, Issue :17, Pages 1-17.

[29] Pattavee Sanchol and Somchart Fugkeaw, "A Fully Outsourced Attribute-Based Signcryption Scheme Supporting Privacy-Preserving Policy Update in Mobile Cloud Computing", IEEE Access, Volume: 11, December 2023, Page(s): 145915 – 145930, DOI: 10.1109/ACCESS.2023.3341095.

[30] Vishal Garg, "Enhancing mobile data security using red panda optimized approach with chaotic fuzzy encryption in mobile cloud computing", Concurrency and Computation Practice and Experience, Volume 36, Issue 1, July 2024, DOI:10.1002/cpe.8243.

[31] Xiuqing Lu, Zhenkuan Pan and Hequn Xian, "An efficient and secure data sharing scheme for mobile devices in cloud computing", Journal of Cloud Computing volume 9, Article number: 60, 2020.

[32] Dilip Venkata Kumar Vengala, D. Kavitha and A. P. Siva Kumar, "Three factor authentication system with modified ECC based secured data transfer: untrusted cloud environment", Complex & Intelligent Systems, Springer 2023, Volume 9, pages: 2915–2928, https://doi.org/10.1007/s40747-021-00305-0.

[33] Linsheng Yu, Mingxing He, Hongbin Liang, Ling Xiong and Yang Liu, "A Blockchain-Based Authentication and Authorization Scheme for Distributed Mobile Cloud Computing Services", Sensors 2023, Volume 23, Issue 3, 1264; https://doi.org/10.3390/s23031264.

[34] Madireddy Swetha and M. Latha, "Security on mobile cloud computing using cipher text policy and attribute based encryption scheme", Materials Today: Proceedings, Volume 80, Part 3, 2023, Pages 3059-3063.

[35] Jie Cui, Bei Li, Hong Zhong, Yan Xu, Lu Liu, "Achieving Revocable Attribute Group-Based Encryption for Mobile Cloud Data: A Multi-Proxy Assisted Approach" IEEE Transactions on Dependable and Secure Computing, Volume: 20, Issue: 4, 01 July-Aug. 2023, Page(s): 2988 – 3001, DOI: 10.1109/TDSC.2022.3204549.

Authors

Sowmya V L is currently pursuing a full-time PhD in the Department of Computer Science and Engineering at the BMS Institute of Technology & Management. She holds a Bachelor of Engineering (B.E.) and a Master of Technology (M.Tech.) in Computer Science. With five years of teaching experience and three years of dedicated research experience, her academic contributions include several book chapters published by Springer and a few journal publications. Sowmya's research interests are primarily focused on cloud computing and radiogenomics, where she has made significant contributions through her published works.

Dr Shankar has a PhD in the area of Sentiment Analysis via Machine Learning, has published more than 12 papers, and authored a book on "Understanding Artificial Intelligence," which is available on platforms like Amazon and Flipkart. With over 15 years of teaching experience complemented by a year in the industry, he brings a diverse skill set to the table. His tenure as an RPA Consultant for Ecocash in Zimbabwe honed his ability to adapt technology to meet real-world challenges. He has successfully translated complex concepts into easily accessible learning materials through modern tools. Additionally, his role as a corporate trainer for leading organizations such as WIPRO, CISCO, MRIU, and UPES reflects his proficiency in imparting knowledge to diverse audiences. His experience as a web developer for the BMS Institute of Technology & Management website further underscores his versatility in both technical and educational domains.

Dr. Anitha Premkumar is currently working as an Assistant Professor-Senior scale in the Department of Information Technology, Manipal Institute of Technology, Bengaluru, MAHE. She earned her PhD from Vellore Institute of Technology, Vellore, Tamil Nadu, in the year 2024. She also completed her M.Tech (2005) in CSE at the same university. She graduated with a B.E(2001) in Computer Science& Engineering from Madras University in Tamil Nadu. She presented her studies in conferences and international journals. Blockchain, machine learning, and cloud computing are among her areas of interest in the study.

Dr B. Umi Salma received an MCA from the University of VTU Belgaum, Karnataka, India, in 2007, an M. Phil in 2015, and a PhD in Computer Science & Engineering from Bharathiar University, India, in 2019. She is currently a lecturer at the College of Engineering and Computer Science at Jazan University, Kingdom of Saudi Arabia. Her research interests are in Information Security, Cloud Computing, Artificial intelligence, and IoT.

Dr Rajesh Natarajan completed his PhD in Computer Science from Bharathiar University, Master of Computer Application from Thiruvalluvar University and BSc in Computer Science from Madras University. He works as a Lecturer at the University of Technology and Applied Sciences-Shinas, Sultanate of Oman. His founder and chief editor for the International Journal of Data Informatics and Intelligent Computing. His research interests include Data Mining, Machine Learning, Big Data Analytics, Blockchain Technology, and Data Privacy and Security. He has presented articles at national and international conferences and has published articles in reputed indexed journals like WoS and SCOPUS.

RESEARCH ARTICLE

How to cite this article:

Sowmya V L, Shankar R, Anitha Premkumar, Umi Salma Basha, Rajesh Natarajan, “Enhancing Secured Data Sharing in Mobile Cloud Environments Using Self-Generative Schnorr Certificateless Signcryption”, International Journal of Computer Networks and Applications (IJCNA), 11(6), PP: 896-914, 2024, DOI: 10.22247/ijcna/2024/54.