Random Self-Generative Schnorr Certificate less Signcryption for Secure Data Sharing in Mobile Cloud Environments

V.HARI PRASAD[1], MOHAMMED RAFI[2], MOHAMMAD KHALEEL AHMED[3],
ASHRAF BEN MILED[2], MAHAMMAD SHABANA[4]

[1]Govt. Polytechnic for Women Kadapa,
Andhra Pradesh,
INDIA

[2]Northern Border University,
SAUDI ARABIA

[3]Computer Science &amp; Engineering (Data Science) Department,
Vignan Institute of Science &amp; Technology,
Hyderabad,
INDIA

[4]Department of Computer Science &amp; Engineering,
Neil Gogte Institute of Technology,
Hyderabad,
INDIA

Abstract: - To deliver applications and services required by mobile devices, mobile cloud computing adopts cloud computing. Cloud computing assists data sharing among authorized users through data stored on cloud servers. Data sharing implies distributing data resources to multiple users or applications with the utmost security. But maintaining security while transferring data in cloud computing is an important challenge. In conventional algorithms, data sharing in a secure environment and control policies in accessing the data for authorized users are the biggest concerns to deal with. To handle these sensitive issues and to eliminate drawbacks by conventional algorithms, a novel technique termed Random Self-Generative Schnorr Certificateless Signcryption-based Secured Data Sharing (RSGSCS-SDS) is proposed in this work. The proposed work improves the required security in transferring data in mobile cloud environments. In this approach, initially the information of registered mobile cloud users can be stored in cloud server for getting access of permission for various services. Using the proposed RSGSCS algorithm, for each registered user, the cloud server generates private and public keys. Depending on the attributes of the policy, the authorization of the user can be verified by the cloud server after receiving the data access request. After receiving the authorization of the user, the data requested can be transmitted in the form of cipher text with a digital signature. This digital signature is verified by the user at his end, and the data is decrypted. This ensures that only users with authorization can access the original content, and this step improves the security of data sharing. The efficacy of the RSGSCS-SDS procedure is evaluated through key metrics like storage overhead, computational time, and data confidentiality rate. Experimental results demonstrate the effectiveness of the proposed method in enhancing secure data access in mobile cloud environments.

Key-Words: - Mobile cloud computing, cloud storage, data security, Random Self-Generative Schnorr Certificate less Signcryption, encrypted data, digital signature.

Received: May 19, 2024. Revised: December 16, 2024. Accepted: January 21, 2025. Published: April 10, 2025.

1 Introduction

Due to increased progress in cloud computing, many enterprises depend on cloud storage for storing their data efficiently and for sharing this with their authorized employees. However, security and privacy of data in mobile cloud environments are becoming difficult due to the increased requirement of lightweight and efficient operations by resource-constrained devices. Sharing sensitive data in these types of devices creates issues like retaining confidentiality, reduced computation time, and minimized storage overhead. Researchers proposed several methods to tackle these issues. Multi-Authority Ciphertext-Policy Attribute-Based Encryption with Elliptic Curve Cryptography (MA-CPABE-ECC) is proposed in [1] which offers reduced key sizes and minimized computation time with improved security. The main drawback of this procedure is it does not appropriately enhance data confidentiality.

The Lightweight Data Sharing Scheme (LDSS) proposed in [2] employs CP-ABE with access control. Because of the allocation of access control tasks to external proxy servers, LDSS faces challenges with increased storage overhead. In [3], the MGPV protocol is proposed which prevents potential attacks on data and protects document access for authorized groups of users by minimizing computational complexity. However, MCPV fails to effectively deal with the issue of time consumption. In [4], for the health sector authors proposed a procedure called Medi-Block which uses a bilinear mapping-based authentication system for tamper-proof record sharing. However, limitations in storage efficiency are the main drawback of this approach. In [5], biocomputing-based data security is future by adopting the polymerase chain reaction method. This method addresses data confidentiality and integrity efficiently, but its practical execution faces some challenges.

Privacy protection strategy which advances secured data sharing without resource duplication is proposed in [6]. For big data processing, hybrid cloud models are intended in [7]. This approach uses hybrid cloud models and peer-to-peer cloud systems (P2PCS). However, these models are not effectively adjusting data security. Federated learning-based secure data sharing (FL2S) is suggested in [8] for better data protection in IoT. In [9], privacy-preserving access control schemes are explained and in [10] DNA-based cryptographic techniques are advised for secured data sharing. These methods are complex and require more computational time.

In [11], the initial objective is to improve the security of cloud computing systems by an advanced attribute-based encryption technique. This method aims to address vulnerabilities combined with traditional encryption schemes by incorporating more granular control over access based on user attributes. In [12], researchers focus on presenting a role-based access control (RBAC) enabled public key encryption with keyword search (PEKS) mechanism to ensure secure and efficient data access in cloud environments. The scheme allows for secure keyword search within encrypted data while restricting access based on user roles.

In [13], the objective is to present a privacy-preserving outsourced cryptographic scheme for cloud-based IoT applications that support attribute-based encryption and verifiable access control policies. This scheme also allows for dynamic policy updates to accommodate changing IoT requirements. In [14], researchers combine hybrid cryptography methods for ensuring end-to-end encryption (E2EE) in multimedia cloud computing systems. This aims to provide data integrity and confidentiality for multimedia files stored in the cloud.

In [15], the authors propose a secure data access and sharing scheme for cloud storage that confirms the privacy and integrity of data while allowing authorized users to access and share it well. In [16], authors provide a privacy-preserving mechanism for attribute-based data sharing that allows dynamic group membership management in cloud environments. In [17], authors objective is to arrive a resilient cryptographic framework that aids secure data sharing within dynamic cloud groups while ensuring effective user revocation.

In [18], authors imply an attribute-based secure data sharing scheme that aims to give a reliable environment for sharing sensitive data in the cloud. In [19], authors objective is to improve security and privacy in cloud data sharing by using attribute-based encryption mechanisms to control access to sensitive information. In [20], authors aim to propose an effective framework for file sharing in cloud environments by using asymmetric key distribution management to ensure the secure exchange of files.

From the literature, some articles are reviewed and the definitions facing data security are increased computational complexity, more storage requirement, computation time, inadequate data privacy, and security differences. To address these problems, in this paper a data sharing method named as Random Self-Generative Schnorr Certificate less Signcryption-based Secured Data Sharing (RSGSCS-SDS) for mobile cloud environments.

2 Random Self-Generative Schnorr Certificate less Signcryption for Secure Data Sharing (RSGSCS-SDS) Technique

Random Self-Generative Schnorr Certificate-less Signcryption is a public-key cryptography method in which encryption and digital signature processes are included in one operation. Especially for data exchange mobile cloud environments, this proposed approach offers efficacy and security.

The security of the data is the major point in cloud data sharing, notably for servers and mobile users. The RSGSCS-SDS approach increases data security by merging public key encryption along with digital signature generation and verification algorithms. The sequence of these methods makes sure data protection and data authentication in a constant and applicable way. This makes sure that the future approach is fit for data sharing with the security in mobile cloud settings.

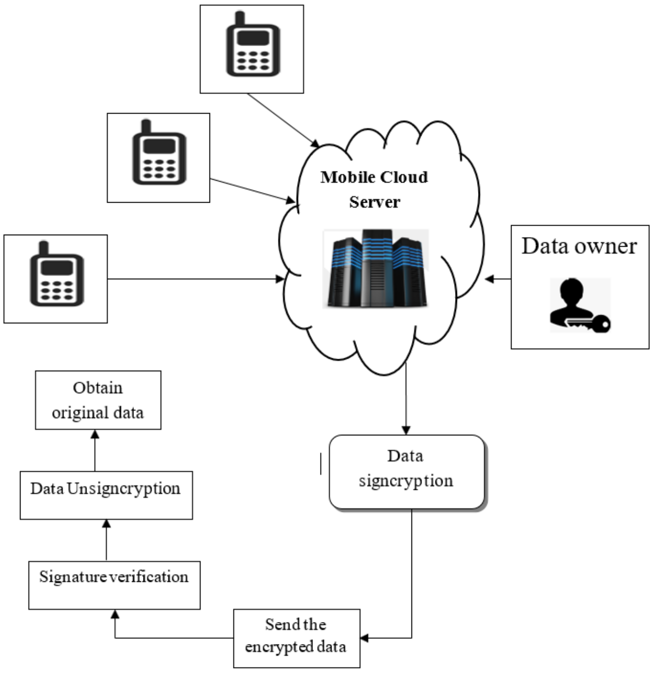

Fig. 1: Architecture Diagram of Certificate less Signcryption based Data Sharing in Mobile Cloud

The architecture of the RSGSCS-SDS approach to achieve data security in a mobile cloud environment is in Figure 1. The cloud-based architecture implies 3 main entities:

1. Mobile Cloud Users (MCUs): '$mcu_1, mcu_2, mcu_3, \ldots\ldots mcu_n$'
2. Mobile Cloud Server (MCS)
3. Data Owner (DO)

For enhanced protection for the data these presented components work together. The data can be uploaded to the MCS by the DO. When a MCU needs access to the data, initially the MCS verifies the user's identity. After validation of the user, the server sends the encrypted data and the digital signature to the mobile user. Then the mobile user performs sign verification and decryption to retrieve the original data.

2.1 Registration and Key Generation

The RSGSCS-SDS method works in 2 main steps: registration and key generation.

2.1.1 Registration

In this registration step, the MCS collects the data of the user. This data can be stored in the server's database. A one-time password (OTP) can be sent to the registered user by the cloud service provider. This OTP should be input by the user within pre specified time. If the OTP is not entered in time, the user must log in again and re-enter their details, as the OTP remains valid only for a limited duration. Upon successful entry of the OTP, the cloud server sends a registration confirmation message to the mobile user.

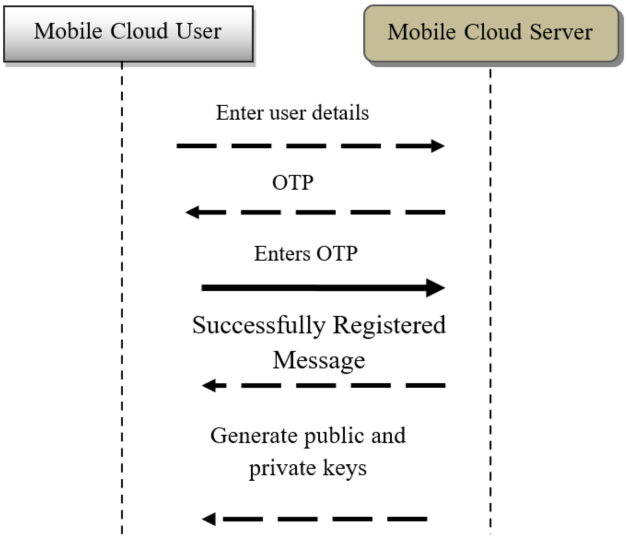

Fig. 2: Diagrammatic Representations of Registration and Key Generation

2.1.2 Key Generation

After the registration is complete, the mobile cloud server generates a random self-generative private key and a random self-generative public key for each registered user. These keys are essential for the encryption, decryption, and digital signature processes, ensuring secure communication between the server and users.

Figure 2 explains the registration and key generation process of the proposed RSGSCS-SDS technique.

$$mcu \xrightarrow{\textbf{Details}} MCS \tag{1}$$

From (1), 'MCS' represents the mobile cloud server, 'mcu' symbolizes the mobile cloud user.

The mobile cloud server transmits the OTP to the registered mobile number.

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

From (2), the mobile cloud user enters the OTP within a particular time period given by the mobile cloud server. After that, the mobile cloud server sent a Successfully Registered Message (SRM) to the mobile user. At that time, the mobile cloud server stored their details and generated the keys. Let us consider that, secret signature key (i.e., private key) 'PI' be the positive integer and it is generated with help of the Schnorr key generation algorithm.

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

From (3), 'Pri_{Key}' symbolizes the private key where 'PI' selected in random manner. After private key generation, the public verification key is attained as:

$$Pub_{Key} = f(PI) \qquad (4)$$

$$f(PI) = PI + 1\ mod\ 16 \qquad (5)$$

From (4) and (5), '$f(PI)$' represent the one-way function. 'Pub_{Key}' symbolize the public verification key attained from 'PI'. The public verification keys are employed as the User_id. The user id is considered as the public key, and it is distributed to the registered user. In Id generation process, the mobile cloud server generates different policy attributes. The keys are distributed to registered mobile users in the mobile cloud environment.

2.2 Signcryption

The RSGSCS-SDS technique uses signcryption to enable efficient and secure data sharing with authorized entities based on policy attributes. Signcryption is a public-key cryptosystem that integrates both digital signature and data encryption processes into a single, efficient operation.

2.2.1 Encryption and Decryption Process

- **Encryption**: The receiver's public key is used to encrypt the data, ensuring that only the intended recipient can decrypt it.
- **Decryption**: The receiver's private key is needed to decrypt the data, inducing it back to its original form, readable only by authorized users.

2.2.2 Data Access Workflow

When a registered mobile user requests access to a resource on the mobile cloud server, the user must first log in to the server. The mobile cloud server verifies the policy attributes that were generated during the registration phase to ensure the user has the necessary permissions. Once the policy attributes are validated, the server grants access to the encrypted resource. The process of attribute-based data access is depicted in Figure 3, illustrating how the policy attributes control secure data access and ensure only permissible users can read the encrypted data.

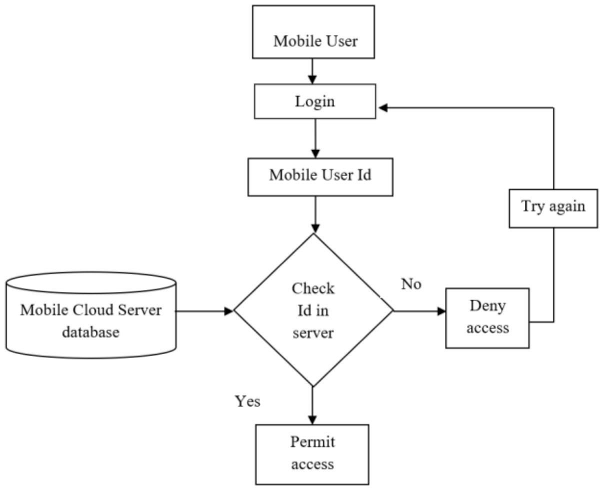

Fig. 3: Policy Attribute Schnorr Certificate less Signcryption Process

Whenever a registered mobile user needs to access data on the mobile cloud server, the user must enter a valid User ID in the login window. The mobile cloud server then verifies if the entered User ID matches the one stored in the server's database from the registration process.

2.2.3 Access Control through Policy Attributes

The mobile cloud server grants access rights based on policies that combine various attributes associated with the user. The Policy Attribute Schnorr Certificate less Signcryption employs 'if-then' rules to determine access:

- If the entered User ID matches the stored User ID, the user is considered authorized, and access is granted.
- If not, the user is considered unauthorized, and access is denied.

2.2.4 Data Sharing as Ciphertext

Once the mobile cloud server authenticates the user, the requested data is shared in the form of ciphertext. The mobile cloud server transmits the encrypted data to the authorized user, ensuring confidentiality and security during the data exchange. Figure 4 illustrates the architecture of the

Signcryption Process, detailing the steps involved in user authentication, encryption, and secure data sharing.

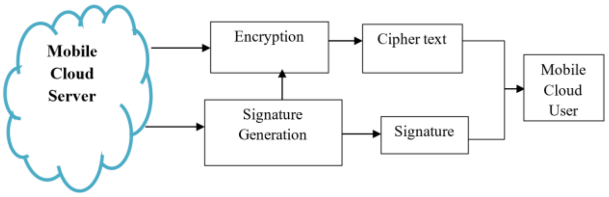

Fig. 4: Signcryption Process

Let us consider, the data is considered as $d_1, d_2, d_3, \dots . d_n$'. The ciphertext of data shared by mobile cloud server is given as:

$$Cipher(d) \leftarrow Encryption\langle Pub_{Key}, d\rangle \quad (6)$$

From (6), $Cipher(d)$ indicates the ciphertext of data 'd'. An encryption is carried out with the public key (Pub_{Key}) of the receiver i.e. user_Id. A digital signature is generated with the sender's private key. A valid digital signature presents the clarification to consider that data is created by the recognized sender (i.e. mobile cloud server) and the data is not altered by any intruders. Schnorr Certificateless Signcryption algorithm is carried out with the help of a private key. Let us consider that, the data $d = d_1, d_2, d_3, \dots . d_m$be (0, 1) and the signature generation is formulated as:

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

From (7), '$Signature_d$' denotes the signature of mobile cloud data 'd'. '$(PI\|)$' represent 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.

2.3 Un Signcryption

RSGSCS-SDS technique performs the unsigncryption depending on the signature verification and decryption to attain the original data. By applying the Schnorr Certificateless Signcryption, the signature verification is carried out.

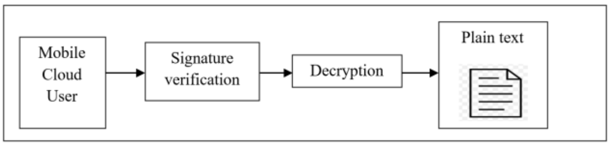

Fig. 5: Unsigncryption Process

Figure 5 describes the block diagram of the signature verification and decryption to attain the plain text. The signature verification is carried out with public key to attain the plain text. It is formulated as:

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

$$f\,(x) = if\ (Signature_d = Signature_d'')\ ;\ signature\ is\ valid \quad otherwise; \quad signature\ is\ not\ valid \quad (9)$$

From (8) and (9), '$Signature_d''$' represents the signature generated at receiver side, $f(x)$ represents the one-way function of signature. 'PI_v' represents the positive integer, h is the cryptographic hash function. Finally, checks the generated signature is $Signature_d''$ is verified with the public key 'Pub_{Key}'. When both the signature gets matched, it is valid and then mobile cloud user decrypts the ciphertext. Otherwise, the signature is not matched, and it is invalid. The mobile cloud user does not decrypt the ciphertext. This in turn helps to increase the data sharing security between the two entities. The authorized user decrypts the ciphertext and obtained the original data as:

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

From (10), 'd' denotes an original data. 'Pri_{Key}' denotes the private key of mobile user. Finally, the original data is attained after decryption and the output is shown at the output layer. The RSGSCS-SDS technique allows the authorized user to access the data from the mobile cloud server. By this way, data confidentiality gets improved by RSGSCS-SDS technique.

```
//Algorithm 1 Random Self-Generative Schnorr Certificateless Signcryption based Secured Data Sharing
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 input layer
// Registration and key generation
    For each mobile cloud user $mcu_i$
        Register the details to 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 '$mcu_i$'
                $MCS$ generates the private and public key
            End for
    End for
// Signcryption
        If mobile cloud user access the data then
            Login to the mobile cloud server with valid '$Id$'
        End if
            If (Id matched with server database) then
```

```
                    Mobile Cloud User is said to be an authorized user
                    $MCS$ permits the access
                else
                    Mobile Cloud User is said to be an 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 with the help of a private key
            Obtain the original data
        End if
End
```

Algorithm 1 outlines the step-by-step process of Random Self-Generative Schnorr Certificateless Signcryption (RSGSCS-SDS) for secure data sharing.

2.3.1 Process Overview:

1. Input:
   The algorithm begins by accepting the number of mobile cloud users as input.
2. User Registration and Key Generation:
   Each user undergoes registration, and the system generates a random self-generative private key and public key for each registered user.

2.3.2 Signcryption Process:

- ➢ Detection and validation of the users can be done by random self-generative Schnorr certificateless signcryption.
- ➢ When a MCU goes to access data, they must first log in with their User_ID.

2.3.3 User Authentication:

- ➢ The system shows if the entered User_ID matches the one stored in the server database in registration.
- ➢ If the IDs meet, the user is authentic and deemed allowed, giving them access.
- ➢ Then, the user is considered unauthorized, and access is denied.

2.3.4 Data Access and Signcryption:

- ➢ Once authenticated, the MCS grants access and makes the signcryption process on the data.

2.3.5 Signature Verification and Decryption

- ➢ On the user's side, the digital signature is shown by the public key.
- ➢ The user then decrypts the data with their private key to rescue the original text.

2.3.6 Outcome

The RSGSCS-SDS method confirms secure data access in the mobile cloud environment by uniting user authentication, encryption, and digital signature verification. This procedure advances security and ensures that only registered and authorized users can access the data.

3 Experimental Evaluation

Java programming language with CloudSim network simulator is used to perform the experimental assess of the RSGSCS-SDS technique. The Amazon Access Sample dataset is assumed to enable secure data sharing and access in the mobile cloud environment. This dataset is stored from the UCI Machine Learning Repository at UCI. The dataset has anonymized samples of access logs specified by the company. This data set contains many policy attributes, resource ID, group ID, and system support ID. These details are generated for registered users stored in the cloud server for read or write access. Its primary objective is to grant access to authorized users based on historical data.

4 Result Analysis

The efficacy of proposed RSGSCS-SDS technique is evaluated by comparing it with two other existing algorithms. One is Multi-Authority Ciphertext Policy Attribute-Based Encryption with Elliptic Curve Cryptography (MA-CPABE-ECC) [1] and second one is Lightweight Data Sharing Scheme (LDSS) [2]. Storage overhead, computational time, and data confidentiality are the three metrics used to compare existing algorithms and proposed algorithm .

4.1 Impact of Storage Overhead

Storage overhead is defined as the amount of memory required for the key generation process to achieve enhanced data security in a mobile cloud environment. It is calculated using the following formula:

$$Storage_{Over} = Number\ of\ data * Memory[private\ key + public\ key] \quad (11)$$

From (11), '$Storage_{Over}$' denotes the storage overhead. Units for the storage overhead is kilobytes (KB).

Table 1. Tabulation of Storage Overhead

Number of Data	Storage Overhead (KB)		
	MA-CPABE-ECC	LDSS	RSGSCS-SDS Technique
25	385	354	300
50	398	365	315
75	415	379	328
100	426	391	340
125	437	412	355
150	452	435	380
175	473	458	400
200	496	479	421
225	507	495	445
250	519	512	459

The storage overhead of proposed and existing methods is presented in Table 1 for the dataset ranging from 25 to 250. As presented in the comparison the storage overhead of the proposed RSGSCS-SDS technique is less than MA-CPABE-ECC and LDSS approaches. With a data set of 25 while sharing the data from the cloud server proposed RSGSCS-SDS technique achieves a storage overhead of 300 KB. But existing methods Multi-Authority Ciphertext Policy Attribute-Based Encryption with Elliptic Curve Cryptography (MA-CPABE-ECC) and the Lightweight Data Sharing Scheme (LDSS) achieve 385 KB and 354 KB respectively. The results comparison between the proposed RSGSCS-SDS technique and existing approaches MA-CPABE-ECC and LDSS are graphically represented in Figure 6.

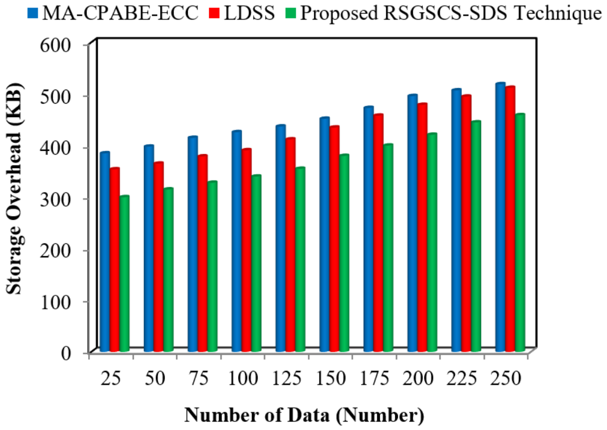

Fig. 6: Measurement of Storage Overhead

Figure 6 illustrates the comparative results of storage overhead concerning the number of data items. In the graph, the storage overhead of the three methods—RSGSCS-SDS Technique, MA-CPABE-ECC [1], and LDSS [2]—is represented in three different colors: blue, red, and green, respectively. The results indicate that the proposed RSGSCS-SDS Technique achieves a lower storage overhead compared to the existing methods.

This reduction is attributed to the Random Self-Generative Schnorr Certificateless Signcryption employed by the proposed technique. When a mobile cloud user needs to access data, they must verify their authenticity to ensure secure data sharing. The data owner conducts the authentication based on the user's identity. If the current mobile user ID matches the one stored during registration, the data owner requests data sharing from the cloud server. This approach allows the proposed RSGSCS-SDS Technique to accurately identify authorized and unauthorized mobile users. The average comparison results clearly demonstrate that the RSGSCS-SDS Technique reduces storage overhead by 17% compared to MA-CPABE-ECC [1] and by 13% compared to LDSS [2].

4.2 Impact of Computation Time

Computation Time is defined as amount of time consumed to perform secure data sharing between the cloud servers and user. The computation time is given as:

$$Comp_{Time} = [Number\ of\ data * Time\ consumed\ to\ share\ one\ data] \qquad (12)$$

From (12), '$Comp_{Time}$' denotes the computation time. Computational time is measured in terms of milliseconds (ms).

Table 2. Tabulation of Computation Time

Number of Data (Number)	Computation Time (ms)		
	MA-CPABE-ECC	LDSS	RSGSCS-SDS technique
25	41	34	25
50	43	37	27
75	45	39	29
100	47	41	30
125	50	43	31
150	52	45	33
175	55	47	34
200	57	49	37
225	59	51	39
250	61	53	41

Table 2 illustrates the computation time for three methods, using a dataset ranging from 25 to 250. The results indicate that the computation time of the proposed RSGSCS-SDS Technique is lower than that of the other two methods. For instance, when considering a dataset of 25 for sharing data from the mobile cloud server, the proposed RSGSCS-SDS Technique achieves a computation time of 25 ms. In contrast, the existing Multi-

Authority Ciphertext Policy Attribute-Based Encryption with Elliptic Curve Cryptography (MA-CPABE-ECC) [1] and the Lightweight Data Sharing Scheme (LDSS) [2] require 41 ms and 34 ms, respectively. Additionally, the remaining nine results have been obtained and are presented in the graphical representation below.

Figure 7 presents the comparative results of computation time concerning the number of data items. The graph displays the computation time for three methods: the RSGSCS-SDS Technique, MA-CPABE-ECC [1], and LDSS [2], represented in blue, red, and green, respectively. The results indicate that the proposed RSGSCS-SDS Technique achieves a reduction in computation time compared to the existing methods. This improvement is attributed to the use of the Random Self-Generative Schnorr Certificateless Signcryption process in the proposed technique. When a mobile cloud user needs to access data from the cloud environment, they must verify their authenticity for secure data sharing. The data owner performs the authentication based on the user's identity. If the mobile user ID matches the one stored at the time of registration, the data owner requests efficient data sharing from the cloud server. This process enables the RSGSCS-SDS Technique to accurately identify authorized and unauthorized mobile users while minimizing time consumption. The average comparison results clearly demonstrate that the RSGSCS-SDS Technique reduces computation time by 36% compared to MA-CPABE-ECC [1] and by 26% compared to LDSS [2].

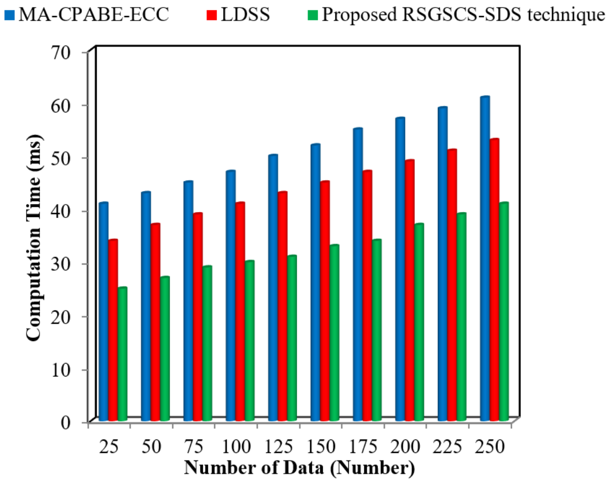

Fig. 7: Measurement of Computation Time

4.3 Impact of Data Confidentiality Rate

Data confidentiality rate is defined as the ability to preserve the data from unauthorized access in the cloud server. Data confidentiality is computed as the ratio of number of data accessed by authorized users to the total number of cloud data. The data confidentiality rate is formulated as:

$$Data\ Con_{Rate} = \left(\frac{Number\ of\ data\ accessed\ by\ the\ authorized\ users}{Total\ number\ of\ cloud\ data}\right) * 100 \quad (13)$$

From (13), '$Data\ Con_{Rate}$' represent the data confidentiality rate. The data confidentiality rate is computed in terms of percentage (%).

Table 3 presents the data confidentiality rate for three methods, using a dataset ranging from 25 to 250. The results indicate that the data confidentiality rate of the proposed RSGSCS-SDS Technique is higher than that of the two existing methods. For instance, when considering a dataset of 25 for data sharing from the mobile cloud server, the proposed RSGSCS-SDS Technique achieves a data confidentiality rate of 82%. In comparison, the existing Multi-Authority Ciphertext Policy Attribute-Based Encryption with Elliptic Curve Cryptography (MA-CPABE-ECC) [1] and the Lightweight Data Sharing Scheme (LDSS) [2] achieve rates of 75% and 78%, respectively. Similarly, the remaining nine results have been obtained and are depicted in the graphical representation for the data confidentiality rate.

Table 3. Tabulation of Data Confidentiality Rate

Number of Data (Number)	Data Confidentiality Rate (%)		
	MA-CPABE-ECC	LDSS	RSGSCS-SDS technique
25	75	78	82
50	77	79	83
75	78	81	85
100	81	83	86
125	82	85	88
150	84	86	89
175	85	87	91
200	88	90	93
225	89	91	95
250	90	93	96

Figure 8 presents the comparative results of the data confidentiality rate concerning the number of data items. The graph illustrates the data confidentiality rates for three methods: the RSGSCS-SDS Technique, MA-CPABE-ECC [1], and LDSS [2], represented in blue, red, and green, respectively. The results indicate that the data confidentiality rate is higher with the proposed RSGSCS-SDS Technique compared to the existing methods.

This improvement is attributed to the implementation of the Random Self-Generative Schnorr Certificateless Signcryption process. For secured data sharing, the authenticity of the mobile cloud user should be verified by the data owner to get access to the data. This authentication by the data owner is based on the user's identity. The data owner accepts the request for data sharing by the user after his ID matches the details given while the registration process. This approach of RSGSCS-SDS technique accurately identifies authorized and unauthorised users.

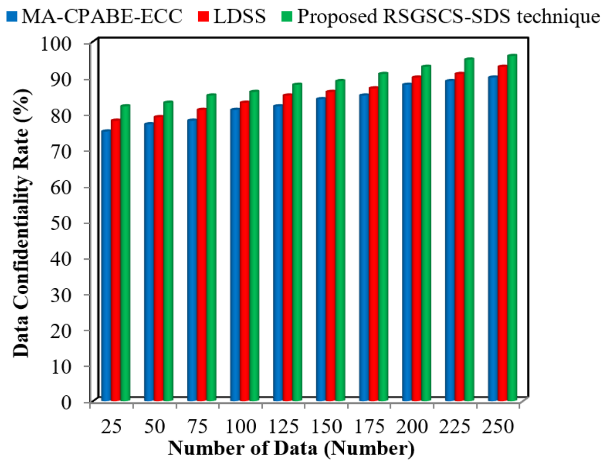

Fig. 8: Measurement of Data Confidentiality Rate

This technique results in efficient data sharing with increased security to the shared data. Experimental results presented in Table 1, Table 2 and Table 3 demonstrate that the proposed RSGSCS-SDS technique increases the data security rate by 7% compared to MA-CPABE-ECC [1] and by 4% compared to LDSS, [2].

5 Conclusion

In this paper, an efficient cryptographic technique, termed the RSGSCS-SDS approach, is proposed for secure data sharing in the mobile cloud with improved data security. The secured sharing of the data from the server to the user in a mobile cloud environment is achieved by employing Random Self-Generative Schnorr Certificateless Signcryption. In this approach, the accessing of the data from the mobile cloud server by the user is possible only when the data owner verifies the authenticity of the user by comparing it with the data provided by the user during the registration process. Signature generation and encryption are adopted to protect the data from unauthorized users and improve data security. The receiver then decrypts the ciphertext and retrieves the original data after the digital signature is valid during verification. Amazon access sample data set is adopted to check the efficacy of the proposed technique by performing experimental evaluation. Performance metrics such as storage overhead, computational time, and data confidentiality are compared between the proposed technique and existing approaches. The results confirm that the RSGSCS-SDS technique achieves increased data security with minimal storage overhead and computational time compared to the other two existing methods.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the authors used ChatGPT in order to proofread the paper. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

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, Volume 13, Issue 1, 2021, Pages 3893–3902 https://doi.org/10.1007/s12652-021-03287-6

[2] Ruixuan Li, Chenglin Shen, Heng He, Xiwu Gu, 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 https://doi.org/10.1109/tcc.2017.2649685

[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 https://doi.org/10.1016/j.future.2019.01.034

[4] Chaitanya Singh, DeepikaChauhan, 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 https://doi.org/10.1016/j.imu.2021.100624

[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 https://doi.org/10.1016/j.jksuci.2018.03.008

[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 https://doi.org/10.1016/j.dcan.2021.09.003

[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 https://doi.org/10.1016/j.jksuci.2019.05.007

[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 https://doi.org/10.1016/j.comnet.2021.108327

[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 https://doi.org/10.1109/access.2018.2844829

[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 https://doi.org/10.1016/j.jksuci.2018.09.024

[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 https://doi.org/10.1002/cpe.4364

[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 https://doi.org/10.1109/access.2019.2941560

[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 2019, Pages 899-918 https://doi.org/10.1016/j.future.2019.11.012

[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 https://doi.org/10.35940/ijitee.j9001.0881019

[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 https://doi.org/10.1007/s11277-016-3742-6

[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 https://doi.org/10.1109/jsyst.2018.2865221

[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 https://doi.org/10.17762/turcomat.v12i3.2144

[18] Nabeil Eltayieb, Ping Wang, Alzubair Hassan, Rashad Elhabob and Fagen Li "ASDS: Attribute-based secure data sharing scheme for reliable cloud environment", Security and Privacy, Volume 2, Issue 2, March/April 2019, Pages 1-11 https://doi.org/10.1002/spy2.57

[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 https://doi.org/10.1109/jsyst.2019.2911391

[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, Issue 1, 2019, Pages 1-15 https://doi.org/10.1155/2019/9852472

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- V. Hari Prasad carried out conceptualization, data curation, formal analysis, investigation and software.
- Mohammed Rafi carried out methodology, project administration and software.
- Mohammad Khaleel Ahmed carried out methodology, resources, software and validation.
- Achrar Ben Miled carried out methodology, software, validation, visualization and writing original draft.
- Mahammad Shabana carried out methodology, software, validation, visualization and writing original draft.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.