Sharing PHR Data in Cloud Using Sigmoid Key and Median Support Signature-Based Cryptosystem

A. Jain Austin

Accepted: 6 January 2022 / Published online: 18 January 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
In recent years it is a major and developing challenge to provide the security of the data that have been stored in personal health records (PHR). Some encryption and key generation techniques in classical works have been created for this purpose. However, it has disadvantages such as ineffective and decreased security rules in access control regulations. To improve the security performance of PHR data for Sigmoid-based signature cryptosystem is proposed to transfer the data stored in a third-party cloud server securely. In the proposed work, the PHR data is taken as an input and the key encrypting key algorithm is implemented for the attribute key generation. After the attribute, the key generation sigmoid is calculated for the PHR data to keep the key secured. To keep the personal data of the patient in a secured manner the median support-based signature is generated and without displaying the generated signature the cipher text is displayed. To gain access to information stored in the cloud, the users have to verify the both generated private key and signature. On the successful verification of both the private key and signature, later, if the data requested by the user is a valid user then the user will be getting permission to access the cloud. The experimental findings show that our proposed technique outperforms other current methods in key computation time, communication complexity, encryption time, decryption time, and cloud operating mechanism.

Keywords: Attribute key · Sigmoid measure · Signature generation · KEK · Attribute-based encryption (ABE)

1 Introduction

PHR has evolved as a patient-centric paradigm of health data sharing in recent years. A PHR service enables a patient to generate, complete, and maintain her health information in one location through the internet, making the storage, retrieval, and exchange of medical data much more cost-effective. Every patient, in particular, is safe in owning complete control of their medical data and may share their health information with a good type of user, such as health care traders, members of the family, or friends [1–4]. Because of the
importance of structure and maintaining particular knowledge centers, some PHR services are outsourced to or offered by third-party service providers, such as Microsoft Health-Vault1. Recently, cloud computing systems for storing PHRs have been planned. Many IT firms are reaping the benefits of cloud servers by outsourcing their database [5–7].

The cloud is a commonly utilized analytical area that is mostly used to store and access data through the internet. It enables network connectivity with greater flexibility and portability at a cheap cost. Providing security for cloud storage is a very significant and necessary task. Cloud storage provides virtualized storage for a specific quality of service based on demand (QoS). Cloud storage may be used in a variety of ways. For example, native knowledge may be protected to cloud storage; a virtual disc could be the cloud associate degree dispersed to many computers, and so the cloud could be utilized as an archive to maintain knowledge [8]. Nowadays, a lot of and a lot of health records (EHRs) systems square measure increasingly victimizing cloud storage services (e.g., Google Health) for storing and retrieving the records to boost interoperability, which may avoid exposing individuals to additional examination and supererogatory costs. Information exchange in EHR systems is critical for improving the service quality. However, information sharing has created certain security and privacy concerns by storing sensitive health information in third-party cloud service providers, as a result of which health care information is likely to be accessed by a wide range of users, potentially exposing patients’ privacy [9–11].

There are several sorts of coding algorithms that have been created for encrypting information before it is sent to the cloud [12, 13]. The primary goal of coding approaches is to provide strong security, access control, and information secrecy to cloud data [14]. PHR makes use of cloud computing technologies to allow the transmission of health data in a safe environment [15]. The key generation algorithmic rule was abused by the public key. Every certificate with time T has a time seal described by the information owner. To generate the ciphertext, the ECC algorithmic rule [16] method is used. To improve the secrecy of users with little expertise, multi-certificate routing is used [17].

Various techniques are projected to safeguard the information contents’ privacy via access management. Identity-based encryption (IBE) associate with causing an email. It makes the cryptographic aspects of the communication nearly clear to the user, and it may be used effectively even by laymen United Nations agency unskilled person regarding keys or protocols. When a message is sent, the sender specifies the recipient’s identity, and only a receiver with the same identity decodes it [18]. Fuzzy Identity-Based secret writing, also known as ABE, is planned for a few years later. Identity is seen as a set of descriptive qualities in such a secret writing scheme, and decipherment is achievable if a decrypt’s identity overlaps with the one per the ciphertext [19]. Following that, in more generic tree-based ABE schemes such as Key-Policy Attribute-Based Encryption (CP-ABE) [20] and Cipher text-Policy Attribute-Based Encryption (CP-ABE) [20], a user’s private key is associated with an indefinite number of attributes represented as strings. On the other side, once a party encrypts a message in our framework, they provide an access structure over characteristics. A user can only be ready to rewrite a ciphertext if the user’s characteristics match the access arrangement of the ciphertext. The remainder of the paper contribution is as follows:

- Develop a KEK is implemented for the attribute key generation.
- Keep the generated key in a secured manner the sigmoid value is calculated for the identified attribute.
• Encryption technique is mainly used to keep the patient health records in a secured manner.
• Keep the information in a more secured manner the signature feature associated with the cryptosystem is added along with the calculated sigmoid value.

The remainder of the article is organized as follows: Sect. 2 displays the related work. Section 3 introduces the suggested methodological design, as well as the implementation stages for the proposed algorithm. Section 4 goes over the experimental setup, findings, and comparative analysis. Section 5 brings the whole project to a close.

2 Related Work

Rui Guo et al. [21] had proposed a convention for some unique access structures into a solitary one, which the various leveled PHR is scrambled dependent on. There are numerous specialists to create and convey the client’s private key altogether. As per this model, it empowers to stay away from the issue of key escrow and adjust to the conveyed normally for cloud administration frameworks. Be that as it may, it has no confided in single or focal one in these specialists.

Deepa1 and Pandiaraja [22] proposed that data be recovered from the cloud via an attribute-based file encoding technique. In comparison to current protocols, the ERFC mechanism requires the least amount of computing and communication quality for four operating mechanisms: patient key computation, doctor index building computation, cloud operating mechanism, and finally patient report secret writing. All four working techniques have been designed to provide successful file recovery to end-users.

Mythili1 et al. [23] had recommended that ABSC use ABE and attribute-based marks to achieve protection-focused categorization in addition to authenticity. Surprisingly, a considerable percentage of the current ABE and ABSC designs impose significant processing overheads to the key length, ciphertext size, or expressive access structures used. Another ABSC scheme is being devised to reduce computational overheads, particularly in the cloud, by offering the more sensitive hypergraph access structure, Attribute Hypergraph.

Hong Zhong et al. [24] had proposed an effective and attribute-based admittance control plot upheld by rethinking, for edge empowered keen medical services. Contrasted and the conventional characteristic encryption conspire, re-appropriate fractional encryption and unscrambling tasks to the edge hubs, decrease the figuring heap of asset compelled gadgets, and ensure the protection and security of clinical information. Simultaneously, the plan underpins trait refreshing, which improves the effectiveness of the plan.

Azath Mubarakali et al. [25] had proposed a huge number of customers and information proprietors in the PHR structure who had potentially weighty computational and organization load on the components in the system; which confine the PHR data openness and convenience. To find a better solution to the aforementioned difficulties, Attribute-based Health Record Protection (AHRP) computation is familiar with providing data access control categorization, validity, and mystery. It is an access control arrangement that scrambles the data and is a useful method for validating a message without disclosing the patient’s personal information.
Man Ho Aua et al. [26] have suggested an affirmative solution to the problem by proposing a generic architecture for secure PHR sharing. The suggested solution allows patients to safely store and share their PHR in the cloud server, and treating clinicians can refer the patients’ medical data to experts for research commitments as necessary, all while keeping the patients’ information private. The suggested method is also concerned with cross-domain activities.

Liu et al. [27] an authentication method was ensured that the system was login in and system service resources were obtained only by genuine users through verification. In this study, a safe, confident user authentication method for a personal health records system in the cloud environment provides appropriate people with cloud-based access to PHRs. A password in combination with a smart card employed the suggested method of authentication, which enables the owner and authorized users to log in and access the appropriate personal documents to the system.

Liu, Yaru et al. [28] The DSSE Scheme for IIoTH is presented for privacy conservation. It is the first DSSE strategy for the security PHR database. The secure index is based on a Hash Chain and updates trapdoor file resistance. We also search fine-grained PHR files of an attribute value type database using encrypted PHR files. If the user does search operations, the attribute value only matches the whole file. Consequently, the costs of communication are lowered and the privacy of the patient minimized. The proposed scheme has to accomplish access control attributes, which allows users to ascribe different access authorities. The comparison analysis of existing papers is given in Table 1.

### 3 Proposed Sigmoid and Median Support Based Signature-Based Cryptosystem

This proposed method presents a Sigmoid and median support-based signature-based cryptosystem for sharing the PHR data secured in the cloud environment the proposed method has two phases; the initial phase is key generation. In attribute selection then based on the

| References                           | Technique                              | Efficiency | Security level | Computational overhead |
|--------------------------------------|----------------------------------------|------------|----------------|------------------------|
| Rui Guo et al. [21]                  | CP-ABE                                 | Medium     | High           | High                   |
| Deepa1 and Pandiaraja [22]           | ERFC                                   | Medium     | High           | Medium                 |
| Mythili1 et al. [23]                 | ABSC                                   | High       | Medium         | Medium                 |
| Hong Zhong et al. [24]               | Decisional bilinear diffie–Hellman (DBDH) | High       | Medium         | High                   |
| Azath Mubarakali et al. [25]         | AHRP                                   | Medium     | High           | High                   |
| Man Ho Aua et al. [26]               | Linear Secret-Sharing Schemes (LSSS)   | Low        | Medium         | High                   |
| Liu et al. [27]                      | Bilinear pairing                       | Medium     | High           | High                   |
| Liu and Yaru et al. [28]             | DSSE                                   | Medium     | High           | High                   |
attribute key is generated. After the attribute, key generation Sigmoid is calculated for the key secured. The ciphertext is displayed instead of displaying the sigmoid value.

In the second phase to keep the personal data of the patient in a secured manner the median support value-based signature is generated and without displaying the generated signature the ciphertext is displayed. In the Decryption process to users access the information in the cloud, they could have to verify the both generated private key and signature. Once the private key verification is successful, then the generated signature is validated to keep the data and send it to the user. The entire work procedure is shown in Fig. 1.

### 3.1 Attribute-Based Key Encryption Key Algorithm

To keep the information, secure in the cloud environment the security key has to be generated for the attribute $AB$ is identified using the KEK algorithm. In previous, Pairwise Temporal Key (PTK) is the set of assigned session keys. So in this work consider the KEK algorithm the proposed encryption process two types of keys, public and private keys are used [27]. The keys are mainly of two types they are public and private keys. In such a way the public key is used for signature generation and sigmoid calculation but in this case of the private key is used for decryption. KEK algorithm is efficient on behalf of both the encryption and decryption process. The KEK algorithm-identified characteristic may be used to create keys for both the encryption and decryption processes. The Attribute-based KEK algorithm steps are defined as follows.

![Block diagram of sigmoid and median support-based signature cryptosystem](image-url)
Step 1 Choose a random number $M$ and computer public key using $bk = 2^{M+1}$.
Step 2 Select the attribute value in the dataset $\nabla H \in \{AB_1, AB_2, AB_3, \ldots, AB_k\}$.
Step 3 Generate two large random and distinct prime numbers $p, q$.
Step 4 Compute the $N_i = p \times q$ and $\phi(i) = (p - 1) \times (q - 1)$.
Step 5 Compute private key using $Pki = N_i \times 2 \mod \phi(i)$.
Step 6 Repeat the process Step 2 to Step 5 until the value $k$. The Attribute KEK algorithm which generates the key is shown in algorithm 1.

\begin{algorithm}
\textbf{Input} : $\nabla H \in \{AB_1, AB_2, AB_3, \ldots, AB_k\}$
\textbf{Output} : $\{Pk\}, \{bk\}$
\textbf{Begin}
       Randomly choose a number $M$
       Compute the public key value $bk = 2^{M+1}$
       For $i = 1$ to $k$
       Select attribute $AB_i$
       Make two big random and separate prime numbers $p, q$
       Such that $(p - 1)/2, (q - 1)/2$ be also large prime numbers.
       Compute $N_i = p \times q$ and $\phi(i) = (p - 1) \times (q - 1)$
       Compute $Pki = N_i \times 2 \mod \phi(i)$
       $AB_i \leftarrow \{Pk\}$ is the private key $AB_i \leftarrow \{bk\}$ is the public key of the cryptosystem.
\textbf{End for}
\textbf{End}

\textbf{Algorithm 1:} Attribute-based KEK algorithm

3.2 Key Based Sigmoid Calculation

To keep the generated key in a secured manner the sigmoid value is calculated for the identified attribute. The Sigmoid calculation formula is given below.

$$\text{Sig}(Pki) = \frac{\log_2(Pki)}{1 + Pki}$$

(1)

From the PHR database $\nabla H$, using the above attribute-based KEK algorithm the most important is selected for the selected attribute the sigmoid value is obtained. The Sigmoid value is kept in the form of the secret key. The generated sigmoid value is not visible to everyone. Only the valid user has permission to access the sigmoid secret value.

3.3 Encryption Based on Key Based Sigmoid

Encryption technique is mainly used to keep the patient health records in a secured manner. Initially, the PHR $\nabla H$ database was converted into ASCII values. Then XOR by a random number 1 or 2 and encryption using step 3. The algorithm steps for the proposed encryption based on the key are given in algorithm 2.
Input : \( \forall H \in \{ PH_1, PH_2, PH_3, \ldots, PH_n \}, \text{Sig}(P_k) \)
Output : \( CM \)

Begin
PK = \text{Sig}(P_k)
For \( i = 1 \) to \( n \)
  For \( j = 1 \) to \( k \)
    Step 1: \( \forall H \in \{ PH_1, PH_2, PH_3, \ldots, PH_n \} \) is converted into ASCII.
    Step 2: Each data is XOR by a random number either 1 or 2.
    \( \text{Msg} = PH(i, j) \oplus \text{rnd} \)
    Step 3: \( \text{EM}(i, j) = \text{Msg} \oplus H_j \)
  End for
End For
End

Where, \( \text{EM} \) is the encrypted message, \( \text{Msg} \) is the input message, \( PK \) is the Encrypted private key, \( n \) is the number of records, \( j \) is the number of attributes

Algorithm 2: Encryption based on Key Based Sigmoid

3.4 Generate the Signature-Based Cryptosystem

To keep the information in a more secured manner the signature feature associated with the cryptosystem is added along with the calculated sigmoid value. A signature is used to verify the information between both the sender and receiver sides. In this case, the signature is used for verifying the public key which is kept secret. The algorithm used to generate the signature-based cryptosystem is shown in algorithm 3.

Algorithm 3: Encryption using median Support based Signature

Using this signature algorithm, the key size is considerably reduced. The main reason for using this signature-based cryptosystem algorithm is the integrity and non-reputation are considerably increased. Thus the key size is also minimized.
3.5 Cipher Text

The data in its encrypted form is referred to as ciphertext. The ciphertext is also known as encrypted information since it is unintelligible by humans or computers without the correct algorithm. The encrypted message \((EM)\) using the KEK algorithm in the form of ciphertext and the generated signature \((\theta)\) based on support value is stored in the cloud.

3.6 Decryption Based Signature Verification

After storing the information in the cloud database, if the hospital or the Insurance Policy Company needs to access the information stored in the database then the signature verification process has to be performed. In the signature verification process, the public key is used to verify the signature to get the information access. The algorithm used for the signature verification process is shown in algorithm 4.

| Input | \(\sigma\) |
|-------|------------|
| Output: | Authenticated or Unauthenticated |
| Begin | |
| For \(i=1\) to \(n\) | |
| For \(j=1\) to \(k\) | |
| Step1: | Split the signature value \(\sigma\) |
| Step2: | Compute the median values of \(M_{1} = (PK - R_{1})\) and \(M_{2} = (PK - R_{2})\) |
| Step3: | Compute XOR values \(\hat{S}_{1} = M_{1} \oplus \alpha\) and \(\hat{S}_{1} = M_{2} \oplus \beta\) |
| Step4: | Compute median support value \(SV = \frac{\hat{S}_{1} + \hat{S}_{2}}{\hat{S}_{1} \times \hat{S}_{2}}\) |
| Step5: | Compute an inverse defy value \(CHL \leftarrow H(bk, EM, SV, \hat{S}_{1}, \hat{S}_{1})\) |
| Step6: | Check that the challenge \([CHL = = \rightarrow CHL]\) |
| Step7: | If it is Yes, authenticated. Otherwise, unauthenticated. |
| End For | |
| End For | |
| Algorithm 4: Signature verification process |

If the defy value is equal, the verification step leads to the signature being considered effective, and the data is shared with the signature verified user. If the challenge value is not equal, the user results in an invalid signature, and the data is not shared with that user. Once the signature verification procedure is finished, the decryption process will begin.

3.6.1 Decryption Using Key

Once all the above process is completed the authenticated finally the decryption stage is carried out. The decryption stage only permits to access the information stored in the cloud database and a matching public key is used to decrypt the message. The decryption algorithm is shown in algorithm 5.
Input : CHL
Output: Decrypted Data $D_{msg}$

Begin
For $i=1$ to $n$
    For $j=1$ to $k$
        Step1: $Msg(i, j) = CHL(i, j) \oplus bk$
        Step2: Each data is XOR by a random number either 1 or 2.
            $D_{msg} = Msg(i, j) \oplus rnd_i$
        Step3: The Decrypted Data \{ $D_{msg1}$, $D_{msg2}$, ..., $D_{msgn}$ \} is converted into String.
    End For
End For
Where, $Msg(i, j)$ is the decrypted message, CHL is the Encrypted message, $bk$ is the public key
End

Algorithm 5: Decryption using Key

The decryption stage is performed with the help of an asymmetric key. Using the asymmetric key, the user gets permission to use the information stored in the cloud database. The algorithm output is valid only if both the key verification and signature verification process accept, otherwise it is invalid. Once the verification process is completely verified the insurance company or the hospital gets the full rights to access the information which is stored in the cloud database. By performing both the encryption and decryption technique only the user gets the right to upload or download the contents from the cloud database. Cloud databases cannot be easily destructed the information stored in the cloud database can withstand for so many years without doing any modification. If the user wants to do any alteration in the information stored in the cloud database, then also the authentication step has to be completed before uploading the contents to the cloud database.

4 Results and Discussion

The suggested Sigmoid and signature-based cryptosystem method has been implemented in several database sizes ranging from 1000 to 10,000 records [18]. The database has been designed in such a way that it can accommodate a wide variety of difficulty and discrimination levels. In this section, the experimental results of the proposed framework are achieved using an Intel Core I3 processor with chipsets 2.0 GHz and 6 GB RAM 500 GB transfer and 300 GB storage by developing a security framework to prevent unauthorized data access using JAVA cloudsim. This section discusses the outcomes of both the existing and suggested techniques in terms of key computation time and communication complexity, Encryption time, decryption time, and Cloud working mechanism. Each criterion is checked and its performance is shown in the form of a graph. Compared to the other technique ERFC [22], FFE [22], FIBE [22], AHRPA [25], and CP-ABSC [23] the proposed technique provides better results.
4.1 Key Computation Time

The Key Computation time is depicted in Fig. 2. For 10,000 record sizes, the calculation time for ERFC is 5 ms, for FFE 42 ms, for AHRPA 34 ms, and using CP-ABSC 54 ms, for our suggested approach. The computation time is 3 ms. When compared to the existing technique, our proposed technique takes the least amount of time.

4.2 Communication Complexity

The time length between the beginning and ending time of a selected task is outlined because of the communication value. The procedure quality is computed mistreatment the derivation,

$$B_c = d_e(b) - d_f(b)$$

(2)

where $B_c$ is the communication cost, $d_e(b)$ is the finish time of every assumed task,$d_f(b)$ is the initial time of any assumed task. The communication complexity is authorized with the existing technique and the graph is exposed in Fig. 3. The computing complexity for ERFC is 34 ms for 10,000 record sizes, FFE is 56 ms, AHRPA is 15 ms, CP-ABSC is 33 ms, and
our proposed technique is 13 ms, demonstrating that our proposed technique has a lower computation complexity than other techniques.

4.3 Encryption Time

Figure 4 exemplifies the encryption time of patients. For the encryption portion, the Encryption period for ERFC is 55 ms for 10,000 record sizes, FFE proceeds 93 ms, AHRPA needs 44 ms, for CP-ABSC the Encryption time 85 ms, for our proposed technique the Encryption time is 27 ms, which exhibitions that our proposed technique has small Encryption time than other technique.

4.4 Decryption Time

The time taken to decrypt the secret message is decryption time. It is important in terms of giving access permission to the cloud. Figure 5 signifies the decryption time of the patient record. For the Decryption section, the Decryption time for ERFC is 59 ms for 10,000 record sizes, FFE earns 84 ms, AHRPA requires 52 ms, CP-ABSC requires 78 ms, and
our proposed technique requires 28 ms, demonstrating that our proposed technique has a shorter Decryption time than other techniques.

### 4.5 Cloud Working Mechanism

The Cloud Working Mechanism is depicted in Fig. 6. For 10,000 record sizes, the calculation time for ERFC is 31 ms, FFE is 33 ms, AHRPA is 44 ms, CP-ABSC is 47 ms, and our suggested protocol approach is 15 ms. When compared to the existing technique, our proposed technique mechanism requires the least amount of time.

### 4.6 Security Levels

The security value calculated using the proposed technique is kept in the form of ciphertext. Key sizes for equivalent security levels for different encryption methods are listed in the following Table 2. The comparison Table 2 proves that our proposed technique-based encryption is better than the existing technique in terms of different parameters.

Table 3 displays the comparative findings for the area unit. This demonstrates that our approach achieves high levels of on-demand cancellation and private securities. The conjunctive rule restricts PUD to single permission, but in PSD, a user’s access arrangement might still be a chance monotonic formula. As a file access policy, the CP-ABSC supports any monotonic Boolean formula. The proposed method, on the other hand, is more effective in terms of communication overhead, as seen by the comparison in Table 3. In CP-ABSC, the data holder must recompute and then send fresh ciphertext components matching canceled characteristics to all remaining users after each cancellation event.

### 5 Conclusion

This paper describes how we obtained the signature-based cryptosystem method for keeping the information secured in the cloud database. Various steps are involved for determining the variation in technique on behalf of ERFC, FFE, FIBE, AHRPA, and CP-ABSC. Initially, the KEK algorithm is carried out to generate the key for keeping the information in a
| Factors             | ERFC Key size          | FFE Key size          | AHRPA Key size          | CP-ABSC Key size          | Proposed technique          |
|---------------------|------------------------|-----------------------|-------------------------|---------------------------|-----------------------------|
| Key size            | 128, 192, 256 bits     | 256 bits              | 512 bits                | 512 bits                  | > 1024 bits                 |
| Block size          | 128 bits               | 64 bits               | Minimum 256 bits        | Minimum 256 bits          | Minimum 512 bits            |
| Ciphering and deci-| 128 bits               | Same                  | Different               | Different                 | Different                   |
| phering key         | Same                   | Same                  |                         |                           |                             |
| Encryption          | Faster                 | Faster                | Faster                  | Faster                    | Faster                      |
| Decryption          | Slower                 | Slower                | Faster                  | Faster                    | Faster                      |
| Key used            | The same key used for | The same key used for | The many keys used for  | The many keys used for    |                             |
|                     | encrypting and decrypt | encrypting and decrypt | encrypting and decrypt  | encrypting and decrypting |                             |

Table 2: Key sizes for equivalent security levels (in bits)
| Scheme       | Security                      | User domains     | Access policy                        | Revocation means                  |
|--------------|-------------------------------|------------------|--------------------------------------|-----------------------------------|
| ERFC         | Against N—2 AA collusion      | All (PSD&PUD)    | Conjunctive form including a wildcard| Level of attribute, immediate     |
| FFE          | Against N—1 AA collusion      | PUD              | Any Boolean formula that is monotone  | Level of attribute, immediate     |
| AHRPA        | Not against user-server, single TA | PUD              | Any Boolean formula that is monotone  | Level of attribute, immediate     |
| CP-ABSC      | Not against user-server, single TA | Signature        | Policy on Attributes and Signatures   | Immediate ACL level               |
| Proposed technique | Single TA                  | Signature        | Policy on Attributes and Signatures   | Immediate ACL level               |
secured manner. Sigmoid is calculated for the newly generated key and the key-encryption process is carried out for the private key. On another side for the data retrieved from the PHR database after the encryption step the signature is calculated and kept secure. The valid users only have access permission to access the information from the cloud database. The verification step is carried out before accessing the cloud. Performance analysis of our proposed work is computed concerning the Key computation time, communication Complexity, Encryption time, decryption time, Cloud working mechanism, and security values. The experimental outcomes verify that our signature-based cryptosystem system has a better outcome comparing with the existing technique.

Author Contributions The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

Funding In this research article has not been funded by anyone.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Preethi, P., & Asokan, R. (2019). An attempt to design improved and fool proof safe distribution of personal healthcare records for cloud computing. Mobile Networks and Applications, 24(6), 1755–1762.
2. Qian, H., Li, J., Zhang, Y., & Han, J. (2015). Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation. International Journal of Information Security, 14(6), 487–497.
3. Liu, X., Xia, Y., Yang, W., & Yang, F. (2018). Secure and efficient querying over personal health records in cloud computing. Neurocomputing, 274, 99–105.
4. Gondkar, D. A., & Kadam, V. S. (2014). Attribute based encryption for securing personal health record on cloud. In 2014 2nd International Conference on devices, circuits and systems (ICDCS) (pp. 1–5). IEEE.
5. Wang, X. A., Ma, J., Xhafa, F., Zhang, M., & Luo, X. (2017). Cost-effective secure E-health cloud system using identity based cryptographic techniques. Future Generation Computer Systems, 67, 242–254.
6. Li, M., Shucheng Y., Ning C., & Wenjing L. (2011) Authorized private keyword search over encrypted data in cloud computing. In 2011 31st international conference on distributed computing systems, pp. 383–392. IEEE.
7. Li, M., Shucheng, Yu., Zheng, Y., Ren, K., & Lou, W. (2012). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE transactions on parallel and distributed systems, 24(1), 131–143.
8. Stephen, O. (2014). The study of the application of data encryption techniques in cloud storage to ensure stored data integrity and availability. International Journal of Scientific and Research Publications, 4(10), 1–7.
9. Eludiora, S., Abiona, O., Oluwatope, A., Oluwaranti, A., Onime, C., & Kehinde, L. (2011). A user identity management protocol for cloud computing paradigm. International Journal of Communications, Network and System Sciences, 4(3), 152.
10. Mishra, N., Tarun, K. S., Varun, S., & Vrince, V. (2018). Secure framework for data security in cloud computing. In Soft computing: Theories and applications, pp. 61–71. Springer, Singapore, 2018.
11. Florence, M. L., & Dhina, S. (2019). Enhanced secure sharing of PHR’s in cloud using user usage based attribute based encryption and signature with keyword search. Cluster Computing, 22(6), 13119–13130.
12. Shirole, B. S., & Vishwamitra, L. K. (2020). Review paper on data security in cloud computing environment. In 2020 9th International conference system modeling and advancement in research trends (SMART), pp. 79–84. IEEE.

13. Athena, J., Sumathy, V., & Kumar, K. (2018). An identity attribute–based encryption using elliptic curve digital signature for patient health record maintenance. International Journal of Communication Systems, 31(2), e3439.

14. Chen, D., Linling, C., Xiaowei, F., Liwen, H., Su, P., & Ruoxiang, H. (2014). Securing patient-centric personal health records sharing system in cloud computing. China Communications, 11(13), 121–127.

15. Rodrigues, J. J. P. C., De La Torre, I., Fernández, G., & López-Coronado, M. (2013). Analysis of the security and privacy requirements of cloud-based electronic health records systems. Journal of medical Internet research, 15(8), e2494.

16. Mhatre, S., & Nimkar, A. V. (2019) "Secure cloud-based federation for EHR using multi-authority ABE. In Progress in advanced computing and intelligent engineering, pp. 3–15. Springer.

17. Sultan, N. H., Kaaniche, N., Laurent, M., & Barbhuiya, F. A. (2019). Authorized keyword search over outsourced encrypted data in cloud environment. IEEE Transactions on Cloud Computing. https://doi.org/10.1109/TCC.2019.2931896

18. Ezhilarasan, E., & Dinakaran, M. (2021). Privacy preserving and data transpiration in multiple cloud using secure and robust data access management algorithm. Microprocessors and Microsystems, 82, 103956.

19. Mishra, P., & Vandani, V. (2020) Study of identity-based encryption for cloud data security. In Decision analytics applications in industry, pp. 401–408. Springer.

20. Ali, M., Abbas, A., Khan, M. U. S., & Khan, S. U. (2018). SeSPHR: A methodology for secure sharing of personal health records in the cloud. IEEE Transactions on Cloud Computing, 9(1), 347–359.

21. Guo, R., Li, X., Zheng, D., & Zhang, Y. (2020). An attribute-based encryption scheme with multiple authorities on hierarchical personal health record in cloud. The Journal of Supercomputing, 76(7), 4884–4903.

22. Deepa, N., & Pandiaraja, P. (2020). E health care data privacy preserving efficient file retrieval from the cloud service provider using attribute based file encryption. Journal of Ambient Intelligence and Humanized Computing, 12, 4877.

23. Mythili, R., Revathi, V., & Sai-Raj, T. (2020). An attribute-based lightweight cloud data access control using hypergraph structure. The Journal of Supercomputing, 76(8), 6040–6064.

24. Zhong, H., Zhou, Y., Zhang, Q., Yan, Xu., & Cui, J. (2020). An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare. Future Generation Computer Systems, 115, 486–496.

25. Mubarakali, A., Ashwin, M., Mavaluru, D., & Dinesh Kumar, A. (2020). Design an attribute based health record protection algorithm for healthcare services in cloud environment. Multimedia Tools and Applications, 79(5), 3943–3956.

26. Au, M. H., Yuen, T. H., Liu, J. K., Susilo, W., Huang, X., Xiang, Y., & Jiang, Z. L. (2017). A general framework for secure sharing of personal health records in cloud system. Journal of Computer and System Sciences, 90, 46–62.

27. Liu, C.-H., Chen, T.-L., Chang, C.-Y., & Zhen-Yu, W. (2021). A reliable authentication scheme of personal health records in cloud computing. Wireless Networks. https://doi.org/10.1007/s11276-021-02743-7

28. Liu, Y., Jia, Y., Jianxi, F., Vijayakumar, P., & Victor, C. (2021). Achieving privacy-preserving DSSE for intelligent IoT healthcare system. IEEE Transactions on Industrial Informatics., 18, 2010.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
A. Jain Austin did his B.E. degree in Computer Science Engineering from Anna University, Chennai in 2014. He is currently pursuing the M.E. degree in Computer Science Engineering from Anna University, Chennai. He is a research head in AJACO Softwares Pvt Ltd., Tamil Nadu, India. His interests include Cloud computing, Data mining, Machine Learning and Big Data etc.