Automated Software Engineering (2022) 29:4
https://doi.org/10.1007/s10515-021-00299-6

l–PEES-IMP: lightweight proxy re-encryption-based identity management protocol for enhancing privacy over multi-cloud environment

Sunitha Pachala[1,2] · Ch. Rupa[3] · L. Sumalatha[1]

Received: 10 May 2021 / Accepted: 13 September 2021 / Published online: 1 October 2021

Abstract
The user authenticity with proper identification is a significant challenge where the defects on the authenticity scheme can directly influence the sensitive data over the multi-cloud data (environment). It leads to severe information breaches and data loss over the cloud environment. Thus, a cloud user identity management protocol has to be designed in a secured manner using the proxy-encryption scheme, i.e., proxy transmits a cipher to another with a different encryption key by preserving the plain text secrets. Therefore, the intervention of third-party is avoided efficiently. This research provides an identity management protocol based on a proxy re-encryption scheme, an improved version of the existing identity management protocol, and named Lightweight Proxy re-encryption-based identity management protocol (*l*– PEES-IMP). It resolves the computational overhead that occurs during the encryption operation performed by the data owners and decryption due to asymmetric mode. It integrates symmetric and asymmetric encryption to establish secure communication. It is applied over the multi-cloud environment to develop privacy and security among sensitive data to avoid data loss or data breaching. It is also a trustworthy identity protocol for service providers and users. It addresses the problem related to the reliance on a third party, commonly identified in existing identity management protocol. Finally, the evaluation of the proposed PEES-IMP is done with existing ECC, RSA, hybrid model and EIDM, and various metrics to guarantee privacy and security of the data. The simulation is performed using MATLAB environment and shows better outcomes compared to prevailing approaches. This model is flexible which can be adopted practically. The encryption time of *l*– PEES-IMP is 0.819 ms, decryption time is 3.872 ms and re-encryption time is 28.18 ms which is better compared to other approaches.

Keywords Identity management protocol · Proxy re-encryption · Asymmetric · Symmetric · Computational overhead · Privacy · Security

✉ Sunitha Pachala
sunithamadhavi.patchala@gmail.com

Extended author information available on the last page of the article

1 Introduction

With the broader range of Internet evolution, many applications intend to fulfill the need of the users. Generally, people update themselves with the emergence of diverse mobile phone applications (Shaikh and Sasikumar 2013). These applications require some initial screening processes like account registration and password settings. It shows that people need to remember the credentials in a secured manner. However, there are various web-based tools provided in an in-built way, for example, cookies. It is precisely for security purposes (Khalil and Azeem 2014). Recently, the Internet-based companies model has launched various apps by launching the application with the app registration using the account generated by the Internet Service Providers (IPS). It is performed with the assistance of the Identity Management System (IMS) (Hanna 2018). It is an integrated version that occupies certain aspects like engineering, programming, and policies for facilitating the authorized resources for determining the user's identity in a precise manner and manages the users' information in the privacy-preservation way (Fan and Liu 2019).

The IMS utilizes an identity measurement policy to validate the identity; by establishing the functions for predicting the service provider to authorize the user's for using the services. In general, identity management has three diverse phases to handle users' personal information (Sun et al. 2018). They are: the information which is more transparent for both the service providers and the users, i.e., password; next, is the information that gives better understanding towards the identity management system and the user. It is validated with the Security Number (S.N.) (Maitra and K. Yelamarthi 2484). Finally, the verification with the identity measures like iris, user's fingerprint, etc. When it comes to the cloud environment, it is more transparent by establishing identity management in public, private, and hybrid clouds (Miao et al. 2019). The major transformation is provided with the concept of information sharing, which shows its significance towards the most dominating Internet companies by establishing the cloud-based product service (Yu et al. 2019). When it comes to the higher-end applications of cloud computing, the process of identity management is highly crucial. When identity management is not performed satisfactorily, the cloud users' or service provider needs to face huge loss (Yu et al. 2019). The cloud environment's computing process is the fusion of diverse computing processes extremely obscure. Thus, it leads to the complexity in both the hardware and software services. For the past few decades, cloud users have had a considerable increase, leading to the massive Identity Management technology (IMS) to offer privacy and security to the cloud environment.

The recent advancements over the cloud-based Identity Management System (IMS) are the advanced version of the conventional identity management system, which adopts massive innovative technologies like the signature model and user's security to facilitate the cloud for validating the user's legitimacy. Mohd et al. (Mohd and T. Hayajneh 2018) summarizes the conventional identity management system and includes various steps. At first, the user needs to log into the IMS with

essential credentials like username and password; next, the user needs to access the CSP request regarding the applications and data. Thirdly, the CSP needs to generate a request for a token from the user-end; fourthly, the user has to create a token request from IDM. Then, the tokens are generated for CSP and users'. The user makes use of token issued by the CSP for IDM. After receiving the token, the transactions need to be evaluated and compared. When the token comparison is performed successfully, the user is provided with the legalization certificate that makes the CSP visit the system legally (Mohd and T. Hayajneh 2018). Some conventional IMS possess diverse issues. For instance, various attacks are indirectly connected with the IMS server, which causes seizing or interception of IMS messages exchanged among the CSP and the users. Sometimes, the malicious attacker involves themselves in activities like loss, theft, or injecting the malicious code towards the mobile devices for capturing the user's data. For handling these issues, an effective cryptographic method needs to be adopted to improve the security of the users' information to achieve security and privacy (Fan and F. Liu 2019). The adoption of a better cryptographic model is exploited for offering some newer technologies. Owing to the nature of sure standardization and CSP structure decentralization, the applications are provided with a better solution for establishing trust.

Specifically, the concept of proxy re-encryption is introduced over the Identity Management System (IMS) for establishing a security enclave to fulfil the requirements of various applications and intends to attain superior performance than prevailing PRE schemes. The anticipated PRE assists in constant ciphertext size and decryption efficiency. However, there are some real-time scenarios with the average user to real-IMS. The IMS for certain social entities needs to ensure trust among the user-identity management system for cloud computing applications where the user's trust problem is higher than the reality. The work's objective is to handle the security issues caused by the user with extensive cloud server centralization. The target of this research is to model a lightweight algorithm to enhance the cloud security-based on proxy re-encryption. The work intends to offer an identity management system with proxy re-encryption scheme and models an improved version of the traditional IMS model. The re-encryption process provides complete data protection and moves the data securely over the complex environment. Also, it maintains the integrity. Therefore, the computational time is not that much higher than the encryption process. The significance of the work is listed below:

1) With the adoption of proxy- re-encryption and the anticipated identity management system facilitates the system to carry out identity authentication without any external influence and maintains the system reputation by eliminating the damage caused by the external factors.
2) A novel approach termed as Lightweight Proxy re-encryption-based identity management protocol ($l-$ PEES-IMP). It resolves the computational overhead that occurs during the encryption operation performed by the data owners and decryption.

3) It is also a trustworthy identity protocol for service providers and users. It addresses the problem related to the reliance on a third party commonly identified in the existing identity management protocol.
4) The evaluation is done by comparing the prevailing model's performance to project the significance of the anticipated $l-$ PEES-IMP model. The simulation is performed using MATLAB, and $l-$ PEES-IMP model shows better outcomes in contrast to the prevailing approaches.

The remainder of the work includes the following sections: Sect. 2 provides an elaborate discussion regarding the traditional IMS model and the level of security offered by them, along with the advantages and disadvantages. Section 3 explains the anticipated $l-$ PEES-IMP model concept extensively to achieve security and privacy preservation. In Sect. 4, the numerical results attained by evaluating the anticipated model are elaborated by measuring the computational complexities. Section 5 provides the conclusion with future research ideas.

2 Related works

This section provides an extensive review of the identity management system and proxy re-encryption. Symlin et al. (Salim and Sakurai 2011) present an analysis of the PRE variants. This model is extensively partitioned into two diverse stages like unidirectional and bi-directional. The former model includes conditional PRE, time-based, attribute-based, and identity-based PRE, while the latter model includes threshold-based PRE and type-based PRE strategy. The security properties include proxy re-encryption with original access, non-transferability, uni-directionality, non-interactivity, proxy invisibility, non-transitivity, key optimality collision resistance, which is broadly analyzed in Salim and Sakurai (2011). Based on the integration of these properties, some PRE models include various features modeled with the PRE evaluation, which is carried out based on these properties' occurrence.

Weng et al. (Weng et al. 2010) present an approach known as attribute-based PRE to offer access control towards the outsourced data by facilitating the ciphertext proxy transform with attributes to successive ciphertext from another attributes. Sun et al. (Sun et al. 2018) introduce the conditional PRE with ciphertext by fulfilling specific criteria that are transformed by the proxy server. Liang et al. (Liang et al. 2014) discuss the improved version of C-PRE for facilitating the ciphertext that the specified sender transforms for the proxy model. It provides the delegator with exclusive policies for authorizing the delegation. Chandran et al. (Chandran et al. 2014) explain the identity-based PRE with delegators' identity and ciphertext, which transforms the ciphertext under its identity. Phong et al. (Phong et al. 2016) anticipate IB-based PRE devoid of any random oracles.

Yao et al. (Yao et al. 2017) propose an improved version of IB-PRE for assisting the properties of conditional re-encryption that offer security among the identity and condition among the ciphertext attacks. Shi et al. (Shi and Fu 2015) discuss a type-based PRE in which every ciphertext delegate is merged with the proxy re-encryption, and type is measured if and only if the delegate shows some identical

measure over the public key. It assists the data owners in attaining a fine-grained delegation model. Egorov et al. (Egorov and M. Wilkison 2017) merge the certificate-based PRE and PRE with the certificate-based encryption model. It assists in providing resistivity towards the chosen adaptive ciphertext attacks. Kim et al. (He ge 2006) anticipates PRE scheme with keyword search. The ciphertext performs re-encryption with matching keyword and information associated with re-encryption. Bertino et al. (Poomagal and Sathish Kumar 2020) provide extensive analysis on broadcast PRE for facilitating the user A to user B the decryption property to the set of users. The anticipated model relies on conditional broadcasting PRE for dynamically addressing the users with a set of shared groups devoid of the necessity to vary the encryption public key (Kim and I. Lee 2018).

Moreover, some general approaches are used for establishing secure data sharing and communication over the cloud environment that suffers from certain constraints. The delay is measured among the user-generated request and response owing to the augmentation over the cryptographic functionality and outsourced data with number of connected users (Bertino et al. 2009). Chu et al. (Ateniese et al. 2006) discuss various security constraints, requirements, and threats over CC. With the analysis of prevailing models, it is observed that PRE for cloud computing is given with various technologies that are alike of conventional approaches. Liang et al. (Chow et al. 2010) discuss the cipher text-based policy attribute encryption for establishing the security access control to the encrypted data. Thus, it facilitates the owner to represent access policy over the universal attributes to perform ciphertext decryption. Further analysis is performed on (Chu and W.-G. Tzeng 2007) with the identity-based cryptographic model's adoption to secure communication against unauthorized users. The author discusses the ID-based PRE model with essential inputs. These approaches offer resistivity against the secret key leakage due to side-channel attacks over the cloud environment.

The crucial drawback associated with using ID-based cryptography, CP-ABE, and ABE models over cloud computing is the cost of computation during the decryption process. It includes various pairing functionality that is integrated with the policy complexities. However, the computing process consists of enormous concurrent and dynamic communication between the connected nodes. Due to the specific resource constraints in the computing environment, the adoption of conventional public key and critical management primitives for providing security during information exchange between the failed connected devices for assisting the computing process. Some prevailing cryptographic methods are computationally costly, and it does not fulfill the computing requirements (Xu et al. 2016).

It is noted that the lightweight cryptographic model is highly compatible while it operates in the cloud computing environment. Phong et al. (Liang et al. 2014) offer a PRE strategy on symmetric ciphers. The significant drawbacks are based on the individual secret key assumption with significant complexity during key exposure. Some approaches facilitate the trusted symmetric key distribution for offering secure communication. Dey et al. (Phong et al. 2016) provide a PRE scheme to deal with these constraints using a lightweight asymmetric encryption model. It is validated that the anticipated model offers an efficient outsourcing security model in a cloud computing environment. However, some shared outsourcing information with many

concurrent users will cause the encryption process with extreme computational load towards the proxy server (Dey and S. Weis 2010; Shao et al. 2011). Based on the above-analysis, it is observed that the existing approaches lacks in fulfilling the security at the end-level due to computational cost and lack of integrity. Similarly, the computing resources like processing power, software and hardware are used in an unauthorized manner. However, it introduces the computational cost for the proxy with added delay in the cloud objects' response time. Thus, the lightweight cryptographic model shows higher significance, and it is analyzed extensively.

3 Methodology

This section discusses the performance properties like security and privacy-preserving using the lightweight cryptographic model. The functionality of the anticipated Lightweight Proxy re-encryption-based identity management protocol (*l*− PEES-IMP) model is described with asymmetric and asymmetric cipher form for assisting security measures and performance. Cloud computing (CC) devices possess certain computational constraints, which is a significant factor for providing a better cryptographic model for CC. This process is termed lightweight cryptography. The adoption of a standard encryption model deals with the resources of the connected devices. These resources include energy, power consumption, memory size, and processing power. The lightweight model requires lesser resources and provides a better trade-off compared to the performance and security of the model. The significance of the model relies on waiting for time/latency, throughput, and power consumption. The higher-end version of the lightweight model is based on factors given below:

1) **Block size** the block size should be around 80 bit where the smaller key size leads to reduced power consumption and higher efficiency.
2) **Key size** the block size should be lesser than 80 bit as the smaller key size leads to reduced power consumption or higher efficiency.
3) **Number of rounds** the function carried out during every lightweight ciphers round are more straightforward than standard encryption ciphers. When the numbers of rounds are more significant, then it leads to performance degradation.
4) **Key schedules** With the provided key, the key scheduling process is used for computing the sub-keys for performing the round. Various encryption algorithms are used to attain higher feasibility with a secure model when identifying the external attacks. Moreover, it is essential to select an effectual encryption cipher in every aspect. Similarly, the encryption algorithm should offer appropriate protection against the injected attacks in a computing environment.

3.1 Lightweight Proxy re-encryption-based identity management protocol (*l*− PEES-IMP)

The system model for the anticipated Lightweight Proxy re-encryption-based identity management protocol (*l*− PEES-IMP) is used for constructing the security and

privacy preservation model. The anticipated $l-$ PEES-IMP model relies on proxy re-encryption facilitates trusted authority for forwarding data. Here, symmetric/asymmetric encryption ciphers are lightweight when both the process is accepted by the devices (lightweight). The system model is composed of a set of users connected with the connected nodes with unique identifiers. The users are provided with private and public keys. The set of cloud-connected nodes acts like a proxy server facilitating communication and connection among the servers and users. The anticipated model is analyzed over the distributed environment with the fully trusted authority who takes system parameters like connected users, cloud-connected nodes, and users' credentials (username and password). The trusted party does not involve it in any PRE. It is accountable for maintaining the secret keys. It comprises four diverse phases: key generation, encryption, re-encryption, and decryption (see Fig. 1). In the initial process, secret keys and system parameters are produced and transmitted to various parties. For specific functionalities, the user (delegator) communicates with the nearby devices and shares it with other users (delegate). The connected nodes communicate with trusted authorities for generating the re-encryption key and transfer to the proxy server securely by maintaining privacy. Data is transformed to both the users (secret key). In the encryption phase: there are two diverse phases, they are asymmetric and symmetric encryption. Here, a random integer is considered as symmetric cipher key. Then, it is used for message encryption with an asymmetric encryption cipher. This key is encrypted using asymmetric cipher and ready for transmission (including the ciphertext) to the targeted location. In re-encryption process, encrypted symmetric key is re-encrypted as another key cipher key form devoid of disturbing any attached data (message). Finally, the end-user needs to decrypt the re-encrypted key over the decryption process and recover the symmetric

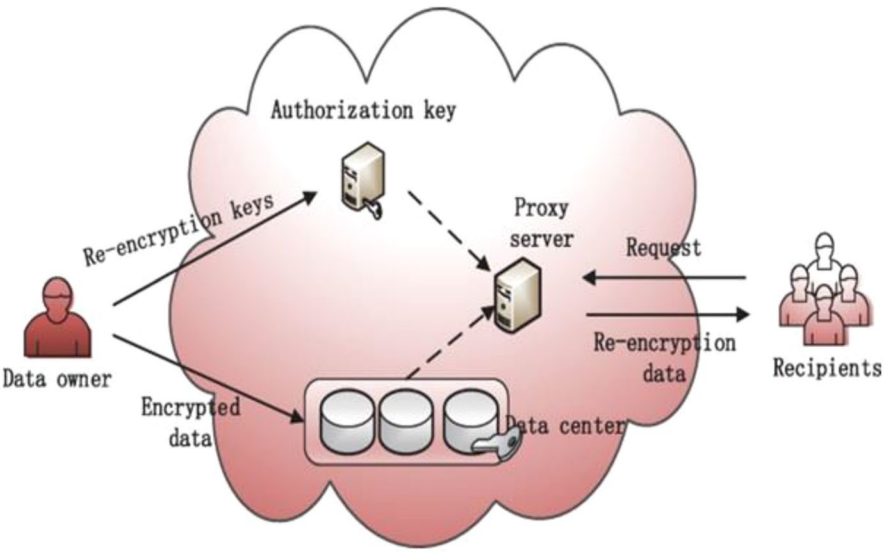

Fig. 1 Proxy re-encryption process

key utilized for ciphertext decryption and attaining the source message. Algorithm 1 depicts the functionality of the anticipated $l-$ PEES-IMP model.

Algorithm 1: l −PEES-IMP

a. Setup phase:
1) Input parameters $'n.'$
2) System parameters (E, p, q, e, G, s) //initialize $(1^n) \rightarrow SP$;
b. Key generation phase:
1) Input parameters $'SP^{.P.'}$
2) Secret key generation for users $'A'$ and $'B'$ (sk_A, sk_B) and key re-encryption $'rek'$ for the proxy server $//keygen\ (S).P. \rightarrow (sk_A, sk_B, rek)$
c. Encryption phase:
1) Message content $'m'$, public key pk of user $'A'$ is represented as SP_{PKA};
2) Generating ciphertext and cipher key (CT_A, CK_A); $//Encryption\left(m, SP_{.P.,PKA}\right) \rightarrow (CT_A, CK_A)$;
d. Re-encryption phase:
1) cipher key CK_A, S.P., and re-encryption key $'rek'$;
2) re-encrypted cipher key CK_{A1}; $//Re-encryption\ (CK_A, SP, rek) \rightarrow CK_{A1}$;
e. Decryption phase:
1) ciphertext $CT_A \rightarrow message\ content; re-encrypted\ cipher\ key \rightarrow CK_{A1};\ secret\ key\ of\ end-user \rightarrow sk_B$;
2) decrypt the original message content; $//decryption\ (CT_A, CK_{A1}, sk_B, SP) \rightarrow m$;

3.2 System model of Identity Management System (IMS)

The IMS system model is composed of CSP, set of users, data centre (D.C.) authority, and trusted party. The user has to register with the appropriate CSP and deploy over the cloud environment. The user transfers the data to the cloud environment with an authorized identity. The cloud system performs two diverse roles: D.C. authority and CSP. It represents CSP deployment where the interactions among the nodes are done with the CSP for verification. The legitimacy of the user's identity is verified along with the system token and the private key provided by D.C. The public and the private keys are provided to satisfy privacy and security. The latter model is used for specifying the CSP deployment and communication with the CSP for verification (See Fig. 2a and b). The hierarchical flow of the IMS is explained below:

1) The user has to generate a key and session request, which includes random integer values, IMS information, CSP information, and users' trust identity information). Then the key is used for encrypting the message content $'M'$ for generating $E(.K., M)$.
2) The user needs to register with the IMS model and log in to the system with proper user credentials. The user initiates transmitting ciphertext to the IMS and generates a request for attaining a token from the IMS for establishing authentication with the CSP.
3) With the received request, IMS intends to produce a token and transfer ciphertext and tokens to the CSP.
4) The CSP wants to fulfill specific security requirements (discussed in the section given below 'c').

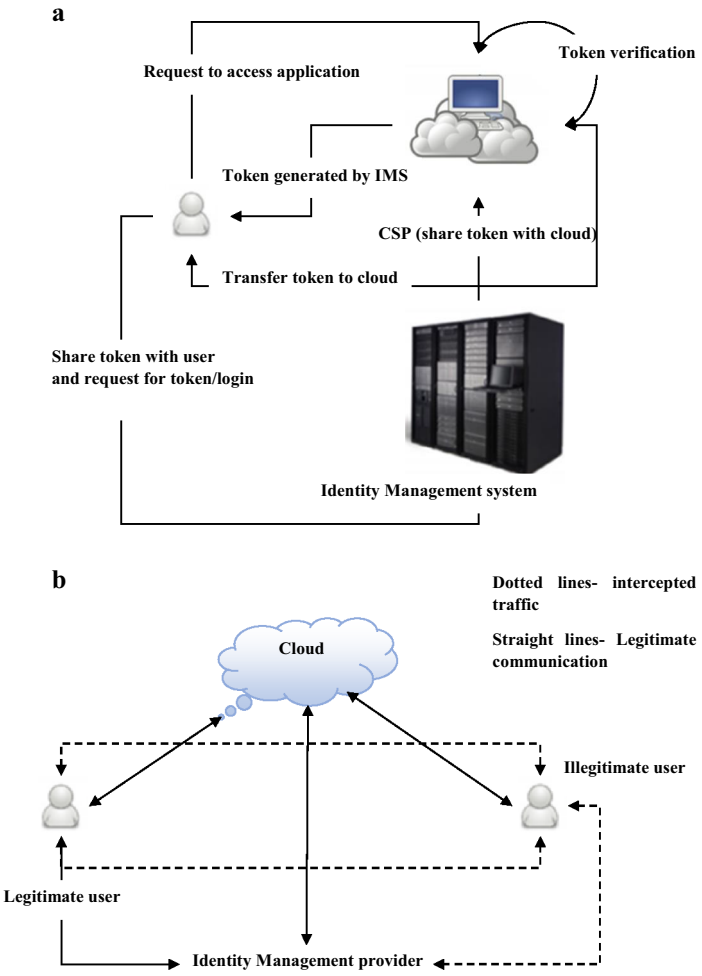

Fig. 2 **a** System model of IMS. **b** IMS communication process

5) The user transfers the encrypted message content and the public key to the CSP, i.e., fulfilling the security requirements.
6) CSP decrypts the key to meet the security requirements and uses the key for encrypting the message content and performing verification.
7) When the comparison is performed correctly, the user receives superior service from the CSP.

When compared to the conventional IMS model, the anticipated lightweight model integrated with IMS shows significant enhancements. For instance, it eliminates data duplication and avoids the threat from external sources. The data exchange has to be performed with a lesser size to achieve an efficient computational cost. However, some

flaws are indirectly connected with the system model as it cannot verify any particular kind of external attacks. When the CSP generates the legitimacy request towards the security model, the user plays the role to fulfill the request with the proper response. When the security measure is not fulfilled, the plain text is transmitted without privacy or security to the message content. Generally, the CSP performs the verification based on the distributed environment, and the reputation of the cloud environment is improved. A better design is retained with the anticipated ($l-$ PEES-IMP) and executed with the identity authentication. The provided model gives better usability and flexibility with proper security measures.

3.3 Security model for privacy preservation

Based on the lightweight model's significance, the computational complexity has to be lesser with the available resources. Generally, higher computational complexity is identified during the offloading process. The symmetric/asymmetric encryption and decryption of the message content are performed on the cloud-connected nodes. The proposed ($l-$ PEES-IMP) model is designed for reducing computational complexity. Simultaneously, the delegation is restricted during the re-encryption process with symmetric cipher key indeed of re-encrypting the complete message content. In setup phase, the key authority needs to run the algorithm with specific security measures as input and output (E, p, q, e, G, s). Here, $'q' \rightarrow$ *prime number*;$'p' \rightarrow$ *order of finite field*, $G \rightarrow$ *multiplication group of prime numbers*, $s \rightarrow$ *setupphase*. Over the key generation phase, the key generation authority runs with system parameters that include random numbers, pair of public and private key evaluation $s(pk, S.K.)$. The re-encryption key from secret key is evaluated using the public key. The returned value is provided with $\begin{pmatrix} SP \\ sk_A, sk_{PKA}, sk_{A1}, pk_{A1}, rek \end{pmatrix}$. Figure 3 depicts the graphical representation of the IMS security model.

The delegator (user A) needs to run the message $'m'$, system parameter $'S.P.,'$ and public key. The lightweight model is executed to encrypt the symmetric key and the message content. It holds the following steps: select symmetric key randomly with uniform distribution; the message content is partitioned into a set of blocks (smaller block size to reduce the computational complexity). The partitioned messages content (blocks) are encrypted for $'n'$ rounds with the generated symmetric key. The developed symmetric key functionality is provided as $f(k) \rightarrow P_k$, and the secret key is chosen randomly. Finally, P_k is encrypted with the public key P_{PKA} and returns the cipher key and ciphertext $\left(CK_A, CT_A\right)$. With the re-encryption process, the targeted node acquires the (rek, CK_A). Thus, it converts the CK_A into another form by re-encryption as CK_{A1}. Then, return CK_B(towards other delegators). With the final decryption process, the user receives the (CK_B, CT_A) from the proxy server. The secret key generated from the other delegate (sk_B) is use/d for decrypting and retrieving the symmetric key and the original message. It is mathematically expressed as in Eq. (1):

$$f^{-1}P_k = k \tag{1}$$

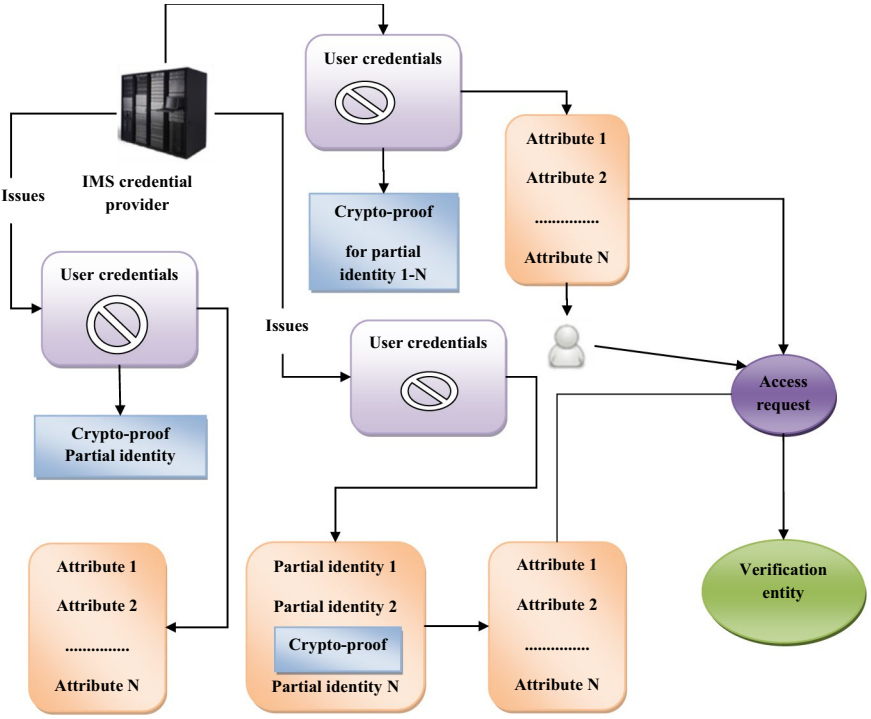

Fig. 3 IMS-based Security model

The ciphertext is decrypted using the proposed ($l-$ PEES-IMP), and it is specified as $dec\left(CT_A\right) = m$ respectively. Finally, return the message content $'m'$ The anticipated ($l-$ PEES-IMP) model's significance is validated by privacy preservation, confidentiality, and correctness. The encrypted message is generated from one user and moves to other user, maintaining correctness. It is based on the standardization of the cryptographic model for developing the nominal outcomes with appropriate keys. It deals with symmetric and asymmetric encryption and decryption with correctness. The correctness of the proposed ($l-$ PEES-IMP) model is asserted as in Eq. (2):

$$Decrypt\left(CT_A, k\right)m \tag{2}$$

The data confidentiality is measured based on the security fulfillment with proper private keys and system parameters. The trusted authority produces the secret key and system parameter. It correctly transmits the message to the user where the symmetric key is generated randomly to message encryption and asymmetric key. The nodes are provided in the trusted environment for performing the re-encryption process. During decryption, delegate identity is pre-defined using the authorized node. The end-user possesses a valid secret key and decrypts symmetric key and data decryption. Thus, the proposed model ($l-$ PEES-IMP) satisfies data confidentiality. The trusted authority produces the secret key with diverse users (See Fig. 4). The

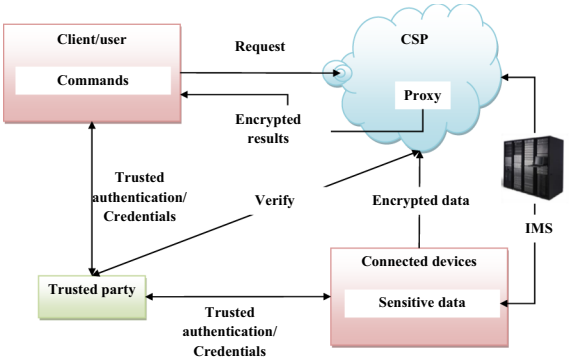

Fig. 4 Overall framework of Lightweight Proxy re-encryption-based identity management protocol (l–PEES-IMP)

secret key is generated using a random number with unique identifiers. The illegal users are unable to attain the decryption key.

4 Performance evaluation

This section helps to analyze the computational complexity of the proposed (l– PEES-IMP) model. Here, simulation is performed with a MATLAB environment. The experimentation is performed to evaluate the time needed for the encryption, decryption, and re-encryption process. PC is equipped with an Intel Core i3 processor, 3.3 GHz, 4 GB RAM, and Windows 7 OS where the computational time for encrypting and decrypting the proposed (l– PEES-IMP) model is compared with the standard RSA, ECC, and Hybrid Lightweight Proxy Re-Encryption algorithms. It is observed that the proposed (l– PEES-IMP) model shows lesser execution time when compared with symmetric encryption and decryption of RSA and ECC over diverse data sizes. Figure 5 shows the graphical model of encryption and decryption process of the image files taken from the private storage. The original input file is taken from the private cloud storage for performing the lightweight proxy re-encryption process. The original file is encrypted and decrypted with any loss and breaches.

Table 1 depicts the comparison of encryption time for various file sizes, i.e., 1 K.B., 100 KB, and 1000 KB. The evaluation is performed among the prevailing standard models like hybrid lightweight proxy re-encryption, ECC, RSA, and the proposed l– PEES-IMP. The proposed l– PEES-IMP model consumes significantly lesser time for encrypting the given input when compared to the other models. The encryption time was measured in milliseconds, i.e., it takes 0.000371, 0.000374, and 0.000375 ms for encrypting the image file of sizes 1 K.B., 100 KB, and 1000 KB, respectively. It is 0.0176, 0.0116, and 0.0046 ms lesser than the other models for 1 K.B.; similarly, for 100 KB, the encryption time is 0.7596, 0.4096, 0.1476 ms lesser than the prevailing models. While in the case of 1000 KB, the encryption time is 0.000375, which is 3.2396, 1.9796, and 0.5216 ms lesser than the hybrid, ECC, and RSA model (See Fig. 6).

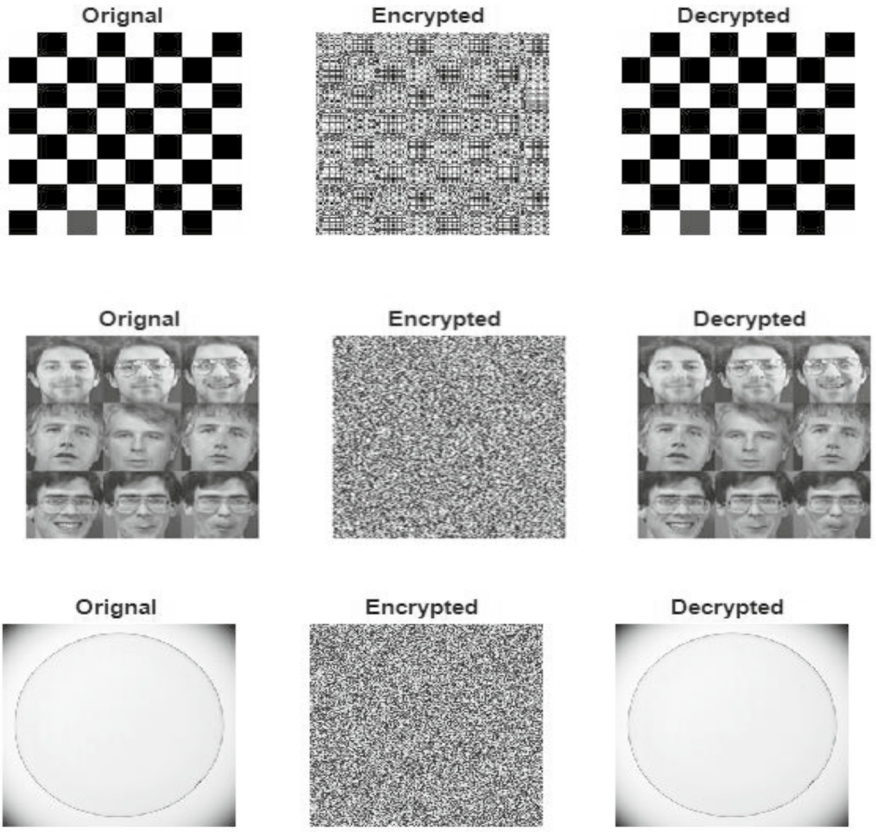

Fig. 5 *l*–PEES-IMP for image file

Table 1 Comparison of encryption time of the given image files

File size	1 KB	100 KB	1000 KB
Hybrid Lightweight Proxy Re-Encryption (Miao et al. 2019)	0.018	0.76	3.24
ECC (Poomagal and Sathish Kumar 2020)	0.012	0.41	1.98
RSA(He ge 2006)	0.005	0.148	0.522
l– PEES-IMP (ours)	0.000371	0.000374	0.000375

Table 2 depicts the comparison of decryption time for various file sizes, i.e., 1 K.B., 100 KB, and 1000 KB. The evaluation is performed among hybrid lightweight proxy re-encryption, ECC, RSA, and *l*– PEES-IMP. Here, *thel*– PEES-IMP model consumes significantly lesser time for decryption when compared to the other models. The decryption time was measured in milliseconds, i.e., the proposed model takes 0.001739, 0.001761, and 0.001796 ms for decrypting the image file of sizes 1 K.B., 100 KB, and 1000 KB, respectively. It is 0.02126,

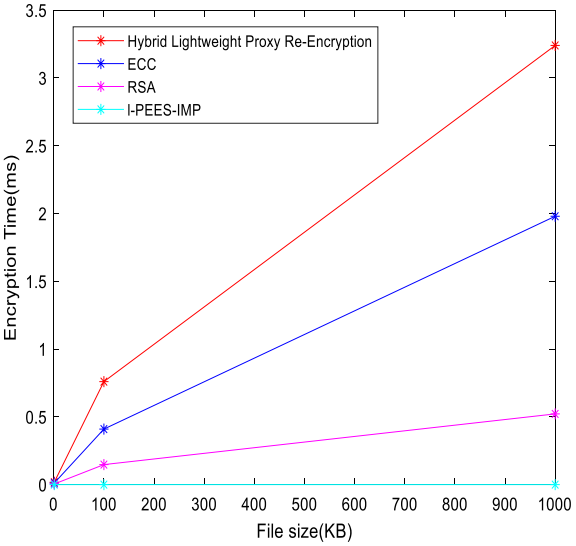

Fig. 6 Graphical representation of Encryption time (ms)

Table 2 Comparison of decryption time of the given image files

File size	1 KB	100 KB	1000 KB
Hybrid Lightweight Proxy Re-Encryption (Miao et al. 2019)	0.023	0.857	3.877
ECC (Poomagal and Sathish Kumar 2020)	0.016	0.523	2.33
RSA (He ge 2006)	0.006	0.219	0.735
l–PEES-IMP (ours)	0.001739	0.001761	0.001796

0.000022, and 0.004261 ms lesser than the other models for 1 K.B.; similarly, for 100 KB, the decryption time is 0.85523, 0.5212, and 0.2172 ms lesser than the prevailing models. While in the case of 1000 KB, the decryption time of *l*– PEES-IMP is 0.001796, which is 3.8752, 2.3282, and 0.7332 ms lesser than the hybrid, ECC, and RSA model (See Fig. 7).

Table 3 depicts the comparison of re-encryption time for various file sizes, i.e., 1 K.B., 100 KB, and 1000 KB. The evaluation is performed among hybrid lightweight proxy re-encryption, ECC, RSA, and *l*– PEES-IMP. Here, the *l*– PEES-IMP model consumes significantly lesser time for decryption when compared to the other models. The re-encryption time was measured in milliseconds, i.e., the proposed model takes 12.08, 13.95, and 14.76 ms for re-encrypting the image file of sizes 1 K.B., 100 KB, and 1000 KB, respectively. It is 22.7, 166.88, and 182.13 ms lesser than the other models for 1 K.B.; similarly, for 100 KB, the re-encryption time is 21.26, 318.34, and 577.96 ms lesser than the prevailing models. While in the case of 1000 KB, the re-encryption time of *l*– PEES-IMP is 14.76, 21.18, 983.84, and 1743.77 ms lesser than the hybrid, ECC, and RSA model (See Fig. 8). This analysis observed that the anticipated *l*– PEES-IMP

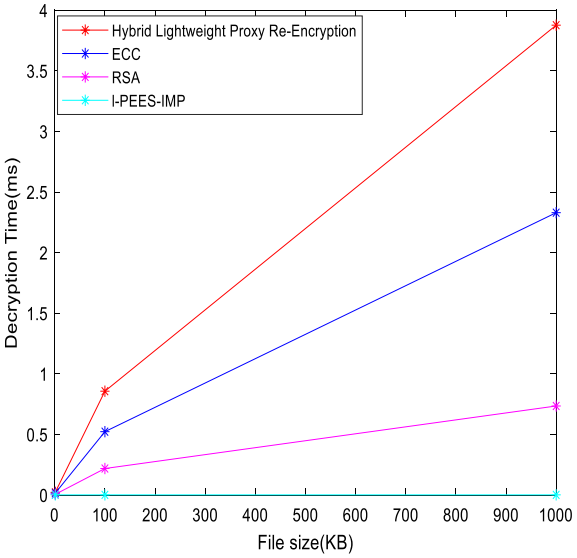

Fig. 7 Graphical representation of decryption time (ms)

Table 3 Comparison of re-encryption time of the given image files

File size	1 KB	100 KB	1000 KB
Hybrid Lightweight Proxy Re-Encryption (Miao et al. 2019)	34.78	35.21	35.94
ECC (Poomagal and Sathish Kumar 2020)	178.96	332.29	998.63
RSA (He ge 2006)	194.21	591.86	1758.53
l–PEES-IMP (ours)	12.08	13.95	14.76

model consumes less time for encryption, decryption, and re-encryption of data over the proxy server.

Table 4 depicts the comparison of encryption, decryption, and re-encryption time of various approaches like EIDM, PRE, hybrid model, and ICN, where the proposed *l*– PEES-IMP model shows the consistently lesser time for computation which gives a better trade-off among the other models. Figure 9 shows the graphical representation of the execution time evaluation. Similarly, Table 5 shows the key generation time of EIDM, PRE, hybrid model, ICN, and *l*– PEES-IMP models. The time taken for the key generation is 15.91 ms, 22.56 ms, 19.22 ms, 139.66 ms, and 21 ms, respectively, where *l*– PEES-IMP consumes less time. The EIDM and hybrid model's key generation time is lesser than an *l*– PEES-IMP model for 5.09 and 1.78 ms. However, the proposed *l*– PEES-IMP model shows a significantly lesser time when compared to PRE and ICN models, i.e., 1.56 ms and 118.66 ms, respectively (See Fig. 10). The encryption time of *l*– PEES-IMP is 0.819 ms which is 19.951 ms, 16.011 ms, 9.821 ms, and 22.491 ms lesser than EIDM, PRE, hybrid lightweight proxy re-encryption and ICN. The decryption time of *l*– PEES-IMP is 3.872 ms

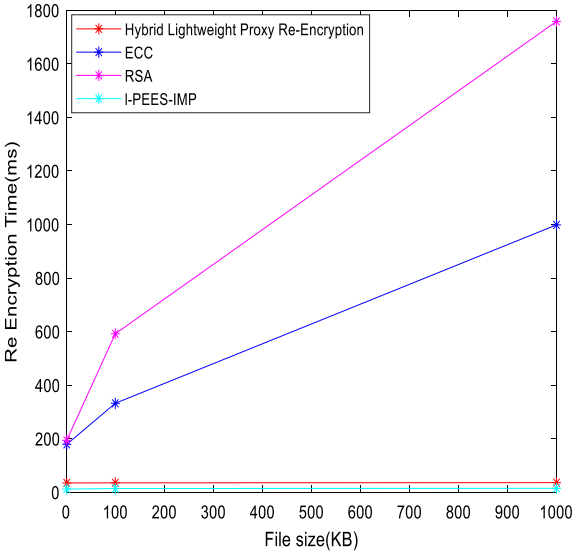

Fig. 8 Graphical representation of re-encryption time (ms)

Table 4 Comparison of execution time of proposed l-PEES-IMP with existing approaches

Algorithms	Encryption time	Decryption time	Re-encryption time
EIDM (Shi and Fu 2015)	20.77	33.91	33.90
PRE (Sun et al. 2018)	16.83	17.43	32.67
Hybrid Lightweight Proxy Re-Encryption (Miao et al. 2019)	10.64	11.09	108.32
ICN (Egorov and M. Wilkison 2017)	23.31	14.28	90.09
l– PEES-IMP (ours)	0.819	3.872	28.18

which is 30.038 ms, 13.558 ms, 7.218 ms, and 22.491 ms lesser than EIDM, PRE, hybrid lightweight proxy re-encryption and ICN. The re-encryption time of *l*– PEES-IMP is 28.18 ms which is 5.72 ms, 4.49 ms, 80.14 ms, and 61.91 ms lesser than EIDM, PRE, hybrid lightweight proxy re-encryption and ICN.

The proposed (*l*– PEES-IMP) model's efficiency is evaluated to show the performance efficiency, and the internal cloud environment generates the necessary keys and transfers them to the end-users (nodes). Then, the re-encryption process is performed for sharing the ciphered key—the time needed by the cloud for generating and transferring the keys to the other party. From the observed results, it is noted that the time for symmetric encryption and decryption linearly increases when the size of message content increased. However, the time needed for asymmetric encryption and decryption is constant with the fixed key length of 128 bits for all sized messages. The bit rate of considered for evaluation is 1 KB, 100 KB and 1000 KB; however, the data transmission process is not restricted to this level. The bit rate can be higher with MB and GB. But, there are changes in the execution time of encryption, decryption and re-encryption.

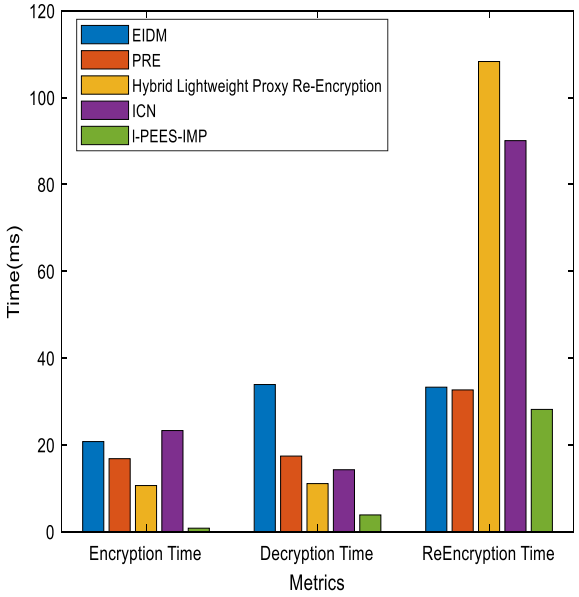

Fig. 9 Graphical representation of execution time (ms)

Table 5 Key generation time

Algorithms	Re-key generation time
EIDM (Shi and Fu 2015)	15.91
PRE (Sun et al. 2018)	22.56
Hybrid Lightweight Proxy Re-Encryption (Miao et al. 2019)	19.22
ICN (Egorov and M. Wilkison 2017)	139.66
l–PEES-IMP (ours)	21

4.1 Discussion

The computation time required for encryption and decryption process is evaluated using some standard methods like RSA and ECC. It is known that the execution time for encryption and decryption (symmetric) over diverse data size is lesser than encryption and decryption (asymmetric) using RSA which is longer for data of same size. The computation time for key-generation is required for generating and sending keys to other nodes. From the results, it is known that the time is increased linearly during symmetric encryption and decryption process in proportional to the message size. Similarly, the asymmetric encryption and decryption time is constant and the key size is fixed as 128 bits. This work compares RSA, ECC, and hybrid model as these model shows consistency during the process of evaluation. Table 1, 2, 3, 4, 5 depicts the efficiency during encryption and decryption with the specification of

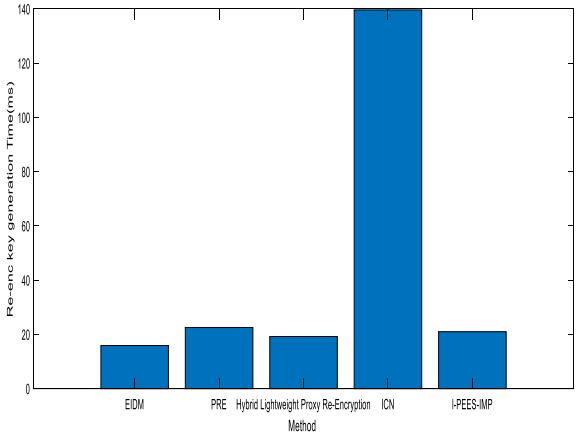

Fig. 10 Graphical representation of key generation time (ms)

total data/unit time. The anticipated *l*–PEES-IMP of varied data size is compared with standard ciphers of ECC, RSA and hybrid model. The proposed model outperforms the existing ECC, RSA, and hybrid model during encryption and decryption process. The complexity of the proposed *l*–PEES-IMP model is analysed based on the execution time, i.e. encryption, decryption and re-encryption time. The model consumes lesser time while performing all these process compared to other approaches.

Table 3 shows the computation time of re-encryption process. It is observed that the proposed *l*–PEES-IMP model shows average time during re-encryption process. There is some significant variation in the re-encryption time among the proposed *l*–PEES-IMP, hybrid, RSA, and ECC model. These models are extensively analyzed with proxy re-encryption process. The user credentials are re-encrypted with private key and specifically provided for the individual users. The faster execution time during re-encryption process explains that *l*–PEES-IMP model does not re-encrypt the complete input data where the symmetric key is re-encrypted with proxy re-encryption process. The trust is build using the crypto-proof model by reducing the overheads, cost, and reduce the need for the third party. When the complete message is re-encrypted then it leads to computational overhead and delay in processing. Tables 1, 2, 3 shows the comparison for data size of 1 KB, 100 KB, and 1000 KB. However, Tables 4, 5 shows the comparison of existing models with various file size. Table 5 shows the comparison of encrypted IDM, PRE, ICN (information-based network model for re-encryption), and hybrid lightweight model with proposed *l*–PEES-IMP. These models show higher significance during the data sharing process among the users and these are directly connected with proxy re-encryption process. However, ECC, RSA models are some of the traditional encryption and decryption models. Thus, the significances of the anticipated *l*–PEES-IMP are finer compared to other models. The end-users are benefitted with this process with reduced computational complexity and computational cost. Therefore, multiple devices attain security during data processing over complex environment. The lightweight model possesses

the functionality of enabling the application of secure encryption with reduced resource utilization. Some other metrics related to the lightweight algorithm are memory, resource computation, and less power supply. However, this work concentrates only on encryption, decryption and re-encryption process.

5 Conclusion

In the cryptographic concept, the proxy re-encryption process is determined to be a powerful tool. The functionality of proxy re-encryption helps to re-encrypt the provided ciphertext to other forms. There is diverse proficiency that is related to proxy re-encryption. However, the conventional model fails in handling all the characteristics while functioning over the real-time scenario. Therefore, this work attempts to improve proxy re-encryption functionality by integrating the concept of the identity management system (IMS). This model intends to reduce the computational complexity with reduced block size and the number of rounds.

This research provides an improved version named Lightweight Proxy re-encryption-based identity management protocol (l–PEES-IMP) which resolves the computational overhead that occurs during encryption and decryption. It integrates symmetric and asymmetric encryption to establish secure communication. The improved version shows the consistency of the model while functioning over the cloud environment. It also attains specific metrics like correctness, privacy, data confidentiality, and so on. This model tries to overcome the drawbacks identified in the traditional approaches. The model helps the ciphertext transform from one form to another with the re-encryption concept's adoption. The simulation is performed using MATLAB and the outcome shows better trade-off while comparing with other models. The proposed (l–PEES-IMP) is applied over the multi-cloud environment to establish privacy and security among the sensitive data to avoid data loss or data breaching. Finally, the evaluation of the proposed PEES-IMP is done with existing CIMP, EIDM, and various metrics to guarantee privacy and security of the data. The encryption time of l–PEES-IMP is 0.819 ms, decryption time is 3.872 ms and re-encryption time is 28.18 ms which is better compared to other approaches.

The limitations faced while modeling the proposed (l–PEES-IMP) is the lack of evaluation with the real-time cloud environment. With the simulation setup, the private cloud is set over the P.C., and the assessment is performed. This simulation environment provides better results; however, the model significance needs to be validated with real-time cloud-like (Amazon). Some other drawbacks like computational cost analysis need to be done in the future. In the future, an attempt will be made to achieve this limitation. The future research direction includes the analysis of the proxy re-encryption with authentication protocols like SAML.

Declarations

Conflict of interest The authors declared that there is no conflict of interest.

References

Ateniese, K., Fu, M.G., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Sec. (TISSEC) **9**(1), 1–30 (2006)

Bertino, et al.: Privacy-preserving digital identity management for cloud computing. Bullet. IEEE Comput. Soc. Tech. Committ. Data Eng. **32**(1), 21–27 (2009)

Chandran, M. Chase, F. Liu, R. Nishimaki, and K. Kagawa, "Reencryption, functional re-encryption, and multi-hop re-encryption: A framework for achieving obfuscation-based security and instantiations from lattices," vol. 8383, pp. 95–112, (2014).

Chow, J. Weng, Y. Yang, and R. H. Deng, "Efficient unidirectionalunidirectional proxy re-encryption," in International Conference on Cryptology in Africa. Springer, (2010), pp. 316–332

Chu and W.-G. Tzeng, "Identity-based proxy re-encryption without random oracles," in International Conference on Information Security. Springer, (2007), pp. 189–202.

Dey, Weis, S., "PseudoID: Enhancing privacy in federated login," in Hot Topics in Privacy Enhancing Technologies, (2010), pp. 95–107.

Egorov and Wilkison M., "Nucypher kms: Decentralized key management system." arXiv: Cryptography and Security, (2017)

Fan and Liu F., "Proxy re-encryption and re-signatures from lattices," pp. 363–382, (2019).

Fan X., Liu, F. "Proxy re-encryption and re-signatures from lattices," pp. 363–382, (2019).

Hanna, ``Systems and methods for an incremental, reversible and decentralized biometric identity management system,'' U.S. Patent 10 078 758 B1, Sep. 18, (2018).

He GE, "An Anonymous Authentication Scheme for Identification Card ", Int. Conf on information and communication security, pp. 238–248, (2006).

Khalil, A.K., Azeem, M.: Consolidated identity management system for secure mobile cloud computing. Comput. Netw. **65**(2), 99–110 (2014)

Kim, S., Lee, I.: IoT device security based on proxy re-encryption. J. Ambient Intell. Human. Comput. **9**(4), 1267–1273 (2018)

Liang, C., Chu, X., Tan, D.S., Wong, C.T., Zhou, J.: Chosen-ciphertext secure multi-hop identity-based conditional proxy re-encryption with constant-size ciphertexts. Theoret. Comput. Sci. (2014). https://doi.org/10.1016/j.tcs.2014.04.027

Maitra, S., Yelamarthi, K.: Rapidly deployable IoT architecture with data security: implementation and experimental evaluation. Sensors **19**(11), 2484 (2019)

Miao, J., Ma, X., Liu, J., Weng, H.L., Li, H.: Lightweight fine-grained search over encrypted data in fog computing. IEEE Trans. Services Comput. **12**(5), 772–785 (2019)

Mohd, B. J., & Hayajneh, T. (2018). Lightweight block ciphers for IoT: energy optimization and survivability techniques. IEEE Access, 6, 35966-35978.

Phong, L. Wang, Y. Aono, M. H. Nguyen, and X. Boyen, "Proxy re-encryption schemes with key privacy from lwe." IACR Cryptology ePrint Archive, vol. 2016, p. 327, (2016).

Poomagal, C.T., Kumar, G.S.: ECC based lightweight secure message conveyance protocol for satellite communication in internet of vehicles (IoV). Wirel. Person. Commun. **113**(2), 1359–1377 (2020)

Salim, T.N., Sakurai, K., "Realizing proxy re-encryption in the symmetric world," in International Conference on Informatics Engineering and Information Science. Springer, (2011), pp. 259–274.

Shaikh. R., Sasikumar.M.: ``Identity management in cloud computing,'' Int. J. Comput. Appl., vol. 63, no. 11, (2013).

Shao, Wei G., Ling, Y., and Xie, M., "Identity-based conditional proxy re-encryption," in 2011 IEEE International Conference on Communications (ICC). IEEE, (2011), pp. 1–5.

Shi, R.X., Fu, A.M.: Multi-element based on proxy re-encryption scheme for mobile cloud computing. J. Commun. **36**(11), 73–79 (2015)

Sun, M., Ge, C., Fang, L., Wang, J.: A proxy broadcast re-encryption for cloud data sharing. Multim. Tools Appl. **77**(9), 455–469 (2018)

Sun, C., Ge, L.F., Wang, J.: A proxy broadcast re-encryption for cloud data sharing. Multim. Tools Appl. **77**(9), 10455–10469 (2018)

Weng, M., Chen, Y., Yang, R., Deng, K.C., Bao, F.: CCA secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles. Sci. Chin. Inf. Sci. **53**(3), 593–606 (2010)

Xu, P., Jiao, T., Wu, Q., Wang, W., Jin, H.: Conditional identity-based broadcast proxy re-encryption and its application to cloud email. IEEE Trans. Comput. **65**(1), 66–79 (2016)

Yao, W., Zhang, Y., Qian, H., Han, J.: Flexible and fine-grained attribute-based data storage in cloud computing. IEEE Trans. Serv. Comput. **10**(5), 785–796 (2017)
Yu, R., Chen, H., Li, Y.L., Tian, A.: Toward data security in edge intelligent IIoT. IEEE Netw. **33**(5), 20–26 (2019)

Authors and Affiliations

Sunitha Pachala[1,2] · Ch. Rupa[3] · L. Sumalatha[1]

Ch. Rupa
rupamtech@gmail.com

L. Sumalatha
sumalatha.lingamgunta@gmail.com

[1] Department of CSE, JNTU College of Engineering, Kakinada, Andhra Pradesh, India

[2] Department of CSE, Dhanekula Institute of Engineering and Technology, Vijayawada, Andhra Pradesh, India

[3] Department of CSE, Velagapudi Ramakrishna Siddhartha Engineering. College, Knuru, Vijayawada, Andhra Pradesh, India