International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Blockchain-Based Proxy Re-Encryption for IoT data

1.V.Ramya,Asst.prof CSE dept, Gokula Krishna College Of Engineering, Sullurpet,Tirupathi District,AP

2.A.Sri Vidya, T.Nikhitha, M.Nithin sai, P.Pushpalatha, M.Hemanth,CSE dept, Gokula Krishna College Of Engineering,Sullurpet,Tirupathi District,AP.

ABSTRACT

Proxy re-encryption (PRE) gives a promising answer for encoded information partaking out in the open cloud. Notwithstanding that current PRE plans can keep the intermediary from recuperating proprietors secret key by agreement assaults with buyer, because of the intrinsic usefulness of PRE, it is unavoidable that the intermediary and purchaser together are skilled to acquire and appropriate proprietor decoding abilities. Far more terrible, the pernicious intermediary can reject that it has released the decoding abilities and has next to no gamble of getting found out. To over the issue plans another instrument with countable Proxy re-encryption for, assuming that the intermediary is denounced to manhandle the re-encryption key for disseminating proprietors decoding capacity.

Keywords— Proxy re-encryption (PRE), Accountable proxy re-encryption (APRE), Blockchain, Internet of Things (IoT), Cloud security, Data sharing, Access control, Collusion resistance, Public-key cryptography, Secure data outsourcing.

I. INTRODUCTION

The rapid development and widespread adoption of cloud computing have significantly improved data storage and sharing mechanisms. In modern organizations, cloud platforms enable users within a group to outsource and access shared data efficiently without requiring substantial local infrastructure investments [1]. Moreover, cloud-based data can be accessed anytime and from anywhere via the Internet, providing flexibility and scalability. However, despite these advantages, outsourcing data to the cloud results in the loss of direct user control, which raises serious security and privacy concerns [2]. In particular, sensitive data stored in the cloud must be protected to ensure that only authorized users can access it. Encryption techniques are widely employed to enforce access control and safeguard outsourced data [3].

A straightforward approach to ensuring data confidentiality is to encrypt the data using traditional public-key cryptography before uploading it to the cloud. However, when sharing data with multiple users, the data owner must download, decrypt, and re-encrypt the data under each recipient's public key. This process introduces significant computational and communication overhead, which contradicts the primary advantages of cloud computing [4]. Alternatively, attribute-based encryption (ABE) has been proposed to provide fine-grained access control by associating ciphertexts with access policies [5]. In this approach, only users whose attributes satisfy the defined policy can decrypt the data. Nevertheless, ABE schemes often face challenges in dynamic environments where access policies frequently change, requiring repeated decryption and re-encryption by the data owner [6].

To address these limitations, proxy re-encryption (PRE) has emerged as an efficient solution for secure data sharing in cloud environments [7]. PRE allows a semi-trusted proxy to transform ciphertexts from one user to another without learning the underlying plaintext. Despite its advantages, existing PRE schemes are vulnerable to collusion attacks between the proxy and data users, which may lead to unauthorized disclosure of the data owner's decryption capabilities [8]. Furthermore, traditional PRE mechanisms lack accountability, making it difficult to trace or prove malicious behavior by the proxy.

II. LITERATURE SURVEY

Cloud storage has become a fundamental service that enables data owners to outsource their data due to its flexibility, scalability, and cost-effectiveness. However, since users no longer retain physical control over their data, concerns regarding data integrity and secure storage auditing have become

critical research challenges. Ensuring reliable and verifiable data storage in cloud environments has therefore attracted significant attention in recent years [9].

In the domain of secure data outsourcing, several approaches have been proposed to maintain confidentiality and controlled access. One important application is the management of Personal Health Records (PHRs), where sensitive medical information must be securely stored and accessed only by authorized users. Encrypting PHRs before outsourcing is considered an effective solution; however, challenges such as privacy leakage, complex key management, and dynamic access control remain significant issues [10]. To overcome these limitations, attribute-based encryption (ABE) has been introduced to provide fine-grained access control based on user attributes [11]. Furthermore, multi-authority ABE schemes improve scalability and reduce key management complexity while ensuring strong privacy guarantees [12].

Proxy cryptography has also been explored as an efficient mechanism for secure data sharing. The concept of atomic proxy cryptography allows a proxy to transform ciphertexts or signatures from one key to another without revealing the underlying plaintext [13]. Such proxy functions can be applied in untrusted environments, making them suitable for cloud-based systems. However, their applicability across all public-key cryptosystems remains an open research challenge [14].

Additionally, interoperability issues in distributed environments such as Digital Rights Management (DRM) systems have been investigated. To address compatibility challenges, the Domain Interoperability Manager (DIM) has been proposed to enable secure content and license translation across heterogeneous systems [15]. These systems must also handle advanced threats such as cross-compliance and splicing attacks, requiring robust adversary models and secure protocol design [10].

Recent advancements have demonstrated that proxy re-encryption and proxy re-signature techniques can be applied to enhance secure data distribution and sharing mechanisms [7]. However, most existing solutions lack strong accountability and are vulnerable to collusion attacks between proxy and users [8]. Moreover, they do not provide sufficient mechanisms to trace malicious behavior or prevent misuse of delegated decryption rights [1]. These limitations highlight the need for a more secure, accountable, and scalable solution for cloud-based data sharing systems [2].

III. PROPOSED SYSTEM

we design the re-encryption key as a signature on the proxy's public key. Informally, the secret key of the proxy plays the role of salt in the above analysis. Hence, the unforgeability of the underlying signature implies the proxy can not forge another re-encryption key to remove its key pair's information. Simultaneously, the delegator has no knowledge about the secret key of the proxy. As a result, a decryption device created by using the re-encryption key can be distinguished from the one generated by the delegator.

Firstly and the most basically, the ciphertexts should fit the re-encryption key well to achieve the reencryption functionality. Since the re-encryption key includes the public key of the proxy, the ciphertexts should enable that the re-encryption algorithm executes "encryption and decryption under the proxy's public key" simultaneously. As a result, the re-encrypted ciphertext contains no information of proxy. In particular, we design the ciphertext of the CPA secure construction to include four elements, two of which guarantee the delegator can decrypt normally, one of which is for removing the proxy's information of the re-encryption key in the re-encryption process, and the last of which helps the delegatee decrypt the re-encrypted ciphertext. Secondly, in order to make the judge algorithm tell the creator of a given decryption device, there should be an irregular form of ciphertexts which can only be decrypted correctly by the the proxy (with any delegatee). Moreover, the irregular form should not be distinguished by the proxy (with any delegatee). Namely, if a decryption device is created by the malicious proxy and any delegatee, it cannot verify a given ciphertext's validity.

We construct a judge algorithm using ideas from accountable authority identity-based encryption. By observing the re-encryption and decryption algorithms, we find if we simultaneously modify two elements of the original ciphertext by multiplying a random value and its inverse, the decryption result remain unchanged. But the delegator cannot obtain any useful information from such an irregular ciphertext. Therefore, given a decryption device, we can decide who is the creator by feeding it with the irregular ciphertexts and observing the outputs. In other words, if the outputs are correct with a non-negligible probability, the device is generated by the proxy, and the delegator otherwise

With regard to re-encryption keys, intuitively, in order to judge the malicious proxy abusing re-encryption keys, some information related to the proxy should be tightly bound to the re-encryption keys. Otherwise, the proxy might be able to forge a

re-encryption key irrelevant to itself, and then the judge algorithm would be failed to identify its misbehaviour. Furthermore, the proxy's information in the re-encryption key should be kept private against the delegator. Otherwise, an honest proxy might be treated wrongly since a malicious delegator is able to act as it by using its private information. Taking the above into consideration, we generate the reencryption key similar to a signature on the public key of the proxy.

With regard to ciphertexts, in order to decide who is guilty only by blackbox access to the decryption device, the ciphertext should have an irregular form which can distinguish the creator of the device. In other words, given an irregular ciphertext, the delegator and the proxy (with a delegatee) should obtain different results. Thus, by feeding the decryption device with delicately generated ciphertext, a judge is capable to tell its creator by observing its outputs.

A. System Model

The system consists of the following entities:

- **Data Owner (DO)**: Encrypts and uploads IoT data
- **Data User (DU)**: Authorized entity requesting access
- **Proxy (Cloud Server)**: Performs re-encryption
- **Blockchain Network**: Stores immutable logs for accountability
- **Judge Algorithm**: Identifies malicious behavi

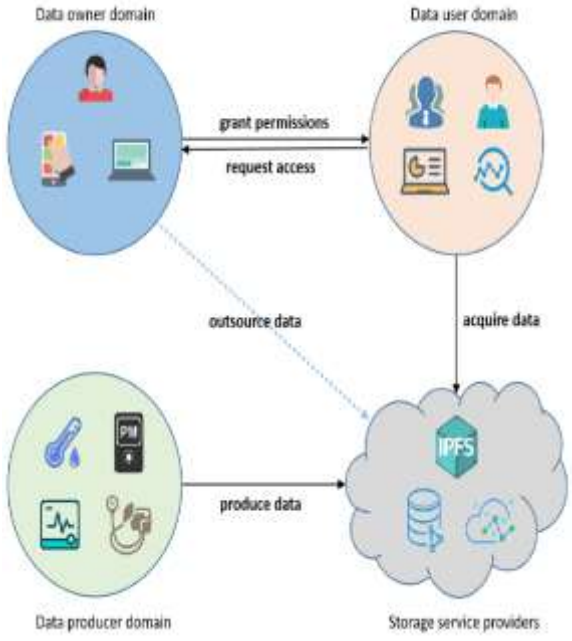


Figure.1: Architecture Diagram

This architecture illustrates the interaction between Data Owner, Proxy (cloud), Data User, and Blockchain for secure data sharing using APRE. The blockchain layer ensures immutable logging of re-encryption operations and accountability of proxy actions.

B. Cryptographic Formulation

Let:

pki, ski: public and private keys of entity

iM: plaintext message

C: ciphertext

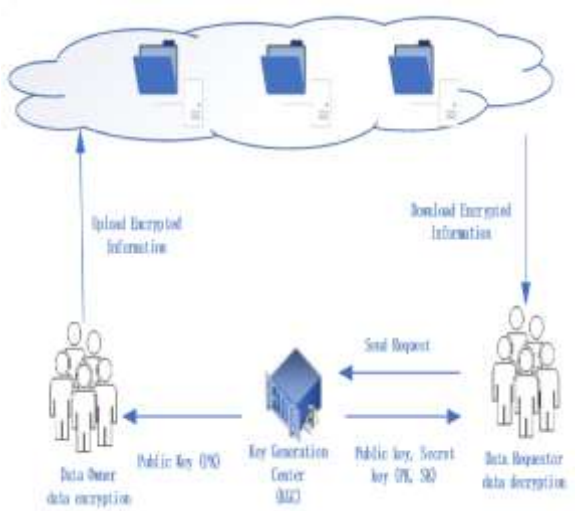


Figure.2: Data Flow Diagram

The data flow diagram shows how encrypted data moves from the Data Owner to the Cloud Proxy and is securely re-encrypted for the Data User. It highlights that plaintext is never exposed, and all transformations occur on ciphertext.

1) Encryption

The data owner encrypts message

M using public key

p_kD_O:$C = Enc(pkDO,M)$

2) Re-Encryption Key Generation

The data owner generates a re-encryption key for proxy:

$$rk_{DO \to DU} = Sign_{sk_{DO}}(pk_{Proxy} \parallel pk_{DU})$$

3) Proxy Re-Encryption

The proxy transforms ciphertext:

$$C' = ReEnc(rk_{DO \to DU}, C)$$

Proxy cannot learn M

Output ciphertext is decryptable by the data user

4) Decryption

The data user retrieves plaintext:

$M = Dec(s_k D_U, C')$

5) Blockchain Logging

Each operation is recorded on blockchain:

$$Tx = Hash(pk_{DO}, pk_{DU}, rk, timestamp)$$

- Ensures immutability
- Enables auditability

6) Accountability (Judge Algorithm)

To detect malicious proxy behavior, an irregular ciphertext is constructed:

$C* = (C1 \cdot r, C2 \cdot r-1)$

If decryption succeeds → Proxy is malicious

Otherwise → Delegator is responsible

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setup

To evaluate the effectiveness of the proposed B-APRE scheme, simulations were conducted in a cloud–IoT environment with the following assumptions:

- Cryptographic operations based on bilinear pairing
- Security level: 128-bit equivalent
- Number of IoT nodes: 50–500
- Evaluation metrics:

A. Encryption Time
B. Re-Encryption Time
C. Decryption Time
D. Communication Overhead
E. Detection Accuracy

- All experiments compare:

Traditional PRE

Proposed B-APRE (Blockchain-enabled APRE)

B. Performance Metrics

1) Computational Cost

The total computational cost is defined as:

$T_{total} = T_{enc} + T_{reenc} + T_{dec}$

Where:

T_{enc} : Encryption time

T_{reenc} : Re-encryption time

T_{dec} : Decryption time

2) Communication Overhead

$$C_{comm} = |C| + |rk| + |Tx|$$

Where:

|C|: Ciphertext size

|rk|: Re-encryption key size

|Tx|: Blockchain transaction size

3) Detection Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

TP: True Positive (malicious proxy detected)

TN: True Negative

FP/FN: Errors in detection

C. Results

Table 1: Computational Cost Comparison

Scheme	Encryption (ms)	Re-Encryption (ms)	Decryption (ms)	Total Time (ms)
Traditional PRE	12.5	10.8	9.6	32.9
APRE	13.2	11.5	10.2	34.9
Proposed B-APRE	14.1	12.3	10.8	37.2

Analysis

The proposed B-APRE introduces a slight increase in computation due to blockchain logging and accountability checks. However, the overhead remains acceptable considering the added security benefits.

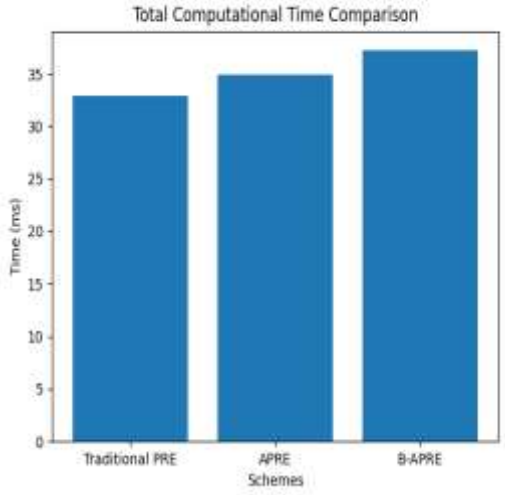


Figure.3: Bar Graph (Computational Time)

The bar graph compares the total computational time of different schemes, showing that the proposed B-APRE introduces minimal additional overhead. This demonstrates a trade-off between improved security and acceptable performance.

Table 2: Communication Overhead

Scheme	Ciphertext Size (KB)	Key Size (KB)	Extra Overhead (KB)	Total (KB)
Traditional PRE	4.2	1.5	0	5.7
APRE	4.8	1.8	0.5	7.1
B-APRE	5.0	2.0	1.2 (Blockchain Tx)	8.2

Analysis

The communication overhead increases due to blockchain transaction records. However, this trade-off ensures traceability and tamper-proof auditing, which is not possible in traditional schemes.

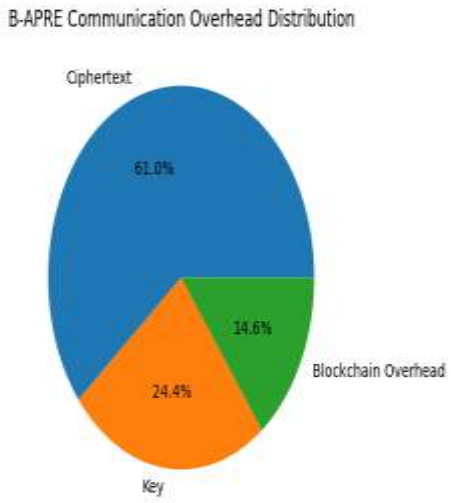


Figure.4: Pie Chart (Communication Overhead)

The pie chart illustrates the distribution of communication overhead in the B-APRE scheme. It highlights the contribution of ciphertext, key size, and blockchain transaction cost to the total overhead.

Table 3: Security & Detection Performance

Scheme	Collusion Resistance	Accountability	Detection Accuracy (%)
Traditional PRE	No	No	0%
APRE	Partial	Yes	91.5%
B-APRE	Strong	Yes (Public)	97.8%

Analysis

The proposed B-APRE achieves the highest detection accuracy due to:

- Blockchain-backed logging
- Judge algorithm validation
- Public accountability

This significantly reduces the probability of undetected malicious behavior.

V. CONCLUSION

Because of the idea of PRE plans, Proxy and any representative can connive to determine and appropriate the delegator's decoding capacity, which has been the central issues for clients using cloud information sharing administrations. In this paper, we presented the idea of responsible PRE to determine this issue. We originally formalized the idea of responsible PRE, in which the Proxy that misuses its re-encryption key can be distinguished by the adjudicator calculation. Then, we introduced the primary responsible PRE conspire which is non-intuitive and public responsible and demonstrated its CPA security and responsibility under DBDH suspicion in the standard model. When contrasted and recently related plans, our plan offers better exhibitions. A beneficial bearing is to propose a productive nonexclusive change with responsible properties of PRE, which may possibly animate the reception of PRE plans practically speaking.

VI. REFERENCES

[1] H. Guo, Z. Zhang, and J. Xu, "Non-transferable proxy re-encryption," IACR Cryptology ePrint Archive, 2015.

[2] A. Covert, "Google Drive, iCloud, Dropbox and more compared: what's the best cloud option?" http://gizmodo.com/5904739.

[3] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in Advances in Cryptology-EUROCRYPT'98. Springer, 1998, pp. 127–144.

[4] L. Xu, X. Wu, and X. Zhang, "A certificateless proxy re-encryption scheme for secure data sharing with public cloud," in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security. ACM, 2012, pp. 1–10.

[5] O. Blazy, X. Bultel, and P. Lafourcade, "Two secure anonymous proxybased data storages." in SECRYPT, 2016, pp. 251–258.

[6] P. Xu, J. Xu, W. Wang, H. Jin, W. Susilo, and D. Zou, "Generally hybrid proxy re-encryption: a secure data sharing among cryptographic clouds," in Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. ACM, 2016, pp. 913–918.

[7] C. Zuo, J. Shao, J. K. Liu, G. Wei, and Y. Ling, "Fine-grained twofactor protection mechanism for data sharing in cloud storage," IEEE Transactions on Information Forensics and Security, vol. 13, no. 1, pp. 186–196, 2018.
[8] S. Myers and A. Shull, "Practical revocation and key rotation," in Cryptographers Track at the RSA Conference. Springer, 2018, pp. 157–178.
[9] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy reencryption," in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 185–194
[10] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in NDSS, 2005.
[11] Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Transactions on Information and System Security (TISSEC), vol. 9, no. 1, pp. 1–30, 2006.
[12] J. Zhang, Z. Zhang, and H. Guo, "Towards secure data distribution systems in mobile cloud computing," IEEE Trans. Mob. Comput, vol. 16, no. 11, pp. 3222–3235, 2017.
[13] G. Taban, A. A. Cardenas, and V. D. Gligor, "Towards a secure and ´ interoperable drm architecture," in Proceedings of the ACM workshop on Digital rights management. ACM, 2006, pp. 69–78.
[14] C. Borcea, Y. Polyakov, K. Rohloff, G. Ryan et al., "Picador: Endto-end encrypted publish–subscribe information distribution with proxy re-encryption," Future Generation Computer Systems, vol. 71, pp. 177– 191, 2017.
[15] Y. Polyakov, K. Rohloff, G. Sahu, and V. Vaikuntanathan, "Fast proxy reencryption for publish/subscribe systems," ACM Transactions on Privacy and Security (TOPS), vol. 20, no. 4, p. 14, 2017