Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: Crossref, ROAD & Google Scholar

5

Review of Encryption Techniques to Enhance Data Protection in Cloud

Aniket Pattanshetti[1], Shounak Sugave[2] and Balaso Jagdale[3]

[1,2,3] Department of Computer Science and Engineering,

MIT World Peace University, Pune, India

Abstract

Cloud computing has become an essential source of data storage for both companies and individuals, making the protection of sensitive information increasingly necessary. The paper explores ways to protect data in the cloud through encryption, including symmetric, asymmetric, and hybrid encryption. Large-sized datasets can be handled quickly and effectively using symmetric encryption, like AES, while asymmetric encryption, like RSA, provides higher security using key pairs. Hybrid encryption combines these two methods, balancing speed and security for better protection. Advanced techniques like Proxy Re-Encryption (PRE) and Attribute-Based Encryption (ABE) help manage access control and scalability in cloud environments. This study examines the mentioned encryption methods, identifying challenges and proposing advancements to improve encryption solutions for cloud storage.

Keywords: Cloud Security, cryptography, data protection, confidentiality

1. Introduction

In recent years, data consumption has reached new heights as the number of online users has grown beyond comprehension. The legacy computing framework proved expensive and challenging to manage, resulting in low data availability. Hence, a separate storage system has

become a requisite for retaining data. Despite their server capabilities, conventional systems struggle to accommodate increasing users on online networking sites, social media, multimedia sharing, etc. As the World Wide Web has gradually expanded to every corner of the globe, the growing demand for service provisioning has resulted in a new form of computing called cloud computing. It is a pool of connected computers that enable processing power, data, and other resources on a shared basis anytime without needing to maintain infrastructure. The National Institute of Standards and Technology (NIST) [1] defines cloud computing as, "a model for enabling convenient, resource pooling, ubiquitous, on-demand access which can be easily delivered with different types of service provider interaction". Modern-day information technology infrastructure is inconceivable without using cloud computing. It has changed how individuals and organizations save, process, and access data. Data storage has shifted from traditional hard drives to cloud computing, which offers virtualization resources and cost-effective storage solutions. With the leading cloud service providers today, the integration of cloud services has expanded to numerous activities including data hosting and high-end processing services of enterprises. This trend is crucial because it encourages businesses to enhance their processes, collaborate, and innovate. Cloud storage has various merits that promote its popularity. The foremost advantage is the availability of the data. Users can access the information they need from anywhere, enabling remote work and allowing a scattered team to collaborate effectively. One notable advantage is its cost-effectiveness. With cloud storage, making unnecessary large changes and upfront investments in hardware and basic infrastructure is not required. Hence, one pays only for the storage or resources required, allowing for effective budget management. Another important aspect of cloud storage is its scalability. Eventually, most organizations face the need to increase the capacity of their storage resources without facing the challenges of the storage systems that were used previously. This aspect is crucial for companies whose data load increases drastically or fluctuates. Cloud storage provides data redundancy and data backup. Major cloud providers store customer data across multiple geographical regions to improve durability and protect against data loss caused by hardware failures or natural disasters. Such redundancy improves data accessibility while simultaneously reducing the workload of the institution's backup system.

From the standpoint of storage service, there are three major types of storage: block, file, and object type [2]. Block type offers a straightforward model by providing storage as fixed-size blocks, enabling low-latency access. File type organizes data in a hierarchical structure of files and folders making it effective in shared access scenarios. In contrast, the object type is designed for scalability, storing data as objects with unique identifiers, which is ideal for handling unstructured data. When it comes to data management on a PB scale, massive datasets create many issues for organizations; for instance, embracing cost-effective storage infrastructure, enabling data interoperability and transmission quickly, and even protecting the risk of losing data. They do not entirely address the essential needs of the enterprise when it comes to protecting and managing mass data storage in terms of many factors such as reliability, availability, security, and so on.

Cloud storage offers easy-to-use interfaces, the ability to grow as needed, and ways to measure resources. It includes many layers [3], summarized as follows: 1) The storage layer consists of devices and a central oversight system. 2) The primary management layer is the main and most challenging part of cloud storage. 3) The application interface layer is the most bendable part of cloud storage. 4) The access layer, handles data computation, user authentication, and authorization to ensure only the right users interact with the storage. Given the nature of cloud storage, which involves distributed data and remote access, safeguarding data security and privacy is critically important.

The primary security concerns of cloud storage [4] fall into three key categories: confidentiality, integrity, and availability (CIA). Protecting classified sensitive data or information from unauthorized access or disclosure is known as confidentiality. Integrity ensures that data remains unchanged in form and content over time. Availability is making sure that the data in the cloud is accessible at any time and from anywhere. Usually, cloud service providers achieve high availability by introducing redundancy, fail-over mechanisms, and regular data backup. For the concern of confidentiality, data encryption before sending it to the cloud storage is the easiest way [5]. Encryption ensures that even if data is accessed maliciously, the malicious entity cannot read without the decryption key. There are different types of cryptographic techniques available that enable data encryption.

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: Crossref, ROAD & Google Scholar

1.1 Symmetric Encryption

Symmetric encryption can be considered one of the most ancient and simplest. For both encryption and decryption, a single shared key is used. This technique is accepted because of its high speed and effectiveness, especially when the amount of information being encrypted is huge, and this includes cases such as entire databases or storage volumes. Currently, AES (Advanced Encryption Standard) is the most commonly used symmetric encryption algorithm for cloud storage. This algorithm is in widespread use due to its provision for enough security while at the same time enhancing efficiency. The advantage of symmetric encryption is that it can process huge quantities of data without using too many computational resources, which makes it perfectly suited to conditions such as cloud storage where time is of the essence. For example, it is common for cloud service providers to use the advanced encryption standard (AES) scheme to protect information while at rest on the disks and in motion between servers or between the user and cloud systems, respectively. Encryption at rest protects data from being compromised if physical storage devices are stolen or accessed. Similarly, encryption while transmitting data makes it impossible for anyone to intercept and access data, even when sent over the internet or other networks. However, symmetric encryption poses a significant challenge: key management. This is because the same key is used to encrypt and decrypt the data. Hence, the processes of distributing and storing the key, have to be very secure. If an intruder obtains the key, they will decrypt the information. This risk is heightened in cloud environments, where keys may need to be shared among users or across multiple servers, increasing the likelihood of attackers compromising the keys.

1.2 Asymmetric Encryption

Symmetric encryption is speedy but may not be so secure owing to the keys that must be shared. That is where the need for asymmetric encryption arises. Asymmetric encryption involves a public key and the other one is a private key. Data is encrypted using a public key and decrypted using a private key. The good thing about this system is that the public key can be shared with virtually everybody without any risk of the secured information being compromised since only the person who has the private key can get such information decrypted. It is because it is more

involved and time-consuming than its counterpart symmetric encryption, which is why it is not preferred for direct encryption of large quantities of data. Instead, it is principally employed in the encryption of small-sized data, for example, encryption keys in hybrid systems or for establishing a secure line of communication between clients and servers at the initial connection setting.

1.3 Hybrid Encryption

In cloud settings, it is very common to use an amalgamation of symmetric and asymmetric encryption to help minimize the weaknesses of both forms of encryption. In a hybrid model, symmetric encryption encrypts the data because it is much faster and more efficient. Following this step, the symmetric encryption key is transmitted over a secure channel using asymmetric encryption. In this way, symmetric encryption is used without compromising the security of the key exchange process.

The topic of encryption techniques used to secure cloud storage is very important in today's digital landscape. The efficient handling of large-scale, dynamic systems is studied across domains, such as web crawling, where incremental and personalized strategies ensure relevance and efficiency [6]. Similarly, our study aims to analyze the existing encryption strategies and propose novel methods to meet the demands of scalability and user-centric performance.

2. Existing Work

In the paper [7], the authors have attempted to improve data security within the cloud by proposing a hybrid model of cryptography using AES for encryption and DES for decryption. AES provides strong security and high-speed performance. In contrast, DES was preferred for decryption because it is less complex and useful in many applications. The work provides a performance analysis of DES and AES and the effect of data volume on the speed of encryption and decryption of the data, noting the increase in the time taken for encryption and decryption with an increase in data volume. The system protects the data by ensuring its confidentiality,

integrity, and authentication. A key limitation noted is the increasing data processing delay with larger volumes, which could be a challenge in scaling this system.

The system proposed in [8] presents a multilevel hybrid cryptography model combining both symmetric (DES) and asymmetric (RSA) encryption algorithms. DES was used for data storage security, and RSA enabled safe data transmission. They found out that the hybrid offered better security; hence, users and cloud providers trusted each other more than before. File upload and download times were lower than previously developed systems. However, the paper does not speak of scalability as data volumes increase, nor performance concerns relating to using DES and RSA concurrently. The authors suggest future work to improve the hybrid approach and address these performance concerns.

This paper [9] addresses security challenges in cloud-based medical data storage by proposing a hybrid cryptographic scheme that combines AES, Enhanced Honeypot (EH) Algorithm, SHA3 hashing, and OTP (One-Time Password) techniques. The goal of this method is to safeguard health information from potential cyber-attacks, principally by providing data integrity and privacy. The medical data to be protected undergoes some preprocessing stage at the onset using Enhanced Principal Component Analysis or EPCA model to remove redundant information and salient data. Moreover, data classification is done where the Adaptive AlexNet CNN classifier classifies the data into trained classes to assist in intrusion or security threat detection. In the hybrid cryptography systems employed in this study, data encryption is done using file security standard Encryption AES, which is highly secure. At the same time, records in tables are protected using the SHA3 hashing algorithm. Enhanced Honeypot Algorithm also enables network traffic monitor to detect any possible intrusion.

The work done in [10] involves a hybrid encryption method that merges the Advanced Encryption Standard (AES), Blowfish (BF), and the Message-Digest algorithm (MD5). It consists of splitting the data file into three parts, encrypting each part using a different algorithm, and finally combining them for cloud storage. Experimental results show that employing such a hybrid solution cuts down the encryption and decryption speed by around 33% as compared to the available solutions while decreasing the amount of storage needed.

The HCAC-EHR: Hybrid Cryptographic Access Control for Secure EHR Retrieval in Healthcare Cloud [11] presents a methodology that uses the Improved Key Generation Scheme of RSA (IKGSR) for encrypting electronic health records (EHR) and the Blowfish algorithm for encrypting the keys generated by IKGSR. This approach enhances security by incorporating four large prime numbers during key generation, following which steganography-based access control is integrated where the secret keys are hidden within images to avert unauthorized usage. The framework supports effective substring indexing and keyword searching of the encrypted information, while a security risk management analysis using the DREAD operational threat modeling technique provides evidence of the framework's resistance to several attack types as well. These findings portray that this HCAC method can enhance both the security aspect and retrieval effectiveness when compared to other existing hybrid cryptographic methods in use, indicating its viability as a base for future healthcare data security and integrity improvement research.

In the work of [12], the authors present a hybrid cryptosystem methodology that employs an amalgamation of the Advanced Encryption Standard (AES) and the Elliptic Curve Diffie-Hellman (ECDH) for securing and optimizing the storage of data in the cloud. ECDH key management is equipped at the client side, which first requests a file from the server, and then the server encrypts the requested file using AES based on the short-time ECDH-derived key. This allows for avoiding the long encryption time that naturally arises with using only AES in such scenarios and provides adequate protection, ensuring that the user data existing in the cloud is private and accessible without the concern of being attacked by users with evil motivations.

The encryption algorithm design discussed in [13] utilizes the strengths of symmetric and asymmetric architectures, which involve AES, DES, and m-RSA algorithms. To start the encryption process, the original data is divided into three parts. The first part is encrypted using AES, the second part employs DES, and the final part utilizes m-RSA. The resultant ciphertext, which is a sum of three ciphertexts, is transmitted to the intended recipient. In the decryption process, the message is cut into three, and the respective algorithms are used to decrypt the parts. The final message is the same as the original data entered by the user.

The authentication protocol articulated in [14] is a security protocol for RFID systems that uses symmetric and asymmetric cryptography. In this case, a data block is encrypted through the Advanced Encryption Standard (AES) algorithm. The key used in the AES algorithm is encrypted using Elliptic-Curve Cryptography (ECC) during transmission from the tag to its corresponding central server. Rabin, ECC, RSA algorithms, and AES key lengths and security efficacies are analyzed in the paper, where it was established that AES could provide the same level of security with shorter keys, therefore, it is optimal for providing mutual authentication in RFID systems.

The encryption schema that is under consideration in [15] uses a combination of symmetric (AES-128) and asymmetric (RSA), even though the focus is mainly on data security for contact tracing. It relies on RSA to provide a public key primarily to encrypt a symmetric key and a private key that is used in creating digital signatures. It involves hashing the data using SHA-256, which provides a 256-bit hash. The hashed data is divided into two blocks, each 128 bits long; one serves as the AES-128 key for data encryption, while the other is protected by RSA encryption using the public key. A digital signature is created when half of the hash not used for encryption is hashed together with the RSA private key. This method improves security in that it guarantees confidentiality and data integrity for the use of a single unique key to prevent misuse.

File storage on the cloud server is made safe using symmetric key encryption with Shamir's secret-sharing approach as proposed in the model [16]. To begin with, a symmetric key is sought to encrypt the file, which produces a form of text called ciphertext. The conversion of this ciphertext into several parts or shares, via Shamir's Secret Sharing (SSS) distribution in which shares are formed by a polynomial with random coefficients. Lagrange interpolation determines the key using a specified range of shares, which are then used to modify the ciphertext. The shares can be kept in distinct partitions across different cloud servers so that the service provider can do homomorphic operations on the data without needing to access the original data and protect it.

This study [17] employs a research methodology focused on developing a secure data-sharing system through Proxy Re-Encryption (PRE). In this method, a proxy assists the re-encryption of previously generated encrypted data. When the data user requests access, the proxy re-encrypts the existing ciphertext using a new re-encryption key generated for that specific request. It creates a new ciphertext which only the intended receiver can decrypt. The methodology includes detailed phases such as key generation, re-encryption key generation for creating new ciphertext, and a two-phase decryption process that ensures only verified individuals can access the original data.

The method put forward in [18] presents a Direct Revocation scheme. The process begins with the data owner encrypting his data with an access list (P1) and uploading the resulting ciphertext C1 to the cloud storage. Whenever Alice wants to make the encrypted content accessible to others, she forms a re-encryption key, which enables the cloud server to transform C1 into another ciphertext C2 meant for users whose policy P2 is less restrictive than that of C1. The cloud service stores the ciphertext and performs re-encryption and revocation. In this case, revocation applies to users who conform to the terms of policy P2, but not an even stricter policy P3. The intended recipients decrypt the re-encrypted ciphertext using their private keys.

The approach discussed in [19] uses the Paillier Homomorphic Encryption and the Blowfish algorithm to provide data security in the multi-cloud environment. The client-side encryption uses Paillier HE, followed by server-side encryption with Blowfish, ensuring that data remains confidential even if cloud providers are compromised. The system has a Role-Based Access Control mechanism. Data integrity is achieved through hash functions such as MD5, SHA-1, and SHA-256. Performance analysis indicated that due to data compression, the hybrid encryption scheme reduced the time taken in both encryption and decryption with obviously lowered memory costs. Entropic measures of security proved that using both algorithms was secure enough.

The following table presents a comparative study of these approaches to clarify further the advantages and limitations of the various encryption techniques discussed.

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: Crossref, ROAD & Google Scholar

Table 1. Comparative Study of Encryption Techniques

Paper	Techniques Used	Strengths	Limitations
[7]	AES and DES	High security and speed	Increased processing delay with larger data volumes
[8]	DES and RSA	Improved security and transparency	Computational load increases with data size
[9]	AES, Enhanced Honeypot, SHA3, OTP	Strong file security, intrusion detection	Complex preprocessing steps may affect performance
[10]	AES, Blowfish, MD5	Reduces encryption/decryption times by approximately 33%.	Performance bottlenecks from multiple algorithms
[11]	IKGSR and Blowfish	Enhances security with a steganography-based access control mechanism	Complexity of key management with many algorithms
[12]	AES and ECDH	Reduced encryption time, robust security	Dependency on key management efficiency
[13]	AES, DES, m-RSA	Fast encryption through parallel processing.	Not suitable for large datasets.
[14]	AES and ECC	Strong security and efficient key exchange.	Slower ECC operations may affect the performance
[15]	AES-128 and RSA	Enhanced security with digital signatures	High computational complexity

[16]	AES and Shamir's Secret Sharing	Allows homomorphic operations without accessing original data	Share management can complicate recovery
[17]	Proxy Re-Encryption (PRE)	Suitable for cloud due to increased scalability	Reliance on a trusted authority, single point of failure.
[18]	CP-ABPRE-DR	Allows secure sharing with direct revocation of access, enhancing control over permissions.	Potential latency in re-encryption.
[19]	Paillier HE and Blowfish	Multi-cloud fragmentation and role-based access control.	Performance overhead in multi-cloud environments.

3. Discussion

Although there has been considerable advancement in providing security to cloud data, the existing literature on the subject shows substantial gaps and limitations that need further research and development to address. One of the central issues is that of scalability. Most of the present techniques, like those with AES, RSA, or other similar encryption techniques, require every data owner to encrypt each file separately for each of the allowed users. This becomes cumbersome and inefficient as users increase, especially in cases where there is a need to share data on large scales. To address this issue, the Proxy Re-Encryption (PRE) method is developed, allowing a proxy to handle the re-encryption process on behalf of the data owner. It resolves the scalability issue but introduces a new vulnerability called a single point of failure. Existing PRE methods depend on a single proxy, making the entire security mechanism vulnerable if the proxy is compromised. This is a major limitation in environments where data security and reliability are paramount. Therefore, there is a need for distributed or decentralized

approaches to re-encryption that do not place all trust in a single entity. Also, since systems keep growing, managing keys becomes difficult and resource-consuming.

1. Proposed Technique

Implementing a Key Management System (KMS) that is not centralized and uses blockchain technology would promote easier management of keys since they will be distributed thus reducing the risks of single-point failures in the following way.

- The Key is segmented into multiple pieces and kept at different nodes. Hashes of each share are recorded on the blockchain for verification.
- A user or an application calls a smart contract on the blockchain to access the key.
- The smart contract automatically triggers key rotation on a schedule to generate a new key and revoke the old one.
- Every transaction involving a key is recorded on the blockchain, to provide an indelible record for compliance purposes (for instance, to support GDPR, HIPAA, etc.)

4. Conclusion

In conclusion, this review paper analyzes various encryption techniques to secure data in cloud environments. It discusses the symmetric, asymmetric, and hybrid approaches offered for data protection and explains the merits and demerits of each. Hybrid systems that combine both encryption types mitigate the weaknesses of single encryption methods but often introduce complexity in implementation and performance bottlenecks. The paper also explored advanced methodologies such as Proxy Re-Encryption (PRE) and Attribute-Based Encryption (ABE), which provide scalable mechanisms crucial for cloud security. Nevertheless, obstacles such as single points of failure and dependence on central authorities persist as open issues. New approaches that share trust among multiple entities, or circumvent heavy computations while preserving security, are important for solving future challenges to provide robust cloud data protection architectures.

References

1. Mell P, Grance T (2018) SP 800-145, The NIST Definition of cloud computing | CSRC (online) Csrc.nist.gov. https://csrc.nist.gov/publications/detail/sp/800-145/final.
2. IBM. Block. Accessed: Feb. 2020. [Online]. Available: https://www.ibm.com/cloud/learn/block-storage
3. D. Zhe, W. Qinghong, S. Naizheng, and Z. Yuhan, "Study on Data Security Policy Based on Cloud Storage," 2017 IEEE 3rd international conference on big data security on Cloud (big data security) 2017, pp. 145-149, doi: 10.1109/BigDataSecurity.2017.12.
4. Hamed Tabrizchi and Marjan Kuchaki Rafsanjani. 2020. A survey on security challenges in cloud computing: issues, threats, and solutions. J. Supercomput. 76, 12 (Dec 2020), 9493–9532.https://doi.org/10.1007/s11227-020-03213-1
5. K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," in IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, Jan.-Feb. 2012, doi: 10.1109/MIC.2012.14.
6. S. M. Nakashe and K. R. Kolhe, "Smart Approach to Crawl Web Interfaces Using a Two-Stage Framework of Crawler," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-6, doi: 10.1109/ICCUBEA.2018.8697592.
7. G. Sudhakar, H. Azath, P. A. Priya M and P. B. Edwinprabhakar, "A Hybrid Cloud Security System using Cryptography," 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 12-16, doi: 10.1109/ICECAA58104.2023.10212352
8. S. Kumar, G. Karnani, M. S. Gaur and A. Mishra, "Cloud Security using Hybrid Cryptography Algorithms," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 599-604, doi: 10.1109/ICIEM51511.2021.9445377.

A. Priya, S. Saradha, (2021). Implementation of hybrid cryptographic schemes in a cloud environment for enhanced medical data security, International Journal Of Nonlinear Analysis And Applications, 12(2), 1785-1800. magiran.com/p2330058

9. Bermani, Ali & Murshedi, Tariq & Abod, Zaid. (2021). A hybrid cryptography technique for data storage on cloud computing. Journal of Discrete Mathematical Sciences and Cryptography. 24. 1-12. 10.1080/09720529.2020.1859799.
10. Ponnusamy, Chinnasamy & Deepalakshmi, P. (2022). HCAC-EHR: hybrid cryptographic access control for secure EHR retrieval in the healthcare cloud. Journal of Ambient Intelligence and Humanized Computing. 13. 1-19. 10.1007/s12652-021-02942-
11. K. Negi, R. Shrestha, T. L. Borges, S. Sahana, and S. Das, "A Hybrid Cryptographic Approach for Secure Cloud-Based File Storage," 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), London, United Kingdom, 2023, pp. 1-7, doi: 10.1109/GlobConET56651.2023.10150148.
12. Pooja, and R. K. Chauhan. 2020. “Triple Phase Hybrid Cryptography Technique in a Wireless Sensor Network.” International Journal of Computers and Applications 44 (2): 148–53. doi:10.1080/1206212X.2019.1710342
13. Chegeni, Vahid & Haj Seyyed Javadi, Hamid & Goudarzi, M. & Rezakhani, Afshin. (2021). Providing a hybrid cryptography algorithm for lightweight authentication protocol in RFID with urban traffic usage cases. 10.48550/arXiv.2104.07714.
14. Eldouh, Ahmed & Amein, Ahmed & Elkouny, A. & Lu, Songfeng. (2022). Hybrid Cryptography with a One–Time Stamp to Secure Contact Tracing for COVID–19 Infection. International Journal of Applied Mathematics and Computer Science. 10.34768/amcs-2022-0011
15. Ali, Sijjad & Wadho, Shuaib & Yc, Aun & Gan, Ming & Lee, Chen. (2024). Advancing cloud security: Unveiling the protective potential of homomorphic secret sharing in secure cloud computing. Egyptian Informatics Journal. 27. 100519. 10.1016/j.eij.2024.100519
16. J. Adilakshmi, B. Rithika, C. Pushpalatha, T. Venkatesh and M. Lohitha, "Secure Data Sharing in the Cloud Through Proxy Re-Encryption Technique," in 2024 4th International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 2024, pp. 700-705, doi: 10.1109/ICPCSN62568.2024.00117.

17. C. Ge, W. Susilo, Z. Liu, J. Baek, X. Luo and L. Fang, "Attribute-Based Proxy Re-Encryption With Direct Revocation Mechanism for Data Sharing in Clouds," in IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 2, pp. 949-960, March-April 2024, doi: 10.1109/TDSC.2023.3265979.
18. Seth, Bijeta & Dalal, Surjeet & Jaglan, Vivek & Le, Dac-Nhuong & Mohan, Senthilkumar & Srivastava, Gautam. (2022). Integrating encryption techniques for secure data storage in the cloud. Transactions on Emerging Telecommunications Technologies. 33. 10.1002/ett.4108