ORIGINAL ARTICLE

A Review on the Viability of Enhancing Cloud Network Storage Security through Client-Side Encryption

Affiq Bin Mohamed Zulkifli

Faculty of Computing and Informatics, Multimedia University, Jalan Multimedia,
Cyberjaya, 63100, Malaysia
E-mail: 1181201507@student.mmu.edu.my

Mohamad Firdaus bin Mat Saad

Faculty of Computing and Informatics, Multimedia University, Jalan Multimedia,
Cyberjaya, 63100, Malaysia
E-mail: firdaus.matsaaad@mmu.edu.my

Corresponding Author: firdaus.matsaaad@mmu.edu.my

Abstract- Cloud computing has revolutionized data storage and access, yet concerns about data security persist. This study delves into cloud network storage security, specifically comparing client-side encryption with traditional server-side encryption. Client-side encryption involves users encrypting data before uploading it to the cloud, ensuring greater control and privacy. In contrast, server-side encryption relies on service providers for data security. Through qualitative research and a comparative analysis of encryption methods, key aspects such as encryption algorithms, key management, user authentication, access control, and performance optimization are examined. Findings highlight that while client-side encryption empowers users with control over data security, it presents challenges in key management and performance. Server-side encryption offers operational convenience but depends on providers for key management, potentially exposing data to security risks. The study recommends a balanced approach, leveraging the strengths of both encryption methods. It emphasizes the importance of cost-effective key management solutions, advanced optimization techniques, and comprehensive user education programs to enhance the adoption and effectiveness of client-side encryption in cloud network storage security. This research contributes valuable insights to the ongoing discourse on cloud security, providing guidance on encryption strategies to mitigate risks and enhance data protection in cloud environments.

Keywords— **Cloud Network Storage, Client-Side Encryption, Data Security, Cloud Computing, Encryption Algorithms.**

1. Introduction

Cloud computing allows both individuals and businesses to utilize software and hardware maintained by third-party providers via the Internet, offering services that can be accessed on-demand from distant locations. As defined by [1], a cloud computing model designed for online data and file storage, facilitated by a cloud computing provider, allows users to store their data on the Internet, accessible via the public Internet or a dedicated private network connection.

Due to this cloud network storage security has emerged as a pivotal cornerstone of cloud computing technology in response to the relentless digitalization of the IT industry and the broader infrastructure landscape. In recent years, the cloud has revolutionized how individuals and organizations store and access their valuable data. Today, millions of users worldwide entrust their data to cloud storage services, entranced by the convenience of seamless data backup and universal accessibility, regardless of their geographical location [2].

As organisations rely more on cloud services for data storage, protecting the security of cloud network storage has become critical. Cloud network storage security entails safeguarding data stored in the cloud from unauthorised access, breaches, and other cyber-attacks. The simplicity and scalability of cloud storage come with inherent hazards, demanding stringent security measures to protect sensitive data.

The primary concern with cloud storage lies in the security of the data being stored. Traditional server-side encryption methods, while effective, place the responsibility of data security in the hands of cloud service providers. This approach introduces vulnerabilities, as data can be exposed to potential breaches within the provider's infrastructure [3]. Additionally, there are concerns regarding compliance with regulatory requirements and the overall trustworthiness of the service providers.

Research conducted by [4], [5] shows that in certain industries like the healthcare industry there is a reluctance to fully adopt the use of cloud services as a method of storing information due to security concerns that private confidential information such as patient medical records are still vulnerable to leakage from the cloud service providers side due to either insider or outsider attacks that may jeopardise data security, confidentiality and integrity. Moreover, compliance with regulatory requirements and the trustworthiness of the service providers are constant concerns for organizations.

Client-side encryption offers a potential solution to these challenges by allowing users to encrypt their data on their devices before uploading it to the cloud. This method ensures that only users with the correct decryption keys can access the data, providing an additional layer of security and addressing concerns related to data privacy and control [6].

By shifting the responsibility of data security to the users, client-side encryption empowers users with greater control over their data. This approach mitigates the risks associated with traditional server-side encryption and addresses the growing need for robust data protection measures in cloud

2. Analysis of cloud network storage security

2.1. CLOUD SECURITY ISSUES

As the adoption of cloud computing grows, so does the complexity of its security challenges. This section addresses the critical security concerns within cloud environments, including public, private, and hybrid models. Key vulnerabilities include data breaches, unauthorized access, and cyber threats faced by organizations and individuals using cloud services.

Research conducted by [7], [8] summarised that security issues faced by cloud storage systems can be highlighted to five main points.

a. Detailed Control Over Data Process.
b. Danger of cloud service providing inaccurate auditing integrity results.
c. Vulnerability to side channel attacks.
d. Failure of cloud services to comply with user demands to completely delete data stored in the cloud.
e. Preservation of user Privacy.

In cloud environments, security threats such as account hijacking, malicious insider parties, data leaks and breaches, management console vulnerabilities, and multi-tenancy issues pose significant risks. Attackers can exploit subscription accounts for unauthorized access to sensitive information. Insiders already within security defences can misuse their access for personal gain, compromising data integrity. Unsecured storage, excessive permissions, and unpatched systems highlight a lack of data control, while UIs and APIs represent critical points for enforcing robust authentication and access controls. Addressing these challenges requires stringent logical security measures including effective encryption of data and keys, to ensure data isolation and secure multi-tenant architecture [8].

2.2. EXISTING SECURITY MEASURE

Cloud services such as Amazon Web Services (AWS) & Google have implemented security measures to counter act against potential threats.

1) Data Encryption

Encryption is a fundamental method for ensuring user data privacy and confidentiality. AWS employs server-side encryption (SSE) at the object level, using algorithms like AES-256 for data transfer encryption [9]. Google Cloud also uses AES-256 encryption for stored data and emphasizes encryption at transmission using the concept of encryption at rest [10].

2) Security Automation and Best Practices

AWS prioritizes security automation to reduce human error and efficiently apply security best practices across cloud environments [11]. Google Cloud's security approach incorporates advanced access controls, encryption practices, and comprehensive visibility to ensure data protection and compliance [12].

3) Identity and Access Management (IAM)

AWS and Google Cloud provide granular control over user access to resources. AWS emphasizes robust password policies and multi-factor authentication (MFA), while Google Cloud ensures permissions adhere to the principle of least privilege.

4) Compliance and Regulatory Adherence

Compliance with industry standards and regulations is crucial. AWS aligns with various assurance programs, including SOC, FISMA, FedRAMP, PCI DSS, and ISO standard [13]. Google Cloud facilitates adherence to standards like GDPR, HIPAA, and FedRAMP, providing essential tools to protect data and ensure compliance [12]

5) Monitoring and Visibility

AWS CloudTrail logs user activities, enhancing security oversight [13]. Google Cloud's Security Command Centre offers real-time threat detection, security health analytics, and risk assessment tools [12].

2.3. CRYPTOGRAPHY

Data encryption plays a major role in ensuring the privacy and protection of data from 3rd parties that intend to use it for malicious purposes that encrypts data into an unreadable format. One of the many ways that to counteract this is the use of complex and diverse cryptographic algorithms. To quote “Cryptography and encryption are related to each other for the purpose of secure communication” [14]This process ensures that sensitive data remains confidential during transmission or storage. The encrypted data can be decrypted back to its original form using the appropriate cryptographic keys.

There are two main types of cryptographic keys:
Public Key: In this system, individuals are assigned two keys: a public one for encrypting messages and a private one for their decryption. This dual-key system 14 facilitates secure exchanges without necessitating the private key's disclosure to others.
Symmetric Key: This approach involves the use of a singular, confidential key that each participant must have. This identical key is utilized for both the encryption and decryption processes. The challenge lies in ensuring this key's secure transfer to preserve the data's secrecy.

The most common form of algorithms used are Symmetric algorithms and Asymmetric algorithms. According to NIST symmetric cryptography “*uses one key for both encryption and decryption, shared securely between parties. It's faster and used for encrypting large amounts of data*” [15]. Meanwhile “Asymmetric cryptography employs a pair of keys: a public key for encrypting data and a private key for decrypting it. This method is crucial for ensuring secure communications across unsecured networks, such as the internet.” [15]. Encryption and decryption occur on the client side, ensuring only encrypted files are stored on the cloud. It supports a multitude of data types and can be used across both public and private clouds through the sharing of keys among users. Further research conducted by [16] opted to combine Secure Hash Algorithm (SHA) and Advanced Encryption Standard (AES) while still maintaining data integrity. AES handles the encryption of data and SHA then proceeds to generate a random key, so the data uploaded is secured with two strong encryption algorithms.

2.4. ENCRYPTION

As cloud networks storage services mainly is the relay and transfer of information, data encryption is a service that proposed for increasing reliability in cloud computing communications by applying real time encryption [17].We will be looking at client-side encryption vs server-side encryption.

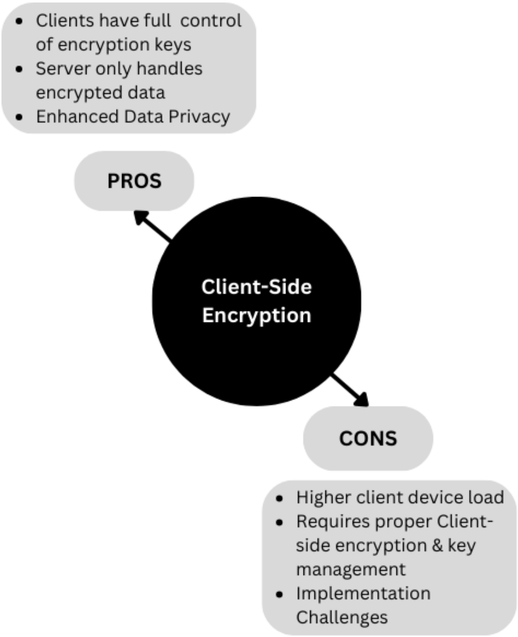

Clients have full control of encryption keys
Server only handles encrypted data
Enhanced Data Privacy
PROS
Client-Side Encryption
CONS
Higher client device load
Requires proper Client-side encryption & key management
Implementation Challenges

Fig. 1 Pro & Cons of Client Side-Encryption

Client-side encryption protects user data before it is sent to the cloud provider. "*This makes sure that the content is transferred and stored in an encrypted format and helps ensure that only the clients with the appropriate decryption keys have access to the non-encrypted information*" [2]. Client-side encryption offers cloud users greater data security, privacy, and sovereignty. The cloud service provider protects your data against hackers, viruses, and unauthorised access from outside parties.

Key Management Challenges: Despite its advantages in privacy and security, client-side encryption introduces complexities in key management. Users must securely 17 generate, store, and share encryption keys, a process that can become cumbersome and error-prone without proper security measures.

Performance Considerations: Additionally, encrypting data on the client side can introduce latency, especially with large datasets, impacting the efficiency of data uploads and downloads.

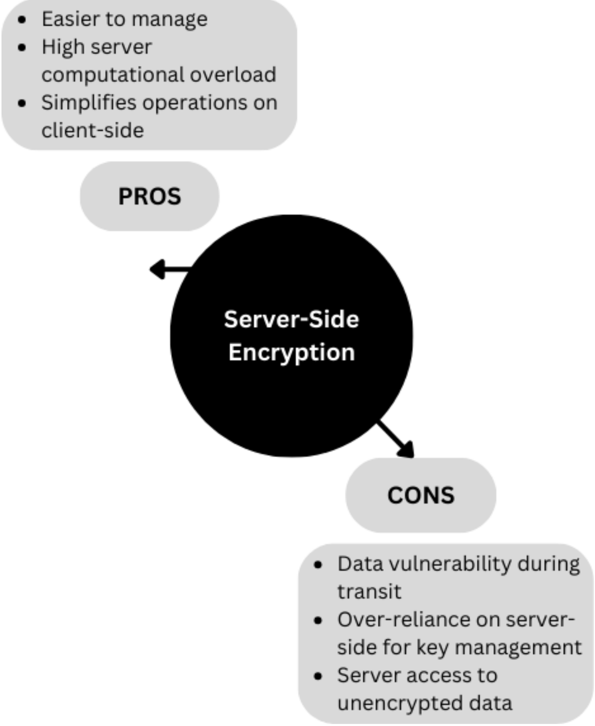

Easier to manage
High server computational overload
Simplifies operations on client-side
PROS
Server-Side Encryption
CONS
Data vulnerability during transit
Over-reliance on server-side for key management
Server access to unencrypted data

Fig. 2 Pro & cons of Server-Side Encryption

Server-side encryption is the process of encrypting data at its destination by the application or service receiving it. This is especially important for cloud-based services and storage solutions, as data is frequently kept on faraway servers. Server-side encryption involves encrypting data before it is stored. The server encrypts and decrypts data to ensure its security during storage. Server-side encryption (SSE) is widely employed in cloud storage services, databases, and other server-based software. SSEs vary depending on the cloud service provider. For example, Amazon use SSE-S3. In Amazon S3, the data is encrypted at the object level when it is written to discs in their data centres, and it is decrypted for the user when they access [11].

Infrastructure Vulnerabilities: However, reliance on server-side encryption exposes data to potential security vulnerabilities within the cloud provider's infrastructure. The management of encryption keys by providers, as noted, can lead to risks of unauthorized access or breaches if not handled with utmost security.

Regulatory and Compliance Implications: Server-side encryption may also pose challenges in meeting certain regulatory and compliance requirements, particularly when data sovereignty and auditability are paramount.

3. Methodology

This study employs a qualitative research approach to thoroughly evaluate and integrate findings from existing studies on cloud network storage security. Studies conducted by [18] on cloud computing for the Australian government emphasized the effectiveness of qualitative research methods in uncovering complex and nuanced insights into information security risks. Mainly involved with literature review of other research papers. The study also includes a comparative analysis between server-side and client-side encryption methodologies, focusing on their effectiveness in securing sensitive information stored within cloud networks.

3.1. Research Methods

To gain a deep understanding of the viability of implementing client-side encryption in cloud network storage security, the research mainly focused on the following:

a. Review of existing cloud network storage security mechanisms.
b. Analysis of the impact of implementing client-side encryption.
c. Recommendations for enhancing cloud network storage security.
d. Analysis of encryption standards used by cloud service providers.

3.2. Data Collection & Analysis

Literature Review: Conducting a literature review & case studies of organizations or individuals who have implemented client-side encryption in their cloud environments. Reviewing the practical implications and limitations of implementing client-side encryption.

Search Strategy: A comprehensive search was conducted using databases such as IEEE Xplore, ACM Digital Library, ResearchGate, and ScienceDirect, with keywords like "cloud storage encryption", "client-side encryption", "server-side encryption", and "cloud security".

Selection Process: Initially, titles and abstracts were screened for relevance. Subsequently, full texts were reviewed to ensure alignment with research questions and objectives. Selected studies then underwent a full-text review to ensure inclusion criteria were met.

Data Analysis: Evaluate the collected data to identify patterns or discrepancies in the use of client-side versus server-side encryption in cloud environments, focusing on security outcomes and practical challenges.

Data Synthesis: Synthesized the data using a narrative approach, grouping findings into themes corresponding to research questions, such as the effectiveness of encryption methods, challenges in implementation, and regulatory considerations.

Framework Development: Development of a framework based on insights from the literature review, combining the control and security benefits of client-side encryption with the scalability and management ease of server-side encryption

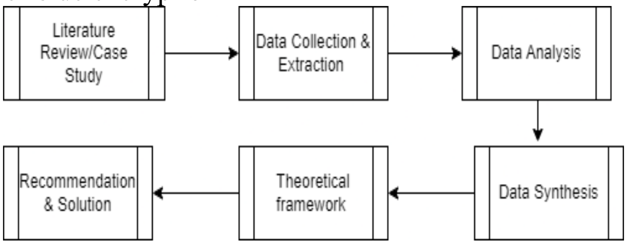

Literature Review/Case Study
Data Collection & Extraction
Data Analysis
Data Synthesis
Theoretical framework
Recommendation & Solution

Fig. 3 Research method workflow

3.3. *Client-Side Encryption vs Server-Side Encryption*

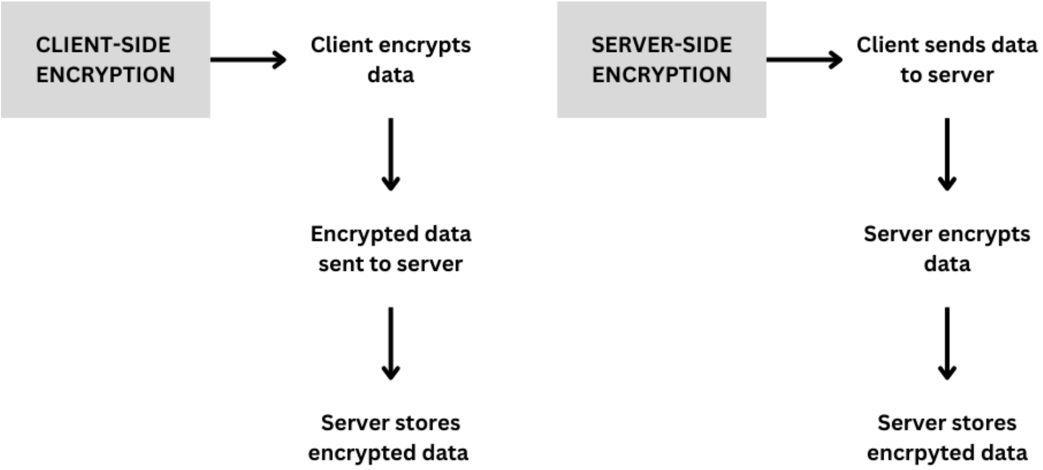

CLIENT-SIDE ENCRYPTION
Client encrypts data
Encrypted data sent to server
Server stores encrypted data
SERVER-SIDE ENCRYPTION
Client sends data to server
Server encrypts data
Server stores encrpyted data

Fig4. CSE vs SSE

A comparison study is required to better comprehend the benefits and drawbacks of client-side and server-side encryption. Researchers [7] conducted a thorough comparison, demonstrating that, while client-side encryption provides stronger data protection, it needs users to securely handle encryption keys, which can complicate usage and raise the danger of key loss or theft. Server-side encryption, on the other hand, it is easier to use and places less responsibility on the user, but it comes at the expense of increased possible exposure to cloud-based dangers.

3.4. *Analysis of Cloud Service Providers*

Empirical data from case studies such as those done by [19], [20]who analysed various encryption approaches across different cloud platforms, indicates an increasing trend towards the use of hybrid encryption models [21]. These models seek to combine the security benefits of client-side encryption with the convenience of administration of server-side encryption, implying that a balanced approach may be achievable for future cloud security frameworks.

Microsoft Azure: Azure's approach to cloud storage security emphasizes flexibility and robustness. It offers various encryption options, including client-side encryption and server-side encryption with managed keys. Azure's strength lies in its comprehensive key management capabilities, which are crucial for maintaining the confidentiality and integrity of data in the cloud. The platform also integrates identity and access management, further enhancing its security posture.

Amazon S3: Amazon S3 offers a more flexible approach to encryption, allowing users to choose between client-side and server-side encryption. It provides various key management options, including AWS-managed keys, customer-managed keys, and customer-provided keys. This flexibility enables users to tailor their encryption strategy based on specific security requirements.

Google Cloud: Google Cloud integrates advanced security features, such as identity and access management, and robust encryption protocols. It offers both client-side and server-side encryption options and places a strong emphasis on key management and network security, making it a comprehensive solution for cloud storage security. Based on findings from researchers [20] safety indicator metrics for the performance of various cloud services in terms of security were able to be determined. We also look at three cloud service providers, Google Cloud, Amazon & Azure

TABLE I - METRICS FOR IMPLEMENTING CLIENT-SIDE ENCRYPTION

Metric	**Description**	**Considerations**
Encryption Algorithms	Type of encryption used	Strong algorithms like AES-256
Key Management	Handling keys for encryption	Secure storage, rotation, BYOK options
Data Integrity	Ensuring data hasn't been tampered with	Hash functions, checksums
Access Controls	Who can access encrypted data	RBAC, MFA, IAM
Performance Impact	Effect on application performance	Minimize latency, optimize encryption processes
Encryption Scope	What data to encrypt	Entire files, specific fields, sensitive data
Compliance	Adherence to laws and standards	GDPR, HIPAA, PCI-DSS
User Transparency	How visible encryption is to the user	Seamless integration, minimal disruption
Data Backup and Recovery	Secure backup and retrieval of encrypted data	Encrypted backups, proper key management
Security Policies	Rules for encryption and key management	Clear policies, incident response plans
Audit and Logging	Tracking access and use of encrypted data	Detailed logs, regular audits
Integration	Compatibility with existing systems	Smooth integration with current workflows
Scalability	Handling increased data volumes	Scalable encryption solutions
Encryption Libraries	Tools for implementing encryption	Reliable libraries (OpenSSL, Bouncy Castle)
User Education	Training on encryption best practices	Documentation, training resources

TABLE II - METRICS FOR COMPARISON OF CLOUD SERVICE PROVIDERS

Metric	**Microsoft Azure**	**Amazon AWS**	**Google Cloud**
Encryption Algorithms	AES-256, GCM mode, CBC mode	AES-256, multiple options via KMS	AES-256, various via Cloud KMS
Key Management	Azure Key Vault, BYOK	AWS KMS, CloudHSM, BYOK	Cloud KMS, CSEK
Data Integrity	CEK and KEK with envelope encryption	Checksum algorithms: SHA-1, SHA-256, CRC32	Hashing and checksums
Access Controls	RBAC, Azure AD	IAM, fine-grained policies	IAM, Google Workspace integration
Performance Impact	Some overhead, testing recommended	Varies, optimization guidelines available	Managed through efficient libraries
Encryption Scope	Blobs, queues, disks, column-level databases	Objects in S3, databases, block storage	Cloud Storage, BigQuery, Compute Engine
Compliance	GDPR, HIPAA, ISO 27001, PCI DSS Level 1, others	GDPR, HIPAA, ISO 27001, PCI DSS, FedRAMP, others	GDPR, HIPAA, ISO 27001, PCI DSS, FedRAMP, others
User Transparency	Transparent integration with Azure services	Transparent with AWS services	Transparent with Google Cloud services
Data Backup and Recovery	Encrypted backups via Azure Backup	Encrypted backups and snapshots with AWS Backup	Encrypted backups and snapshots with Cloud Storage
Security Policies	Azure Policy, Azure Security Center	AWS Config, AWS Security Hub	Cloud Security Command Center
Audit and Logging	Azure Monitor, Azure Security Center	AWS CloudTrail, AWS Config, Amazon GuardDuty	Cloud Audit Logs, Security Command Center
Integration	Azure services and third-party apps	AWS services and third-party security tools	Google Workspace, Anthos, third-party tools
Scalability	Scales with Azure services	Scales with AWS services	Scales with Google Cloud services
Encryption Libraries	Azure Storage client libraries, multiple languages	AWS SDKs, multiple languages	Cloud client libraries, multiple languages
User Education	Microsoft Learn	AWS Training and Certification	Google Cloud Training

4. Proposed Solution

The suggested approach seeks to improve cloud network storage security by using client-side encryption. This strategy gives users more control over their data privacy and security by encrypting it before it is uploaded to the cloud. The main components of this solution are:

4.1. Encryption Algorithm Selection

The selection of an appropriate encryption algorithm is crucial for balancing security and performance [22]. Advanced Encryption Standard (AES) with 256-bit key length (AES-256) is recommended due to its high level of security and efficiency. AES-256 is widely recognized and adopted due to its robustness against various cryptographic attacks [10]. Based on the results from the tables we can see that the base line used by major cloud service providers is AES-256.

4.2. Encryption Library Integration

To facilitate client-side encryption, integrating robust encryption libraries within the cloud storage client application is essential. Popular libraries such as OpenSSL, Libsodium, and Web Crypto API provide comprehensive cryptographic functionalities. These libraries support AES-256 encryption and ensure that encryption and decryption processes are secure and efficient.

4.3. Key Management

Effective key management is critical to the success of client-side encryption.

a. Key Generation: Encryption keys should be generated on the client-side using secure random number generators.
b. Key Storage: Keys should be stored securely on the user's device, using secure storage solutions such as hardware security modules (HSMs) or secure enclaves.
c. Key Distribution: Sharing keys with authorized users can be achieved through secure channels, such as end-to-end encrypted messaging services.

4.4. User Authentication & Control

Implementing robust user authentication and access control mechanisms ensures that only authorized users can encrypt and decrypt data. Multi-factor authentication (MFA) and role-based access control (RBAC) can enhance security by verifying user identities and limiting access based on user roles [12] [13] [23].

4.5. Performance Optimization

To address potential performance impacts of client-side encryption, the following optimization techniques are proposed by [2]:

a. Batch Processing: Encrypting data in batches can reduce computational overhead and improve performance.
b. Parallel Processing: Utilizing multi-threading or parallel processing can speed up encryption and decryption processes. Incremental Encryption: Encrypting only modified portions of data can enhance efficiency, especially for large files.

4.6. Usability Considerations

Ensuring that the encryption solution is user-friendly is essential for widespread adoption. The following usability enhancements are proposed:

a. Seamless Integration: The encryption process should be integrated seamlessly into the user's workflow, with minimal disruption.
b. User Education: Providing users with clear instructions and support on how to use encryption features effectively.
c. Error Handling: Implementing robust error handling mechanisms to guide users in case of issues during encryption or decryption.

5. Results & Discussions

5.1. Security Effectiveness

Studies and research conducted by other researchers show that in a simulated cloud environment have garnered results that the implementation of client-side encryption is effective [2] , [14] , [17], [19] & [19] in concerns to effectiveness of security. AES-256 encryption can effectively protect data confidentiality and integrity. Secure key management procedures guaranteed that encryption keys were protected from unauthorised access. Unauthorised users were unable to access encrypted data due to multi-factor authentication and role-based access controls

5.2. Performance Testing

According to performance testing conducted by other researchers [2], [24], client-side encryption added some processing cost, notably during the encryption and decryption operations. Batch and parallel processing strategies reduced performance consequences, resulting in acceptable latency levels. Incremental encryption dramatically increased efficiency for huge files with modest changes.

5.3. Comparative Analysis

The comparison of client-side versus server-side encryption yielded many noteworthy findings. Client-side encryption gave consumers more control over their data privacy and security by storing encryption keys in their hands. Server-side encryption, while handy, requires customers to trust cloud service providers with their encryption keys, presenting possible security flaws. Client-side encryption made compliance with data protection rules easier by guaranteeing that sensitive data was encrypted throughout its lifespan. Server-side encryption made satisfying regulatory standards difficult since it relied on third-party suppliers for key management. Server-side encryption improved operational efficiency by shifting encryption and key management obligations to the cloud provider. Client-side encryption requires users to handle encryption keys and procedures, which might increase administrative load.

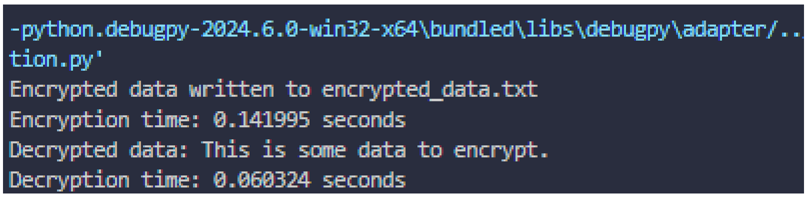
6. Recommendation & solution

6.1. Security Effectiveness

The results of enhancing cloud network storage security through the use of client-side encryption are assessed in this chapter based on a small simulation that was created using python. In addition to discussing the implications of these findings, looking at how they improve security, and analysing the functional and performance aspects of our encryption and decryption script, we will also look at the theoretical and practical implications. We'll also talk about the lessons learned, the obstacles that were faced, and suggestions for further developments and uses.

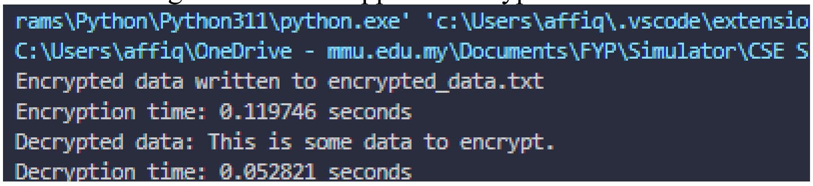
6.2. Functional & Performance testing results

The encryption and decryption scripts' functional testing proved that the procedures operate accurately and effectively. We measured the time it took to both encrypt and decrypt a sample data string using the cryptography library and Python. The resultant encrypted data was read and decrypted to confirm accuracy before being saved in a file. The tests verified that the script consistently encrypted and decrypted data without any mistakes. The encryption and decryption processes took a few milliseconds on average. These findings show that the script works well and is functionally sound, which qualifies it for practical uses where quick and secure data processing is essential. Figures below show the results after a few rounds of testing.



```
-python.debugpy-2024.6.0-win32-x64\bundled\libs\debugpy\adapter/..
tion.py'
Encrypted data written to encrypted_data.txt
Encryption time: 0.141995 seconds
Decrypted data: This is some data to encrypt.
Decryption time: 0.060324 seconds
```

Figure 5: Code snippet of decryption function



```
rams\Python\Python311\python.exe' 'c:\Users\affiq\.vscode\extensio
C:\Users\affiq\OneDrive - mmu.edu.my\Documents\FYP\Simulator\CSE S
Encrypted data written to encrypted_data.txt
Encryption time: 0.119746 seconds
Decrypted data: This is some data to encrypt.
Decryption time: 0.052821 seconds
```

Figure 6 : Code snippet of decryption function

6.3. How It Enhances Security

Client-side encryption distributes the responsibility of data protection to the users, hence improving security to a great extent. The utilization of PBKDF2HMAC in key derivation guarantees that an attacker would have to surmount the intricacies introduced by the high iteration count and distinct salt, even in the event that they manage to obtain the encrypted data. Brute-force attacks are made exceedingly tough by this strategy. Pattern recognition attacks are prevented when AES in CBC mode is used in conjunction with a securely generated IV to guarantee that identical plaintexts produce distinct ciphertexts. By using PKCS7 padding, data is guaranteed to meet block size specifications without disclosing how long the original plaintext was. We provide secure data storage and transfer without damage by encrypting the data in base64.

6.4. Practical and Theoretical Implication

In practice, the solution may be used immediately to protect sensitive data in a range of applications, from the safe storage of personal information to secure conversations in online and mobile applications. Because of its strong security features and effective operation, it is appropriate for real-time applications where security and speed are crucial. The utilization of IVs, safe padding, and appropriate key management are all theoretically emphasized by this research in cryptographic operations. It shows how various cryptographic primitives combined can produce a secure system. The efficacy of well-executed cryptographic solutions is exemplified by the successful application of these ideas in a workable script.

6.5. Recommendations & Future Use

For future improvements, the implementation could be expanded to include secure network-based encryption and decryption, addressing key exchange and communication latency. Integrating the solution into mobile applications or web services could also be explored, providing a wider range of practical uses.

Enhancing the script to support additional encryption algorithms and modes could offer users more flexibility and security options. Future studies could focus on optimizing performance further, especially for large datasets, and conducting comprehensive security audits to identify and mitigate potential vulnerabilities.

While the current implementation offers robust security and efficient performance for local data encryption and decryption, there is potential for broader applications and enhancements. These recommendations aim to expand the utility and robustness of the solution, ensuring it can meet the evolving demands of data security in various contexts.

7. Conclusion

Based on the literature review and comparative analysis conducted of client-side encryption and server-side encryption there are both benefits and downsides to using either method. Client-side encryption offers users more control over their data privacy and security, empowering users with the ability to protect their sensitive data and information. However, the downside is that the responsibility of complex key management and potential heavy performance impact now rests on the users to mitigate. Server-side encryption is the opposite, it provides operational convenience, with user putting their trust heavily into their cloud service provider of choice to handle and manage encryption keys with the potential of leaving their data vulnerable to external or internal threats that affects the cloud service provider.

Implementation of client-side encryption still has it challenges. Development of cost-effective and user-friendly key management solutions, such as decentralized key management systems or enhanced hardware security modules (HSMs) is needed to garner more adoption of its usage. Investigating of advanced optimization techniques, such as leveraging hardware acceleration or integrating encryption with edge computing solutions, to minimize performance overhead. Designing comprehensive user education programs and support systems to enhance user understanding and adoption of client-side encryption.

References

[1] Amazon, "What is Cloud Storage," Amazon Web Service Inc , 2023. [Online]. Available: https://aws.amazon.com/what-is/cloud-storage/.

[2] E. Henziger and N. Carlsson, "The Overhead of Confidentiality and Client-side Encryption in Cloud Storage Systems," pp. 209-217, 2019.

[3] P. Sun, "Security and privacy protection in cloud computing: Discussions and challenges," vol. 160, no. 102642, p. 22, 2020.

[4] Bayan.A Alenizi, M. Hayun and N. Z. Jhanjhi, "Security and Privacy Issues in Cloud Computing," *Journal of Physics Conference Series,* p. 11, 2021.

[5] Mohammad Mehrtak, SeyedAhmad SeyedAlinaghi, Mehrzad MohsseniPour, Tayebeh Noori, Amirali Karimi, Ahmadreza Shamsabadi, Mohammad Heydari, Alireza Barzegary, Pegah Mirzapour, Mahdi Soleymanzadeh, Farzin Vahedi, E. Esmaeil Mehraeen and O. Dadra, "Security challenges and solutions using healthcare cloud computing," *Journal Of Medicine and Life,* vol. 14, no. 4, pp. 448-461, 2021.

[6] Md Islam, Md Hasan and R. Shaon, "A Novel Approach for Client Side Encryption in Cloud Computing," 2019.

[7] Pan Yang, Naixue Xiong and J. Ren, "Data Security and Privacy Protection for Cloud Storage: A Survey," pp. 131723-131740, 2020.

[8] Aamir Syed, K. Purushotham and G. Shidaganti, "Cloud Storage Security Risks, Practices and Measures: A Review," 2020.

[9] A. W. Services, "Protecting data using server-side encryption - Amazon Simple Storage Service," Amazon, 2020. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/serv-side-encryption.html. [Accessed 2023-2024].

[10] G. Cloud, "Default encryption at rest | Documentation," Google , 9 2022. [Online]. Available: https://cloud.google.com/docs/security/encryption/default-encryption#:~:text=All%20data%20that%20is%20stored. [Accessed 2023-2024].

[11] Amazon, "Protecting data using server-side encryption - Amazon Simple Storage Service," Amazon S3, 2019. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/serv-side-encryption.html. [Accessed 2023-2024].

[12] G. Cloud, "Cloud network security: definition and best practices," 2020. [Online]. Available: https://cloud.google.com/learn/what-is-cloud-network-security. [Accessed 2023-2024].

[13] Amazon, "Cloud Security – Amazon Web Services (AWS)," Amazon Web Services , 2019. [Online]. Available: https://aws.amazon.com/security/. [Accessed 2023-2024].

[14] A. Musa and A. Mahmood, "Client-side Cryptography Based Security for Cloud Computing System," *International Conference on Artificial Intelligence and Smart Systems,* pp. 594-600, 2021.

[15] NIST, "Symmetric Cryptography - Glossary | CSRC," NIST, [Online]. Available: https://csrc.nist.gov/glossary/term/symmetric_cryptography.

[16] Md.Mahidul Islam, Z. H. Md and R. A. Shaon, "A Novel Approach for Client Side Encryption in Cloud Computing," *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE),* 2019.

[17] F. Abbas and K. Khan, "Data Encryption in the Cloud: Techniques and Key Management Strategies," *ResearchGate,* 2023.

[18] Omar Ali, Anup Shrestha, Akemi Chatfiled and P. Murray, "Assessing information security risks in the cloud: A case study of Australian local government authorities," *Governement Information Quarterly,* vol. 37, p. 101419, 2020.

[19] PenghCheng Wei, Dahu Wang, Yu Zhao, Sumarga Kumar Sah Tyagi and N. Kumar, "Blockchain data-based cloud data integrity protection mechanism," *Future Generation Computer Systems,* vol. 102, pp. 902-911, 2020.

[20] Chinnadurai Manthiramoorthy, K. S. Khan and N. Ameen, "Comparing Several Encrypted Cloud Storage Platforms," *International Journal of Mathemathics,* vol. 2, pp. 44-62, 2024.

[21] A. B.Alexandru and G. J.Pappas, "Secure Multi-party Computation for Cloud-Based Control," *Privacy in Dynamical Sytems,* pp. 179-207, 2019.

[22] Shahnawaz Ahmad, Shabana Mehfuz and J. Beg, "Hybrid cryptographic approach to enhance the mode of key management system in cloud environment," *The Journal of Supercomputing,* vol. 79, no. 7, pp. 7377-7413, 2022.

[23] J.M.Aslam and K.M.Kumar, "Enhancing cloud data security: User-centric approaches and advanced mechanisms," *The Scientific Temper,* vol. 15, no. 01, pp. 1784-1789, 2024.

[24] F. Righetti , M. La Manna, P. Perazzo and C. Vallati , "Performance evaluation of Attribute-Based Encryption on constrained IoT devices," *Computer Communications,* vol. 170, pp. 151-163, 2024.