IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657
Volume 7, Issue 2, 2019

Implementing AES Encryption Algorithm to Enhance Data Security in Cloud Computing

Poovendran Alagarsundaram,

Project Lead, IBM, Sacramento,

North Carolina, United States

Email: poovasg@gmail.com

ABSTRACT

Implementing the Advanced Encryption Standard (AES) algorithm in cloud computing is critical for improving data security in the face of increasing cyber threats and the exponential development of sensitive data storage and processing on remote servers. AES, a symmetric encryption method, uses a variety of cryptographic transformations to encrypt and decrypt fixed-length data blocks while maintaining strong confidentiality and integrity. AES emerged as the encryption standard in 2001, replacing the outdated Data Encryption Standard (DES), following a public competition organized by the United States National Institute of Standards and Technology (NIST) in the late 1990s. Its widespread acceptance across sectors and cloud computing environments can be attributed to its effectiveness in data security. This paper describes the methods for implementing AES, with a focus on key expansion, algorithm phases, and practical concerns for cloud deployment. While AES provides major security benefits, issues such as compatibility, performance overhead, and key management remain, necessitating constant research and innovation to solve these concerns and enhance AES encryption solutions in cloud computing. Organizations that apply AES principles can strengthen their data security posture, comply with regulatory standards, and promote confidence among cloud users by protecting critical information from unauthorized access and cyber threats.

Keywords: AES algorithm, Cloud computing, Cyber threats, Symmetric encryption, Cryptographic transformations, Confidentiality, Integrity, Regulatory standards, Cryptanalysis, Encryption solutions.

1. INTRODUCTION

Strong security measures are essential to protect sensitive data from cyber attacks and unauthorized access in the age of cloud computing, where enormous volumes of data are processed and stored across remote servers. The use of the Advanced Encryption Standard (AES) algorithm is one such crucial security precaution. One of the mainstays of contemporary cryptography is AES, which

ISSN 2347–3657
Volume 7, Issue 2, 2019

provides an advanced but effective way to encrypt data. Organizations can strengthen their data security posture and reduce the risks of data breaches and cyberattacks by implementing AES encryption into cloud computing environments. This improves confidentiality and integrity of data.

Symmetric encryption known for its security and effectiveness is the Advanced Encryption Standard (AES) algorithm. It uses a number of cryptographic transformations to safely encrypt and decrypt data, working with fixed-length data blocks. Key Expansion, the First Round, Rounds, and the Final Round are the main parts of the AES algorithm. The cipher key is subjected to Key Expansion during the encryption process, producing round keys that are utilized in further cryptographic procedures. In the first round, bitwise XOR is used to combine each byte of the data with the round key. Non-linear substitution (Sub Bytes), transposition (Shift Rows), mixing (Mix Columns), and extra XOR operations (Add Round Key) are all included in subsequent rounds. The Final Round's structure is the same as that of the earlier rounds, with the exception of the Mix Columns procedure. AES provides strong encryption of data through this series of cryptographic procedures, making it extremely resistant to efforts at decryption and unauthorized access.

A public competition was held by the U.S. National Institute of Standards and Technology (NIST) in the late 1990s to choose a new encryption standard to replace the outdated Data Encryption Standard (DES). This competition marked the beginning of the creation of the Advanced Encryption Standard (AES). The AES standard was established in 2001 when the Rijndael algorithm, created by Belgian cryptographers Vincent Rijmen and Joan Daemen, emerged as the winner of the competition following a thorough examination and analysis. As the de facto encryption method for protecting sensitive data in a variety of applications, including cloud computing, AES has been widely adopted across industries since it was standardized.

A plethora of cryptographic libraries and programming languages can be used to implement the AES encryption algorithm. Libraries for AES encryption and decryption can be found in third-party or built-in packages for popular programming languages including Python, Java, and C/C++. Furthermore, extensive support for AES encryption in a variety of settings and platforms is offered by specialist cryptographic libraries like Bouncy Castle and OpenSSL. To improve data security, cloud service providers might also include built-in encryption features or incorporate external encryption programs into their framework.

The Rijndael cipher was put forth by Belgian cryptographers Vincent Rijmen and Joan Daemen as a contender for the Advanced Encryption Standard (AES) competition, which was started by the National Institute of Standards and Technology (NIST) in the United States. This led to the development of the AES encryption algorithm. AES has been widely implemented and adopted by the global cryptographic community since it was chosen as the standard in 2001. Researchers, developers, and industry stakeholders have all contributed to this acceptance.

ISSN 2347–3657
Volume 7, Issue 2, 2019

The major goal of using the AES encryption algorithm in cloud computing environments is to improve data security by providing strong encryption techniques to safeguard sensitive information from unwanted access and cyber threats. Organizations want to protect the confidentiality, integrity, and authenticity of data stored and sent within cloud infrastructures by incorporating AES encryption into cloud-based applications and services. Furthermore, deploying AES encryption corresponds with regulatory compliance standards and industry best practices for data safety, which fosters trust and confidence among cloud users.

There are some obstacles and factors to take into account while implementing AES encryption, even though it provides a high level of security for data in cloud computing environments. Among these could be compatibility with current systems and protocols, performance overhead, and key management. In addition, it is crucial to continuously assess and update encryption procedures in order to preserve security efficacy due to the dynamic nature of threats and developments in cryptanalysis procedures. Further developments in data security can result from research initiatives aimed at resolving these issues and improving the effectiveness and usability of AES encryption in cloud computing circumstances.

There are still worries about the security of sensitive data handled and stored in cloud environments, even though cloud computing has become very popular. The confidentiality and integrity of data are seriously threatened by unauthorized access, data breaches, and cyberattacks. Although encryption is a vital tool for protecting data, its proper use in cloud computing necessitates giving careful thought to a number of variables, such as algorithm choice, key management, and performance effects. A reliable and effective method for improving data security is offered by the AES encryption algorithm, which is being implemented in cloud computing to try and overcome these difficulties. For AES encryption methods to be optimized and new security risks in cloud computing settings to be reduced, further research and innovation are nonetheless required.

2. LITERATURE SURVEY:

The benefits of the Advanced Encryption Standard (AES) are outlined by Abdullah (2017), who also point out that AES is widely used in industries like government communications, e-commerce, and banking. AES offers different levels of security and works with fixed-size data blocks that have key lengths of 128, 192, or 256 bits. NIST recognizes it as an international standard and it is built to withstand cryptographic attacks. The algorithm's reliability and ongoing security improvements are attributed to its effectiveness, adaptability, and transparency.

A study comparing the security of data using the encryption methods AES, DES, and RSA was carried out by Mahajan and Sachdeva (2013). They discovered that AES performs exceptionally well in terms of speed and security, accommodating varying key lengths to meet various security

ISSN 2347–3657
Volume 7, Issue 2, 2019

needs. However, because to its tiny key size and vulnerability to brute force assaults, DES is seen as less secure and has lost value in favor of AES. While RSA is slower than symmetric encryption methods like AES, it provides strong asymmetric encryption that is appropriate for digital signatures and key exchange. Among the three, AES is thought to be the most effective and safest, whereas RSA is mostly utilized for specialized encryption applications and DES is considered to be mostly outmoded.

AES (Advanced Encryption Standard) outperformed DES (Data Encryption Standard) in terms of speed and security, according to Rihan et al. (2015) performance analysis. Compared to DES, which has a fixed key size of 56 bits and an antiquated design, AES has an optimized design, supports greater key sizes, and is resistant to brute force attacks. These factors account for AES's supremacy. AES has become the industry standard for protecting sensitive data in a variety of applications because it performs better than DES in terms of efficiency, provides better security with variable key lengths, and integrates more sophisticated cryptographic approaches. Due to its security flaws, DES is typically discouraged for new deployments, despite the fact that it is still utilized in some older systems.

Lu and Tseng (2002) suggest a simplified system/module that combines functionalities into an integrated AES encryption and decryption. This method incorporates data handling, cryptographic operations, key management, security features, and compatibility assurance. It also increases efficiency. Scalability and flexibility across several deployment scenarios are supported by the design, which is validated against known vulnerabilities and subjected to stringent compliance testing.

In order to improve data security, Kumar and Rana (2016) suggest creating a modified version of the AES (Advanced Encryption Standard) algorithm that is compatible with the original AES standard but makes adjustments to key scheduling, substitution-permutation networks (SPNs), and other elements. Enhancement of resistance against cryptographic assaults, customization to particular application requirements, performance optimization, and key management are the goals of modifications. Comprehensive cryptanalysis, conformity to standards, public review, documentation, and ongoing improvement based on developments and input are all part of this iterative process.

AES encryption, which encrypts data before uploading and decrypts it upon retrieval to ensure secrecy and integrity, is the method of safe cloud storage described by Babitha and Babu (2016). Encryption key production, storage, and sharing must all be done securely as part of key management. This method makes use of data partitioning for fine-grained access control, access control methods, backup/disaster recovery plans, and HTTPS or TLS for secure transfer. Ongoing surveillance guarantees the security of data that is stored.

ISSN 2347–3657
Volume 7, Issue 2, 2019

AES encryption, which encrypts data before remote storage and ensures confidentiality and integrity, is a key component of data security in cloud storage, according to Mendonca (2018). Encryption keys must be generated, stored, and shared securely. This requires key management. In addition, access control, secure transmission, backup, recovery, and ongoing monitoring are all facilitated by AES encryption. Its scalability upholds security standards while satisfying the requirements of cloud storage settings.

Islam et al. (2008) support the use of longer key lengths and more sophisticated cryptographic approaches to provide improved security in symmetric data encryption using the AES (Advanced Encryption Standard) methodology. Higher levels of data confidentiality are ensured by extending the length of the AES key to 192 or 256 bits, which greatly increases resistance against different cryptographic assaults including brute force attacks. Performance may be marginally impacted by greater key lengths, but these impacts are mitigated by contemporary enhancements. Longer key length AES has become the industry standard, used in government, healthcare, and finance to ensure regulatory compliance and compatibility. Ongoing assessment guarantees efficacy against changing risks, supporting overall security posture and user trust.

AES, RSA, ECC, and other cryptographic algorithms are compared and evaluated for security, performance, scalability, and applicability for cloud systems in Semwal and Sharma (2017) study. AES is the best symmetric encryption algorithm for data that is not in use, whereas RSA and ECC are the most secure and effective options for asymmetric encryption and key management. Key management, scalability, and performance impacts are taken into account. Strong encryption for cloud storage is provided by Blowfish and Twofish, and data integrity is guaranteed by SHA-3. In order to provide flexibility to changing threats and requirements, cloud computing cryptography algorithms must take into account performance, scalability, regulatory compliance, and continual review.

Arora et al. (2013) support the use of encryption methods like AES and RSA to secure user data in cloud computing, guaranteeing its confidentiality and integrity. Strong algorithms are used to encrypt data before it is stored or sent, protecting it against manipulation or unwanted access. While RSA is used for safe asymmetric encryption during data transmission, AES is used for effective symmetric encryption of data at rest. Ensuring safe key generation, storage, and distribution requires the implementation of strong key management protocols. Interception is prevented via secure communication protocols like HTTPS or TLS, and unwanted data access is limited by access control measures. Consistent audits and surveillance identify and address security risks, guaranteeing adherence to industry norms and data protection laws.

Shimbre and Deshpande (2015) suggest integrating the AES (Advanced Encryption Standard) algorithm with Third-Party Auditing (TPA) to improve distributed data storage security for cloud computing. While AES provides encryption for data confidentiality, bolstering security against

ISSN 2347–3657
Volume 7, Issue 2, 2019

unauthorized access and tampering in distributed cloud storage systems, TPA maintains data integrity through impartial audits. By ensuring transparency, scalability, regulatory compliance, constant monitoring, and resistance against assaults, this combined strategy strengthens user confidence in cloud storage services.

3. METHODOLOGY

Improving data security in cloud computing environments requires the implementation of the Advanced Encryption Standard (AES) algorithm. AES, a symmetric encryption technique, is frequently used due to its reliability and effectiveness. This section details the implementation methods for AES, including an overview of its components, key expansion, algorithm phases, table explanations, architecture diagrams, and equations. In addition, practical issues for cloud deployment are highlighted to enable effective and secure encryption.

3.1. Overview of the AES algorithm.

The AES method uses a symmetric key to operate on fixed-length data blocks, which are typically 128 bits long. The encryption process consists of well-defined processes, such as key expansion, start and final rounds, and several transformation rounds. These processes ensure that the data is fully encrypted, making it extremely difficult to decrypt without the correct key.

Key Expansion: This phase creates a succession of round keys from the initial cipher key. It ensures that each encryption round utilizes a unique key, hence improving security.

In the *initial round*, each byte of data is merged with the initial round key via a bitwise XOR operation.

Rounds are the repetitive application of four major operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey.

Final Round: This stage, like the other rounds, includes SubBytes, ShiftRows, and AddRoundKey but skips the MixColumns phase.

These stages all contribute to the powerful encryption process that distinguishes AES, making it the preferred choice for data security in cloud computing environments.

3.2. Key Expansion

Key Expansion is a fundamental step in the AES algorithm that converts the original cipher key into a succession of round keys. This procedure employs Rijndael's key scheduling, which ensures that each encryption round uses a distinct key, so increasing security.

The key expansion process consists of the following steps:

ISSN 2347–3657
Volume 7, Issue 2, 2019

Key Schedule Core: This comprises rotating the key's bytes, applying the S-box to each byte, and XORing the result with a round constant.

Word Generation: The previous words are combined with the key schedule core to create new words.

Round Key Formation: The words are then combined to create round keys for each encryption round.

This systematic key expansion protects the encryption process's security and resistance to attacks.

Table 1: Components and Steps of the AES Algorithm.

Step	Description	Details
Key Expansion	Creates round keys using the cipher key	Uses the key schedule of Rijndael to generate a different key for every encryption cycle.
Initial Round	AddRoundKey	Employs bitwise XOR to combine each byte of the state with the round key.
Main Rounds	SubBytes, ShiftRows, MixColumns, and AddRoundKey transformations are used in several rounds.	SubBytes: S-box-based non-linear replacement step. ShiftRows: Rearranges the state's rows cyclically. MixColumns: Employs a fixed polynomial to blend the bytes in every column. - AddRoundKey: Incorporates the round key and the state together.
Final Round	Similar to the main rounds, but no MixColumns.	SubBytes Shift Rows- AddRoundKey

S-box

- Goal: Offers non-linearity throughout the encryption procedure.
- In the SubBytes stage, a predetermined replacement table called the S-box is utilized. Every byte in the state matrix is swapped out with a matching byte from the table.
- Generation: Made resistant to both linear and differential cryptanalysis with the use of a mathematical transformation.

The S-box introduces non-linearity into the encryption process, which is essential for resisting certain types of cryptanalysis, and is carefully designed to be difficult to reverse-engineer, thereby enhancing security.

MixColumns Polynomial

- Purpose: By combining bytes inside each column, it provides diffusion.

ISSN 2347–3657
Volume 7, Issue 2, 2019

- The state matrix's columns are each transformed using a fixed polynomial in the MixColumns step.
- Polynomial:

2 3 1 1 1 2 3 1 1 1 2 3 3 1 1 2

The influence of every byte in the ciphertext is distributed over a number of bytes thanks to this polynomial. The polynomial enables diffusion by distributing the influence of each byte across many bytes in the ciphertext, while its fixed structure allows for consistent and reliable translation of the state matrix columns.

1. Input: At the start of the AES encryption process, the plaintext message and the initial key are used.
2. Key Expansion Module: This module uses a special algorithm to create round keys from the initial cipher key.
3. Initial Round: The first round uses the AddRoundKey operation, which combines each byte of the state with a round key via bitwise XOR.
4. Main Rounds: Subsequent rounds, known as the main rounds, include a sequence of operations such as SubBytes (byte substitution), ShiftRows (row-wise shifting), MixColumns (column mixing), and AddRoundKey (XOR with round key).
5. Final Round: The final round of the AES algorithm contains SubBytes, ShiftRows, and AddRoundKey operations, but not the MixColumns phase.
6. Output: After the encryption procedure is completed, the ciphertext is generated, which represents the encrypted version of the original plaintext message.

ISSN 2347–3657
Volume 7, Issue 2, 2019

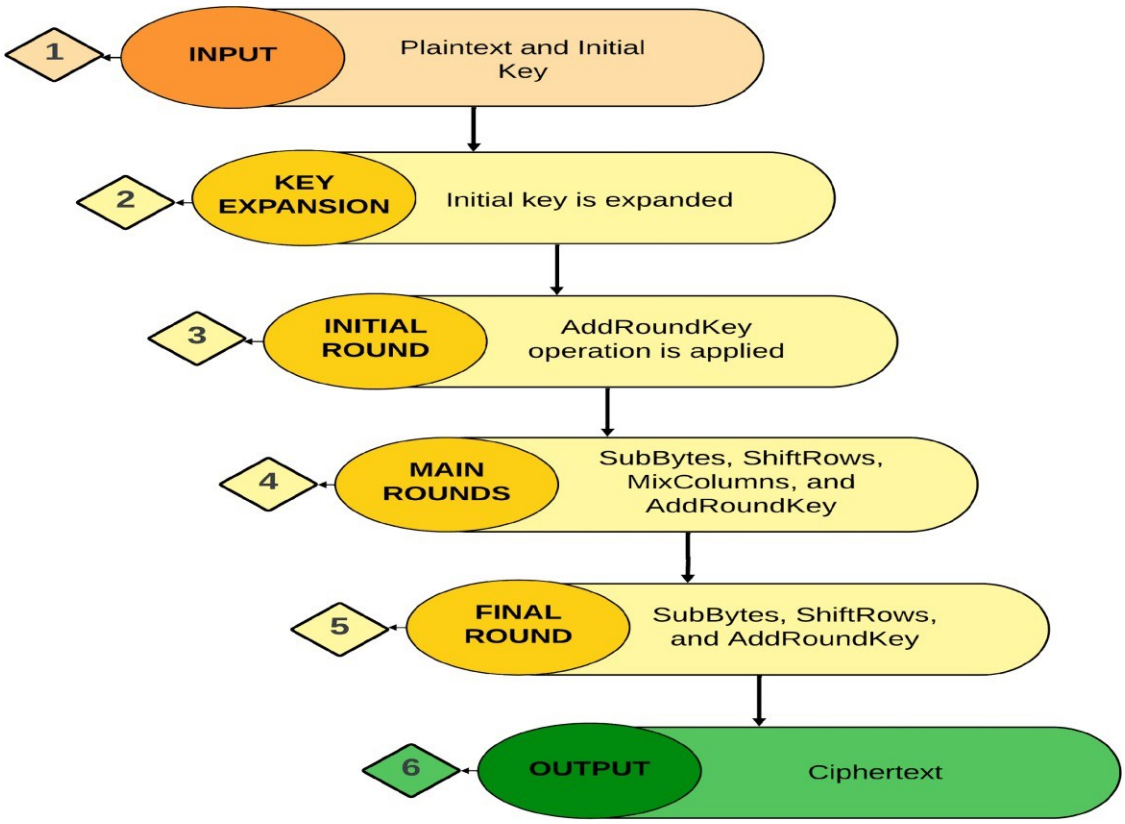

Figure 1: Visualizing the Encryption Process: AES Algorithm Overview

3.3. Algorithm Steps

The AES encryption process is divided into numerous parts, each of which contributes to the algorithm's overall security and efficiency.

SubBytes: This is a non-linear substitution phase in which each byte in the state matrix is replaced with a corresponding byte from a fixed lookup table called the S-box. This stage assures non-linearity in the encryption process, increasing its security.

ShiftRows: In this stage, the state matrix's rows are cyclically shifted. The degree of shift varies by row number, with the first row remaining intact, the second row shifted by one byte, and so on. This step promotes dissemination by transposing the bytes.

MixColumns alters each column of the state matrix with a fixed polynomial. This phase guarantees that the bytes are further combined, resulting in further diffusion.

ISSN 2347–3657
Volume 7, Issue 2, 2019

A bitwise XOR operation is used to combine the state matrix with the round key generated during Key Expansion. This stage complicates the encryption process by merging the plaintext and round key.

These processes are carried out for a fixed number of rounds, which is specified by the key length. The final round bypasses the MixColumns step, which is necessary to complete the encryption.

3.4. Equations

Equations are the mathematical foundation of AES transformations, ensuring accurate and consistent actions throughout the encryption process.

SubBytes: In this transformation, each byte in the state matrix is replaced using the S-box.

$$S'(i,j) = S(S(i,j))$$

Where S is the S-box lookup function.

ShiftRows: This transformation cycles across the rows of the state matrix.

$S'(i,j)=S(i,(j+shift(i))modNb)$

Where shift(i) defines the cyclic shift for row i and Nb represents the number of columns in the state matrix.

MixColumns: This transformation uses a fixed polynomial to mix the bytes in each column.

$$S'(i,j) = 2\ 3\ 1\ 1\ 1\ 2\ 3\ 1\ 1\ 1\ 2\ 3\ 3\ 1\ 1\ 2 \quad \times \quad S(0,j)\ S(1,j)\ S(2,j)\ S(3,j)$$

These equations ensure that each transformation step is mathematically valid, contributing to the AES algorithm's overall security and efficiency.

3.5. Implementation Steps

Implementing the AES algorithm in a cloud computing environment requires many critical steps:

Choose a Suitable Library: Choose from well-known cryptographic libraries such as Bouncy Castle and OpenSSL, or language-specific libraries like PyCrypto for Python and Java Cryptography Extension (JCE) for Java.

Initialize Parameters: Set the key size (128, 192, or 256 bits), plaintext, and initialization vectors as needed. Initialization vectors are very crucial for several types of AES operation, such as Cipher Block Chaining (CBC).

ISSN 2347–3657
Volume 7, Issue 2, 2019

Key generation: Generate the initial cipher key using a secure random number generator. This ensures that the key remains surprising and secure.

Encrypt data:

Key Expansion: Take the initial cipher key and generate round keys.

Encryption Process: Transform the plaintext through the Initial Round, Multiple Rounds, and Final Round.

Ciphertext is the output of the encryption process.

Decrypt Data: To recover the original plaintext, repeat the previous steps using the same round keys. The decryption procedure employs the inverse operations of the encryption processes, such as InvSubBytes, InvShiftRows, and InvMixColumns.

3.6. Practical Considerations

When implementing AES in a cloud computing environment, numerous practical considerations must be made to ensure optimal speed and security:

Performance Overhead: The encryption and decryption operations add computational overhead, which might affect performance. It is critical to improve these operations to reduce latency and prevent them from becoming bottlenecks.

Key administration: The secure storage and administration of encryption keys is crucial. Consider employing hardware security modules (HSMs) or key management services (KMS) offered by cloud platforms. These tools provide secure key generation, storage, and administration, which reduces the danger of key compromise.

Integration with Cloud Services: Many cloud service providers have encryption options. Use these features or incorporate third-party encryption solutions to improve security without sacrificing performance. For example, Amazon Web Services (AWS) offers AWS Key Management Service (KMS) and AWS CloudHSM for key management and encryption.

Compliance and legislation: Ensure that the encryption solution adheres to applicable industry standards and legislation. Compliance with GDPR, HIPAA, or PCI-DSS may necessitate specific encryption processes and key management protocols.

Scalability: Consider the encryption solution's scalability. Cloud environments frequently contain massive amounts of data, and the encryption solution must be capable of handling this size without degrading performance.

ISSN 2347–3657
Volume 7, Issue 2, 2019

Backup and recovery: Ensure that the encryption keys and encrypted data are securely stored. Implement robust recovery methods to avoid data loss in the event of a key compromise or hardware failure.

Monitoring and auditing: Set up mechanisms to track encryption key usage and detect unwanted access attempts. This aids in ensuring the security and integrity of the encrypted data.

Implementing the AES algorithm in cloud computing environments improves data security by offering strong encryption techniques. This technique covers the necessary processes and considerations for successful deployment, ensuring that sensitive data is safeguarded from unauthorized access and cyber threats. Understanding and utilizing AES principles allows enterprises to achieve a high level of data confidentiality and integrity in their cloud computing infrastructures.

4. RESULT AND DISCUSSION

Data security is effectively addressed by cloud computing settings through the use of the AES algorithm. Organizations may guarantee that their information is safe from cyber threats and unlawful access by encrypting important data using AES. The AES algorithm uses well-defined encryption procedures and symmetric encryption techniques to provide a high level of security. By guaranteeing that every encryption cycle utilizes a different key, key expansion improves security. Strong data encryption is achieved through a series of well-defined phases in the encryption process, such as AddRoundKey, MixColumns, SubBytes, and ShiftRows. For AES to be successfully deployed in cloud environments, practical factors including performance overhead, key administration, integration with cloud services, compliance, scalability, backup and recovery, and monitoring and auditing are essential. The results demonstrate how well the AES algorithm protects data in cloud computing environments, giving businesses a dependable way to protect their important data.

5. CONCLUSION

Finally, implementing the Advanced Encryption Standard (AES) algorithm in cloud computing environments is a critical step toward improving data security against rising cyber threats. Organizations can strengthen their cloud infrastructures and instill user confidence in data security and integrity by exploiting AES's strong encryption algorithms and addressing practical factors such as performance overhead and key management. Continuous research and innovation are required to overcome hurdles and optimize AES encryption methods, assuring their effectiveness in managing changing security concerns. Finally, incorporating AES into cloud computing frameworks is an important step toward protecting sensitive data from unwanted access and cyber threats.

ISSN 2347–3657
Volume 7, Issue 2, 2019

6. FUTURE ENHANCEMENT

Future improvements to AES implementation in cloud contexts might concentrate on enhancing security and speed even more. This could entail creating more effective encryption and decryption algorithms designed especially for cloud infrastructures and incorporating cutting-edge key distribution and management strategies. Furthermore, improving AES's scalability and compatibility with new cloud technologies will guarantee that it stays a trustworthy and efficient encryption option for developing cloud computing environments.

7. REFERENCE

1) Abdullah, A. M. (2017). Advanced encryption standard (AES) algorithm to encrypt and decrypt data. Cryptography and Network Security, 16(1), 11.
2) Mahajan, P., & Sachdeva, A. (2013). A study of encryption algorithms AES, DES and RSA for security. Global journal of computer science and technology, 13(15), 15-22.
3) Rihan, S. D., Khalid, A., & Osman, S. E. F. (2015). A performance comparison of encryption algorithms AES and DES. International Journal of Engineering Research & Technology (IJERT), 4(12), 151-154.
4) Lu, C. C., & Tseng, S. Y. (2002, July). Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Proceedings IEEE International Conference on Application-Specific Systems, Architectures, and Processors (pp. 277-285). IEEE.
5) Kumar, P., & Rana, S. B. (2016). Development of modified AES algorithm for data security. Optik, 127(4), 2341-2345.
6) Babitha, M. P., & Babu, K. R. (2016, September). Secure cloud storage using AES encryption. In 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT) (pp. 859-864). IEEE.
7) Mendonca, S. N. (2018). Data security in cloud using AES. Int. J. Eng. Res. Technol, 7.
8) Islam, M. N., Mia, M. M. H., Chowdhury, M. F., & Matin, M. A. (2008, August). Effect of security increment to symmetric data encryption through AES methodology. In 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (pp. 291-294). IEEE.
9) Semwal, P., & Sharma, M. K. (2017, September). Comparative study of different cryptographic algorithms for data security in cloud computing. In 2017 3rd international conference on advances in computing, communication & automation (ICACCA)(Fall) (pp. 1-7). IEEE.
10) Arora, R., Parashar, A., & Transforming, C. C. I. (2013). Secure user data in cloud computing using encryption algorithms. International journal of engineering research and applications, 3(4), 1922-1926.

ISSN 2347–3657
Volume 7, Issue 2, 2019

11) Shimbre, N., & Deshpande, P. (2015, February). Enhancing distributed data storage security for cloud computing using TPA and AES algorithm. In 2015 International Conference on Computing Communication Control and Automation (pp. 35-39). IEEE.