

Using Hybrid Hashing Algorithm to Secure Cloud Services (AES, ChaCha20-Poly1305 , AES GCM, Fernet, AES CCM)

Anubhav Agarwal, Maulik Bahri, Dr. Parthasarathy G
Vellore Institute of Technology
Vellore – 632014, Tamil Nadu, India

Abstract

The increasing reliance on computers to store sensitive information and the need for automation have necessitated the development of secure systems to protect data. With the growing amount of data being generated, there is a demand for secure high-capacity storage solutions. Security is particularly crucial as more people become dependent on the digital world for various purposes. However, sending and retrieving data from the cloud can be risky, as it may be intercepted or altered by an eavesdropper. To address these concerns, a proposed method for storing data securely in the cloud involves a multilevel cryptography-based security system. This model is a hybrid approach that combines both symmetric and asymmetric key cryptography algorithms. While there are already many secure systems in existence, the rapidly advancing technology demands the development of new, more secure systems. Therefore, this proposed model aims to enhance security in cloud computing, where security is of utmost importance.

1. Introduction

Traditional storage devices such as flash drives and hard disks are gradually becoming outdated due to the need for global expansion of businesses, which requires data sharing among employees for collaborative work. Also, individuals now have multiple devices, including mobile phones, tablets, laptops, and desktop computers, and cloud storage provides a convenient way to access personal data across all these devices.

Despite its numerous benefits, cloud computing faces security challenges, making it difficult to use. One major challenge is the time and memory limitations during encryption and decryption processes due to the shared performance of cloud servers. Additionally, data transmitted to the cloud may be intercepted and manipulated by unauthorized persons.

To address these security concerns, the proposed approach uses hybrid cryptographic mechanisms for key exchange and encryption/decryption. This approach enhances security in the cloud while also optimizing time and memory usage during encryption and decryption processes. Ultimately, the goal is to ensure the safe and secure storage and transmission of sensitive data in the cloud.

2. Literature Survey

- i. Paper titled "Data Storage Security Issues in Cloud Computing" was presented at the 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM) by Devang Pratap Singh, Prakarsh Kaushik, Manjari Jain and published by IEEE in January 2022. The main focus of the paper is to discuss the various security issues related to data storage in cloud computing. The authors have identified various security issues such as data privacy, data confidentiality, data integrity, and data availability, which are crucial for ensuring secure data storage in cloud computing. The authors have provided a comprehensive literature review of previous studies and research works related to data storage security issues in cloud computing. They have also discussed various techniques and approaches used for ensuring secure data storage in cloud computing. The paper provides an in-depth analysis of the various security issues and challenges faced by cloud computing users and service providers. The authors have suggested several measures that can be taken to ensure secure data storage in cloud computing. These measures include encryption, access control, backup and recovery, and intrusion detection and prevention systems. Overall, the paper provides valuable insights into the security issues related to data storage in cloud computing and proposes several techniques and approaches for ensuring secure data storage.
- ii. Paper titled "Security Threats and Challenges in Public Cloud Storage" was presented at the 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) by G. Nagarajan, Dr. K. Sampath Kumar and published by IEEE in June 2021. The main focus of the paper is to discuss the various security threats and challenges that are associated with public cloud storage. The authors have identified several security threats such as unauthorized access, data breaches, data loss, and data theft, which pose a significant risk to public cloud storage. The authors have provided a comprehensive literature review of previous studies and

research works related to security threats and challenges in public cloud storage. They have also discussed various techniques and approaches used for mitigating these security threats. The paper provides an in-depth analysis of the various security threats and challenges faced by users of public cloud storage and service providers. The authors have suggested several measures that can be taken to mitigate these security threats. These measures include the use of encryption, access control, data backup and recovery, and intrusion detection and prevention systems. Overall, the paper provides valuable insights into the security threats and challenges associated with public cloud storage and proposes several techniques and approaches for mitigating these security threats.

- iii. The paper titled "Securing File Storage on the Cloud using Cryptography" was published in the International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE) by Aman Singh, Shivashankar Reddy Ginni, Dr. Advin Manhar in April 2021. The main focus of the paper is to propose a secure file storage mechanism for cloud computing using cryptography. The authors have identified the importance of file security in cloud computing and have proposed a solution to ensure secure file storage. The authors have provided a detailed description of the proposed file storage mechanism, which uses cryptographic techniques to ensure the confidentiality and integrity of the stored files. The proposed mechanism uses a combination of symmetric and asymmetric encryption techniques to ensure secure file storage on the cloud. The paper provides an in-depth analysis of the proposed file storage mechanism, including the encryption and decryption processes. The authors have also discussed the advantages of using cryptography for securing file storage in cloud computing. However, the paper lacks a comprehensive evaluation of the proposed mechanism's performance and scalability, which could be an area for future research.
- iv. The paper titled "Blockchain Techniques for Secure Storage of Data in Cloud Environment" by Praveen Kumar Kollu, Monika Saxena, Khongdet Phasinam, Thanwamas Kassinuk, Malik Mustafa was published in the Turkish Journal of Computer and Mathematics Education in May 2021. The paper focuses on the use of blockchain techniques for secure data storage in cloud computing environments. The authors have identified the security issues related to data storage in the cloud and proposed a solution based on blockchain technology to address these security concerns. The authors have provided a comprehensive literature review of previous studies and

research works related to the use of blockchain in cloud computing. They have also discussed the advantages of using blockchain for secure data storage in cloud computing. The paper provides a detailed description of the proposed solution, which involves the use of blockchain-based distributed storage to ensure secure data storage in the cloud. The authors have also discussed the implementation and performance evaluation of the proposed solution. Overall, the paper proposes a promising solution for secure data storage in the cloud using blockchain technology. However, the paper could have included more detailed information on the implementation and evaluation of the proposed solution, which could have added more value to the paper.

- v. The paper titled "Secure File Storage using Hybrid Cryptography" by P. Bharathi, G. Annam, J. B. Kandi, V. K. Duggana and A. T was presented at the 6th International Conference on Communication and Electronics Systems (ICCES) in 2021. The paper proposes a hybrid cryptography-based solution for secure file storage in cloud computing environments. The authors have identified the importance of secure file storage in cloud computing and have proposed a solution to address the security concerns related to file storage. The authors have provided a comprehensive literature review of previous studies and research works related to cryptography and cloud computing. They have also discussed the advantages of using hybrid cryptography for secure file storage in cloud computing. The paper provides a detailed description of the proposed solution, which involves the use of both symmetric and asymmetric cryptography techniques to ensure secure file storage in the cloud. The authors have also discussed the implementation and evaluation of the proposed solution. Overall, the paper proposes a promising solution for secure file storage in cloud computing using hybrid cryptography techniques. The proposed solution can be a useful resource for researchers and practitioners working in the field of cloud computing and data security. However, the paper could have provided more details on the performance evaluation of the proposed solution, which could have added more value to the paper.

Other Literature Surveys Include:

- i. Pan Yang, Neal N. Xiong, Jigli Ren, "Data Security and Privacy Protection for Cloud Storage: A Survey", IEEE Access, vol 4, pp(99) : 1-1, July 2020.

- ii. Bharati Mishra, Debsish Jena, "Security of Cloud Storage: A Survey", 2019 International Conference on Information Technology (ICIT), 2019 IEEE, March 2020.
- iii. Bijeta Seth, Surjeet Dalal, Da-Nhuong Le, Vivek Jaglan, Neeraj Dahiya, Akshat Agrawal, Mayank Mohan Sharma, Deo Prakash, K.D. Verma, "Secure Cloud Data Storage System Using Hybrid Paillier-Blowfish Algorithm", Computers, Materials and Continua (CMC), Vol. 67, no. 1, November 2020.
- iv. Manoj V. Brahme, Dr. Milind V. Sarode, Dr. Meenakshi S. Arya, "Design and Implementation of Secure File Storage using Distributed Cloud Mechanism", International Journal of Research and Analytical Reviews (IJRAR) , Vol. 6, Issue 1, February 2019.
- v. S. Pavithra, S. Ramya, Soma Prathibha, "A Survey On Cloud Security Issues and Blockchain", 3rd International Conference on Computing and Communication Technologies (ICCT), 2019 IEEE, 136-140, 2019.
- vi. S. A. Ahmad and A. B. Garko, "Hybrid Cryptography Algorithms in Cloud Computing: A Review," 2019 15th International Conference on Electronics, Computer and Computation (ICECCO), 2019, pp. 1-6
- vii. Chauhan, A. and Gupta, J., 2017, September. A novel technique of cloud security based on hybrid encryption by Blowfish and MD5. In 2017 4th International conference on signal processing, computing and control (ISPCC) (pp. 349-355). IEEE.
- viii. Chhabra, A. and Arora, S., 2017, October. An elliptic curve cryptography based encryption scheme for securing the cloud against eavesdropping attacks. In 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC) (pp. 243-246). IEEE.
- ix. Kanatt, S., Jadhav, A. and Talwar, P., 2020. Review of Secure File Storage on Cloud using Hybrid Cryptography. International Journal of Engineering Research & Technology (IJERT).
- x. Swarna, C. and Eastaff, M.S., 2018. Secure File Storage in Cloud Computing Using Hybrid Cryptography Algorithm. Iaetsd Journal for Advanced Research in Applied Science.
- xi. Maitri, P.V. and Verma, A., 2016, March. Secure file storage in cloud computing using hybrid cryptography algorithm. In 2016 international conference on wireless communications, signal processing and networking (WiSPNET) (pp. 1635-1638). IEEE.

3. Proposed Model

The main objective of the proposed system is to build a secure cloud-protected file storage system that satisfies the basic principles of cryptography, namely confidentiality, integrity, and verification, without compromising the speed of advanced applications. To achieve this goal, the system employs hybrid cryptography, which uses three encryption algorithms: AES, Fernet, and ChaChaPoly1305. The system also utilizes the Round Robin algorithm, which is a load-balancing algorithm that distributes incoming requests evenly among a group of servers. This algorithm helps to optimize the system's performance and reduce processing time, which is crucial for advanced applications. Furthermore, digital signatures are used to ensure data integrity and verification. Digital signatures help to detect any unauthorized changes made to the data, ensuring that the data remains unaltered and trustworthy.

In summary, the proposed cloud-protected file storage system employs a combination of hybrid cryptography, load-balancing algorithms, and digital signatures to ensure data confidentiality, integrity, and verification, while also optimizing system performance for advanced applications.

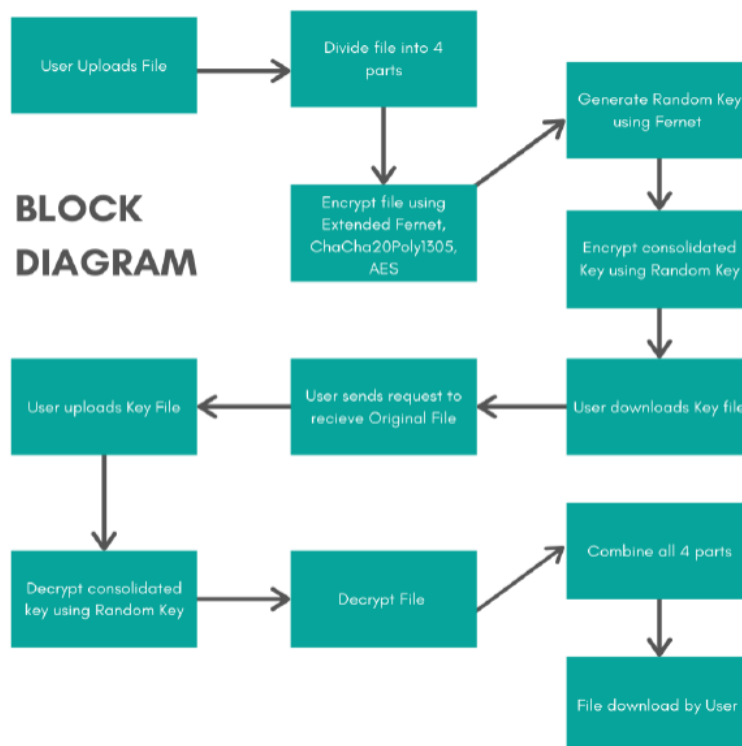


Figure 3.0.1 : Block Diagram

PROCESS FLOW DIAGRAM

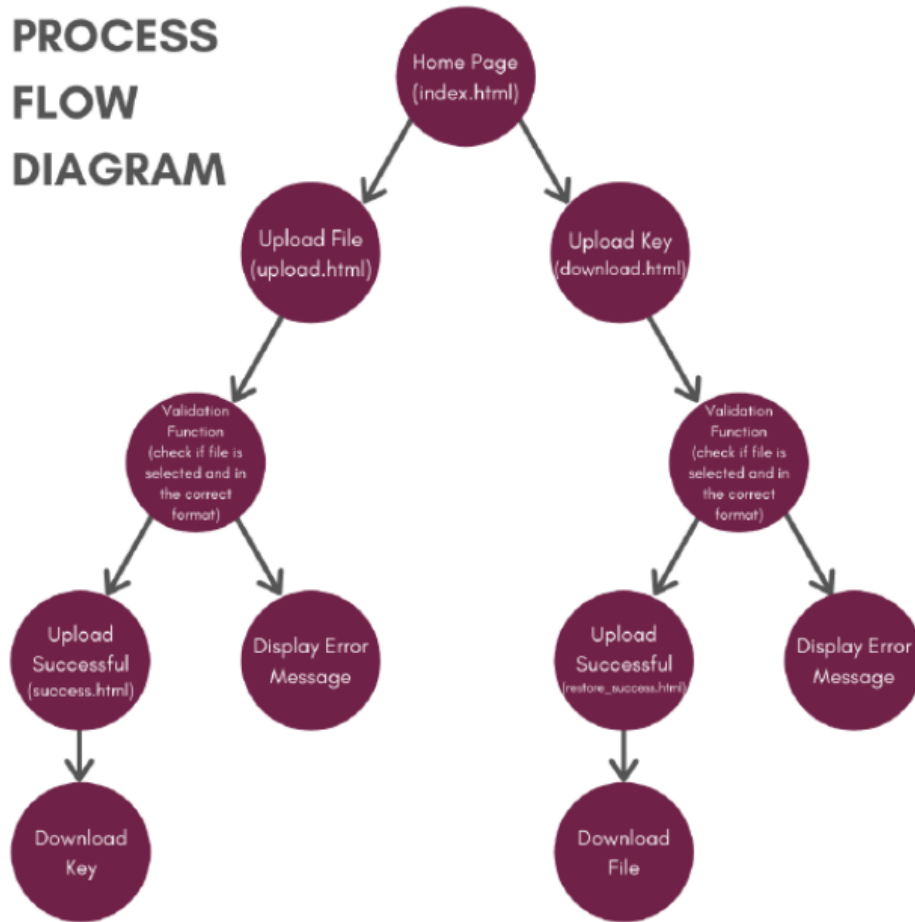


Figure 3.0.2 : Process Flow Diagram

In this project, we first split the file into 5 segments and assign one cryptographic algorithm to each segment. The algorithms we use in this project are:

1. AES CCM.
2. ChaCha20-Poly1305.
3. AES GCM.
4. AES.
5. FERNET.

3.1 AES CCM

CCM mode (counter with cipher block chaining message authentication code; counter with CBC-MAC) is a mode of operation for cryptographic block ciphers. It is an authenticated encryption algorithm designed to provide both authentication and confidentiality. CCM mode is only defined for block ciphers with a block length of 128 bits.

The nonce of CCM must be carefully chosen to never be used more than once for a given key. This is because CCM is a derivation of counter (CTR) mode and the latter is effectively a stream cipher.

DATA INPUT:

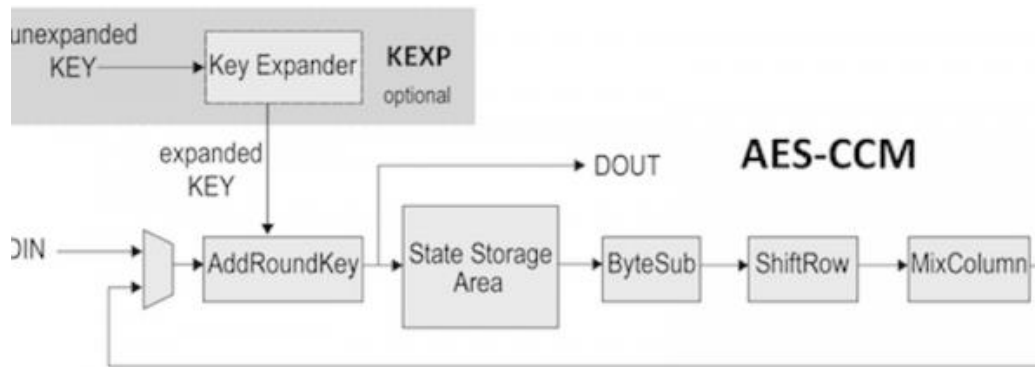


Figure 3.1.1 : AES CCM

As the name suggests, CCM mode combines counter (CTR) mode for confidentiality with cipher block chaining message authentication code (CBC-MAC) for authentication. These two primitives are applied in an "authenticate-then-encrypt" manner: CBC-MAC is first computed on the message to obtain a message authentication code (MAC), then the message and the MAC are encrypted using counter mode. The main insight is that the same encryption key can be used for both, provided that the counter values used in the encryption do not collide with the (pre-)initialization vector used in the authentication. A proof of security exists for this combination, based on the security of the underlying block cipher.

CCM requires two block cipher encryption operations on each block of an encrypted-and-authenticated message, and one encryption on each block of associated authenticated data. According to Crypto++ benchmarks, AES CCM requires 28.6 cycles per byte on an Intel Core 2 processor in 32-bit mode.

3.2.2. ChaCha20-Poly1305

ChaCha20-Poly1305 is an authenticated encryption with additional data (AEAD) algorithm, that combines the ChaCha20 stream cipher with the Poly1305 message authentication code. Its usage in IETF protocols is standardized in RFC 8439. It has fast software performance, and without hardware acceleration, is usually faster than AES-GCM.

The ChaCha20-Poly1305 algorithm as described in RFC 8439 takes as input a 256-bit key and a 96-bit nonce to encrypt a plaintext, with a ciphertext expansion of 128-bit (the tag size). In the ChaCha20-Poly1305 construction, ChaCha20 is used in counter mode to derive a key stream that is XORed with the plaintext. The ciphertext and the associated data is then authenticated using a variant of Poly1305 that first encodes the two strings into one.

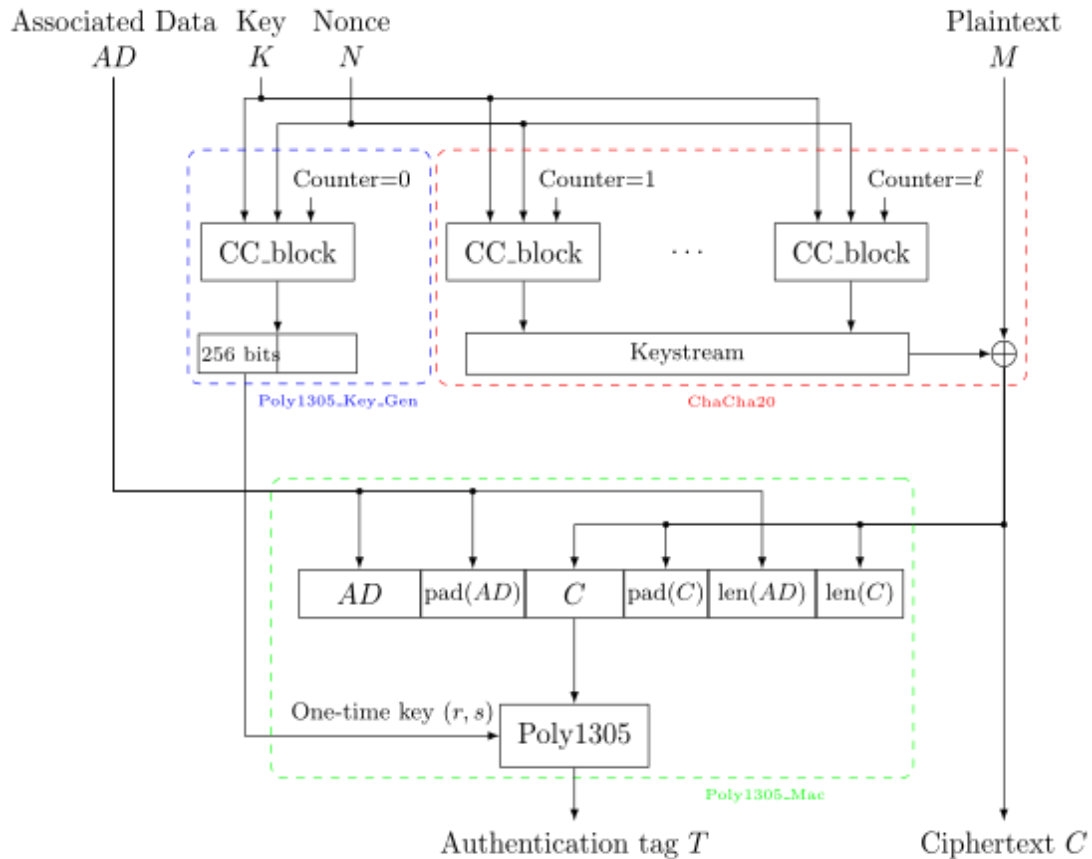


Figure 3.2.1 : ChaCha20-Poly1305

ChaCha20-Poly1305 usually offers better performance than the more prevalent AES-GCM algorithm on systems where the CPU(s) does not feature the AES-NI instruction set extension.[2] As a result, ChaCha20-Poly1305 is sometimes preferred over AES-GCM due to its similar levels of security and in certain use cases involving mobile devices, which mostly use ARM-based CPUs.

3.3 AES GCM

In cryptography, Galois/Counter Mode (GCM) is a mode of operation for symmetric key cryptographic block ciphers which is widely adopted for its performance.

GCM throughput rates for state-of-the-art, high-speed communication channels can be achieved with inexpensive hardware resources. The GCM algorithm provides both data authenticity (integrity) and confidentiality and belongs to the class of authenticated encryption with associated data (AEAD) methods.

Like in normal counter mode, blocks are numbered sequentially, and then this block number is combined with an initialization vector (IV) and encrypted with a block cipher E , usually AES. The result of this encryption is then XORed with the plaintext to produce the ciphertext. Like all counter modes, this is essentially a stream cipher, and so it is essential that a different IV is used for each stream that is encrypted.

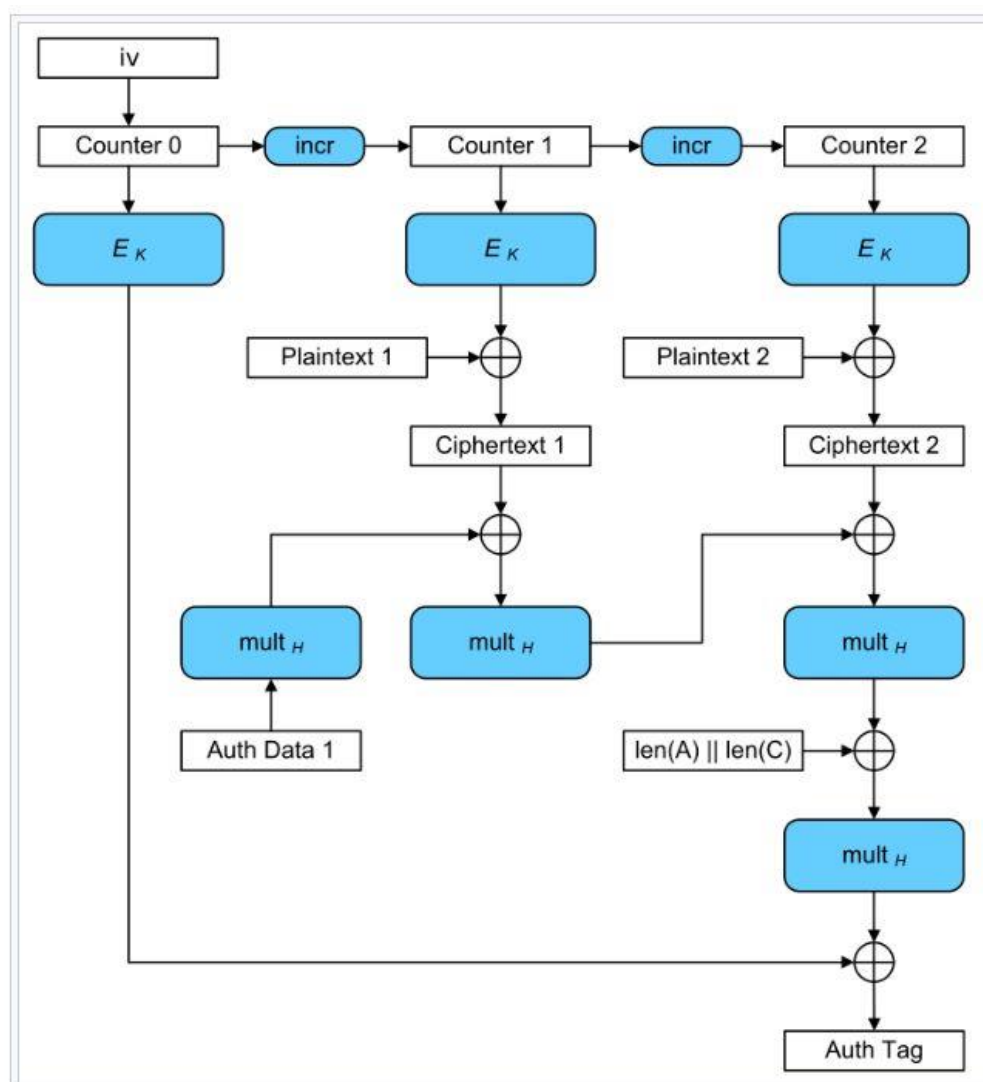


Figure 3.3.1 : AES GCM

3.4 AES

- AES uses bytes rather than bits to perform operations. The cipher handles 128 bits (or 16 bytes) of incoming data at a time since the block size is 128 bits.
- The number of rounds always depends on the key length:
 - 128-bit key – 10 rounds
 - 192-bit key – 12 rounds
 - 256-bit key – 14 rounds

Creation of Round keys

- To calculate all the round keys from the key, a Key Schedule algorithm is employed. As a result, the initial key is used to generate a number of other round keys, each of which will be used in the encryption round that follows. A diagrammatic representation of the same is shown on the next slide.

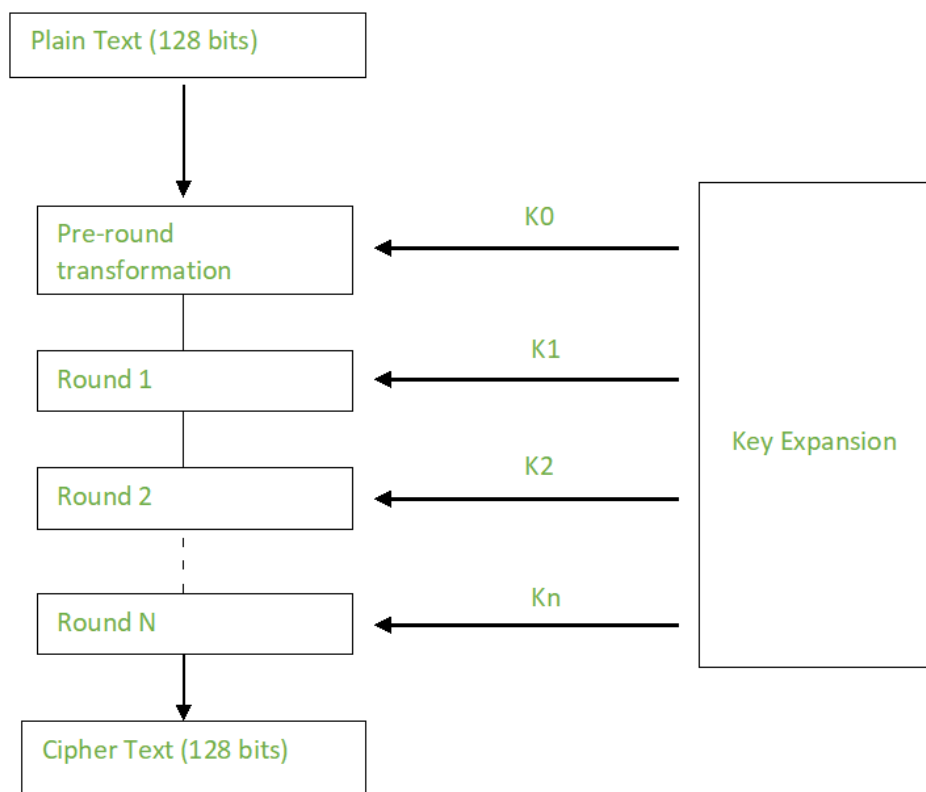


Figure 3.4.1 : AES Key Generation

Encryption:

- In a column major structure, AES treats each block as a 16-byte (4 bytes x 4 bytes = 128) grid.
- Each round consists of 4 steps:
 - ❑ SubBytes
 - ❑ ShiftRows
 - ❑ MixColumns
 - ❑ Add Round Key
- The MixColumns round is not included in the final round.
- In the algorithm, SubBytes does the substitution, while ShiftRows and MixColumns perform the permutation.
- The diagrammatic representation of the Encryption in AES is shown on the next slide.
- After all of the rounds, the output is 128 bits of encrypted data. This method is repeated until all of the data that needs to be encrypted has been encrypted.

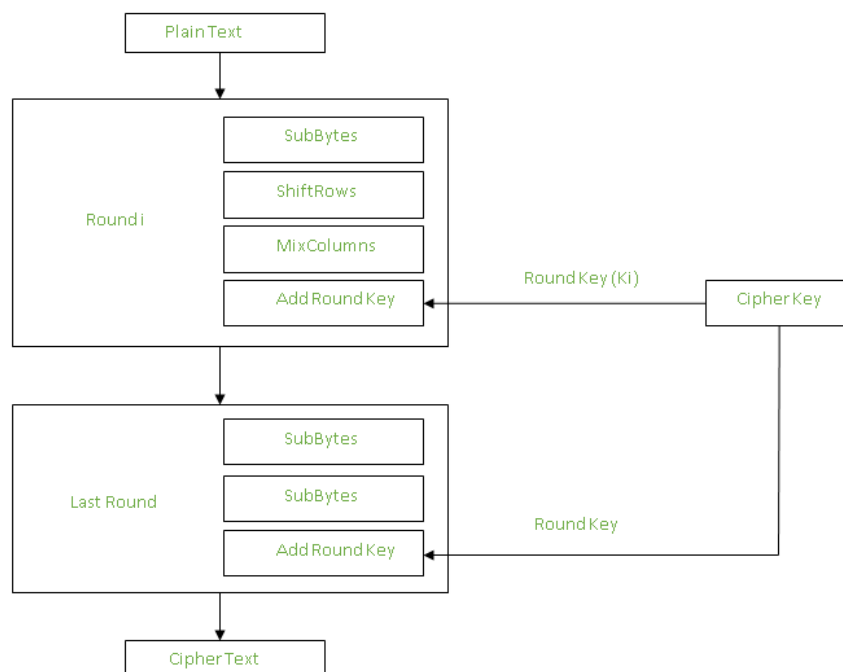


Figure 3.4.2 : AES Encryption

Decryption:

- The rounds' phases can be readily undone because they have an opposite that, when completed, reverses the modifications. Depending on the key size, each of the 128 blocks goes through 10, 12, or 14 rounds.
- The stages of each round in decryption are:
 - ☐ Add round key
 - ☐ Inverse MixColumns
 - ☐ ShiftRows
 - ☐ Inverse SubByte

3.5 FERNET

Fernet guarantees that a message encrypted using it cannot be manipulated or read without the key. Fernet is an implementation of symmetric (also known as “secret key”) authenticated cryptography.

- Fernet is built on top of a number of standard cryptographic primitives. Specifically it uses AES in CBC mode with a 128-bit key for encryption; using PKCS7 padding
- HMAC using SHA256 for authentication.
- Initialization vectors are generated using `os.urandom()`.

Key Format

A fernet key is the base64url encoding of the following fields:

Signing-key || Encryption-key

- Signing-key, 128 bits
- Encryption-key, 128 bits

A fernet token is the base64url encoding of the concatenation of the following fields:

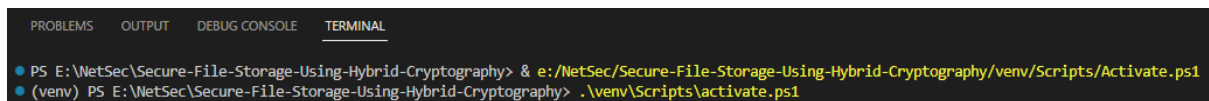
Version || Timestamp || IV || Ciphertext || HMAC

- Version, 8 bits
- Timestamp, 64 bits
- IV, 128 bits
- Ciphertext, variable length, multiple of 128 bits
- HMAC, 256 bits

Given a key and message, generate a fernet token with the following steps, in order:

1. Record the current time for the timestamp field.
2. Choose a unique IV.
3. Construct the ciphertext:
 1. Pad the message to a multiple of 16 bytes (128 bits).
 2. Encrypt the padded message using AES 128 in CBC mode with the chosen IV and user-supplied encryption-key.
4. Compute the HMAC field as described above using the user-supplied signing-key.
5. Concatenate all fields together.

5. Experiments and Results



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS E:\NetSec\Secure-File-Storage-Using-Hybrid-Cryptography> & e:/NetSec/Secure-File-Storage-Using-Hybrid-Cryptography/venv/Scripts/Activate.ps1
(venv) PS E:\NetSec\Secure-File-Storage-Using-Hybrid-Cryptography> .\venv\Scripts\activate.ps1
```

Figure 5.1 : How to activate a virtual environment

Secure File Storage Using Hybrid Cryptography

[Upload](#) | [Restore](#)

A Project for:
Network Security
B.Tech (CSE)

Figure 5.2 Homepage of Cloud Storage PLatform

Secure File Storage Using Hybrid Cryptography

SUCCESS

[Download Key](#)

[Back to HOME](#)

A Project for:

**Network Security
B.Tech (CSE)**

Figure 5.3 Succesfully Uploaded File and Key Download Available

Secure File Storage Using Hybrid Cryptography

SUCCESS

[Download File](#)

[Back to HOME](#)

A Project for:

**Network Security
B.Tech (CSE)**

Figure 5.4 Download file option available

6. Conclusion

In conclusion, implementing a secure hashing algorithm is an essential step to ensure the security of cloud storage. The use of strong encryption algorithms such as AES CCM, AES GCM, ChaChaPoly1315, and Fernet, can enhance the security of data in transit and at rest. Each of these algorithms has its strengths and weaknesses, and the choice of which one to use will depend on the specific security requirements of the project.

AES CCM is a block cipher mode of operation that provides both confidentiality and authenticity of data. It is efficient and has low computational overhead, making it suitable for use in resource-constrained environments.

AES GCM, on the other hand, combines the AES block cipher with the Galois/Counter Mode (GCM) of operation to provide both confidentiality and integrity. It is a popular choice for secure communication protocols and has been standardized by the National Institute of Standards and Technology (NIST).

ChaChaPoly1315 is a stream cipher algorithm that provides both confidentiality and integrity. It is a lightweight algorithm that is suitable for use in resource-constrained environments, and it has been adopted as a standard by the Internet Engineering Task Force (IETF).

Fernet is a simple symmetric encryption algorithm that uses the AES cipher in CBC mode and provides both confidentiality and integrity. It is a secure and easy-to-use algorithm that is suitable for encrypting small amounts of data.

Overall, the use of a secure hashing algorithm and strong encryption algorithms can provide robust security for cloud storage. It is essential to choose the right algorithm that meets the specific security requirements of the project and to ensure that the implementation is properly configured and tested to prevent any vulnerabilities or weaknesses.

References

1. Devang Pratap Singh, Prakarsh Kaushik, Manjari Jain, "Data Storage Security Issues in Cloud Computing", 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM), 2021 IEEE, January 2022.
2. G. Nagarajan, Dr. K. Sampath Kumar, "Security Threats and Challenges in Public Cloud Storage", 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021 IEEE, June 2021.
3. Aman Singh, Shivashankar Reddy Ginni, Dr. Advin Manhar, "Securing File Storage on the Cloud using Cryptography", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 10, Issue 4, April 2021.
4. Praveen Kumar Kollu, Monika Saxena, Khongdet Phasinam, Thanwamas Kassanuk, Malik Mustafa, "Blockchain Techniques for Secure Storage of Data in Cloud Environment", Turkish Journal of Computer and Mathematics Education, Vol.12 No. 11(2021), 1515-1522, May 2021.

5. P. Bharathi, G. Annam, J. B. Kandi, V. K. Duggana and A. T., "Secure File Storage using Hybrid Cryptography," 2021 6th International Conference on Communication and Electronics Systems (ICCES), 2021, pp. 1-6.
6. Pan Yang, Neal N. Xiong, Jigli Ren,"Data Security and Privacy Protection for Cloud Storage: A Survey", IEEE Access,vol 4, pp(99) : 1-1, July 2020.
7. Bharati Mishra, Debsish Jena, "Security of Cloud Storage: A Survey", 2019 International Conference on Information Technology (ICIT),2019 IEEE, March 2020.
8. Bijeta Seth, Surjeet Dalal, Da-Nhuong Le, Vivek Jaglan, Neeraj Dahiya, Akshat Agrawal, Mayank Mohan Sharma, Deo Prakash, K.D. Verma, "Secure Cloud Data Storage System Using Hybrid Paillier-Blowfish Algorithm", Computers, Materials and Continua (CMC), Vol. 67, no. 1, November 2020.
9. Manoj V. Brahme, Dr. Milind V. Sarode, Dr. Meenakshi S. Arya, "Design and Implementation of Secure File Storage using Distributed Cloud Mechanism", International Journal of Research and Analytical Reviews (IJRAR) , Vol. 6, Issue 1, February 2019.
10. S. Pavithra, S. Ramya, Soma Prathibha, "A Survey On Cloud Security Issues and Blockchain", 3rd International Conference on Computing and Communication Technologies(ICCCT), 2019 IEEE, 136-140, 2019. `
11. S. A. Ahmad and A. B. Garko, "Hybrid Cryptography Algorithms in Cloud Computing: A Review," 2019 15th International Conference on Electronics, Computer and Computation (ICECCO), 2019, pp. 1-6
12. Chauhan, A. and Gupta, J., 2017, September. A novel technique of cloud security based on hybrid encryption by Blowfish and MD5. In 2017 4th International conference on signal processing, computing and control (ISPCC) (pp. 349-355). IEEE.
13. Chhabra, A. and Arora, S., 2017, October. An elliptic curve cryptography based encryption scheme for securing the cloud against eavesdropping attacks. In 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC) (pp. 243-246). IEEE.
14. Kanatt, S., Jadhav, A. and Talwar, P., 2020. Review of Secure File Storage on Cloud using Hybrid Cryptography. International Journal of Engineering Research & Technology (IJERT).
15. Swarna, C. and Eastaff, M.S., 2018. Secure File Storage in Cloud Computing Using Hybrid Cryptography Algorithm. Iaetsd Journal for Advanced Research in Applied Science.

16. Maitri, P.V. and Verma, A., 2016, March. Secure file storage in cloud computing using hybrid cryptography algorithm. In 2016 international conference on wireless communications, signal processing and networking (WiSPNET) (pp. 1635-1638). IEEE.