Fortidrop: Cloud-based secure file-sharing tool using Cryptography

Sumit Bakane[1], Abhijeet Lande[2], Prof. Vinayak Sathe[3]
[1,2]B.Tech Student School of Computer Science Engineering and Applications, DYPIU Pune
[3]Professor of Practice at DYPIU Pune
D Y PATIL INTERNATIONAL UNIVERSITY, AKURDI, PUNE

Abstract

The increasing prevalence of data breaches and cyber risks has made secure file sharing a crucial requirement in today's digital world. A cloud-based file-sharing solution has been developed, employing various cryptographic techniques to address this. This tool ensures the secrecy, integrity, and authenticity of shared files. It utilizes symmetric encryption to encrypt file contents, allowing only authorized users with the decryption key to access the file. Asymmetric encryption is used to securely share the decryption key, while digital signatures verify the integrity and authenticity of transferred files. Robust authentication technologies, such as multi-factor authentication, and role-based access control are implemented to ensure authorized user access and manage permissions. End-to-end encryption further enhances security by encrypting files on the sender's device and decrypting them on the recipient's device without intermediary server involvement. This technology significantly improves the security and privacy of shared files, reducing the risk of data breaches and unauthorized access.

Keywords - Cyber Security, File Sharing, MFA Cryptography, Symmetric Encryption, Cloud Computing

Introduction

The ever-evolving nature of network applications has led to a growing need for a solution that is scalable, reusable, interoperable, and easily deployable. Web services have emerged as a highly acclaimed solution in both industry and academia to fulfill these requirements[1,2]. With the rapid progress of wireless networks and mobile devices, individuals now have seamless access to a diverse range of services at any time and from any location. Among the various applications, file sharing stands out as a commonly used one. In the past, file sharing was predominantly carried out using traditional protocols such as FTP (File Transfer Protocol) or NFS (Network File System)[3].

Cloud computing enables the sharing of resources, information, and software with computers and other devices based on user demand[4], you can use a cloud provider like Amazon Web Services (AWS), Google Cloud, or Microsoft Azure to have access to technological services like computing power, storage, and databases as needed[5]. Cloud-based computing has become widely adopted across various organizations. However, there is a rising concern regarding the safety and confidentiality of data in the cloud. Users often lack control over the security measures and processes implemented to protect their data[6]. Globally, organizations have shown a keen interest in leveraging information technology to safeguard their valuable data and protect critical assets[7]. The decision has been made to ensure the confidentiality and simplicity of message transmission through the use of cryptography. Cryptography involves employing a set of algorithms to encrypt and decrypt data[8].

Cryptography

Cryptography is a field dedicated to safeguarding data by transforming it into an unreadable format, allowing authorized users to access information securely at the intended destination. It employs mathematical techniques to encrypt and decrypt data effectively[13]. This encrypted data can only be restored to its original form by decrypting it with the same secret key on the recipient's end[9]. Access to the encrypted data and the authority to decrypt it are limited to individuals possessing knowledge of the secret key. The fundamental components of any cryptographic process include the original data, secret key, encryption algorithm, encrypted data (cipher), and decryption algorithm[10]. Cryptography can be broadly classified into two main types: symmetric (also known as private or single-key cryptography) and asymmetric (also known as public key cryptography)[11]. The regular flow of a cryptography process is depicted in [Fig.1], illustrating the sequential steps involved in encrypting and decrypting data.

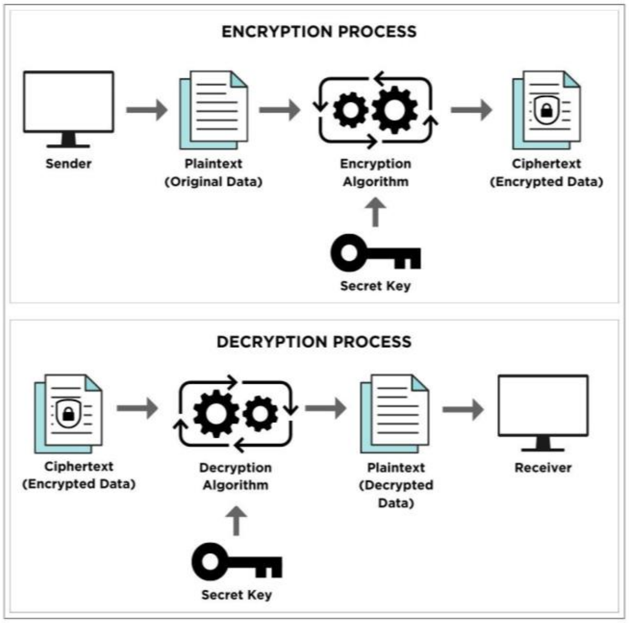


Fig.1 Simple Cryptography Process

I. Symmetric Key Encryption

In symmetric key cryptography, a single key is utilized for data encryption and decryption. When a person encrypts the data, they share the encryption key with the intended recipient who uses it to decrypt the ciphered information. Examples of symmetric key encryption processes include the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). In cloud services, providers often employ a single key for encryption and decryption algorithms to safeguard user data. (Fig.2) illustrates the flow of the symmetric key approach[12].

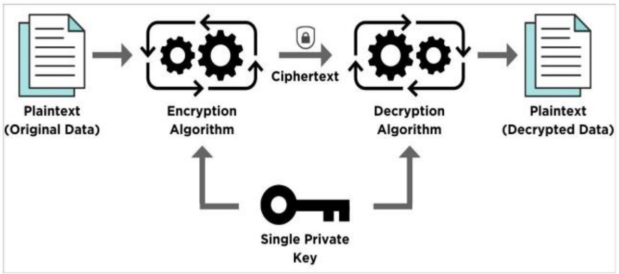


Fig.2 Symmetric Key Encryption

II. Asymmetric Key Encryption

In asymmetric key cryptography, two distinct keys are employed for the encryption and decryption of data. A public key encrypts plaintext information and is accessible to all parties involved in the communication. On the other hand, a private key is exclusively used for decrypting the ciphered information and is only known to the intended recipient. While the public and private keys are mathematically related, possessing knowledge of the public encryption key alone is insufficient for determining the private decryption key. Examples of asymmetric encryption algorithms include RSA (Rivest-Shamir-Adleman) encryption and the Diffie-Hellman algorithm. (Fig.3) illustrates the flow of the asymmetric key methodology[12].

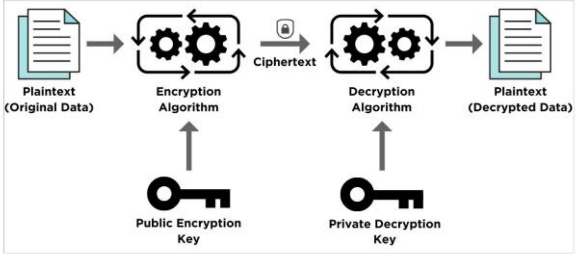


Fig.3 Asymmetric Key Encryption

Methodology

It is an AES algorithm-based application. In our proposed method, encryption and decryption of data are performed using keys. The process of securing data involves several steps, starting with encoding the plaintext into binary format. This conversion ensures that the data can be processed and manipulated in its fundamental binary representation. Once the data is in binary format, encryption operations are applied using secret encryption keys, resulting in the ciphertext. Ciphertext is the encrypted form of the data, which is not easily readable or understandable without the appropriate decryption keys.

To retrieve the original data, decryption operations are performed on the binary ciphertext, utilizing the correct decryption keys. These operations reverse the encryption process, transforming the ciphertext back into the original plaintext.

At this stage, the data is still in binary format. To view the file content in a human-readable form, additional operations such as Base64 decoding are applied to the decrypted binary data.

In summary, the overall process involves encoding the plaintext into binary format, encrypting the binary data using encryption keys to generate ciphertext, decrypting the ciphertext using decryption keys to obtain the original binary plaintext, and finally applying Base64 decoding to view the file content in a readable format.

As a result, the total size and quality of the data remain unaltered throughout the encryption and decryption process, preserving the file's characteristics. (Fig.4) outlines the processing model for file encryption and decryption.

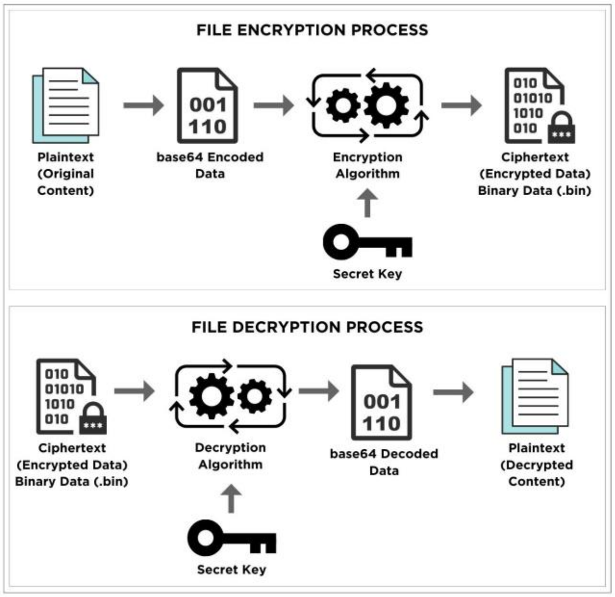


Fig.4 Model for file encryption and decryption

The application also provides a sharing feature that allows users to specify the recipients with whom they want to share the encrypted file. The sharing service ensures that only authorized users can access the file and decrypt its contents.

I. AES Implementation

The AES algorithm consists of three main components: encryption, decryption, and key expansion. During key expansion, a derived key schedule is generated from the secret key, which is then utilized in the encryption and decryption processes. AES is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in 128-bit blocks. The number of rounds in the algorithm is determined by the size of the cryptographic key, with 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key.

The AES algorithm utilizes a 4x4 bytes matrix called the "state." It employs four distinct transformations, namely Sub Bytes, Shift Rows, Mix Columns, and Add Round Key, which is sequentially applied to the data blocks in a series of rounds. These rounds consist of repetitive application of the transformations on the state matrix. The Sub Bytes, Shift Rows, Mix Columns, and Add Round Key operations play crucial roles in the encryption and decryption process of the algorithm. AES implementation is shown in (Fig.5).

A. SubByte

The SubBytes transformation in AES is a non-linear byte substitution function. It involves replacing each byte of the state matrix by referencing a substitution table called the S-box. The S-box is generated through the multiplicative inverse operation in a finite field.

B. ShiftRows

The ShiftRows transformation in AES is a permutation function. It performs a leftward shift on each row of the state matrix, with the offset for each row determined by its line number.

C. MixColumns

The MixColumns transformation in AES is a mixing function. It operates on the state matrix column by column, where each column's four bytes are combined using an invertible linear transformation. This transformation involves multiplying the state with a constant matrix, resulting in a modified state matrix. The MixColumns function plays a crucial role in achieving diffusion and increasing the cryptographic strength of AES.

D. AddRoundKey

The AddRoundKey transformation in AES is an XOR function. In each round, a sub-key is derived from the main key using the Rijndael key schedule. This sub-key is then XORed with the state matrix, resulting in a modified state matrix for further processing in the algorithm.

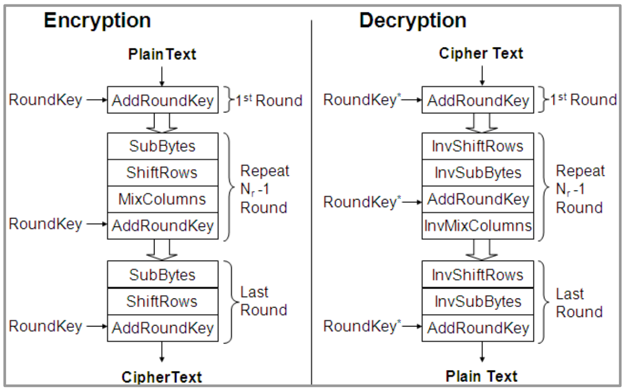


Fig.5 AES implementation model

II. Steps Involved

A. Key selection

The sender and receiver establish a shared 128-bit key for the encryption and decryption of images. This key is used in a symmetric key encryption technique, requiring both parties to securely exchange the key. The key is represented as blocks k[0], k[1], ..., k[15], where each block is 8 bits long. In total, the key consists of 16 blocks, resulting in a 128-bit key.

B. Generation of Multiple Keys

Using the Modified AES Key Expansion technique explained above, the sender and receiver have the capability to independently generate the necessary keys for the encryption and decryption process. This key generation is a one-time process, and the resulting expanded keys can be utilized for future communications indefinitely, as long as the initial key value remains unchanged.

C. Encryption

In the encryption process, we divide the image into spans, with each span consisting of 16 pixels. For each set of pixels, we perform two XOR operations using the expanded key and apply the SubBytes Transformation. By performing these XOR operations with the expanded key, it becomes infeasible to retrieve the key from the plain image and cipher image. To enhance non-linearity, we incorporate the s-box values utilized in AES.

D. Decryption

The decryption process closely resembles encryption, but with a few differences. We utilize the Inverse SubByte Transformation instead of the regular SubBytes Transformation. Additionally, the order of the XOR operations using the expanded key is reversed compared to encryption. These modifications allow us to effectively decrypt the cipher image and retrieve the original plain image using the same expanded key.

III. Cloud Implementation

The web application data is hosted on the AWS Cloud, leveraging its scalable and reliable infrastructure. The database instance is running in a private subnet, ensuring that it is not directly accessible from the internet. This setup adds an additional layer of security by isolating the database from potential external threats.

To facilitate user engagement and provide a seamless frontend experience, an EC2 instance is deployed within a public subnet. This EC2 instance serves as the frontend server, handling user requests and rendering the application interface. By placing it in a public subnet, the front end remains accessible to users while still benefiting from the security measures provided by AWS.

Furthermore, the cloud architecture incorporates cloud security services like the Web Application Firewall (WAF) and Defender. The WAF acts as a shield, protecting the web application from common web-based attacks, such as cross-site scripting (XSS) and SQL injection. It monitors and filters incoming traffic, ensuring that only legitimate requests reach the application.

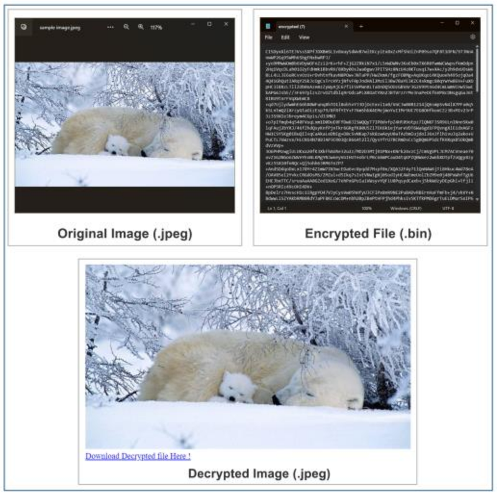
Defender, another cloud security service, enhances the overall security of the system. It provides advanced threat detection and helps identify potential vulnerabilities within the infrastructure. By continuously monitoring the environment, Defender detects suspicious activities or potential security breaches, allowing for proactive mitigation measures.

The combination of hosting the web application data on AWS, utilizing private and public subnets, and integrating cloud security services such as WAF and Defender ensures the CIA triad of information security: confidentiality, integrity, and availability. Confidentiality is maintained by isolating sensitive data in the private subnet, while integrity is safeguarded through the protection mechanisms offered by the WAF and Defender. Availability is ensured by leveraging AWS's reliable infrastructure and scalable services.

Result

The proposed methodology involves the development of a user-friendly web-based application that allows users and owners to easily encrypt and decrypt various types of files. Extensive testing of the method in web-based applications has confirmed its capability to handle essential tasks such as encrypting and decrypting image (.jpeg) files (fig.6), encrypting and decrypting PDF files(.pdf) (fig.7), and encrypting and decrypting word document (.docx) files (fig.8).

The application provides a sharing feature that allows users to specify the recipients with whom they want to share the encrypted file. The sharing service ensures that only authorized users can access the file and decrypt its contents.



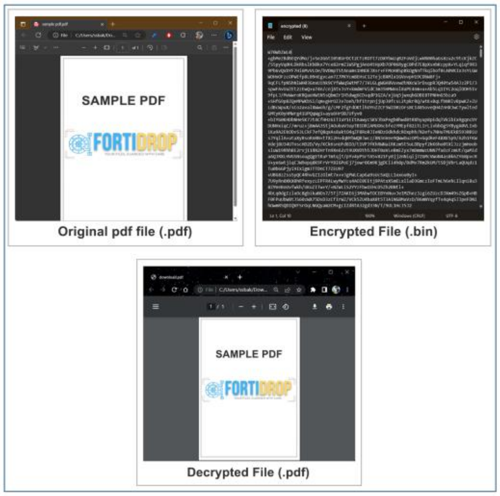
Original Image (.jpeg)

Encrypted File (.bin)

Download Decrypted file Here !

Decrypted Image (.jpeg)

Fig.6 Encryption and Decryption of Image



Original pdf file (.pdf)

Encrypted File (.bin)

Decrypted File (.pdf)

Fig.7 Encryption and Decryption of Pdf

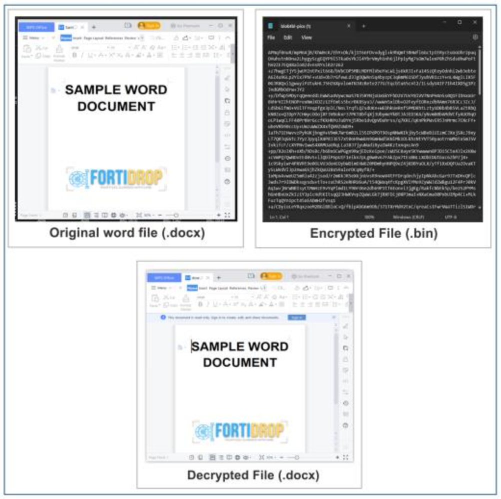


Fig.8 Encryption and Decryption of Word

Conclusion

This paper presents the development of an application that emphasizes ensuring confidentiality for files shared over the internet. The proposed method involves converting user files into binary format and enabling users to download and decrypt them using an encryption key provided by the file owner. The application's authenticity and effectiveness have been successfully verified, particularly for small-scale usage. The method adopts a straightforward and user-friendly approach to ensure ease of use.

References

[1] Roy T. Fielding and Richard N Taylor, "Principled Design of the Modern Web Architecture", ACM Transactions on Internet Technology, vol. 2 no. 2, May 2002, pp. 115–150

[2] Douglas K. Barry, “Web Services, Service-Oriented Architectures, and Cloud Computing: The Savvy Manager's Guide”, 2nd ed., Morgan Kaufmann, 2013.

[3] Chin-Chih Chang, Wen-Xiang Wu, “Digital Information and Communication Technology and it's Applications (DICTAP)”, 2014 Fourth International Conference.

[4] Veerraju Gampala, Muppidi Satish, “Data Security in Cloud Computing with Elliptic Curve Cryptography”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3, July 2012

[5] A Comparative Analysis of Public Cloud Platforms and Introduction of Multi-Cloud [Khot, A. R. (2020). A Comparative Analysis of Public Cloud Platforms and Introduction of Multi-Cloud. In International Journal of Innovative Science and Research Technology (Vol. 5, Issue 9). www.ijisrt.com]

[6] Nagababu Garigipati, Dr. Krishna Reddy V, “A Study on Data Security and Query Privacy in Cloud”, Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE Xplore Part Number: CFP19J32-ART; ISBN: 978-1-5386-9439- 8

[7] Nur Syafiqah Mohd Shamsuddin and Sakinah Ali Pitchay, “LocationBased Cryptographic Techniques for Data Protection”, MJoSHT 2019, Volume 4, Special Issue, eISSN: 2601-0003

[8] Volume 4, Special Issue, eISSN: 2601-0003 [5] Mrs. Sumitra Samal, Abhishek Tandon, “A New Efficient Digital Signature Scheme Algorithm Based On Block Cipher”, Journal of the Gujarat Research Society, ISSN: 0374-8588 Volume 21 Issue 17, December 2019

[9] S. Gunavathy, and C. Meena, "A Survey: Data Security In Cloud Using Cryptography And Steganography". International Research Journal of Engineering and Technology, Vol.6, No. 5, pp. 6792- 6797, 2019

[10] Hashem and H. Ramadan, “Using Cryptography Algorithms to Secure Cloud Computing Data and Services”, Amer. J Eng. Res. (AJER), vol. 6, no. 10, pp.334-337, 2017.

[11] Y. Peng, et al, "Secure Cloud Storage Based on Cryptographic Techniques", J China Univer. Posts Telecomm, Vol. 19, pp. 182- 189, 2012. Doi: 10.1016/s1005-8885(11)60424-x.

[12] Samer A. Noah, “ Cloud Cryptography: User End Encryption”, 2020 International Conference on Computing and Information Technology, Volume: 01, Issue: ICCIT-1441, Page No.: 397 - 400, 9th & 10th Sep. 2020.

[13] Gurdeep Singh, Prateek Kumar, Nishant Taneja, Gurpreet Kaur, “A Research Paper On Cryptography”, International Journal For Technological Research In Engineering Volume 7, Issue 4, December-2019