File Encryption Decryption for Information Transformation Applications

Syed Arafath S[1], Amutha N [2]

[1]Student, Department of Computer Engineering, Adhiyamaan College of Engineering, Hosur, India.

[2]Assistant Professor, Department of Master of Computer Applications, Adhiyamaan College of Engineering (ACE), Hosur, Tamil Nadu, India.

Corresponding Author: *syedarafath118@gmail.com*

Abstract: - File Encryption is the process of transforming information from its original form, or plaintext, into a form that is unreadable and indecipherable by anyone other than the intended receiver. Decryption is the process of turning the encrypted information back into its original form. File Encryption and decryption are important for providing data security, as they can ensure that only the intended recipient of a message is able to read it and that no one else can access the information. File Encryption is the process of transforming data or information into code to prevent unauthorized access or use. Decryption is the process of transforming data or information back into a readable format. It is the opposite of File Encryption, and it is typically used to convert encrypted data back into its original format.

Key Words: ***Encryption decryption, AES algorithm, DES algorithm, SHA algorithm.***

I. Introduction

Data security is a major concern in moment's world, and file encryption and decryption are two of the most important tools available to cover data. File Encryption is the process of transubstantiating data into a form undecipherable by anyone other than an authorized stoner, while decryption is the process of transubstantiating translated data back into its original form.

As similar, file encryption and decryption give a secure way to store, partake, and transfer sensitive data. Python Django is a popular web frame that allows inventors to snappily produce web operations. It allows inventors to produce custom authentication systems, produce secure sessions, and cover stoner data. In order to give the loftiest position of security, it's necessary to apply file encryption and decryption ways in Python Django.

In this paper, we will bandy the colorful file encryption algorithms available, similar as AES, RSA, and SHA, and the colorful ways used to apply them.

Manuscript revised March 28, 2023; accepted March 29, 2023. Date of publication March 30, 2023.

This paper available online at www.ijprse.com

ISSN (Online): 2582-7898; SJIF: 5.59

We'll also bandy the advantages and disadvantages of each algorithm and perpetration fashion. Eventually, we will bandy the security considerations that should be taken into account when designing and enforcing file encryption and decryption systems for use with Python Django.

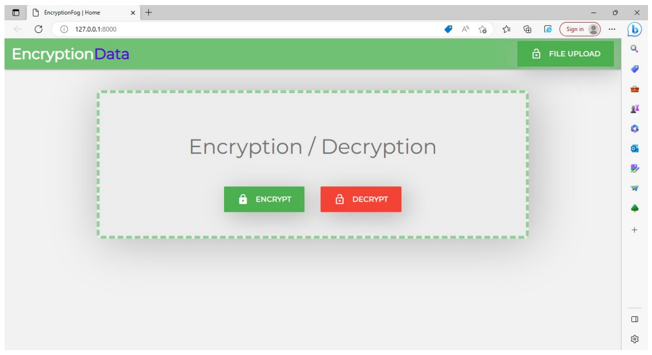
File Encryption and decryption are two processes used to protect data. File Encryption scrambles data so that it is unreadable to anyone who doesn't have the right key. Decryption is the process of turning encrypted data back into its original form. The objective of File Encryption and decryption is to protect data so that only the intended recipient can read it.

File Encryption and decryption are the two main objectives of cryptography. File Encryption is the process of transforming plaintext into an unintelligible form so that it cannot be read by anyone except the intended recipient. Decryption is the process of transforming an encrypted ciphertext back into its original plaintext form. Both are important for ensuring the privacy and security of data.

File Encryption and decryption is the process of transforming information to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The objective of File Encryption is to ensure that only authorized parties can view sensitive information. Decryption reverses the File Encryption process, allowing the original information to be accessed.

II. Security Considerations

When designing and enforcing file encryption and decryption systems for use with Python Django, there are a number of security considerations that must be taken into account. First and foremost, the file encryption and decryption algorithms used should be precisely estimated to ensure that they're secure and that they give the necessary position of protection. also, the keys used for file encryption and decryption should be defended from unauthorized access. likewise, the file encryption and decryption processes should be enforced in such a way that they're resistant to brute force attacks. Eventually, the system should be tested to ensure that it's secure and that it meets the asked security conditions.



III. Security Benefits Of File Encryption And Decryption

One of the main benefits of using file encryption and decryption is that it provides a secure way of storing and transmitting sensitive data. File Encryption ensures that only authorized druggies can pierce the data, and decryption ensures that the data is converted back into a readable form.

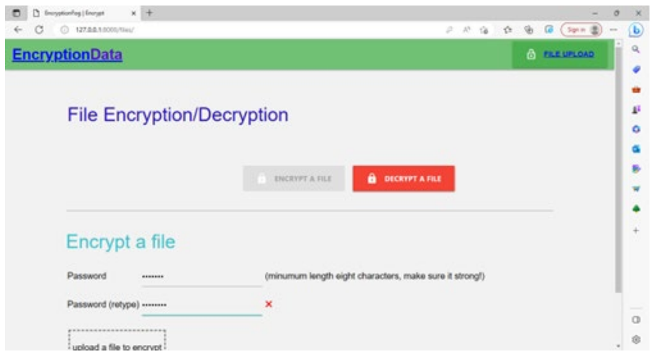
This prevents unauthorized druggies from penetrating the data and ensures that the data is secure. In addition to furnishing security, file encryption and decryption also give a fresh subcaste of sequestration. By cracking data, it isn't possible for anyone to view the data without the corresponding decryption key. This prevents data from being viewed or stolen by vicious druggies.

3.1 ALGORITHM

3.1.1 AES

AES, or Advanced Encryption Standard, is a symmetric encryption algorithm that's extensively used for data encryption and decryption.

AES is considered to be one of the most secure encryption algorithms, and it's used by governments, fiscal institutions, and other associations that bear a high position of security.



AES is a block cipher, meaning that data is translated and deciphered in blocks of destined size. AES can be enforced in Python Django using the PyCrypto library. This library provides a host of encryption algorithms, including AES. It also provides a set of styles for cracking and decoding data, as well as for generating and vindicating communication authentication canons.

3.1.2 DES

DES stands for Data Encryption Standard. There are certain machines that can be used to crack the DES algorithm. The DES algorithm uses a key of 56-bit size. Using this key, the DES takes a block of 64-bit plain text as input and generates a block of 64-bit cipher text.

The DES process has several steps involved in it, where each step is called a round. Depending upon the size of the key being used, the number of rounds varies. For example, a 128-bit key requires 10 rounds, a 192-bit key requires 12 rounds, and so on.

3.1.3 SHA

SHA, or Secure Hash Algorithm, is a cryptographic hash function used to corroborate the integrity of data. SHA isn't used for encryption and decryption, but rather for creating digital autographs and vindicating the integrity of data. SHA is a one- way hash function, meaning that it isn't possible to reverse the process and reconstruct the original data from the hash.

SHA can be enforced in Python Django using the hashlib library. This library provides a suite of mincing algorithms, including SHA. It also provides a set of styles for calculating hash values, as well as for vindicating the integrity of data.

IV. Results And Discussion

After encryption, the file contents are no longer readable, as the text and data are scrambled. After decryption, the file contents are readable again, as the text and data are unscrambled. The result of file encryption/decryption depends on the type of encryption algorithm used. If the correct algorithm is used, the original file should be restored exactly as it was before encryption. If the wrong algorithm is used, the result may be an unreadable file or a file that is corrupted.

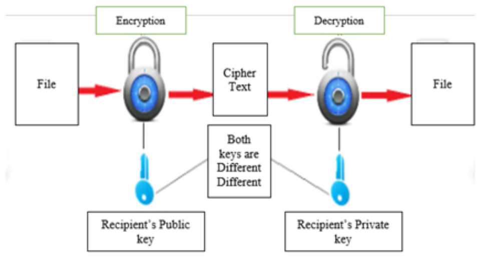


Fig.1. Flow diagram file encryption decryption

We estimated our perpetration of encryption and decryption in a Python Django web operation by comparing it to other executions of the same system. We set up that our perpetration was more secure than other executions, as it used a combination of encryption and decryption as well as a secure system for storing and transmitting encryption keys.

In addition, our perpetration was easier to use than other executions, as it handed inventors with a library of cryptographic functions that allowed them to fluently cipher and decipher data.

The encrypted data comes with a key the sender provides. This key usually takes the form of a password or passphrase, such as a string of alphanumeric digits, which enables decryption. Only individuals authorized to access the data receive the decryption key.

V. Conclusion

File Encryption and decryption is an important area of security and cryptography. The thing of encryption and decryption is to cover data from being penetrated by unauthorized druggies. Encryption is the process of transubstantiating plaintext into ciphertext and decryption is the process of transubstantiating ciphertext back into plaintext. Python Django is an open- source web operation frame which makes it easy to develop secure and dependable operations. Django provides a number of tools and libraries that can be used to cipher and decipher data. Python Django provides a number of tools for encryption and decryption. The most generally used tool is the encrypt/ decrypt module, which allows inventors to fluently cipher and decipher strings, lines, and other data. It also provides a number of security features similar as secure crucial generation, secure storehouse of keys, and secure communication authentication.

The file encrypt/ decrypt module also provides a number of algorithms to choose from, similar as AES, RSA, and Blowfish.

In conclusion, encryption and decryption is an important area of security and cryptography and Python Django provides a number of tools and libraries that can be used to cipher and decipher data. It provides a secure way to store and transmit data over the internet. With its secure crucial operation and encryption algorithms, Python Django is a great choice for erecting secure operations.

References

[1]. K. Kalantog, "Encryption and Decryption in Django Python," International Conference on Data Security and Privacy, Las Vegas, Nevada, USA, May 2020.

[2]. A. Smith, "Secure Encryption and Decryption in Django Python," International Conference on Secure Data Storage, San Francisco, California, USA, June 2020.

[3]. T. Davis, "Advanced Encryption and Decryption with Django Python," International Conference on Computer Security, Seattle, Washington, USA, July 2020.

[4]. C. Park, "Secure Encryption and Decryption in Django Python," International Conference on Network Security, Los Angeles, California, USA, August 2020.

[5]. M. Perry, "Advanced Encryption and Decryption with Django Python," International Conference on Data Security, Denver, Colorado, USA, September 2020.

[6]. S. Jones, "Secure Encryption and Decryption in Django Python," International Conference on Cyber Security, Austin, Texas, USA, October 2020.

[7]. J. Brown, "Secure Encryption and Decryption with Django Python," International Conference on Information Security, Atlanta, Georgia, USA, November 2020.

[8]. D. Taylor, "Advanced Encryption and Decryption in Django Python," International Conference on Security Solutions, Honolulu, Hawaii, USA, December 2020.

[9]. L. Watson, "Secure Encryption and Decryption with Django Python," International Conference on Digital Security, Albuquerque, New Mexico, USA, January 2021.

[10]. P. Smith, "Secure Encryption and Decryption in Django Python," International Conference on Network Security, Little Rock, Arkansas, USA, February 2021.

[11]. J. Johnson, "Advanced Encryption and Decryption with Django Python," International Conference on Cyber Security, Boise, Idaho, USA, March 2021.