***File Encryption Using Noise Images as Key***

***Abstract***

Everything in today's highly connected digital world is increasingly dependent on instantaneous global data transfers. Internet efficiency facilitates our daily lives. Sharing information online presents significant security dangers and difficulties in the modern day. The use of cryptography is the means through which sensitive information can be protected from various threats. Improved cryptosystems technology is a no-brainer for securing communication networks. Our work focuses on developing a more effective cryptosystem, and to that end, we have introduced a new approach to data encryption and decryption. This work proposes a noise picture encryption and decryption strategy in which the noise signal is chosen at random to determine the starting values for a chaotic system, which both increases the system's security and its complexity. for figuring out how well the suggested approach works. The experimental findings validate the effectiveness and suitability of the proposed chaos-based cryptosystem for the safe transfer of information (images).We have created a file encryption system where we save encrypted file and decrypt and download it later. The main difference between the existing and proposed system is the file encryption system is implemented using noise images that stores key values inside it.

***Keywords***— Encrypting, Key, Decryption, Noise picture, Cryptographic

**INTRODUCTION**

Every day, a corporation handles files, so it needs reliable data security methods. 32% of organizations are affected by cybercrime, which costs $6 trillion globally. You need sophisticated security capabilities that interact nicely with your existing platforms and fit within your company budget to mitigate this risk. One of the best security measures is file encryption. With advanced security measures, it protects your business's data. Encrypting a file involves encoding not only the file itself but also any sensitive information that it may contain. This allows the file to be sent via an insecure channel without compromising its integrity. Because of the encoding, nefarious actors are unable to get unauthorized access to the data or modify it in any way. It prevents a file from being accessed by anyone other than the person or individuals for whom it was intended, whether it be one person or multiple people. Encryption of files is a useful solution to this problem. It gives you the ability to make any sensitive information you save unreadable to anybody other than the authorized recipients.

The encryption of files is accomplished by the application of complicated algorithms. A file is considered to be encrypted if an encryption technique has been applied to it in order to jumble the data contained within the file. After the file has been scrambled, it will no longer be readable; however, this will only be a temporary impact. The sender is responsible for providing the decryption key for the data that has been encrypted. This key, which enables decryption, takes the form of a password or passphrase most of the time. For example, it might be a string of alphanumeric numbers. The decryption key is only given to those people who are legally permitted to access the data. The file will become readable once again after the authorized receiver has entered the appropriate password or passphrase. The vast majority of computer operating systems and file systems are equipped with integrated support for file encryption. The system encrypts and stores sensitive data in a secure location, and the decryption key is required to access those files.

Keys are typically required in order to successfully complete the encryption process. Cryptography is the practice of protecting sensitive information by scrambling it and then reassembling it using secret keys. Symmetric and asymmetric cryptography are the two primary forms that it can take. The use of asymmetric cryptography, sometimes known as public-key cryptography, is commonplace in many businesses. These encryption methods make use of a public key in addition to a private key. Anyone who possesses the public key has the ability to utilize it to encrypt files. However, only users who possess the private key can decrypt the data, therefore the files will continue to be protected from access by unauthorized users. The process of encrypting and decrypting data with symmetric encryption only requires the usage of a single private key. To be able to decrypt the encoded files, the two parties that are exchanging information through symmetric encryption need to exchange keys.

In most cases, symmetric encryption can be made to function more quickly and effectively than asymmetric encryption. Whenever an organization has to encrypt a large amount of information, such as an entire database, they typically turn to this method. However, symmetric encryption has the drawback of having a lower level of overall security efficacy. This word indicates that it will likely be more difficult to maintain the secrecy of the key. For instance, if the process of encrypting and decrypting data takes place in various locations, the private key will need to be moved between those sites, which leaves it open to the possibility of being attacked. Programs that are written to encode the data in a variety of predetermined ways are typically used to carry out encryption. Certain sectors have adopted particular encryption standards as their norm. Others are compatible with a limited number of different kinds of databases. The efficiency of the encryption is determined by a number of distinct aspects, including the appropriateness of the encryption system, the efficacy of the algorithm, and the magnitude of the key. In this article, we are going to discuss the process of encrypting data using noise images as the key, and we are also going to have a look at various cases. Experiments have indicated that the noise picture that is proposed to be used as a key for encryption and decryption methods is safe and reliable, suggesting that it may have the potential to be adapted to high-security image communication applications.

**LITERATURE SURVEY**

[1] T. M. K. Afandi, D. H. Fandiantoro, Endroyono and I. K. E. Purnama, "Medical Images Compression and Encryption using DCT, Arithmetic Encoding and Chaos-Based Encryption," 2021 International Seminar on Intelligent Technology and Its Applications (ISITIA), 2021

The scientists designed chaotic sequence-based encryption with arithmetic encoding and discrete cosine transform (DCT) compression to meet the demand for privacy and security in medical picture transmission while simultaneously lowering file size. First, a DCT compression is applied to the picture. Another round of arithmetic encoding compression follows. The encrypted file is then compressed once more before being shuffled and scrambled using two chaotic sequence functions. The PSNR for the suggested method is 41.70 dB, and the compression ratio is 0.748.  
[2] C. Qin, J. Hu, F. Li, Z. Qian and X. Zhang, "JPEG Image Encryption with Adaptive DC Coefficient Prediction and RS Pair Permutation," in IEEE Transactions on Multimedia,2022

JPEG image encryption transforms the original picture into a meaningless blur of noise. Our adaptive prediction approach is originally used to make predictions about DC coefficients. Then, we encrypt the histogram of DC coefficient prediction errors using a mixture of prediction errors and random integers, which helps us to cut down on both the length and size of the encoded file. As an added bonus, we implement the permutation for all DCT blocks besides the DC coefficients, which might modify the image's content, and construct the RS (run/size) pairings in each DCT block. In order to obtain the RS pairings, the top left corner of each DCT block is used. In comparison to other methods, our JPEG picture encryption approach is more secure while still being backwards-compatible with JPEG files.  
[3] M. D and S. Vasuhi, "Image Steganography: 2-Bit XOR Algorithm Used In YCbCr Color Model With Crypto-algorithm," 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2020

Steganography studies how messages are hidden. Steganography sends data in an image that humans cannot see to protect it while sending it through any medium. Image processing combines encryption and steganography in this paper. The YCbCr colour model used in this example is based on a 2-bit XOR LSB image steganography. The recommended method uses 2-bit XOR to conceal information in the Cr component of the colour space after converting a picture from RGB to YCbCr. In our method, we encrypt a text file using a simple cryptographic algorithm and then conceal the resultant picture. Files containing text can be encrypted and decrypted using the Rail Fence Cipher with key 2.The YCbCr image 2-bit XOR substitution technique has been extensively studied. Finally, encryption methods were compared to MSE and PSNR. The proposed solution is secure, high PSNR, and low MSE, making it effective for picture data concealment.

[4] J. Tong, Y. Long and Q. Liu, "A File Encryption System Based on Attribute Based Encryption," 2021 17th International Conference on Computational Intelligence and Security (CIS), 2021

Attribute Based Encryption (ABE) is a type of public-key cryptosystem that is commonly used to protect file data security and achieve fine-grained file sharing. However, existing attribute-based encryption algorithms are very complex and require a large amount of additional cryptographic data, making them difficult to implement. This paper proposes an attribute-based data encryption scheme to achieve secure file storage and access control. This system is simple to set up and operate. The attribute private key in this file encryption system is divided into a server-side secret share and a user-side secret share. When a user is granted access to a file, the client and server decrypt the encrypted file collaboratively using the user-side secret shares and server-side secret shares of the user's attribute private keys.

[5] A. N, A. V. K and N. R, "Sharing Confidential Images with Abbreviated Shares using Steganography and AES Algorithm," 2022 2nd International Conference on Intelligent Technologies (CONIT), 2022

Steganography literally means "covered writing" from the Greek steganos (= "covered") and graphy (= "writing"). Communication is "invisible" to it. There are many steganographic methods for hiding critical information in different file formats, some of which are harder than others and have their own pros and cons. The Least Significant Bit (LSB) embedding method hides the cover file's image in the cover picture's least significant bits. 24-bit and 8-bit settings could use this method. BPCS steganography embeds secret data in vessel bit-planes. We substitute every "noise-like" bit-plane of the vessel image with secret data without degrading image quality. "BPCS-Steganography" is the name of this steganography. LSB, BPCS, and AES steganography are used to disguise the secret image in this research.

[6] H. Nazir, I. S. Bajwa, S. Abdullah, R. Kazmi and M. Samiullah, "A Color Image Encryption Scheme Combining Hyperchaos and Genetic Codes," in IEEE Access, 2022

We employ the 4D-hyperchaotic system to generate three S-boxes (red, green, and blue) and a logistic map to transform a regular picture into DNA strands, since the chaotic range and susceptibility of a single chaotic map is constrained. After that, a fake picture of DNA strands is constructed using a logistic map. The DNA strands are then subjected to DNA operations utilising a logistic map sequence, which yields the genetic code. The decoded strands are encrypted using three substitution boxes (s-boxes). This study employs a cryptanalysis-driven design approach to establish the safety of an encryption protocol. The suggested method is compatible with a wide range of picture file formats and sizes (N M). There has been an exhaustive study of the image in terms of its appearance, key space, key sensitivity, energy, homogeneity, contrast, entropy, histogram, correlation, chosen-plaintext attacks, number of pixels changing per second (NPCR), universal average changing intensity (UACI), mean absolute error, resistance to noise and occlusion, and encryption efficiency. Visual and numerical simulations show that the strategy is secure and reliable.

[7] O. Q. J. Al-Thahab and A. A. Hussein, "Implementation Of Stego-Watermarking Technique by Encryption Image Based On Turbo Code For Copyright Application," 2020 1st. Information Technology To Enhance e-learning and Other Application (IT-ELA), 2020

The amount of digital data exchanged and disseminated via internet channels, notably video, is growing rapidly. Maintaining data copyright was a major concern. Therefore, data protection methods are always under question. Video steganography is a means of hiding data in digital video files that the human visual system cannot recognize. Video watermarking is another way to safeguard digital video's intellectual property. In this work, steganography and watermarking will be used to hide the publisher logo picture inside the digital video file by embedding the logo pixel data in the video frames' LSB. The most prominent video steganography technique, watermarking, is now used by attackers. Therefore, a smarter way to protect digital video copyright is needed. In this paper, Turbo code is used to encrypt the logo image bits and the least significant bit (LSB) approach is used to incorporate the encoded logo pixels inside the cover video frames. This method made the system powerful and secure against hacker attempts. The system's Stego-video has a relative rate of 98% and a PSNR of 57dB, with great robustness against salt and pepper and Gaussian noise.

[8] P. Oktivasari, M. Agustin, R. E. M. Akbar, A. Kurniawan, A. R. Zain and F. A. Murad, "Analysis of ECG Image File Encryption using ECDH and AES-GCM Algorithm," 2022 7th International Workshop on Big Data and Information Security (IWBIS), 2022

Healthcare uses image data. Electrocardiograms are health imaging data (ECG). ECGs include personal data that must be protected. Encryption methods protect ECG image data. This thesis uses ECDH and AES-GCM. ECDH generates AES-GCM encryption and decryption keys. Python handles encryption and decoding. ECG image file size increases encryption and decryption times. Decryption requires checking the nonce value and authentication tag. The encrypted ECG picture file's histogram shows pixel homogeneity. PSNR and SSIM tests show the difference between original and encrypted ECG picture files. The NIST Statistical Test Suite shows that the technique generates random output to safeguard ECG picture files.

[9]  S. Patel and T. V, "New Image Encryption Algorithm based on Pixel Confusion-Diffusion using Hash Functions and Chaotic Map," 2022 7th International Conference on Communication and Electronics Systems (ICCES), 2022

Computer technology has made data privacy and security essential. This study proposes a hash function, chaotic map, and two diffusion process picture encryption techniques. Two hash functions generate chaos map initialization keys. These hash functions start with the plain image's rows, columns, and diagonal pixels. Quantization unit scrambles the image. The initial diffusion step XORs the scrambled image's pixels with the normalized chaotic map. Even and odd pixels are XORed with chaotic maps. A second diffusion process XORs the picture with the map a finite number of times to achieve strong encryption. For strong encryption, the pixel array is circular shifted three times after each round. The suggested picture encryption algorithm can withstand statistical and differential attacks in the communication channel, according to state-of-the-art experimental and comparative analyses.

[10] M. E. Kahla, M. Beggas, A. Laouid, M. Kara and M. AlShaikh, "Asymmetric Image Encryption Based on Twin Message Fusion," 2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP), 2021

Encryption is one of the most popular techniques for securing personal information after the digital revolution, including photographs. The speed of encryption and decryption, the size of image encryption, and Safeway's key sharing are only a few of the problems that remain. Based on Magic Number Fragmentation and El-Gamal Encryption, this paper presents a novel image asymmetric encryption scheme (MNF-G). The plain image is converted first into bits, then into decimal, and then into MNF-G. The results of the experiments demonstrate that the proposed method can encrypt/decrypt images with a strong encryption effect. By using this method, a whole distinct image file that can be sent securely over the internet is created. This design increases security.

**TABLE OF COMPARISON**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | TITLE | PROS | CONS |
| [1] | "Medical Images Compression and Encryption using DCT, Arithmetic Encoding and Chaos-Based Encryption" | The proposed method achieves a compression ratio of 0.748 and a peak signal-to-noise ratio (PSNR) of 41.70 dB. | Since chaotic encryption is more sequential, it takes more time than AES encryption, which employs a Matrix. |
| [2] | "JPEG Image Encryption with Adaptive DC Coefficient Prediction and RS Pair Permutation" | Our technique maintains JPEG format compatibility for encrypted images, a tiny file size increase, and greater security. | Future solutions can be found for privacy protection in compressed video data such as H.264/AVC, H.265/HEVC, and H.266/VVC. |
| [3] | "Image Steganography: 2-Bit XOR Algorithm Used In YCbCr Color Model With Crypto-algorithm" | The approach produces the best stego-images when PSNR is above 40 dB. | The output is affected by the quality of the input images. |
| [4] | "A File Encryption System Based on Attribute Based Encryption" | The paper's technique requires less cryptographic data and is easier to implement. | Several enhancements can be made to the aspect of security of data. |
| [5] | "Sharing Confidential Images with Abbreviated Shares using Steganography and AES Algorithm" | The proposed method boosts the level of security by merging two different steganographic approaches. | It is possible to acquire a greater number of shares, which would result in a much-improved level of safety for any future research. |
| [6] | "A Color Image Encryption Scheme Combining Hyperchaos and Genetic Codes" | The proposed approach works on several image dimensions N × M with diverse image sizes and formats. | To make this cipher more efficient, we can investigate computationally intensive operations in the future. |
| [7] | "Implementation Of Stego-Watermarking Technique by Encryption Image Based On Turbo Code For Copyright Application" | The system produced a high-quality Stego-video with a relative rate of 98% and a PSNR of 57dB. It was robust to salt and pepper and Gaussian noise. | After incorporating the logo bits, the frame pixels values change somewhat but are negligible. |
| [8] | "Analysis of ECG Image File Encryption using ECDH and AES-GCM Algorithm" | With low PSNR and SSIM values, the encrypted image file is excellent. | ECG picture file encryption and decryption can be improved with greater size, 16 GB RAM, and longer keys. |
| [9] | "New Image Encryption Algorithm based on Pixel Confusion-Diffusion using Hash Functions and Chaotic Map" | Experimental study demonstrates that the algorithm resists several communication channel statistical and differential attacks. | To strengthen the encryption algorithm, we'll use machine learning. |
| [10] | "Asymmetric Image Encryption Based on Twin Message Fusion" | Analyses of the method's security and efficacy have shown that it has both. | There is a possibility that the image quality will alter the results. |

**EXISTING SYSTEM**

The existing systems encryption and decryption methodologies are not efficient existing system is not done with noise key as an encryption method. The existing system also doesn’t have an scalable architecture and it cant be used for complex use cases.

**PROPOSED SYSTEM**

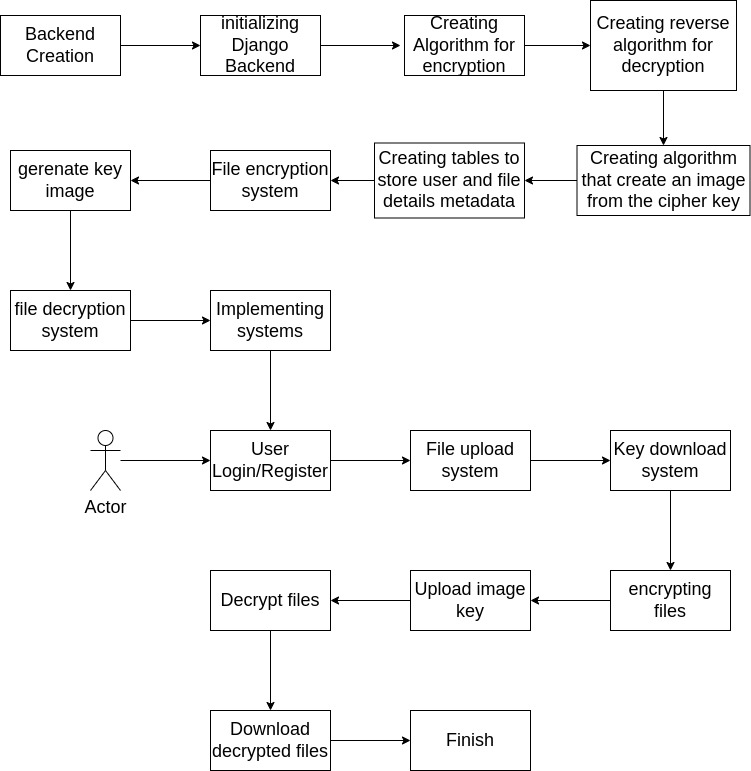
Step 1 : Initializing Backend The first step is to create an django backend to work with the file encryption system.

Step 2: Creating algorithm: We will use AES algorithm for doing the encryption -decryption thing

Step 3: Make a table for the database using: To manage the users and files that are saved on the server, we will develop a user table in addition to a file handling table.

Step 4 :Creating a Two-Factor Authentication System :By utilizing a distinct Django server, we will construct a two-factor auth in order to guarantee the safety of our users. In most cases, the application level is where the encryption of fields at the field level has to take place. This is accomplished by coupling the data with a top-secret key that is familiar only to the application server in question.

**ARCHITECTURE**

****

**Fig 1. Architecture diagram**

**IMPLEMENTATION**

**Module 1: Creating Backend using Django**

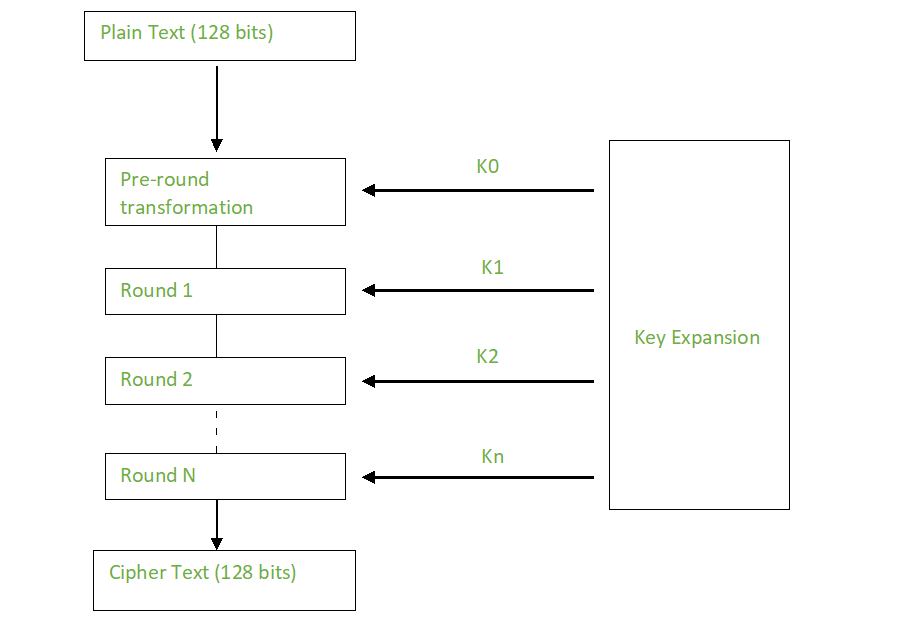
Create a new directory or add a new python package in your app using the name utility once you have finished the initial setup of the project and added the first app. Make a copy of the init .py file and place it within the utility directory. In the utility directory, create a new file and give it the name encryption util.py. Create a new function that encrypts the content that has been provided. ENCRYPT KEY should be kept safe. Do not commit it to git; rather, keep it in the settings production.py file where it belongs. Because it's possible that we'll need to send encoded data to a URL, we're also converting the encoded string to a url-safe base64 format. If there is an error of any kind, make sure to log it and then return null. Just reversing the process will allow us to decode any text that has been encrypted.

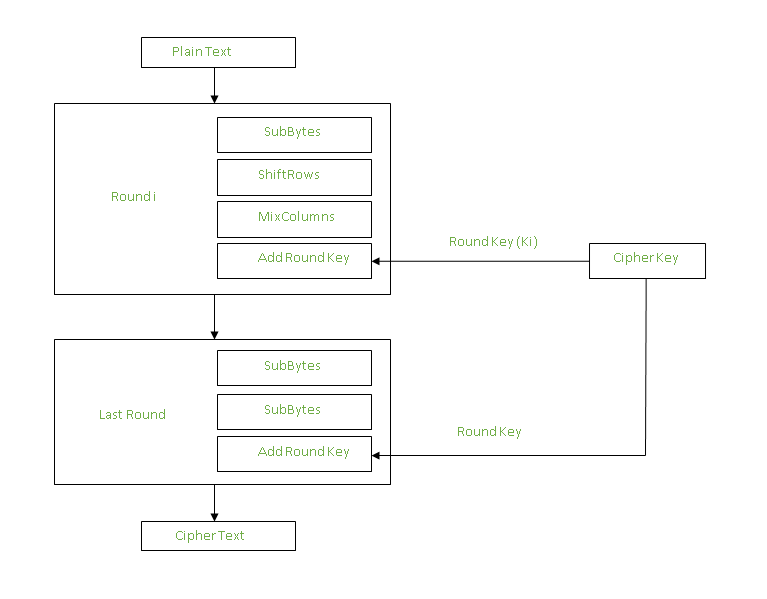
**Module 2: Using AES Algorithm For Encryption-Decryption**

AES is an iterative kind of encryption. This particular algorithm is a symmetrical block cypher. It is capable of encrypting plain text that is 128 bits in length. This technique utilizes keys with lengths of 128, 192, and 256 bits respectively. It is regarded as the algorithm with the highest level of security.

**Putting in work on the AES Algorithm**

1. Get access to the key by using the cypher key.
2. The simple text should be assigned to the state array.
3. Initialize the state array with the round key at the beginning.
4. Carry out the manipulation a total of nine times.
5. The eleventh and last alteration needs to be carried out.
6. Do a copy of the ciphertext.



**Figure 2: The Working of AES Model**

**Module 3: Make a table for database**

A Django model is a built-in component of Django that is used to generate tables, as well as the fields within those tables and the different constraints that apply to those fields. In a nutshell, Django Models is the SQL-based database management system that is used in conjunction with Django. SQL, which stands for "structured query language," is a complicated programming language that requires the execution of a large number of distinct queries when a database needs to be created, deleted, updated, or any other function. The chores can be simplified with Django models, which also organise tables into models. In most cases, one model corresponds to a single table in the database.

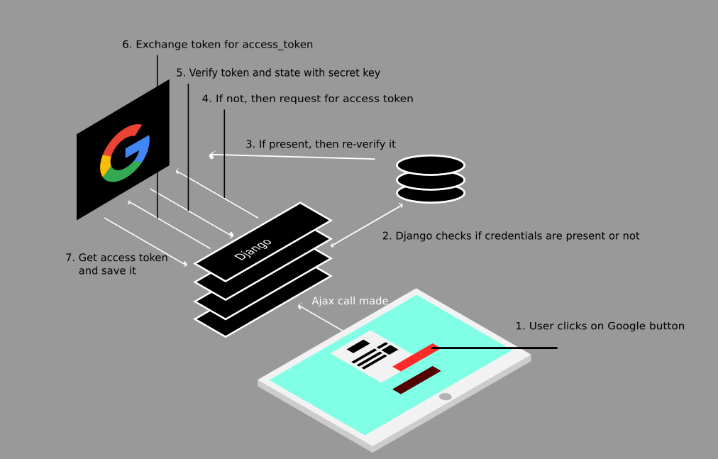
Using Django, we create a table for the database that corresponds to our model: We are going to construct a user table in addition to a file handling table so that we can manage the users who save files on the server and the files themselves.

**Module 4: Creating a two-factor Authentication System**

Then, establish a Django 2FA Project directory in the project's location. Open the newly formed project folder in an IDE or text editor. Let's establish a Python virtual environment for the project's dependencies. Run this command in the root folder terminal to build the virtual environment, depending on your OS. If your IDE or text editor prompts you, click "Yes" to enable the workspace folder virtual environment. Manually activate the virtual environment with one of these OS-specific instructions. Install Django web and Django REST in the virtual environment.

This command creates an otp project Django project in the root directory after installation. Django-admin will generate the project in the current directory if the project name ends with. This command pushes the built-in user model's first migration files to the database. Create otp app, a simple Django app, in the root directory terminal. Create a models.py file in the otp app folder for a web app.

Django requires a username and password for abstract users by default. Nevertheless, by setting username = None, we may remove the username field from the Abstract user. This method lets us log in with our email and password without validation problems. Django must also accept email addresses as usernames. We set the email field to the USERNAME FIELD constant. The REQUIRED FIELDS list tells Django to prompt us for values when creating a user via the createsuperuser management command. Meta class and fields property defined our model. The create() method now hashes the user's password before persisting it to the database to avoid saving plain-text passwords.



**Figure 3: Two Factor Authentication using Django**

Python web applications can utilise PyOTP to enable two-factor (2FA) or multi-factor (MFA) authentication. Two-factor (2FA) authentication varies per language, but the implementation is similar. Open views.py in otp app to start. The first API view class handles /api/auth/register POST requests. Django REST will call this route handler to add a database user. Django will invoke this API View's post () function to process account registration requests. The request body will be checked against the UserSerializer rules, and if any are violated, the client or frontend app will receive a well-formatted validation error.

**COMPARISON STUDY**

|  |  |
| --- | --- |
| **Previous Algorithms** | **Cons as Compared To AES** |
| **DES** | 56-bit key size is DES's main drawback. Chips can encrypt and decrypt a million DES operations per second. |
| **MD5** | Significantly less safe and prone to crashes. It's simple to get the same hash function for two different inputs. |
| **HHEA** | The fact that data encryption only provides a minimal level of protection to data that is already in transit is one of the technology's most significant drawbacks. |
| **SHA1** | It has a lower level of security due to the discovery of a large number of vulnerabilities throughout the years. It is simple and inexpensive to find collisions that it possesses. |

**RESULTS**

Our plan is to apply the encryption methods of the AES after completing some steps in the encryption process in order to produce an outcome of file that will, as a result, be hashed and securely decrypted by the reliable software that is used for file storing. This strategy will ensure that the data is kept secure. The decryption revealed that the software can successfully encrypt binary files, as well as text files and any other type of file. These results demonstrate that the software is capable of providing a high level of file encryption.

**CONCLUSION**

File encryption is achieved with the use of complex algorithms. If the information contained in a file has been scrambled using an encryption method, the file is said to be encrypted. A scrambled file is unreadable, but there are still many cyber-attacks happening in many ways, posing a threat to the confidentiality of the information. In this work, we addressed noise pictures as the key for file encryption and examined numerous case studies. Experimental results reveal that the suggested noise picture as a key for encryption and decryption procedures is both secure and reliable, opening the door for its use in high-security image communication applications.

***REFERENCES:***

1. <https://ieeexplore.ieee.org/document/9502246>

2. <https://ieeexplore.ieee.org/document/9706273>

3. <https://ieeexplore.ieee.org/document/9315245>

4. <https://ieeexplore.ieee.org/document/9701774>

5. <https://ieeexplore.ieee.org/document/9847932>

6. <https://ieeexplore.ieee.org/document/9681070>

7. <https://ieeexplore.ieee.org/document/9253112>

8. <https://ieeexplore.ieee.org/document/9924954>

9. <https://ieeexplore.ieee.org/document/9835957>

10. <https://ieeexplore.ieee.org/document/9337031>

[1] T. M. K. Afandi, D. H. Fandiantoro, Endroyono and I. K. E. Purnama, "Medical Images Compression and Encryption using DCT, Arithmetic Encoding and Chaos-Based Encryption," 2021 International Seminar on Intelligent Technology and Its Applications (ISITIA), 2021

[2] C. Qin, J. Hu, F. Li, Z. Qian and X. Zhang, "JPEG Image Encryption with Adaptive DC Coefficient Prediction and RS Pair Permutation," in IEEE Transactions on Multimedia,2022

[3] M. D and S. Vasuhi, "Image Steganography: 2-Bit XOR Algorithm Used In YCbCr Color Model With Crypto-algorithm," 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2020

[4] J. Tong, Y. Long and Q. Liu, "A File Encryption System Based on Attribute Based Encryption," 2021 17th International Conference on Computational Intelligence and Security (CIS), 2021

[5] A. N, A. V. K and N. R, "Sharing Confidential Images with Abbreviated Shares using Steganography and AES Algorithm," 2022 2nd International Conference on Intelligent Technologies (CONIT), 2022

[6] H. Nazir, I. S. Bajwa, S. Abdullah, R. Kazmi and M. Samiullah, "A Color Image Encryption Scheme Combining Hyperchaos and Genetic Codes," in IEEE Access, 2022

[7] O. Q. J. Al-Thahab and A. A. Hussein, "Implementation of Stego-Watermarking Technique by Encryption Image Based on Turbo Code for Copyright Application," 2020 1st. Information Technology To Enhance e-learning and Other Application (IT-ELA), 2020

[8] P. Oktivasari, M. Agustin, R. E. M. Akbar, A. Kurniawan, A. R. Zain and F. A. Murad, "Analysis of ECG Image File Encryption using ECDH and AES-GCM Algorithm," 2022 7th International Workshop on Big Data and Information Security (IWBIS), 2022

[9]  S. Patel and T. V, "New Image Encryption Algorithm based on Pixel Confusion-Diffusion using Hash Functions and Chaotic Map," 2022 7th International Conference on Communication and Electronics Systems (ICCES), 2022

[10] M. E. Kahla, M. Beggas, A. Laouid, M. Kara and M. AlShaikh, "Asymmetric Image Encryption Based on Twin Message Fusion," 2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP), 2021