Aes enc and dec using openssl flow

+----------------------+

| AES Algorithm |

+----------------------+

|

v

+----------------------+

| Initialize Context |

+----------------------+

|

v

+----------------------+

| Set Key and IV |

+----------------------+

|

v

+----------------------+

| Encrypt or Decrypt |

+----------------------+

|

v

+----------------------+

| Finalize Context |

+----------------------+

The flowchart shows that the AES encryption and decryption process begins by initializing the encryption or decryption context using the EVP\_CIPHER\_CTX\_init() function. The context stores the necessary parameters, such as the key and initialization vector (IV), for the encryption or decryption operation.

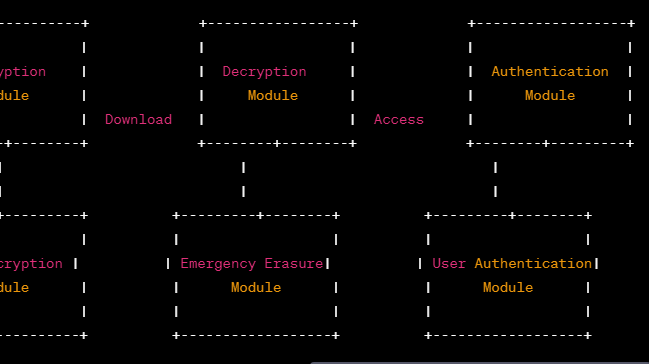
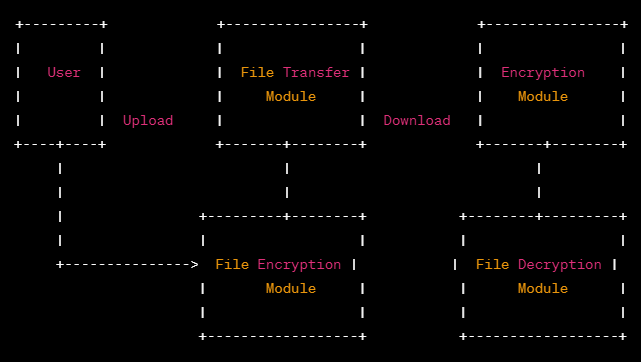
After initializing the context, the key and IV are set using the EVP\_EncryptInit\_ex() or EVP\_DecryptInit\_ex() function. The key and IV values are obtained from a secure key storage location or provided by the user.

Once the key and IV are set, the encryption or decryption process can begin using the EVP\_EncryptUpdate() or EVP\_DecryptUpdate() function. This function reads the plaintext or ciphertext data in blocks and encrypts or decrypts it using the specified AES algorithm.

After processing all the input data, the final step is to call the EVP\_EncryptFinal\_ex() or EVP\_DecryptFinal\_ex() function to finalize the encryption or decryption process and retrieve any remaining ciphertext or plaintext data.

Finally, the context is cleaned up using the EVP\_CIPHER\_CTX\_cleanup() function to ensure that any sensitive data is securely wiped from memory.

Note that this is a high-level flowchart and does not include detailed error handling or memory management steps that may be required in a production implementation.



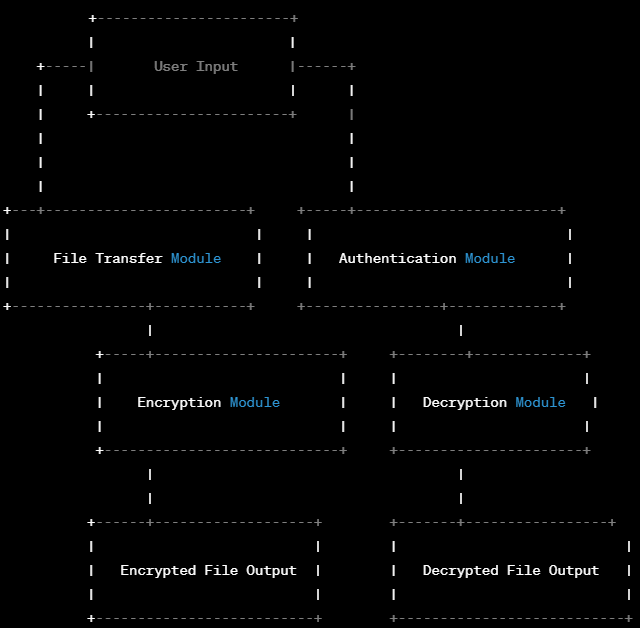
The DFD illustrates the flow of data between the different modules in the system. The User module interacts with the File Transfer module to upload or download files.

When a file is uploaded, it is passed through the File Encryption module to encrypt the data before being stored or transferred. When a file is downloaded, it is passed through the File Decryption module to decrypt the data.

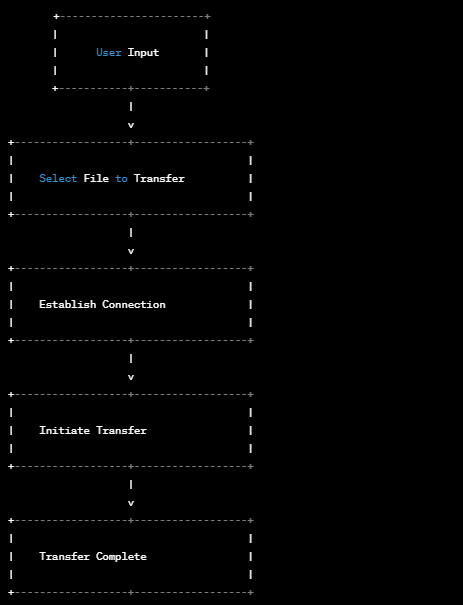
In the case of an emergency, the Emergency Erasure module can be used to erase all data stored in the system.

The Authentication module is responsible for ensuring that only authorized users can access the system. When a user attempts to access the system, the Authentication module verifies their credentials and grants access if they are authorized.

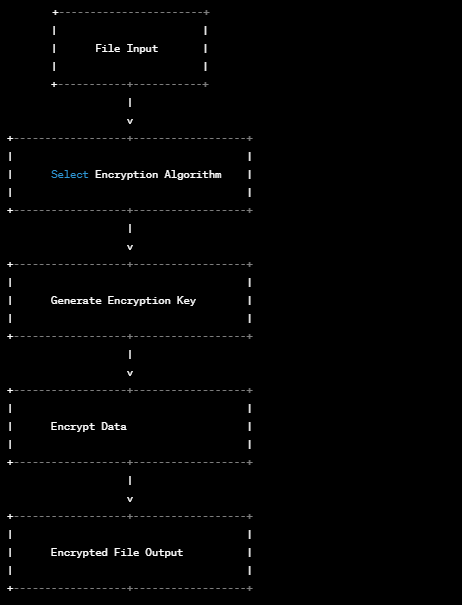
The DFD shows the flow of data between the different modules, but it does not show the specific data processing steps within each module.



**File Transfer Module**



## Encryption Module DFD



**Decryption Module DFD**



**Emergency Erasure Module DFD**



**Authentication Module DFD**

