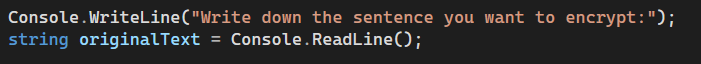
AES cryptographic Algorithm

## AES Algorithm



The user is asked to enter a phrase that they wish to encrypt, and the text supplied is stored in the originalText variable.



In this code segment, an indefinite loop is created to repeatedly prompt the user to choose an encryption key. The user is specifically prompted to enter a number, either 1, 2, 3 or 4. The user's input is carefully validated, checking both its validity as a character and its position within the specified range. If the input is valid, a switching instruction precisely assigns a preset key to the "key" variable according to the user's choice. The loop is immediately interrupted using the "break" instruction. In the event of an invalid entry, an explicit error message is displayed, and the loop continues iteratively until the user provides a valid key. The selected key is then stored precisely in the "key" variable for specific use in subsequent stages of the encryption processes.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

This code snippet implements the Encrypt function, using the Advanced Encryption Standard (AES) algorithm for symmetrical encryption. Here is a detailed explanation of how this code works:

Creating an instance of the Aes class: The code begins by instantiating the Aes class, which represents the AES algorithm. This instance will be used to configure the encryption parameters.

Definition of the encryption key and initialization vector (IV): The encryption key is established by converting the input key (a string) into an array of bytes using UTF-8 encoding. A 16-byte initialization vector (IV) filled with zeros is also created. The IV is used to introduce randomness into the encryption process.

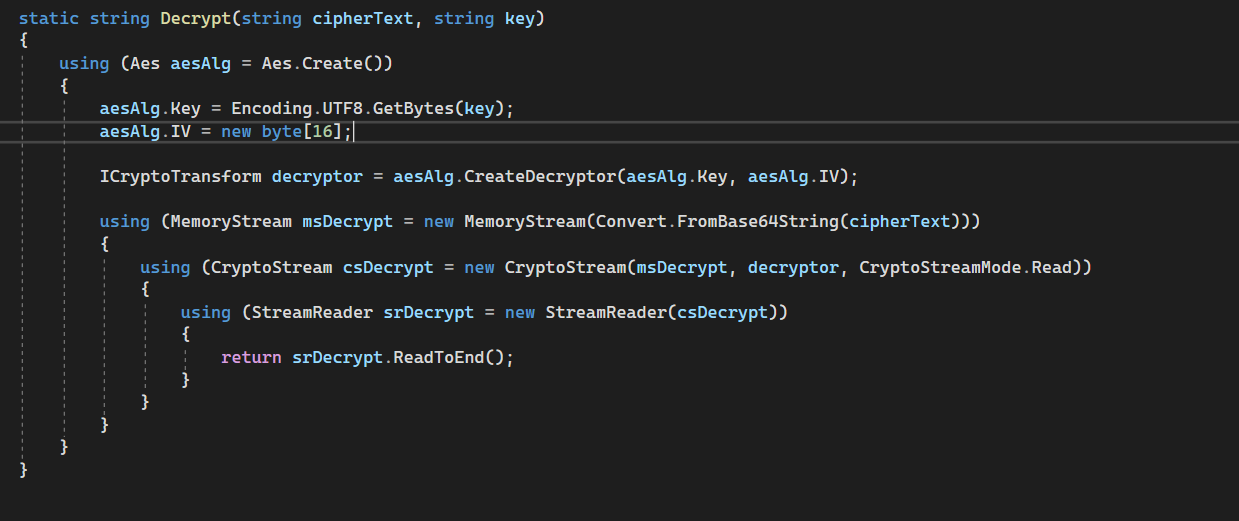
Creation of the ICryptoTransform object: Using the instance of the Aes class, the code creates an ICryptoTransform object via the CreateEncryptor method. This object is responsible for executing the encryption process.

Input text encryption: The input text is written to a CryptoStream, linked to a MemoryStream. This combination enables the encrypted data to be stored in the memory stream.

Convert encrypted data to Base64: The encrypted data, now stored in the MemoryStream, is converted to a Base64 string representation using Convert.ToBase64String. This step is useful for representing binary data in text form, which is useful when there are non-printable characters in the encrypted data.

Result of Encrypt function: The result of the Encrypt function is the ciphertext, represented as a Base64 encoded string. It should be emphasised that this code only deals with the encryption aspect, and a corresponding Decrypt function would be required to recover the original text.

Encryption security: The security of the encryption process depends on several factors, such as the robustness of the key used, the appropriate handling of the initialization vector, and the overall implementation of the cryptographic algorithm. Careful attention to these aspects is crucial to guaranteeing the security of the encryption system.



This code extract presents a Decrypt function dedicated to decrypting a text encrypted using the Advanced Encryption Standard (AES) algorithm and a specific key. The steps involved in this function are described in detail:

Create an instance of the Aes class: The function starts by instantiating the Aes class to represent the AES algorithm.

Definition of the encryption key and initialization vector (IV): The encryption key is defined by converting the input key (a string) into an array of bytes using the UTF-8 encoding. The initialization vector (IV) is also initialized to a 16-byte array filled with zeros. It is crucial to use the same IV during decryption as was used during encryption.

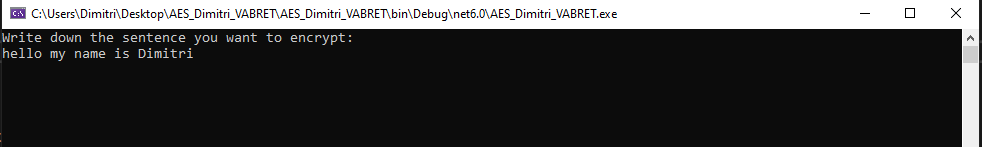
Creation of the ICryptoTransform object: The function then creates an ICryptoTransform object by calling the CreateDecryptor method of the Aes instance. This object is responsible for the actual decryption process.

Text decryption: The decryption process begins by creating a MemoryStream initialised with the ciphertext decoded in Base64. Next, a CryptoStream is created, linked to the memory stream, and configured for reading.

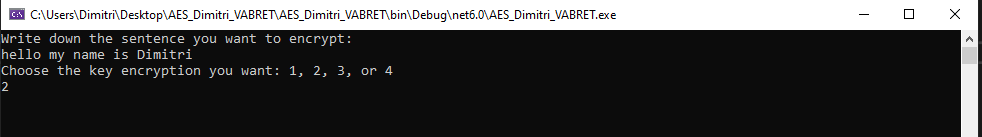
Reading decrypted data: Finally, a StreamReader is used to read the decrypted data from the cryptographic stream, and the decrypted text is returned as the result.

This function provides a detailed and accurate sequence of the entire decryption process, ensuring that the original text is recovered correctly.

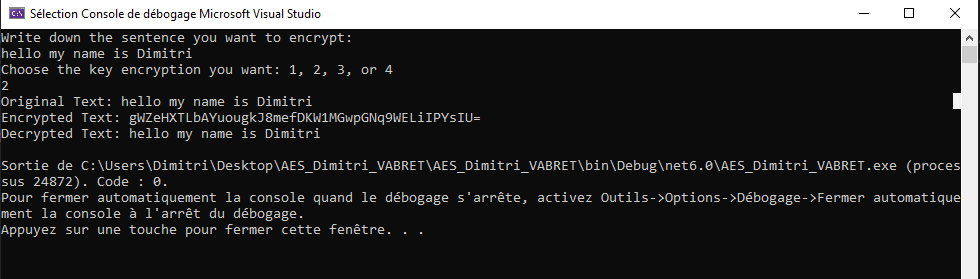
## Console App Results



The first step when running the program is to write the sentence or words you want to encrypted.

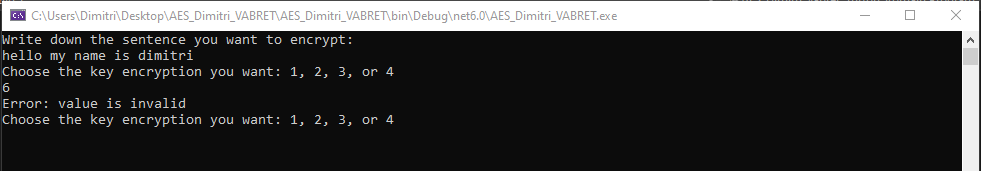


Then, you have the choice between 4 different encryption key which are : “mysecretkey12345”, “abcdefgh12345678”, “abc123XYZ456789!” and “🌟🎉🔐🌺”. Each one of these are 136 bits key. The user choose one of them.



As soon as a valid value is entered, the sentence will be encrypted. The encrypted sentence is displayed, then the encrypted message will be decrypted and the result of this process will also be displayed to be sure that the original sentence is matching with the new one.

To close the Console App and stop debugging, simply press any key on the keyboard.



If the user enters a wrong value (something else than 1, 2, 3 or 4) a message will appear and ask the user to input another value until it is a correct one.

## Conclusion

In conclusion, the code presented offers a detailed view of the implementation of the encryption and decryption processes using the Advanced Encryption Standard (AES) algorithm. The user is asked to provide a phrase to be encrypted, and the choice of an encryption key is made securely. The encryption process uses AES, implementing security measures such as the generation of an initialization vector (IV). Similarly, the decryption function ensures accurate recovery of the original text using the same key and IV.

The approach to the code is methodical, emphasising the importance of security factors such as key strength, correct handling of the IV, and careful implementation of the cryptographic algorithm. In summary, this implementation demonstrates rigorous security management while providing a fluid and intuitive user experience when interacting with the encryption system.