150630D

Tharsanan.K

This application can read a file and encrypt or save a user entered text to database in an encrypted format. For the encryption this application follows some fundamental steps.

1. Create a key
    - This application by default will create 8 letter long key all letters will be capital.
    - First 6 letters of the key will be created randomly.
    - Last 2 letters of the key will be derived from the first 6 randomly created letters.
    - Reason for the deriving 2 letter is to validate the keys when user input keys and prevent the user to see how encrypted text is decrypted with key input. It stops hackers to find key intuitively or using brute force.
    - Another advantage of this mapping functionality (deriving last 2 letters from first 6) is we don't want to store the key to check key is correct or not. And we can display a fake text if a wrong key entered. So, hackers cannot identify that they are getting a right decryption or wrong decryption.
2. Substitute letters in the data.
    - To substitute letters in the data application will generate a key first (as above). By default, key size is 8 but application can support any size of key.
    - This key will be changed to ascii integer value and mapped to $size^2$ number of integers, by a combination of addition between integers derived from key let call this **createSubstituteNumbers()**.
    - These derived integers will be added to data letters-related ascii integers.
    - Ex: data: "Tharsanan is computer science student". Key: HKGJHVB then we will have 64 integers by the **createSubstituteNumbers()** method. Mapped key will be like (68,89,76,98,78, ……..76,89). These 64 digits will be added to the data ascii values in a circular way.
    - After the addition application will have data that is substituted with other ascii characters with respect to the given key.
3. Permute the data
    - Given data will be split into parts of length = size of key.
    - Let say key size == 8.
    - Application will create a rearrangement array, size of 8 by using key. Ex: [3,1,5,2,3,8,4,7].
    - Let see how application create rearrangement array.
      Application will split key into character array.
      Application sort the created array.
      After the sort application will create a array contain 0 to 7 integer array based on the sorted array and real key.
      Ex: key: "NDFNKRFV"
      Sorted array: {D,F,F,K,N,N,R,V}
      Rearrangement array: {4,0,1,5,3,6,2,7}
      Duplicated letters of the keys are well handled.
    - After this split data will be rearranged according to the rearrangement array.

- Key feature of this function hackers cannot get the idea about the permutation. Permutation method is not hardcoded so hackers cannot break the permutation even they have access to the code.

4. Decrypt the data
    - User can decrypt the data by giving key and file path or key and username (from the database).
    - If application find out key is not valid, by using mapping function, it can display a fake information. But for now it only display “not a valid key”.