CSE439 COMPUTER SECURITY

A BASIC HOST BASED IDS

TERM PROJECT

HÜRKAN UĞUR 20160702051

DATA STRUCTURES

```
26 public class HostBasedIDS
27 {
28     private static ArrayList<String> originalFileLocations = new ArrayList<String>(); //[Directory\hurkanugur.txt]
29     private static HashMap<String,byte[]> originalFileContents = new HashMap<String,byte[]>(); //[KEY: Directory\hu
30
31     private static HashMap<String,String> backupFileLocations = new HashMap<String,String>(); //[KEY: Directory\hu
32     private static HashMap<String,byte[]> backupFileContents = new HashMap<String,byte[]>(); //[KEY: Directory\$Di
33
34     private static String originalDirectoryLocation, backupDirectoryLocation;
```

String originalDirectoryLocation [HUKO]

Contains the path of the directory that contains original files.

String backupDirectoryLocation [HUKO\$HUKO]

Contains the path of the backup directory that contains encrypted files.

ArrayList<String> originalFileLocations [HUKO\cse.mp3]

Contains the location of the original files.

HashMap<String, byte[ ]> originalFileContents [K= HUKO\cse.mp3][V= BINARY DATA]

Contains the data of original files.

ArrayList<String, String> backupFileLocations [K= HUKO\cse.mp3][V= HUKO\$HUKO\$cse.mp3]

Contains the location of the encrypted backup files.

HashMap<String, byte[ ]> backupFileContents [K = $HUKO\$cse.mp3][V = BINARY DATA]

Contains the data of the encrypted backup files.

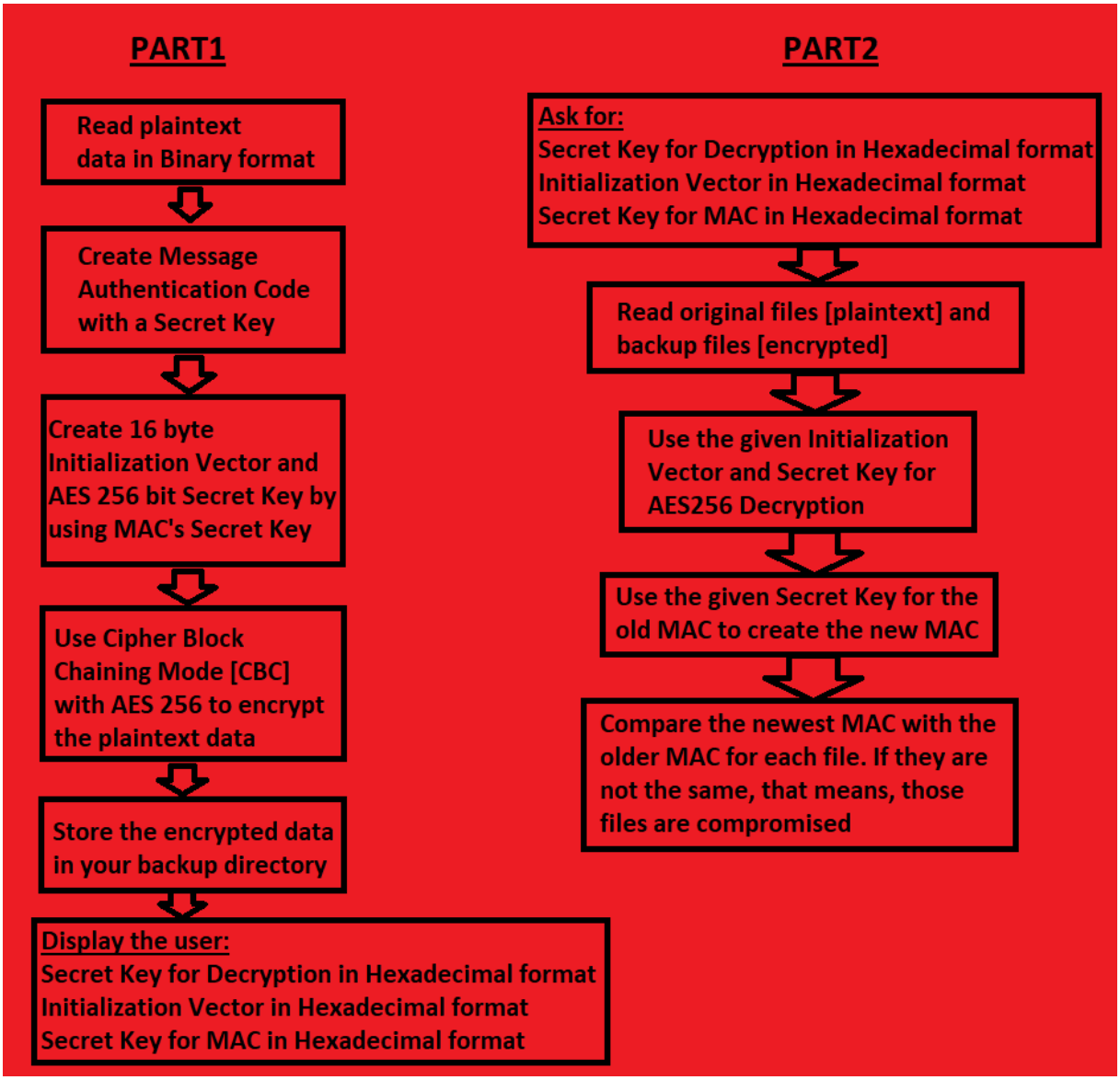
CLASSES AND METHODS

```
public class HostBasedIDS
{
    //AES256 ENCRYPTION & DECRYPTION CLASS
    public static class HukoAES256

    //[FOR PART 1 & PART 2]: READ ORIGINAL DOCUMENTS AND STORE THEIR INFO IN ARRAYLISTS & HASHMAPS
    public static boolean ReadOriginals()

    //[FOR PART 1]: STORE DOCUMENTS THAT ARE CONVERTED FROM ORIGINAL TO CIPHERTEXT
    public static boolean StoreBackup()

    //[FOR PART 2]: READ BACKUP FILES TO CHECK THEIR MAC CODES IN THE FUTURE
    public static boolean ReadBackup()

    //CONVERT BYTE[] -> HEX VALUE STRING
    public static String ConvertByteArrayToHexadecimalString(byte[] byteArray)

    //CONVERT HEX VALUE STRING -> BYTE[]
    public static byte[] ConvertHexadecimalStringToByteArray(String hexadecimalString)

    //[FOR PART 1]: CREATE SECRET KEY FOR MAC OPERATION
    public static SecretKey SecretKeyGenerator()

    //[FOR PART 1]: CREATE MAC AND STORE THEM BY ADDING THEM AT THE END OF THE FILE ALONG WITH SECRETKEY
    public static boolean CreateMessageAuthenticationCode()

    //[FOR PART 2]: COMPARE NEW MAC CODE AND OLD MAC CODE (NEW MAC CODE IS CREATED BY OLD SECRET KEY OF EACH FILE)
    public static void ControlMessageAuthenticationCode()
    public static void main(String[] args)
}
```

MY OWN AES-256 BIT ENCRYPTION CLASS

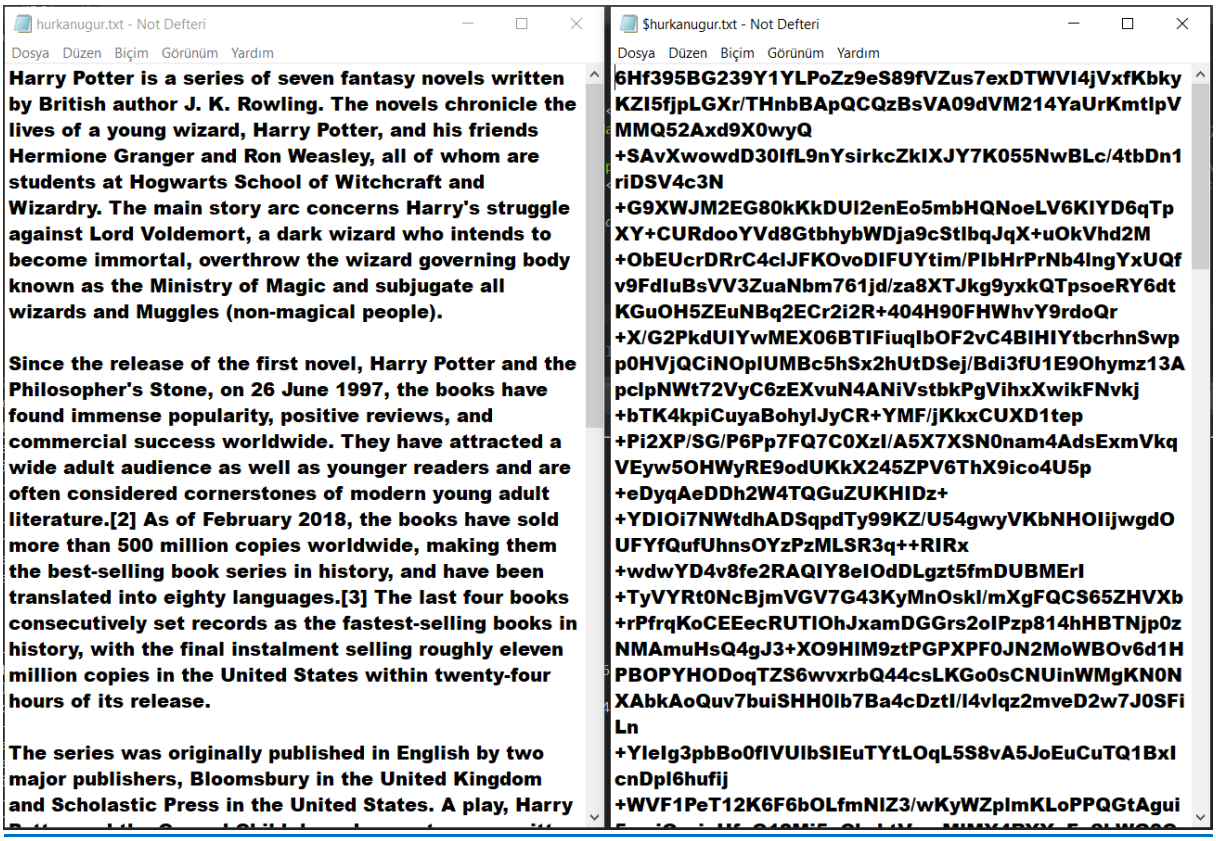
```
//AES256 ENCRYPTION & DECRYPTION CLASS
public static class HukoAES256
{
    private static SecretKey secretKey = null;
    private static IvParameterSpec initializationVector =  null;
    private static String hexadecimalStringSecretKey = null;
    private static String hexadecimalStringInitializationVector = null;

    //ENCAPSULATIONS
    //GET SECRET KEY
    public static SecretKey GetSecretKeyOfAES256()
    //SET SECRET KEY
    public static void SetSecretKeyOfAES256(SecretKey secretKey)
    //GET HEXADECIMAL STRING SECRET KEY
    public static String GetHexSecretKeyAES256()
    //SET HEXADECIMAL STRING SECRET KEY THEN GENERATE AES256 SECRET KEY
    public static void SetHexSecretKeyAES256(String secretKey) throws Exception

    //INITIALIZATION VECTOR GENERATOR
    private static IvParameterSpec InitializationVectorGenerator() throws Exception

    //GET HEXADECIMAL STRING INITIALIZATION VECTOR
    public static String GetHexInitializationVector()
    //SET HEXADECIMAL STRING INITIALIZATION VECTOR THEN GENERATE AES256 - CBC IV
    private static void SetHexInitializationVector(String initializationVector) throws Exception

    //AES256 SECRET KEY GENERATOR (GENERATES AES-256 BIT KEY FROM THE MAC CODE)
    public static void GenerateAES256SecretKeyFromMACSecretKey(String MAC_SECRET_KEY)

    //AES-256 ENCRYPTION FUNCTION
    public static byte[] EncryptionWithAES256(byte[] Data)

    //AES-256 DECRYPTION FUNCTION
    public static byte[] DecryptionWithAES256(String Data) throws Exception
}
```

A BRIEF EXPLANATION ABOUT HOW MY PROGRAM WORKS



PART1: CREATING BACKUP OF THE ALL FILES IN THE DIRECTORY

```
HostBasedIDS [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (8 May 2021 00:34:52)
---------------------------------------------
[0]: Create backup of a Directory
[1]: Check if there is an Intrusion
[2]: Exit
---------------------------------------------
[Your Choice]: 0
---------------------------------------------
Enter the Directory name to be scanned: HUKO
[Original File Reading]: Files are being read...
[Original File Reading]: 6 file(s) are read successfully from [HUKO]
[Backup File Writing]: Create Directory -> HUKO\$HUKO
[Backup File Writing]: Files are being stored...
[Backup File Writing]: 6 file(s) are stored successfully in [HUKO\$HUKO]
---------------------------------------------
[Decryption Secret Key]: 4D69646F6E46665A6B4377514D337566644A4D764D733443647439534171 2B2B474254362B2B6679792F773D
[Initialization Vector]: 76CEBDA71EC8E718AE18FBF4DC9BFFFC
[MAC Secret Key]: 752F79594A5564717A6A6D3555596C64556C714C586B54676A4771335066475633584962434558565 84A733D
```



PART2: IF THERE IS NO INTRUSION

```
HostBasedIDS [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (8 May 2021 00:34:52)
[0]: Create backup of a Directory
[1]: Check if there is an Intrusion
[2]: Exit
-----------------------------------------------
[Your Choice]: 1
-----------------------------------------------
Enter the Directory name to be scanned: HUKO
Enter Decryption Secret Key: 4D69646F6E46665A6B4377514D337566644A4D764D7334436474395341712B2B474254362B2B6679792F773D
Enter Initialization Vector: 76CEBDA71EC8E718AE18FBF4DC9BFFFC
Enter MAC Secret Key: 752F79594A5564717A6A6D3555596C64556C714C586B54676A4771335066475633584962434558565 84A733D
[Original File Reading]: Files are being read...
[Original File Reading]: 6 file(s) are read successfully from [HUKO]
[Backup File Reading]: 6 file(s) are read successfully from [HUKO\$HUKO]
-----------------------------------------------
[Scanning Result]: Everything is OK !
-----------------------------------------------
```

PART2: IF THERE IS AN INTRUSION

```
HostBasedIDS [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (8 May 2021 00:34:52)
-----------------------------------------------
[0]: Create backup of a Directory
[1]: Check if there is an Intrusion
[2]: Exit
-----------------------------------------------
[Your Choice]: 1
-----------------------------------------------
Enter the Directory name to be scanned: HUKO
Enter Decryption Secret Key: 4D7131523349373453 4B43535278634A6B455A485A5A4F48577037666C425661596D5245534A7776434E303D
Enter Initialization Vector: F795D8A7995D64DED38727BF9D343D54
Enter MAC Secret Key: 716A395469514F66516E3043514163656242633 64D4F69492B6B69796E3974587174364344 6B464E4670513D
[Original File Reading]: Files are being read...
[Original File Reading]: 6 file(s) are read successfully from [HUKO]
[Backup File Reading]: 6 file(s) are read successfully from [HUKO\$HUKO]
-----------------------------------------------
[AES256 Decryption]: Something went wrong !
[DECTRYPTION ALERT]: This file is compromised -> HUKO\hurkanugur.txt
-----------------------------------------------
```