



National University
of computer and emerging sciences

ASSIGNMENT 1

INFORMATION SECURITY

Muhammad Ayaz Hasan [20k-1044]

SE-7A

Submitted to:
Dr. Abdul Aziz

FAST-NUCES Karachi

DES ENCRYPTION/DECRYPTION(TXT FILE)

We have a Text file (sample.txt) with dummy data of upto size 10 mb.

```
sample.txt x
sample.txt
1  AYAZ  HASAN  20K-1044  SE-7A  INFORMATION  SECURITY  COURSE
2  TJxrYu3tapRVauGZT3VAPzLU07IbK80pFaUFG636LpCV7xpMTthocmkSwnaAkzhU73LG  709pw
18tZfE5qYKViqvXhGvFJZl9xhYb0rQgHekrUFYvFY6vznZFO6QccLratotCLdpwJEVZMG6YJdDmsVibkgZqvBQ5lWfMd
81jd8T73Wn9Z7zcxIXe0Sxg7yXJiUWhocGkpzs0rGjBCRgwYIZj7jITDUkwP4fpZ5IOL4s6x0LqoCo0hUub8CI82Wnb8M9yjdGnrB238d2IaUa7CSNNqtPdM7HEKvns14DLU4nV9NcG1acfiPhf
CTVp 0B7qH5SY571IhZHiME00tEZOVcwZjBNZUeGRSHGv4kgBwVepZUxVYnaV Xlp3P37p12rq MGKGVI8pu7j31KalD5ddIiN
ey4JagE1PE1k92Mg7sd0Ki2dhma06NqJJXrSVgghsumr7MQTdznsP1LPHAAOYKDRGMcpfE6CUoQ73VePuwyPhJjIHxehspNcCocboiSQ
ZPEfc0DfXg879tlylp2njCJ1E1PTxPnxSSsai fxmIeowIpfOrhXONf0K9GWIaQztIwgeA3huatIomgvdr0XBhwcAdwK4B09xJiXe8rXRqsRAw6ThQbCKTKoFwaGLPD3zy19BQf2jp1RWrr
sOP4L088BTdQ51sALPblayjcvPgkOnB8B8Vfmieolw6bIomCLL5lwmffypilUzYkEQa61KKt
xGcYr6YRIPuAlzxNT0HidwGKzSlyzmIn5xMpF17iYMBwZDsc0SHXGUMesfHgiVAc7pUcfcnejwxRmpo10071rsH63NyOPCukBviUqrivtdIW6FpjtmMiYLIxJAcf3btKiYZMC9WFnu2ez06
Nq73807zeR0 FFXqSPXuA02jpirNIC0c ohxuXVVIKIXmqef1B5koQVgyz34EgkwbQm0mm3IYyeGAzmtGcJMU9SWkyIkMty3 TTRLc2A6CESFXOR1niC4UBBLPt63PF1IVAx2i
fDj28jw00BokduXwsTuofUeiUrekSq4Mg9RVBFQcBQxelVcVM2Tp1dfthMm1wMiEfVENUumfMu3W7LneZALLQ13jLu9rV7P38dF9KyiiPpPwpDYADXzua0H6QWVF6TiVuWrfAbZuhse7ex2
S4yz0jxjnuA089HhgD4k5kckQZT38Bss0i8fh1NBbi6Tu9KfyG3d4GkXuk2IQqj2CQf0rOYoQjVQAdvugbBRFmev7yDpdntLDY8SBG4a4ED00vyeskiRLEq7L4TEiY3mJxMoFNQ3IEMSM0ZwUwY
7hac8xjPU5zPQVFRTHbB9XUWZ8rXar nfs8H50v26GMUL5NB 4r
9jp744AggdSI1l0c8h6kDaCEKN4J9mh31XrIUyDcTFq1RjEL2MmfVMF20e33a0ysuIgpU3HK5IdLy1Zl6jo3domaMmUwG L FRrjunQmj1bqchX
CHC6G6DGAV1QvZ545C4YKZgga4W0fG1RwP CsGa 0Rb2UoToFyrCoeUz0BwXsilfhPp0sRVqB7linB806bKZysXSEnd8T6Zj1l FxCR8sBLq7RPMYURBkrOHT9 VDoGsTc
t107UGPslnjDM7Docnhs41lYstWx81wzWu80sr517cR0xkiTo1grzDREDsN3FdhIhN8K5lti3MCK11T2QeWnUfU46E3b58BS4Sv9JUpHIsuB2SC1DFwxJWuFUVkvzx8a1BNCwFOF
hXkM7T1ixp4W09HhgD4k5kckQZT38Bss0i8fh1NBbi6Tu9KfyG3d4GkXuk2IQqj2CQf0rOYoQjVQAdvugbBRFmev7yDpdntLDY8SBG4a4ED00vyeskiRLEq7L4TEiY3mJxMoFNQ3IEMSM0ZwUwY
0jceTsCR8Hay z9ca4appM8rathyycc23eq2 mlfcmzcx4G32ndWg6uejelTMWZyQ5Q00iXfuy02Mj
mgL1nQWvrqR6Yk1HrG1UqeRAYPzjUoRgP8h440cWm6ATkHLBAPndsQsHepeSuz20k715GTPGTFzDsey8JYV1bdtAxvkZagahDPnRLpnrfIcAwIJKVHGpBRCzZGWEUBU84YXSEzChmntNodvay
8BherJ 9B8dhtwt9tUpRVL0QL4GwH96kQZaEqETiggrGYWsdj TkYf7cEvV4r1l3WNBKMF3WmH5cIGTFPNQYe LhxS64ZQtuxiWcXo
6yQYXUHVafZ8vH1HlGaLxYyV1JGvHmC5z6EcpH5lGrGBRN1r0QaB9i FiddZCYp6iQR4Q1Pdof0NQ605WU0MuI34taxa1l09r1Exego2k7LFx8sdcM2Gea1XmpVnfZiHRvrVHLNCHXaO
eF191G0ukVUGoQgeqTmW09ihhpA91l0MiNHydfhfFAAEFCiNazZXYr9VrnpV085yEtwH0BZst6WfNYTghubJ4eA38G0UUD2J1fFouPbGgc7CT2wM4ptHcFwmNBEOOnSHXDnp7wZex
1t4gZ1IjH5Q0cKz8QY0J7TU bJodMH2L8Foa9Nr5BWNnzBqxfGxJphu3IXAoM3nXUE5XerDnH3H3ELpCifr5RouERNtn1yIiKA68Er7W1ziU1HwzLU7LgkMPZ20H0b16xkvIkNBeyyEO
PF0n9163IK3xS5z1g87jKMuXkz2dGKAvpWrEFnsf6liZ9lejoHbxsG9tdiLnwOib9g dGB0ccYv28I7BfhRELwFISuZ48 qNoG0z6
9dCieGT7Zn6vmCEoT24xIAV4a7SeG59ptHG9Xw0Ry3k69RxB7t0fU9ZJBIn9HPMF Ac2aWQmOZGJ9nCuhks17G8gzI8fEcYP66Hm3Z9DRX w2ZxhsUt2W41licGMKchjQC0xE9Is0qHfTg24m
mQlYbGw4N1K3miHCUkrINM4DMrhmhJG ZhB0pIt7iVpVlK2h958madNkbgotf2fnpH4CTB8HuGv bFRmVbhIM0motiWCEDrnl2U4M3vtveAuio6BVqrdoKcjbWdDyK
DvB67NZu7Jw60QxYrXh3CYrYf9K36G8Cz7HMXGmos23EhokdytLHMKJ86ciSLarq5 7nAPKozmxfsHTMJB8a3Ak1KaxdQStXonly x92WOXI8 cfiH
UvxE1y57RTVwxTfQddVC8RGxEdxmckmlZBCKgQzsvSn3w4bp4qFwmcFOGmbutsZAQfybhj0ex687951dst9Je0mbstJL D4xbzc57s
```

First we generate the secret key:

```
J KeyGeneration.java x
J KeyGeneration.java > KeyGeneration
1  import javax.crypto.KeyGenerator;
2  import javax.crypto.SecretKey;
3  import java.security.NoSuchAlgorithmException;
4
5  public class KeyGeneration {
6      public static void main(String[] args) throws NoSuchAlgorithmException {
7          KeyGenerator keyGenerator = KeyGenerator.getInstance("DES");
8          SecretKey secretKey = keyGenerator.generateKey();
9
10         byte[] keyBytes = secretKey.getEncoded();
11
12         StringBuilder keyHex = new StringBuilder();
13         for (byte b : keyBytes) {
14             keyHex.append(String.format(format:"%02x", b));
15         }
16         System.out.println("Generated Key: " + keyHex.toString());
17     }
18 }
19
```

```

gram Files\Java\jdk-19\bin\java.exe" "-XX:+ShowCodeDetailsInExceptionMessage
ca0830329e7734870d4b9fb7018fc4\redhat.java\jdt_ws\ASSIGNMENT_1_ea12118d\bin
Generated Key: f4196e85cbbcd99d

```

The above generated key will be use for encryption and decryption.after run the file we will have two file generated encrypt.txt (having encrypted data),decrypt.txt (having decrypt/plain text)

```

J DesEncryption.java
J DesEncryption.java > DesEncryption > main(String[])
1  import javax.crypto.Cipher;
2  import javax.crypto.SecretKey;
3  import javax.crypto.SecretKeyFactory;
4  import javax.crypto.spec.DESKeySpec;
5  import java.io.*;
6  import java.security.spec.KeySpec;
7
8  public class DesEncryption {
9
10     Run | Debug
11     public static void main(String[] args) throws Exception {
12         String inputFile = "sample.txt";
13         String encryptedFile = "encrypt.txt";
14         String decryptedFile = "decrypt.txt";
15         String secretKey = "f4196e85cbbcd99d";

```

```

encrypt.txt
encrypt.txt
This document contains many invisible unicode characters  Disable Invisible Highlight
1  GS B5 'EM d" k q % ? SUB c h W SYN ] NUL FS
2  0 d DCI ] i S FF Lp \ GS
3  0 29z S* u ' $ e ] { . US , STX
4  H52 P e > ~ x 4 g @ DEL d RS { z % 81 _ ENQ $ SUBSTXT (# VVnE2
5  DUE P1 7 ETB f e D GS ENQ VtP A " 9 | FS o
6  % SI 1 ! U B B B B 9 r o ? STXSTX & - k , ETB M A C o o U A SP ) < 3 h RS ESC B Ni | G
7  S SI DC2 4 CAN I > R 8 L EM / S GS IX
8  3 NAK I FF SV V ENQ GS ESC * SI X ^ NUL NAK DC3 DCI B RS . u B ) U O P Z NAK W n c
9  , P O ETB T n NUL _ ESC NAK L ! ) e BS k \ d 2 t ; ( SI [ z ;
10  G ( so % k _ ~ # so O Y SOH u O O C \ n 5 f f _ > % @ + F SO _ $ Uc _ ~ EOT u H o y _ V _ o k W ; SI S
11  EM SYN GS k p c BEL _ 9 Y L BS M14 ACK 43 r s SOH G DCI T o ! EOT q FF t g : y & N O H S SUB
12  K US C SUB N ; DCI > > o r S 3 l p DC4 ETB GS ; . & < NUL B 0 K M J BS g PS 5 [ i w 8 Z EN \ G _ $ "
13  d H Y z RS a 7 f a ESC p ? = X L % $ E T a FF C > US y RS C W V T ETB d SEOT US
14  l J J ETX V / c a & { 0 DC3 b [ L B Q O Y | @ B ' VT K V T DC4 a FS 1 DC3 C T S Z DC4
15  b h 0 J V
16  j ! 9 \ FF ESC D SI v " ; 祠 i " 5 & & 0 RS k [ BS ENQ SUB ^ W ; V ~ L ACK STX X "
17  i 7 NUL 47 n SYN ^ ; u * EOT [ j " O R GS CY q _ BS a f H S f ( 5 C = EH j % K US d DC4 @ Q S l # [
18  ~

```

```
decrypt.txt x
decrypt.txt
1  AYZ HASAN 20K-1044 SE-7A INFORMATION SECURITY COURSE
2  TJxrYu3tapRVauGZT3VAPzLU07Ibkr80pFaUFgE36LpCV7xpMTthocmkSwnaAkzhU73LG 709pw
18tZFe5qYKViqvxVhGvfJZL9xhYb0rQqHewrUFYvFY6vnzF060QccLratotCLdpwJEvZWGGYJdDmsVIbkgZqvBQS
81jd8T73Wn9ZYzcxIxEOsXg7yxJiUwhocGkpzsQrGJBCRgwYIZj7jITDUkwP4fpZ5IOL4s6x0LqoCoOhUbBCI82W
CTVp 0B7qh5YSy71IhZHImE00tEZOVcwZJbNZUeGRsHGv4kGbWvEpZUxVYnaV Xlp3P37p12rq MGKGV18pu7j31
ey4jagE1PE1k92Mg7sd0Ki2dhma06NqJJXrSVgghsumr7MQTdZnSp1LPHAAOYKDZRGMcPFE6CUoQ73VePuwyPhJJ
ZPEfc0DfXg879tUy1p2njJCJ1EI2PTxPnxSSsaifxmIeoWipForhXOnF0K9GIaQztIwgeA3huatIomgvdr0XbHw
sOP4La08BTdQS1sALPblayjcvPgkOnBb8VfmieoLw6bIomCL15lwmffypUzYkEQa61KKt
xGCYr6YRIpuAlzxNT0HiWdxGZkSlyZmIn5xMpF17iYmbWzDscOShXGUWesfHgiVAc7pUcfcrejwxRmpol0071rsH
Na728070p0_FYfGSDyuA02Tn0pNTc0c_0bWuY0MTKTyw0ef1R5k0Vgvz34Fgldub0m0mm3TYveCA7mTgc3MU0CL
```

Also we will get the total encryption and decryption time

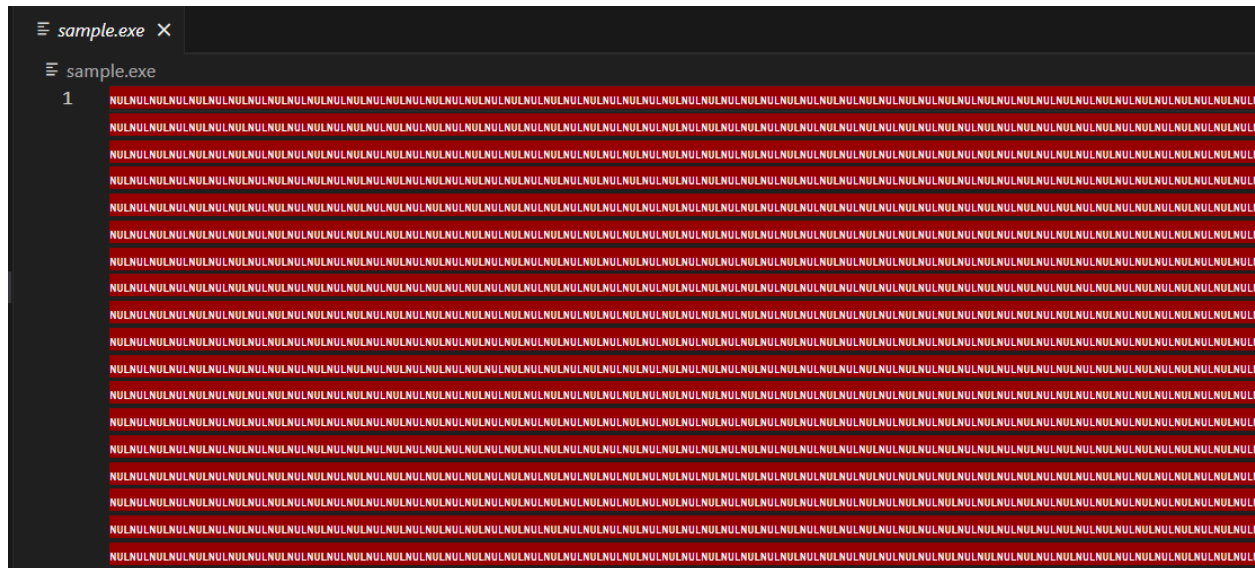
```
ca0830329e7734870d4b9fb7018fc4\redhat.java\jdt_ws\ASSIGNMENT 1_ea12118d\bin
Encryption Time: 1019 milliseconds
Decryption Time: 1037 milliseconds
D:\C++\Users\922725\OneDrive\Desktop\7TH SEMESTER\IC\ASSIGNMENT 1\
```

DES ENCRYPTION/DECRYPTION(EXE FILE)

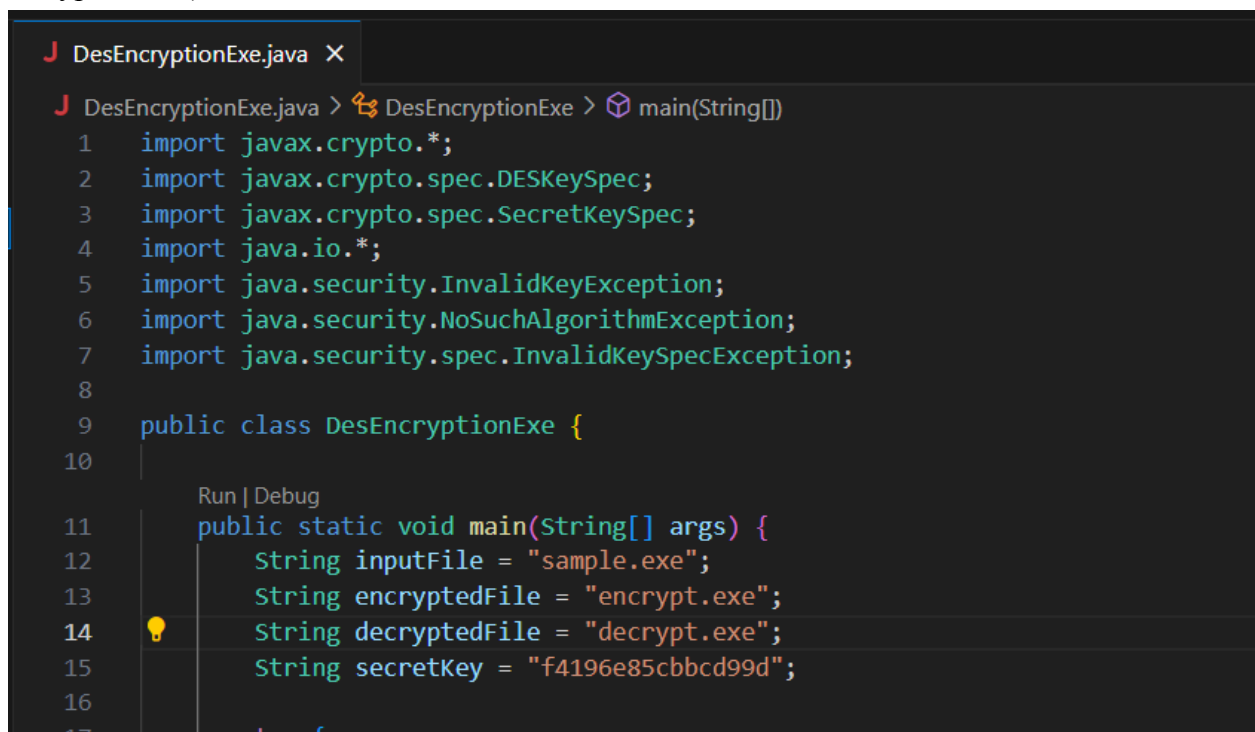
First we generate the exe file(sample.exe) as it is complex to get it from online resources due to its unsecure

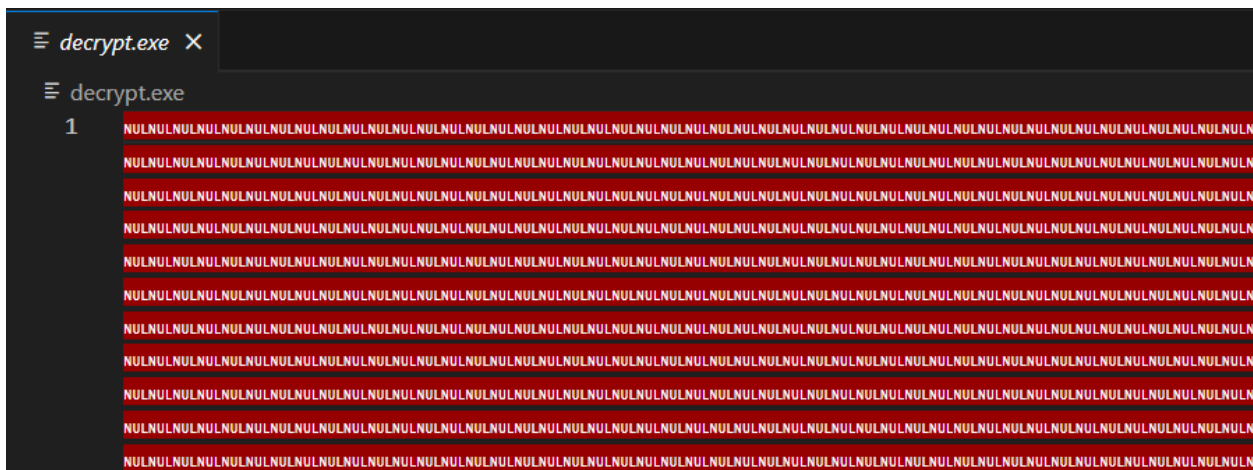
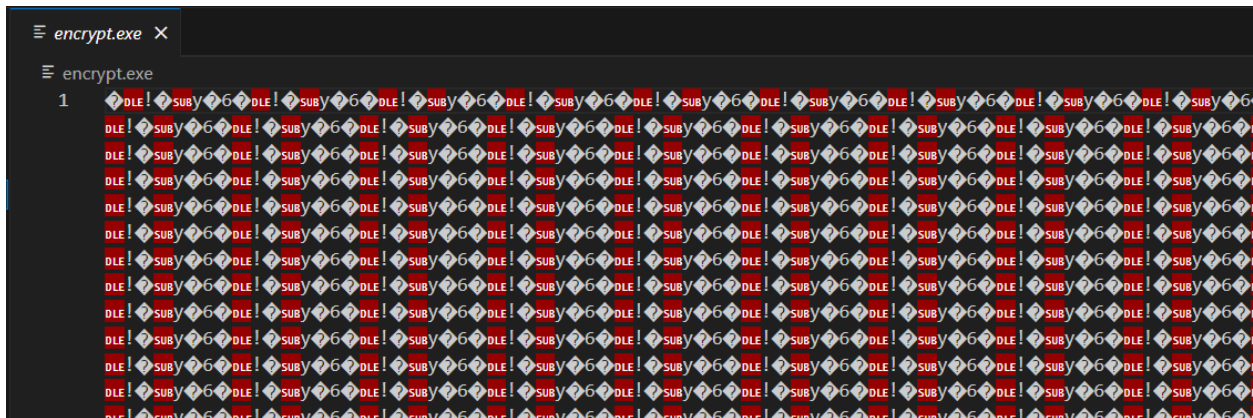
```
GenerateexeFile.java x
GenerateexeFile.java > GenerateexeFile > main(String[])
1  import java.io.FileOutputStream;
2  import java.io.IOException;
3
4  public class GenerateexeFile {
5
6      Run | Debug
7      public static void main(String[] args) {
8          String outputFile = "sample.exe";
9          long fileSizeInBytes = 10 * 1024 * 1024;
10
11          try (FileOutputStream outputStream = new FileOutputStream(outputFile)) {
12              byte[] dummyData = new byte[1024];
13              long bytesWritten = 0;
14              while (bytesWritten < fileSizeInBytes) {
```

Here we have dummy exe file



Now after run the DES .we got two file encrypt.exe(for encrypted data)and decrypt.exe(for decrypted data)





Here is the total time for running this algorithm

```

gram Files (Java\jdk-19\bin\java.exe -XX:+ShowCode
ca0830329e7734870d4b9fb7018fc4\redhat.java\jdt_ws\
Encryption Time: 157 milliseconds
Decryption Time: 188 milliseconds

```

AES ENCRYPTION/DECRYPTION(TXT FILE)

```

PS C:\Users\92335\OneDrive\Desktop\7TH SEMESTER\IS\ASSIGNMENT 17 & C.
ssages' -cp' 'C:\Users\92335\AppData\Roaming\Code\User\workspaceStorag
\bin' 'AesEncryption'
Encryption Time: 31ms
Decryption Time: 24ms

```



```
J AesEncryptionExe.java X
J AesEncryptionExe.java > AesEncryptionExe > main(String[])
1  import javax.crypto.Cipher;
2  import javax.crypto.spec.SecretKeySpec;
3  import java.io.*;
4  import java.security.Key;
5
6  public class AesEncryptionExe {
7      Run | Debug
8      public static void main(String[] args) {
9          try {
10             byte[] keyBytes = "f4196e85cbbcd99d".getBytes();
11             Key secretKey = new SecretKeySpec(keyBytes, "AES");
12             Cipher cipher = Cipher.getInstance("AES");
13
14             String inputFile = "sample.exe";
15             String encryptedFile = "encrypt.exe";
16             String decryptedFile = "decrypt.exe";
17
18             long startTime = System.currentTimeMillis();
19
20             cipher.init(Cipher.ENCRYPT_MODE, secretKey);
21
```

```
J AesEncryptionExe.java  encrypt.exe X
encrypt.exe
1  ^_`GS! b
2  SYN$C`GS b
3  SYN$C`GS b
4  SYN$C`GS b
5  SYN$C`GS b
6  SYN$C`GS b
7  SYN$C`GS b
8  SYN$C`GS b
9  SYN$C`GS b
10 SYN$C`GS b
11 SYN$C`GS b
12 SYN$C`GS b
13 SYN$C`GS b
14 SYN$C`GS b
15 SYN$C`GS b
16 SYN$C`GS b
17 SYN$C`GS b
18 SYN$C`GS b
19 SYN$C`GS b
```