**Lab 2. Symmetric Encryption and Hashing**

**Name:**

Submit your lab to Canvas. Should you have any questions, email Dr. Vance at [anthony@vance.name](mailto:anthony@vance.name).

**Setup:**

If you'd like, I have Cryptool2 and a few other things installed on a ready-to-go virtual machine for students in my class to use. Simply do this:

1. Install one of the VirtualBox platform packages from [here](https://www.virtualbox.org/wiki/Downloads).
2. If you’re on Windows you can skip this step, but if you’re on a Mac:
   * Open VirtualBox, then from the menu, select “Virtual Box” -> “Preferences” -> “Network.”
   * Select “Host-only” tab, then click the plus button on the right to add a new host-only interface. Accept the default name. You just created a host-only virtual interface for VirtualBox guests. This will let your virtual machine guests talk with each other and with your host, but not with anyone else.
3. Get a copy of my Windows .ova file onto your computer. If you didn’t get a copy in class, download it here:

<https://www.dropbox.com/s/dvcib0leancn886/Windows%20InfoSec.ova?dl=0>

1. “File” > “Import Appliance”, then browse to the file obtained in step 3.
2. Try to start the imported appliance. If it complains about network settings needing to be changed:
   * click the “Update network settings” button or whatever it is (I forget).
   * For Adapter 1, select the virtual interface you created in step 2.
   * You should be good-to-go now. (If this is still giving you grief, select “Internal Network” for Adapter 1 instead of “Host-Only” – that will be fine for this lab.)
3. The password is “Password1.”
4. You can now delete the “.ova” file if you’re disk space-starved.

Alternatively, you can install CryptTool 2 (for Windows) here:

<https://www.cryptool.org/modules/mod_ctdownloads/assets/curversion.php>

**Virtual Machine Network Setup**

## **infosec-net Network Map**

The network map is as follows:

| **IP Address** | **Machine** |
| --- | --- |
| 192.168.55.100 | Windows 10 VM |

## **Setting up Virtualbox and the infoset-net network**

I pre-prepare the virtual machines to be on the same virtual network so that they have internet access and so that they can talk to one another. However, you must initialize the virtualbox network after you install virtualbox. To do so:

**Note:** Ideally you will have at least 8 GB memory (RAM)

1. First, install one of the VirtualBox platform packages from [here](https://www.virtualbox.org/wiki/Downloads).
2. Then, create the virtual network.
   * If on Windows, download and run [this script](https://www.dropbox.com/s/923mt76knrherrm/create-infosec-natnetwork.bat?dl=0)
   * If on Mac:
     + Open Terminal application
     + Copy-paste the following lines into the terminal (each bullet is one line), and press enter
     + vboxmanage natnetwork add --netname infosec-net --network 192.168.55.0/24 --enable --dhcp off
     + vboxmanage natnetwork start --netname infosec-net

You can test whether the scripts were successful by navigating to Virtuabox > File > Preferences > Network, where you should see “infosec-net” in the list of networks. If you don’t see that network, then you can manually create the network from this dialog prompt by clicking the plus and using the following options. Then click ‘ok’, ‘ok’.

\* Enable Network: `checked`

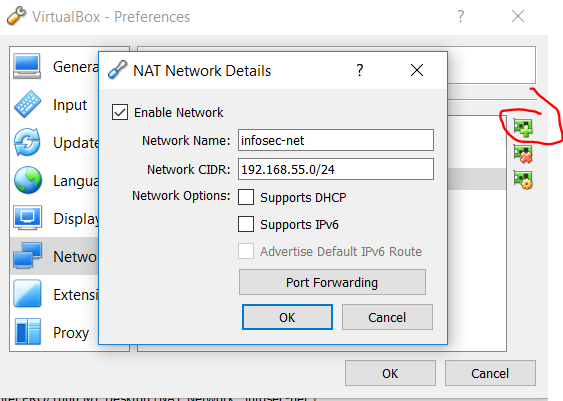
\* Network Name: `infosec-net` (case-sensitive!)

\* Network CIDR: `192.168.55.0/24`

\* Network Options:

\* Supports DHCP: `unchecked`

\* Supports IPv6: `unchecked`



**Part 1**

**Polyalphabetic substitution and Transposition Ciphers**

1. Using polyalphabetic substitution, with alphabet a–z (no caps) and a space (as the last character of the alphabet), use the private key *security* to encrypt the plaintext message:

send money

To help you, use the following formula:

Encryption: ciphertext = (plaintext + key) mod 27

Decryption: plaintext = (ciphertext - key) mod 27

**Important:** Number your alphabet so that it starts with zero, e.g., A = 0.

**Answer:**

2. Consider the transposition method discussed in class. Assume the plaintext is

attack postponed until two am

and the key is {4312567}

Create the ciphertext as was demonstrated in class. Pad the message so that there is an even block of characters, meaning that each column will have the same number of characters. One way to do this is to pad the block by repeating the message until the block is complete. Show your result.

**Answer:**

**Frequency analysis**

[Download](https://www.cryptool.org/modules/mod_ctdownloads/assets/curversion.php) and install CrypTool 2 (Windows only).

Open CrypTool 2.0. Under Templates, select Cryptanalysis, then Classical, and double-click on Frequency Analysis. Paste the cipher text below into the Text Input box and hit the Play button. The results of the analysis will show up in the box on the right.

Use CrypTool’s cryptanalysis function, “Frequency Test,” to analyze the following message:

Hvs Qosgof qwdvsf wg boasr othsf Xizwig Qosgof, kvc, oqqcfrwbu hc Gishcbwig, igsr wh kwhv o gvwth ct hvfss hc dfchsqh asggousg ct awzwhofm gwubwtwqobqs. Kvwzs Qosgof'g kog hvs twfgh fsqcfrsr igs ct hvwg gqvsas, chvsf gipghwhihwcb qwdvsfg ofs ybckb hc vojs pssb igsr sofzwsf.

Wt vs vor obmhvwbu qcbtwrsbhwoz hc gom, vs kfchs wh wb qwdvsf, hvoh wg, pm gc qvobuwbu hvs cfrsf ct hvs zshhsfg ct hvs ozdvopsh, hvoh bch o kcfr qcizr ps aors cih. Wt obmcbs kwgvsg hc rsqwdvsf hvsgs, obr ush oh hvswf asobwbu, vs aigh gipghwhihs hvs tcifhv zshhsf ct hvs ozdvopsh, boaszm R, tcf O, obr gc kwhv hvs chvsfg.

—Gishcbwig, Zwts ct Xizwig Qosgof 56

Vwg bsdvsk, Oiuighig, ozgc igsr hvs qwdvsf, pih kwhv o fwuvh gvwth ct cbs, obr wh rwr bch kfod ofcibr hc hvs psuwbbwbu ct hvs ozdvopsh:

Kvsbsjsf vs kfchs wb qwdvsf, vs kfchs P tcf O, Q tcf P, obr hvs fsgh ct hvs zshhsfg cb hvs goas dfwbqwdzs, igwbu OO tcf L.

—Gishcbwig, Zwts ct Oiuighig 88

Hvsfs wg sjwrsbqs hvoh Xizwig Qosgof igsr acfs qcadzwqohsr gmghsag og kszz, obr cbs kfwhsf, Oizig Uszzwig, fstsfg hc o (bck zcgh) hfsohwgs cb vwg qwdvsfg:

Hvsfs wg sjsb o fohvsf wbusbwcigzm kfwhhsb hfsohwgs pm hvs ufoaaofwob Dfcpig qcbqsfbwbu hvs gsqfsh asobwbu ct zshhsfg wb hvs qcadcgwhwcb ct Qosgof'g sdwghzsg.

—Oizig Uszzwig, Ohhwq Bwuvhg

Wh wg ibybckb vck sttsqhwjs hvs Qosgof qwdvsf kog oh hvs hwas, pih wh wg zwyszm hc vojs pssb fsogcbopzm gsqifs, bch zsogh psqoigs acgh ct Qosgof'g sbsawsg kcizr vojs pssb wzzwhsfohs obr chvsfg kcizr vojs oggiasr hvoh hvs asggousg ksfs kfwhhsb wb ob ibybckb tcfswub zobuious. Hvsfs wg bc fsqcfr oh hvoh hwas ct obm hsqvbweisg tcf hvs gczihwcb ct gwadzs gipghwhihwcb qwdvsfg. Hvs sofzwsgh gifjwjwbu fsqcfrg rohs hc hvs 9hv qsbhifm kcfyg ct Oz-Ywbrw wb hvs Ofop kcfzr kwhv hvs rwgqcjsfm ct tfseisbqm obozmgwg.

Qosgsf\_qwdvsf, Kwywdsrwo

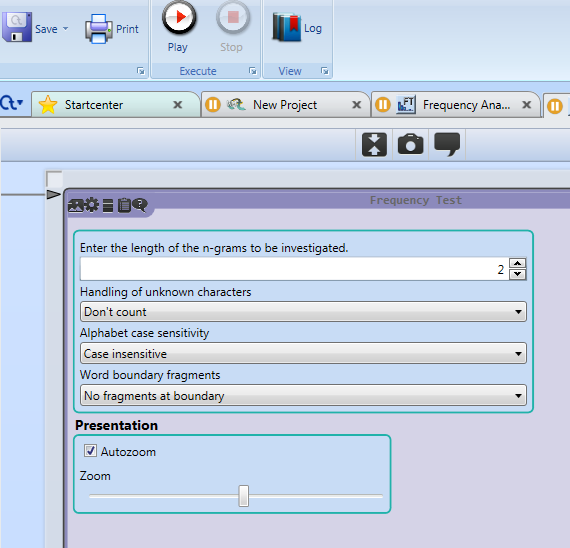
**Questions:**

1. Examine the frequency table in the Frequency Test component (drag the component until the frequency chart is visible. What are the four most frequent characters, in descending order? Examine the frequency table chart at <http://en.wikipedia.org/wiki/Frequency_analysis>. What does this suggest the four letters identified correspond to?

Question: According to CrypTool, what are the four most frequent characters?

**Answer:**

2. Do the same analysis, but for character pairs (bigrams) or trigrams. You can either open up a new Frequency Test or hit the Stop button and modify the test you’re currently running. Under the settings tab of the Frequency Test box, you can change the length of n-grams. Hit Play and then view the Presentation tab for a graphical view of your results. What are the three most frequent N-grams in your analysis? Examine the frequency table chart at <http://en.wikipedia.org/wiki/Bigram> or <http://en.wikipedia.org/wiki/Trigram>. What does this suggest the three N-grams identified correspond to?



**Answer:**

3. Given the above information, what is the key (the number of characters shifted)? Don’t be proud; you can use your fingers to count.

**Answer:**

4. What is the plaintext? (Hint: Which cipher shifts all the characters by a fixed number? It’s listed as one of the Classic Ciphers in this tool.)

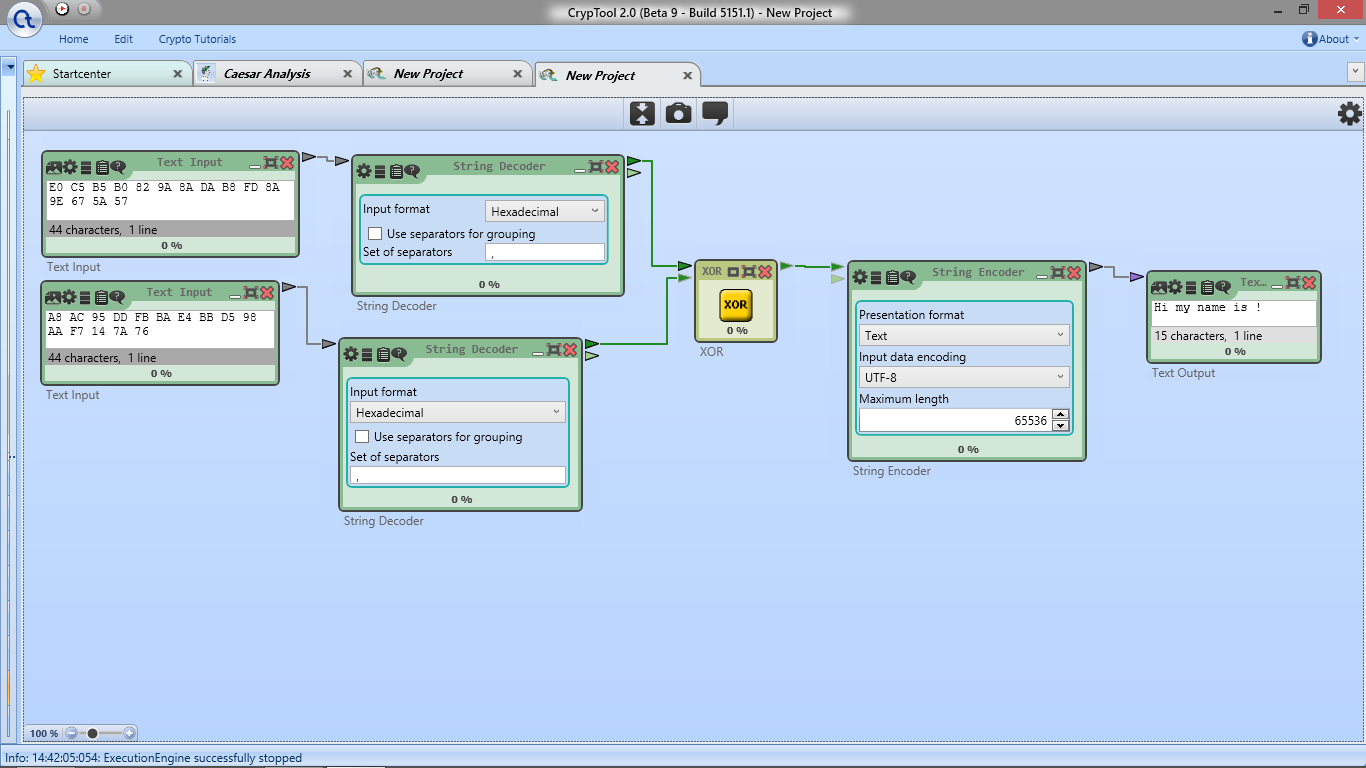
**Answer:**

**One-time Pad**

Using Cryptool2, decrypt the following ciphertext using the following one-time pads.

**Note:** Click “New” > “Workspace” to get started. You’ll need to use the XOR function under “Classic Ciphers” to do the encryption. You’ll also need to use the “String Decoder” to decode the string as hexadecimal before passing the stream to the XOR function (see screenshot below). I like to use the Tools tab on the bottom-left and its search bar to drag components such as XOR and “String Decoder” into my project.

Use the screenshot below to see how to connect the components. Put the ciphertext into one text input box and the one-time pads in the other text input box, one at a time. Click-and-drag the component arrow connectors over to an appropriate connector arrow on the next component (it may not be the same one as shown in the screenshot—the CrypTool2 tooltip guides will help you).



Ciphertext (hexadecimal): E0 C5 B5 B0 82 9A 8A DA B8 FD 8A 9E 67 5A 57

One-time pad 1: A1 B1 C1 D1 E1 F1 AA BB CC DD EE FF 10 34 76

**Answer:**

One-time pad 2: B2 A0 C1 C2 E7 FB FE FA D9 89 AA AF 56 6A 67

**Answer:**

One-time pad 3: B3 B0 C7 C2 E7 F4 EE BF CA DD EC F1 15 2E 76

**Answer:**

One-time pad 4: B4 AD D0 90 E1 FB FE FA D1 8E AA FA 02 3B 33

**Answer:**

Question: How many possible ASCII plaintexts are there for this ciphertext? What does the above tell you about the security of one-time pads, if each of the above one-time pads are equally likely?

**Answer:**

**Random Number Generators**

1. Examine the random number generators built in to your favorite programming language. Give links to online documentation for these generators below:

**Answer:**

2. Would you use these random number generators for cryptographic purposes? Why or why not?[[1]](#footnote-1)

**Answer:**

**Part 2**

**Hashing**

Find a file and hash it with multiple hashing algorithms (e.g., using md5deep.exe, sha1deep.exe, and sha256deep.exe, command-line tools installed on the windows vm).

* You can use “hash-me.txt” on the Windows vm desktop
* Shft+Right-click on the desktop and choose “Open PowerShell window here”
* The command-line tools are available on the “path”, which means you can type their name followed by the filename to get a hash. E.g., “md5deep hash-me.txt”

If you’re not using the lab VM, and using a Mac, use the built-in command “shasum” or “md5”.

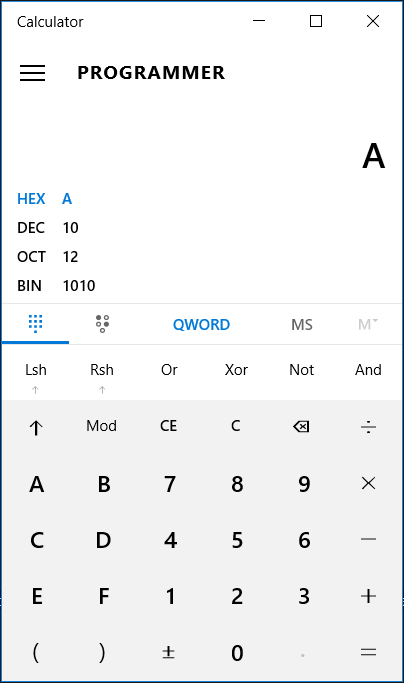
If you’re using Windows, you can download the command-line tools yourself from [here](https://github.com/jessek/hashdeep/releases/download/v4.4/md5deep-4.4.zip). You'll unzip them and then open a command prompt. Then you can either call the full path to the tools (e.g., c:\path\to\md5deep.exe), or you can Windows key > "Edit the System environment variables" > "Environment Variables" > select "Path" > "Edit" > "New" > "Browse" > (Browse to the path where you unzipped your md5deep stuff, the folder where the .exe's are). Hit Ok > Ok > Ok (you have to do all three before it applies) then open a new command prompt / PowerShell window. Voila, you should have the tools available on your path now.

**Questions**:

1. What is the length of each algorithm hash in characters? How many bits does each hash represent?

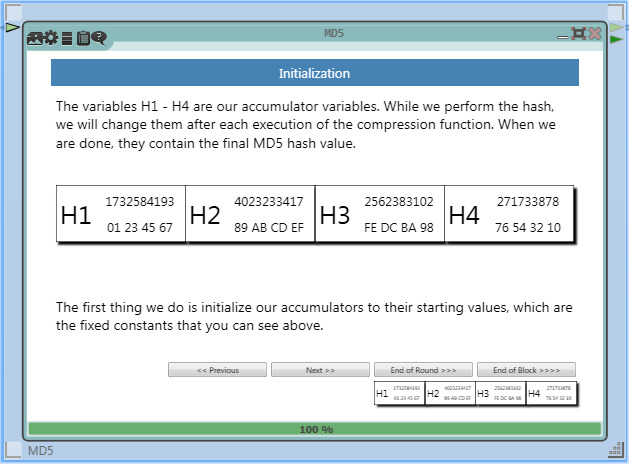
**Answer:**

Make a copy of your file and open the copy in HxD. (HxD is available on the desktop.) Use the hexeditor to change just a single bit (e.g, change an “A” — 1010 to a B — 1011. Use Windows Calculator in the “Programmer” mode to be sure that your hexcode change represents a change of only one bit.) Save the modified file and hash it again. Since you only changed a single bit in a file of billions of bits, you might reason that the hashes would be nearly the same. Are they? Explain why.



**Answer:**

**Optional**: Use the MD5 function in CrypTool to hash a file. Look at the graphical depiction for how it works.

****

**Hash Collisions**

You’ll find the files you need for this section [here](https://drive.google.com/file/d/1zZD-jcBSm-dMuW6OXs8G5DdM6hqlM3tS/view?usp=sharing). Hash the following files using multiple hashing algorithms (e.g., using MD5Deep, SHA1Deep, and SHA256Deep).

programA.exe

programB.exe

Are the files the same? Explain.

**Answer:**

**Symmetric encryption with AES:**

Decrypt the message below using AES and the following parameters:

Message (base64):



Algorithm: AES

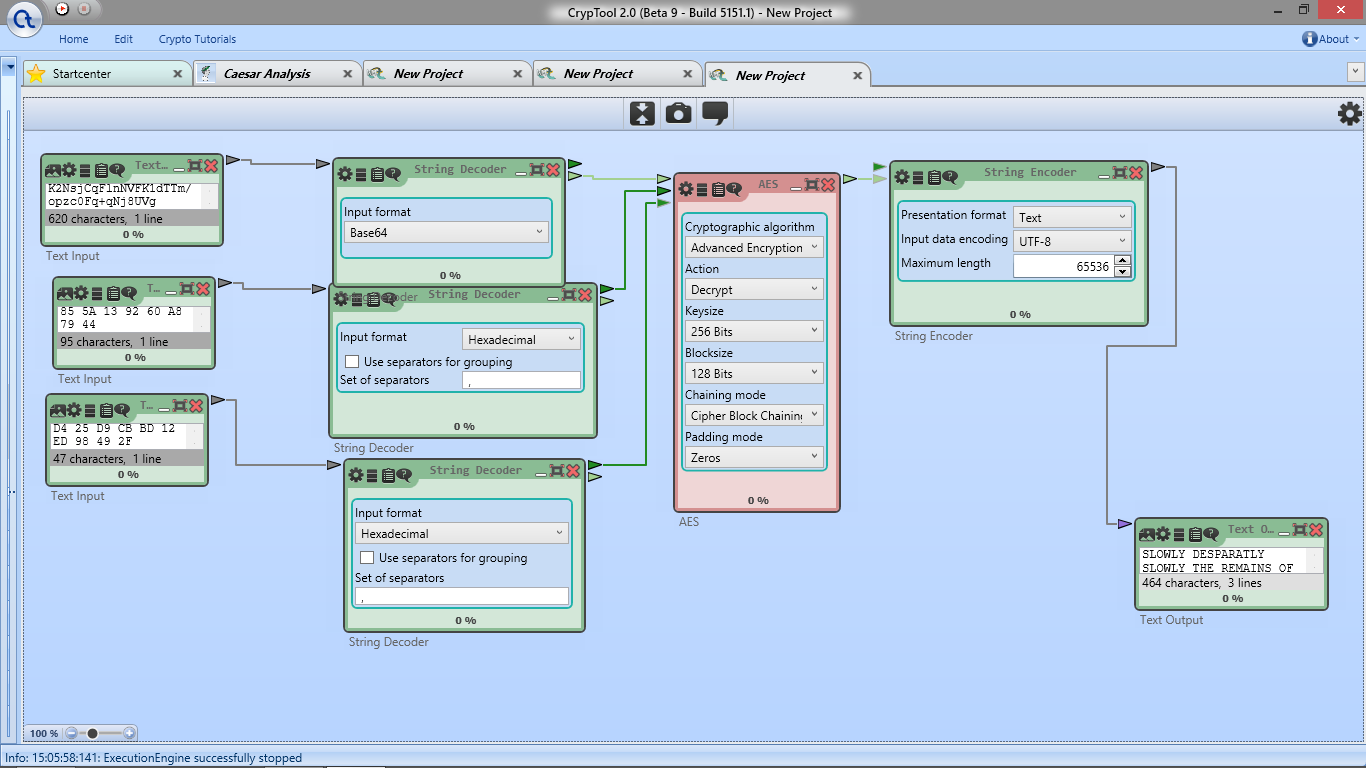
Chaining mode: CBC

Padding mode: Zeros

IV (hexadecimal): 15 7F 14 FF DF D1 D4 25 D9 CB BD 12 ED 98 49 2F

Key (hexadecimal):

FA 71 7D 51 3F D7 32 7D D9 C7 13 84 43 47 C2 5B 2F 9C 1E 48 67 C6 81 32 85 5A 13 92 60 A8 79 44



Questions:

1. What is the plaintext of the message?

**Answer:**

1. How long is the IV in bits?

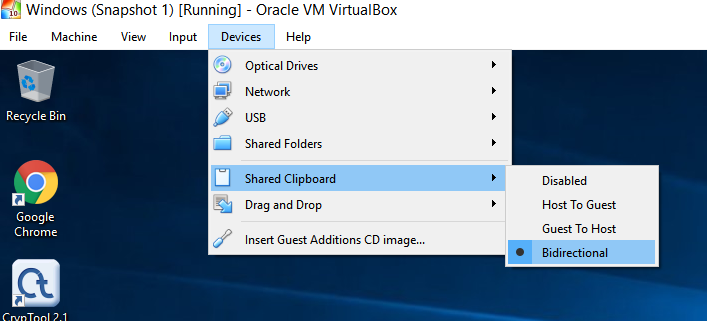
**Answer:**

1. How long is the key in bits?

**Answer:**

**Message Sharing**

Use AES or another symmetric block cipher to encrypt/decrypt messages exchanged between you and another class member. (You could email encrypted messages to one another, and decrypt using CryptTool. You can copy-paste from the host into your VM if you enable the bidirectional clipboard).



Questions:

1. How did you ensure that the key exchange was safe? How would you exchange keys if you weren’t in the same location?

**Answer:**

1. Was it important that the other parameters of the block cipher (algorithm name, mode of operation, IV (if any), key length, etc.) was kept secret?

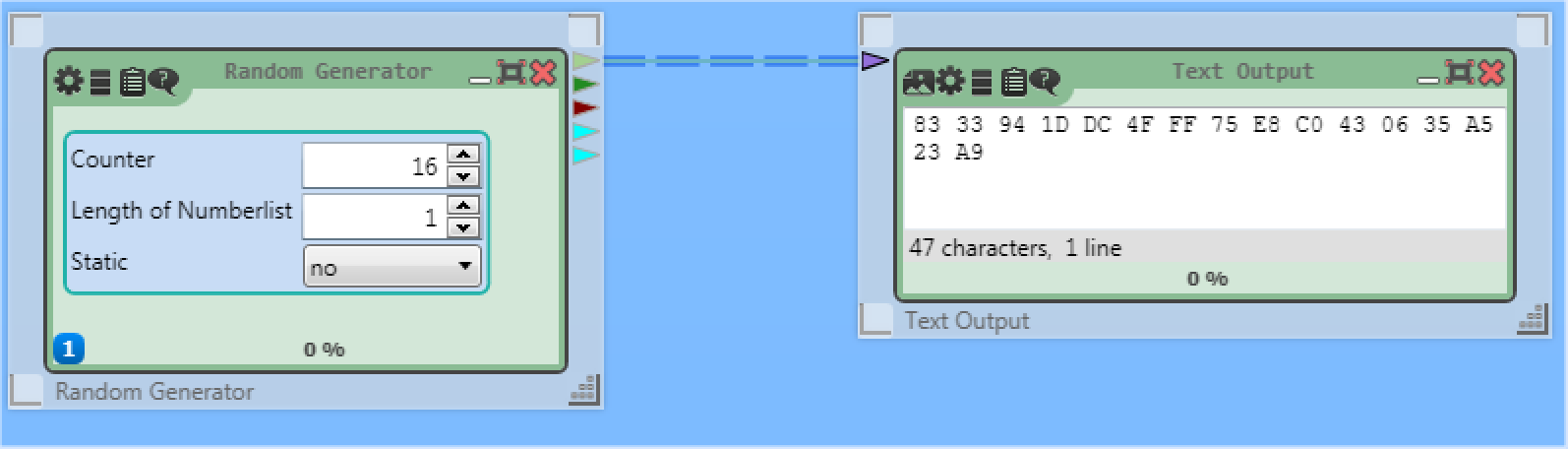
**Answer:**

**Block Cipher Modes of Operation**

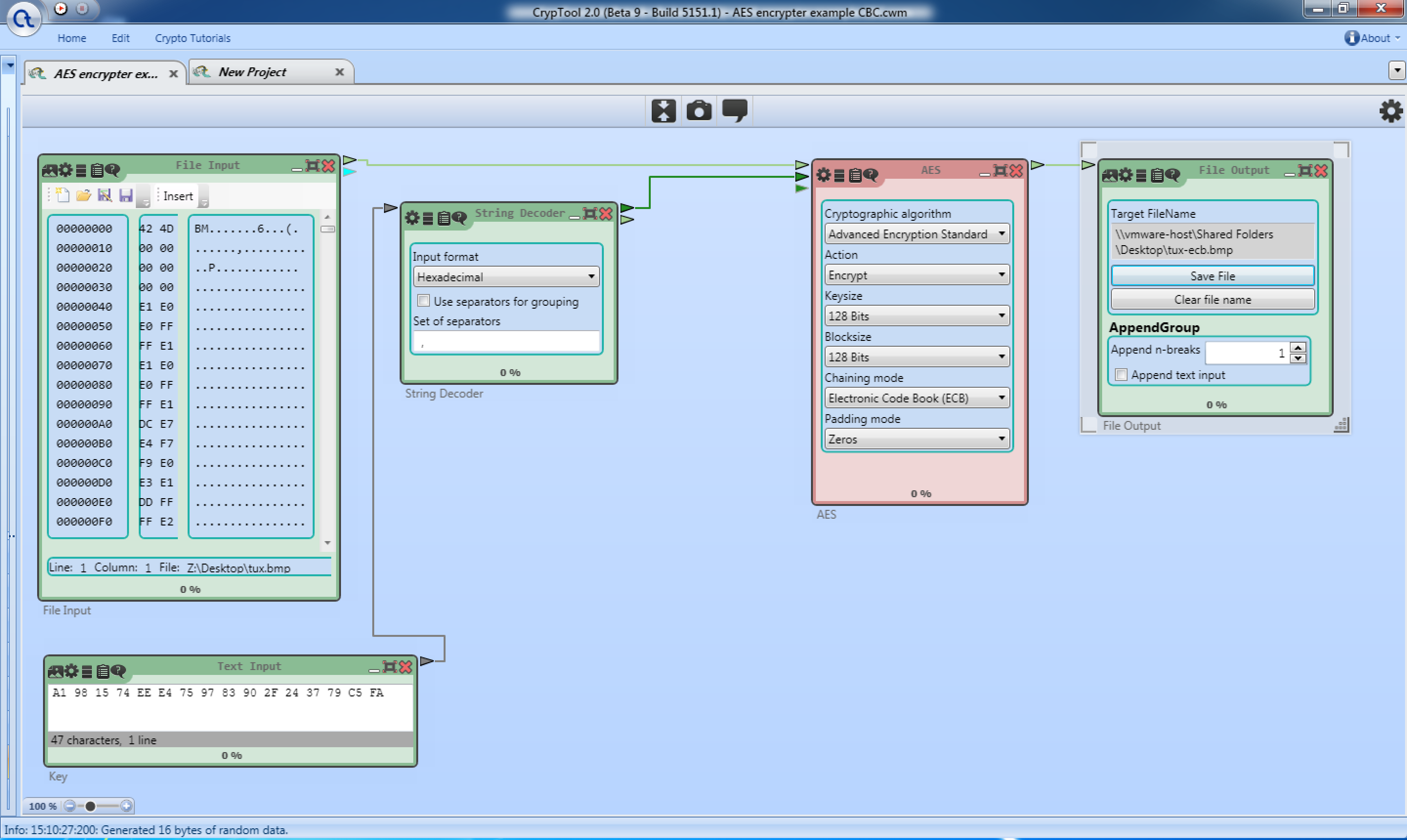
1. Using Cryptool, use the File Input and File Output tools to encrypt the image tux.bmp contained in the zip file you copied from the Source folder. Using AES in Electronic Code Book (ECB) mode, and save the resulting file as “tux\_ecb.bmp”. Then, use HxD or other hexeditor to insert the first 56 bytes (the file signature header that identifies the file as a BMP file) of tux.bmp at the beginning of tux-ecb.bmp. Open both files in an image viewer program and compare.
2. Encrypt tux.bmp again with AES but this time use Cipher Block Chaining (CBC) mode. Save the resulting file as tux\_cbc.bmp. Visually compare tux.bmp, tux\_ebc.bmp, and tux\_cbc.bmp.

**Note**: Remember that CBC requires a random initialization vector (IV) that is the same size as the block.

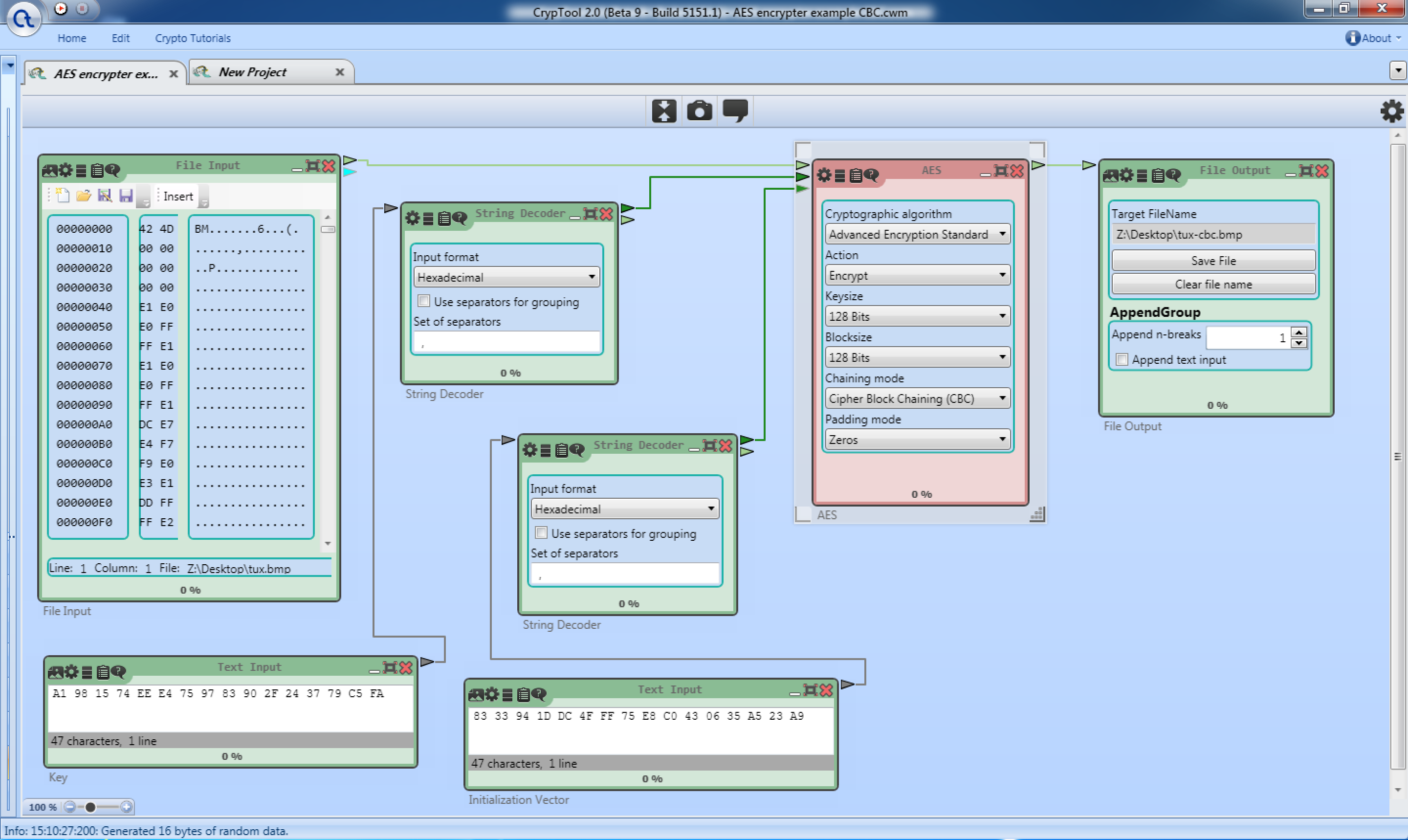
**Note**: You can use Cryptool’s random number generator to create a key and IV:



AES in ECB Mode:



AES in CBC Mode:



Question: Explain the results you see for the EBC and CBC modes of AES. What advantages do you see for using CBC over EBC?

**Note:** BMP files use no compression, meaning there is a lot of redundant information. The above test won’t work with a format like JPEG that uses compression.

**Answer:**

**Lifelong Learning**

To learn more about cryptography, the following are some of the best and most readable books.

|  |  |
| --- | --- |
| Macintosh HD:Users:anthony:Desktop:51RRKXRP0RL._SY346_PJlook-inside-v2,TopRight,1,0_SH20_.jpg | “The Code Book” by Simon Singh.  This is a very interesting and gripping book about the history and intrigue of cryptography and cryptanalysis.  Available at [Temple Library](https://librarysearch.temple.edu/catalog/991004378459703811)  Amazon: <http://amzn.com/0470474246> |
| Macintosh HD:Users:anthony:Desktop:41Tuv9saPgL._SY346_PJlook-inside-v2,TopRight,1,0_SH20_.jpg | “Crypto: How the Code Rebels Beat the Government Saving Privacy in the Digital Age” by Steven Levy.  A very engaging look at the modern history of cryptography, including the development of DES, RSA, and PGP. Also, it describes the fight in the 1990’s to legalize the use of strong cryptography.  Available at [Temple Library](https://librarysearch.temple.edu/catalog/991013963759703811)  Amazon: <http://amzn.com/0140244328> |
| Macintosh HD:Users:anthony:Desktop:51oEJqoe5RL._SY346_PJlook-inside-v2,TopRight,1,0_SH20_.jpg | “The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet” by David Kahn.  The bible of the history of Cryptography. A deep dive (at 1,200 pages) but still readable.  Amazon: <http://amzn.com/0684831309> |
| Macintosh HD:Users:anthony:Desktop:41bQahpPfJL.jpg | “Cryptography Engineering: Design Principles and Practical Applications” by  Niels Ferguson, Bruce Schneier, Tadayoshi “Yoshi” Kohno.  The updated successor to the classic “Applied Cryptography” by Bruce Schneier. This book explains how modern crypto works and how to implement it in your systems.  Amazon: <http://amzn.com/0470474246> |

1. Taken in part from Ferguson et al. 2010. [↑](#footnote-ref-1)