**Lab 8**

**Secret-Key Encryption Lab**

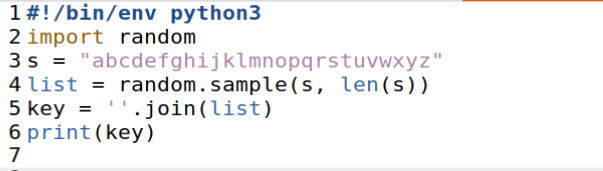
**Name: GAURAV SETTY**

**Email: settgm01@pfw.edu**

**Task 1 : Frequency Analysis**

**Step 1 : Generating the Key :**

**Code :**





**Step 2 : Simplification**

Article.txt

Since they are still preserved in the rocks for us to see, they must have been formed quite recently, that is, geologically speaking. What can explain these striations and their common orientation? Did you ever hear about the Great Ice Age or the Pleistocene Epoch? Less than one million years ago, in fact, some 12,000 years ago, an ice sheet many thousands of feet thick rode over Burke Mountain in a southeastward direction.



Plaintext.txt:

since they are still preserved in the rocks for us to see they must have been formed quite recently that is geologically speaking what can explain these striations and their common orientation did you ever hear about the great ice age or the pleistocene epoch less than one million years ago in fact some years ago an ice sheet many thousands of feet thick rode over burke mountain in a southeastward direction

**Step 3: Encryption**



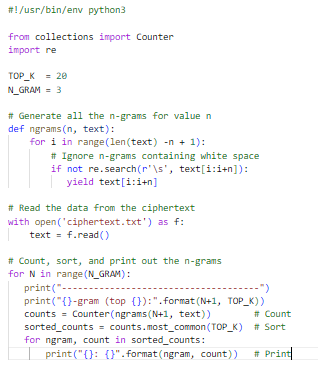
Ciphertext.txt:

lbgtw hqwz saw lhbpp fawlwauwr bg hqw aktdl ika yl hk lww hqwz vylh qsuw xwwg ikavwr mybhw awtwghpz hqsh bl nwkpknbtsppz lfwsdbgn oqsh tsg wjfpsbg hqwlw lhabshbkgl sgr hqwba tkvvkg kabwghshbkg rbr zky wuwa qwsa sxkyh hqw nawsh btw snw ka hqw fpwblhktwgw wfktq pwll hqsg kgw vbppbkg zwsal snk bg isth lkvw zwsal snk sg btw lqwwh vsgz hqkylsgrl ki iwwh hqbtd akrw kuwa xyadw vkyghsbg bg s lkyhqwslhosar rbawthbkg

The text was encrypted.

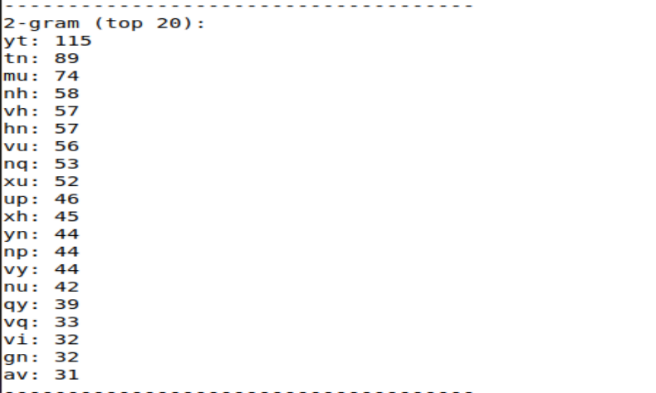
**Deciphering:**

**Code:**

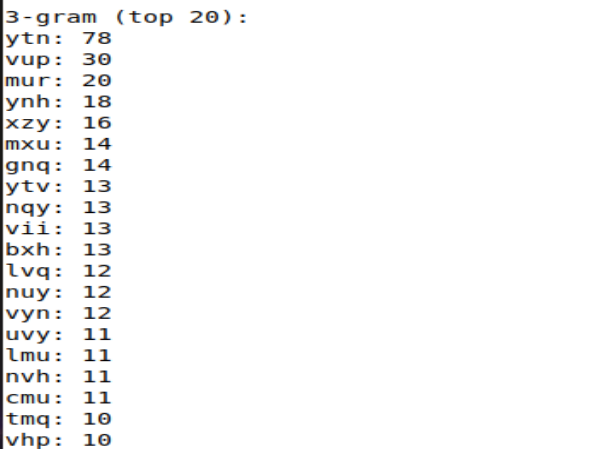


Now we’ll check the n-grams:

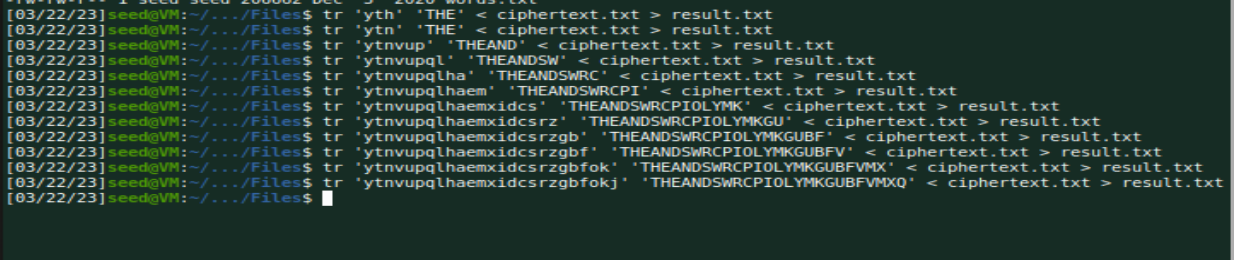
1-gram: 2-gram:

3-gram:



Now, we’ll run the commands for deciphering:



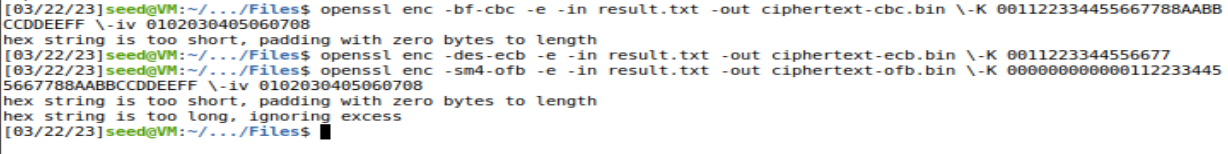
Deciphered text:



**Task 2: Encryption using Different Ciphers and Modes**

I’m using the following 3 encryption modes:

1. sm4-ofb
2. bf-cbc
3. des-ecb



Ciphertext-ofb.bin:



Ciphertext-cbc.bin:



Ciphertext-ecb.bin:



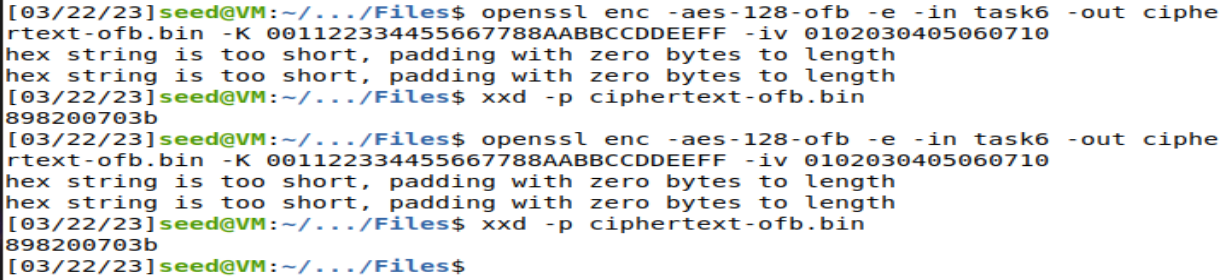
**Task 6: Initial Vector (IV) and Common Mistakes**

**Task 6.1: IV Experiment**

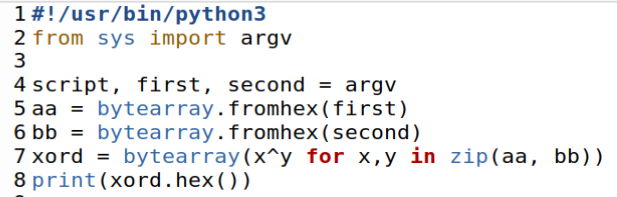
**Using different IV:**



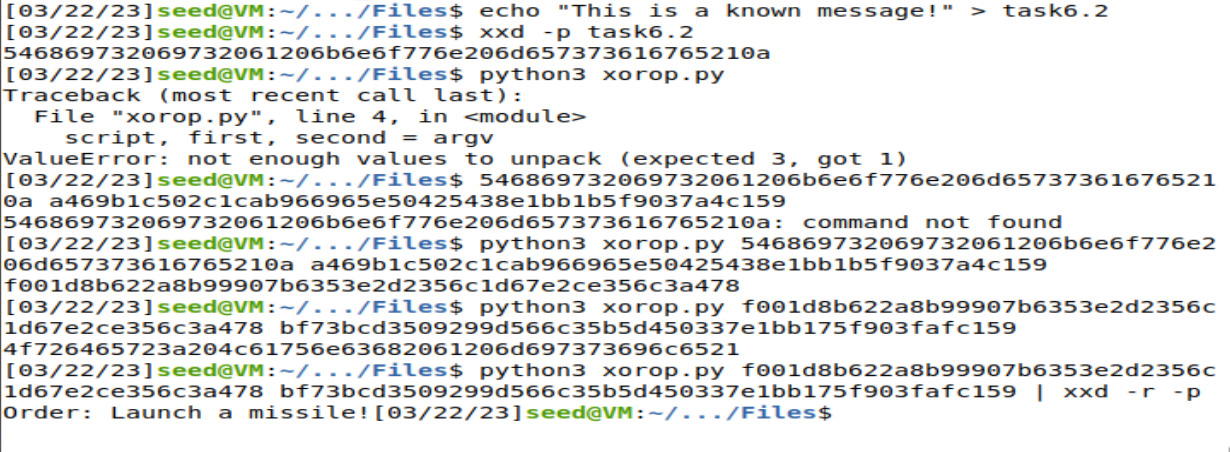
**Using same IV:**



**Task 6.2: Common Mistake: Use the Same IV**



Inserting the message into P1 and obtaining P1's HEX value. P1 and C1 are XORed, and the output is an XOR with C2. The outcome is the same.

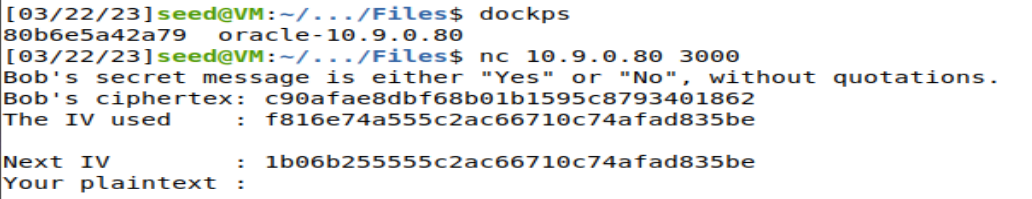


In CFB mode, by performing an XOR between P1 and the encryption function's output using the same Initialization vector, an attacker with access to both the plain text P1 and the cipher text C1 can obtain the keystream used to encode P1.

The attacker, however, is unable to directly decrypt C2 using the keystream because it relies on the ciphertext of the block before it when using CFB mode to encrypt one block of plaintext. The attacker is unable to recover the complete message from it as a result.

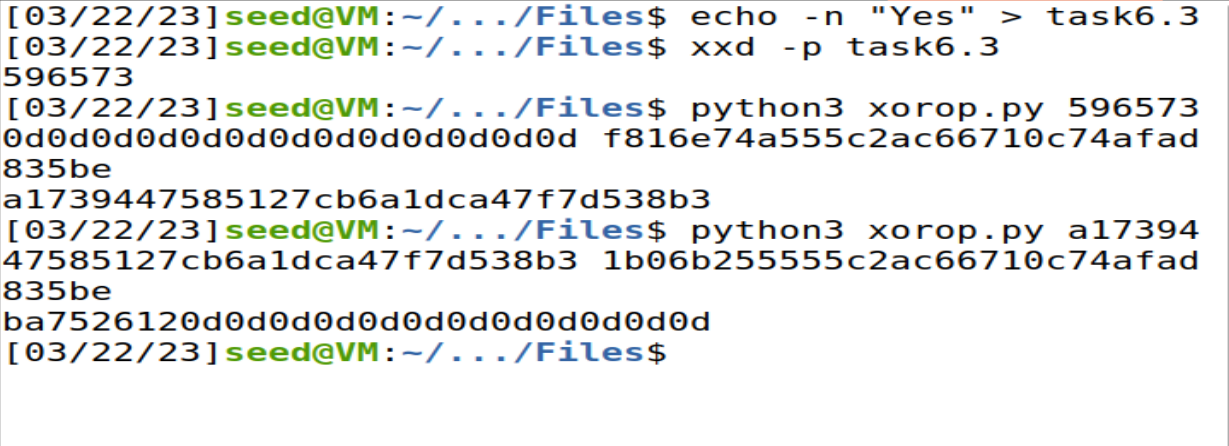
**Task 6.3: Common Mistake: Use the Same IV**

Run the nc command:

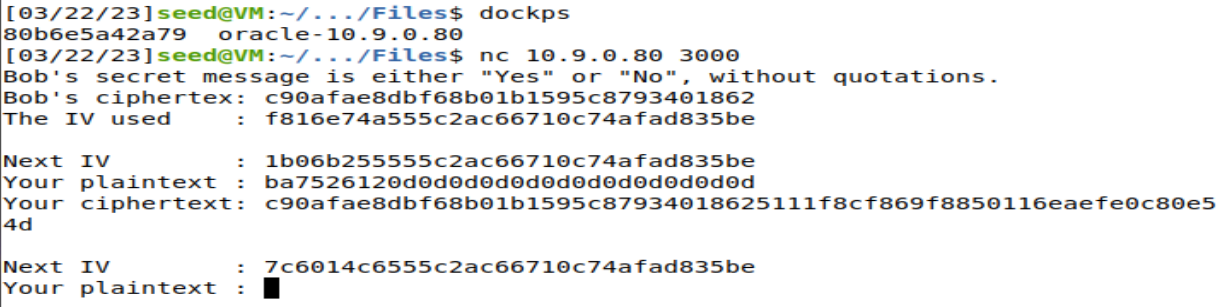


Putting Yes into PT1 to see if Bob's Cipher text fits that. PT1's hex code is verified. Padding added to PT1 unicode value: **5965730d0d0d0d0d0d0d0d0d0d0d0d0d**

XORing the PT1 HEX value with the IV and the resulting value with the Next IV.



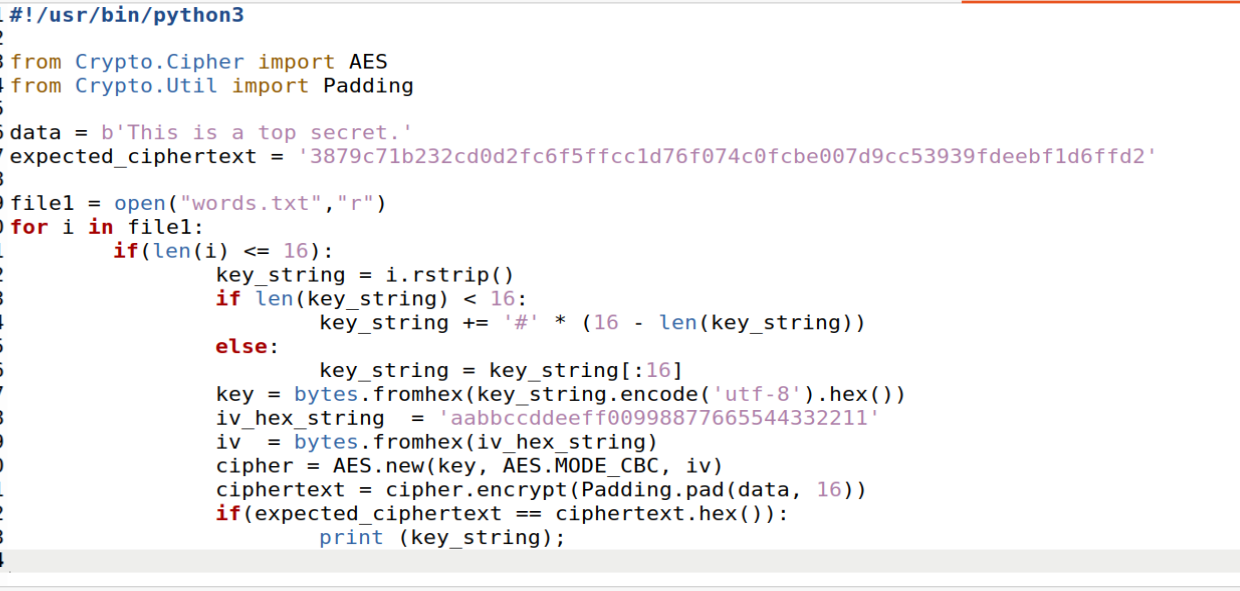
Now, let’s check the result:



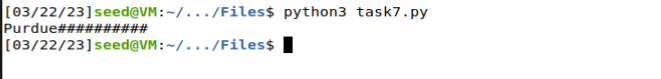
We can see that Bob's Cipher text matches the new "Your Ciphertext" column. Bob's Simple Text was Yeah, as shown by His. As the remaining bytes are for padding, we just take into account the first 16 bytes.

**Task 7: Programming using the Crypto Library**

**Code:**



Result:



We can see that Purdue used as the encryption key.