Jonathan Kenney (M08837382) and Brennan Thomas (M10668733)
Data Security and Privacy
Project 3: Searchable Encryption

Setup

Install a stable version of Python 3.6+

Use pip to install the cryptography package

```
pip3 install cryptography
```

The following experiment ran using both Ubuntu 18.04 and Windows 10, however the program should run on any Python-enabled OS with the above requirements (i.e. Python 3.6+ and dependencies).

Execution

First, create desired input files. These files should be located in the data/files folder and follow the naming convention f#.txt where # is a positive integer greater than 0.

From **TOP DIRECTORY**, run in the following order:

keygen.py

```
python3 build/keygen.py 32
```

NOTE: key size parameter can be 16, 24, or 32

enc.py

```
python3 build/enc.py
```

tokengen.py

```
python3 build/tokengen.py [KEYWORD]
```

NOTE: select keyword from various input files, punctuation is stripped

search.py

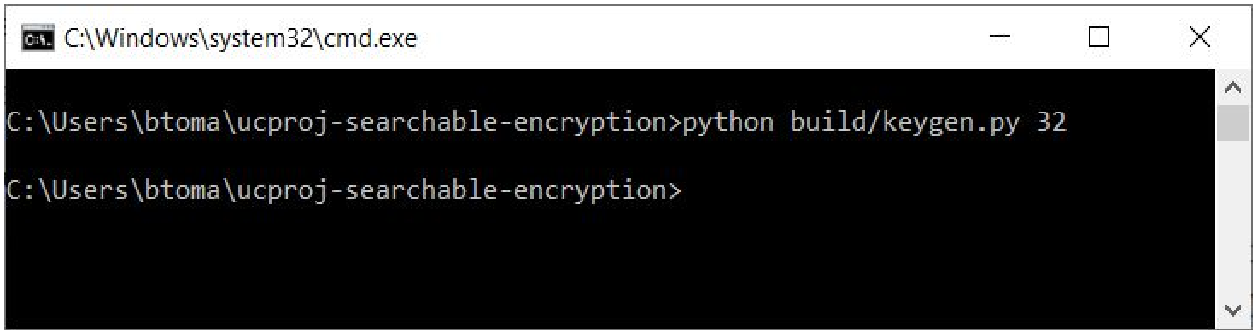
```
python3 build/search.py
```

Search results will be in data/result.txt

Results

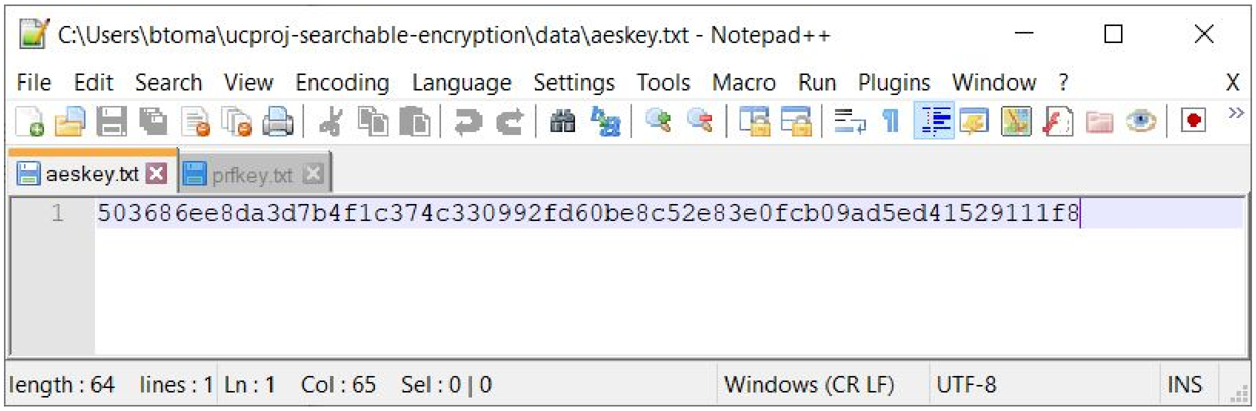
Key Generation

The key generation function generates uniformly distributed random numbers to use as keys with a number of bytes equal to the given input argument. In this case, the argument given is 32 bytes (256 bits).



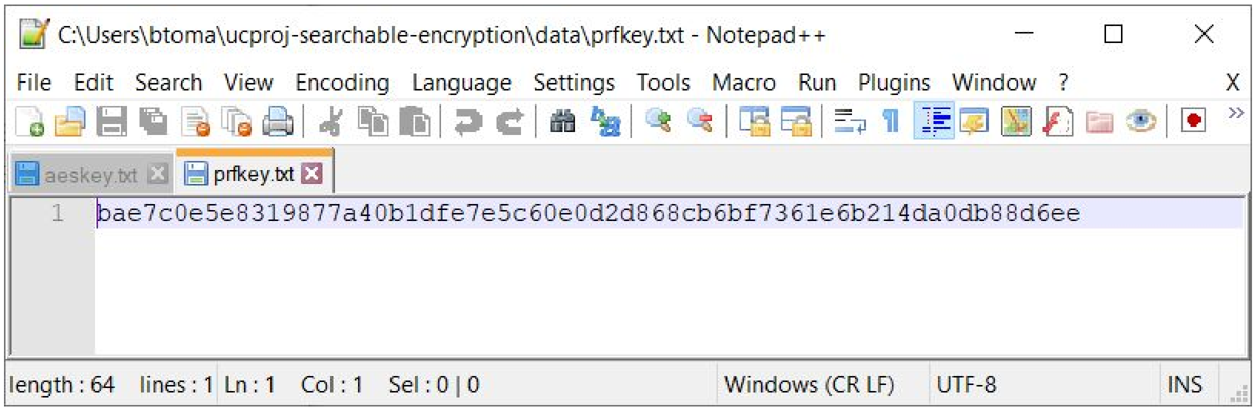
```
C:\Users\btoma\ucproj-searchable-encryption>python build/keygen.py 32

C:\Users\btoma\ucproj-searchable-encryption>
```

Figure 1: *Running the keygen process with a parameter of 256 bits.*


C:\Users\btoma\ucproj-searchable-encryption\data\aeskey.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
aeskey.txt
prfkey.txt
length : 64 lines : 1 Ln : 1 Col : 65 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

```
503686ee8da3d7b4f1c374c330992fd60be8c52e83e0fcb09ad5ed41529111f8
```

Figure 2: *The resulting AES key.*

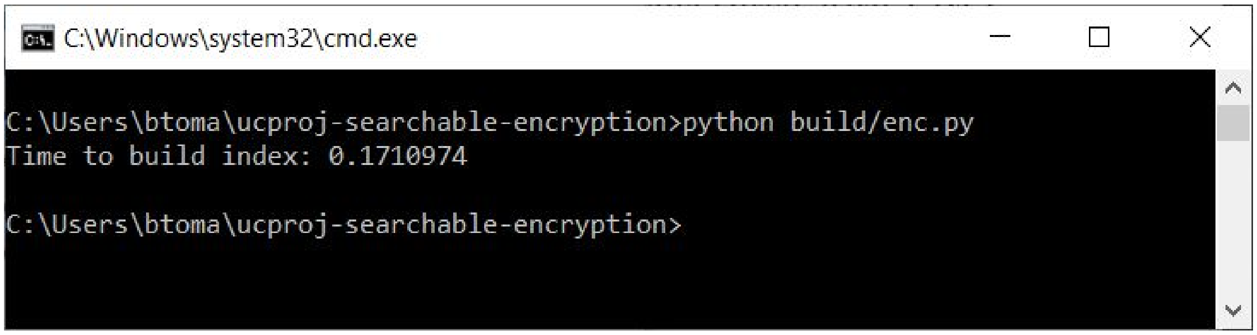

C:\Users\btoma\ucproj-searchable-encryption\data\prfkey.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
aeskey.txt
prfkey.txt
length : 64 lines : 1 Ln : 1 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

```
bae7c0e5e8319877a40b1dfe7e5c60e0d2d868cb6bf7361e6b214da0db88d6ee
```

Figure 3: *The resulting PRF key.*

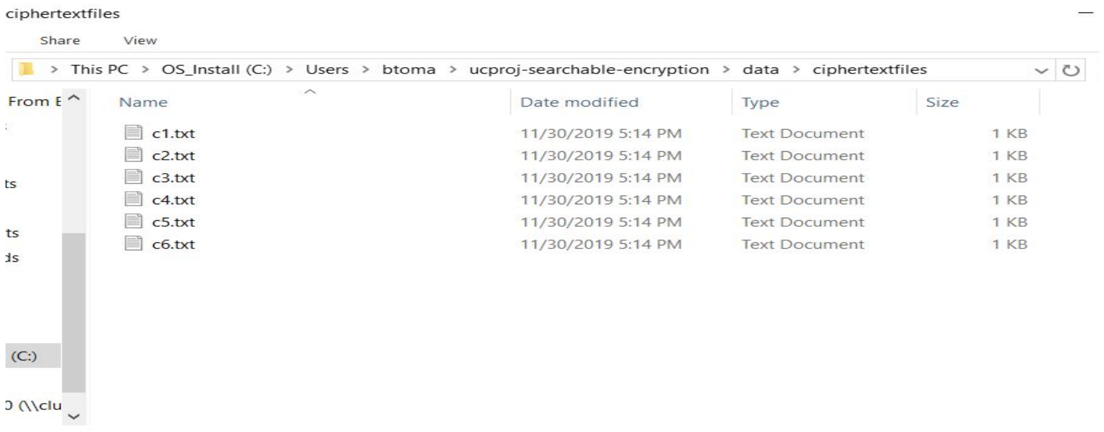
Encryption

The encryption process uses AES with the key generated in the previous step to encrypt all the files in the data/files directory and outputs the encrypted files into the data/ciphertextfiles directory. It also builds an encrypted index using the words found in the plaintext files. Each word is encrypted with a PRF and a corresponding entry is placed in the index. If a word is found in a plaintext file, the corresponding ciphertext file is placed under the word's entry in the index. The index is saved into data/index.txt.



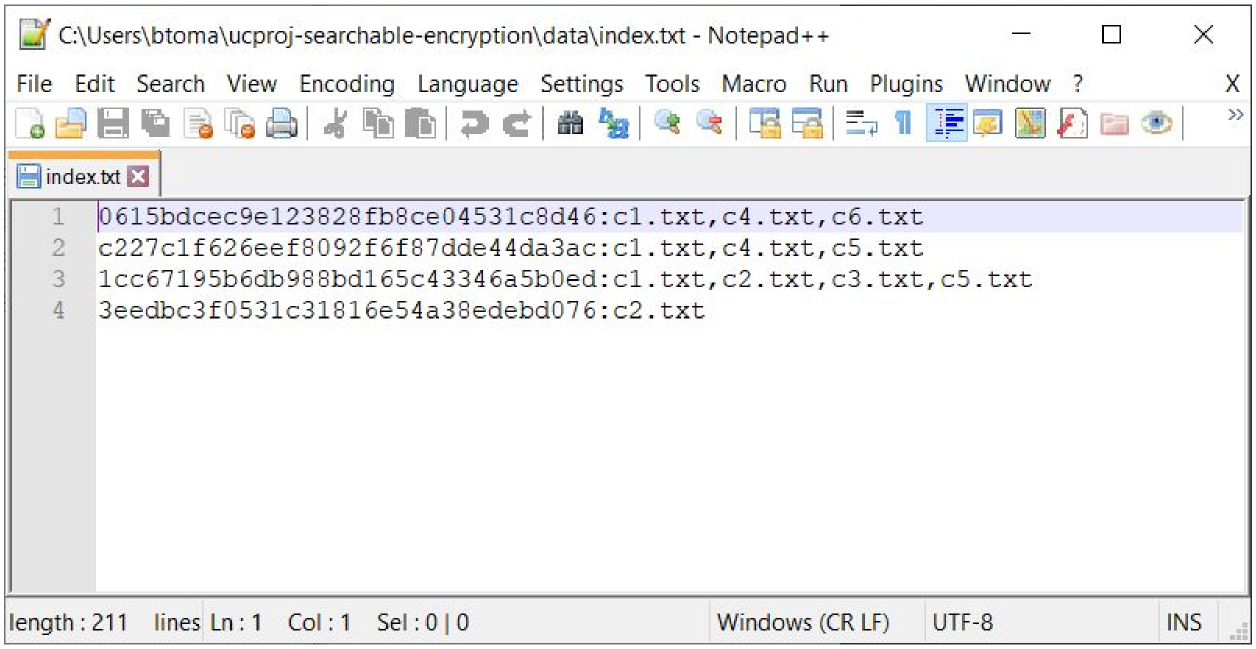
```
C:\Users\btoma\ucproj-searchable-encryption>python build/enc.py
Time to build index: 0.1710974

C:\Users\btoma\ucproj-searchable-encryption>
```

Figure 4: *Running the encryption process. The process of encrypting the files and generating the index took around 0.17 seconds in this case.*


ciphertextfiles
Share
View
This PC > OS_Install (C:) > Users > btoma > ucproj-searchable-encryption > data > ciphertextfiles

Name	Date modified	Type	Size
c1.txt	11/30/2019 5:14 PM	Text Document	1 KB
c2.txt	11/30/2019 5:14 PM	Text Document	1 KB
c3.txt	11/30/2019 5:14 PM	Text Document	1 KB
c4.txt	11/30/2019 5:14 PM	Text Document	1 KB
c5.txt	11/30/2019 5:14 PM	Text Document	1 KB
c6.txt	11/30/2019 5:14 PM	Text Document	1 KB

Figure 5: *The first output of the encryption process; these ciphertext files are generated from the corresponding plaintext files.*

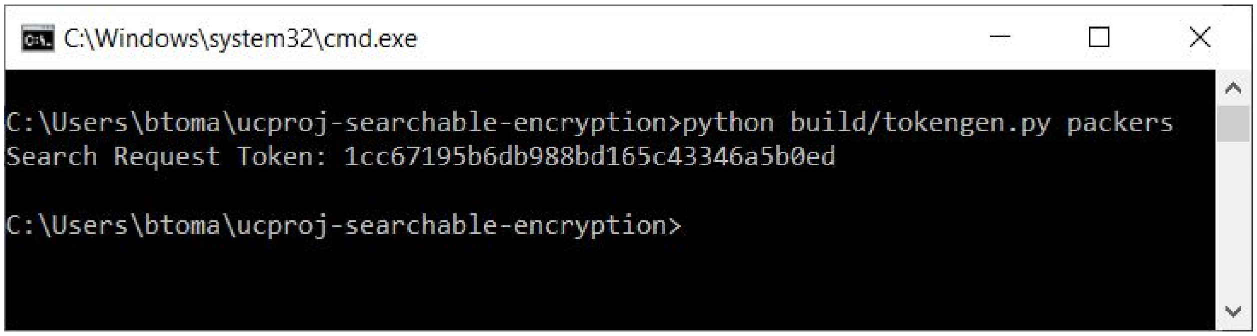

C:\Users\btoma\ucproj-searchable-encryption\data\index.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.txt
length : 211 lines Ln : 1 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

```
0615bdcec9e123828fb8ce04531c8d46:c1.txt,c4.txt,c6.txt
c227c1f626eef8092f6f87dde44da3ac:c1.txt,c4.txt,c5.txt
1cc67195b6db988bd165c43346a5b0ed:c1.txt,c2.txt,c3.txt,c5.txt
3eedbc3f0531c31816e54a38edebd076:c2.txt
```

Figure 6: *The second output of the encryption process, the index. Each line is an entry for a specific token, followed by the ciphertext files that contain that token.*

Token Generation

The token generation process accepts a keyword and encrypts it with the PRF to generate a token that can be used to search the index.

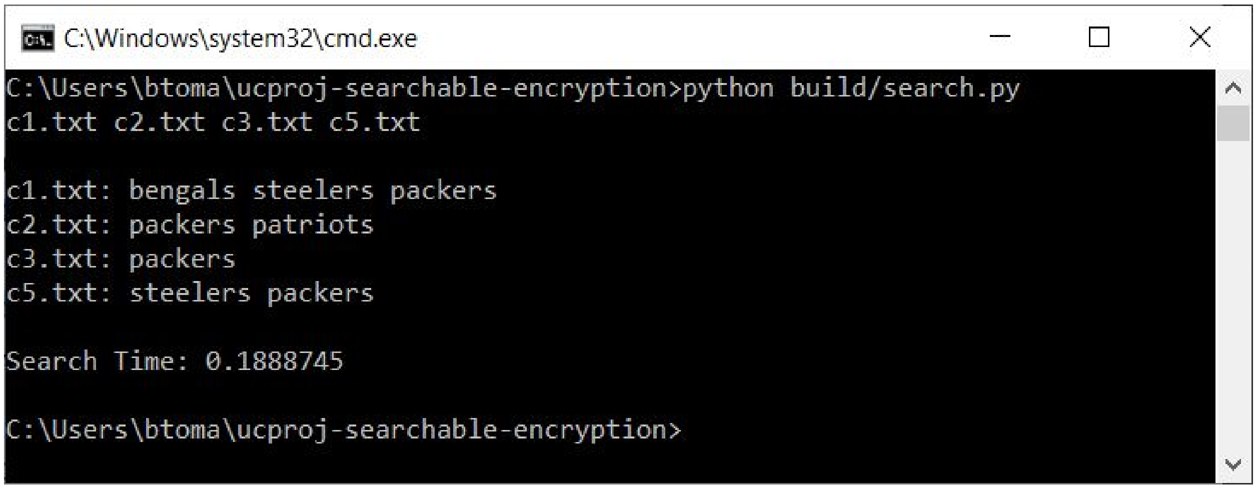


```
C:\Users\btoma\ucproj-searchable-encryption>python build/tokengen.py packers
Search Request Token: 1cc67195b6db988bd165c43346a5b0ed

C:\Users\btoma\ucproj-searchable-encryption>
```

Figure 7: *Generating a search token based on the search keyword "packers."*

Search

The search process uses the search token generated in the last step to search through the index and returns the ciphertext files that contain the search keyword.



```
C:\Users\btoma\ucproj-searchable-encryption>python build/search.py
c1.txt c2.txt c3.txt c5.txt

c1.txt: bengals steelers packers
c2.txt: packers patriots
c3.txt: packers
c5.txt: steelers packers

Search Time: 0.1888745

C:\Users\btoma\ucproj-searchable-encryption>
```

Figure 8: *Running the search process to search for the keyword "packers." Every ciphertext with a corresponding plaintext that contains the keyword is returned. The search process took about 0.19 seconds in this instance.*