VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

CRYPTOGRAPHY AND NETWORK SECURITY
AAT REPORT
on

Autokey Cipher

Submitted by

Sayed Ayman Bukhari (1BM18CS095)
Shaan Subbaiah B C (1BM18CS096)

Under the Guidance of
Prof. Lohith J J
Assistant Professor, BMSCE

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Mar-2021 to Jun-2021

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the AAT work entitled “**Autokey Cipher**" is carried out by **Sayed Ayman Bukhari (1BM18CS095), and Shaan Subbaiah B C (1BM18CS096)** who are bonafide students of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visveswaraya Technological University, Belgaum during the year 2021. The AAT report has been approved as it satisfies the academic requirements in respect of **Cryptography and Network Security (20CS6PCCNS)** work prescribed for the said degree.

Signature of the Guide	Signature of the HOD
Prof. Lohith J J	Dr. Umadevi V
Assistant Professor	Associate Prof. & Head, Dept. of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

B. M. S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, Sayed Ayman Bukhari (1BM18CS095) and Shaan Subbaiah B C (1BM18CS096), students of 6th Semester, B.E, Department of Computer Science and Engineering, B. M. S. College of Engineering, Bangalore, hereby declare that, this AAT entitled "Autokey Cipher" has been carried out by us under the guidance of Prof. Lohith J J, Assistant Professor, Department of CSE, B. M. S. College of Engineering, Bangalore during the academic semester Mar-2021-Jun-2021

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

Signature

Sayed Ayman Bukhari (1BM18CS095)

Shaan Subbaiah B C   (1BM18CS096)

Chapter 1

Introduction

Autokey Cipher

Autokey Cipher also known as Autoclave Cipher is a polyalphabetic substitution cipher that incorporates the message (PlainText) into the key. The key is generated from the message in an automated fashion.

There are two forms of autokey ciphers: key-autokey and text-autokey ciphers. A key-autokey cipher uses previous members of the keystream to determine the next element in the keystream. A text-autokey uses the previous message text to determine the next element in the keystream. In modern cryptography, self-synchronizing stream ciphers are autokey ciphers.

Attacks

For Autokey cipher frequency analysis methods such as Kasiski examination or index of coincidence analysis will not work on the Ciphertext, unlike for similar ciphers that use a single repeated key.

The key can be attacked by using a dictionary of common words and n-grams by attempting the decryption of the message by moving that word through the key until potentially-readable text appears.

Dictionary attack - In cryptanalysis and computer security, a dictionary attack is a form of brute force attack technique for defeating a cipher or authentication mechanism by trying to determine its decryption key or keyphrase by trying thousands or millions of likely possibilities, such as words in a dictionary or previously used words.

Motivation

It is closely related to the Vigenere cipher but uses a different method of generating the key. It was invented by Blaise de Vigenere in 1586 and is a much more secure way of generating the keystream than the Vigenere Cipher, which is amazing since for over 200 years it was believed that the Vigenere was unbreakable. The weakness of the Vigenere Cipher was the repeating nature of the keystream, which allowed us to work out the length of the keyword and thus perform frequency analysis on the different parts.

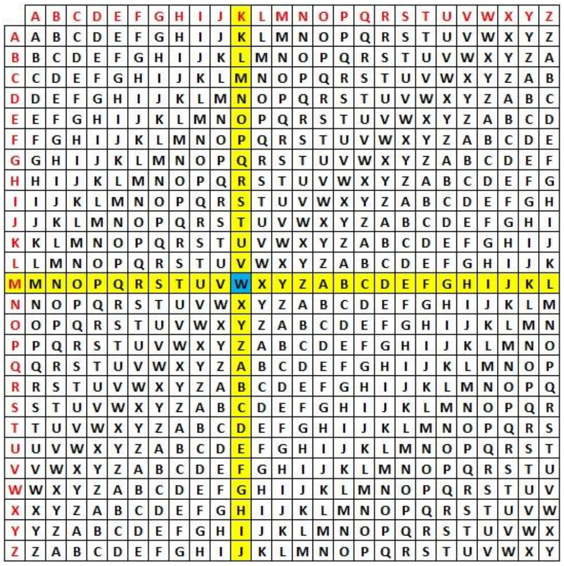
Various aspects of the algorithm chosen

Encryption using the Autokey Cipher is very similar to the Vigenere Cipher, except in the creation of the keystream.

The keystream is made by starting with the keyword or keyphrase, and then adding the plaintext itself to the end of the keyword.

We then use a Tabula Recta to find the PlainText letter across the top, and the keystream letter down the left, and use the crossover letter as the ciphertext letter.

PlainText = K,   Key = M,   CipherText = W



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Fig 1 - Autokey Ciphering using Tabula Recta.

Chapter 2
Methodology

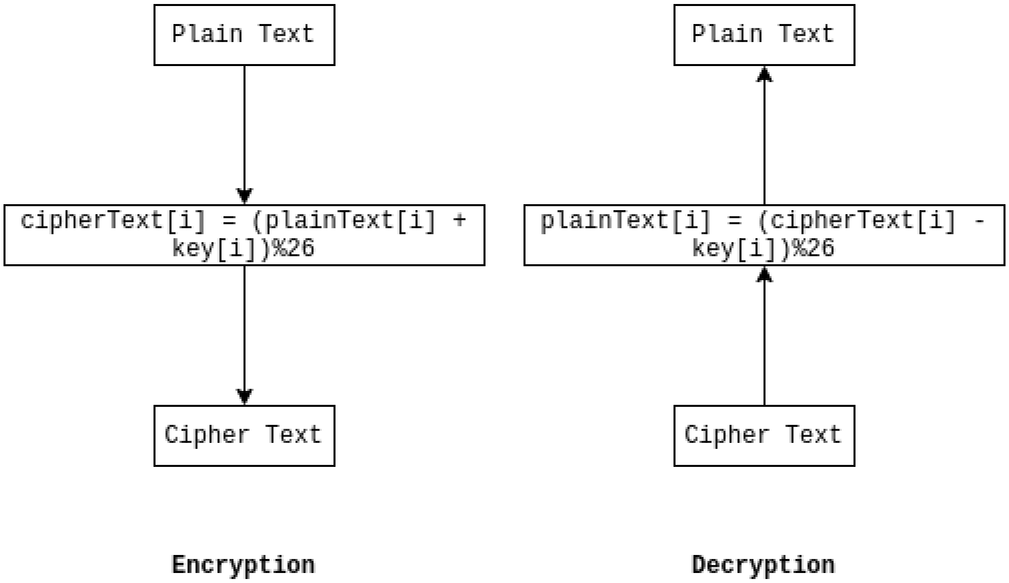

Plain Text
cipherText[i] = (plainText[i] + key[i])%26
Cipher Text
Encryption
Plain Text
plainText[i] = (cipherText[i] - key[i])%26
Cipher Text
Decryption

Fig 2 - Autokey encryption and decryption flowchart

Encryption: $C_i = (K_i - P_i) \bmod 26$

Decryption: $P_i = (K_i + C_i) \bmod 26$

The above depicts the workflow of the Autokey algorithm. Autokey cipher is a polyalphabetic substitution cipher and a variant of Vigenère cipher. The Autokey cipher encodes the Key & Plaintext to gives us the keystream which we use to encrypt the Plaintext into the Ciphertext

Encryption :

Step 1: The keystream is made by starting with the keyword or keyphrase, and then appending to the end of this the plaintext itself.

Example :

Plaintext : *DUSK*

Keyword: *HI*

Keystream: *HIDU*

Step 2: We then use a Tabula Recta to find the keystream letter across the top, and the plaintext letter down the left, and use the crossover letter as the ciphertext letter.

Plaintext	D	U	S	K
Keystream	H	I		
Ciphertext				

Plaintext	D	U	S	K
Keystream	H	I	D	U
Ciphertext				

Plaintext	D	U	S	K
Keystream	H	I	D	U
Ciphertext	K	C	V	E

Decryption :

Step 1: To decrypt a ciphertext using the Autokey Cipher, we start just as we did for the Vigenere Cipher, and find the first letter of the key across the top

Step 2: Find the ciphertext letter down that column, and take the plaintext letter at the far left of this row

Step 3: Continuing to decode each letter, we add them to the end of the keystream each time

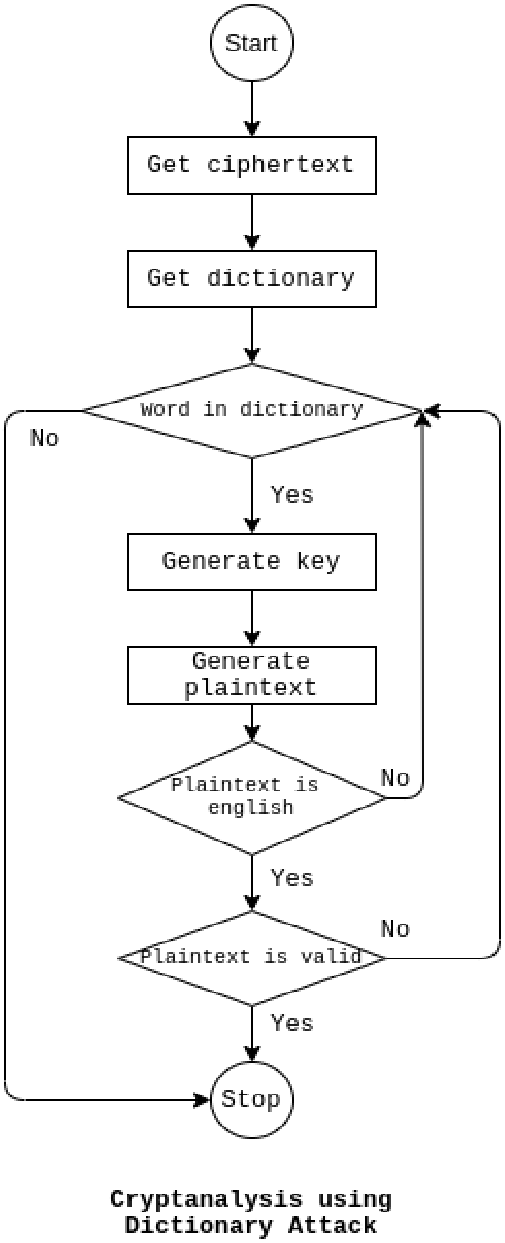
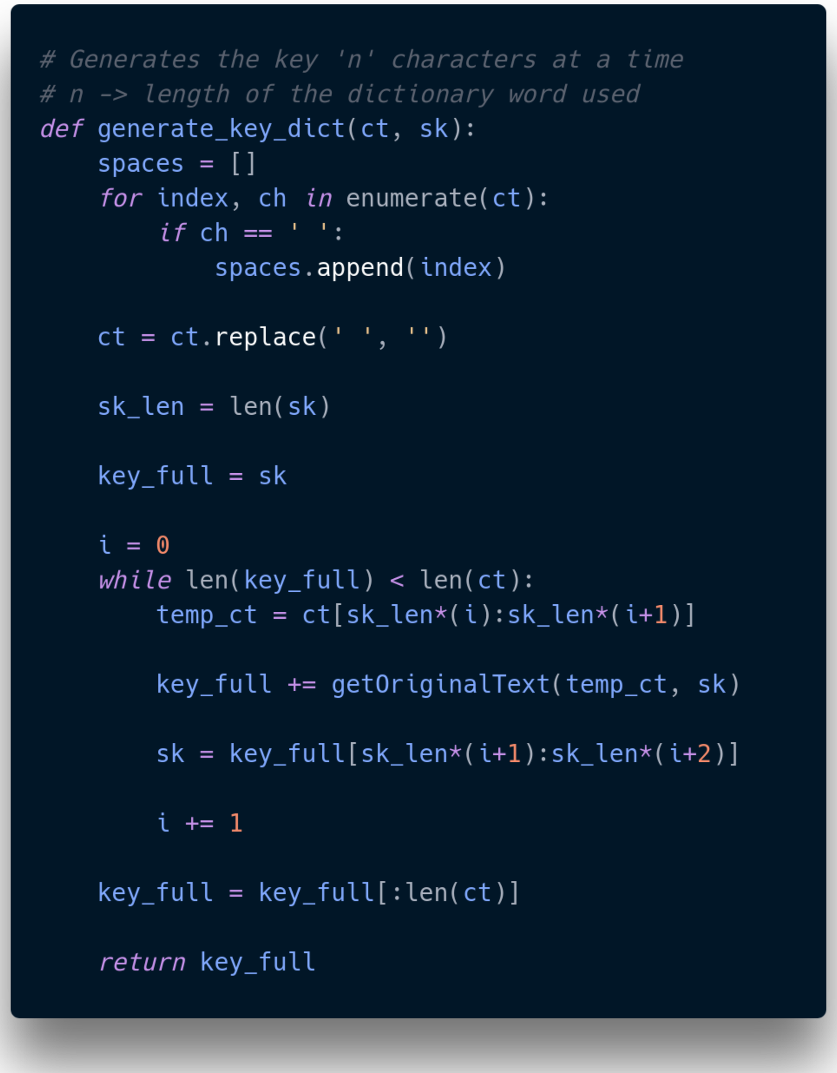


Fig 3 - Dictionary method cryptanalysis flowchart

The above depicts the workflow of dictionary attacks. It is a form of brute force attack.

- Ciphertext is taken as an input.
- The input is run against a list of dictionary words in order to detect the key used while encryption.
- If any one of the words in the dictionary decrypts the ciphertext by the said percentage (say, 70%), the plaintext is displayed.
- If it makes a legitimate sentence,we exit the loop else it loops back to run it against the rest of the words, until the message gets decrypted.



```python
# Generates the key 'n' characters at a time
# n -> length of the dictionary word used
def generate_key_dict(ct, sk):
    spaces = []
    for index, ch in enumerate(ct):
        if ch == ' ':
            spaces.append(index)

    ct = ct.replace(' ', '')

    sk_len = len(sk)

    key_full = sk

    i = 0
    while len(key_full) < len(ct):
        temp_ct = ct[sk_len*(i):sk_len*(i+1)]

        key_full += getOriginalText(temp_ct, sk)

        sk = key_full[sk_len*(i+1):sk_len*(i+2)]

        i += 1

    key_full = key_full[:len(ct)]

    return key_full
```

Fig 4 - Snippet of cryptanalysis key generation function

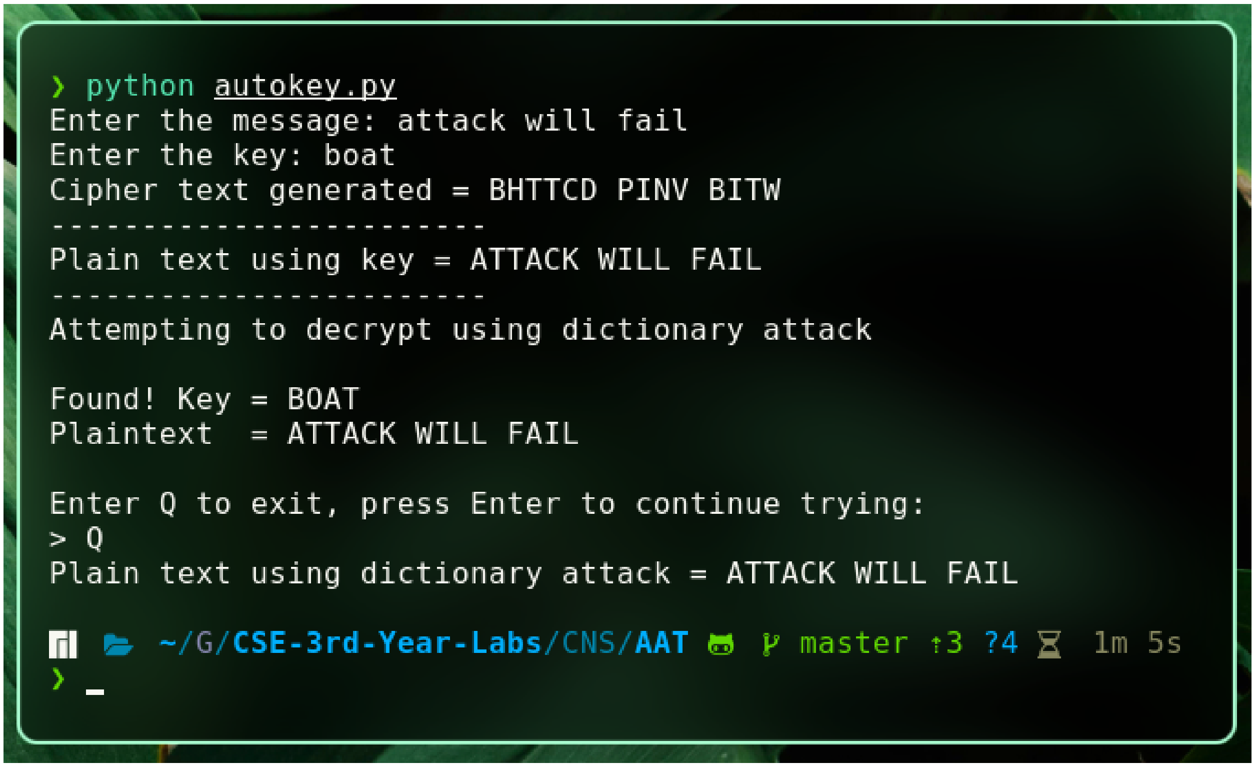
Cryptanalysis key generation function working:

- The dictionary is iterated and each word is used as a short-key.
- The full-key is assigned to the short-key.
- While the length of the full key is less than the length of the ciphertext to be decrypted, the following steps are repeated:
    - n is the length of the short-key.
    - The temporary-ciphertext is assigned to the next n characters of the ciphertext.
    - The full key is concatenated with the n length key obtained when decrypting the temporary-ciphertext with the short-key.
    - The short-key is assigned to the next n letters of the full-key.
- The full-key obtained is sliced to the length of the ciphertext.

Chapter 3

Results and Discussion

Screenshots of the results.



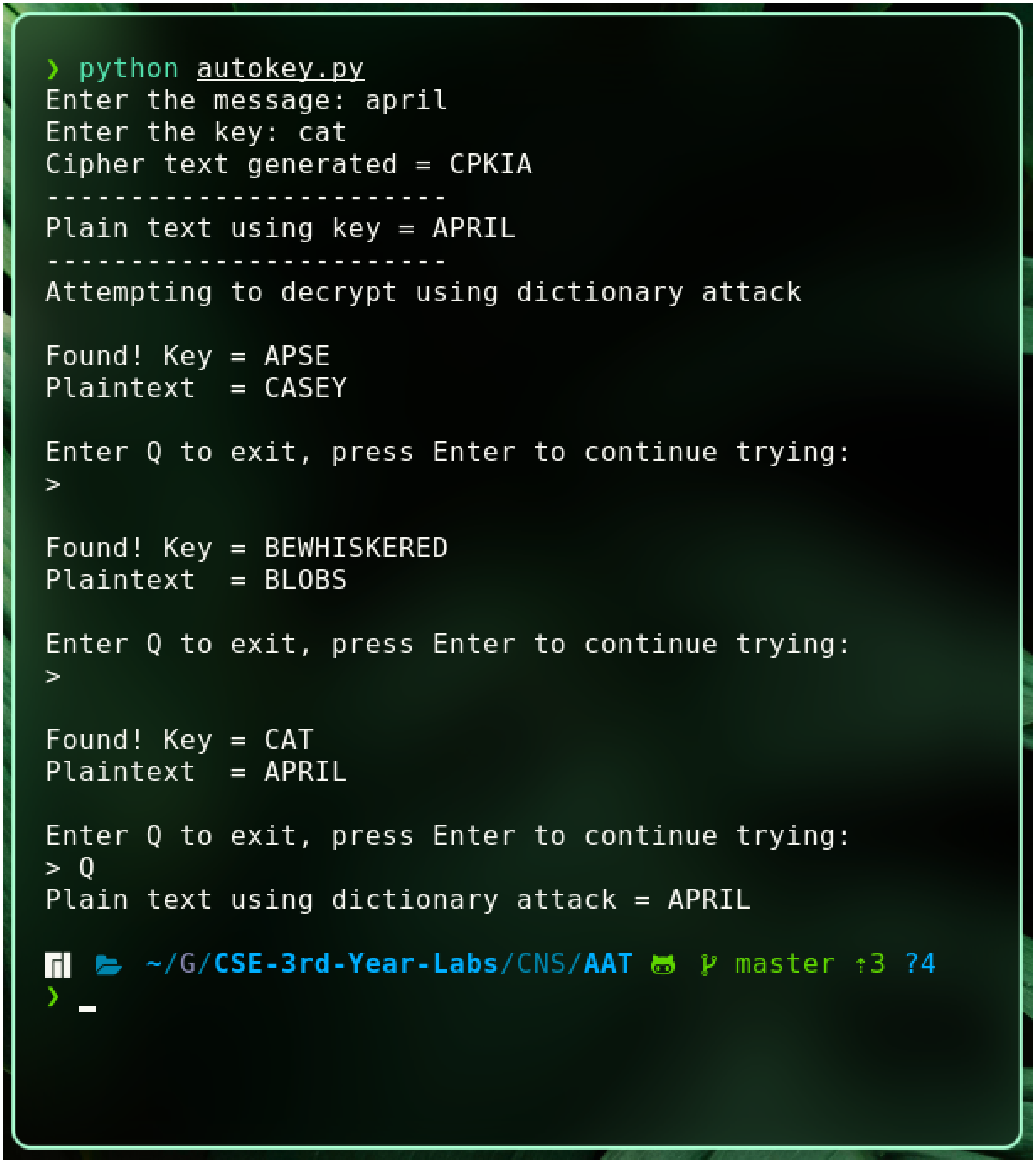
```
❯ python autokey.py
Enter the message: attack will fail
Enter the key: boat
Cipher text generated = BHTTCD PINV BITW
------------------------
Plain text using key = ATTACK WILL FAIL
------------------------
Attempting to decrypt using dictionary attack

Found! Key = BOAT
Plaintext  = ATTACK WILL FAIL

Enter Q to exit, press Enter to continue trying:
> Q
Plain text using dictionary attack = ATTACK WILL FAIL

~/G/CSE-3rd-Year-Labs/CNS/AAT  master ↑3 ?4  1m 5s
❯ _
```

Fig 5 - Message 'Attack will fail' with key 'boat'



```
❯ python autokey.py
Enter the message: april
Enter the key: cat
Cipher text generated = CPKIA
------------------------
Plain text using key = APRIL
------------------------
Attempting to decrypt using dictionary attack

Found! Key = APSE
Plaintext  = CASEY

Enter Q to exit, press Enter to continue trying:
>

Found! Key = BEWHISKERED
Plaintext  = BLOBS

Enter Q to exit, press Enter to continue trying:
>

Found! Key = CAT
Plaintext  = APRIL

Enter Q to exit, press Enter to continue trying:
> Q
Plain text using dictionary attack = APRIL

~/G/CSE-3rd-Year-Labs/CNS/AAT  master ↑3 ?4
❯ _
```

Fig 6 - Message ‘April’ with key ‘cat’

Chapter 4

Conclusion and Future Work

Autokey Cipher which is a variation of the Vignere Cipher, is a cryptographic approach. It is a polyalphabetic cipher and is quite secure. The cipher is a relatively old yet widely used cipher. The Cipher incorporates the message (PlainText) into the keystream after the Key.

Autokey cipher is more secure than any other poly-alphabetic ciphers that use defined keys since the key does not repeat again in cipher text/message. So, some frequency analysis methods like Kasiski examination/index of coincidence analysis will not work on this type of ciphertext, except for similar ciphers that use a single repeated key.

A main weakness of the system is that the plaintext is also a part of the key. which means that the key will likely contain common words at multiple/various positions. The key can be attacked by using a dictionary of common words, bigrams, trigrams(any brute force), etc, by attempting the decryption of the message by moving that word through the key until the actual readable text appears.

Cryptographers can work on countering the Brute force attacks such as the dictionary attack by improving the appending of the plaintext(message) and the key into the keystream to make it more undetectable and are hence developing more enhanced enciphering techniques that are immune to such attacks.

References

1. ‘Autokey Cipher’ (2021) Wikipedia. Available at: https://en.wikipedia.org/wiki/Autokey_cipher (Accessed: 02 June 2021).

2. ‘Detecting English Programmatically’ Available at: http://inventwithpython.com/hacking/chapter12.html