1

In [27]:
```
import matplotlib.pyplot as plt
import itertools
import string
from tqdm import tqdm
```

In [28]:
```
cipher_text = "PRCSOFQX FP QDR AFOPQ CZSPR LA JFPALOQSKR. QDFP FP ZK LIU BROJZK MOL
```

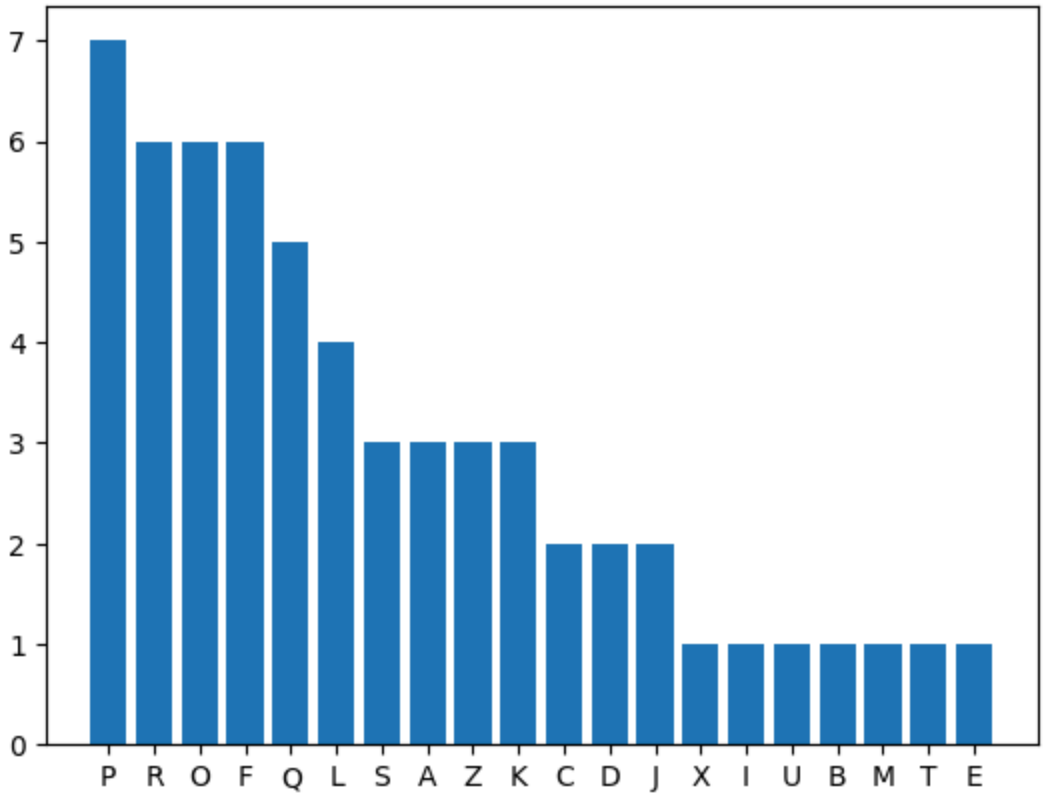
In [29]:
```
freq = {}
for c in cipher_text:
    if c.isalpha():
        freq[c] = freq.get(c, 0) + 1
```

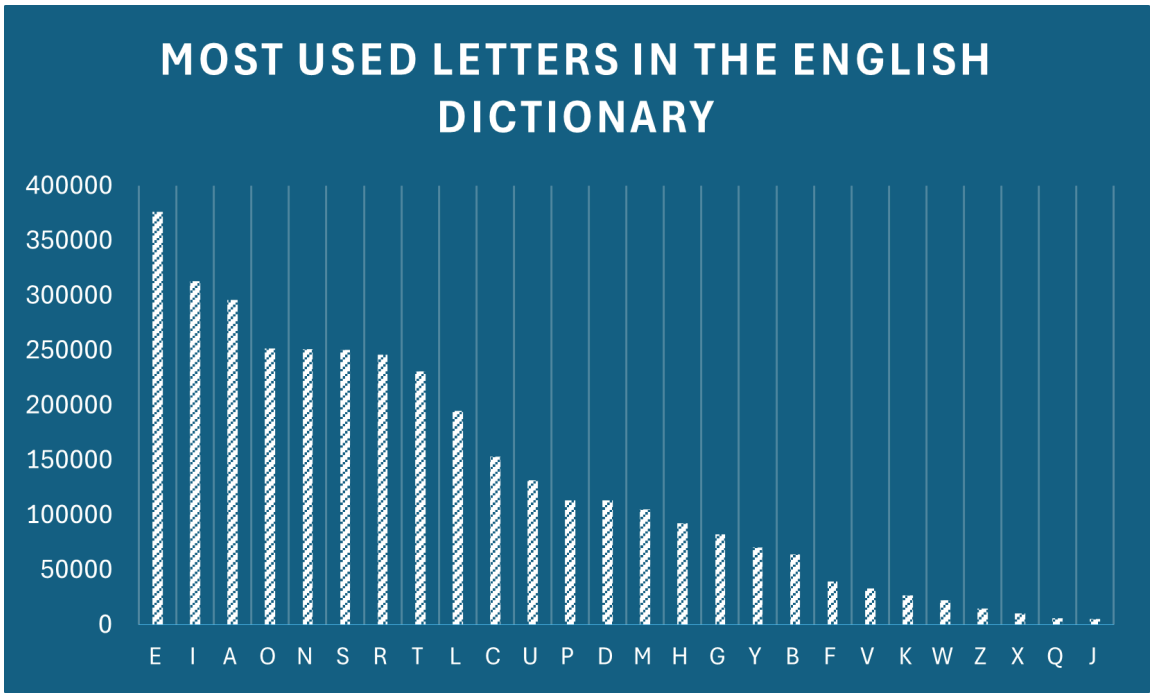
In [30]:
```
sorted_items = sorted(freq.items(), key=lambda x: x[1], reverse=True)
print(sorted_items)

letters_sorted = [item[0] for item in sorted_items]
counts_sorted = [item[1] for item in sorted_items]

bars = plt.bar(letters_sorted, counts_sorted, color='C0')
plt.show()
```

```
[('P', 7), ('R', 6), ('O', 6), ('F', 6), ('Q', 5), ('L', 4), ('S', 3), ('A', 3),
('Z', 3), ('K', 3), ('C', 2), ('D', 2), ('J', 2), ('X', 1), ('I', 1), ('U', 1),
('B', 1), ('M', 1), ('T', 1), ('E', 1)]
```





a. ANS: P, R, O

In [31]:
```
#FP -> IS, QDR -> THE
sub_dic1 = {"P":"S","F":"I","Q":"T","D":"H","R":"E" }
print("encoded text : "+cipher_text)
def decode(cipher_text, dic):
    decoded_text = []
    isSubstituted = []
    for c in cipher_text:
        if c in dic:
            decoded_text.append(dic[c])
            isSubstituted.append("1")
        else:
            decoded_text.append(c)
            isSubstituted.append("0")
    print("decoded text : "+"".join(decoded_text))
    return decoded_text, isSubstituted
decoded_text, isSubstitute = decode(cipher_text, sub_dic1)
print(decoded_text)
print(isSubstitute)
```

```
encoded text : PRCSOFQX FP QDR AFOPQ CZSPR LA JFPALOQSKR. QDFP FP ZK LIU BROJZK MOLT
ROE
decoded text : SECSOITX IS THE AIOST CZSSE LA JISALOTSKE. THIS IS ZK LIU BEOJZK MOLT
EOE
['S', 'E', 'C', 'S', 'O', 'I', 'T', 'X', ' ', 'I', 'S', ' ', 'T', 'H', 'E', ' ',
'A', 'I', 'O', 'S', 'T', ' ', 'C', 'Z', 'S', 'S', 'E', ' ', 'L', 'A', ' ', 'J', 'I',
'S', 'A', 'L', 'O', 'T', 'S', 'K', 'E', '.', ' ', 'T', 'H', 'I', 'S', ' ', 'I', 'S',
' ', 'Z', 'K', ' ', 'L', 'I', 'U', ' ', 'B', 'E', 'O', 'J', 'Z', 'K', ' ', 'M', 'O',
'L', 'T', 'E', 'O', 'E']
['1', '1', '0', '0', '0', '1', '1', '0', '0', '1', '1', '0', '1', '1', '1', '0',
'0', '1', '0', '1', '1', '0', '0', '0', '0', '1', '1', '0', '0', '0', '0', '0', '1',
'1', '0', '0', '0', '1', '0', '0', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
'0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0', '0',
'0', '0', '1', '0', '0']
```

In [32]:
```
#FP -> IS, QDR -> THE
#ZK -> AN
sub_dic2 = {"P":"S","F":"I","Q":"T","D":"H","R":"E","Z":"A","K":"N"}
decoded_text, isSubstitute = decode(cipher_text, sub_dic2)
print(decoded_text)
print(isSubstitute)
```

```
decoded text : SECSOITX IS THE AIOST CASSE LA JISALOTSNE. THIS IS AN LIU BEOJAN MOLT
EOE
['S', 'E', 'C', 'S', 'O', 'I', 'T', 'X', ' ', 'I', 'S', ' ', 'T', 'H', 'E', ' ',
'A', 'I', 'O', 'S', 'T', ' ', 'C', 'A', 'S', 'S', 'E', ' ', 'L', 'A', ' ', 'J', 'I',
'S', 'A', 'L', 'O', 'T', 'S', 'N', 'E', '.', ' ', 'T', 'H', 'I', 'S', ' ', 'I', 'S',
' ', 'A', 'N', ' ', 'L', 'I', 'U', ' ', 'B', 'E', 'O', 'J', 'A', 'N', ' ', 'M', 'O',
'L', 'T', 'E', 'O', 'E']
['1', '1', '0', '0', '0', '1', '1', '0', '0', '1', '1', '0', '1', '1', '1', '0',
'0', '1', '0', '1', '1', '0', '0', '1', '0', '1', '1', '0', '0', '0', '0', '0', '1',
'1', '0', '0', '0', '1', '0', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
'0', '1', '1', '0', '0', '0', '0', '0', '0', '1', '0', '0', '1', '1', '0', '0', '0',
'0', '0', '1', '0', '0']
```

In [33]:
```
#FP -> IS, QDR -> THE
#ZK -> AN
#LA -> IN/OF
sub_dic3 = {"P":"S","F":"I","Q":"T","D":"H","R":"E","Z":"A","K":"N", "L":"O","A":"F
decoded_text, isSubstitute = decode(cipher_text, sub_dic3)
print(decoded_text)
print(isSubstitute)
```

```
decoded text : SECSOITX IS THE FIOST CASSE OF JISFOOTSNE. THIS IS AN OIU BEOJAN MOOT
EOE
['S', 'E', 'C', 'S', 'O', 'I', 'T', 'X', ' ', 'I', 'S', ' ', 'T', 'H', 'E', ' ',
'F', 'I', 'O', 'S', 'T', ' ', 'C', 'A', 'S', 'S', 'E', ' ', 'O', 'F', ' ', 'J', 'I',
'S', 'F', 'O', 'O', 'T', 'S', 'N', 'E', '.', ' ', 'T', 'H', 'I', 'S', ' ', 'I', 'S',
' ', 'A', 'N', ' ', 'O', 'I', 'U', ' ', 'B', 'E', 'O', 'J', 'A', 'N', ' ', 'M', 'O',
'O', 'T', 'E', 'O', 'E']
['1', '1', '0', '0', '0', '1', '1', '0', '0', '1', '1', '0', '1', '1', '1', '0',
'1', '1', '0', '1', '1', '0', '0', '1', '0', '1', '1', '0', '1', '1', '0', '0', '1',
'1', '1', '1', '0', '1', '0', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
'0', '1', '1', '0', '1', '0', '0', '0', '0', '1', '0', '0', '1', '1', '0', '0', '0',
'1', '0', '1', '0', '0']
```

In [34]:
```
#FP -> IS, QDR -> THE
#ZK -> AN
#LA -> IN/OF
#X->Y, O->R (guess)
sub_dic4 = {"P":"S","F":"I","Q":"T","D":"H","R":"E","Z":"A","K":"N", "L":"O","A":"F
decoded_text, isSubstitute = decode(cipher_text, sub_dic4)
print(decoded_text)
print(isSubstitute)
```

```
decoded text : SECSRITY IS THE FIRST CASSE OF JISFORTSNE. THIS IS AN OIU BERJAN MROT
ERE
['S', 'E', 'C', 'S', 'R', 'I', 'T', 'Y', ' ', 'I', 'S', ' ', 'T', 'H', 'E', ' ',
'F', 'I', 'R', 'S', 'T', ' ', 'C', 'A', 'S', 'S', 'E', ' ', 'O', 'F', ' ', 'J', 'I',
'S', 'F', 'O', 'R', 'T', 'S', 'N', 'E', '.', ' ', 'T', 'H', 'I', 'S', ' ', 'I', 'S',
' ', 'A', 'N', ' ', 'O', 'I', 'U', ' ', 'B', 'E', 'R', 'J', 'A', 'N', ' ', 'M', 'R',
'O', 'T', 'E', 'R', 'E']
['1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1', '1', '0',
'1', '1', '1', '1', '1', '0', '0', '1', '0', '1', '1', '0', '1', '1', '0', '0', '1',
'1', '1', '1', '1', '1', '0', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
'0', '1', '1', '0', '1', '0', '0', '0', '0', '1', '1', '0', '1', '1', '0', '0', '1',
'1', '0', '1', '1', '0']
```

In [35]:
```
#FP -> IS, QDR -> THE
#ZK -> AN
#LA -> IN/OF
#X->Y, O->R (guess)
# first word is SECURITY : C->C,S->U
sub_dic5 = {"P":"S","F":"I","Q":"T","D":"H","R":"E","Z":"A","K":"N", "L":"O","A":"F
decoded_text, isSubstitute = decode(cipher_text, sub_dic5)
print(decoded_text)
print(isSubstitute)
```

```
decoded text : SECURITY IS THE FIRST CAUSE OF JISFORTUNE. THIS IS AN OIU BERJAN MROT
ERE
['S', 'E', 'C', 'U', 'R', 'I', 'T', 'Y', ' ', 'I', 'S', ' ', 'T', 'H', 'E', ' ',
'F', 'I', 'R', 'S', 'T', ' ', 'C', 'A', 'U', 'S', 'E', ' ', 'O', 'F', ' ', 'J', 'I',
'S', 'F', 'O', 'R', 'T', 'U', 'N', 'E', '.', ' ', 'T', 'H', 'I', 'S', ' ', 'I', 'S',
' ', 'A', 'N', ' ', 'O', 'I', 'U', ' ', 'B', 'E', 'R', 'J', 'A', 'N', ' ', 'M', 'R',
'O', 'T', 'E', 'R', 'E']
['1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1', '1', '0',
'1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '0', '1',
'1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
'0', '1', '1', '0', '1', '0', '0', '0', '0', '1', '1', '0', '1', '1', '0', '0', '1',
'1', '0', '1', '1', '0']
```

In [36]:
```
#FP -> IS, QDR -> THE
#ZK -> AN
#LA -> IN/OF
#X->Y, O->R (guess)
# first word is SECURITY : C->C,S->U
# misfortune : J->M
sub_dic6 = {"P":"S","F":"I","Q":"T","D":"H","R":"E","Z":"A","K":"N", "L":"O","A":"F
decoded_text, isSubstitute = decode(cipher_text, sub_dic6)
print(decoded_text)
print(isSubstitute)
```

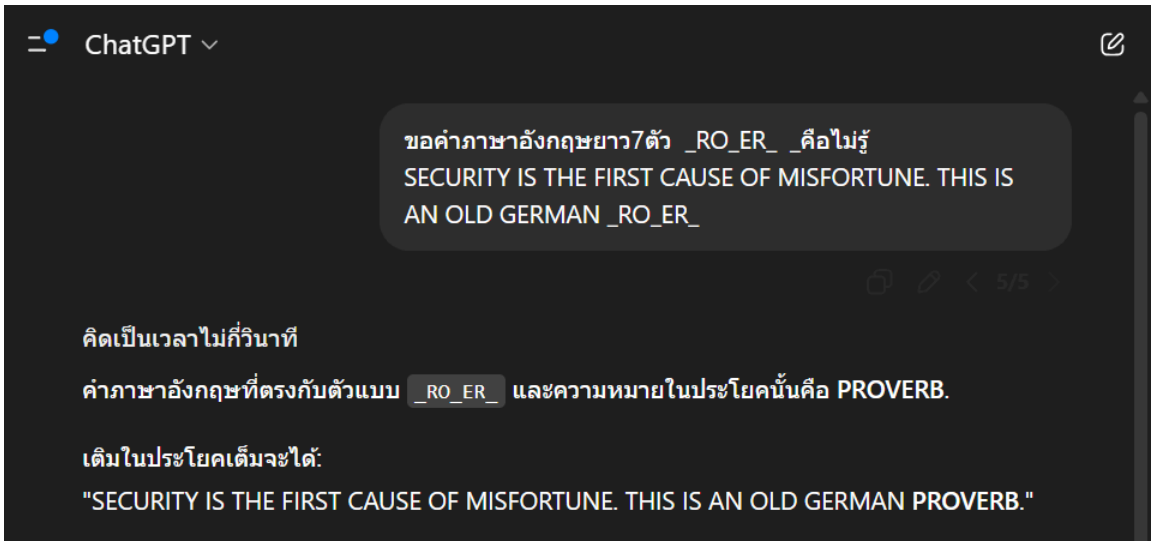
```
decoded text : SECURITY IS THE FIRST CAUSE OF MISFORTUNE. THIS IS AN OIU BERMAN MROT
ERE
['S', 'E', 'C', 'U', 'R', 'I', 'T', 'Y', ' ', 'I', 'S', ' ', 'T', 'H', 'E', ' ',
'F', 'I', 'R', 'S', 'T', ' ', 'C', 'A', 'U', 'S', 'E', ' ', 'O', 'F', ' ', 'M', 'I',
'S', 'F', 'O', 'R', 'T', 'U', 'N', 'E', '.', ' ', 'T', 'H', 'I', 'S', ' ', 'I', 'S',
' ', 'A', 'N', ' ', 'O', 'I', 'U', ' ', 'B', 'E', 'R', 'M', 'A', 'N', ' ', 'M', 'R',
'O', 'T', 'E', 'R', 'E']
['1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1', '1', '0',
'1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1',
'1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
'0', '1', '1', '0', '1', '0', '0', '0', '0', '1', '1', '1', '1', '1', '0', '0', '1',
'1', '0', '1', '1', '0']
```

In [37]:
```
#FP -> IS, QDR -> THE
#ZK -> AN
#LA -> IN/OF
#X->Y, O->R (guess)
# first word is SECURITY : C->C,S->U
# misfortune : J->M
# german -> B->G
sub_dic7 = {"P":"S","F":"I","Q":"T","D":"H","R":"E","Z":"A","K":"N", "L":"O","A":"F
decoded_text, isSubstitute = decode(cipher_text, sub_dic7)
print(decoded_text)
print(isSubstitute)
```

```
decoded text : SECURITY IS THE FIRST CAUSE OF MISFORTUNE. THIS IS AN OIU GERMAN MROT
ERE
['S', 'E', 'C', 'U', 'R', 'I', 'T', 'Y', ' ', 'I', 'S', ' ', 'T', 'H', 'E', ' ',
'F', 'I', 'R', 'S', 'T', ' ', 'C', 'A', 'U', 'S', 'E', ' ', 'O', 'F', ' ', 'M', 'I',
'S', 'F', 'O', 'R', 'T', 'U', 'N', 'E', '.', ' ', 'T', 'H', 'I', 'S', ' ', 'I', 'S',
' ', 'A', 'N', ' ', 'O', 'I', 'U', ' ', 'G', 'E', 'R', 'M', 'A', 'N', ' ', 'M', 'R',
'O', 'T', 'E', 'R', 'E']
['1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1', '1', '0',
'1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1',
'1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
'0', '1', '1', '0', '1', '0', '0', '0', '1', '1', '1', '1', '1', '1', '0', '0', '1',
'1', '0', '1', '1', '0']
```

In [38]:
```
#FP -> IS, QDR -> THE
#ZK -> AN
#LA -> IN/OF
#X->Y, O->R (guess)
# first word is SECURITY : C->C,S->U
# misfortune : J->M
# german -> B->G
# old(must be an adj.) : I->L, U->D
sub_dic8 = {"P":"S","F":"I","Q":"T","D":"H","R":"E","Z":"A","K":"N", "L":"O","A":"F
decoded_text, isSubstitute = decode(cipher_text, sub_dic8)
print(decoded_text)
print(isSubstitute)
```

```
decoded text : SECURITY IS THE FIRST CAUSE OF MISFORTUNE. THIS IS AN OLD GERMAN MROT
ERE
['S', 'E', 'C', 'U', 'R', 'I', 'T', 'Y', ' ', 'I', 'S', ' ', 'T', 'H', 'E', ' ',
'F', 'I', 'R', 'S', 'T', ' ', 'C', 'A', 'U', 'S', 'E', ' ', 'O', 'F', ' ', 'M', 'I',
'S', 'F', 'O', 'R', 'T', 'U', 'N', 'E', '.', ' ', 'T', 'H', 'I', 'S', ' ', 'I', 'S',
' ', 'A', 'N', ' ', 'O', 'L', 'D', ' ', 'G', 'E', 'R', 'M', 'A', 'N', ' ', 'M', 'R',
'O', 'T', 'E', 'R', 'E']
['1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1', '1', '0',
'1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1',
'1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
'0', '1', '1', '0', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '0', '1',
'1', '0', '1', '1', '0']
```



In [39]:
```python
#FP -> IS, QDR -> THE
#ZK -> AN
#LA -> IN/OF
#X->Y, O->R (guess)
# first word is SECURITY : C->C,S->U
# misfortune : J->M
# german -> B->G
# old(must be an adj.) : I->L, U->D
# GPT : M->P, T->V, E->B
sub_dic9 = {"P":"S","F":"I","Q":"T","D":"H","R":"E","Z":"A","K":"N", "L":"O","A":"F
decoded_text, isSubstitute = decode(cipher_text, sub_dic9)
print(decoded_text)
print(isSubstitute)
```

```
decoded text : SECURITY IS THE FIRST CAUSE OF MISFORTUNE. THIS IS AN OLD GERMAN PROV
ERB
['S', 'E', 'C', 'U', 'R', 'I', 'T', 'Y', ' ', 'I', 'S', ' ', 'T', 'H', 'E', ' ',
'F', 'I', 'R', 'S', 'T', ' ', 'C', 'A', 'U', 'S', 'E', ' ', 'O', 'F', ' ', 'M', 'I',
'S', 'F', 'O', 'R', 'T', 'U', 'N', 'E', '.', ' ', 'T', 'H', 'I', 'S', ' ', 'I', 'S',
' ', 'A', 'N', ' ', 'O', 'L', 'D', ' ', 'G', 'E', 'R', 'M', 'A', 'N', ' ', 'P', 'R',
'O', 'V', 'E', 'R', 'B']
['1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1', '1', '0',
'1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '1',
'1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
'0', '1', '1', '0', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '1', '1',
'1', '1', '1', '1', '1']
```

QUESTION

a: P, R, O

b: is, in, of, the, an

c: SECURITY IS THE FIRST CAUSE OF MISFORTUNE. THIS IS AN OLD GERMAN PROVERB

d: this take me about 1 hour

2

Vigenère **เป็นการเข้ารหัสแบบ** polyalphabetic substitution **ใช้ คีย์หลายตัว หมุนซ้ำ**

Kasiski Examination **ใช้ การหาลวดลายซ้ำของ** ciphertext **เพื่อคาดเดาความยาวของคีย์**

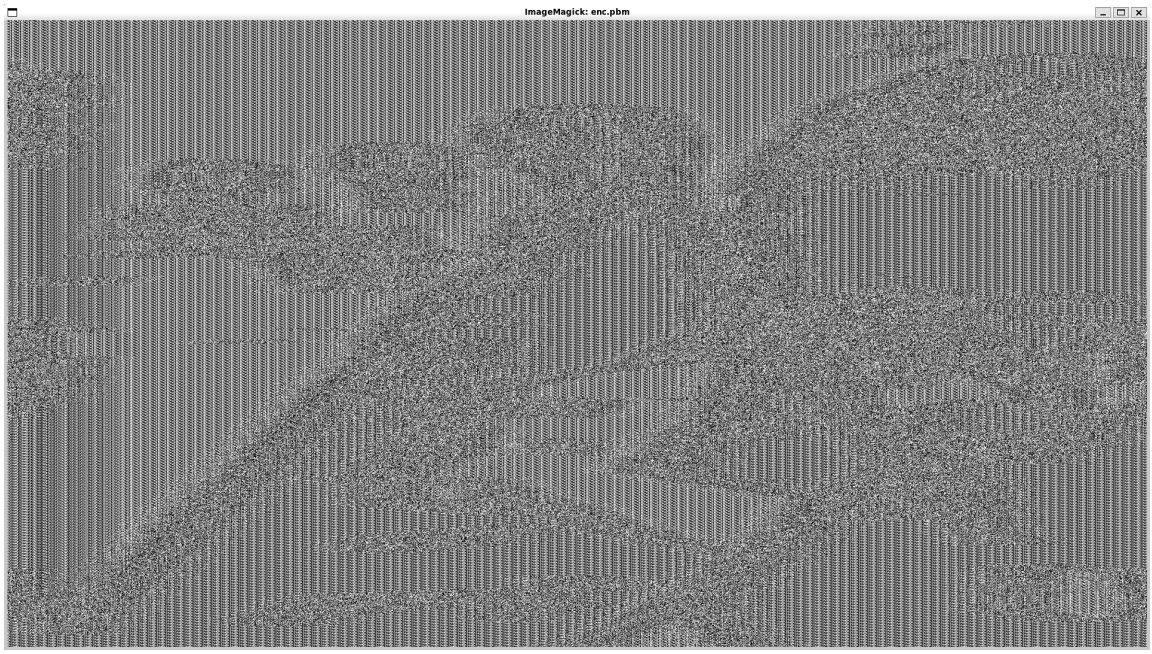
เมื่อรู้ความยาวคีย์แล้ว สามารถแยก ciphertext **เป็น ชุดของ** Caesar cipher **แต่ละชุด → ถอดรหัสได้ง่ายขึ้น**

3

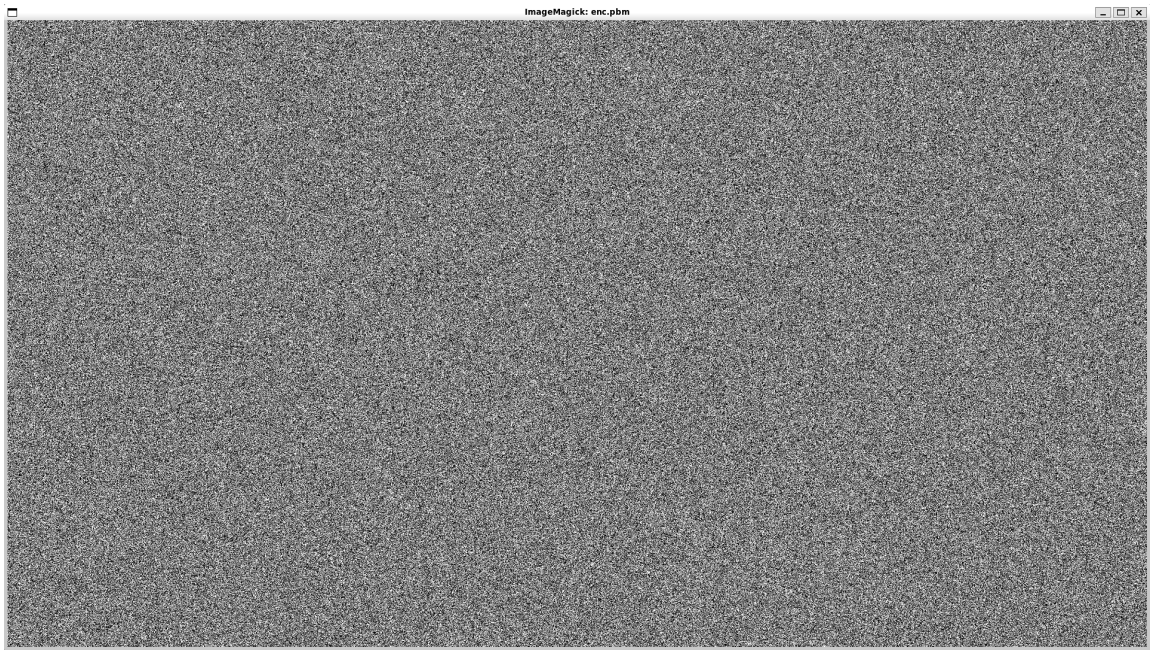
original

org.pbm

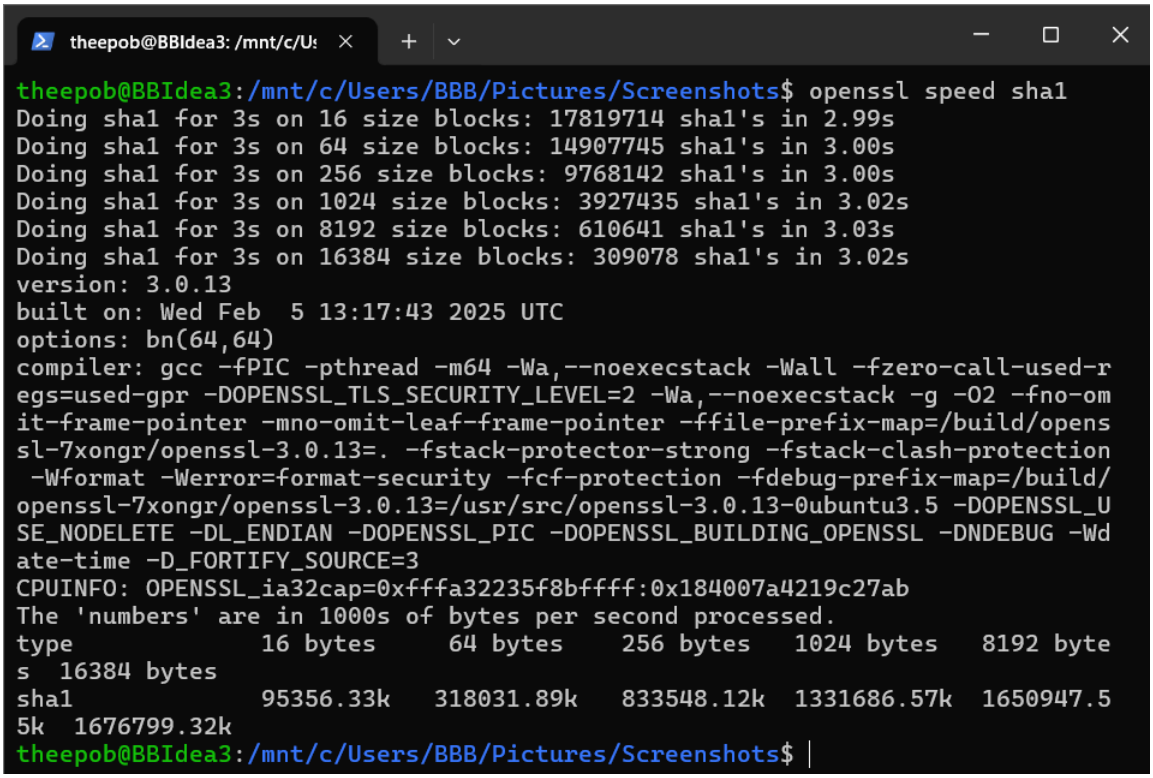
ecb_enc.pbm

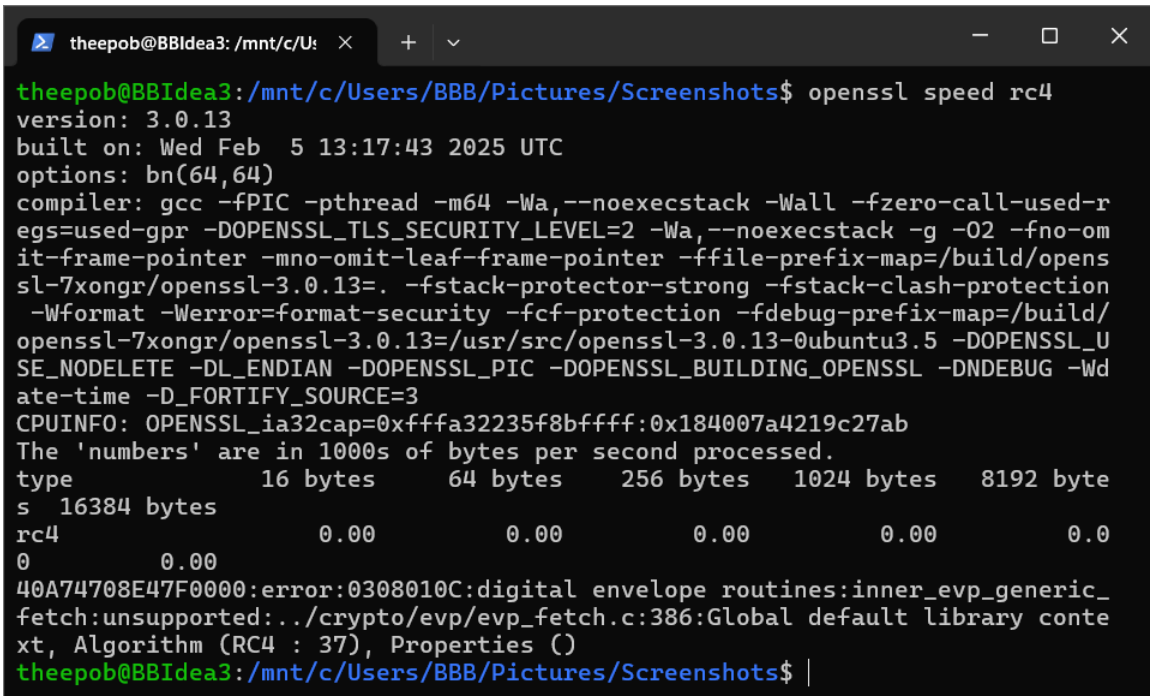


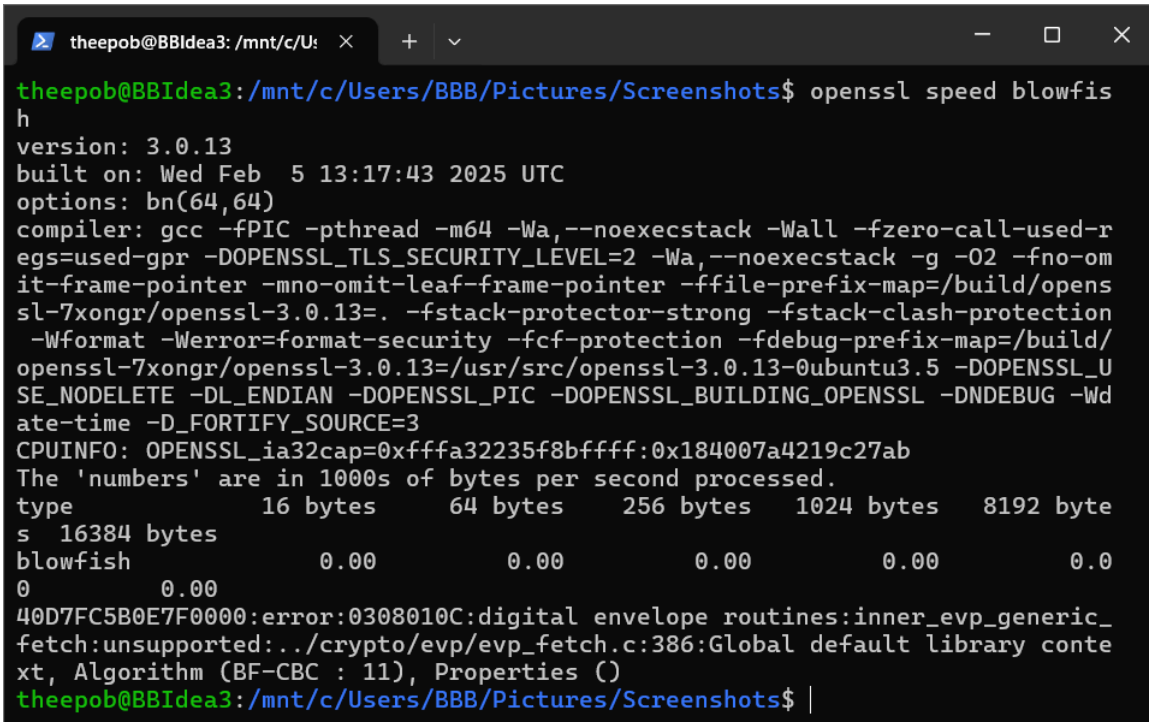
cbc_enc.pbm

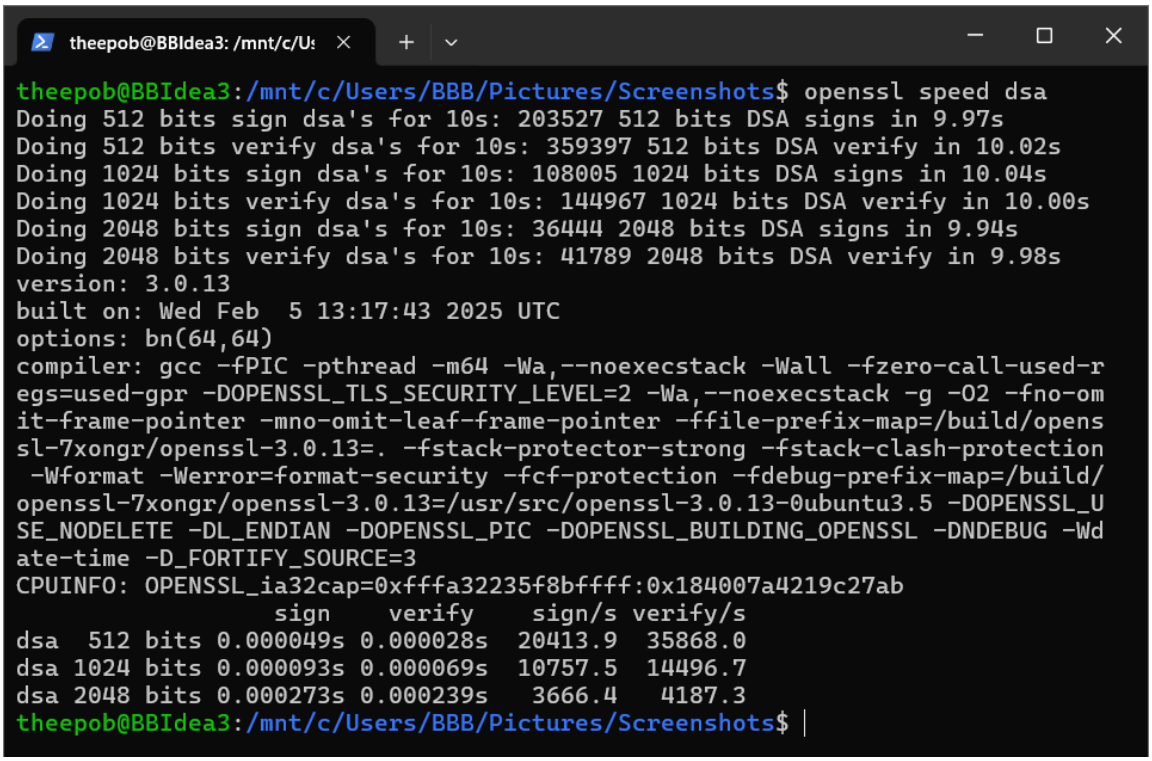


4


theepob@BBIdea3: /mnt/c/U
theepob@BBIdea3:/mnt/c/Users/BBB/Pictures/Screenshots$ openssl speed sha1
Doing sha1 for 3s on 16 size blocks: 17819714 sha1's in 2.99s
Doing sha1 for 3s on 64 size blocks: 14907745 sha1's in 3.00s
Doing sha1 for 3s on 256 size blocks: 9768142 sha1's in 3.00s
Doing sha1 for 3s on 1024 size blocks: 3927435 sha1's in 3.02s
Doing sha1 for 3s on 8192 size blocks: 610641 sha1's in 3.03s
Doing sha1 for 3s on 16384 size blocks: 309078 sha1's in 3.02s
version: 3.0.13
built on: Wed Feb 5 13:17:43 2025 UTC
options: bn(64,64)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -fzero-call-used-regs=used-gpr -DOPENSSL_TLS_SECURITY_LEVEL=2 -Wa,--noexecstack -g -O2 -fno-omit-frame-pointer -mno-omit-leaf-frame-pointer -ffile-prefix-map=/build/openssl-7xongr/openssl-3.0.13=. -fstack-protector-strong -fstack-clash-protection -Wformat -Werror=format-security -fcf-protection -fdebug-prefix-map=/build/openssl-7xongr/openssl-3.0.13=/usr/src/openssl-3.0.13-0ubuntu3.5 -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDEBUG -Wdate-time -D_FORTIFY_SOURCE=3
CPUINFO: OPENSSL_ia32cap=0xfffa32235f8bffff:0x184007a4219c27ab
The 'numbers' are in 1000s of bytes per second processed.
type 16 bytes 64 bytes 256 bytes 1024 bytes 8192 bytes 16384 bytes
sha1 95356.33k 318031.89k 833548.12k 1331686.57k 1650947.55k 1676799.32k
theepob@BBIdea3:/mnt/c/Users/BBB/Pictures/Screenshots$


theepob@BBIdea3: /mnt/c/U
theepob@BBIdea3:/mnt/c/Users/BBB/Pictures/Screenshots$ openssl speed rc4
version: 3.0.13
built on: Wed Feb 5 13:17:43 2025 UTC
options: bn(64,64)
compiler: gcc -fPIC -pthread -m64 -Wa,--noexecstack -Wall -fzero-call-used-regs=used-gpr -DOPENSSL_TLS_SECURITY_LEVEL=2 -Wa,--noexecstack -g -O2 -fno-omit-frame-pointer -mno-omit-leaf-frame-pointer -ffile-prefix-map=/build/openssl-7xongr/openssl-3.0.13=. -fstack-protector-strong -fstack-clash-protection -Wformat -Werror=format-security -fcf-protection -fdebug-prefix-map=/build/openssl-7xongr/openssl-3.0.13=/usr/src/openssl-3.0.13-0ubuntu3.5 -DOPENSSL_USE_NODELETE -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDEBUG -Wdate-time -D_FORTIFY_SOURCE=3
CPUINFO: OPENSSL_ia32cap=0xfffa32235f8bffff:0x184007a4219c27ab
The 'numbers' are in 1000s of bytes per second processed.
type 16 bytes 64 bytes 256 bytes 1024 bytes 8192 bytes 16384 bytes
rc4 0.00 0.00 0.00 0.00 0.00 0.00
40A74708E47F0000:error:0308010C:digital envelope routines:inner_evp_generic_fetch:unsupported:../crypto/evp/evp_fetch.c:386:Global default library context, Algorithm (RC4 : 37), Properties ()
theepob@BBIdea3:/mnt/c/Users/BBB/Pictures/Screenshots$





Question

b

SHA1,RC4,Blowfish ถูกออกแบบมาเพื่อความเร็ว โดยมีอัตราการส่งข้อมูลสูงในระดับเมกะไบต์ต่อวินาที (MB/s) ประสิทธิภาพของพวกมันค่อนข้างใกล้เคียงกัน โดยรหัสแบบสตรีมอย่าง RC4 มักจะเหนือกว่าเล็กน้อยเนื่องจากการออกแบบที่เรียบง่ายกว่า

DSA ต้องใช้การคำนวณที่ซับซ้อนมาก ประสิทธิภาพของมันจึงช้ากว่า

c

1.สร้างค่าแฮชของข้อความ

2.ใช้ กุญแจส่วนตัวของผู้ส่งมาลงลายเซ็นบนค่าแฮช เพื่อยืนยันว่า ผู้ส่งเป็นเจ้าของกุญแจจริง

3.ผู้รับคำนวณค่าแฮชของข้อความที่ได้รับและตรวจสอบลายเซ็นด้วย กุญแจสาธารณะของผู้ส่ง หากตรงกัน → ข้อความถูกต้องและไม่ถูกแก้ไข

hase function มีจุดแข็งคือทำให้ตรวจสอบข้อความเร็ว, ลดขนาดข้อมูล แต่ถ้า hash อ่อน → อาจถูกโจมตี

Asymmetric encryption มีจุดแข็งคือยืนยันตัวตนผู้ส่งและป้องกันการปฏิเสธ แต่ช้า, ต้องพึ่งพาความปลอดภัยของกุญแจ