Homework 1: Applied Cryptography - Theory and Practice

Names: Mateo Fuertes (00321987)
Melisa Guerrero (00322205)
Joel Cuascota (00327494)
Course: CMP 5006 - Information Security (NRC: 1230)
Institution: Universidad San Francisco de Quito

1 Pico CTF

1.1 The Numbers

First technique: Direct Number-to-Letter Mapping

This method involves a simple numerical-to-letter conversion based on the alphabetical position of letters. Each number in the given list directly represents a letter in the English alphabet, where 1 corresponds to 'A', 2 corresponds to 'B', ..., up to 'Z'. The numbers are simply mapped back to their respective letters to reconstruct the plaintext message.

Second Technique: Caesar Cipher with Dynamic Shift

This method is similar but incorporates an additional step: a dynamically calculated shift based on a known phrase. Instead of directly mapping numbers to letters, the process assumes that the first part of the encrypted message corresponds to the phrase `"picoCTF"`.

The method proceeds as follows:

1. Convert the encrypted message into a list of numbers representing the positions of letters.
2. Extract two lists:
   - One representing the known phrase `"picoCTF"` (converted to numerical positions).
   - The second containing the encrypted flag.
3. Compute the Caesar Cipher shift by comparing the first number of the known phrase with the first number of the encrypted flag.
4. Apply the shift in reverse to each number in the encrypted flag to recover the original letters.

1.2 C3

First Technique: Manual Decryption and Execution

The first method involves manually decrypting the given ciphertext using a modified version of the `convert.py` script. The original script processes text using a predefined lookup table. The encryption method maps characters between these lookup tables while keeping track of a dynamic shift based on the previous character's position. To decrypt the text, the inverse transformation was applied.

After running this modified script, the output was a Python 2 script containing a hidden message extraction mechanism. Since the script was written in Python 2, minor modifications were made to ensure compatibility with Python 3. By running this script with the obtained Python 2 code as input, the final flag was successfully extracted.

Second Technique: Automating the Entire Process

To streamline the decryption process, the second method automated all steps into a single pipeline. Instead of manually executing the individual scripts, a program was developed.

The automated solution eliminates the need for manual execution, as it:

- Performs the decryption step using the inverse transformation of `convert.py`.
- Extracts the hidden message by identifying characters located at cubic indices in the decrypted text.
- Executes the entire process in a single run.

By integrating both decryption and extraction processes into a unified pipeline, this method provides a more efficient and reproducible solution to the problem.

1.3 rsa_oracle

First Technique: Server Interaction via Pwn Library

This technique involves interacting with a remote server using Python's `pwn` library to decrypt a password. The process begins by establishing a connection with the server and sending the command `'E'` to trigger the encryption function. The server responds with an encrypted value, which is stored for further processing.

Next, the ciphertext is retrieved from a local file and sent to the server. The decryption process is initiated by sending the command `'D'`. The server returns a modular result, which is then manipulated by performing mathematical operations: multiplying this value by the ciphertext and sending it back to the server. The server returns the decrypted password in hexadecimal format.

The final step involves adjusting the result by dividing it by 2 and converting it from bytes to a UTF-8 string, successfully recovering the original password. This approach exploits modular arithmetic and remote server interaction to decrypt the message efficiently.

Second Technique: Server Automation via Telnet

This technique automates interaction with a remote server using the `telnetlib` module in Python. The process starts by establishing a Telnet connection to the server and retrieving the initial banner, which indicates where the decryption process begins.

The encrypted password is loaded from a local file and used as input for the decryption process. Commands (`'E'` for encryption and `'D'` for decryption) are sent to the server, with responses processed at each step. The server returns values such as a modular result (c_a), which is then used in further calculations.

To decrypt the password, the necessary mathematical operations are applied, including multiplying values and dividing the result by 2. The final server response, given as a hexadecimal string, is converted into the original password using byte manipulation and UTF-8 decoding. This method leverages remote communication, modular arithmetic, and automated interaction to successfully decrypt the password.

1.4 interencdec

First Technique: Caesar Cipher Brute-Force

The first step involved decoding a *Base64-encoded message* twice. The given encoded string:

```
YidkM0JxZGtwQlRYdHFhR3g2YUhsZmF6TnFlVGwzWVROclh6YzRNalV3YUcxcWZRPT0nCg==
```

was decoded once to produce another Base64-encoded string:

```
b'd3BqdkpBTXtqaGx6aHlfazNqeTl3YTNrXzc4MjUwaG1qfQ=='
```

After a second Base64 decoding, we obtained:

```
wpjvJAM{jhlzhy_k3jy9wa3k_78250hmj}
```

At this stage, the flag format was visible, but the text remained unreadable.
To decrypt it, we applied a *Caesar cipher brute-force approach*. The script attempted all possible shifts (0-25), searching for the occurrence of `"picoCTF"`. The correct shift found was 19, revealing the final message.

Second Technique: Affine Cipher Brute-Force Decryption

The second technique used after decoding with Base64, was using an *Affine Cipher*. The decryption process involved:

1. Identifying the encryption formula:
$$E(x) = (a \cdot x + b) \mod 26$$
2. Brute-forcing the values of a and b:
   - Only values of a that have a modular inverse in $\mathbb{Z}_{26}$ were considered.
   - Each valid (a, b) combination was tested until meaningful plaintext.

By iterating through all possible decryption keys, the script successfully recovered the original text.

1.5 HideToSee

First Technique: Steghide and Atbash

In this approach, `Steghide`, a steganography tool, was used to extract a hidden message from an encrypted image. Since no password was provided, it was assumed that the image contained pre-encrypted data that did not require authentication for extraction.

Once the hidden data was retrieved, it was found to be encoded using the Atbash cipher, a classical substitution cipher where each letter in the alphabet is replaced by its opposite (e.g., 'A' becomes 'Z', 'B' becomes 'Y'). Decrypting with Atbash revealed the plaintext message.

Second Technique: Stegseek and Affine

The second approach involved using `Stegseek`, a brute-force steganography tool, to recover the password required to extract the hidden data. Since the password was unknown, `Stegseek` was run with a dictionary attack to identify the correct key.

After successfully extracting the encrypted message, it was determined that the text was encoded using the Affine cipher. The Affine cipher is a mathematical encryption method that applies a transformation based on two parameters. Using the correct decryption keys, the hidden message was recovered.

2 Theory

2.1 Exercise 1

If an encryption function e_K is identical to the decryption function d_K, then the key K is said to be an involutory key. Find all the involutory keys in the Shift Cipher.

A key K is involutory if applying the encryption function twice results in the original plaintext:

$$d_K(e_K(M)) = M$$

Substituting the encryption and decryption functions:

$$(M + K) - K \equiv M \mod 26$$

$$M \equiv M \mod 26$$

This equation always holds for any K, which means that every K correctly decrypts the ciphertext. However, for a key to be truly involutory, we require:

$$e_K(e_K(M)) = M$$

which means:

$$e_K(C) = (C + K) \mod 26$$

since the cypher function for Shift Cypher is $C = (M + K) \mod 26$. Applying e_K again:

$$e_K(e_K(M)) = ((M + K) + K) \mod 26$$

$$e_K(e_K(M)) = (M + 2K) \mod 26$$

For K to be involutory, we must have:

$$M + 2K \equiv M \mod 26$$

Canceling M from both sides:

$$2K \equiv 0 \mod 26$$

Now, solving for K, we have

$$2K \equiv 0 \mod 26$$

This means $2K$ must be a multiple of 26:

$$K = \frac{26n}{2} = 13n, \quad \text{where } n \text{ is an integer}$$

Since K is an integer in $\{0, 1, 2, ..., 25\}$, the only valid values of K are:

$$K = 0, 13$$

- $K = 0$: No shift occurs; the message remains unchanged.
- $K = 13$: This corresponds to the ROT13 cipher, where each letter is shifted by 13 places, making it its own inverse.

Thus, the involutory keys in the Shift Cipher are:

$$K = 0, 13$$

2.2 Exercise 2

Suppose that π is the following permutation of $x \in \{1, \cdots, 8\} : \pi(x) = \{4, 1, 6, 2, 7, 3, 8, 5\}$

a. Compute π^{-1}

Consider the following permutation:

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 1 & 6 & 2 & 7 & 3 & 8 & 5 \end{pmatrix}$$

In cyclic notation:

$$\pi = (142)(36)(578)$$

In order to find its inverse we will compute $\pi^{-1} = [(142)(36)(578)]^{-1}$

$$\begin{aligned} \pi^{-1} &= [(142)(36)(578)]^{-1} \\ &= (142)^{-1}(36)^{-1}(578)^{-1} \\ &= (241)(63)(875) \end{aligned}$$

Thus, $\pi^{-1} = (241)(63)(875)$ or:

$$\pi^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 4 & 6 & 1 & 8 & 3 & 5 & 7 \end{pmatrix}$$

b. Decrypt the following ciphertext, for a Permutation Cipher with m = 8, which was encrypted using the key π: TGEEMNELNNTDROEOAAHDOETCSHAEIRLM

We first partition the ciphertext into groups of eight letters:

$$|TGEEMNEL|NNTDROEO|AAHDOETC|SHAEIRLM|$$

Now each group of six letters is rearranged according to the permutation π:

$$|GENTLEME|NDONOTRE|ADEACHOT|HERSMAIL|$$

the plaintext is:

$$GENTLEMENDONOTREADEACHOTHERSMAIL$$

In conclusion:

GENTLEMEN DO NOT READ EACH OTHERS MAIL

2.3 Exercise 3

Below are given four examples of ciphertext, one obtained from a Substitution Cipher, one from a Vigenere Cipher, one from an Affine Cipher, and one unspecified. In each case, the task is to determine the plaintext. Give a clearly written description of the steps you followed to decrypt each ciphertext. This should include all statistical analysis and computations you performed.

2.3.1 Substitution

1. Frequency Analysis

To begin, we analyze the frequency of each letter in the ciphertext.

Letter	Count	Letter	Count	Letter	Count
A	5	K	18	X	7
B	0	L	7	Y	15
C	37	M	5	Z	13
D	8	N	13		
E	12	O	10		
F	9	P	6		
G	24	Q	1		
H	5	S	20		
I	15	U	14		
J	7	W	5		

Table 1: Letter frequency analysis of the ciphertext.

2. Identifying Common Letters

Ciphertext characters that occur at least 10 times are:

G (24), S (20), K (18), Y (15), I (15), Z (13), E (12).

We expect these letters to correspond to a subset of the most frequent English letters: {t, a, o, i, n, s, h, r}.

3. Bigram and Trigram Analysis

To refine our decryption process, we examine the most common **bigrams** (two-letter sequences) and **trigrams** (three-letter sequences) in the ciphertext.

Most Frequent Bigrams in the Ciphertext

CG (7), ZC (7), NC (5), YS (5), CK (5), GO (5), AC (5), CN (5), SF (4), GY (4), GK (4), FZ (4).

Most Frequent Trigrams in the Ciphertext

YSF (3), GOI (3), FZC (3), ZCC (3), CCN (3), CYK (2), JCK (2), GOL (2), ICG (2), CGI (2), NCG (2), GAC (2), CKS (2), SAC (2), CKX (2), KSH (2), ZCN (2), KGO (2), CND (2), NDG (2), DGY (2), GYY (2), YYS (2), JNC (2), CJU (2), UZC (2), CFZ (2), ZEJ (2).

For comparison, the most common bigrams and trigrams in English are:

Bigrams: th, he, in, en, nt, re, er, an, ti, es, on, at, se, nd, or, ar, al, te, co, de, to, ra, et, ed, it, sa, em, ro.
Trigrams: the, and, tha, ent, ing, ion, tio, for, nde, has, nce, edt, tis, oft, sth, men.

The most common trigram in English is 'the', and since 'UZC' appears twice, we might conjecture that

$$d_k(U) = t$$

4. Deciphering the Text

Using our analysis, we make the following substitutions and explain each step:

- We first notice that **ZC** appears frequently. Since **he** is one of the most common bigrams in English, we conjecture:

$$d_k(Z) = h, \quad d_k(C) = e$$

- The trigram **UZC** appears twice, and **the** is the most frequent trigram in English:

$$d_k(U) = t$$

- The letter **G** is one of the most frequent in our ciphertext, and **A** is common in plaintext:

$$d_k(G) = a$$

- The letters **S** and **K** appear frequently in bigrams like **cK**. Since **s** is common in English words, we assign:

$$d_k(S) = o, \quad d_k(K) = s$$

- The bigram **cN** appears frequently. Since **l** is common in bigrams, we assume:

$$d_k(N) = l$$

- The bigram **Ys** appears often. Since **r** is a common letter, we deduce:

$$d_k(Y) = r$$

- Observing the partial plaintext, fragments like **heel** and **arro** resemble **wheel** and **arrow**. Thus:

$$d_k(F) = w$$

- Fragments like **heel** and **arro** resemble **wheel** and **arrow**, leading to:

$$d_k(F) = w$$

- Further analyzing letter frequencies, contextual matches suggest:

$$d_k(E) = i, \quad d_k(J) = c, \quad d_k(X) = p, \quad d_k(H) = f$$
$$d_k(M) = m, \quad d_k(I) = d, \quad d_k(L) = y, \quad d_k(O) = n, \quad d_k(W) = g$$
$$d_k(D) = b, \quad d_k(P) = u, \quad d_k(A) = v, \quad d_k(Q) = j$$

```
--a--ot-ea-leto-row-lowers--t---ar-e--ro---es
EMGLOSUDCGDNCUSWYSFHNSFCYKDPUMLWGYICOXYSIPJCK

--st-s-----e--le--esol---ershoes--e-eso-ro-ea
QPKUGKMGOLICGINCGACKSNISACYKZSCKXECJCKSHYSXCG

----shelso--ea--ra--a-a---o--sa--to-a---o--ht
OIDPKZCNKSHICGIWYGKKGKGOLDSILKGOIUSIGLEDSPWZU

awheel-arrowtohel----lear----t----h-ealw--slo
GFZCCNDGYYSFUSZCNXEOJNCGYEOWEUPXEZGACGNFGLKNS

-e-a--res-e-te-thewheel-arrow-t-stheo-ewheele
ACIGOIYCKXCJUCIUZCFZCCNDGYYSFEUEKUZCSOCFZCCNC

--eh--leo-wh--h-a--e--e-t-a-ter
IACZEJNCSHFZEJZEGMXCYHCJUMGKUCY
```

$$d_k(X) = p$$
$$d_k(H) = f$$

```
i-a--ot-ea-leto-rowflowers--t---ar-e-pro--ces
EMGLOSUDCGDNCUSWYSFHNSFCYKDPUMLWGYICOXYSIPJCK

--stas-a---ea-lea-esol---ershoespiecesofropea
QPKUGKMGOLICGINCGACKSNISACYKZSCKXECJCKSHYSXCG

----shelso--ea--rassasa---o--sa--to-a-i-o--ht
OIDPKZCNKSHICGIWYGKKGKGOLDSILKGOIUSIGLEDSPWZU

awheel-arrowtohelpi-cleari--it-pi-h-ealwa-slo
GFZCCNDGYYSFUSZCNXEOJNCGYEOWEUPXEZGACGNFGLKNS

-e-a--respecte-thewheel-arrowitistheo-ewheele
ACIGOIYCKXCJUCIUZCFZCCNDGYYSFEUEKUZCSOCFZCCNC

--ehicleofwhichia-perfect-aster
IACZEJNCSHFZEJZEGMXCYHCJUMGKUCY
```

5. Deciphered Text

Applying these substitutions, we obtain:

```
imaynotbeabletogrowflowersbutmygardenproduces
EMGLOSUDCGDNCUSWYSFHNSFCYKDPUMLWGYICOXYSIPJCK

justasmanydeadleavesoldovershoespiecesofropea
QPKUGKMGOLICGINCGACKSNISACYKZSCKXECJCKSHYSXCG

ndbushelsofdeadgrassasanybodysandtodayibought
OIDPKZCNKSHICGIWYGKKGKGOLDSILKGOIUSIGLEDSPWZU

awheelbarrowtohelpinclearingitupihavealwayslo
GFZCCNDGYYSFUSZCNXEOJNCGYEOWEUPXEZGACGNFGLKNS

vedandrespectedthewheelbarrowitistheonewheele
ACIGOIYCKXCJUCIUZCFZCCNDGYYSFEUEKUZCSOCFZCCNC

dvehicleofwhichiamperfectmaster
IACZEJNCSHFZEJZEGMXCYHCJUMGKUCY
```

Plaintext: I may not be able to grow flowers but my garden produces just as many dead leavesl, old overshoes, pieces of rope, and bushels of dead grass as anybody's. And today I bought a wheelbarrow to help in clearing it up. I have always loved and respected the wheelbarrow. It is the one wheeled vehicle of which I am perfect master.

2.3.2 Vigenere Cipher

```
KCCPKBGUFDPHQTYAVINRRTMVGRKDNBVFDETDGILTXRGUD
DKOTFMBPVGEGLTGCKQRACQCWDNAWCRXIZAKFTLEWRPTYC
QKYVXCHKFTPONCQQRHJVAJUWETMCMSPKQDYHJVDAHCTRL
SVSKCGCZQQDZXGSFRLSWCWSJTBHAFSIASPRJAHKJRJUMV
GKMITZHFPDISPZLVLGWTFPLKKEBDPGCEBSHCTJRWXBAFS
PEZQNRWXCVYCGAONWDDKACKAWBBIKFTIOVKCGGHJVLNHI
FFSQESVYCLACNVRWBBIREPBBVFEXOSCDYGZWPFDTKFQIY
CWHJVLNHIQIBTKHJVNPIST
```

1. Kasiski Examination

First, we apply the Kasiski test by looking for repeated trigrams. The trigram `HJV` occurs at positions [107, 125, 263, 317, 329]. The distances between these positions are:

$$18, \quad 138, \quad 54, \quad 12.$$

The greatest common divisor (GCD) of these distances is 6, suggesting that the keyword length is $m = 6$.

2. Index of Coincidence (IC)

Next, we divide the ciphertext into 6 subcolumns (one for each position modulo 6) and compute the Index of Coincidence for each subcolumn:

$$Ic(x) \;=\; \frac{\sum_{i=0}^{25} f_i\,(f_i - 1)}{n\,(n-1)},$$

where f_i is the frequency of the ith letter in that subcolumn, and n is the subcolumn's length. For English text under a monoalphabetic mapping, we expect $Ic \approx 0.065$. In contrast, a random string would have $Ic \approx 0.038$.

The table below summarizes the IC values computed for various assumed key lengths m:

Column	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$	$m = 7$	$m = 8$
1	0.038	0.056	0.037	0.043	0.063	0.031	0.033
2	0.047	0.048	0.043	0.043	0.084	0.044	0.041
3		0.048	0.038	0.033	0.049	0.043	0.034
4			0.049	0.035	0.065	0.041	0.041
5				0.043	0.043	0.044	0.039

Table 2: Index of Coincidence for Different Key Lengths (m)

The values for $m = 6$ are consistent with those expected for English text, further confirming that the key length is 6.

3. Determining the Key

Assuming $m = 6$, each subcolumn is treated as a simple shift cipher. By comparing the frequency distribution in each subcolumn with standard English frequencies (using the M_g correlation method), the optimal shifts were found to be:

$$[2,\ 17,\ 24,\ 15,\ 19,\ 14\,].$$

Mapping $0 \to$ `A`, $1 \to$ `B`, ... these shifts correspond to the letters:

`C`, `R`, `Y`, `P`, `T`, `O`.

Hence, the recovered keyword is `CRYPTO`.

4. Decrypted Plaintext

```
ILEARNEDHOWTOCALCULATETHEAMOUNTOFPAPERNEEDEDF
ORAROOMWHENIWASATSCHOOLYOUMULTIPLYTHESQUAREFO
OTAGEOFTHEWALLSBYTHECUBICCONTENTSOFTHEFLOORAN
DCEILINGCOMBINEDANDDOUBLEITYOUTHENALLOWHALFTH
ETOTALFOROPENINGSSUCHASWINDOWSANDDOORSTHENYOU
ALLOWTHEOTHERHALFFORMATCHINGTHEPATTERNTHENYOU
DOUBLETHEWHOLETHINGAGAINTOGIVEAMARGINOFERRORA
NDTHENYOUORDERTHEPAPER
```

Plaintext:

I learned how to calculate the amount of paper needed for a room when I was at school. You multiply the square footage of the walls by the cubic contents of the floor and ceiling combined and double it. You then allow half the total for openings such as windows and doors. Then you allow the other half for matching the pattern. Then you double the whole thing again to give a margin of error, and then you order the paper.

2.3.3 Affine Cipher

```
KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP
KRIOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCCAFIEABDKP
BCPFEQPKAZBKRHAIBKAPCCIBURCCDKDCCJCIDFUIXPAFF
ERBICZDFKABICBBENEFCUPJCVKABPCYDCCDPKBCOCPERK
IVKSCPICBRKIJPKABI
```

1. Frequency Analysis

Table 3 shows the count of each letter in the ciphertext. This count helps us guess which plaintext letters map to the most frequent letters in the ciphertext.

Letter	Count	Letter	Count	Letter	Count
A	13	H	1	O	2
B	21	I	16	P	20
C	32	J	6	Q	4
D	9	K	20	R	12
E	13	N	1	S	1
F	10	U	6	V	4
G	-	X	2	Y	1
Z	4				

Table 3: Letter frequency analysis of the ciphertext.

2. Hypothesis and Equation Setup

In English, the letters e and t tend to be the most frequent. Using the numeric convention

$$A = 0,\ B = 1,\ C = 2,\ \ldots,\ Z = 25,$$

we have:

$$\mathrm{e} \mapsto 4, \quad \mathrm{t} \mapsto 19.$$

We assume that the most frequent letter in the ciphertext (for instance, C, which is 2) corresponds to e (4). Hence, in the Affine Cipher function

$$e_K(x) \ = \ a\,x + b \pmod{26},$$

we impose

$$e_K(4) = 2 \quad (\text{“e” encrypts to “C”}).$$

Next, we pick another highly frequent letter (or the next most likely one) and assume it encrypts from t (19). This gives us a second pair and, therefore, a system of two linear equations modulo 26:

$$\begin{cases} 4a + b \equiv 2 \pmod{26}, \\ 19a + b \equiv 1 \pmod{26}. \end{cases}$$

3. System Solution and Key Recovery

3.1. Subtracting the Equations

We subtract the first equation from the second to eliminate b:

$$(19a + b) - (4a + b) \ = \ 15a,$$

and on the right-hand side,

$$1 - 2 = -1 \ \equiv \ 25 \pmod{26}.$$

Thus,

$$15a \equiv 25 \pmod{26}.$$

3.2. Finding a and b

To solve $15a \equiv 25 \pmod{26}$, we seek the inverse of 15 in $\mathbb{Z}_{26}$. It can be checked that $15 \times 7 = 105 \equiv 1 \pmod{26}$, so $15^{-1} \equiv 7$. Multiplying both sides by 7:

$$a \equiv 25 \times 7 = 175 \equiv 19 \pmod{26}.$$

Next, substitute $a = 19$ into $4a + b \equiv 2$:

$$4 \cdot 19 + b = 76 + b \equiv 2 \pmod{26}.$$

Since $76 \equiv 24 \pmod{26}$,

$$24 + b \equiv 2 \pmod{26} \implies b \equiv 2 - 24 \equiv -22 \equiv 4 \pmod{26}.$$

Hence, the Affine Cipher key is

$$\boxed{a = 19, \quad b = 4.}$$

The condition $\gcd(a, 26) = 1$ is satisfied, so this key is valid.

Additional Note: Summary of Three Systems

We can also summarize the step-by-step resolution for the three related systems:

$$\begin{cases} 4a + b \equiv 2 \pmod{26}, \\ 19a + b \equiv t \pmod{26}, \end{cases}$$

where t takes the values 1, 15, and 10. In each case, subtracting the first equation from the second gives us a different right-hand side (-1, 13, or 8), leading to distinct values for a and b. Only when $t = 1$ do we obtain $(a, b) = (19, 4)$, which is a valid Affine key.

- **System 1** ($t = 1$):
$$15a \equiv -1 \equiv 25 \pmod{26} \implies a = 19,\ b = 4 \text{ (valid)}.$$
- **System 2** ($t = 15$):
$$15a \equiv 13 \pmod{26} \implies a = 13,\ b = 2\ (\gcd(13, 26) \neq 1,\ \text{invalid}).$$
- **System 3** ($t = 10$):
$$15a \equiv 8 \pmod{26} \implies a = 4,\ b = 12\ (\gcd(4, 26) \neq 1,\ \text{invalid}).$$

Thus, $\boxed{(a, b) = (19, 4)}$ is the only suitable pair among these options.

4. Encryption and Decryption Functions

4.1. Encryption With $(a, b) = (19, 4)$, the encryption function is

$$e_K(x) = 19x + 4 \pmod{26}.$$

4.2. Decryption To decrypt, we need the inverse of 19 in $\mathbb{Z}_{26}$. We verify that $19 \times 11 = 209 \equiv 1 \pmod{26}$, so $19^{-1} \equiv 11$. Hence, the decryption function is

$$d_K(y) = 11\,(y - 4) \pmod{26}.$$

Equivalently,

$$d_K(y) \equiv 11y - 44 \equiv 11y + 8 \pmod{26}$$

(because $-44 \equiv 8$ in $\mathbb{Z}_{26}$).

5. Plaintext

Once we have the decryption function $d_K(y)$, we apply it to each letter of the ciphertext:

$$y = \text{(cipher)} \longmapsto x = d_K(y) \longmapsto \text{(plaintext letter)}.$$

This process reveals a perfectly readable English message, confirming that the choice $(a, b) = (19, 4)$ was correct and that our assumption about the most frequent letters (e $\mapsto$ C, t $\mapsto$ B) was accurate.

5.1 Decrypted Plaintext

```
OCANADATERREDENOSAIEUXTONFRONTESTCEINTDEFLEUR
ONSGLORIEUXCARTONBRASSAITPORTERLEPEEILSAITPOR
TERLACROIXTONHISTOIREESTUNEEPOPEEDESPLUSBRILL
ANTSEXPLOITSETTAVALEURDEFOITREMPEEPROTEGERANO
SFOYERSETNOSDROITS
```

Plaintext: ocanada terredenosa ieuxton frontest ceintdefleur onsglorieux cartonbrassait porterlepee ilsaitpor terlacroixt onhistoire estuneep oopeedesplusbrill antsexploit settavaleurdefoit rempeeprotegerano sfoyerset nosdroits

2.3.4 Unspecified Cipher

```
BNVSNSIHQCEELSSKKYERIFJKXUMBGYKAMQLJTYAVFBKVT
DVBPVVRJYYLAOKYMPQSCGDLFSRLLPROYGESEBUUALRWXM
MASAZLGLEDFJBZAVVPXWICGJXASCBYEHOSNMULKCEAHTQ
OKMFLEBKFXLRRFDTZXCIWBJSICBGAWDVYDHAVFJXZIBKC
GJIWEAHTTOEWTUHKRQVVRGZBXYIREMMASCSPBNLHJMBLR
FFJELHWEYLWISTFVVYFJCMHYUYRUFSFMGESIGRLWALSWM
NUHSIMYYITCCQPZSICEHBCCMZFEGVJYOCDEMMPGHVAAUM
ELCMOEHVLTIPSUYILVGFLMVWDVYDBTHFRAYISYSGKVSUU
HYHGGCKTMBLRX
```

1. Kasiski Test and Index of Coincidence (IC)

Kasiski Test: By looking for repeated sequences (e.g., trigrams) in the ciphertext and measuring the distances between their occurrences, one can often guess the key length. For example, if the same trigram appears at positions 38 and 170, the distance is $170 - 38 = 132$. The greatest common divisor (GCD) of these distances typically suggests divisors that may be the actual key length.

Index of Coincidence (IC): The IC indicates how likely it is for two randomly picked letters in a text to be the same. For English, the IC is about 0.066. When we split a Vigenère ciphertext correctly according to its key length, each substring resembles a simple substitution cipher whose IC tends to match that of English.

IC Table Example Below is an example of the IC values computed for candidate key lengths $m \in \{2, 3, 4, 5, 6, 7, 8\}$. Rows (0–7) often represent different internal splits or offsets:

Index	2	3	4	5	6	7	8
0	0.046333	0.044258	0.043468	0.045045	0.051203	0.039832	0.058279
1	0.046207	0.047863	0.056335	0.042523	0.061343	0.044724	0.055504
2		0.048387	0.046517	0.039640	0.054997	0.042747	0.050879
3			0.047920	0.043317	0.070862	0.037736	0.046253
4				0.037023	0.055526	0.046444	0.037928
5					0.068904	0.033382	0.064734
6						0.046237	0.046377
7							0.050242

Table 4: Index of Coincidence (IC) values for different candidate key lengths.

When one of these candidate lengths shows consistently higher IC values (closer to the expected IC for English), it becomes a strong guess for the true key length.

Analysis with $m = 6$ After using the above approaches (Kasiski distances and IC checks), we hypothesize that $m = 6$. We then split the ciphertext into 6 substrings:

$$y_0,\ y_1,\ y_2,\ y_3,\ y_4,\ y_5,$$

and compute a score for each possible shift $g \in \{0, \ldots, 25\}$ on each y_i. The shift maximizing the score is chosen as k_i. Converting k_i to letters (0 = A, 1 = B, etc.) gives the recovered key.

2. Results (Scores)

Substring y_0 (length 63): best shift $g = 19$ (T)

Substring y_1 (length 62): best shift $g = 7$ (H)

Substring y_2 (length 62): best shift $g = 4$ (E)

Substring y_3 (length 62): best shift $g = 14$ (O)

Substring y_4 (length 62): best shift $g = 17$ (R)

Substring y_5 (length 62): best shift $g = 24$ (Y)

Hence, the final key is **THEORY**.

3. Decrypted Plaintext

```
IGREWUPAMONGSLOWTALKERSMENINPARTICULARWHODROP
PEDWORDSAFEWATATIMELIKEBEANSINAHILLANDWHENIGO
TTOMINNEAPOLISWHEREPEOPLETOOKALAKEWOBEGONCOMM
ATOMEANTHEENDOFASTORYICOULDNTSPEAKAWHOLESENTE
NCEINCOMPANYANDWASCONSIDEREDNOTTOOBRIGHTSOIEN
ROLLEDINASPEECHCOURSETAUGHTBYORVILLESANDTHEFO
UNDEROFREFLEXIVERELAXOLOGYASELFHYPNOTICTECHNI
QUETHATENABLEDAPERSONTOSPEAKUPTOTHREEHUNDREDW
ORDSPERMINUTE
```

Plaintext: I grew up among slow talkers. Men in particular who dropped words a few at a time like beans in a hill. And when I got to Minneapolis, where people took a Lake Wobegon comma to mean the end of a story, I couldn't speak a whole sentence in company and was considered not too bright. So I enrolled in a speech course taught by Orville Sand, the founder of reflexive relaxology, a self hypnotic technique that enabled a person to speak up to three hundred words per minute.

2.4 Exercise 4

Prove that the Affine Cipher achieves perfect secrecy if every key is used with equal probability 1/312.

Proof. The **Affine Cipher** encrypts a plaintext letter M (represented as an integer mod n) using a key (a, b) as follows:

$$C = (aM + b) \mod n$$

where $n = 26$ is the size of the alphabet; a is an integer such that $\gcd(a, n) = 1$ (ensuring invertibility); and b is any integer in $\mathbb{Z}_n$.

A cipher satisfies **perfect secrecy** if, for all possible plaintexts M and ciphertexts C, we have:

$$P(M|C) = P(M)$$

This means that observing the ciphertext C does not provide any additional information about the original message M.

By Bayes' Theorem:

$$P(M|C) = \frac{P(C|M)P(M)}{P(C)}$$

In consequence, if **P(C | M) = P(C)**, perfect secrecy holds.

Given a plaintext M, the number of keys (a, b) that transform M into a specific ciphertext C is determined by the equation:

$$C = (aM + b) \mod 26$$

For each valid a, there exists a unique b satisfying this equation. Since a must be coprime to 26, there are $\phi(26) = 12$ valid choices for a, and for each a, there is exactly one corresponding b for any C. Thus, each plaintext M can be mapped to a given ciphertext C in exactly 12 ways.

Since there are a total of 312 possible keys (as a has 12 choices and b has 26 choices), and each key is chosen with equal probability $\frac{1}{312}$, the probability that a specific plaintext M encrypts to a specific ciphertext C is:

$$P(C|M) = \frac{12}{312} = \frac{1}{26}.$$

On the other hand, by the law of total probability, we sum over all possible plaintexts:

$$P(C) = \sum_M P(C|M)P(M)$$

If we assume that all plaintexts are equally probable (i.e. without considering actual frequencies of use each character), then:

$$P(M) = \frac{1}{26}, \quad \forall M.$$

Since there are 26 possible plaintexts and each transforms into C with probability $\frac{1}{26}$, we obtain:

$$P(C) = \sum_M \frac{1}{26} P(M) = \frac{1}{26} \sum_M P(M) = \frac{1}{26} \cdot 1 = \frac{1}{26}.$$

Since we have established that:

$$P(C|M) = P(C) = \frac{1}{26},$$

this implies that:

$$P(M|C) = P(M)$$

and therefore, the Affine Cipher satisfies the definition of perfect secrecy.

Thus, we have proven that the Affine Cipher achieves perfect secrecy when each key is chosen with equal probability $\frac{1}{312}$.

□

2.5 Exercise 5

Prove that if a cryptosystem has perfect secrecy and |K|= |C|= |P|, then every ciphertext is equally probable.

Proof. Let:

- P be the plaintext space,
- C be the ciphertext space,
- K be the key space,
- $P(p)$ be the probability distribution over the plaintexts,
- $P(c)$ be the probability distribution over the ciphertexts.

A cryptosystem is said to have perfect secrecy if, for any two plaintexts p_1 and p_2, the probability of obtaining a ciphertext c given p_1 is the same as the probability of obtaining c given p_2. In mathematical terms:

$$P(c \mid p_1) = P(c \mid p_2) \quad \text{for all } p_1, p_2 \in P \text{ and } c \in C.$$

For **perfect secrecy**, the ciphertext must reveal no information about the plaintext. This means that the probability distribution over the ciphertexts must be independent of the plaintext. Hence, for all $p \in P$, the ciphertext distribution $P(c \mid p)$ should be identical.

Mathematically:

$$P(c \mid p) = P(c) \quad \text{for all } p \in P.$$

This means that the probability of observing ciphertext c should not depend on the plaintext p.

Using **Bayes' theorem**, the conditional probability of ciphertext c given plaintext p is:

$$P(c \mid p) = \frac{P(p \mid c)P(c)}{P(p)}.$$

For perfect secrecy to hold, $P(c \mid p)$ must be **independent** of p. Therefore, the numerator $P(p \mid c)P(c)$ must be independent of p, meaning the distribution $P(c)$ must be uniform across all ciphertexts.

If every plaintext p leads to the same distribution over ciphertexts, then the ciphertext distribution $P(c)$ must be **uniform**, meaning that all ciphertexts are equally probable. Specifically:

$$P(c) = \frac{1}{|C|} \quad \text{for all } c \in C.$$

Since the ciphertext distribution $P(c)$ is independent of the plaintext and every ciphertext is equally probable, we can conclude that, for a cryptosystem with perfect secrecy and $|K| = |C| = |P|$, every ciphertext is equally likely. □

2.6 Exercise 6

Use the EXTENDED EUCLIDEAN ALGORITHM to compute the following multiplicative inverses:

- $17^{-1} \mod 101$
- $357^{-1} \mod 1234$
- $3125^{-1} \mod 9987$

1. $\mathbf{17}^{-1} \mod \mathbf{101}$

We want to find x such that:

$$17x \equiv 1 \mod 101$$

First, we use the Euclide's Algorithm to compute $gcd(17, 101)$:
Dividing successively:

$$101 = 5 \times 17 + 16$$

$$17 = 1 \times 16 + 1$$

$$16 = 16 \times 1 + 0$$

Since $gcd(17, 101) = 1$, then 17 has an inverse modulo 101.
Now, we go backwards to find x. So we express 1 as a linear combination:

$$1 = 17 - 1 \times 16$$

substituting $16 = 101 - 5 \times 17$:

$$1 = 17 - 1 \times (101 - 5 \times 17)$$

$$1 = 17 - 1 \times 101 + 5 \times 17$$

$$1 = 6 \times 17 - 1 \times 101$$

Therefore:

$$x = 6, \quad y = -1$$

and the solution is

$$17^{-1} \equiv 6 \mod 101$$

2. $\mathbf{357}^{-1} \mod \mathbf{1234}$

We want to find x such that:

$$357x \equiv 1 \mod 1234$$

First, we apply the Euclide's Algorithm:

$$1234 = 3 \times 357 + 163$$

$$357 = 2 \times 163 + 31$$

$$163 = 5 \times 31 + 8$$

$$31 = 3 \times 8 + 7$$

$$8 = 1 \times 7 + 1$$

$$7 = 7 \times 1 + 0$$

Since $gcd(357, 1234 = 1$, then 357 has an inverse.
Then, we move back, using these equations to express 1 as a linear combination:

$$1 = 8 - 1 \times 7$$

Substituting $7 = 31 - 3 \times 8$:

$$1 = 8 - 1 \times (31 - 3 \times 8)$$

$$1 = 4 \times 8 - 1 \times 31$$

Substituting $8 = 163 - 5 \times 31$:

$$1 = 4 \times (163 - 5 \times 31) - 1 \times 31$$

$$1 = 4 \times 163 - 21 \times 31$$

Substituting $31 = 357 - 2 \times 163$:

$$1 = 4 \times 163 - 21 \times (357 - 2 \times 163)$$

$$1 = 46 \times 163 - 21 \times 357$$

Substituting $163 = 1234 - 3 \times 357$:

$$1 = 46 \times (1234 - 3 \times 357) - 21 \times 357$$

$$1 = 46 \times 1234 - 159 \times 357$$

Therefore:

$$x = -159$$

which in positive numbers is equivalent to 1234:

$$x = 1075$$

so the solution is

$$357^{-1} \equiv 1075 \mod 1234$$

3. $\mathbf{3125}^{-1} \mod \mathbf{9987}$

We want to find x such that:

$$3125x \equiv 1 \mod 9987$$

First, we use the Euclidean Algorithm:

$$9987 = 3 \times 3125 + 6162$$

$$3125 = 1 \times 6162 + 1187$$

$$6162 = 5 \times 1187 + 414$$

$$1187 = 2 \times 414 + 359$$

$$414 = 1 \times 359 + 55$$

$$359 = 6 \times 55 + 29$$

$$55 = 1 \times 29 + 26$$

$$29 = 1 \times 26 + 3$$

$$26 = 8 \times 3 + 2$$

$$3 = 1 \times 2 + 1$$

$$2 = 2 \times 1 + 0$$

Since $gcd(3125, 9987) = 1$, then 3125 has an inverse.
Now we want to express 1 as a linear combination:

$$1 = 3 - 1 \times 2$$

Substituting $2 = 26 - 8 \times 3$:

$$1 = 3 - 1 \times (26 - 8 \times 3)$$

$$1 = 9 \times 3 - 1 \times 26$$

Substituting $3 = 29 - 1 \times 26$:

$$1 = 9 \times (29 - 1 \times 26) - 1 \times 26$$

$$1 = 9 \times 29 - 10 \times 26$$

Substituting $26 = 55 - 1 \times 29$:

$$1 = 9 \times 29 - 10 \times (55 - 1 \times 29)$$

$$1 = 19 \times 29 - 10 \times 55$$

Substituting $29 = 359 - 6 \times 55$:

$$1 = 19 \times (359 - 6 \times 55) - 10 \times 55$$

$$1 = 19 \times 359 - 124 \times 55$$

Substituting $55 = 414 - 1 \times 359$:

$$1 = 19 \times 359 - 124 \times (414 - 1 \times 359)$$

$$1 = 143 \times 359 - 124 \times 414$$

Substituting $359 = 1187 - 2 \times 414$:

$$1 = 143 \times (1187 - 2 \times 414) - 124 \times 414$$

$$1 = 143 \times 1187 - 410 \times 414$$

Substituting $414 = 6162 - 5 \times 1187$:

$$1 = 143 \times 1187 - 410 \times (6162 - 5 \times 1187)$$

$$1 = 2153 \times 1187 - 410 \times 6162$$

Substituting $1187 = 3125 - 1 \times 6162$:

$$1 = 2153 \times (3125 - 1 \times 6162) - 410 \times 6162$$

$$1 = 2153 \times 3125 - 2563 \times 6162$$

Substituting $6162 = 9987 - 3 \times 3125$:

$$1 = 2153 \times 3125 - 2563 \times (9987 - 3 \times 3125)$$

$$1 = 2153 \times 3125 - 2563 \times 9987 + 7689 \times 3125$$

$$1 = (2153 + 7689) \times 3125 - 2563 \times 9987$$

$$1 = 1844 \times 3125 - 577 \times 9987$$

Therefore:

$$x = 1844$$

so the solution is:

$$3125^{-1} \equiv 1844 \mod 9987$$