Assignment 1

MATHS 3026 / 4026 / 7026 Cryptography

Samuel Chau
a1799298

August 10, 2025

1. a) Given $e(x) = 5x + 11 \pmod{26}$ and the word 'ALMOND'.

i. Encrypt the plaintext ALMOND.

First, we map each letter to its numerical position (A=0, B=1, ..., Z=25):

Letter	A	L	M	O	N	D
Position	0	11	12	14	13	3

Then, using the affine cipher, we can calculate the position for each mapped cipher letter:

$$\begin{aligned}
e(0) &= 5(0) + 11 &&= 11 = 0 \cdot 26 + 11 &&= 11 \pmod{26} \to \text{L} \\
e(11) &= 5(11) + 11 &&= 66 = 2 \cdot 26 + 14 &&= 14 \pmod{26} \to \text{O} \\
e(12) &= 5(12) + 11 &&= 71 = 2 \cdot 26 + 19 &&= 19 \pmod{26} \to \text{T} \\
e(14) &= 5(14) + 11 &&= 81 = 3 \cdot 26 + 3 &&= 3 \pmod{26} \to \text{D} \\
e(13) &= 5(13) + 11 &&= 76 = 2 \cdot 26 + 24 &&= 24 \pmod{26} \to \text{Y} \\
e(3) &= 5(3) + 11 &&= 26 = 1 \cdot 26 + 0 &&= 0 \pmod{26} \to \text{A}
\end{aligned}$$

The mapped ciphertext:

Plaintext	A	L	M	O	N	D
Ciphertext	L	O	T	D	Y	A

Therefore, the encrypted word is **LOTDYA**.

ii. Find the decryption algorithm in the form $d(x) = cx + d$ for some $c, d \in \mathbb{Z}_{26}$.

Starting with the encryption function, for clarity we write it as:

$$c = 5p + 11 \pmod{26}$$

where p is the plaintext position and c is the ciphertext position.
To find the decryption function, we solve for p:

$$\begin{aligned}
c &= 5p + 11 \pmod{26} \\
c - 11 &= 5p \pmod{26}
\end{aligned}$$

To isolate p, we need to divide both sides by 5. However, division is not defined in modular arithmetic, so we instead multiply by the multiplicative inverse of 5. The inverse 5^{-1} satisfies $5 \cdot 5^{-1} \equiv 1 \pmod{26}$.

Testing values, we find:

$$5 \cdot 21 = 105 = 4 \cdot 26 + 1 \equiv 1 \pmod{26}$$

Therefore $5^{-1} = 21$. Multiplying both sides of our equation by 21:

$$\begin{aligned} 21(c - 11) &= 21 \cdot 5p \pmod{26} \\ 21(c - 11) &= p \pmod{26} \quad (\text{since } 21 \cdot 5 \equiv 1 \pmod{26}) \\ 21c - 231 &= p \pmod{26} \end{aligned}$$

To reduce $-231 \pmod{26}$:

$$-231 = -9 \cdot 26 + 3 \equiv 3 \pmod{26}$$

Therefore:

$$p = 21c + 3 \pmod{26}$$

In the required form: $\mathbf{d(x) = 21x + 3}$ where $c = 21, d = 3 \in \mathbb{Z}_{26}$.

To verify, we decrypt the ciphertext letter L (position 11) from part i:

$$\begin{aligned} d(11) &= 21(11) + 3 \pmod{26} \\ &= 231 + 3 \pmod{26} \\ &= 234 \pmod{26} \\ &= 9 \cdot 26 + 0 \pmod{26} \\ &= 0 \pmod{26} \rightarrow \text{A} \end{aligned}$$

This correctly decrypts L back to A, confirming our decryption algorithm.

iii. Decrypt the ciphertext VSLJF.

First, we map each ciphertext letter to its numerical position:

Letter	V	S	L	J	F
Position	21	18	11	9	5

Using the decryption algorithm $d(x) = 21x + 3 \pmod{26}$:

$$\begin{aligned}
d(21) = 21(21) + 3 &= 441 + 3 = 444 &&= 17 \cdot 26 + 2 = 2 \pmod{26} &&\to \text{C} \\
d(18) = 21(18) + 3 &= 378 + 3 = 381 &&= 14 \cdot 26 + 17 = 17 \pmod{26} &&\to \text{R} \\
d(11) = 21(11) + 3 &= 231 + 3 = 234 &&= 9 \cdot 26 + 0 = 0 \pmod{26} &&\to \text{A} \\
d(9) = 21(9) + 3 &= 189 + 3 = 192 &&= 7 \cdot 26 + 10 = 10 \pmod{26} &&\to \text{K} \\
d(5) = 21(5) + 3 &= 105 + 3 = 108 &&= 4 \cdot 26 + 4 = 4 \pmod{26} &&\to \text{E}
\end{aligned}$$

Therefore, the decrypted plaintext is **CRAKE**.

b) Show that you cannot use the affine cipher with the encryption rule $e(x) = 6x + 11 \pmod{26}$ by finding two plaintext letters which encrypt to the same ciphertext letter.

To show that this affine cipher doesn't work, we need to find two different plaintext letters that encrypt to the same ciphertext letter.

Let's test the letters A and N:

For letter A (position 0):

$$e(0) = 6(0) + 11 = 11 \pmod{26} \to \text{L}$$

For letter N (position 13):

$$e(13) = 6(13) + 11 = 78 + 11 = 89 \equiv 11 \pmod{26} \to \text{L}$$

Both A and N encrypt to the same letter L. This is a collision, which makes decryption impossible.

2. a) Encrypt the word CHERRY using the Vigenère cipher.

i. Find the ciphertext when the keyword is BROLGA.

Plaintext Position	2	7	4	17	17	24
Plaintext	C	H	E	R	R	Y
Keyword Position	1	17	14	11	6	0
Keyword	B	R	O	L	G	A

Adding the positions modulo 26:

$$\begin{aligned} 2+1&=3\\ 7+17&=24\\ 4+14&=18\\ 17+11&=28=1\cdot 26+2=2 \pmod{26}\\ 17+6&=23\\ 24+0&=24 \end{aligned}$$

Ciphertext Position	3	24	18	2	23	24
Ciphertext	D	Y	S	C	X	Y

Therefore, the ciphertext is **DYSCXY**.

ii. Find the ciphertext when the keyword is MARTIN.

Plaintext Position	2	7	4	17	17	24
Plaintext	C	H	E	R	R	Y
Keyword Position	12	0	17	19	8	13
Keyword	M	A	R	T	I	N

Adding the positions modulo 26:

$$\begin{aligned} 2+12&=14\\ 7+0&=7\\ 4+17&=21\\ 17+19&=36=1\cdot 26+10=10 \pmod{26}\\ 17+8&=25\\ 24+13&=37=1\cdot 26+11=11 \pmod{26} \end{aligned}$$

Ciphertext Position	14	7	21	10	25	11
Ciphertext	O	H	V	K	Z	L

Therefore, the ciphertext is **OHVKZL**.

b) You receive the ciphertext URWZZZMXZZV.

i. What is the keyword if the plaintext is INFORMATION?

To find the keyword, we subtract plaintext positions from ciphertext positions modulo 26.

Ciphertext	U	R	W	Z	Z	Z	M	X	Z	Z	V
Ciphertext Position	20	17	22	25	25	25	12	23	25	25	21
Plaintext	I	N	F	O	R	M	A	T	I	O	N
Plaintext Position	8	13	5	14	17	12	0	19	8	14	13

Finding keyword positions by subtraction:

$$
\begin{aligned}
k_1 &= 20 - 8 = 12 \to \text{M} \\
k_2 &= 17 - 13 = 4 \to \text{E} \\
k_3 &= 22 - 5 = 17 \to \text{R} \\
k_4 &= 25 - 14 = 11 \to \text{L} \\
k_5 &= 25 - 17 = 8 \to \text{I} \\
k_6 &= 25 - 12 = 13 \to \text{N}
\end{aligned}
$$

The pattern repeats: $k_7 = 12 - 0 = 12$ (M), $k_8 = 23 - 19 = 4$ (E), etc.

Therefore, the keyword is **MERLIN**.

ii. What is the keyword if the plaintext is APPROPRIATE?

Ciphertext	U	R	W	Z	Z	Z	M	X	Z	Z	V
Ciphertext Position	20	17	22	25	25	25	12	23	25	25	21
Plaintext	A	P	P	R	O	P	R	I	A	T	E
Plaintext Position	0	15	15	17	14	15	17	8	0	19	4

Finding keyword positions by subtraction:

$$
\begin{aligned}
k_1 &= 20 - 0 = 20 \to \text{U} \\
k_2 &= 17 - 15 = 2 \to \text{C} \\
k_3 &= 22 - 15 = 7 \to \text{H} \\
k_4 &= 25 - 17 = 8 \to \text{I} \\
k_5 &= 25 - 14 = 11 \to \text{L} \\
k_6 &= 25 - 15 = 10 \to \text{K} \\
k_7 &= 12 - 17 = -5 \equiv 21 \pmod{26} \to \text{V} \\
k_8 &= 23 - 8 = 15 \to \text{P} \\
k_9 &= 25 - 0 = 25 \to \text{Z} \\
k_{10} &= 25 - 19 = 6 \to \text{G} \\
k_{11} &= 21 - 4 = 17 \to \text{R}
\end{aligned}
$$

The keyword is **UCHILKVPZGR**.

iii. How many possible plaintexts are there for this ciphertext? What percentage of these are a single English word?

Since the ciphertext URWZZZMXZZV has 11 letters, and each letter can be any of 26 letters in the plaintext (determined by the choice of keyword), there are 26^{11} possible plaintexts.

Calculating: $26^{11} = 3,670,344,486,987,776$

Therefore, there are **3,670,344,486,987,776** possible plaintexts for this ciphertext.

To find how many are English words, we can check a dictionary:

```
grep -E '^.{11}$' /usr/share/dict/words | wc -l
8845
```

On Ubuntu, this gives us 8,845 eleven-letter English words out of 3,670,344,486,987,776 total possible plaintexts.

Percentage: $\frac{8,845}{3,670,344,486,987,776} \times 100\% \approx 0.000000000241\%$

3. Suppose you have access to two encryption algorithms, called DUV-56 and DUV-69:

- DUV-56 has a 56 bit key
- DUV-69 has a 69 bit key

Suppose you have sufficient computing power to use an exhaustive key search to find the key of DUV-56 in 24 hours.

a) Assuming the two algorithms have a similar computational complexity, approximately how many days would you expect to take to find the key of DUV-69 using an exhaustive key search?

First, find the time to test a single key:

DUV-56 has 2^{56} possible keys.

Time to test all DUV-56 keys = 24 hours

Time per key = $\frac{24 \text{ hours}}{2^{56}}$

Now for DUV-69:

DUV-69 has 2^{69} possible keys.

Time for DUV-69 = $2^{69} \times \frac{24 \text{ hours}}{2^{56}}$

Simplifying:

$$\begin{aligned}
\text{Time for DUV-69} &= \frac{2^{69} \times 24}{2^{56}} \text{ hours} \\
&= 2^{69-56} \times 24 \text{ hours} \\
&= 2^{13} \times 24 \text{ hours} \\
&= 8192 \times 24 \text{ hours} \\
&= 196,608 \text{ hours} \\
&= \frac{196,608}{24} \text{ days} \\
&= 8,192 \text{ days}
\end{aligned}$$

Therefore, it would take approximately **8,192 days** to find the key of DUV-69.

b) Suppose that DUV-69 has been designed so that it can be run in two separate stages, so that it is possible to conduct an exhaustive key search for the first 56 bits of a DUV-69 key, followed by a separate exhaustive key search for the last 13 bits. Approximately how many days would you now expect to take to find the key of DUV-69 using an exhaustive key search?

With the two-stage approach:

Stage 1: Test 2^{56} possible keys for the first 56 bits = 24 hours (given)

Stage 2: Test 2^{13} possible keys for the last 13 bits

From part a, time per key = $\frac{24 \text{ hours}}{2^{56}}$

Time for Stage 2 $= 2^{13} \times \frac{24 \text{ hours}}{2^{56}}$

Calculating Stage 2:

$$\begin{aligned}\text{Time for Stage 2} &= \frac{2^{13} \times 24}{2^{56}} \text{ hours} \\ &= 2^{13-56} \times 24 \text{ hours} \\ &= 2^{-43} \times 24 \text{ hours} \\ &= \frac{24}{2^{43}} \text{ hours} \\ &= \frac{24}{8,796,093,022,208} \text{ hours} \\ &\approx 2.73 \times 10^{-12} \text{ hours (negligibly small)}\end{aligned}$$

Total time = 24 hours + negligible time $\approx$ 24 hours = 1 day

Therefore, it would take approximately **1 day** to find the key of DUV-69 using the two-stage approach.

4. a) Compute the following, and show your working out.

i. Exactly how many decimal digits are needed to write the number 10^{20}?

$10^{20} = 100,000,000,000,000,000,000$

Counting the digits: 1 followed by 20 zeros = **21 digits**

ii. Approximately how many binary digits (bits) are needed to write the number 10^{20}?

Using the approximation $2^{3.3} \approx 10$:

$$\begin{aligned}10^{20} &\approx (2^{3.3})^{20} \\ &= 2^{3.3 \times 20} \\ &= 2^{66}\end{aligned}$$

Therefore, approximately **67 bits** are needed (66 bits can represent up to $2^{66} - 1$, so we need 67 bits to represent 2^{66}).

iii. Exactly how many binary digits (bits) are needed to write the number 2^{20}?

2^{20} in binary is 1 followed by 20 zeros: $1\underbrace{00...00}_{\text{20 zeros}}$

Therefore, exactly **21 bits** are needed.

iv. Approximately how many decimal digits are needed to write the number 2^{20}?

Using the approximation $2^{3.3} \approx 10$:

$$\begin{aligned} 2^{20} &= 2^{3.3\times 6.06\ldots} \\ &\approx 2^{3.3\times 6} \\ &= (2^{3.3})^6 \\ &\approx 10^6 \\ &= 1,000,000 \end{aligned}$$

Therefore, approximately **7 decimal digits** are needed.

v. Summarise your answer using a table.

	10^{20}	2^{20}
Number of decimal digits	21	7 (approx)
Number of binary digits	67 (approx)	21

b) For each of the following, find the approximate number. Work out the approximate number of decimal and binary digits needed to represent the number.

Part	Number	Decimal digits	Binary digits
A	8,000,000,000	10	33
B	10^{11}[α]	12	$\approx 10^{11} = (2^{3.3})^{11} \approx 2^{36.3} \to 37$
C	10^{24}[β]	25	$\approx 10^{24} = (2^{3.3})^{24} \approx 2^{79.2} \to 80$
D	10^{7}[γ]	8	$\approx 10^{7} = (2^{3.3})^{7} \approx 2^{23.1} \to 24$
E	10^{80}[δ]	81	$\approx 10^{80} = (2^{3.3})^{80} \approx 2^{264} \to 265$
F	31,536,000	8	$\approx 10^{7} = (2^{3.3})^{7} \approx 2^{23.1} \to 24$
G	2^{64}	$2^{64} = (2^{3.3})^{19.4} \approx 10^{19.4} \to 20$	65
H	2^{128}	$2^{128} = (2^{3.3})^{38.8} \approx 10^{39} \to 39$	129
I	2^{256}	$2^{256} = (2^{3.3})^{77.6} \approx 10^{78} \to 78$	257

[α] Stars in the Milky Way (NASA)
[β] Stars in the Universe (ESA)
[γ] Estimated insect species on Earth (Smithsonian)
[δ] Atoms in the observable universe (Wikipedia)

5. Suppose we have a symmetric key algorithm with encryption rule $E_k(x)$, and we want to increase the security. An obvious approach is to try a 'double encryption'.

a) Suppose we encrypt a plaintext by first using the encryption rule E_{k_1}, then using E_{k_2} on the result, that is, use the encryption rule $E(x) = E_{k_2}(E_{k_1}(x))$.

Show that there is a single encryption rule $E_{k_3}(x) = a_3x + b_3 \pmod{26}$ which performs exactly the same encryption.

Solution:

Starting with the two encryption rules:

$$E_{k_1}(x) = a_1x + b_1 \pmod{26}$$
$$E_{k_2}(x) = a_2x + b_2 \pmod{26}$$

Now we compose them by applying E_{k_2} to the output of E_{k_1}:

$$\begin{aligned} E(x) &= E_{k_2}(E_{k_1}(x)) \\ &= E_{k_2}(a_1x + b_1) \\ &= a_2(a_1x + b_1) + b_2 \pmod{26} \\ &= a_2a_1x + a_2b_1 + b_2 \pmod{26} \\ &= (a_2a_1)x + (a_2b_1 + b_2) \pmod{26} \end{aligned}$$

This is in the form $a_3x + b_3 \pmod{26}$ where:

$$a_3 = a_2a_1 \pmod{26}$$
$$b_3 = a_2b_1 + b_2 \pmod{26}$$

Therefore, the double encryption is equivalent to a single affine encryption with key $k_3 = (a_3, b_3)$.

b) Find the values for $a_3, b_3 \in \mathbb{Z}_{26}$ when $E_{k_1}(x) = 11x+3 \pmod{26}$ and $E_{k_2}(x) = 5x + 15 \pmod{26}$.

From the given encryption rules: $a_1 = 11$, $b_1 = 3$, $a_2 = 5$, $b_2 = 15$

Using the formulas from part (a):

$$a_3 = a_2a_1 = 5 \cdot 11 = 55 = 2 \cdot 26 + 3 \equiv 3 \pmod{26}$$
$$b_3 = a_2b_1 + b_2 = 5 \cdot 3 + 15 = 30 = 1 \cdot 26 + 4 \equiv 4 \pmod{26}$$

Therefore, $E_{k_3}(x) = 3x + 4 \pmod{26}$.

c) Check your solution to part b by:

i. Encrypt the plaintext OK first using E_{k_1} and then encrypt the result using E_{k_2}

ii. Encrypt the plaintext OK using E_{k_3}

Converting OK to numbers: O = 14, K = 10

i. First apply $E_{k_1}(x) = 11x + 3 \pmod{26}$:

$$E_{k_1}(14) = 11 \cdot 14 + 3 = 157 = 6 \cdot 26 + 1 \equiv 1 \pmod{26} \to \text{B}$$
$$E_{k_1}(10) = 11 \cdot 10 + 3 = 113 = 4 \cdot 26 + 9 \equiv 9 \pmod{26} \to \text{J}$$

Then apply $E_{k_2}(x) = 5x + 15 \pmod{26}$ to BJ (positions 1, 9):

$$E_{k_2}(1) = 5 \cdot 1 + 15 = 20 \pmod{26} \to \text{U}$$
$$E_{k_2}(9) = 5 \cdot 9 + 15 = 60 = 2 \cdot 26 + 8 \equiv 8 \pmod{26} \to \text{I}$$

Result: **UI**

ii. Apply $E_{k_3}(x) = 3x + 4 \pmod{26}$:

$$E_{k_3}(14) = 3 \cdot 14 + 4 = 46 = 1 \cdot 26 + 20 \equiv 20 \pmod{26} \to \text{U}$$
$$E_{k_3}(10) = 3 \cdot 10 + 4 = 34 = 1 \cdot 26 + 8 \equiv 8 \pmod{26} \to \text{I}$$

Result: **UI**

Both methods give the same ciphertext, confirming our solution.

d) Suppose an exhaustive key-search attack is applied to a double-encrypted affine ciphertext, is the effective key space increased? (Explain your answer.)

No, the effective key space is not increased.

As shown in part (a), double encryption with two affine ciphers is equivalent to a single affine cipher with parameters (a_3, b_3) where $a_3 = a_2 a_1 \pmod{26}$ and $b_3 = a_2 b_1 + b_2 \pmod{26}$.

Since the result can always be expressed as a single affine encryption, an attacker only needs to try the same $12 \times 26 = 312$ possible keys to break the cipher, not 312^2 keys. Therefore, double encryption provides no additional security for the affine cipher.

— End of Assignment 1 —