THE UNIVERSITY OF
SYDNEY

Assignment 2

INFO3616: Principles of Security and Security Eng

510415022

Problem 1. Breaking the Vigenère Cipher (25 marks)

- **Key length:** 17
- **Key:** `XVFERTQXSWIFMZQAB`
- **Final plaintext:** NO MATTER HOW MANY TIMES YOU LEFT EARTH, DR. HEYWOOD FLOYD TOLD HIMSELF, THE EXCITEMENT NEVER REALLY PALLED. HE HAD BEEN TO MARS ONCE, TO THE MOON THREE TIMES, AND TO THE VARIOUS SPACE STATIONS MORE OFTEN THAN HE COULD REMEMBER. YET AS THE MOMENT OF TAKEOFF APPROACHED, HE WAS CONSCIOUS OF A RISING TENSION, A FEELING OF WONDER AND AWE - YES; AND OF NERVOUSNESS - WHICH PUT HIM ON THE SAME LEVEL AS ANY EARTHLUBBER ABOUT TO RECEIVE HIS FIRST BAPTISM OF SPACE. THE JET THAT HAD RUSHED HIM HERE FROM WASHINGTON, AFTER THAT MIDNIGHT BRIEFING WITH THE PRESIDENT, WAS NOW DROPPING DOWN TOWARD ONE OF THE MOST FAMILIAR, YET MOST EXCITING, LANDSCAPES IN ALL THE WORLD. THERE LAY THE FIRST TWO GENERATIONS OF THE SPACE AGE, SPANNING TWENTY MILES OF THE FLORIDA COAST TO THE SOUTH, OUTLINED BY WINKING RED WARNING LIGHTS, WERE THE GIANT GANTRIES OF THE SATURNS AND NEPTUNES, THAT HAD SET MEN ON THE PATH TO THE PLANETS, AND HAD NOW PASSED INTO HISTORY. CYBERSECURITY ENGINEERING ASSIGNMENT. NEAR THE HORIZON, A GLEAMING SILVER TOWER BATHED IN FLOODLIGHTS, STOOD THE LAST OF THE SATURN V'S, FOR ALMOST TWENTY YEARS A NATIONAL MONUMENT AND PLACE OF PILGRIMAGE. NOT FAR AWAY, LOOMING AGAINST THE SKY LIKE A MAN-MADE MOUNTAIN, WAS THE INCREDIBLE BULK OF THE VEHICLE ASSEMBLY BUILDING, STILL THE LARGEST SINGLE STRUCTURE ON EARTH. BUT THESE THINGS NOW BELONGED TO THE PAST, AND HE WAS FLYING TOWARD THE FUTURE. AS THEY BANKED, DR. FLOYD COULD SEE BELOW HIM A MAZE OF BUILDINGS, THEN A GREAT AIRSTRIP, THEN A BROAD, DEAD-STRAIGHT SCAR ACROSS THE FLAT FLORIDA LANDSCAPE - THE MULTIPLE RAILS OF A GIANT LAUNCH-LUG TRACK. AT ITS END, SURROUNDED BY VEHICLES AND GANTRIES, A SPACEPLANE LAY GLEAMING IN A POOL OF LIGHT, BEING PREPARED FOR ITS LEAP TO THE STARS. IN A SUDDEN FAILURE OF PERSPECTIVE, BROUGHT ON BY HIS SWIFT CHANGES OF SPEED AND HEIGHT, IT SEEMED TO FLOYD THAT HE WAS LOOKING DOWN ON A SMALL SILVER MOTH, CAUGHT IN THE BEAM OF A FLASHLIGHT.

Approach

My hacking script follows the below approach:

1. **Normalise the ciphertext:** I remove non-letters and convert the text to uppercase so that punctuation doesn't interfere with the rest of the script.

2. **Estimate the key length using the Kasiski examination:**
    - Find all repeated substrings of length 3-5 and record the spacings between their occurrences.
    - For each spacing, collect factors between 2 and the maximum key length of 20.
    - Tally factor frequencies and sort them to produce a ranked list of likely key lengths $k_1, k_2, \ldots, k_n$.
3. **Split by position:** For each candidate key length k, I build k strings containing all letters from the ciphertext spaced out by $1, 2, \ldots, k$ letters, as these strings will contain characters encrypted by the same Caesar shift.
4. **Per-position Caesar cracking:** For each stream and each possible subkey A-Z, I "decrypt" the stream with that single letter and score it with a frequency analysis function.
    - The frequency analysis function returns a score of up to 12 points, with +1 point for each of the six most common English letters that appears among the stream's six most frequent, and +1 for each of the six rarest that appear among the stream's six least frequent.
    - I keep the top 3 candidate letters per key position.
5. **Combine candidates and pick the best key:** I take the Cartesian product of the top letters for each position to form full candidate keys, decrypt the entire ciphertext with each, and rescore the result. If a candidate passes a string English dictionary check, I preview it to the terminal and allow the runner to accept or reject.

You can test my script by running:

```
python hack.py
```

Reject the first two results by pressing the Return key on your keyboard. The third result will be the correct key. This entire process should take < 15 seconds.

Problem 2. AES Calculation **(25 marks)**

Given:

- **Plaintext:** `0E0D0C0B 0A090807 06050403 0201000F`
- **First round key:** `03030303 03030303 03030303 03030303`

we are required to manually calculate the initial steps of the AES Rijndael cipher.

i. We first convert the plaintext and round key into block form using column-major order, which becomes our initial State.

$$\texttt{State}_1 = \begin{bmatrix} \texttt{0E} & \texttt{0A} & \texttt{06} & \texttt{02} \\ \texttt{0D} & \texttt{09} & \texttt{05} & \texttt{01} \\ \texttt{0C} & \texttt{08} & \texttt{04} & \texttt{00} \\ \texttt{0B} & \texttt{07} & \texttt{03} & \texttt{0F} \end{bmatrix} \tag{1}$$

$$\texttt{RoundKey} = \begin{bmatrix} \texttt{03} & \texttt{03} & \texttt{03} & \texttt{03} \\ \texttt{03} & \texttt{03} & \texttt{03} & \texttt{03} \\ \texttt{03} & \texttt{03} & \texttt{03} & \texttt{03} \\ \texttt{03} & \texttt{03} & \texttt{03} & \texttt{03} \end{bmatrix} \tag{2}$$

ii. Then, we perform the `AddRoundKey` operation, which simply involves an element-wise XOR between the `State` and `RoundKey` blocks.

$$\texttt{State}_2 = \texttt{AddRoundKey}(\texttt{State}_1) = \texttt{State}_1 \oplus \texttt{RoundKey} \tag{3}$$

$$= \begin{bmatrix} \texttt{0E} & \texttt{0A} & \texttt{06} & \texttt{02} \\ \texttt{0D} & \texttt{09} & \texttt{05} & \texttt{01} \\ \texttt{0C} & \texttt{08} & \texttt{04} & \texttt{00} \\ \texttt{0B} & \texttt{07} & \texttt{03} & \texttt{0F} \end{bmatrix} \oplus \begin{bmatrix} \texttt{03} & \texttt{03} & \texttt{03} & \texttt{03} \\ \texttt{03} & \texttt{03} & \texttt{03} & \texttt{03} \\ \texttt{03} & \texttt{03} & \texttt{03} & \texttt{03} \\ \texttt{03} & \texttt{03} & \texttt{03} & \texttt{03} \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} \texttt{0E} \oplus \texttt{03} & \dots & \dots & \texttt{02} \oplus \texttt{03} \\ \texttt{0D} \oplus \texttt{03} & \ddots & & \texttt{01} \oplus \texttt{03} \\ \texttt{0C} \oplus \texttt{03} & & \ddots & \texttt{00} \oplus \texttt{03} \\ \texttt{0B} \oplus \texttt{03} & \dots & \dots & \texttt{0F} \oplus \texttt{03} \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} \texttt{0D} & \texttt{09} & \texttt{05} & \texttt{01} \\ \texttt{0E} & \texttt{0A} & \texttt{06} & \texttt{02} \\ \texttt{0F} & \texttt{0B} & \texttt{07} & \texttt{03} \\ \texttt{08} & \texttt{04} & \texttt{00} & \texttt{0C} \end{bmatrix} \tag{6}$$

For example, $\texttt{0E} \oplus \texttt{03} = \texttt{0b00001110} \oplus \texttt{0b00000011} = \texttt{0b00001101} = \texttt{0x0D}$.

iii. The `SubBytes` operation uses a pre-constructed mapping called an *S-box* to perform a byte-by-byte substitution of the block (see Table 6.2 of the Stallings textbook). It is just a simple table lookup: for each entry in $\texttt{State}_2$, the most-significant byte is taken as the row and the least-significant byte is taken as the column.

For example, for the first entry of $\texttt{State}_2$ `0D`, `0` represents the row of the S-box while `D` represents the column. After looking up the corresponding value in the table, we get `D7`.

$$\texttt{State}_3 = \texttt{SubBytes}(\texttt{State}_2) \tag{7}$$

$$= \begin{bmatrix} \texttt{D7} & \texttt{01} & \texttt{6B} & \texttt{7C} \\ \texttt{AB} & \texttt{67} & \texttt{6F} & \texttt{77} \\ \texttt{76} & \texttt{2B} & \texttt{C5} & \texttt{7B} \\ \texttt{30} & \texttt{F2} & \texttt{63} & \texttt{FE} \end{bmatrix} \tag{8}$$

iv. We now perform the `ShiftRows` operation on $\texttt{State}_3$. For the first row, nothing is altered. The second row undergoes a 1-byte circular left shift, $\lll 1$. The third row undergoes a 2-byte circular left shift ($\lll 2$) and the fourth row undergoes a 3-byte circular left shift ($\lll 3$).

$$\texttt{State}_4 = \texttt{ShiftRows}(\texttt{State}_3) \tag{9}$$

$$= \begin{bmatrix} \texttt{D7} & \texttt{01} & \texttt{6B} & \texttt{7C} \\ \texttt{67} & \texttt{6F} & \texttt{77} & \texttt{AB} \\ \texttt{C5} & \texttt{7B} & \texttt{76} & \texttt{2B} \\ \texttt{FE} & \texttt{30} & \texttt{F2} & \texttt{63} \end{bmatrix} \tag{10}$$

v. Finally, the $\texttt{State}_4$ block undergoes the `MixColumns` operation. The `MixColumns` operation, along with `ShiftRows`, is the Rijndael cipher's main mechanism for achieving diffusion in the ciphertext. It's goal is to shuffle each column individually such that, in tandem with `ShiftRows`, each value in the previous state is well-randomised throughout the next block. It achieves this by performing the following matrix multiplication:

$$\texttt{MixColumns}(S) := \begin{bmatrix} \texttt{02} & \texttt{03} & \texttt{01} & \texttt{01} \\ \texttt{01} & \texttt{02} & \texttt{03} & \texttt{01} \\ \texttt{01} & \texttt{01} & \texttt{02} & \texttt{03} \\ \texttt{03} & \texttt{01} & \texttt{01} & \texttt{02} \end{bmatrix} \begin{bmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{bmatrix} \tag{11}$$

Note that this is not an ordinary matrix multiplication. AES arithmetic during `MixColumns` is done in the finite field $\mathrm{GF}(2^8)$ to allow every nonzero byte to have a multiplicative inverse, as well as to contain the math within 1 byte. This has several consequences for our calculations.

Firstly, we can now treat bytes as polynomials: A byte $a = a_7 a_6 \cdots a_0$ (bits $a_i \in \{0, 1\}$) is identified with the polynomial

$$a(x) = a_7 x^7 + a_6 x^6 + \cdots + a_1 x + a_0. \tag{12}$$

For example, $\texttt{D7} = \texttt{11010111}_2$ corresponds to $x^7+x^6+x^4+x^2+x+1$. One of the reasons we do this is so that we can define the operation of multiplication between two bytes: it is simply the multiplication of their corresponding polynomials. Note that all arithmetic done here is computed modulo 2 because all polynomial coefficients must be in $\{0, 1\}$ by definition of the finite field.

However, in certain cases, this multiplication results in a polynomial with degree > 7 which represents a number greater than 8 bits. To remediate this, we introduce the *reducing polynomial*:

$$m(x) = x^8 + x^4 + x^3 + x + 1 = \texttt{100011011}_2 = \texttt{11B}. \tag{13}$$

Whenever a product produces a polynomial of degree $\geq x^8$, we reduce *modulo* $m(x)$. Operationally, this is just the XOR of $m(x)$ and the polynomial.

Additionally, over GF(2^8), we define addition as the bitwise XOR operation:

$$a(x) + b(x) \iff a \oplus b \tag{14}$$

Let us work through the calculation for the first column:

$$\begin{bmatrix} \texttt{02} & \texttt{03} & \texttt{01} & \texttt{01} \\ \texttt{01} & \texttt{02} & \texttt{03} & \texttt{01} \\ \texttt{01} & \texttt{01} & \texttt{02} & \texttt{03} \\ \texttt{03} & \texttt{01} & \texttt{01} & \texttt{02} \end{bmatrix} \begin{bmatrix} \texttt{D7} \\ \texttt{67} \\ \texttt{C5} \\ \texttt{FE} \end{bmatrix} = \begin{bmatrix} \texttt{02} \cdot \texttt{D7} \oplus \texttt{03} \cdot \texttt{67} \oplus \texttt{01} \cdot \texttt{C5} \oplus \texttt{01} \cdot \texttt{FE} \\ \texttt{01} \cdot \texttt{D7} \oplus \texttt{02} \cdot \texttt{67} \oplus \texttt{03} \cdot \texttt{C5} \oplus \texttt{01} \cdot \texttt{FE} \\ \texttt{01} \cdot \texttt{D7} \oplus \texttt{01} \cdot \texttt{67} \oplus \texttt{02} \cdot \texttt{C5} \oplus \texttt{03} \cdot \texttt{FE} \\ \texttt{03} \cdot \texttt{D7} \oplus \texttt{01} \cdot \texttt{67} \oplus \texttt{01} \cdot \texttt{C5} \oplus \texttt{02} \cdot \texttt{FE} \end{bmatrix} \tag{15}$$

Row 4

We begin with the fourth row of the first column to validate our approach (as the correct answer of 27 has already been provided for us). We explain each step thoroughly for this row; future calculations will be more terse.

We are required to solve:

$$\texttt{03} \cdot \texttt{D7} \oplus \texttt{01} \cdot \texttt{67} \oplus \texttt{01} \cdot \texttt{C5} \oplus \texttt{02} \cdot \texttt{FE} \tag{16}$$

Let us first focus on the 4 multiplications separately before ultimately XORing the results.

$\texttt{03} \cdot \texttt{D7}$

$$\texttt{0x03} = \texttt{0b00000011} = x + 1 \tag{17}$$

$$\texttt{0xD7} = \texttt{0b11010111} = x^7 + x^6 + x^4 + x^2 + x + 1 \tag{18}$$

Multiplying the two polynomials together in GF(2^8):

$$\texttt{0x03} \cdot \texttt{0xD7} = (x+1)(x^7+x^6+x^4+x^2+x+1) \mod 2 \tag{19}$$
$$= x^8+2x^7+x^6+x^5+x^4+x^3+2x^2+2x+1 \mod 2 \tag{20}$$
$$= x^8+x^6+x^5+x^4+x^3+1 \tag{21}$$
$$= \texttt{0b101111001} \tag{22}$$

As this result is greater than 7 bits, we will reduce it using $m(x) = x^8+x^4+x^3+x+1 = \texttt{0b100011011}$:

$$\texttt{0b101111001} \oplus \texttt{0b100011011} = \texttt{0b001100010} = \texttt{0x62} \tag{23}$$

$\texttt{01} \cdot \texttt{67}$

$$\texttt{0x01} = \texttt{0b00000001} = 1 \tag{24}$$
$$\texttt{0x67} = \texttt{0b01100111} = x^6+x^5+x^2+x+1 \tag{25}$$
$$\texttt{0x01} \cdot \texttt{0x67} = 1 \cdot (x^6+x^5+x^2+x+1) \mod 2 \tag{26}$$
$$= x^6+x^5+x^2+x+1 \tag{27}$$
$$= \texttt{0b01100111} = \texttt{0x67} \tag{28}$$

$\texttt{01} \cdot \texttt{C5}$

$$\texttt{0x01} = \texttt{0b00000001} = 1 \tag{29}$$
$$\texttt{0xC5} = \texttt{0b11000101} = x^7+x^6+x^2+1 \tag{30}$$
$$\texttt{0x01} \cdot \texttt{0xC5} = 1 \cdot (x^7+x^6+x^2+1) \mod 2 \tag{31}$$
$$= x^7+x^6+x^2+1 \tag{32}$$
$$= \texttt{0b11000101} = \texttt{0xC5} \tag{33}$$

$\implies$ In general, $\texttt{0x01} \cdot \texttt{0xYY} = \texttt{0xYY}$.

$\texttt{02} \cdot \texttt{FE}$

$$\texttt{0x02} = \texttt{0b00000010} = x \tag{34}$$
$$\texttt{0xFE} = \texttt{0b11111110} = x^7+x^6+x^5+x^4+x^3+x^2+x \tag{35}$$

Multiplying in GF(2):

$$\texttt{0x02} \cdot \texttt{0xFE} = x\,(x^7+x^6+x^5+x^4+x^3+x^2+x) \mod 2 \tag{36}$$
$$= x^8+x^7+x^6+x^5+x^4+x^3+x^2 \tag{37}$$
$$= \texttt{0b111111100} \tag{38}$$

Reduce modulo $m(x) = \texttt{0b100011011}$:

$$\texttt{0b111111100} \oplus \texttt{0b100011013} = \texttt{0b011100111} = \texttt{0xE7} \tag{39}$$

Combine

Combining the four results, we get:

$$\texttt{03} \cdot \texttt{D7} \oplus \texttt{01} \cdot \texttt{67} \oplus \texttt{01} \cdot \texttt{C5} \oplus \texttt{02} \cdot \texttt{FE} = \texttt{0x62} \oplus \texttt{0x67} \oplus \texttt{0xC5} \oplus \texttt{0xE7} \tag{40}$$

$$= \texttt{0x27} \tag{41}$$

as required. □

We now repeat the calculations above for the first three rows.

Row 1

$$\texttt{02} \cdot \texttt{D7} \oplus \texttt{03} \cdot \texttt{67} \oplus \texttt{01} \cdot \texttt{C5} \oplus \texttt{01} \cdot \texttt{FE} \tag{42}$$

$\texttt{02} \cdot \texttt{D7}$

$$\texttt{0x02} = x \tag{43}$$

$$\texttt{0xD7} = x^7 + x^6 + x^4 + x^2 + x + 1 \tag{44}$$

$$\implies \texttt{0x02} \cdot \texttt{0xD7} = x(x^7 + x^6 + x^4 + x^2 + x + 1) \tag{45}$$

$$= x^8 + x^7 + x^5 + x^3 + x^2 + x \tag{46}$$

$$= \texttt{0b110101110} \tag{47}$$

Reducing using $m(x)$:

$$\texttt{0b110101110} \oplus \texttt{0b100011011} = \texttt{0b010110101} = \texttt{0xB5} \tag{48}$$

$\texttt{03} \cdot \texttt{67}$

$$\texttt{0x03} = x + 1 \tag{49}$$

$$\texttt{0x67} = x^6 + x^5 + x^2 + x + 1 \tag{50}$$

$$\implies \texttt{0x03} \cdot \texttt{0x67} = (x + 1)(x^6 + x^5 + x^2 + x + 1) \tag{51}$$

$$= x^7 + x^6 + x^3 + x^2 + x \ + \ x^6 + x^5 + x^2 + x + 1 \tag{52}$$

$$= x^7 + x^5 + x^3 + 1 \tag{53}$$

$$= \texttt{0b10101001} \tag{54}$$

$\texttt{01} \cdot \texttt{C5}$

From above, $\texttt{01} \cdot \texttt{C5} = \texttt{0xC5}$.

$\texttt{01} \cdot \texttt{FE}$

From above, $\texttt{01} \cdot \texttt{FE} = \texttt{0xFE}$.

Combine

$$\texttt{0xB5} \oplus \texttt{0xA9} \oplus \texttt{0xC5} \oplus \texttt{0xFE} = \texttt{0x27} \tag{55}$$

Row 2

$$\texttt{01} \cdot \texttt{D7} \oplus \texttt{02} \cdot \texttt{67} \oplus \texttt{03} \cdot \texttt{C5} \oplus \texttt{01} \cdot \texttt{FE} \tag{56}$$

$\texttt{01} \cdot \texttt{D7}$

From above, $\texttt{01} \cdot \texttt{D7} = \texttt{0xD7}$.

$\texttt{02} \cdot \texttt{67}$

$$\begin{aligned}
\texttt{0x02} &= x && (57)\\
\texttt{0x67} &= x^6 + x^5 + x^2 + x + 1 && (58)\\
\implies \texttt{0x02} \cdot \texttt{0x67} &= x(x^6 + x^5 + x^2 + x + 1) && (59)\\
&= x^7 + x^6 + x^3 + x^2 + x && (60)\\
&= \texttt{0b11001110} = \texttt{0xCE} && (61)
\end{aligned}$$

$\texttt{03} \cdot \texttt{C5}$

$$\begin{aligned}
\texttt{0x03} &= x + 1 && (62)\\
\texttt{0xC5} &= x^7 + x^6 + x^2 + 1 && (63)\\
\implies \texttt{0x03} \cdot \texttt{0xC5} &= (x + 1)(x^7 + x^6 + x^2 + 1) && (64)\\
&= x^8 + x^6 + x^3 + x^2 + x + 1 && (65)\\
&= \texttt{0b101001111} && (66)
\end{aligned}$$

Reducing using $m(x)$:

$$\texttt{0b101001111} \oplus \texttt{0b100011011} = \texttt{0b001010100} = \texttt{0x54} \tag{67}$$

$\texttt{01} \cdot \texttt{FE}$

From above, $\texttt{01} \cdot \texttt{FE} = \texttt{0xFE}$.

Combine

$$\texttt{0xD7} \oplus \texttt{0xCE} \oplus \texttt{0x54} \oplus \texttt{0xFE} = \texttt{0xB3} \tag{68}$$

Row 3

$$\texttt{01} \cdot \texttt{D7} \oplus \texttt{01} \cdot \texttt{67} \oplus \texttt{02} \cdot \texttt{C5} \oplus \texttt{03} \cdot \texttt{FE} \tag{69}$$

$\texttt{01} \cdot \texttt{D7}$

From above, $\texttt{01} \cdot \texttt{D7} = \texttt{0xD7}$.

$\texttt{01} \cdot \texttt{67}$

From above, $\texttt{01} \cdot \texttt{67} = \texttt{0x67}$.

$\texttt{02} \cdot \texttt{C5}$

$$\begin{aligned}
\texttt{0x02} &= x && (70)\\
\texttt{0xC5} &= x^7 + x^6 + x^2 + 1 && (71)\\
\implies \texttt{0x02} \cdot \texttt{0xC5} &= x(x^7 + x^6 + x^2 + 1) && (72)\\
&= x^8 + x^7 + x^3 + x && (73)\\
&= \texttt{0b110001010} && (74)
\end{aligned}$$

Reducing using $m(x)$:

$$\texttt{0b110001010} \oplus \texttt{0b100011011} = \texttt{0b010010001} = \texttt{0x91} \tag{75}$$

$\texttt{03} \cdot \texttt{FE}$

$$\begin{aligned}
\texttt{0x03} &= x + 1 && (76)\\
\texttt{0xFE} &= x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x && (77)\\
\implies \texttt{0x03} \cdot \texttt{0xFE} &= (x+1)(x^7 + \cdots + x) && (78)\\
&= x^8 + x && (79)
\end{aligned}$$

Reducing using $m(x)$:

$$\texttt{0b100000000} \oplus \texttt{0b100011011} = \texttt{0b00011001} = \texttt{0x19} \tag{80}$$

Combine

$$\texttt{0xD7} \oplus \texttt{0x67} \oplus \texttt{0x91} \oplus \texttt{0x19} = \texttt{0x38} \tag{81}$$

Thus, the final state of this calculation is:

$$\begin{bmatrix} \texttt{27} & \texttt{NA} & \texttt{NA} & \texttt{NA} \\ \texttt{B3} & \texttt{NA} & \texttt{NA} & \texttt{NA} \\ \texttt{38} & \texttt{NA} & \texttt{NA} & \texttt{NA} \\ \texttt{27} & \texttt{NA} & \texttt{NA} & \texttt{NA} \end{bmatrix} \tag{82}$$

Problem 3. RSA Calculation **(25 marks)**

Parameters:

- $p = 4337$
- $q = 6481$
- $n = pq = 28108097$
- $\phi(n) = (p-1)(q-1) = 28097280$
- $e = 65537$

i. We will need to rely on the following theory to find the private key:

Definition 1. $u \equiv v \pmod{m} \implies u$ and v give the same remainder when divided by m.

Definition 2. If two numbers give the same remainder when divided by m, their difference must be a multiple of m.

$$u \equiv v \pmod{m} \implies u - v = km \quad k \in \mathrm{Z} \tag{83}$$

Now, the public key d must be the modular inverse of e:

$$ed = 1 \pmod{\phi(n)} \tag{84}$$

$$\implies ed - 1 = \phi(n) \cdot x \tag{85}$$

$$ed + \phi(n) \cdot x = 1 \tag{86}$$

This is now in the form of Bézout's identity $ax + by = \gcd(a, b)$ with $a = e = 65537$, $b = \phi(n) = 28097280$, and $\gcd(e, \phi(n)) = 1$:

$$65537d + 28097280x = 1 \tag{87}$$

Now that we have reformulated the problem into Bézout's identity, we can first run the Euclidean Algorithm for finding the GCD of e and $\phi(n)$ before backtracking in the *Extended* Euclidean Algorithm to find the modular inverse.

Step 1: Euclidean Algorithm

We first apply the Euclidean Algorithm to $e = 65537$ and $\phi(n) = 28097280$:

$$28097280 = 428(65537) + 47444 \tag{88}$$
$$65537 = 1(47444) + 18093 \tag{89}$$
$$47444 = 2(18093) + 11258 \tag{90}$$
$$18093 = 1(11258) + 6835 \tag{91}$$
$$11258 = 1(6835) + 4423 \tag{92}$$
$$6835 = 1(4423) + 2412 \tag{93}$$
$$4423 = 1(2412) + 2011 \tag{94}$$
$$2412 = 1(2011) + 401 \tag{95}$$
$$2011 = 5(401) + 6 \tag{96}$$
$$401 = 66(6) + 5 \tag{97}$$
$$6 = 1(5) + 1 \tag{98}$$

Step 2: Extended Euclidean Algorithm

We now work backwards through the remainders in order to express 1 as a linear combination of 65537 and 28097280. Starting from the last nonzero remainder in (98):

$$1 = 6 - 1 \cdot 5 \tag{99}$$

Substitute $5 = 401 - 66 \cdot 6$ from (97):

$$1 = 6 - \big(401 - 66 \cdot 6\big) = 67 \cdot 6 - 401 \tag{100}$$

Now substitute $6 = 2011 - 5 \cdot 401$ from (96):

$$1 = 67\big(2011 - 5 \cdot 401\big) - 401 = 67 \cdot 2011 - 336 \cdot 401 \tag{101}$$

Then substitute $401 = 2412 - 2011$ from (95):

$$1 = 67 \cdot 2011 - 336\big(2412 - 2011\big) = 403 \cdot 2011 - 336 \cdot 2412 \tag{102}$$

At this point the process is clear: each substitution replaces a remainder with the corresponding expression from the Euclidean Algorithm. However, this back-substitution is tedious to do by hand. To speed things up, we instead use the tabular form of the algorithm presented in the assignment, which automates this process.

The recurrence relations are:

$$x_i = x_{i-2} - q_i x_{i-1}, \qquad y_i = y_{i-2} - q_i y_{i-1}, \tag{103}$$

with initial values $x_{-1} = 1, y_{-1} = 0, x_0 = 0, y_0 = 1$.

i	q_i	r_i	x_i	y_i
-1	–	28097280	1	0
0	–	65537	0	1
1	428	47444	1	-428
2	1	18093	-1	429
3	2	11258	3	-1286
4	1	6835	-4	1715
5	1	4423	7	-3001
6	1	2412	-11	4716
7	1	2011	18	-7717
8	1	401	-29	12433
9	5	6	163	-69882
10	66	5	-10787	4624645
11	1	1	10950	-4694527
12	5	0	NA	NA

At $r_{11} = 1$, we obtain the Bézout identity

$$65537(-4694527) + 28097280(10950) = 1 \tag{104}$$

Step 3: Modular Inverse

Recalling the identity above,

$$ed = 1 \pmod{\phi(n)} \implies ed + \phi(n)x = 1 \tag{105}$$

we can match coefficients and see that

$$65537(-4694527) \equiv 1 \pmod{28097280} \tag{106}$$

Therefore the modular inverse is

$$d \equiv -4694527 \pmod{28097280} \tag{107}$$

Adding $\phi(n)$ to make d positive:

$$d = -4694527 + 28097280 = 23402753 \tag{108}$$

Thus, the private key exponent is d = 23402753.

ii. If Alice wants to send the message $X = \texttt{3}$ to Bob, she will encrypt the message using Bob's public key:

$$\begin{aligned} Y &= E[PU_b, X] && (109)\\ &= X^e \mod n && (110)\\ &= 3^{65537} \mod 28108097 && (111)\\ & && (112) \end{aligned}$$

To compute this large modular exponentiation, we can use exponentiation by squaring. Firstly, we write the exponent in binary:

$$65537 = 2^{16} + 1 = (10000000000000001)_2 \tag{113}$$

Then,

$$3^{65537} = 3^{2^{16}+1} = 3 \times 3^{2^{16}} \tag{114}$$

Now, let $a_k \equiv 3^{2^k} \pmod{28108097}$ with $a_0 = 3$. Then $a_{k+1} \equiv a_k^2 \pmod{28108097}$. We compute a few squarings explicitly:

$$\begin{aligned} a_0 &= 3\\ a_1 = a_0^2 &\equiv 3^2 \equiv 9 \pmod{28108097}\\ a_2 = a_1^2 &\equiv 9^2 \equiv 81 \pmod{28108097}\\ a_3 = a_2^2 &\equiv 81^2 \equiv 6561 \pmod{28108097}\\ a_4 = a_3^2 &\equiv 6561^2 \equiv 14938624 \pmod{28108097}\\ &\vdots\\ a_{16} = a_{15}^2 &\equiv 12910071 \pmod{28108097} \end{aligned} \tag{115}$$

Hence, to solve Equation 114, we simply compute

$$3 \times 12910071 \pmod{28108097} = 10622116 \pmod{28108097} \tag{116}$$

Using a modular exponentiation calculator, we confirm that the final answer is 10622116. Thus the message Alice sends to Bob is 10622116.

iii. Bob can decrypt Alice's message X using his own private key:

$$\begin{aligned} X &= D[PR_b, Y] && (117)\\ &= Y^d \bmod n && (118)\\ &= 10622116^{23402753} \mod 28108097 && (119)\\ &= 3 && (120) \end{aligned}$$

Problem 4. Message Authentication Codes (25 marks)

a) **Message Authentication Codes vs. Hashes (5 marks)**

i. Authenticated Encryption (AE) combines symmetric encryption and message authentication into one cryptographic system in order to simultaneously protect the confidentiality and authenticity of communications. AE typically uses a symmetric key to encrypt the plaintext and generate a tag to verify authenticity.

AE protects against eavesdropping (via encryption) and tampering (via authentication). Without AE, an attacker could intercept and modify a message, potentially causing unauthorised actions or data corruption.

ii. AE provides confidentiality and integrity for the plaintext. AEAD extends AE by allowed associated data (such as headers) to be authenticated but not encrypted. This is useful when some data needs to be sent unencrypted but still verified for integrity.

b) **Protocol Analysis (10 marks)**

Protocol X is defined as

$$y = E_{k_1}(x||H(k_2||x))$$

When Bob receives the message y, he will:

(a) Use the shared key k_1 to decrypt the message using AES decryption:

$$D_{k_1} = x||H(k_2||x) \tag{121}$$

(b) Split the output into plaintext x and the hash $H(k_2||x)$.

(c) Compute $H(k_2||x)$ using the shared key k_2 and the received x. He compares this with the received plaintext from step b). If they match, the message is authentic and untampered. If they don't match, Bob rejects the message as tampered or invalid.

Using this protocol, confidentiality is achieved because the plaintext x is encrypted with AES using k_1 before transmission. Hence, only Bob, who also has k_1, can decrypt y. This ensures the message is unreadable to eavesdroppers.

Integrity is also achieved because the hash is included in the encrypted message. After decryption, Bob can verify the hash using k_2. Since k_2 is only shared with Alice, an attacker cannot forge a valid hash. The encryption of the hash also prevents tampering during transmission.

Protocol Y is defined as

$$x, y = E_{PK}(H(x))$$

When Bob receives the plaintext x and the encrypted hash $y = E_{PK}(H(x))$, he will:

(a) Decrypt y using his private key SK: $D_{SK}(y) = H(x)$.

(b) Compute $H(x)$ using the received plaintext x.

(c) Compare the computed $H(x)$ with the decrypted hash from step a). If they match, the message is authentic and from someone with access to Bob's public key. If they don't match, Bob rejects the message as tampered or invalid.

Unlike Protocol X, Protocol Y does not achieve confidentiality. This is because the plaintext x is sent unencrypted, so anyone intercepting the message can read it. Only the hash is encrypted with Bob's public key.

However, this protocol does continue to achieve integrity. Because the hash is encrypted with Bob's public key, he can decrypt it using his private key and verify the hash against one he computed from the received x.

c) **Flawed Hash** **(10 marks)**

The hash function processes a message M divided into blocks $M_1, M_2, \ldots, M_N$ using DES without a secret key:

- Initialise: H_0 (arbitrary value).
- For each block: $H_i = H_{i-1} \oplus E(M_i, H_{i-1})$, where E is DES encryption.
- Final hash: $H = H_N$.

To alter the message without changing H_N, modify blocks M_i and M_{i+1}:

(a) Compute $H_i' = H_{i-1} \oplus E(M_i', H_{i-1})$.

(b) By the DES property, $E(M_i', H_{i-1}) = E(M_i, H_{i-1})' = Y'$, so $H_i' = H_{i-1} \oplus Y'$.

(c) Choose M_{i+1}' such that $H_{i+1}' = H_{i+1}$, where $H_{i+1} = H_i \oplus E(M_{i+1}, H_i)$. Set:

$$E(M_{i+1}', H_i') = H_i' \oplus H_i \oplus E(M_{i+1}, H_i)$$

Thus:

$$M_{i+1}' = D(H_i', H_i' \oplus H_i \oplus E(M_{i+1}, H_i))$$

where D is DES decryption.

The modified message $M_1, \ldots, M_i', M_{i+1}', \ldots, M_N$ produces the same hash H_N, as $H_{i+1}' = H_{i+1}$, and subsequent blocks use unchanged inputs. This breaks the hash function's integrity, allowing undetectable message alterations.