

**PSET 1 — 2023-01-27***Prof. Chakrabarti**Student: Amittai Siavava***Credit Statement**

I worked on these problems alone, with reference to class notes and the following books:

- (a) **Introduction to the Theory of Computation** by **Michael Sipser**.
- (b) **A Mathematical Introduction to Logic** by **Herbert Enderton**.

**Problem 1.**

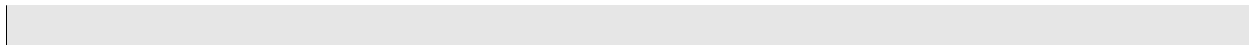
What is the message embedded in the following?

Dear George,  
Greetings to all at Oxford. Many thanks for **your**  
letter and for the Summer examination **package**.  
All Entry Forms and Fees Forms should be **ready**  
for final despatch to the syndicate by **Friday**  
20th or at the very latest, I'm told, by the **21st**.  
Admin has improved here, though there's **room**  
for improvement still; just give us all two or **three**  
more years and we'll really show you! **Please**  
don't let these wretched 16+ proposals **destroy**  
your basic O and A pattern. Certainly **this**  
sort of change, if implemented **immediately**,  
would bring chaos.

I thought it was unlikely that the text was actually encrypted (a strategy that did not work that well on problem 2). On this problem, I tried to look at the words in different positions in the text — every first word in a sentence, every last word, every  $n$ th word, in sentence  $n$ , etc.

Using this strategy, I deciphered this message (highlighted in red above):

**Your package ready Friday 21st, room three. Please destroy this immediately.**



**Problem 2.**

Encrypt the message 001100001010 using two rounds of SDES and (9 bit) key 111000101, as explained in lecture.

Show all your steps! *[Hint: After one round, the output is 001010010011.]*

First, let's define our permutation function:

**permute** (123456) = 12434356

And tables:

| $s_1$ | 1   | 2   | $s_2$ | 1   | 2   |
|-------|-----|-----|-------|-----|-----|
| 000   | 101 | 001 |       | 100 | 101 |
| 001   | 010 | 100 |       | 000 | 011 |
| 010   | 001 | 110 |       | 110 | 000 |
| 011   | 110 | 010 |       | 101 | 111 |
| 100   | 011 | 000 |       | 111 | 110 |
| 101   | 100 | 111 |       | 001 | 010 |
| 110   | 111 | 101 |       | 011 | 001 |
| 111   | 000 | 011 |       | 010 | 100 |

| ROUND 1                               | ROUND 2                                |
|---------------------------------------|--|
| $L_0 = 001100$                        | $L_1 = 001010$                         |
| $R_0 = 001010$                        | $R_1 = 010011$                         |
| $K_0 = 11100010$                      | $K_1 = 11000101$                       |
| <b>permute</b> ( $R_0$ ) = 00010110   | <b>permute</b> ( $R_1$ ) = 01000011    |
| 00010110 <b>xor</b> $K_0$ = 11110100  | 01000011 <b>xor</b> $K_1$ = 10000110   |
| $S_1(1111) = 011$                     | $S_1(1000) = 001$                      |
| $S_2(0100) = 111$                     | $S_2(0110) = 011$                      |
| 011111 <b>xor</b> $L_0$ = 010011      | 001011 <b>xor</b> $L_1$ = 000001       |
| $L_1 \leftarrow R_0$                  | $L_2 \leftarrow R_1$                   |
| $R_1 \leftarrow 010011$               | $R_2 \leftarrow 000001$                |
| Exryption after 1 round: 001010010011 | Exryption after 2 rounds: 010011000001 |

**Problem 3.**

In the Rijndael field  $F = \mathbb{F}_2[X]/(X^8 + X^4 + X^3 + X + 1)$ , where bytes are associated to polynomials modulo  $X^8 + X^4 + X^3 + X + 1$ , compute the product  $01010010 \cdot 10010010 \in F$ .

We can represent polynomials in  $F$  as binary numbers, where the state of each bit (whether 0 or 1) represents whether the corresponding power in the polynomial has a factor of 0 or 1.

Then:

$$X^8 + X^4 + X^3 + X + 1 = 100011011$$

Then, we can perform the multiplication modulo 2:

$$\begin{array}{r} 10010010 \\ 1010010 \\ \times 0010010 \\ \hline 10010010 \dots \\ 10010010 \dots \\ \hline 10110210200100 \end{array}$$

Shifting back to base 2, we get: 10110010000100

We then need to find this number mod 100011011

$$\begin{array}{r|l} \text{mod } 10110010000100, 100011011 & \\ 100011011 & 10110010000100 \\ 100000 & 100011011 \dots \\ & 111111100 \dots \\ 1000 & 100011011 \dots \\ & 111001111 \dots \\ 100 & 100011011 \dots \\ & 110101000 \dots \\ 10 & 100011011 \dots \\ & 101100110 \\ 1 & 100011011 \\ \hline 101111 & 1111101 \end{array}$$

Thus, the product in  $F$  is 1111101