# CPSC 418 / MATH 318 — Introduction to Cryptography
# ASSIGNMENT 1

**Name:** Harjee Johal (replace by your name)
**Student ID:** 30000668 (replace by your ID number)

**Problem 1** — Linear Feedback Shift Register Key Streams (10 marks)

(a) The first 19 bits generated by this sequence are: $z_0 = 1, z_1 = 0, z_2 = 1, z_3 = 0, z_4 = 1, z_5 = 1, z_6 = 0, z_7 = 0, z_8 = 1, z_9 = 0, z_{10} = 0, z_{11} = 0, z_{12} = 1, z_{13} = 1, z_{14} = 1, z_{15} = 1, z_{16} = 0, z_{17} = 1, z_{18} = 0$

(b) We can use linear recurrence (1) provided in the question, along with the known sequence:

$$z_i, z_{i+1}, \ldots, z_{i+2m-1}$$

to generate the following system of $m$ equations:

$$\begin{aligned}
z_{i+m} &= c_{m-1}z_{i+m-1} + c_{m-2}z_{i+m-2} + \ldots + c_0 z_i \\
z_{i+m+1} &= c_{m-1}z_{i+m} + c_{m-2}z_{i+m-1} + \ldots + c_0 z_{i+1} \\
&\ldots \\
z_{i+2m-2} &= c_{m-1}z_{i+2m-3} + c_{m-2}z_{i+2m-4} + \ldots + c_0 z_{i+m} \\
z_{i+2m-1} &= c_{m-1}z_{i+2m-2} + c_{m-2}z_{i+2m-3} + \ldots + c_0 z_{i+m-1}
\end{aligned}$$

Since there are $m$ unknown coefficients $c_0, c_1, \ldots, c_{m-1}$, and we have $m$ equations utilizating these coefficients, we can generate a $m$ x $m$ matrix to solve for these coefficients.

(c)

**Problem 2** — Password Counts (20 marks plus 5 bonus marks)

(a) The total number of passwords of length 8 that can be generated using these 94 printable characters is $94^8$

(b) There are two scenarios in this question that need to be considered.

**Case 1: The first four letters of the child's name are all lowercase:**
If the child's name is in all lowercase, then there are $26^3 * 366 = 6,432,816$ possible password candidates. We know that the first four characters of the password are the first four letters of the child's name, and since the first one is known to be 'L', that leaves the next three slots unknown. Each of these slots could be any of the lowercase letters of the alphabet, so there are $26^3$ possibilities.

We also know that the last four characters of the password are the date and month that the user was born in, in DDMM format. The number of unique DDMM combinations is equal to the number of days in the year - there is one possible DDMM combination for each day. Since we know that the user was born in 2008, that means that there are 366 unique DDMM possibilities, since 2008 was a leap year.

By multiplying the number of possibilities for the first four characters of the password by the number of possibilities for the second half of the password, we get $26^3 * 366 = 6,432,816$.

**Case 2: The first four letters of the child's name are all uppercase:**
The number of possibilities in this case is the same as case one: $26^3 * 366 = 6,432,816$. This scenario is almost identical to case one, with the exception that we're working with uppercase letters instead of lowercase letters. However, since there are an equal number of uppercase and lowercase letters in the alphabet, the calculations are the exact same.

Therefore, the total number of password candidates is simply the sum of the number of possibilities of each case: $6,432,816 + 6,432,816 = 12,865,632$ total candidates.

(c) The total number of passwords of length 8 that have at least one numerical digit and at least one special character can be computed by finding the number of passwords that violate this condition, and subtracting them from the total number of passwords of length 8.

First, we compute the total number of passwords of length 8: $94^8$.

Next, we find the number of passwords of length 8 without a numerical digit in them. Since there are 10 numerical digits, the total number of passwords of length 8 that use none of them is: $(94 - 10)^8 = 84^8$.
Next, we find the number of passwords of length 8 without a special character in them. There are 32 special characters, so the number of passwords of length 8 that don't use them is: $(94 - 32)^8 = 62^8$.

Within these two computed subsets of the total passwords of length 8, we've double-counted a specific category. Namely, we've double counted the total number of passwords of length 8 that have neither a numerical digit nor a special characer in them. Therefore, to avoid these subset being subtracted from the total number of passwords of length 8 twice, it must be added once. The total number of passwords of length 8 that don't use a numercal digit or a special character is $(94 - 32 - 10)^8 = 52^8$.

Combining all of this information together, we can compute the number of passwords of length 8 with at least one numerical digit and at least one special character as:

*Total passwords - passwords with no digits - passwords with no special characters + passwords with neither digits nor special characters.*

Substituting in the values that have been computed, we get $94^8 - 84^8 - 62^8 - 52^8$ passwords of length 8 that have at least one numerical digit and at least one special character.

(d) The percentage of 8-character passwords that satisfy the requirements of part (c) is simply a ratio of the value computed in the previous question, and the total number of passwords of length 8:

$$(94^8 - 84^8 - 62^8 - 52^8)/94^8 = 0.5487698\ldots \approx 54.9\%.$$

Therefore, approximately 54.9% of passwords satisfy the requirement in (c).

(e) (**Bonus**)

(f) If each permissable character in a password is chosen with equal likelihood, that means that each password in that password space is chosen with equal likelihood. Therefore, the entropy of the password space is $log_2(n)$, where $n$ is the number of passwords in the password space.

**For part (a):** In part (a) we asserted that the number of passwords of length 8 was $94^8$. Therefore, the entropy of this password space is:

$$\begin{aligned} entropy &= log_2(94^8) \\ &= 52.43671\ldots \\ &\approx 52.4 \end{aligned}$$

**For part (c):** In part (c) we asserted that the number of passwords of length 8 with at least one numerical digit and at least one special character was $(94^8 - 84^8 - 62^8 - 52^8)$. Therefore, the entropy of this password space is:

$$\begin{aligned} entropy &= log_2(94^8 - 84^8 - 62^8 - 52^8) \\ &= 51.5709\ldots \\ &\approx 51.6 \end{aligned}$$

(g) Assuming that each password candidate is chosen with equal likelihood, we would need $2^128$ passwords in order to achieve an entropy of 128. Since we have 94 printable characters to use in our passwords, that means we just have to determine $x$ such that $94^x = 2^{128}$. Solving this equation, we get:

$$\begin{aligned} 94^x &= 2^{128} \\ log_2(94^x) &= log_2(2^{128}) \\ log_2(94^x) &= 128 \\ xlog_2(94) &= 128 \\ x &= 128/log_2(94) \\ x &= 19.5283\ldots \\ x &\approx 19.5. \end{aligned}$$

However, there's no such thing as a password that's 19.5 characters long. Therefore, we must take the ceiling of this value. We cannot take the floor, because a password length of 19 would have a entropy below 128. Therefore the minimum password length that guarantees a password space with entropy 128 is 20.

**Problem 3** — Probabilities of Non-Collisions (26 marks)

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

**Problem 4** — Equiprobability maximizes entropy for two outcomes, 10 marks

(a)

(b)

(c)

**Problem 5** — Cryptanalysis of a class of linear ciphers, 34 marks)

***** Remove the text for this problem if you don't attempt it. *****

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

**Problem 7** — Mixed Vigenère cipher cryptanalysis, 10 marks

***** Remove the text for this problem if you don't attempt it. *****