Worksheet 3

# Exercise 19 (P)

With the set of English letters, S, and the corresponding set of letter-probabilities, P, we compute the Entropy, H, using the function

where is the probability of the corresponding letter , to occur in a text written in the English language.

To compute the Entropy, a C#-program was written. The letter-probabilities from the lecture slides are paired with the corresponding letter in a Dictionary-datatype. The computation is then done for each entry in the dictionary, and summed in an entropy-value. Using this approach, we compute the Entropy to be approximately **4,18**. This number indicates the average number of bits needed to store one letter of an English plaintext.

The code used to calculate the entropy, is found on the following page.

# Code for Exercise 19

using System;

using System.Collections.Generic;

namespace Cryptanalysis.Entropy

{

class Program

{

static void Main(string[] args)

{

//Probabilities extracted from the lecture slides.

Dictionary<char, double> probabilities = new Dictionary<char, double>()

{

{'A', 0.082F},

{'B', 0.015F},

{'C', 0.028F},

{'D', 0.043F},

{'E', 0.127F},

{'F', 0.022F},

{'G', 0.020F},

{'H', 0.061F},

{'I', 0.070F},

{'J', 0.002F},

{'K', 0.008F},

{'L', 0.040F},

{'M', 0.024F},

{'N', 0.067F},

{'O', 0.075F},

{'P', 0.019F},

{'Q', 0.001F},

{'R', 0.060F},

{'S', 0.063F},

{'T', 0.091F},

{'U', 0.028F},

{'V', 0.010F},

{'W', 0.023F},

{'X', 0.001F},

{'Y', 0.020F},

{'Z', 0.001F}

};

double Entropy = 0.0F;

for (char c = 'A'; c <= 'Z'; c++)

Entropy += probabilities[c] \* Math.Log((1 / probabilities[c]), 2);

Console.WriteLine("Entropy: " + Entropy);

//Entropy: 4,18024503236223

}

}

}