COSC412: Assignment 1

Due: 4/9/2022

Instructions

- All work is to be submitted by email to michael.albert@otago.ac.nz by midnight on the due date.
- PDF format for documents is (strongly) preferred. If submitting code, please submit as a separate source file (with instructions for compilation/use if not entirely obvious).
- For the RSA question, the use of sympy is highly recommended (though any language with support for arbitrary-precision integer arithmetic would be suitable).
- There are 12 points indicated—the maximum possible score is 10. You are free to choose your strategy. That could include submitting perfect work for the whole assignment, getting a feeling of quiet satisfaction from having done so, achieving insurance against having made a silly error, and obtaining a score of 10.

Problems

1. Solving simple substitution ciphers in English is easy – there are plenty of online resources, or you can find letter and digram frequency tables and use those. In this problem I'm going to ask you to explore the process of attacking simple substitution ciphers, but the twist is that the original plain text is in Te Reo. You will be given a corpus of plain text in Te Reo which you can use to generate whatever data you require and also two encoded messages. One will include spaces and punctuation, the other (as would be more usual) will have spaces and punctuation removed. The keys used for the two messages (i.e., the actual substitutions) will be different. Your objective is to try and determine as much as you can about the original plain texts. What you will submit (and be graded on) is a short report documenting how you went about this process, as well as any (useful) code produced. I want to see evidence in the report of: a clear plan of how to attack the problem, the results of implementing the plan, and a reflection on how you might have changed things (or did change things). The expected length of the report text is between 500 and 1000 words.

   It is possible to get full marks for this question even if you are unable to decrypt either one of the messages. If your report is poorly written or unconvincing as to the process, it is also possible to get very few marks for this question even if you decrypt both messages. [6 points]

2. Neither of the following pseudo-random generators are secure (for fairly trivial reasons). In each case, demonstrate this fact by giving an efficient statistical test with a significant advantage over the generator.

(a) $G : \{0,1\}^{64} \to \{0,1\}^{70}$ where $G(k)$ is the concatenation of k and the (binary representation of) the remainder when k (as a sixty-four bit integer) is divided by 64. [1 point]

(b) $G : \{0,1\}^{64} \to \{0,1\}^{128}$ where $G(k)$ is the concatenation of k with its bitwise complement (the result of converting all 0s in k to 1s and vice versa). [1 point]

3. Suppose that $G : \{0,1\}^s \to \{0,1\}^n$ *is* a secure pseudo-random generator (assuming that such a thing exists). Which of the following are also secure? Give arguments for or against, not just answers.

(a) $H(k) = G(k) \oplus w$ where w is the string of length n beginning with a 0 and containing 0's and 1's alternately (i.e., $w = 0101\cdots$ until we get n bits). [1 point]

(b) $H(k)$ defined as $G(k)c$, the string of $n+1$ bits where c is the exclusive or of the bits of k (note carefully - the bits of k, not of $G(k)$). [1 point]

4. Consider the two primes:

$$p = 190836086241037,$$
$$q = 569824609824697.$$

Let $N = (p-1)(q-1)$ and consider the basic form of RSA encoding modulo pq with public encoding key

$$e = 436217.$$

(a) Compute the private decoding key d i.e., find d so that $de = 1 \mod N$. [1 point]

(b) The encoding of a "message", m i.e., the value of m^e modulo pq where m is a positive integer less than pq, is

$$98400072373862522893031464992.$$

What was m? [1 point]