University of the Witwatersrand, Johannesburg
School of Electrical Engineering
ELEN 330 – Information Engineering Techniques

Laboratory I1
Classical Cryptography

Note: This lab requires some preparation, in terms of theoretical background as well as the use of the tools (use of the D-Lab, Matlab, the m-files, etc.). Students who are unable to do the lab because they have not prepared will be asked to leave.

Object of the lab: The object of this lab is to give the student practical exposure to some of the classical cryptographic theory presented in class and to introduce some rudimentary cryptographic tools and their use on real data.

Instructions, source material and preparation required:

- All of the Matlab programs referred to in this lab sheet are available on the ELEN330 home page. Please download these files and familiarise yourself with them (i.e. learn how to use them) before your lab session.

- You are required to generate a sample message text between 10000 and 20000 characters long and ensure that it is available in the lab. The message text must be ordinary English as an ordinary ASCII text file (i.e. *not* a Microsoft Word DOC or the like), and can be selected from any text source. The ELEN330 home page makes standard sample texts available – a good choice would be a passage randomly selected from the novel "Far From the Madding Crowd" which is part of the Calgary Corpus Suite (book1.txt in the downloadable zip file).

- You are also required to write a Matlab program for calculating the "index of coincidence" of a sample text. This program must be written and tested before you enter the lab. It may make use of programs on the home page (hint: this can be accomplished quite easily by making use of the 'freqget' function). The program must be written by you - you may *not* copy someone else's program.

- Lab partners must operate groups of two (and no larger) and may help each other during the lab but each should use his/her own sample text in all the exercises and write his/her own lab report.

Report: Reports will be completed in the lab and emailed to the demonstrator before you leave the lab. The report will take the form of the following group of files which should all be attached to a single email for dispatch to your demonstrator:

- The report itself – you must generate an ASCII file with your name and your lab partner's name and student numbers, the date and experiment number, and your response to all the questions posed below. Although your report does not have to be a "long report" (one that is done at home), it should nevertheless conform to the School's requirements for English spelling and grammar, and it should have a conclusion written up in the lab. It should also have an introduction which, along with the student information mentioned above, must have been be prepared *prior* to entering the lab. Your lab report must contain every Matlab command line you used to obtain the results you present.

- Other files as called for in the procedure given below.

---

1. (a) Using the Matlab m files 'readfile', 'strip' and 'freqget', calculate the letter frequency distribution of your sample text file. Reproduce your table of values in your lab report. *Your sample text file must be submitted along with your lab report.* For the remainder of the exercises, assume that all operations are to be carried out on plaintext which has been 'stripped' – using the 'strip' function – of all characters which are not one of the 26 characters of the alphabet.

   (b) Plot the letter frequency distribution found in (a) on top of (i.e. on the same graph as) the expected letter frequency distribution for English. The table of expected values for English as defined in the m file 'freqmatch' should be used. Save this plot as a file of type "fig" and submit it with your lab report.

   (c) Compare the two graphs and make predictions about which letters would likely be confused (if any) in using 'freqmatch' to decrypt a monoalphabetically encrypted version of your message text.

   (d) Generate a translation table suitable for use with 'enmono' and use it to encrypt your sample text. Using 'demono' and the same translation table, check that your ciphertext decrypts correctly. Now use 'freqmatch' and its built-in frequency table for English to break your ciphertext. Compare the plaintext recovered by 'freqmatch' with the original plaintext – explain any disagreements between the two in terms of your observations in 1 (c). How well does the recovered plaintext match the original plaintext? Submit your recovered translation table in your lab report.

(e) Comment on how the result obtained in (d) may be improved.

(f) Repeat your ciphertext break on the ciphertext generated in (d), but this time with the frequency distribution calculated for your sample plaintext in (a) above. Comment on the result.

2. Using your sample plaintext, the function ‘enpoly’ and the function you have written for index of coincidence, create your own IC table for “number of alphabets” from 1 to 5. In your report, describe the method you adopt and comment on your result, comparing it to the table given in class:

Alphabets	1	2	3	4	5	10	Large
IC	0.068	0.052	0.047	0.044	0.044	0.041	0.038

Submit your program for calculating IC with your lab report.

3. (a) Create a short sample text which is devoid of all punctuation and is exactly 22 characters long. Use the ‘encolumn’ function to do a columnar transposition of 7 columns on the sample text. Notice how padding characters have been automatically inserted into the ciphertext. Confirm that the ciphertext is correctly deciphered by using the ‘decolumn’ function and once again take note of the padding characters and their position. In a columnar transposition we expect the letter frequency distribution for the plaintext and ciphertext to be identical because the letters have simply been rearranged. Determine the distribution for both you plaintext and ciphertext. Are they the same? – explain your result. What problem can this give rise to and how can the problem be avoided? Submit all sample texts and results for this exercise.

(b) Create sample text which is exactly 91 characters long. Encipher with a columnar of width 7 and record your result in your report. Double-up the security of your cipher text with a double-columnar by enciphering a second time, this time using a width of 13. Explain and comment on your result. Use what you have learned to formulate and propose a general rule which can be used to avoid this problem. Submit all sample texts and results for this exercise.

(c) * This last exercise is not mandatory. Create a sample text which is exactly 20 characters long. Do a multiple-columnar encipherment of the sample text with repeated application of the ‘encolumn’ function and a column number of 4. How many times must the columnar be applied before the plaintext re-appears? Repeat this exercise for a column width of 5. How does the number (of applications of ‘encolumn’) relate to the sample size and the number of columns used?