The University of Sydney
School of Mathematics and Statistics

Computer Tutorial 3 (Week 3)

MATH2068/2988: Number Theory and Cryptography — Semester 2, 2017

Web Page: http://www.maths.usyd.edu.au/u/UG/IM/MATH2068/
Lecturer: Dzmitry Badziahin

A *translation cipher* is a substitution cipher in which each letter is replaced by the letter that occurs k steps later in the alphabet. More precisely, the ith letter of the alphabet is replaced everywhere by the jth letter, where $j \equiv i+k \pmod{26}$. (The case $k = 3$ is *Caesar's Cipher*. The case $k = 1$ occurred in Question 3 of Computer Tutorial 2.) A translation cipher is completely determined by the letter that replaces A, and so it is traditional to call this single letter the key.

A *Vigenère cipher* is constructed from n translation ciphers. To encipher the first letter of a message the first cipher is used, for the second letter the second cipher is used, and so on; for the $(n+1)$st letter you return to the first cipher. To be precise, the ith letter of the message is enciphered with the jth translation cipher, where $j \equiv i \pmod{n}$. For a Vigenère cipher the key is the n-letter word made up of the keys of the n translation ciphers.

Start MAGMA and type `load "tut3data.txt";`

1. Type `V:=VigenereCryptosystem(7);` and `k:=V!"CARSLAW";`, thus setting up a Vigenère cipher with keyword `CARSLAW`, and then use it to encipher a message of your choice. (For example, the commands `M:="Computer Tutorial Three"; P:=Encoding(V,M); C:=Enciphering(k,P);` would do.) Get MAGMA to print your enciphered message, and examine a few letters to check that it is right. (To find where various letters occur in the alphabet, you can use commands like `alphabet[15];` and `Index(alphabet,"Q");`.) Type `m:=InverseKey(k);` then print `m` and check that it is right. Finally, check that the command `Enciphering(m,C);` recovers `P`.

The probability that two randomly chosen letters from a piece of text are the same is called the *coincidence index* for the text. Suppose there are N letters in the text, of which N_1 are A's, N_2 are B's, and so on. So $N = \sum_{i=1}^{26} N_i$. Let $p_i = N_i/N$. If we choose a letter at random then the probability that it is an A is p_1, the probability that it is a B is p_2, and so on. If we choose one letter and then independently choose another then the probability that they are both A's is p_1^2, and the probability that they are both B's is p_2^2, and so on. The probability that the two randomly chosen letters are the same is the probability that they are both A's plus the probability that they are both B's plus ... plus the probability that they are both Z's. That is, the coincidence index is $\sum_{i=1}^{26} p_i^2$. For a random string of letters from a 26 letter alphabet the expected value of the CI is $1/26 \approx 0.0385$, but for English text it is usually about 0.066. (Why the difference?)

2. The file `tut3data.txt` that you have loaded contains an excerpt from the short story "The Dancing Men" by Sir Arthur Conan-Doyle. Type `dancemen;` to see it, and then type the command `CoincidenceIndex(dancemen);` to see the value of the coincidence index for this text. Observe that it is close to typical. Type `S:=SubstitutionCryptosystem();` followed by `dm:=Encoding(S,dancemen);`, then `rk:=RandomKey(S);`, then `rk;`(to see the key MAGMA has chosen) and `cdm:=Enciphering(rk,dm);` (to encipher the message). Get MAGMA to tell you `CoincidenceIndex(cdm)`, and compare the result with the CI of the unenciphered text; they are the same. Can you explain this?

Now encipher `dancemen` using the Vigenère cipher from Exercise 1, and calculate the CI of

the result. (Type `dmv:=Encoding(V,dancemen);` and `cdmv:=Enciphering(k,dmv);`, then `CoincidenceIndex(cdmv);`.) Would using a different key give a different CI? (To check this, do `k:=RandomKey(V);`, encipher `dmv` with this new key, and find the CI. Repeat a couple of times.) You will probably find that the CI is less for the ciphertext than it is for the plaintext, but still greater than $\frac{1}{26} \approx 0.0385$. Can you explain why? How would the CI be affected if a longer Vigenère period was chosen? Try it and see! (Start with the commands `VV:=VigenereCryptosystem(20);` and `dmvv:=Encoding(VV,dancemen);`.)

The file tut3data.txt contains some Vigenère-enciphered ciphertexts, named `vt1`, `vt2`, `vt3` and `vt4`. In the remaining exercises we shall attempt to decipher them via frequency analysis. Our main tool is a function called `Decimation` that is defined in our MAGMA cryptography package.

3. Try to figure out what the `Decimation` function does: type

```
alph:="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
Decimation(alph,1,3);
Decimation(alph,2,3);
Decimation(alph,2,6);
```

and examine the output. (There is a bug: it doesn't print as many terms as it should.)

4. All objects that MAGMA works with must have a "type". For example, `dm` and `cdmv` above are, in MAGMA's opinion, objects of type "CryptTxt", while `k` and `rk` are of type "CryptKey". (Type the command `Type(dm);`.) A string of characters, such as `alph` above, is an object of some other type. (Type `Type(alph);`.) The `Decimation` function can only be used on strings, not on cryptographic text. However, in the next exercise we shall want to decimate `cdmv`. So that we can do so, type `scdmv:=String(cdmv);` to create a string `scmdv` consisting of the same letters as `cdmv`. Check that `vt1`, `vt2`, `vt3` and `vt4` are already strings.

5. Before starting work on `vt1` let us examine decimations of the ciphertext `cdmv;`. Recall that the Vigenère period for `V` is 7. Type

```
CoincidenceIndex(Decimation(scdmv,1,7));
CoincidenceIndex(Decimation(scdmv,2,7));
CoincidenceIndex(Decimation(scdmv,3,7));
```

Do the same again using a decimation period that is not equal to the Vigenère period. (For example, use the same commands with 6 instead of 7.) Repeat for several choices of the decimation period. What do you observe?

6. We now try to find the Vigenère period for `vt1` by checking the CI for decimations of periods from 2 to 20. Type

```
for i:=2 to 20 do
  print "Period:",i,"CI:",CoincidenceIndex(Decimation(vt1,1,i));
end for;
```

7. The output from the above commands should enable you to correctly deduce that the Vigenère period for `vt1` is 13. Use `SortedFreqDist(Decimation(vt1,1,13));` to find which letter occurs most frequently in this decimation. This letter presumably represents E in the first of the 13 translation ciphers that make up the Vigenère cipher. Work out what represents A. Repeat this idea for `Decimation(vt1,i,13)`, for all values of `i` from 1 to 13, and hence determine the key.

8. Type `V1:=VigenereCryptosystem(13);` and `k1:=V1!"BATMANFOREVER";` to define `k1` to be the key you (should have) found in the previous exercise. After this, typing the command `Enciphering(InverseKey(k1),Encoding(V1,vt1));` should give you something readable.

9. Do the same for at least one of `vt2`, `vt3` and `vt4`. (Be warned that sometimes the most frequent letter in a decimation does not represent E. So you should check the assumption that it does by looking at some other frequencies. For example, in one case the most frequent letter is B, followed by I and M. If B represents E then I represents L and M represents P. This can't be right: maybe L could be the second most frequent letter in some unusual piece, but surely P cannot be the third most frequent letter! If I represents E then B represents X. This can't be right either. In fact M represents E, which means that B represents T and I represents A.)