Feedback on Topic 2 Ex

MANCHESTER 1824

The University of Manchester

Exercise Question – E2.1

- Given the following ciphertext which has been generated using the Caesar cipher (but a different key), use the frequency analysis method to work out the encryption key and the corresponding plaintext.

Ciphertext:

bpmzm wvkm eia iv cotg lckstqvo eqbp nmibpmza itt abcjjg ivl jzwev …

Key: ??

Plaintext: ??

Exercise Question – E2.1 (this is plaintext distribution)

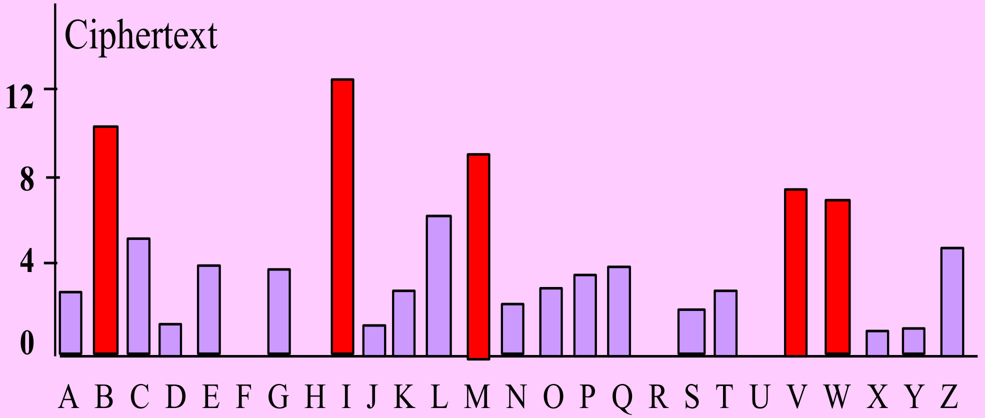
- **Letter Frequency Distribution in English (in percentage) (**this may vary depending on the content/size of the text**)**

- a b c d e f g h i
- 8.2 1.5 2.8 4.2 12.7 2.2 2.0 6.1 7.0

- j k l m n o p q r
- 0.1 0.8 4.0 2.4 6.7 7.5 1.9 0.1 6.0

- s t u v w x y z
- 6.3 9.0 2.8 1.0 2.4 2.0 0.1 0.1

Exercise Question – E2.1: Plain/cipher-text distributions



- To guess the key K=?
- Do a letter frequency distribution for the ciphertext and compare it with the plaintext distribution.
- The more frequently occurring letters in the ciphertext are likely to be among the more frequently occurring letters in the plaintext.
- Look at Plaintext 'E', P(E): if P(E) is C(I), C(M), C(V), C(W), or C(B), then the keys would be respectively:
  - 4, 8, 17, 18 or 23
- Do the same for P(A), we have:
  - 1, 8, 12, 21, or 22
- Among the two sets of possible keys, one key (**K = 8**) appears in both sets.
- So try to use K=8 to decrypt …
- This is just one of the possible methods.

Exercise Question – E2.1

Ciphertext:

bpmzm wvkm eia iv cotg lckstqvo eqbp nmibpmza itt abcjjg ivl jzwev

Plaintext:

There once was an ugly duckling

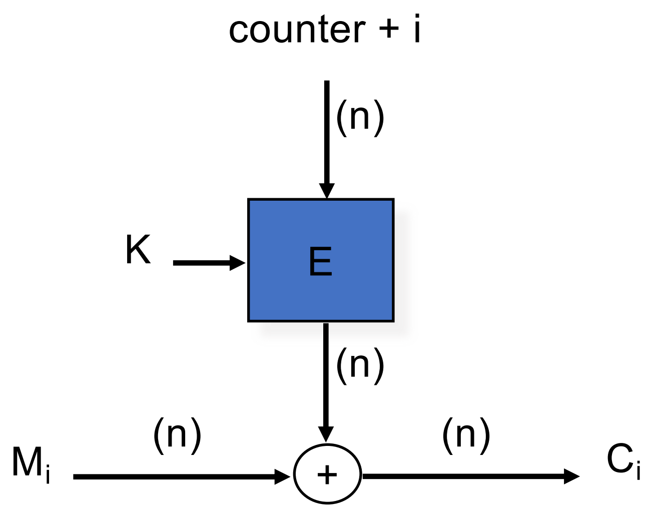
With feathers all stubby and brown …

Exercise Question – E2.2

(i) Comment on the benefit(s) of this approach, i.e. why is the key stream generated from K?

- This approach addresses the following issue:
  - The key stream must be unique for each encryption, i.e. a key stream must not be used twice, as, otherwise, the encryption will not be secure:
  - $K = M \text{ xor } C \quad \Rightarrow \quad M' = K \text{ xor } C' = (M \text{ xor } C) \text{ xor } C'$
  - This is a dangerous property and we **must never *ever* reuse the same keystream** to encrypt two different messages.
  - To ensure a key stream non-repeating can be challenging: (a) their distributions are expensive – a key stream should be as long as the message to be protected and this is too expensive for long messages; (b) managing and storing a large number of key streams may also be problematic; (c) there is an synchronisation issue too – the key stream used by a sender/receiver pair for a particular message must be the same.

Exercise Question – E2.2 (cont)



(ii) How to ensure (or to minimize the chances) that the output of the pseudo-random generator (i.e. the key stream) is non-repeating?

(a) Use a strong mixing function as the pseudo-random generator.

(b) add another input into the function, a counter, which changes (e.g. increment by 1) for each iteration.

(c) If the counter value reaches its maximum, then change the key, K.