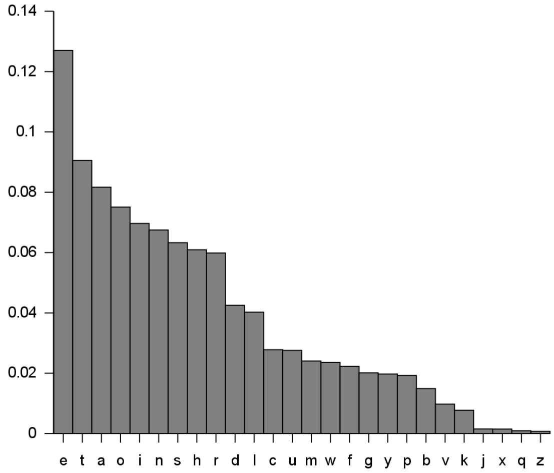
Institut of Computer Science, University of Bern
Cryptography, HS 2019

Prof. Christian Cachin
Luca Zanolini

Exercise 1

1.1 Hiding the plaintext statistics of natural language (3pt)

The letter frequencies of natural-language English text are shown in the diagram below. A monoalphabetic substitution cipher for encrypting text uses a random permutation K of the 26 letters as key and replaces each letter ℓ by $K[\ell]$. Since there is a one-to-one correspondence between every plaintext letter and ciphertext letter, the statistics of the plaintext are leaked. Therefore the cipher can be cryptanalyzed if the plaintext is sufficiently long.



a) Suppose one uses *text compression* to preprocess the plaintext before encryption with monoalphabetic substitution. For example, one could use a utility like GZIP, suitably modified to output text in a 26-letter alphabet. The compressed text will have almost uniform letter statistics. How secure is this method?

b) Another encoding method to disguise plaintext letter frequencies is called *homophonic substitution*. It works by mapping plaintext letters to more than one ciphertext letter before encryption. Every letter is associated with a number of homophones, say, e is mapped to $e_1, e_2, \ldots, e_{13}$. Letters with high frequency are given more equivalents than letters with low frequency. Encoding picks one of the homophones uniformly at random, such that some e_j has about the same relative frequency ≈ 0.01 as the letter k. In this way, the frequency distribution is flattened. How secure is this?

c) (Bonus question) Assume instead that one has a proper, secure encryption scheme (unlike monoalphabetic substitution) and one wants to compress and encrypt the plaintext. Should compression be applied to the plaintext, before encryption, or to the ciphertext, after encryption?

Justify your answers.

1.2 Does the one-time pad leak information? (2pt)

Alice is using one-time pad (OTP) and notices that when her key is the all-zeroes string $k = 0^\lambda$, then $\mathrm{Enc}(k, m) = m$ and her message is sent in the clear! The message is also easily visible with the key $k = 1^\lambda$. To avoid this problem, she decides to modify *KeyGen* to exclude the all-zeroes and all-ones key. She modifies *KeyGen* to choose a key uniformly from $\{0, 1\}^\lambda \setminus \{0^\lambda, 1^\lambda\}$, the set of all λ-bit strings except 0^λ and 1^λ . In this way, she guarantees that her plaintext is not leaked directly. Is it still true that the eavesdropper's ciphertext distribution is uniform? Prove or disprove.

1.3 One-time pad using the same key (3pt)

Suppose Alice encrypts two plaintexts m and m' using one-time pad with the same key k. What information about m and m' is leaked to an eavesdropper by doing this (assume the eavesdropper knows that Alice has reused k)? Be as specific as you can!

1.4 Attack at one-time pad (2pt)

A known-plaintext attack refers to a situation where an eavesdropper sees a ciphertext $c = \mathrm{Enc}(k, m)$ and also learns or knows what plaintext m was used to generate c.

a) Show that a known-plaintext attack on OTP results in the attacker learning the key k.

b) Can OTP be secure if it allows an attacker to recover the encryption key? Is this a contradiction to the security we showed for OTP? Explain.