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

Prof. Christian Cachin
Luca Zanolini

Solutions for exercise 1

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

a) Using text compression to preprocess the plaintext before encryption with monoalphabetic substitution makes frequency analysis less useful. The reason is that the sequences of the plaintext are first compressed to letters with a close-to-uniform distribution and the letters that enter encryption no longer follow the skewed distribution of English text. So, concerning this attack, compression enhances the security.

b) It generally makes an encryption method more secure because the plaintext statistics are "smoothed out," that is, the ciphertext letters have a more uniform frequency distribution than in natural language.

However, one can still analyze frequencies by guessing sets of homophones which map to the same plaintext letter. With this one would consider single letters, two-letter pairs (bigrams) and longer text patterns. For example, if plaintext is written in English, then a very common bigram to test is *th*. If an alphabet including the homophones has 100 symbols, for example, then the number of bigrams is $100^2 = 10000$ and every bigrams should have frequency $0,0001$. It turns out that the structure of a language makes bigrams to have an higher frequency, allowing frequency analysis.

A detailed explanation of cryptanalysis for homophonic substitution ciphers appears at: `http://www.cs.sjsu.edu/faculty/stamp/RUA/homophonic.pdf`.

c) (Bonus question) The plaintext should be compressed *before* it is encrypted, so that one can benefit from saving space by using compression. This is because compression algorithms take advantage from repeated patterns in uncompressed text. GZIP, for example looks for such redundancies and shortens source code or English text considerably. If one encrypts a text (with a secure encryption scheme) and then applies compression to the ciphertext, the compression algorithm will no longer find such repeated patterns and compression no longer works.

Note: Questions a) and b) only refer to historic encryption methods, which are *insecure* according to the definition of secure encryption presented in the course.

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

With this modification the eavesdropper's distribution over the possible plaintexts is no longer uniform. In fact, when the eavesdropper learns that the ciphertext is c, she knows that the plaintext cannot have been c nor its complement, i.e., that $m \neq c$ and $m \neq \bar{m}$. In other words, $\mathrm{P}[\textsc{eavesdrop}(m) = m] = \mathrm{P}[\textsc{eavesdrop}(m) = \bar{m}] = 0$ for every $m \in \{0,1\}^\lambda$. With the original OTP, every possible plaintext is equally probably for the eavesdropper, including the plaintext itself.

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

Let k be the key used to encrypt two messages m and m'. It follows that $c = k \oplus m$ and $c' = k \oplus m'$. One can XOR the two ciphertexts such that $c \oplus c' = (k \oplus m) \oplus (k \oplus m') = m \oplus m'$. From this, it is possible to have information about the two plaintexts.

Furthermore, several attacks are possible against $m \oplus m'$ such as frequency analysis (the 0s in $m \oplus m'$ are matching characters in m and m') or the so-called *crib-dragging* where the attacker tries to XOR "meaningful" pieces of data in different positions of $m \oplus m'$ and sees if the result leads to useful information about the two plaintexts.

Historical fact: A concrete example where the use of the same key in OTP brought at the decryption of sensible informations is the Project Venona.
(`https://www.nsa.gov/news-features/declassified-documents/venona/`)

1.4 Attack at one-time pad (2pt)

a) If an attacker knows m and c, then it is possible to derive the key by XORing them, i.e, $k = m \oplus c$.

b) The definition of OTP implies that every instance of the OTP uses an independent key. So, even if the attacker knows the encryption key for a particular plaintext/ciphertext pairs, this is useless for any further instance.