Homework 1—Solutions

1. (As usual, we associate the English alphabet with the set \{0, \ldots, 25\}.) Gen: choose the period \(t\) uniformly from \(\{1, \ldots, t^*\}\) (where \(t^*\) is some upper bound on the period); then choose \(k_1, \ldots, k_t\) uniformly and independently from \(\{0, \ldots, 25\}\). (Note that this is not the same as choosing a uniform key from the key space.)

\(\text{Enc}:\) on input a key \(k_1 \cdots k_t\) and a message \(m_1 \cdots m_\ell\), where each \(m_i \in \{0, \ldots, 25\}\), output the ciphertext \(c_1 \cdots c_\ell\) where

\[ c_i := \left[ m_i + k_{[i \mod t]} \mod 26 \right]. \]

\(\text{Dec}:\) on input a key \(k_1 \cdots k_t\) and a ciphertext \(c_1 \cdots c_\ell\), where each \(c_i \in \{0, \ldots, 25\}\), output the message \(m_1 \cdots m_\ell\) where

\[ m_i := \left[ c_i - k_{[i \mod t]} \mod 26 \right]. \]

2. If \(abcd\) is encrypted with the shift cipher, the second character of the ciphertext will always be one more (modulo 26) than the first character of the ciphertext; if \(bedg\) is encrypted, the second character of the ciphertext will always be three more (modulo 26) than the first character of the ciphertext. This makes it easy to tell which of the two messages was encrypted.

3. Period 2. It is not possible to tell which password was encrypted. The reason is because in this case the shift used in the first and third positions is the same, and the shift used in the second and fourth positions is the same, but the shifts used in the first/third and second/fourth positions are independent. And the difference between the first and third characters (resp., the second and fourth characters) of the first password is the same as the difference between the first and third characters (resp., the second and fourth characters) of the second password.

Period 3. Here it \textit{is} possible to tell which password was encrypted, because the shift used in the first and fourth positions is the same. So if \(abcd\) is encrypted, the fourth character of the ciphertext will always be three more (modulo 26) than the first character of the ciphertext; if \(bedg\) is encrypted, the fourth character of the ciphertext will always be five more (modulo 26) than the first character of the ciphertext.

Period 4. It is not possible to tell which password was encrypted, because using a 4-character key to encrypt a 4-character plaintext is perfectly secret (by analogy with the one-time pad).
4. First note that encryption with two uniform keys \( k_1, k_2 \) of the same length is equivalent to encrypting with one uniform key \( K^* \) of that length: XORing by one uniform byte followed by another is the same as just XORing with one uniform byte.

If the keys \( k_1, k_2 \) have different lengths \( \ell_1, \ell_2 \), then this can be viewed as encrypting with one key \( K^* \) of length \( \text{lcm}(\ell_1, \ell_2) \). In this case, however, the distribution of \( K^* \) is not uniform because there are dependencies among its bytes. This can only make it easier to break.

5. If the attack is run assuming English plaintext, but the plaintext was actually in some other language, then the first step would work fine since all we are doing there is maximizing the sum-of-squares (and there is no specific dependence on English). The second step, however, would fail because the frequencies of letters in another language will not (in general) be the same or even close to those in English.