COMS30023 / Cryptology

Problem Sheet 2

François Dupressoir *

2025/26

In this problem sheet, we explore blockciphers a bit more in depth by considering classical designs and how they fail. This helps you practice attack finding skills on simple designs before we move on to grander things. As such, the knowledge we exercise here isn't particularly important to secure

We will consider blockciphers for which $\mathcal{M} = \mathcal{C} = \{\mathtt{a}, \ldots, \mathtt{z}\}^9$ (that is, plaintexts are 9-letter strings).

1. a) $\star$ Determine $|\mathcal{M}|$ and estimate, to one decimal, $\lg(|\mathcal{M}|)$.

   b) $\star$ Determine $|\mathcal{M} \to \mathcal{M}|$. What can you say about $\lg(|\mathcal{M} \to \mathcal{M}|)$? (In relation to the previous calculation.)
   ($\mathcal{X} \to \mathcal{X}$ is the set of *functions* with $\mathcal{X}$ as both domain and codomain.)

   c) $\star$ What do those quantities represent—in terms of objects we defined in the lecture, and why might we be interested in them?

2. You may have heard of transposition (or *shuffling*) ciphers (for example, columnar transposition), that operate by changing the *position* of letters in a text. For example, one could use the following table to define a shuffle on nine positions.

in	1	2	3	4	5	6	7	8	9
out	4	6	1	5	3	7	9	2	8

   With this shuffle, `abcdefghi` would encipher to `cheadbfig`, so the first letter in the plaintext (`a`) goes to the fourth position in the ciphertext, and the first letter in the ciphertext (`c`) originates from the third position in the plaintext.

   a) $\star$ Decipher `vlooibuys` using the shuffle given above as key.

   b) $\star$ What is the keyspace $\mathcal{K}$ and how large are keys (in bits)?

   c) $\star$ Describe an adversary that distinguishes the shuffling cipher from a random permutation with an overwhelming advantage in a single query and minimal computation. (Calculate its advantage.)

   d) $\star$ Suppose the adversary is more ambitious than simply distinguishing and wants to recover the key using a chosen plaintext attack. Explain how you would recover the key. Try to maximize the key recovery advantage while minimizing the number of queries and adversarial runtime.

3. Shuffling ciphers aren't very good. Another class of historical ciphers is known as *substitution* ciphers, where each letter of the alphabet is substituted by another one. For instance, one could use the following table to define the substitution.

[*] Based on material by Dr. Martijn Stam, Dr David Bernhard and others

in	`abcdefghijklmnopqrstuvwxyz`
out	`francoiszyxwvutqpmlkjhgedb`

a) $\star$ Decipher `atvqjkcml` using the substitution give above as key.

b) $\star$ What is the keyspace $\mathcal{K}$ and how large are keys (in bits)?

c) $\star$ Describe an adversary that distinguishes the substitution cipher from a random permutation with an overwhelming advantage. Try to maximize the distinguishing advantage while minimizing the number of queries and adversarial runtime.

d) $\star\star$ Suppose the adversary is more ambitious and wants to recover the key using a chosen plaintext attack. Explain how you would recover the key. Try to maximize the key recovery advantage while minimizing the number of queries and adversarial runtime.

4. Shuffling once or substituting once is rubbish as an enciphering mechanism. But can we instead combine both operations, and iterate them a couple of times? Let's consider that we use P_k with $k \in \mathrm{Perm}(\{1,\ldots,9\})$ to denote a shuffle, and S_k with $k \in \mathrm{Perm}(\{\mathtt{a},\ldots,\mathtt{z}\})$ to denote a substitution. We can create an enciphering scheme E by composing shuffles and substitutions as follows.

```
Kg
$k_1 \leftarrow_\$ \mathrm{Perm}(\{1,\ldots,9\})$
$k_2 \leftarrow_\$ \mathrm{Perm}(\{\mathtt{a},\ldots,\mathtt{z}\})$
$k_3 \leftarrow_\$ \mathrm{Perm}(\{1,\ldots,9\})$
$k_4 \leftarrow_\$ \mathrm{Perm}(\{\mathtt{a},\ldots,\mathtt{z}\})$
return $(k_1, k_2, k_3, k_4)$
```

```
$\mathsf{E}_{(k_1,k_2,k_3,k_4)}(m)$
$c \leftarrow \mathsf{P}_{k_1}(\mathsf{S}_{k_2}(\mathsf{P}_{k_3}(\mathsf{S}_{k_4}(m))))$
return $c$
```

a) $\star$ Give the deciphering algorithm.

b) $\star\star$ Argue (don't prove) that the repetition in the enciphering scheme is pointless, so we can consider only a two-key scheme $\mathsf{E}_{k_5,k_6} = \mathsf{P}_{k_5} \circ \mathsf{S}_{k_6}$ instead, without loss of generality (or security).

c) $\star\star$ Give a distinguishing attack. (Describe an adversary that distinguishes this enciphering scheme from a random permutation with high probability and low cost in queries or running time.)

d) $\star\star\star$ Sticking to the simplification from (b), describe a key recovery attack under chosen plaintext attack that has advantage 1. You don't have to try to minimize the number of queries, but try to avoid exhaustive search, while still being guaranteed to recover the key (k_5, k_6).

5. Shuffling and substitution on their own, and taken together, are simply not good enough. We throw another ingredient into the mix. The Vigenère cipher is a generalization of Caesar's cipher, where letters of the alphabet are added together, identifying `a` with 1, `b` with 2, all the way up to identifying `z` with 0. Additions are done modulo 26. Given two words of the same length, we can add them letter by letter.

```
Kg
$k \leftarrow_\$ \{\mathtt{a},\ldots,\mathtt{z}\}^9$
return $k$
```

```
$\mathsf{V}_k(m)$
$c \leftarrow m + k$
return $c$
```

a) $\star$ Use Shannon's theorem to demonstrate that Vigenère's scheme is perfectly secret.

As mentioned, we'd like to combine the Vigenère cipher with shuffles and substitutions. The hope is that having three different mechanisms in play will work better than only the two.

We first consider whether repetition helps, or whether it is as pointless as with substitutions and shuffles.

a) ⋆⋆⋆ Argue that when combining Vigenère with shuffles, repetition is pointless. That is, for all k_1, k_2, k_3, k_4, there exist k_5 and k_6 such that

$$\mathsf{P}_{k_5} \circ \mathsf{V}_{k_6} = \mathsf{P}_{k_1} \circ \mathsf{V}_{k_2} \circ \mathsf{P}_{k_3} \circ \mathsf{V}_{k_4}$$

b) ⋆⋆ When combining Vigenère with substitutions, repetitions *do* in fact add complexity. However, you can still find a distinguishing attack. Do so.

c) ⋆⋆⋆ Suppose that a cipher consists of ten repetitions, or rounds, each consisting of a substitution followed by Vigenère—all with independent keys. Describe an efficient chosen plaintext attack that recovers a complete description of the keyed encryption and decryption algorithms. (As lookup tables or functions—this is easier than recovering all 20 subkeys!)
(Hint 1: Which letters of the plaintext does the ith letter of the ciphertext depend on? — Answering this will help you understand how you can recover an algorithm to decrypt without recovering the entire key.)
(Hint 2: 26 chosen plaintexts will suffice. Attacks with a lot less exist for those of you who have nothing better to do.)