Universität Leipzig
Institute of Computer Science
Winter term 2026/2027

Andreas Maletti

Exercises for the lecture

Cryptography

Solutions to Exercise Sheet 3

for the exercise on November 6th, 2026

Exercise 3.1

Decrypt. (will not be done during exercise)

```
CfwnjhyvdlsfxxJuxpqsjyaqqxjQcuazhxtqpSdrylafSfftxzukmaktwnkuz
kqkywgNlsfhfguakgkbjEsaeojzniAsagkCgnxbftbtgkgsfsvxxZglzfsEhj
hymnbbyzkwkbhqOdekPkgozvwgchxwakPjnkgkkSlrwbtvggbqsjYrxkzaikw
brwCxiasjzxrylaTijymhrnjdsfXoqklBosmgjhyYnbifwfxyxekKwkkltaki
rwlChouaofmgmhtFepqzxyhogakXmukokbjGooxxdavdkijxoqkLabzwLkltg
akbltaijxiBfvxtuagzRbvpkuygebuwduqtmzbfkmkkrxSbwfxtgybyeomljl
klajPmgjIxxqaslkoqqxjxzdxChyxjXbvxtExnaphwgjhxGwqijTrokZqqsft
rokUkbgwgLrrzakwzkkuXhobbkiauQnapgwzgeybarbkntgXxxbbWbthtYnbi
fwmhvkqbtlbsWuwSlzdnywctnarwfCxxfcbuwukqagzasjVnhxnxphwazyukC
lhlYxrnpebgwbthYhjksfyrlkzakRmkikjxoEwefkoyinxsuazlmxjMzsgRda
yaqPjnkgkkarfwUgktYnbivbmzoxafbZxrgfniPwwzkqYxeaielikrnjdsfFo
orbkksfWohyxjHikljhxzwknwgChrmXoiwwkuaxxbfeLzhxgaknwezPalobwf
eohhxnSolxxzuajbbAaxvznaonlgohjxnJwdeortxjXvfxywjnzbbKvnrkibb
fOxrwYnyewzgahhxnjGlxxqkgvbzlNkekkOqsjgkqsnopsjpuktxj
```

(Ciphertext available on my website; capitalization only for readability; lower- and upper-case letters can be treated the same)

Solution: It is a Vigenère cipher and decrypts to part of the German text of "Ode to Joy" (Symphony No. 9, Beethoven).

Exercise 3.2

Suppose a one-time pad encryption scheme with key length $\ell \in \mathbb{N}$. Obviously, the key space has size 2^ℓ, so we just try all 2^ℓ keys and decrypt a ciphertext with all those keys and accept the decryption if it returns proper English. This procedure obviously takes exponential time, but is clearly computable. However, we proved that one-time pad is perfectly secret. Explain the flaw in the above brute-force approach.

Solution: The problem is that for any proper English sentence m' of length ℓ there exists exactly one key that turns the given ciphertext into m'. Since these keys are all equally likely, all those messages are also equally likely, so brute-forcing one-time pad reduces to an arbitrary selection among all proper English sentences of length ℓ.

Exercise 3.3

Let $\ell \in \mathbb{N}$. Consider the following program: Draw k from a uniform distribution on $\{0,1\}^\ell$, draw m from any unknown distribution on $\{0,1\}^\ell$, and return $k \oplus m$. The two draws are independent. Prove that the program returns values uniformly distributed in $\{0,1\}^\ell$. (In other words, as long as one summand of an XOR is uniformly distributed, the XOR operation removes any bias; i.e., turns any arbitrary distribution of the other summand into a uniform output distribution.)

Solution: Let $c \in \{0,1\}^\ell$ and $D\colon \{0,1\}^\ell \to [0,1]$ be the unknown distribution. Let us compute the probability of c.

$$\begin{aligned}\mathsf{P}(c) &= \sum_{\substack{k,m\in\{0,1\}^\ell \\ c=k\oplus m}} \mathsf{P}(k,m) = \sum_{\substack{k,m\in\{0,1\}^\ell \\ c=k\oplus m}} \mathsf{P}(k)\cdot P(m) = \sum_{\substack{k,m\in\{0,1\}^\ell \\ c=k\oplus m}} 2^{-\ell}\cdot D(m) \\ &= 2^{-\ell}\cdot \sum_{\substack{k,m\in\{0,1\}^\ell \\ c=k\oplus m}} \mathsf{D}(m) = 2^{-\ell}\cdot \sum_{\substack{k,m\in\{0,1\}^\ell \\ k=c\oplus m}} \mathsf{D}(m) = 2^{-\ell}\cdot \underbrace{\sum_{m\in\{0,1\}^\ell} \mathsf{D}(m)}_{=1} = 2^{-\ell}\end{aligned}$$

Note that for every $m \in \{0,1\}^\ell$ there exists a unique $k = c \oplus m$ such that $c = k \oplus m$. Thus, the probability for each $c \in \{0,1\}^\ell$ is $2^{-\ell}$ and thus the outputs are uniformly distributed.

Exercise 3.4

Are ciphertexts uniformly distributed in a perfectly secret encryption scheme?

Prove or refute: Given a perfectly secret encryption scheme $\mathcal{E}$ with message space M and ciphertext space C and an arbitrary scenario $(\mathcal{E}, \mathsf{P}_M)$, we have $\mathsf{P}(c) = \mathsf{P}(c')$ for all ciphertexts $c, c' \in C$.

Solution: The statement is false. Let $\mathcal{E}' = (\mathrm{gen}, \mathrm{enc}', \mathrm{dec}')$ be any perfectly secret encryption scheme (e.g., one-time pad) with key space K, message space M, and ciphertext space C'. We construct a new encryption scheme $\mathcal{E} = (\mathrm{gen}, \mathrm{enc}, \mathrm{dec})$ with the same key space K, same message space M, and (potentially different) ciphertext space $C = C'\{0,1\}$ as follows:

- $\mathrm{enc}_k(m)$ is given for every $k \in K$ and $m \in M$ by the following procedure:
  (a) $c' \leftarrow \mathrm{enc}'_k(m)$
  (b) draw a from uniform distribution on $\{0,1\}^2$
  (c) if $a = 00$ return $c'0$ else return $c'1$

- $\mathrm{dec}_k(c)$ is given for every $k \in K$ and $c \in C$ by
  (a) removing the last bit b from c; i.e. $c = c'b$
  (b) returning $\mathrm{dec}'_k(c')$

We start by proving perfect correctness. Let $k \in \mathrm{ran}(\mathrm{gen})$, $m \in M$, and $c \in C$ such that $\mathsf{P}\big[c \leftarrow \mathrm{enc}_k(m)\big] \neq 0$. Since $C = C'\{0,1\}$, we can write $c \in C$ as $c = c'b$ with $c' \in C'$ and $b \in \{0,1\}$. Consequently, $\mathsf{P}\big[c' \leftarrow \mathrm{enc}'_k(m)\big] \neq 0$. Since $\mathcal{E}'$ is perfectly correct, we have $\mathrm{dec}'_k(c') = m$. Since $\mathrm{dec}_k(c) = \mathrm{dec}'_k(c') = m$, this proves correctness. Perfect secrecy and non-uniform ciphertext distribution remain as proof obligations. We start with perfect secrecy. Let $c \in C$ be a ciphertext with $c = c'b$ as usual, and $m, m' \in M$ be messages. Now we distinguish two cases:

(a) Let $b = 0$.

$$\begin{aligned}\sum_{k\in K} \mathsf{P}_K(k) \cdot \mathsf{P}\big[c \leftarrow \mathrm{enc}_k(m)\big] &= \frac{1}{4}\sum_{k\in K} \mathsf{P}_K(k) \cdot \mathsf{P}\big[c' \leftarrow \mathrm{enc}'_k(m)\big] \\ &\overset{\dagger}{=} \frac{1}{4}\sum_{k\in K} \mathsf{P}_K(k) \cdot \mathsf{P}\big[c' \leftarrow \mathrm{enc}'_k(m')\big] \\ &= \sum_{k\in K} \mathsf{P}_K(k) \cdot \mathsf{P}\big[c \leftarrow \mathrm{enc}_k(m')\big]\end{aligned}$$

where we use the perfect secrecy of $\mathcal{E}'$ at $\dagger$.

(b) Let $b = 1$. The proof is analogous with $\frac{1}{4}$ replaced by $\frac{3}{4}$.

Thus we established that also in $\mathcal{E}$ the probability of the ciphertext does not depend on the message, which proves perfect secrecy of $\mathcal{E}$. Now we finally move to the ciphertext distribution. Let $(\mathcal{E}, \mathsf{P}_M)$ be any scenario. Since we are perfectly secret, the message distribution P_M actually does not matter. Let $c' \in C'$ be a ciphertext and $m \in M$ such that $O_{\mathcal{E}'}(c, m) \neq 0$; i.e., a possible ciphertext in the original encryption scheme $\mathcal{E}'$. Then

$$\begin{aligned}\mathsf{P}_C(c'0) = O_{\mathcal{E}}(c'0, m) &= \sum_{k\in K} \mathsf{P}_K(k) \cdot \mathsf{P}\big[c'0 \leftarrow \mathrm{enc}_k(m)\big] \\ &= \frac{1}{4}\sum_{k\in K} \mathsf{P}_K(k) \cdot \mathsf{P}\big[c' \leftarrow \mathrm{enc}'_k(m)\big] = \frac{1}{4} O_{\mathcal{E}'}(c', m) \\ \mathsf{P}_C(c'1) = O_{\mathcal{E}}(c'1, m) &= \sum_{k\in K} \mathsf{P}_K(k) \cdot \mathsf{P}\big[c'1 \leftarrow \mathrm{enc}_k(m)\big] \\ &= \frac{3}{4}\sum_{k\in K} \mathsf{P}_K(k) \cdot \mathsf{P}\big[c' \leftarrow \mathrm{enc}'_k(m)\big] = \frac{3}{4} O_{\mathcal{E}'}(c', m)\end{aligned}$$

Since $O_{\mathcal{E}'}(c', m) \neq 0$, we have $\mathsf{P}_C(c'0) = \frac{1}{4} O_{\mathcal{E}'}(c', m) \neq \frac{3}{4} O_{\mathcal{E}'}(c', m) = \mathsf{P}_C(c'1)$, which shows that the ciphertexts $c'0$ and $c'1$ have different probabilities, which shows that the ciphertext distribution is not uniform, as desired.

Exercise 3.5

Prove that decryption can always be derandomized in every encryption scheme $\mathcal{E}$. Formally, given encryption scheme $\mathcal{E} = (\mathrm{gen}, \mathrm{enc}, \mathrm{dec})$ with $\mathrm{dec}\colon R \times K \times C \to M$, prove that $\mathrm{dec}(r,k,c) = \mathrm{dec}(r',k,c)$ for all $r, r' \in R$, every possible key $k \in \mathrm{ran}(\mathrm{gen})$, and every suitable ciphertext $c \in C$; i.e., there exists $m \in M$ such that $\mathsf{P}\big[c \leftarrow \mathrm{enc}_k(m)\big] \neq 0$. In other words, decryption does not depend on the random bit tape. Mind that $\mathcal{E}$ need not be perfectly secret.

Solution: Let $k \in \mathrm{ran}(\mathrm{gen})$ and $r, r' \in R$ be arbitrary and $c \in C$ a ciphertext that is possible for this key; i.e., there exists $m \in M$ such that $\mathsf{P}\big[c \leftarrow \mathrm{enc}_k(m)\big] \neq 0$. Then by the definition of correctness we need to have

$$\mathrm{dec}(r,k,c) = m = \mathrm{dec}(r',k,c)$$

as required.

For those that are interested: $k \in \mathrm{ran}(\mathrm{gen})$ indeed implies $P(k) \neq 0$; i.e., that the key k is possible. Here is a quick proof. Suppose that $k \in \mathrm{ran}(\mathrm{gen})$; i.e., there exists $r \in R$ such that $\mathrm{gen}(r) = k$. Since gen is computable and k is returned, the corresponding Turing machine $\mathcal{A}$ computing 'gen' terminates after ℓ steps when run on r. In ℓ steps, the Turing machine can inspect at most the ℓ tape cells $r(0) \cdots r(\ell-1)$ of r and all remaining tape cells $r(\ell)r(\ell+1)\cdots$ of r do not influence the final result; i.e., the same key is returned for all random tapes that start with the prefix $r(0) \cdots r(\ell-1)$ of length ℓ. Avoiding the definition of a suitable space and measure, it should be obvious (by a suitable partition of the space) that the probability of all $r \in R$ with this prefix is exactly $2^{-\ell}$ and thus $P(k) \geq 2^{-\ell}$ (other prefixes might also generate the key k).

Exercise 3.6

Is one-time pad still perfectly secret if multiple messages are encrypted with the same key? Justify your answer, so either prove or refute perfect secrecy. If it is not perfectly secret, then also answer whether realistic attacks are possible and outline how you would proceed to break such a cipher.

Solution: Obviously not. The name one-time pad strongly suggests this answer. Indeed key reuse is relatively easy to detect in one-time pad (which is the reason why so many "two-time" pads were detected and broken). Suppose that we encrypt messages $m, m' \in M = \{0,1\}^\ell$ with one-time pad using key $k \in K = \{0,1\}^\ell$. We eavesdrop on the generated ciphertexts

$$c = m \oplus k \qquad \text{and} \qquad c' = m' \oplus k \ .$$

If we suspect that these two ciphertexts were generated using the same key, then we can XOR the ciphertexts and obtain

$$c \oplus c' = (m \oplus k) \oplus (m' \oplus k) = (m \oplus m') \oplus (k \oplus k) = m \oplus m' \ ,$$

which essentially cancels the key. The XOR of the ciphertexts is then the XOR of the original messages (without any key impact anymore). Since none of the operands is now uniform (both are English messages), the resulting distribution need not be uniform and indeed the distribution of XOR-ed English messages is easy to recognize (for example, for ASCII coding the first bit of standard letters is universally 0, so that would also be the case in the XOR of such messages).

Let us now formally prove our answer. To this end, we first need to model our two-time pad. To this end, let $\ell \in \mathbb{N}_+$ and $(\mathrm{gen}, \mathrm{enc}, \mathrm{dec})$ be our standard one-time pad encryption scheme with key space $K = \{0,1\}^\ell$, message space $M = \{0,1\}^\ell$, and ciphertext space $C = \{0,1\}^\ell$. We construct the two-time pad encryption scheme $\mathcal{E}' = (\mathrm{gen}, \mathrm{enc}', \mathrm{dec}')$ with the same key space K, message space $M' = MM = \{mm' \mid m, m' \in M\} = \{0,1\}^{2\ell}$, and ciphertext space $C' = CC = \{cc' \mid c, c' \in C\} = \{0,1\}^{2\ell}$. Key generation remains the same and encryption (as well as decryption) $\mathrm{enc}' = \mathrm{dec}'$ are given for every $k \in K$ and $m, m' \in M$ by

$$\mathrm{enc}'_k(mm') = \mathrm{enc}_k(m)\ \mathrm{enc}_k(m') \ .$$

Correctness remains obvious. We can now quickly argue that

$$2^\ell = |K| < \left|M'\right| = |M|^2 = 2^{2\ell} = 2^\ell \cdot 2^\ell \ , \qquad \text{(since } \ell \geq 1\text{)}$$

which by Shannon's theorem, yields that $\mathcal{E}'$ cannot be perfectly secure (key space smaller than message space).

How do we practically break two-time pad? Naturally we could guess messages and XOR them to check whether this combination of messages was the right choice, but this naturally scales terribly. The standard method is called "crib dragging" and assumes that you guess a short sequence (the crib) that occurs in one of the messages. This crib is then "dragged" across the XOR of the messages, which is the XOR of intercepted ciphertexts, to identify positions where it "fits" (i.e., occurs in one of the messages as indicated by obtaining a reasonable letter sequence). Let us illustrate this on a quick example, in which we now use alphabet position encoding (`A` being 1, `B` being 2, etc. with space `_` being 0), which we then code in the usual way in 5 bits, but present here as numbers or letters for ease of readability. In other words, the letter T is at position 20 in the alphabet, which corresponds to bit sequence 10100. Numbers outside the letter range, like 29 are presented as $^2_9$ in order to preserve the alignment. (As discussed in the exercise, modular addition works in principle as illustration, but elements are not self-inverse in that case, which is the case for XOR; modular subtraction can also be used for the illustration, but modular subtraction naturally is neither commutative nor associative, so the earlier steps would not be legal in this setup) Suppose that the XOR of intercepted ciphertexts is $\mathtt{OP}^3_1\mathtt{AS}^2_9\mathtt{FSTAHN}^{32}_{17}\mathtt{BIE}^{22}_{98}\mathtt{HNH}$. Dragging the crib "`_THE_`" across it we obtain for example

$\mathtt{OP}^3_1\mathtt{AS}^2_9\mathtt{FSTAHN}^{32}_{17}\mathtt{BIE}^{22}_{98}\mathtt{HNH}$	$\mathtt{OP}^3_1\mathtt{AS}^2_9\mathtt{FSTAHN}^{32}_{17}\mathtt{BIE}^{22}_{98}\mathtt{HNH}$	$\mathtt{OP}^3_1\mathtt{AS}^2_9\mathtt{FSTAHN}^{32}_{17}\mathtt{BIE}^{22}_{98}\mathtt{HNH}$
`_THE_`	`_THE_`	`_THE_`
`ODWDS`	$\mathtt{PKIV}^2_9$	`NKSGI`

The third example seems promising, so we guess that the word is "Thanksgiving" and partially decrypt.

$$\begin{array}{r} \mathrm{OP}^{3}_{1}\mathrm{AS}^{2}_{9}\mathrm{FSTAHN}^{32}_{17}\mathrm{BIE}^{22}_{98}\mathrm{HNH} \\ \mathtt{_THANKSGIVING_} \\ \hline \mathtt{S_II_THE_STRON} \end{array}$$

At this point we can completely guess one of the messages “Augustus II the Strong” and also decrypt the other message to “Next is Thanksgiving o”.