+1/1/60+

EPFL

1

Teacher : Prof. M. Gastpar
COM-102 (Advanced ICC II) - MA
June 20, 2024
Duration : 180 minutes

Student 1

SCIPER : **999000**

Do not turn the page before the start of the exam. This document is double-sided, has 16 pages, the last ones possibly blank. Do not unstaple.

- Place your student card on your table.
- **No other paper materials** are allowed to be used during the exam.
- Using a **calculator** or **any electronic device** is not permitted during the exam.
- The exam has **4 parts.** The fourth part is a single, comprehensive open problem (containing three sub-problems).
- For each question there is **exactly one** correct answer. We assign **negative points** to the **wrong answers** in such a way that a person who chooses a wrong answer loses **25 %** of the points given for that question.
- Use a **black or dark blue ballpen** and clearly erase with **correction fluid** if necessary.
- If a question is wrong, the teacher may decide to nullify it.

Respectez les consignes suivantes | Observe this guidelines | Beachten Sie bitte die unten stehenden Richtlinien		
choisir une réponse | select an answer Antwort auswählen	ne PAS choisir une réponse | NOT select an answer NICHT Antwort auswählen	Corriger une réponse | Correct an answer Antwort korrigieren
ce qu'il ne faut **PAS** faire | what should **NOT** be done | was man **NICHT** tun sollte		

Pour votre examen, imprimez de préférence les documents compilés à l'aide de auto-multiple-choice.

First part: Entropy and Source Coding

Question 1

[2 Points] In a binary Huffman code, symbols with equal probability always have codewords of equal length.

☐ TRUE ☐ FALSE

Question 2:

[6 Points] About a binary source code for a source whose alphabet has 5 letters, we only know the codeword lengths: $\ell_1 = 1, \ell_2 = 2, \ell_3 = 4, \ell_4 = 5, \ell_5 = 6$.
Answer the following true/false questions.

The average codeword length of this source code could be equal to the entropy of the source.

☐ TRUE ☐ FALSE

It is possible that this source code is uniquely decodable.

☐ TRUE ☐ FALSE

This source code could be a Huffman code.

☐ TRUE ☐ FALSE

Question 3

[4 Points] Consider a random variable $X \in \mathcal{X}$ and denote its alphabet size by $M = |\mathcal{X}|$. The symbol probabilities are not known. A binary Huffman code has been produced for this source. Now, suppose we observe a string of M letters : The string contains *each symbol of the source alphabet exactly once.* We compress the string with the given binary Huffman code. What is the maximum number of bits the binary Huffman compressor could produce for the entire string? Note: $\lceil x \rceil$ is rounding x up to the nearest integer.

☐ $2^{M\lceil \log_2 M \rceil}$
☐ $\lceil M \log_2 M \rceil$
☐ We do not have enough information.
☐ $\frac{1}{2}M(M+1) - 1$
☐ $M\lceil \log_2 M \rceil - 1$

Question 4

[3 Points] Alice and Bob are playing an outfit guessing game. Alice uniformly randomly chooses a top, a bottom and a pair of shoes from her wardrobe. Alice's wardrobe has the following unique items:

- Tops: 3 Shirts, 4 Pullovers
- Bottom: 4 Jeans, 2 Skirts
- Shoes: 2 Pairs of Sneakers (sometimes *baskets* in French), 1 Pair of Boots

Bob needs to find out Alice's outfit precisely, including which of the 4 pullovers or which of the 3 shirts she's wearing, and so on. What is the minimum required number of Yes/No questions such that Bob is guaranteed to find out Alice's outfit? Note that arbitrarily complex questions are allowed — no restrictions — but the answer can only be either Yes or No.

☐ 6 ☐ 9 ☐ 4
☐ 5 ☐ 7 ☐ 8

Question 5

[3 Points] Continued from previous question: Alice wants to give the following clue to reduce the number of questions that Bobs needs to ask:

"If I am wearing a pullover, I am also wearing a skirt and boots."

Given the clue, what is the minimum required number of Yes/No questions such that Bob is guaranteed to find out Alice's outfit?

- ☐ 7
- ☐ 4
- ☐ 8
- ☐ 9
- ☐ 6
- ☐ 5

Question 6

[2 Points] Consider a certain source symbol $x \in \mathcal{X}$. Claim: the length of the codeword corresponding to x in an optimal binary code (optimal in the sense of average codeword length) must be smaller or equal to the length of the codeword corresponding to x in the binary Shannon-Fano code.

- ☐ TRUE
- ☐ FALSE

Second part: Cryptography

Question 7
[4 Points] How many $x \in \{0, 1, 2, \ldots, 34\}$ satisfy the equation $x^2 - 5x + 4 \mod 35 = 0$?

- ☐ 4
- ☐ 0
- ☐ 1
- ☐ 2

Question 8:

[8 points] Alice and Bob are studying cryptography for the first time. They each did a project and each got a score. The score is out of 30 points. They represent it with 5 bits, using the natural binary representation. They want to tell their project scores to their friend Charlie, using the one-time pad. Charlie has a single uniformly sampled binary string K of length 5. He sends this *same* string to both Alice and Bob. Nobody else ever gets to know K.
Class policy dictates that students get a grade of 6 on the project if they score more than 27 points and a grade of 5.75 if they score more than 23 points.
Several people have partial information and try to infer Alice's and Bob's grades:

- David intercepts only Bob's transmission.
- Eve intercepts both Alice's and Bob's transmissions.
- Frank does not intercept anything, but he saw Bob very happy on finding his project score. So Frank knows that Bob received a grade of 6.

David can tell if Bob got a grade of 6 on the project

☐ TRUE ☐ FALSE

Eve can tell for sure if either of them received full points.

☐ TRUE ☐ FALSE

Eve can tell who scored more points between Alice and Bob.

☐ TRUE ☐ FALSE

Frank can help Eve infer if Alice got a grade of 5.75 or more.

☐ TRUE ☐ FALSE

Question 9
[2 Points] If a finite commutative group contains an element that is its own inverse but is *not* the identity element, then the group must have an even number of elements.

☐ TRUE ☐ FALSE

Question 10
[4 Points] How many elements does the multiplicative group $(\mathbb{Z}/2535\mathbb{Z}^*, \cdot)$ have?

- ☐ 845
- ☐ 1248
- ☐ 0
- ☐ 1252

Question 11
[4 Points] Which of the following is **not** a valid RSA encoding exponent for the modulus $m = 3403 = 83 \times 41$?

- ☐ 999
- ☐ 949
- ☐ 7
- ☐ 943

Question 12
[3 Points] Find $4^{14553} \mod 13$.

- ☐ 12
- ☐ 3
- ☐ 9
- ☐ 2

Question 13
[4 Points] Let $(m_1 = p_1q_1, e)$ be the public parameters of an RSA cryptosystem, with (secret) decrypting exponent d_1. Moreover, let $(m_2 = p_2q_2, e)$ be the public parameters of another RSA cryptosystem, with (secret) decrypting exponent d_2. We assume that p_1, q_1, p_2 and q_2 are four distinct prime numbers.
Alex has encrypted the plaintext t with the RSA cryptosystem with public parameters $(m = p_1q_2, e)$ to form the cryptogram c. Which of the following is a correct decryption?
(Note: Here, we use $(a)_k^{-1}$ to denote the multiplicative inverse of a in the multiplicative group $(\mathbb{Z}/k\mathbb{Z}^*, \cdot)$ if it exists. If it does not exist, we set $(a)_k^{-1} = 0$.)

- ☐ $\left[a \cdot \left(c^{d_1} \mod p_1\right) + b \cdot \left(c^{d_2} \mod q_2\right)\right] \mod m$,
  where $a = q_2v$ and $b = p_1u$ for some integers u and v satisfying $p_1u + q_2v = 1$.
- ☐ $c^d \mod m$,
  where $d = d_1(q_2 - 1)(q_2 - 1)_{p_1-1}^{-1} + d_2(p_1 - 1)(p_1 - 1)_{q_2-1}^{-1}$.
- ☐ None of the given formulas work.
- ☐ $\left(c^{d_1} \mod m_1 + c^{d_2} \mod m_2\right) \mod m$

Question 14
[2 Points] A finite cyclic group with an even number of elements has exactly one element of order 2.

☐ TRUE ☐ FALSE

Third part: Channel Coding

Question 15

[4 points] Suppose we want to send a k-bit message, where we assume that $k = rc$ for two positive integers r and c. We can shape the $k = rc$ bits into a rectangular array with r rows and c columns. For example, if $k = 12$, then the array could be of size 1×12, 2×6, 3×4, 4×3, 6×2, or 12×1. The following table shows an example for the case of 3×4.

x_{11}	x_{12}	x_{13}	x_{14}	q_1
x_{21}	x_{22}	x_{23}	x_{24}	q_2
x_{31}	x_{32}	x_{33}	x_{34}	q_3
p_1	p_2	p_3	p_4	

Define q_i to be the parity bit of the message bits in row i of the array, i.e.,

$$q_i = x_{i1} + \cdots + x_{ic} \bmod 2$$

and p_j to be the parity bit of the message bits in column j of the array, i.e.,

$$p_j = x_{1j} + \cdots + x_{rj} \bmod 2$$

The codeword w is the concatenated sequence of the message bits $x_{11}, \ldots, x_{rc}$, the row parity bits, and the column parity bits:

$$w = [x_{11}, x_{12}, \ldots, x_{rc}, q_1, \ldots, q_r, p_1, \ldots, p_c]$$

w is then transmitted over the (binary) error channel. Which of the following statements is correct?

- ☐ The minimum distance of the rectangular parity code depends on the choice of the positive integers k and r.
- ☐ Suppose we use the 1×12 rectangular parity code. Suppose the error channel output has a single position j for which $x_{1j} + p_j = 1$, while all other column parity bits are consistent, and the row parity bit is consistent, too. Then, minimum distance decoding amounts to changing bit x_{1j} and nothing further.
- ☐ For $k \geq 2$, no error pattern of weight two or more can be corrected.
- ☐ For any $k \geq 1$, the rectangular parity code can correct any error pattern of weight 1.

Question 16

[4 Points] Suppose a Reed-Solomon code over $\mathbb{F}_{11}$ of length $n = 7$ and dimension $k = 3$ was used with $a_1 = 0, a_2 = 1, a_3 = 2, a_4 = 3, a_5 = 4, a_6 = 5$, and $a_7 = 6$. We intercept the following output of the erasure channel:

$$\vec{y} = (2, 3, 6, ?, ?, ?, ?)$$

Which one of the following sequences could be the codeword that generated this channel output?

- ☐ $\vec{c} = (2, 3, 6, 0, 7, 5, 5)$
- ☐ $\vec{c} = (2, 3, 6, 1, 3, 5, 5)$
- ☐ None of the listed sequences.
- ☐ $\vec{c} = (2, 3, 6, 0, 7, 5, 6)$

Question 17:

[8 points] Consider a binary linear (n, k) block code $\mathcal{C}_1$ with (full-rank) generator matrix G and (full-rank) parity check matrix H. The binary linear block code $\mathcal{C}_2$ has generator matrix H. Answer the following True/False questions.

G is a parity check matrix of $\mathcal{C}_2$.

☐ TRUE ☐ FALSE

$\mathcal{C}_1$ and $\mathcal{C}_2$ have no codewords in common.

☐ TRUE ☐ FALSE

The union of $\mathcal{C}_1$ and $\mathcal{C}_2$ is $\mathbb{F}_2^n$, that is, $\mathcal{C}_1 \cup \mathcal{C}_2 = \mathbb{F}_2^n$.

☐ TRUE ☐ FALSE

$\mathcal{C}_2$ is a binary linear $(n, n-k)$ block code.

☐ TRUE ☐ FALSE

Question 18

[4 points] Consider a binary linear code spanned by the following vectors:

$$\{v_1 = (110010), v_2 = (011100), v_3 = (111001), v_4 = (100101)\}$$

Which one of the following statements is correct?

- ☐ If we receive from the Binary Symmetric Channel (with crossover probability less than 0.5) a word (101111), to minimize the error probability, we will deduce the transmitted codeword is (101110)
- ☐ Every three columns in its parity-check matrix H in systematic form are linearly independent.
- ☐ The cardinality of the code is 16.
- ☐ If we receive from the erasure channel a word (?10?11), the transmitted codeword is (010011)

Question 19

[4 points] Consider a binary code $\mathcal{C}$ of length $n = 5$. The code is made up of all sequences that contain exactly three ones. Which of the following statements is true about this code?

- ☐ The minimum distance of $\mathcal{C}$ is 2.
- ☐ Overall there are 8 codewords in $\mathcal{C}$.
- ☐ It can correct every error with weight no more than 1.
- ☐ It is a linear code.

Fourth part: open questions

Answer in the empty space below. Your answer should be carefully justified, and all the steps of your argument should be discussed in details. If you have the correct answer, but do not show the full work, you will not receive full credit. Leave the check-boxes empty, they are used for the grading.

Question 20: *This question is worth* $3 \times 8 = 24$ *points.*

In this problem you are an eavesdropper and codebreaker! You eavesdrop on a communication and manage to overhear the following (noisy) channel output sequence:

$$\vec{y} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Your task in this problem is to find the original message that was transmitted.
Here is what you know about the system used for transmission:

- Source Coding: You know the original message was either '*foo*' or '*bar*'. You know that the original message was compressed letter-by-letter using a Huffman code, but you don't know which Huffman code. You only know that the Huffman code was designed for the symbol probabilities $p(X = r) = 0.5, p(X = o) = 0.25, p(X = a) = 0.125, p(X = f) = 0.0625, p(X = b) = 0.0625$.

- Encryption: The binary string coming from the Huffman code is now encrypted using RSA. This is a little tricky since RSA does not directly take in binary strings. Here is how they proceeded: The binary string is first cut into chunks of 4 bits each. Each chunk is converted into an integer following the usual binary expansion rule, and then adding '1' (so that we do not feed a '0' to RSA encryption), as in the following table:

chunks of 4 bits of source coding output	input to RSA	RSA output	towards input to channel code
0000	1	1	00001
0001	2	2	00010
0010	3	3	00011
⋮	⋮	⋮	⋮
1111	16	20	10100

  The integer "input to RSA" is encrypted with RSA with public parameters $(m = 21, e = 5)$. The output of the RSA encryption is an integer between 1 and 20 (since we never use the input '0'). We convert this integer into 5 bits as in the above table (using standard binary representation). In this way, note that if the original string (source coding output) was $2 \times 4 = 8$ bits long, then the output of the RSA encryption will be $2 \times 5 = 10$ bits long.

- Channel Coding: From the RSA stage, we get 10 bits. They simply append a '0' at the end so as to make it 11 bits. Then, what was used is the *systematic* $(15, 11)$ Hamming code, specifically with the following parity check matrix:

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\vec{y} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

  where we have included, purely for your convenience, again the noisy channel output sequence $\vec{y}$.

0 .5 1 .5 2 .5 3 .5 4
.5 5 .5 6 .5 7 .5 8

(a) [8 Points] *Reprinted for your convenience*
Channel Coding: From the RSA stage, we get 10 bits. They simply append a '0' at the end so as to make it 11 bits. Then, what was used is the *systematic* $(15, 11)$ Hamming code, specifically with the following parity check matrix:

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\vec{y} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

where we have included, purely for your convenience, again the noisy channel output sequence $\vec{y}$.

(b) [8 Points] *Reprinted for your convenience*

Encryption: The binary string coming from the Huffman code is now encrypted using RSA. This is a little tricky since RSA does not directly take in binary strings. Here is how they proceeded: The binary string is first cut into chunks of 4 bits each. Each chunk is converted into an integer following the usual binary expansion rule, and then adding '1' (so that we do not feed a '0' to RSA encryption), as in the following table:

chunks of 4 bits of source coding output	input to RSA
0000	1
0001	2
0010	3
⋮	⋮
1111	16

RSA output	towards input to channel code
1	00001
2	00010
3	00011
⋮	⋮
20	10100

The integer "input to RSA" is encrypted with RSA with public parameters $(m = 21, e = 5)$. The output of the RSA encryption is an integer between 1 and 20 (since we never use the input '0'). We convert this integer into 5 bits as in the above table (using standard binary representation). In this way, note that if the original string (source coding output) was $2 \times 4 = 8$ bits long, then the output of the RSA encryption will be $2 \times 5 = 10$ bits long.

Backup plan for Part *(b):* [Maximum 7 points] If you could not solve Part *(a)*, you can do Part *(b)* assuming that the answer to Part *(a)* was $0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0$ (which are the 11 original transmitted bits making up the input of channel coding). Recall that the last '0' was only appended to get to 11 bits and can thus be removed.

0 .5 1 .5 2 .5 3 .5 4
.5 5 .5 6 .5 7 .5 8

(c) [8 Points] *Reprinted for your convenience*
You know the original message was either '*foo*' or '*bar*'. You know that the original message was compressed letter-by-letter using a Huffman code, but you don't know which Huffman code. You only know that the Huffman code was designed for the symbol probabilities $p(X = r) = 0.5, p(X = o) = 0.25, p(X = a) = 0.125, p(X = f) = 0.0625, p(X = b) = 0.0625$.
Backup plan for Part *(c):* [Maximum 7 points] If you could not solve Part *(b)*, you can do Part *(c)* assuming that in Part *(b)*, RSA decrpytion gave you the integers 2 and 6 (in this order).
(*Note:* If you solved Part *(b)* with the "Backup plan" and continue with your solution from Part *(b)*, then you can get a score of up to 8 Points for Part *(c)*.)

Pour votre examen, imprimez de préférence les documents compilés à l'aide de auto-multiple-choice.