QUANTUM CRYPTOGRAPHY
Master of Logic, University of Amsterdam, 2017
TEACHERS: Jan Czajkowski, Yfke Dulek, and Christian Schaffner

Practice problem set 5

You do not have to hand in these exercises, they are for practicing only.

Problem 1: Generating key using an anonymous message board (Problem 1.1)

Imagine that Alice and Bob have discovered an anonymous message board in the hallway. It allows both Alice and Bob to post messages in such a way that nobody can ever find out who the message came from. In particular, any eavesdropper Eve cannot learn whether the message came from Alice or from Bob. The message board simply creates a list of messages posted to it, without indicating a sender. We further assume that only Alice and Bob can write on the board! That is, Eve can observe all messages that appear but she can not alter them or write her own messages.

Can you think of a way that Alice and Bob can use the anonymous message board to exchange a key? I.e., at the end of the day, we want that Alice and Bob both share an n-bit key, but Eve is ignorant about the key. Which of the following protocols generates key? (There is only one right answer)

Protocol 1:

- Alice and Bob write a random bit on the board
- If the bit of Alice is the same as the bit of Bob then they erase and start from point 1.
- If the bit of Alice is different than Bob's bit then the next bit of their key is Alice's bit.
- Alice or Bob erase the bits and repeat from point 1 until they have n bits of key

Protocol 2:

- Alice starts by writing two bits on the board
- If the second bit is 0 they take the first bit as a key bit and they repeat point 1
- If the second bit is 1 they take the XOR of the two bits as a key bit and start from point 1 but now Bob writes instead of Alice.
- Alice or Bob execute this alternating protocol until they have n bits of key

Protocol 3:

- Alice and Bob each write $k \leq n$ random strings of n bits on the board in a random order
- If Alice sees one of her strings followed by a Bob string she XOR's the two strings.
- If Bob sees one of his strings followed by an Alice string he XOR's the two strings.
- Alice and Bob toss all strings that were never XOR'ed.
- Alice and Bob XOR all remaining strings together thus obtaining n bits of key.

Problem 2: Key rate with special channels

(a) **Problem 2.1** In class you saw how Alice and Bob could establish key in the presence of a limited Eve. In particular you saw a situation where Alice and Bob possessed a channel which allowed them to send classical bits such that Eve would obtain the bit with probability q (which is known to Eve!) and would obtain the flipped bit with probability $1-q$. Just to check if you were paying attention, please calculate the amount of min entropy Eve would have about a bit that Alice sent to Eve for the following values of q (up to two decimal places): $q = 1/4$, $q = 3/5$, $q = 9/10$.

(b) **Problem 2.2** For which values of q would we be able to use this channel to create keys?

(c) **Problem 2.3** Now imagine we are in the situation where Eve has a limited classical memory of size k bits. Imagine Alice sends Bob n bits through a public channel (of which Eve can copy and store k). Let's take for example $k = 1000$. What would Eve's min-entropy be (about the string of n bits) in the following situations? (enter a number!). I case of $n < k$, $n = k$, and $n = 10k$.

Problem 3: Information reconciliation

(a) **Problem 3.1** In class you saw an information reconciliation protocol based on the parity check matrix

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

This protocol could reliably correct a single bit flip error on blocks of three bits. If we assume that every key bit distributed is flipped with probability p and remains unchanged with probability $1 - p$ we could derive that the probability of correctly distributing a three bit string without error correction was $p_{succ} = (1-p)^3$ while using the error correction scheme based on H we had $p_{succ} = 1 - 3p^2 + 2p^3$ which is of course quite a bit better for small p. Now the question is, can we do even better? Here we will look at a simple expansion of the three bit linear code from class and look at the seven bit code generated by the parity check matrix

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{2}$$

This code, which can also correct a single error, can also be found in the Julia labs. Here we will investigate the robustness of this code to errors. Let's set a baseline by looking at the probability of successfully distributing a seven bit string using no error correction when all bits in the string are affected by a binary symmetric channel which flips bits with probability p. What is the probability of successfully distributing an error-free string?

(b) **Problem 3.2** Of course this code is not magical, i.e. we will never be able to reliably correct all errors. To see why this is the case let is look at the error strings $S = 1000000$ and $S' = 0110000$. Can we correct both S and S', what are the syndromes of those error strings?

(c) **Problem 3.3** Now, assuming we use the information reconciliation scheme from the videos with the matrix H and a string of seven bits. Assuming the probability of flipping a bit is again given by p, and we can reliably correct single bit errors, what is the probability that we can successfully distribute an error free key?