Quantum Cryptography (spring 2019) Dominique Unruh

Exercise Sheet 8

Out: 2019-05-17 Due: 2019-05-24

Problem 1: Missing claims from QKD proof

(a) **(Bonus problem)** In the practice we showed (or will show) that in our QKD protocol, after the Bell test and after measuring the n-bit raw key, we have

$$H_\infty(K_A|E)_{\rho_{raw}} \geq -\log(N2^{-n})$$

where $N := |\{xy \in \{0,1\}^{2n} : |xy| \leq t\}|$. (Note: $|xy|$ does not refer to the Hamming weight of xy here, but to the number of non-00 bitpairs.)

Show that $N \leq (3n+1)^t$.

(b) In the lecture, we claimed that if $\rho \in S^{\text{test}}_{\text{Ideal}}$, and we measure A's and B's system in the computational basis, then with probability 1, we have $|K_A \oplus K_B| \leq t$.

Show that this is true.

Problem 2: Inverting cyclic functions

Consider a function $H : [N] \to [N]$ where $[N] := \{0, \dots, N-1\}$. Let $H^i(x)$ denote $H(H(H(\dots H(x)\cdots)))$ (applied i times). For the sake of this problem, we call H cyclic if there exists a value p (the period) such that for all x, $H^p(x) = H(x)$.

(a) Let $U_H|x\rangle|i\rangle|0\rangle = |x\rangle|i\rangle|H^i(x)\rangle$. Give a quantum algorithm involving U_H for finding the period of H (assuming that H is cyclic).

Note: You may assume that the DFT D_N can be implemented as a polynomial-time[1] quantum circuit. (This is, in general, not true for all N. But in the general case, you would be able to use an approximately solution that is only slightly more complicated than the solution needed here.)

Note: "involving U_H" means that you can apply U_H in a single runtime step.

(b) **(Bonus problem)** Given $y = H(x)$ and given the period of p, show that you can find x in polynomial-time. (You may still use U_H.)

(c) The following statement is wrong:

[1] By polynomial-time, I mean that the size of the circuit is bounded by $p(\log N)$ for some polynomial p.

Given a cyclic H and a value $y \in \operatorname{range} H$, using the algorithm from (a), we can find the period p of H, and then using the algorithm from (b), we can compute $H^{-1}(y)$.[2] Moreover, all involved algorithms run in polynomial-time. Hence using quantum computers, cyclic functions can be inverted in polynomial-time.

Why?

[2]Notice that cyclicity implies bijectivity, so H^{-1} is well-defined.