

CSCE689: FDNS of Post-Quantum Crypto

Homework 1

Texas A&M U, Fall 2018
Lecturer: Fang Song

October 11, 2018; *Update: Oct. 21*
Due: October 30, 2018

Instructions. Your solutions will be graded on *correctness* and *clarity*. You should only submit work that you believe to be correct; if you cannot solve a problem completely, you will get significantly more partial credit if you clearly identify the gap(s) in your solution. For this problem set, a random subset of problems will be graded. You may keep working on bonus problems till November 13.

You may collaborate with others on this problem set However, you must *write up your own solutions* and *list your collaborators* for each problem.

1. (10 points) (Birthday bound) Fix a positive integer N , and $q \leq \sqrt{2N}$. Choose elements y_1, \dots, y_q uniformly and independently at random from a set of size N . Show that the probability that there exist distinct i, j with $y_i = y_j$ is $\Theta(q^2/N)$. (Note: you need to prove both lower and upper bounds.)
2. (Density matrix) Given a state $|\psi\rangle$, we introduce a new representation, *density matrix*, defined by $\rho := |\psi\rangle\langle\psi|$. In general, given an ensemble of states $\{p_i, |\psi_i\rangle\}_{i=1}^k$ on a quantum register, where the register is in $|\psi_i\rangle$ with probability p_i , we define its density matrix as $\rho = \sum_{i=1}^k p_i |\psi_i\rangle\langle\psi_i|$. We call a state pure if its density matrix ρ can be written as $|\psi\rangle\langle\psi|$ for some $|\psi\rangle$. Otherwise we say it is a *mixed* state.
 - (a) (Exercise) Compute the density matrices of $|0\rangle, |+\rangle, \frac{1}{\sqrt{3}}|0\rangle + \frac{e^{i\pi/8}\sqrt{2}}{\sqrt{3}}|1\rangle$.
 - (b) (5 points) A density matrix ρ corresponds to a pure state if and only if $\rho = |\psi\rangle\langle\psi|$. Show that ρ corresponds to a pure state if and only if $\text{Tr}(\rho^2) = 1$.
 - (c) (5 points) Show that every 2×2 density matrix ρ can be expressed as an equally weighted mixture of pure states. That is $\rho = \frac{1}{2}|\psi_1\rangle\langle\psi_1| + \frac{1}{2}|\psi_2\rangle\langle\psi_2|$ (Note: the two states need not be orthogonal).
3. (Grover's Search)
 - (a) (10 points) (Claim in lower bound Proof) Let $O_r : |x, y\rangle \mapsto |x, y \oplus f_r(x)\rangle$, where $\forall x \in \{0, 1\}^n, f_r(x) = 1$ iff. $x = r$. $O_\emptyset : |x, y\rangle \mapsto |x, y \oplus f_\emptyset(x)\rangle$, where $\forall x \in \{0, 1\}^n, f_\emptyset(x) = 0$. Let $A = A_k O A_{k-1} \dots A_1 O A_0$ be a k -query quantum algorithm. For $j = 0, \dots, k$, define

$$|\psi_r^{(j)}\rangle = O_r A_{j-1} \dots O_r A_0 |0^n\rangle, \quad |\phi^{(j)}\rangle = O_\emptyset A_{j-1} \dots O_\emptyset A_0 |0^n\rangle;$$

$$D_r^{(j)} := \|\psi_r^{(j)} - \phi^{(j)}\|, \quad E_r^{(j)} := \|O_r |\phi^{(j)}\rangle - |\phi^{(j)}\rangle\|.$$

Show the following:

- $D_r^{(j)} \leq D_r^{(j-1)} + E_r^{(j-1)}, \forall j = 1, \dots, k.$
 - Suppose $|\phi^{(j)}\rangle = \sum_x \alpha_x^{(j)} |x\rangle$. Show that $E_r^{(j)} \leq 2|\alpha_r^{(j)}|, \forall j = 0, \dots, k.$
- (b) (10 points) (Multiple marked items) Given $f : \{0,1\}^n \rightarrow \{0,1\}$. Let $A = f^{-1}(1) = \{x \in \{0,1\}^n : f(x) = 1\}$ and $B = f^{-1}(0) = \{x \in \{0,1\}^n : f(x) = 0\}$. Suppose $|A| \geq 1$ is known. Given $O_f : |x, y\rangle \mapsto |x, f(x) \oplus y\rangle$. Design an algorithm that finds some $x \in A$ with $O(\sqrt{N/a})$ queries to O_f .
- (c) (Bonus 10pts. Unknown size of marked items) What if a is unknown in Part (b)? Design an algorithm that counts a (approximately).
- (d) (Bonus 10pts. Fine performance of quantum search) Let $f : X \rightarrow \{0,1\}$ be a function such that $|f^{-1}(1)| = a$. Describe a q -query quantum algorithm that finds a x s.t. $f(x) = 1$ with probability $\Omega(q^2 a/N)$. (NB. this is also optimal.)
4. (Reduction and Hybrid argument) In this problem, we practice security proofs in cryptography. Let $G : \{0,1\}^n \rightarrow \{0,1\}^{n+1}$ be a pseudorandom generator that expands the seed by 1 bit.
- (a) (10 points) (Multi-sample security i.e., Parallel composition) Consider an adversary A given (r_1, \dots, r_k) generated in one of two ways below:
- i) Pick $s_1, \dots, s_k \leftarrow \{0,1\}^n$ independently and uniformly at random, and output $r_i = G(s_i)$ for $i = 1, \dots, k$.
 - ii) Pick $r_1, \dots, r_k \leftarrow \{0,1\}^{n+1}$ independently and uniformly at random, and output them.
- Show that no efficient A can distinguish the two cases. Namely the parallel composition of G , $G' = G \parallel \dots \parallel G$, is also a PRG.
- (b) (10 points) (Sequential composition) Let $z[i]$ denotes the i th bit of a string z . Consider the construction G^k below for increasing the expansion of G (due to Blum and Micali): on random seed $s \leftarrow \{0,1\}^n$, let $r_0 = s$.
- i) For $i = 1, \dots, k$, compute $y_i := G(r_{i-1})$, and let $r_i = y_i[2, \dots, n]$.
 - ii) Output $r = y_1[1] \parallel y_2[1] \parallel \dots \parallel y_{k-1}[1] \parallel y_k[1] \in \{0,1\}^k$.
- Namely, at each iteration, we save one bit of the output of G and use the rest as a seed for the next invocation. We prove that G^k is a PRG for any polynomially bounded k (in particular we get a length-doubling PRG by setting $k = 2n$) by a *hybrid argument*. For $j = 1, \dots, k$, define H^j as a revised generator:
- i) Pick random $z_j \leftarrow \{0,1\}^j$ and $r_j \leftarrow \{0,1\}^n$.
 - ii) For $i = j+1, \dots, k$, compute $y_i := G(r_{i-1})$, and let $r_i = y_i[2, \dots, n]$.
 - iii) Output $r = z_j \parallel y_{j+1}[1] \parallel \dots \parallel y_{k-1}[1] \parallel y_k[1] \in \{0,1\}^k$.
- Let $H_0 = G^k$ be the original construction. Note that the output in H_k is a truly random $r \leftarrow \{0,1\}^k$.

Show that for all $j = 0, \dots, k - 1$,

$$\left| \Pr_{r \leftarrow H_j} [A(r) = 1] - \Pr_{r \leftarrow H_{j+1}} [A(r) = 1] \right| \leq \text{negl}(n)$$

holds for any efficient A . Conclude from this that G^k is a PRG. (Exercise: what is the advantage of G^k over the parallel G' from part a?)

- (c) (10 points) Let $f : X \rightarrow Y$ be a function. We say f is *one-way* if $f(x)$ can be computed efficiently (i.e., *poly*($|x|$)); but is hard to invert *on-average*, i.e.,

$$\Pr_{x \leftarrow X} [f(x') = f(x) : x' \leftarrow A(f(x))] \leq \text{negl}(|x|),$$

holds for any poly-time algorithm A (Note: it's crucial that x is generated randomly.) Construct a one-way function from the PRG G , and give a proof that it is one-way. Let $F = \{F_k : \{0, 1\}^n \rightarrow \{0, 1\}^n\}_{k \in \{0, 1\}^n}$ be a pseudorandom function family. Construct a one-way function from F .

- (d) (Bonus 10pts. WeakOWF to StrongOWF) The definition of one-way function above is strong in the sense that no efficient algorithm can invert it with non-negligible probability. We consider a weaker notion of one-way function, where we consider it secure as long as no efficient algorithm can invert with probability close to 1: for any poly-time A

$$\Pr_{x \leftarrow X} [f(x') = f(x) : x' \leftarrow A(f(x))] \leq 1 - 1/p(|x|).$$

for some polynomial $p(\cdot)$. Show that if there is a weak one-way function, then there is a strong one-way function. (Hint: given a weak one-way function f , consider $f' : (x_1, \dots, x_m) \mapsto (f(x_1), \dots, f(x_m))$.)

5. (Small-range distribution) Let D be a distribution on set Y . For an arbitrary set X of size N , consider D^X and $\text{SR}_r^D(X)$, which are both distributions on $\{f : X \rightarrow Y\}$. We claimed in class (without proof) that the output distributions of any q -query quantum algorithm to either $\text{SR}_r^D(X)$ or D^X are $O(q^3/r)$ -close.

- (a) (15 points) (Collision finding) We develop a quantum algorithm for finding collision in a function. Given a function $f : X \rightarrow Y$, we call a pair of inputs $(x \in X, x' \in X)$ a collision, if $x \neq x'$ and $f(x) = f(x')$. Let f be a function with k collisions. Consider the following algorithm that uses q queries:

- Pick a random subset $S \subseteq X$ of size $0 < q_1 \leq q$. Query all inputs in S *classically*, and find a collision in S .
- if there were none, then apply a quantum search algorithm to find collision between S and $X \setminus S$.

Describe how to implement the second step (i.e., building the Grover oracle). What is the success probability that the algorithm finds a collision? Optimize the choice of q_1 and q_2 . (Hint: Problem 2.d may be helpful.)

- (b) (15 points) (Quantum distinguisher) Describe a quantum algorithm that distinguishes $\text{SR}_r^D(X)$ from D^X with probability $\Omega(q^3/r)$.
- (c) (10 points) (Classical attack) Give a classical algorithm to distinguish $f \leftarrow \text{SR}_r^D(X)$ from a truly random function with constant probability (e.g., $> 1/4$) with as few queries as possible. (You do not need to prove its optimality, 5 bonus pts if you actually do.)
- (d) (Bonus 10pts. Statistical Oracle Indistinguishability) We have proved in class that if an *efficient* quantum algorithm distinguishes D_0^X and D_1^X with advantage ε , then one can distinguish D_1 and D_2 with advantage $\Omega(\varepsilon^2/q^3)$. Show that this holds *statistically*. Namely, if an *unbounded* quantum algorithm distinguishes D_0^X and D_1^X with advantage ε , then D_1 and D_2 must be $\Omega(\varepsilon^2/q^3)$ far.