

Homework is due by **11pm of Oct 15**. Send by email to both “regev” (under the cs.nyu.edu domain) and “ry849” (under the nyu.edu domain) with subject line “CSCI-GA 3210 Homework 4” and name the attachment “YOUR NAME HERE HW4.tex/pdf”. There is no need to print it. Start early!

1. (3 points) (*Rabin’s permutation*) Assume $p, q \equiv 3 \pmod{4}$. Does Rabin’s function remain one way when its domain is restricted to \mathbb{QR}_N^* (and so becomes a one way *permutation*)?
2. (PRGs.)¹ A (deterministic) function $G : \{0, 1\}^* \rightarrow \{0, 1\}^*$ is called a *pseudorandom generator* (PRG) with output length $\ell(n) > n$ if
 1. G can be computed by a PPT algorithm,
 2. $\forall n, x \in \{0, 1\}^n, |G(x)| = \ell(n)$, and
 3. $\{G(U_n)\}$ is computationally indistinguishable from $U_{\ell(n)}$, the uniform distribution on $\ell(n)$ bits.

Prove or disprove (giving the simplest counterexample you can find) the following statements. In constructing a counterexample, you may assume the existence of another OWF / PRG.

- (a) (4 points) Let G be a PRG with output length $\ell(n) > n$. The function $G'(s) = G(s) \oplus (s|0^{\ell(|s|)-|s|})$ is a PRG, where $|$ denotes concatenation. I need a hint! (ID 99102)
 - (b) (4 points) For a PRG f , define $g(x) = f(x)|f(\bar{x})$, where \bar{x} is the bit-wise negation of x . Then g is a PRG.
 - (c) (5 points) A PRG G with output length $\ell(n) = 2n$ is itself a one-way function. I need a hint! (ID 15489)
 - (d) (4 points) (extra credit) A PRG G with output length $\ell(n) = n + 1$ is itself a one-way function. I need a hint for 1 points! (ID 19634)
3. (a) (2 points) (*Computing square roots efficiently modulo prime*) Let $p > 2$ be a prime. Assume we are given a quadratic residue $x \in \mathbb{Z}_p^*$ and we wish to compute its (two) square roots. Show that when $p \equiv 3 \pmod{4}$, this can be done efficiently by computing $\pm x^{(p+1)/4}$, a formula due to Lagrange. (The case $p \equiv 5 \pmod{8}$ is a bit more difficult; the case of a general prime can also be done efficiently but is more involved; feel free to look it up and summarize it here!)
 - (b) (2 points) (*LSB is not hard.*¹) Show how given a prime $p > 2$, a generator g of \mathbb{Z}_p^* , and $g^x \pmod{p}$ for an unknown $x \in \{0, \dots, p-2\}$, we can efficiently decide if x is odd. (This shows that “least significant bit” [x is odd] is *not* a hard-core predicate for the modular exponentiation function $f_{p,g}(x) = g^x \pmod{p}$.)
 - (c) (2 points) Here is a sketch of an attempt to efficiently compute discrete logs (a problem believed to be hard). Complete the missing details and identify the bug.
We are given $y = g^x \pmod{p}$ for an unknown $x \in \{0, \dots, p-2\}$. Write $x = \sum_{j=0}^{\lceil \log p \rceil} 2^j b_j$ in its binary expansion. Efficiently find b_0 as above. Let $y_1 = y/g^{b_0}$ and notice that it is a quadratic residue. Compute the square root of y_1 , and continue recursively to recover all the bits of x .
4. (2 points) ♣ (*Using hard core predicates to construct PRGs*) We say that an efficiently computable function $h : \{0, 1\}^* \rightarrow \{0, 1\}$ is *hard-core* for a function f if for all non-uniform PPT algorithms \mathcal{A} ,

$$\Pr_{x \leftarrow \{0,1\}^n} [\mathcal{A}(f(x)) = h(x)] \leq \frac{1}{2} + \text{negl}(n).$$

¹A question from Peikert’s class

Assume we're able to show that a certain h is hard-core for a one-way *permutation* f . Suggest a way to construct a PRG from f and h , and try to think what the analysis would entail. (We'll do the analysis in class and in the next homework.)