University of Maryland
CMSC456 — Introduction to Cryptography
Professor Jonathan Katz

Problem Set 2

Due at *beginning* of class on Oct. 9

1. Let p be a prime, let x_1, x_2 be quadratic residues in $\mathbb{Z}_p^*$, and let y be a quadratic non-residue in $\mathbb{Z}_p^*$.

   (a) Prove that $x_1 x_2$ is a quadratic residue.

   (b) Prove that x_1^{-1} is a quadratic residue.

   (c) Prove that $x_1 y$ is *not* a quadratc residue.

2. Consider the following proposed definition of a one-way function. $f : \{0,1\}^* \to \{0,1\}^*$ is one-way if, for all PPT algorithms A the following is negligible:

$$\Pr[x \leftarrow \{0,1\}^k; y = f(x) : A(1^k, y) = x].$$

   Give a function which satisfies the above definition but does *not* satisfy the definition we gave in class for a one-way function. Do you think the definition proposed here is a good one?

3. Consider the following definition of a perfect pseudorandom generator (PRG). $G : \{0,1\}^k \to \{0,1\}^{k+1}$ is a perfect PRG if, for all algorithms A we have:

$$\Pr[x \leftarrow \{0,1\}^k; y = G(x) : A(y) = 1] = \Pr[y \leftarrow \{0,1\}^{k+1} : A(y) = 1].$$

   Show that a perfect PRG does not exist. Namely, for *any* proposed perfect PRG, give an explicit algorithm A for which the above definition of security is not true.

4. Discuss whether the functions G which follow are secure PRGs (in the sense defined in class, not the "perfect" sense described above). When G is a PRG, prove it. When G is not, give an explicit, efficient algorithm which "breaks" the PRG and distinguishes the output of G from random.

   (a) G defined by $G(x) = x \circ b$ where b is the parity of x.

   (b) Let G_1, G_2 be secure PRGs. Define G by $G(x) = G_1(x) \circ G_2(x)$ ($\circ$ denotes concatenation).

   (c) **Graduate students only.** Let G_1, G_2 be secure PRGs. Define G by $G(x_1 \circ x_2) = G_1(x_1) \circ G_2(x_2)$. Note the difference between this and the previous problem.

5. **Graduate students only.** You will prove that if G is a secure PRG, then no algorithm can predict the last bit of the output of G. More precisely, assume $G : \{0,1\}^* \to \{0,1\}^*$ is a secure PRG which stretches its input by one bit. For any string y of length ℓ, let $y_1 \cdots y_\ell$ denote the bits of y. We say that G is *unpredictable* if no PPT algorithm

can predict the last bit of the output of G (with more than probability $1/2$), given all the other bits. That is, for any PPT algorithm A the following is negligible:

$$\left|\Pr[x \leftarrow \{0,1\}^k; y = G(x) : A(y_1 \cdots y_k) = y_{k+1}] - 1/2\right|.$$

The structure of the proof is as follows: We will assume toward a contradiction that there is an efficient algorithm A for which

$$\left|\Pr[x \leftarrow \{0,1\}^k; y = G(x) : A(y_1 \cdots y_k) = y_{k+1}] - 1/2\right| > \epsilon(k)$$

and $\epsilon(\cdot)$ is not negligible. We then construct an efficient algorithm A' that can distinguish the output of G from random (i.e., A' "breaks" G) with probability which is not negligible. This contradicts the security of G as a PRG, implying that our original assumption is false, and hence no such A can exist.

We define A' as follows: on input $y = y_1 \cdots y_{k+1}$, A' runs $A(y_1 \cdots y_k)$. If $A(y_1 \cdots y_k)$ outputs y_{k+1}, then A' outputs 1. Otherwise, A' outputs 0.

- What is the probability that A' outputs 1 given that y is a random string?
- What is the probability that A' outputs 1 given that y is an output of $G(x)$ for some x (express your probability in terms of $\epsilon(k)$)?
- Complete the proof that if $\epsilon(\cdot)$ is not negligible, then A' breaks G with probability which is not negligible.