Introduction to Cryptography
Courant, Fall 2014

Homework 1

Instructor: Oded Regev
Student: YOUR NAME HERE

Homework is due by **noon of Sep 15**. Send both TEX and PDF by email to Oded and bring a printed copy to class. Start early!

Instructions. Solutions must be typeset in LaTeX (a template for this homework is available on the course web page). Your work will be graded on *correctness*, *clarity*, and *conciseness*. 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. It is good practice to start any long solution with an informal (but accurate) "proof summary" that describes the main idea.

You are expected to read all the hints either before or after submission, but before the next class.

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

1. *(Due by noon on Wed, Sep 10.)* Send email to Oded (`regev` at `cims`) with subject `CSCI-GA 3210 student` containing (1) a few sentences about yourself and your background (including your department, graduate program, how long in program), and (2) your comfort level with the following: mathematical proofs, elementary probability theory, big-O notation and analysis of algorithms, Turing machines, and P, BPP, NP, and NP-completeness. Please also mention any courses you've taken covering these topics.

2. (5 points) *(Shannon)* Prove that in any perfectly secret shared-key encryption scheme, $|\mathcal{K}| \geq |\mathcal{M}|$.

3. *(Perfect secrecy.[1])* Prove or disprove (giving the simplest counterexample you can find) the following statements about perfect secrecy for shared-key encryption. You may use any of the facts from class.
   (a) (1 point) There is a perfectly secret encryption scheme for which the ciphertext always reveals 99% of the bits of the key k to the adversary.
   (b) (2 points) There is an encryption scheme that is not perfectly secure, yet the adversary cannot guess the key with probability greater than $1/|\mathcal{K}|$.
   (c) (2 points) In a perfectly secret encryption scheme, the ciphertext is uniformly random. That is, for every $m \in \mathcal{M}$, the probability $\Pr_{k \leftarrow \mathsf{Gen}}[\mathsf{Enc}_k(m) = \bar{c}]$ is the same for every ciphertext $\bar{c} \in \mathcal{C}$.
   (d) (5 points) Perfect secrecy is equivalent to the following definition, which says that the adversary cannot determine which of two messages was encrypted any better than by random guessing. Formally, for any $m_0, m_1 \in \mathcal{M}$, and any function $\mathcal{A} : \mathcal{C} \to \{0,1\}$,
   $$\Pr_{k \leftarrow \mathsf{Gen},\, b \leftarrow \{0,1\}}[\mathcal{A}(\mathsf{Enc}_k(m_b)) = b] = \frac{1}{2}.$$
   (e) (5 points) Perfect secrecy is equivalent to the following definition, which says that the ciphertext and message are independent (as random variables). Formally, for any probability distribution $\mathcal{D}$ over the message space $\mathcal{M}$ and any $\bar{m} \in \mathcal{M}$ and $\bar{c} \in \mathcal{C}$,
   $$\Pr_{m \leftarrow \mathcal{D},\, k \leftarrow \mathsf{Gen}}[m = \bar{m} \wedge \mathsf{Enc}_k(m) = \bar{c}] = \Pr_{m \leftarrow \mathcal{D}}[m = \bar{m}] \cdot \Pr_{m \leftarrow \mathcal{D},\, k \leftarrow \mathsf{Gen}}[\mathsf{Enc}_k(m) = \bar{c}].$$

4. *(Encryption schemes with a computationally bounded adversary.)*

   Consider the scenario of an encryption scheme in which Alice wants to send a message to Bob in such a way that Eve, who monitors the transmission, cannot read the message.

[1] Based on a question from Peikert's class

(a) (1 point) Explain in one sentence why Bob needs to have a secret from Eve.

(b) (1 point) Explain in one or two sentences why Alice needs to have a secret from Eve.

(c) (2 points) Now assume that Eve is computationally bounded (i.e., is restricted to run in polynomial time in the length of the message). Does Bob still need to have a secret from Eve? Does Alice? (your feeling for the latter is enough)

I'm done solving and want to know more! (ID 18764)

5. *(Working with negligible functions.[1])* Recall that a non-negative function $\nu : \mathbb{N} \to \mathbb{R}$ is *negligible* if it decreases faster than the inverse of any polynomial (otherwise, we say that ν is *non-negligible*). More precisely, $\nu(n) = o(n^{-c})$ for every fixed constant $c > 0$, or equivalently, $\lim_{n\to\infty} \nu(n) \cdot n^c = 0$.

   State whether each of the following functions is negligible or non-negligible, and give a brief justification. In the following, $\mathrm{negl}(n)$ denotes some arbitrary negligible function, and $\mathrm{poly}(n)$ denotes some arbitrary polynomial in n. (If you are not comfortable with these notion, read Section 4.2 of Lecture 2 in Peikert's notes)

   (a) (1 point) $\nu(n) = 1/2^{100 \log n}$.

   (b) (1 point) $\nu(n) = n^{-\log\log\log n}$. (Compare with the previous item for "reasonable" values of n.)

   (c) (1 point) $\nu(n) = \mathrm{poly}(n) \cdot \mathrm{negl}(n)$. (State whether ν is *always* negligible, or not necessarily.)

   (d) (1 point) $\nu(n) = (\mathrm{negl}(n))^{1/\mathrm{poly}(n)}$. (Same instructions as previous item.)

   (e) (1 point)

$$\nu(n) = \begin{cases} 2^{-n} & \text{if } n \text{ is composite} \\ 100^{-100} & \text{if } n \text{ is prime.} \end{cases}$$

6. (2 points) *(Defining one-way functions.♣)* Next class we will define the notion of a *one-way function*. Informally, this is a function $f : \{0,1\}^* \to \{0,1\}^*$ that is (1) easy to compute, and (2) hard to invert.

   (a) Suggest a way (or ways) to formally define it. After thinking about this question for at least 10 minutes and before writing your solution, click here for some food for thought (ID 19166)

   Next, for each of the following functions, say if you think it's one way according to your definition.

   (b) The function that given an n-bit string outputs the same string with its first half zeroed out.

   (c) The function f on domain $\{1,\ldots,N\} \times \{1,\ldots,N\}$ that maps a pair (x, y) to their product xy.

   (d) Choose elements $a_1, \ldots, a_n$ uniformly from $\mathbb{Z}_N$ for $N = 2^n$ and define $f : \{0,1\}^n \to \mathbb{Z}_N$ by $f(b_1, \ldots, b_n) = \sum_{i=1}^n b_i a_i$.

   (e) Same as previous part, except $a_1, \ldots, a_n$ are chosen uniformly from $\mathbb{Z}_2^n$.

7. *(Error-correcting codes (optional, no credit).)* This is a bit off topic, but will give you a glimpse to an immensely important topic that also dates back to Shannon's seminal work. These ideas are used in pretty much all digital communication protocols: cell phones, Internet, satellites, etc.

   (a) Assume we choose $2^{n/20}$ strings from the set $\{0,1\}^n$ uniformly at random. Show that with positive probability (in fact, high probability) the Hamming distance (i.e., number of different coordinates) between *any* two strings in the set is more than $n/4$. I need a hint! (ID 84542)

♣Questions marked with a club are more open-ended and meant to encourage you to think in preparation for next class. You are not expected to answer correctly. Instead, you are expected to spend time thinking about it.

(b) Show how Alice can communicate to Bob a message of k bits by sending only $n = 20k$ bits in such a way that Bob can recover the message even if an adversary flips up to $n/8$ bits of the communication. Would simply repeating the message 20 times be good enough?

8. *(The group $\mathbb{Z}_p^*$) (Not yet for submission! Included here so you can start thinking about it, and also for you to get a sense of the level of math expected in this course)* Let p be an odd prime.

   (a) Find an efficient algorithm that given $a \in \mathbb{Z}_p^*$ and an integer $b \geq 0$ computes $a^b \in \mathbb{Z}_p^*$. Can we simply compute a^b as integers and then reduce the result modulo p?

   (b) Find an efficient algorithm to check if a given $a \in \mathbb{Z}_p^*$ is a quadratic residue.

   (c) What fraction of the elements of $\mathbb{Z}_p^*$ are generators? How does it behave asymptotically? (You can use Wikipedia for the latter)

   (d) Describe an efficient algorithm to check if a given $g \in \mathbb{Z}_p^*$ is a generator. Assume that the algorithm is also given a factorization of $p - 1$. (It is not known how to perform this task efficiently without this factorization.)

   (e) There is a known efficient algorithm that given a number n (in unary) outputs a uniform n-bit prime p, together with a generator g of $\mathbb{Z}_p^*$. How can that be in light of what we said earlier about the necessity of the factorization of $p - 1$? Explain the apparent paradox and suggest a solution.