CS 359C: Classics of Cryptography **Spring 2017**

Problem Set 2

Due: May 10, 2017, by 2:30pm (submit hard copy at the *beginning* of lecture)

Instructions: You must typeset your solution in LaTeX using the provided template:

`https://web.stanford.edu/class/cs359c/homework.tex`

Problem 1: Discrete Log Cryptanalysis [10 points]. Let $\mathbb{G}$ be a group of prime order q in which discrete log is hard. Let g be a generator of $\mathbb{G}$.

(a) You are given $h = g^x \in \mathbb{G}$. Your task is to recover $x \in \mathbb{Z}_q$. Show that finding a pair $(a, b) \in \mathbb{Z}_q^2$ such that $g^a = h^b \in \mathbb{G}$ and $a, b \neq 0$ is enough to recover x.

(b) Produce an algorithm for finding one such pair (a, b) that runs in time and space $\tilde{O}(\sqrt{q})$.

(c) You are given two vectors $(g_1, g_2, \ldots, g_n) \in \mathbb{G}^n$ and $(h_1, h_2, \ldots, h_n) \in \mathbb{G}^n$, where $h_i = g_i^{x_i}$ for $1 \leq i \leq n$. Your task is to recover $(x_1, x_2, \ldots, x_n) \in \mathbb{Z}_q^n$. Show that using a precomputed table of size $\tilde{O}(\sqrt{q})$, you can compute $(x_1, \ldots, x_n)$ in time $\tilde{O}(n\sqrt{q})$ with $O(\log q)$ additional space. Your solution should not require making changes to the precomputed table.

(d) **Extra Credit [3 points].** Modify your algorithm from part (b) to use time $\tilde{O}(\sqrt{q})$ time but only $\tilde{O}(1)$ space.

Problem 2: Hard-Core Bit of Discrete Log [10 points]. You are given a group $\mathbb{G}$ of prime order q (in which discrete log is hard), along with a generator g of $\mathbb{G}$. Let $\mathcal{O}(\cdot)$ be an oracle that takes as input $h = g^x \in \mathbb{G}$ and computes the least-significant bit of x.

(a) Show that it is possible to use $\mathcal{O}(\cdot)$ to compute the discrete log of an arbitrary group element in $\mathbb{G}$.

(b) How would you modify your algorithm from Part (a) if $\mathcal{O}$ is correct with probability $\varepsilon = 2/3$, where the probability is taken over the random coins of $\mathcal{O}$? That is, *for all* $h = g^x \in \mathbb{G}$, $\mathcal{O}(h)$ will give you the correct answer 2/3 of the time.

(c) **Extreme Extra Credit [5 points.]** Modify your algorithm to recover x if $\mathcal{O}(\cdot)$ only gives you a correct answer with probability 2/3 on 2/3 of the elements in $\mathbb{G}$. That is, for 1/3 of the elements $h \in \mathbb{G}$, $\mathcal{O}(h)$ will give you arbitrarily wrong answers. For the remaining 2/3 of the elements, $\mathcal{O}(h)$ will give you the right answer 2/3 of the time.

Your solution to this problem shows that computing the least significant bit of x given g^x is as hard as computing discrete logs in $\mathbb{G}$.

Problem 3: Fancy ElGamal [10 points]. Refer to the Boneh-Shoup textbook for a definition of CPA-security (semantic security against a chosen plaintext attack). Let $\mathbb{G}$ be a group of prime order q in which DDH is hard (in particular, $\log q = \text{poly}(\lambda)$, where λ is a concrete security parameter). Fix a generator g of $\mathbb{G}$. We define a public-key encryption scheme whose message and ciphertext spaces are both $\mathbb{G}^n$. We leave the security parameter implicit:

- $\mathsf{KeyGen}() \to (\mathsf{pk}, \mathsf{sk})$. Sample $\vec{a} = (a_1, \ldots, a_n) \xleftarrow{R} \mathbb{Z}_q^n$. Compute $\vec{A} = (g^{a_1}, \ldots, g^{a_n}) \in \mathbb{G}^n$. Output $(\mathsf{pk}, \mathsf{sk}) = (\vec{A}, \vec{a})$.
- $\mathsf{Encrypt}(\mathsf{pk}, (m_1, \ldots, m_n)) \to c$. Sample $b \xleftarrow{R} \mathbb{Z}_q$. Output $c = (g^b, m_1 \cdot A_1^b, m_2 \cdot A_2^b, \ldots, m_n \cdot A_n^b)$.

(a) Prove that this cryptosystem is CPA-secure assuming the DDH assumption holds in $\mathbb{G}$. If you solve the extra credit (Part (b)), you need only include one reduction, but please indicate this when writing up your solution.

(b) **Extra Credit [3 points].** Give a *tight* security reduction that this cryptosystem is CPA-secure assuming the DDH assumption holds in $\mathbb{G}$. Here, we say that a security reduction is "tight" if for every adversary $\mathcal{A}$ that breaks CPA-security of the scheme in time t and advantage ε, there exists an algorithm $\mathcal{B}$ that breaks the DDH assumption in $\mathbb{G}$ in time $t' = t \cdot n \cdot \text{poly}(\lambda)$ and advantage $\varepsilon' = \varepsilon / k$ for some constant k. Notably, the security loss (i.e., the reduction in the advantage of algorithm $\mathcal{B}$) is *independent* of the number of messages n.

Problem 4: Understanding Zero Knowledge [10 points].

(a) Give a protocol that satisfies completeness, soundness, and honest-verifier zero knowledge, but that is *not* zero knowledge.

(b) Give a protocol that is complete and sound but is zero knowledge if and only if factoring is in BPP.