CS355: Topics in cryptography **Spring 2014**

Assignment #1

Due: Tuesday, May. 13, 2014.

Problem 1: Baby Goldreich-Levin. Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ be a one-way permutation. Suppose that for an $x \in \mathbb{Z}_2^n$ we have an algorithm $\mathcal{A}_x$ such that $\Pr[\mathcal{A}_x(r) = \langle x, r\rangle]$ is at least $\frac{3}{4} + \epsilon$ for some $\epsilon > 0$. The probability is over the choice of uniform r in $\mathbb{Z}_2^n$ and $\langle x, r\rangle$ denotes the inner product of x and r over $\mathbb{Z}_2^n$. Show how to construct an algorithm $\mathcal{B}$ that outputs x by calling $\mathcal{A}_x$ about $O(n^2)$ times.
Hint: Your goal is to boost algorithm $\mathcal{A}_x$ to an algorithm such that $\Pr[\mathcal{A}_x(r) = \langle x, r\rangle]$ is close to 1, at which point finding x is easy by linear algebra. To evaluate $\langle x, r\rangle$ try choosing many random $s \in \mathbb{Z}_2^n$ and running $\mathcal{A}_x(s)$ and $\mathcal{A}_x(r \oplus s)$.

Problem 2: Let $\mathbb{G}$ be a cyclic group of known odd order q with generator $g \in \mathbb{G}$. Consider the function $f : \mathbb{Z}_q \to \mathbb{G}$ defined as $f(x) = g^x$. Let $\text{lsb} : \mathbb{Z}_q \to \{0, 1\}$ be the function that outputs the least significant bit of $x \in \mathbb{Z}_q$ when x is treated as a number in $\{0, \ldots, q-1\}$. Show that $\text{lsb}(x)$ is hard-core for $f(x)$, assuming discrete-log in $\mathbb{G}$ is hard.
Hint: first, suppose there is an algorithm $\mathcal{A}$ that takes g^x as input and *always* outputs $\text{lsb}(x)$. Show that $\mathcal{A}$ can be used to compute discrete-log in $\mathbb{G}$. To do so, observe that $(g^x)^{(q+1)/2}$ is the square root of g^x. Second, one would need to show that an algorithm $\mathcal{B}$ that given g^x outputs $\text{lsb}(x)$ with probability $\frac{1}{2} + \epsilon$ can be boosted to an algorithm that outputs $\text{lsb}(x)$ with probability close to 1 by calling $\mathcal{B}$ about $O(1/\epsilon^2)$ times. Here there is no need for you to prove this second part: you may assume it is true. The proof is not hard, but is a little tedious.

Problem 3: Commitments. Fix an RSA modulus $N = pq$, an RSA exponent e, and a random element $g \in \mathbb{Z}_N^*$. Prove that the following commitment scheme is secure: to commit to a message $m \in \{0, \ldots, e-1\}$ choose a random $r \in \mathbb{Z}_N^*$ and output $c \leftarrow g^m \cdot r^e \in Z_N^*$. To open the commitment send m and r to the receiver and the receiver accepts if $c = g^m \cdot r^e$. Prove that this commitment scheme is perfectly hiding. Prove that it is binding assuming that finding the e'th root of g is hard. Note that the factorization of N is not known to the sender.

Problem 4: Private information Retrieval. In class we saw how to use the ϕ-hiding assumption to construct a PIR protocol. Show that this PIR can be used to lookup k bits in the database (for small k, e.g. $k \leq 5$) with no additional communication beyond what is needed to lookup one bit. You may assume that the size of the modulus N is unchanged, even after your modification to the protocol.

Problem 5: Oblivious Transfer. Show that the Bellare-Micali OT protocol is insecure in a group where the Computational Diffie-Hellman problem is easy. That is, show that an algorithm $\mathcal{A}$ for solving the Computational Diffie-Hellman problem in $\mathbb{G}$ can be used to break one of recipient security or sender security.

Problem 6. Offline signatures. One approach to speeding up signature generation is to perform the bulk of the work offline, before the message to sign is known. Then, once the message m is given, generating the signature on m should be very fast. Our goal is to design a signature system with this property (in class we showed how to do something similar for oblivious transfer).

a. We show that any signature system can be converted into a signature where the bulk of the signing work can be done offline. Let $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be a secure signature system and let $\mathbb{G}$ be a group of order q where discrete log is hard. Consider the following modified signature system $(\mathsf{KeyGen}', \mathsf{Sign}', \mathsf{Verify}')$:

a. Algorithm KeyGen' runs algorithm KeyGen to obtain a signing key sk and verification vk. It also chooses a random group element $g \in \mathbb{G}$ and sets $h = g^\alpha$ for some random $\alpha \in \{1, \ldots, q\}$. It outputs the verification key $\mathsf{vk}' = (\mathsf{vk}, g, h)$ and the signing key $\mathsf{sk}' = (\mathsf{vk}', \mathsf{sk}, \alpha)$.

- Algorithm $\mathsf{Sign}'(\mathsf{sk}', m)$ first chooses a random $r \in \{1, \ldots, q\}$, computes $M = g^m h^r \in \mathbb{G}$, and then runs $\mathsf{Sign}(\mathsf{sk}, M)$ to obtain a signature σ. It outputs the signature $\sigma' = (\sigma, r)$.
- Algorithm $\mathsf{Verify}'(\mathsf{vk}', m, \sigma')$, where $\sigma' = (\sigma, r)$, computes $M = g^m h^r \in \mathbb{G}$ and outputs the result of $\mathsf{Verify}(\mathsf{vk}, M, \sigma)$.

Show that the bulk of the work in algorithm Sign' can be done before the message m is given. Hint: Recall that α is part of sk'.

b. Prove that this modified signature scheme is secure. In other words, show that an existential forgery under a chosen message attack on the modified scheme gives an existential forgery under a chosen message attack on the underlying scheme. You may use the fact that $H(m, r) = g^m h^r$ is a collision resistant hash function.