**CS255: Cryptography and Computer Security** **Winter 2018**

# Assignment #3

Due: Monday, Mar. 12, 2018, by Gradescope (each answer on a separate page).

**Problem 1.** Let's explore why in the RSA public key system each person has to be assigned a different modulus $n = pq$. Suppose we try to use the same modulus $n = pq$ for everyone. Each person is assigned a public exponent $e_i$ and a private exponent $d_i$ such that $e_i \cdot d_i = 1 \bmod \varphi(n)$. At first this appears to work fine: to encrypt to Bob, Alice computes $c = x^{e_{\text{bob}}}$ for some value $x$ and sends $c$ to Bob. An eavesdropper Eve, not knowing $d_{\text{bob}}$ appears to be unable to invert Bob's RSA function to decrypt $c$. Let's show that using $e_{\text{eve}}$ and $d_{\text{eve}}$ Eve can very easily decrypt $c$.

**a.** Show that given $e_{\text{eve}}$ and $d_{\text{eve}}$ Eve can obtain a multiple of $\varphi(n)$. Let us denote that integer by $V$.

**b.** Suppose Eve intercepts a ciphertext $c = x^{e_{\text{bob}}} \bmod n$. Show that Eve can use $V$ to efficiently obtain $x$ from $c$. In other words, Eve can invert Bob's RSA function.
**Hint:** First, suppose $e_{\text{bob}}$ is relatively prime to $V$. Then Eve can find an integer $d$ such that $d \cdot e_{\text{bob}} = 1 \bmod V$. Show that $d$ can be used to efficiently compute $x$ from $c$. Next, show how to make your algorithm work even if $e_{\text{bob}}$ is not relatively prime to $V$.

**Note:** In fact, one can show that Eve can completely factor the global modulus $n$.

**Problem 2.** Time-space tradeoff. Let $f : X \to X$ be a one-way one-to-one function. Show that one can build a table $T$ of size $2B$ elements of $X$ ($B \ll |X|$) that enables an attacker to invert $f$ in time $O(|X|/B)$. More precisely, construct an $O(|X|/B)$-time deterministic algorithm $\mathcal{A}$ that takes as input the table $T$ and a $y \in X$, and outputs an $x \in X$ satisfying $f(x) = y$. This result suggests that the more memory the attacker has, the easier it becomes to invert functions.
**Hint:** Pick a random point $z \in X$ and compute the sequence

$$z_0 := z, \;\; z_1 := f(z), \;\; z_2 := f(f(z)), \;\; z_3 := f(f(f(z))), \;\; \ldots$$

Since $f$ is a permutation, this sequence must come back to $z$ at some point (i.e. there exists some $j > 0$ such that $z_j = z$). We call the resulting sequence $(z_0, z_1, \ldots, z_j)$ an $f$-cycle. Let $t := \lceil |X|/B \rceil$. Try storing $(z_0, z_t, z_{2t}, z_{3t}, \ldots)$ in memory. Use this table (or perhaps, several such tables) to invert an input $y \in X$ in time $O(t)$.

**Problem 3.** A commitment scheme enables Alice to commit a value $x$ to Bob. The scheme is *hiding* if the commitment does not reveal to Bob any information about the committed value $x$. At a later time Alice may *open* the commitment and convince Bob that the committed value is $x$. The commitment is *binding* if Alice cannot convince Bob that the committed value is some $x' \neq x$. Here is an example commitment scheme:

**Public values:** A group $\mathbb{G}$ of prime order $q$ and two generators $g, h \in \mathbb{G}$.

**Commitment:** To commit to an integer $x \in \mathbb{Z}_q$ Alice does the following: (1) she chooses a random $r \in \mathbb{Z}_q$, (2) she computes $b = g^x \cdot h^r \in \mathbb{G}$, and (3) she sends $b$ to Bob as her commitment to $x$.

**Open:** To open the commitment Alice sends $(x, r)$ to Bob. Bob verifies that $b = g^x \cdot h^r$.

Show that this scheme is hiding and binding.

**a.** To prove the hiding property show that $b$ reveals no information about $x$. In other words, show that given $b$, the committed value can be any element $x'$ in $\mathbb{Z}_q$.
Hint: show that for any $x' \in \mathbb{Z}_q$ there exists a unique $r' \in \mathbb{Z}_q$ so that $b = g^{x'} h^{r'}$.

**b.** To prove the binding property show that if Alice can open the commitment as $(x', r')$, where $x \neq x'$, then Alice can compute the discrete log of $h$ base $g$. In other words, show that if Alice can find an $(x', r')$ such that $b = g^{x'} h^{r'}$ and $x \neq x'$ then she can find the discrete log of $h$ base $g$. Recall that Alice also knows the $(x, r)$ used to create $b$.

**c.** Show that the commitment is *additively homomorphic*: given a commitment to $x \in \mathbb{Z}_q$ and a commitment to $y \in \mathbb{Z}_q$, Bob can construct a commitment to $z = ax + by$, for any $a, b \in \mathbb{Z}_q$ of his choice.

**Problem 4.** Fast one-time signatures from discrete-log. Let's see another application for the commitment scheme from the previous problem. Let $\mathbb{G}$ be a group of prime order $q$ with generator $g$. Consider the following signature system for signing messages in $\mathbb{Z}_q$:

KeyGen: choose $x, y \xleftarrow{R} \mathbb{Z}_q$, set $h := g^x$ and $u := g^y$.
output $\text{sk} := (x, y)$ and $pk := (g, h, u) \in \mathbb{G}^3$.

Sign($\text{sk}$, $m \in \mathbb{Z}_q$): output $s \in \mathbb{Z}_q$ such that $u = g^m h^s$.

Verify($pk, m, s$): output 'yes' if $u = g^m h^s$ and 'no' otherwise.

**a.** Explain how the signing algorithm works. That is, show how to find $s$ using sk. Note that signing is super fast.

**b.** Show that the signature scheme is weakly one-time secure assuming the discrete-log problem in $\mathbb{G}$ is hard. The weak one-time security game is defined as follows:

> the adversary $\mathcal{A}$ first outputs a message $m \in \mathbb{Z}_q$ and in response is given the public key $pk$ and a valid signature $s$ on $m$ relative to $pk$. The adversary's goal is to output a signature forgery $(m^*, s^*)$ where $m \neq m^*$.

Show how to use $\mathcal{A}$ to compute discrete-log in $\mathbb{G}$. This will prove that the signature is secure in this weak sense as long as the adversary sees at most one signature.

[Recall that in the standard game defined in class the adversary is first given the public-key and only then outputs a message $m$. In the weak game above the adversary is forced to choose the message $m$ *before* seeing the public-key. The standard game from class gives the adversary more power and more accurately models the real world.]

**Hint:** Your goal is to construct an algorithm $\mathcal{B}$ that given a random $h \in \mathbb{G}$ outputs an $x \in \mathbb{Z}_q$ such that $h = g^x$. Your algorithm $\mathcal{B}$ runs adversary $\mathcal{A}$ and receives a message $m$ from $\mathcal{A}$. Show how $\mathcal{B}$ can generate a public key $pk = (g, h, u)$ so that it has a signature $s$ for $m$. Your algorithm $\mathcal{B}$ then sends $pk$ and $s$ to $\mathcal{A}$ and receives from $\mathcal{A}$ a signature forgery $(m^*, s^*)$. Show how to use the signatures on $m^*$ and $m$ to compute the discrete-log of $h$ base $g$.

**c.** Show that this signature scheme is not 2-time secure. Given the signature on two distinct messages $m_0, m_1 \in \mathbb{Z}_q$ show how to forge a signature for any other message $m \in \mathbb{Z}_q$.

**Problem 5.** Oblivious PRF. Let $\mathbb{G}$ be a cyclic group of prime order $q$ generated by $g \in \mathbb{G}$. Let $H : \mathcal{M} \to \mathbb{G}$ be a hash function. Let $F$ be the PRF defined over $(\mathbb{Z}_q, \mathcal{M}, \mathbb{G})$ as follows:

$$F(k, m) := H(m)^k \quad \text{for } k \in \mathbb{Z}_q, m \in \mathcal{M}.$$

It is not difficult to show that this $F$ is a secure PRF assuming the Decision Diffie-Hellman (DDH) assumption holds in the group $\mathbb{G}$ and, the hash function $H$ is modeled as a random oracle.

Show that this PRF $F$ can be evaluated *obliviously*. That is, show that if Bob has the key $k$ and Alice has an input $m$, there is a simple protocol that allows Alice to learn $F(k, m)$ without learning anything else about $k$. Moreover, Bob learns nothing about $m$. You may assume that $g$ and $g^k$ are publicly known values. An oblivious PRF like this is quite handy for many applications.

**a.** To start the protocol, Alice generates a random $r \xleftarrow{R} \mathbb{Z}_q$ and sends to Bob $u := H(m) \cdot g^r$. Show that this $u$ is uniformly distributed in $\mathbb{G}$ and is independent of $m$, so that Bob learns nothing about $m$.

**b.** Show how Bob can respond to enable Alice to learn $F(k, m)$ and nothing else.

**Problem 6.** A bad choice of primes for RSA. Let's see why when choosing an RSA modulus $n = pq$ it is important to choose the two primes $p$ and $q$ *independently* at random. Suppose $n$ is generated by choosing the prime $p$ at random, and then choosing the prime $q$ dependent on $p$. In particular, suppose that $p$ and $q$ are close, namely $|p - q| < n^{1/4}$. Let's show that the resulting $n$ can be easily factored.

**a.** Let $A = (p + q)/2$ be the arithmetic mean of $p$ and $q$. Recall that $\sqrt{n}$ is the geometric mean of $p$ and $q$. Show that when $|p - q| < n^{1/4}$ we have that

$$A - \sqrt{n} < 1.$$

Hint: one way to prove this is by multiplying both sides by $A + \sqrt{n}$ and then using the fact that $A \geq \sqrt{n}$ by the AGM inequality.

**b.** Because $p$ and $q$ are odd primes, we know that $A$ is an integer. Then by part (a) we can deduce that $A = \lceil \sqrt{n} \rceil$, and therefore it is easy to calculate $A$ from $n$. Show that using $A$ and $n$ it is easy to factor $n$.

**Problem 7.** Consider again the RSA-FDH signature scheme. The public key is a pair $(N, e)$ where $N$ is an RSA modulus, and a signature on a message $m \in \mathcal{M}$ is defined as $\sigma := H(m)^{1/e} \in \mathbb{Z}_N$, where $H : \mathcal{M} \to \mathbb{Z}_N$ is a hash function. Suppose the adversary could find three messages $m_1, m_2, m_3 \in \mathcal{M}$ such that $H(m_1) \cdot H(m_2) = H(m_3)$ in $\mathbb{Z}_N$. Show that the resulting RSA-FDH signature scheme is no longer existentially unforgeable under a chosen message attack.

More generally, your attack shows that for security of the signature scheme, it should be difficult to find a set of inputs to $H$ where the corresponding outputs have a known algebraic relation in $\mathbb{Z}_N$. One can show that this is indeed the case for a random function $H : \mathcal{M} \to \mathbb{Z}_N$, which is what we assumed when proving security of RSA-FDH.