CS 388H: Cryptography **Fall 2021**

Homework 4: Public-Key Cryptography

Due: November 11, 2021 at 11:59pm (Submit on Gradescope) **Instructor:** David Wu

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

https://www.cs.utexas.edu/~dwu4/courses/fa21/static/homework.tex

You must submit your problem set via Gradescope (accessible through Canvas).

Collaboration Policy. You may discuss your general *high-level* strategy with other students, but you may not share any written documents or code. You should not search online for solutions to these problems. If you do consult external sources, you must cite them in your submission. You must include the names of all of your collaborators with your submission. Refer to the official course policies for the full details.

Problem 1: Matrix Diffie-Hellman [17 points]. Recall that the DDH assumption in a group $\mathbb{G}$ of prime order p (where $\log p = \text{poly}(\lambda)$) and generator g says that the tuple (g, g^x, g^y, g^{xy}) is computationally indistinguishable from the tuple (g, g^x, g^y, g^z) where $x, y, z \xleftarrow{R} \mathbb{Z}_p$. In this problem, we will consider a matrix generalization of DDH that is useful in many cryptographic settings. For a matrix $\mathbf{A} \in \mathbb{Z}_p^{n \times n}$, let $g^{\mathbf{A}} \in \mathbb{G}^{n \times n}$ denote the matrix of group elements whose $(i,j)^{\text{th}}$ entry is $g^{A_{i,j}}$ (where $A_{i,j}$ is the $(i,j)^{\text{th}}$ entry of $\mathbf{A}$). Show that under the DDH assumption in $\mathbb{G}$, for all integers $n = \text{poly}(\lambda)$, no efficient adversary can distinguish between the following two distributions:

$$\left\{\mathbf{u} \xleftarrow{R} \mathbb{Z}_p^n, \mathbf{v} \xleftarrow{R} \mathbb{Z}_p^n : g^{\mathbf{u}\mathbf{v}^\mathsf{T}}\right\} \text{ and } \left\{\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{n \times n} : g^{\mathbf{A}}\right\}.$$

In particular, the standard DDH assumption (with a random generator) corresponds to the case $n = 2$.
Hint: Consider a sequence of $n+1$ hybrid experiments that iteratively modify the rows of $\mathbf{A}$.

Remark: Essentially, an equivalent way to view the DDH assumption is that it is saying that a random rank 1 matrix is computationally indistinguishable from a random matrix in the exponent.

Problem 2: Computing on Encrypted Data [20 points]. Let $N = pq$ be an RSA modulus and suppose that $\gcd(N, \varphi(N)) = 1$. Consider the following public-key encryption scheme with message space $\mathbb{Z}_N$. The public key $\mathsf{pk} = N$ is the RSA modulus $N = pq$ and the secret key sk is the factorization $\mathsf{sk} = (p, q)$. Let $g = 1 + N \in \mathbb{Z}_{N^2}^*$. To encrypt a message $m \in \mathbb{Z}_N$, sample $h \xleftarrow{R} \mathbb{Z}_{N^2}^*$ and compute $c \leftarrow g^m h^N \in \mathbb{Z}_{N^2}^*$.

(a) Show that the discrete logarithm assumption base g in $\mathbb{Z}_{N^2}$ is easy. Namely, give an efficient algorithm that takes as input (g, h) where $h = g^x$ for some $x \in \mathbb{Z}_N$, and outputs x. Remember to be explicit about the existence of any inverses you use in your solution. **Hint:** Use the binomial theorem: $(a+b)^k = \sum_{i=0}^{k} \binom{k}{i} a^i b^{k-i}$.

(b) Show how to *efficiently* implement the decryption algorithm $\text{Decrypt}(\mathsf{sk}, c)$. Namely, describe an efficient algorithm that given the secret key $\mathsf{sk} = (p, q)$ and a ciphertext $c = g^m h^N$, outputs the message $m \in \mathbb{Z}_N$. You may use the fact that $\varphi(N^2) = N\varphi(N)$.

(c) Show that this public-key encryption scheme is semantically secure assuming that no efficient adversary is able to distinguish the following two distributions:

$$(N, u) \quad \text{and} \quad (N, v),$$

where $N = pq$ is an RSA modulus, $u \xleftarrow{R} \mathbb{Z}_{N^2}^*$ and $v \xleftarrow{R} \{h \in \mathbb{Z}_{N^2}^* : h^N\}$. Namely, show that the above encryption scheme is semantically secure assuming that it is hard to distinguish random values in $\mathbb{Z}_{N^2}^*$ from random N^{th} powers in $\mathbb{Z}_{N^2}^*$.

(d) Show that given the public key pk and two ciphertexts $c_1 \leftarrow \mathsf{Encrypt}(\mathsf{pk}, m_1)$, $c_2 \leftarrow \mathsf{Encrypt}(\mathsf{pk}, m_2)$, there is an efficient algorithm that outputs a new ciphertext c where $\mathsf{Decrypt}(\mathsf{sk}, c) = m_1 + m_2 \in \mathbb{Z}_N$. Your algorithm should only depend on *public* parameters and *not* the value of the messages m_1, m_2.

Remark: This is an example of an encryption scheme that supports computation on *encrypted* values.

Problem 3: Collision-Resistant Hashing from RSA [18 points]. Let $N = pq$ be an RSA modulus and take $e \in \mathbb{N}$ to be a prime that is also relatively prime to $\varphi(N)$. Let $u \xleftarrow{R} \mathbb{Z}_N^*$, and define the hash function

$$H_{N,e,u} \colon \mathbb{Z}_N^* \times \{0, \ldots, e-1\} \to \mathbb{Z}_N^* \quad \text{where} \quad H_{N,e,u}(x, y) = x^e u^y \in \mathbb{Z}_N^*.$$

In this problem, we will show that under the RSA assumption, $H_{N,e,u}$ defined above is collision-resistant. Namely, suppose there is an efficient adversary $\mathcal{A}$ that takes as input (N, e, u) and outputs $(x_1, y_1) \neq (x_2, y_2)$ such that $H_{N,e,u}(x_1, y_1) = H_{N,e,u}(x_2, y_2)$. We will use $\mathcal{A}$ to construct an efficient adversary $\mathcal{B}$ that takes as input (N, e, u) where $u \xleftarrow{R} \mathbb{Z}_N^*$ and outputs x such that $x^e = u \in \mathbb{Z}_N^*$.

(a) Show that using algorithm $\mathcal{A}$ defined above, algorithm $\mathcal{B}$ can efficiently compute $a \in \mathbb{Z}_N$ and $b \in \mathbb{Z}$ such that $a^e = u^b \pmod N$ and $0 \neq |b| < e$. Remember to argue why any inverses you compute will exist.

(b) Use the above relation to show how $\mathcal{B}$ can *efficiently* compute $x \in \mathbb{Z}_N$ such that $x^e = u$. Note that $\mathcal{B}$ does *not* know the factorization of N, so $\mathcal{B}$ cannot compute $b^{-1} \pmod{\varphi(N)}$. **Hint:** What is $\gcd(b, e)$?

Problem 4: Coppersmith Attacks on RSA [20 points]. In this problem, we will explore what are known as "Coppersmith" attacks on RSA-style cryptosystems. As you will see, these attacks are very powerful and very general. We will use the following theorem:

> **Theorem (Coppersmith, Howgrave-Graham, May).** Let N be an integer of unknown factorization. Let p be a divisor of N such that $p \geq N^\beta$ for some constant $0 < \beta \leq 1$. Let $f \in \mathbb{Z}_N[x]$ be a monic polynomial of degree δ. Then, there is an efficient algorithm that outputs all integers x such that
>
> $$f(x) = 0 \bmod p \qquad \text{and} \qquad |x| \leq N^{\beta^2/\delta}.$$

In the statement of the theorem, when we write $f \in \mathbb{Z}_N[x]$, we mean that f is a polynomial in an indeterminate x with coefficients in $\mathbb{Z}_N$. A *monic* polynomial is one whose leading coefficient is 1.

When $N = pq$ is an RSA modulus (where p, q are identically-distributed primes), the interesting instantiations of the theorem have either $\beta = 1/2$ (i.e., we are looking for solutions modulo a prime factor of N) or $\beta = 1$ (i.e., we are looking for small solutions modulo N).

For this problem, let N be an RSA modulus with $\gcd(\varphi(N),3) = 1$ and let $F_{\text{RSA}}(m) := m^3 \pmod N$ be the RSA trapdoor permutation.

(a) Let $n = \lceil \log_2 N \rceil$. Show that you can factor an RSA modulus $N = pq$ if you are given:

- The low-order $\lceil n/3 \rceil$ bits of p; **or**
- The high-order $\lceil n/3 \rceil$ bits of p.

As usual, remember to be explicit about any inverses you use in your solution.

(b) In class, we saw that there is a simple message-recovery attack against textbook RSA when it is used to encrypt short messages (i.e., take the cube root over $\mathbb{R}$). Here, we show that there is a message-recovery attack even if we apply a simple padding scheme to the message m before applying textbook RSA. Suppose we want to encrypt a message $m \in \{0,1\}^{\lfloor \log_2 N \rfloor / 5}$. We do so using the following version of textbook RSA:

- Set $M \leftarrow 2^\ell + m$ for some integer ℓ so that $N/2 \le M < N$. This corresponds to padding the message M by prepending it with a binary string "$10000\cdots000$."
- Output the ciphertext $c \leftarrow F_{\text{RSA}}(M)$.

Show that there is an efficient message-recovery attack on this scheme. Namely, construct an efficient adversary that takes as input the public key N and a ciphertext c, and outputs m.

(c) To avoid the problem with the padding scheme above, your friend proposes to encrypt the short message $m \in \{0,1\}^{\lfloor \log_2 N \rfloor / 5}$ by setting $M \leftarrow (m \| m \| m \| m \| m) \in \{0,1\}^{\lfloor \log_2 N \rfloor}$ and outputting $c \leftarrow F_{\text{RSA}}(M)$. Show that there is still an efficient message-recovery attack on the scheme.

Remark: This problem hopefully convinces you that you should *never* encrypt messages using variants of textbook RSA. Not only are such schemes not semantically secure (since they are deterministic), they are often vulnerable to message-recovery attacks. In addition, leaking a sufficient-large fraction of the bits of the factors of N also leads to a concrete attack.

Problem 5: Time Spent [3 extra credit points]. How long did you spend on this problem set? This is for calibration purposes, and the response you provide does not affect your score.

Optional Feedback. Please answer the following *optional* questions to help us design future problem sets. You do not need to answer these questions. However, we do encourage you to provide us feedback on how to improve the course experience.

(a) What was your favorite problem on this problem set? Why?

(b) What was your least favorite problem on this problem set? Why?

(c) Do you have any other feedback for this problem set?

(d) Do you have any other feedback on the course so far?