CS 388H: Cryptography **Fall 2026**

Homework 4: Public-Key Cryptography

Due: November 6, 2026 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/fa26/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: Hash Functions from Discrete Log [14 points]. Let $\mathbb{G}$ be a group of prime order p with generator g. Sample $h_1, \ldots, h_n \xleftarrow{R} \mathbb{G}$ and define the hash function $H_{h_1,\ldots,h_n} : \mathbb{Z}_p^n \to \mathbb{G}$ as follows:

$$H_{h_1,\ldots,h_n}(x_1, \ldots, x_n) := h_1^{x_1} h_2^{x_2} \cdots h_n^{x_n} \in \mathbb{G}.$$

(a) Show that $H_{h_1,\ldots,h_n}$ is collision resistant under the discrete log assumption in $\mathbb{G}$. Specifically, in the (keyed) collision-resistant hashing security game, the adversary is first given $h_1, \ldots, h_n \xleftarrow{R} \mathbb{G}$ and it succeeds if it outputs $(x_1, \ldots, x_n) \neq (x'_1, \ldots, x'_n)$ such that $H_{h_1,\ldots,h_n}(x_1, \ldots, x_n) = H_{h_1,\ldots,h_n}(x'_1, \ldots, x'_n)$. In the discrete log security game, the adversary is given $h \xleftarrow{R} \mathbb{G}$, and it wins if it outputs $x \in \mathbb{Z}_p$ such that $h = g^x$. **Hint:** Consider a reduction algorithm that starts by guessing the index $i^* \in [n]$ (uniformly at random) where $x_{i^*} \neq x'_{i^*}$. Remember to compute the advantage of your reduction algorithm.

(b) Show that the function $H_{h_1,\ldots,h_n}$ has a *trapdoor* that can be used to sample pre-images. Specifically, show that if someone knew the discrete logs of $h_1, \ldots, h_n$ (i.e., $z_i \in \mathbb{Z}_p$ where $h_i = g^{z_i}$ for each $i \in [n]$), then given any target $t \in \mathbb{Z}_p$, they can find a pre-image $(x_1, \ldots, x_n) \in \mathbb{Z}_p^n$ where $H_{h_1,\ldots,h_n}(x_1, \ldots, x_n) = g^t$.

Problem 2: Encrypted Group Chat [20 points]. Suppose a group of n people (denoted $P_1, \ldots, P_n$) wants to set up a shared key for an encrypted group chat. At the end of the group key-exchange protocol, everyone within the group should know the key, but an eavesdropper on the network should not. We will use the following variant of Diffie-Hellman over a group $\mathbb{G}$ of prime order p and generator g:

- At the beginning of the protocol, P_1 chooses $s \xleftarrow{R} \mathbb{Z}_p$. We will view P_1 as the group administrator that all of the other parties know.
- Each of the other parties P_i ($2 \leq i \leq n$) samples $r_i \xleftarrow{R} \mathbb{Z}_p$ and sends $x_i \leftarrow g^{r_i}$ to the group administrator P_1. The administrator P_1 replies to P_i with x_i^s.
- The group key is then defined to be $k \leftarrow H(g^s)$, where $H : \mathbb{G} \to \{0,1\}^\lambda$ is a hash function.

Both the group description $(\mathbb{G}, p, g)$ and the hash function H are public and known to everyone (both the protocol participants and the eavesdropper).

(a) Show that both the group administrator P_1 and each of the parties P_i ($2 \le i \le n$) are able to efficiently compute the group key (except perhaps with negligible probability).

(b) We say that the group key-exchange protocol is secure against eavesdroppers if no efficient adversary who sees the transcript of messages sent by the honest parties $P_1, \ldots, P_n$ is able to distinguish the group key k from a uniformly random string over $\{0,1\}^\lambda$, except perhaps with negligible probability. If we model H as an "ideal hash function" (i.e., random oracle), it suffices to argue that the shared Diffie-Hellman secret g^s is *unguessable*: namely, for all efficient adversaries $\mathcal{A}$,

$$\Pr[\mathcal{A}(x_2, x_2^s, \ldots, x_n, x_n^s) = g^s] = \mathrm{negl}(\lambda), \tag{1}$$

where $x_i = g^{r_i}$ and $r_2, \ldots, r_n, s \xleftarrow{R} \mathbb{Z}_p$. This means that an eavesdropper who only observes the messages sent by the honest parties cannot guess g^s, and correspondingly, the shared key $H(g^s)$ is uniformly random and unknown to the adversary.

Show that under the CDH assumption in $\mathbb{G}$, the shared Diffie-Hellman secret g^s in the group key-exchange protocol above is unguessable (i.e., Eq. (1) holds for all efficient adversaries $\mathcal{A}$). As usual, you should consider the contrapositive: show that if there exists an efficient adversary $\mathcal{A}$ that can predict g^s from the above challenge tuple $(x_2, x_2^s, \ldots, x_n, x_n^s)$, then there exists an efficient algorithm $\mathcal{B}$ that breaks CDH in $\mathbb{G}$. Your algorithm should work for any polynomially bounded n (i.e., you should not fix a value of n in your reduction). **Hint:** Your algorithm $\mathcal{B}$ may need to invoke $\mathcal{A}$ *more than once*. Remember to compute the advantage of the adversary you construct.

Problem 3: Reducing the Size of Lamport Signatures [30 points]. Let $f\colon \{0,1\}^\lambda \to \{0,1\}^\lambda$ be an injective one-way function. For an integer $i \ge 0$, write $f^{(i)}$ to denote the i-fold evaluation of f. Namely, for all $x \in \{0,1\}^\lambda$,

- $f^{(0)}(x) = x$; and
- $f^{(i)}(x) = f(f^{(i-1)}(x))$.

Let $w \ge 2$ be the "alphabet size" and let $\{0, \ldots, w-1\}^\ell$ be the message space. Consider the following one-time signature scheme:

- KeyGen: Sample $x_1, \ldots, x_\ell \xleftarrow{R} \{0,1\}^\lambda$. Output $\mathsf{sk} = (x_1, \ldots, x_\ell)$ and $\mathsf{vk} = (y_1, \ldots, y_\ell)$ where $y_i = f^{(w-1)}(x_i)$.
- Sign$(\mathsf{sk}, (d_1, \ldots, d_\ell))$: On input the signing key $\mathsf{sk} = (x_1, \ldots, x_\ell)$ and a message $(d_1, \ldots, d_\ell)$, output the signature $\sigma = (\sigma_1, \ldots, \sigma_\ell)$ where $\sigma_i = f^{(d_i)}(x_i)$.
- Verify$(\mathsf{vk}, (d_1, \ldots, d_\ell), (\sigma_1, \ldots, \sigma_\ell))$: On input the verification key $\mathsf{vk} = (y_1, \ldots, y_\ell)$, a message $(d_1, \ldots, d_\ell)$, and a signature $(\sigma_1, \ldots, \sigma_\ell)$, output 1 if $f^{(w-1-d_i)}(\sigma_i) = y_i$ for all $i \in [\ell]$ and 0 otherwise.

(a) Suppose we want to sign 256-bit message digests. As a function of λ, what is the size of the verification key and the size of the signature for the basic Lamport signature scheme (instantiated with f)? Suppose we instantiate the above signature scheme with $w = 2^4$ and $\ell = 64$. What is the size of the verification key and the size of the signature?

(b) Unfortunately, the above scheme is insecure even as a one-time signature scheme. Give an explicit attack against the *one-time* security of the above scheme.

(c) One way to defend against this attack is by appending a *checksum* to the message before signing. For a message $(d_1, \ldots, d_\ell) \in \{0, \ldots, w-1\}^\ell$, define the checksum $C = \sum_{i \in [\ell]} (w - 1 - d_i)$ and let $(c_1, \ldots, c_{\ell'}) \in \{0, \ldots, w-1\}^{\ell'}$ be the base-w representation of C, where $\ell' = \lfloor \log_w(\ell(w-1)) \rfloor + 1$. Now, to support signing ℓ-block messages, we instantiate the above scheme with an $(\ell + \ell')$-block message space. To sign a message $(d_1, \ldots, d_\ell)$, we use the above scheme to sign the *encoded* message

$$\mathsf{Encode}(d_1, \ldots, d_\ell) = (d_1, \ldots, d_\ell, c_1, \ldots, c_{\ell'}) \in \{0, \ldots, w-1\}^{\ell + \ell'}.$$

Show that for every pair of distinct messages $(d_1, \ldots, d_\ell) \neq (d'_1, \ldots, d'_\ell)$, if we consider their encoded messages

$$(d_1, \ldots, d_\ell, c_1, \ldots, c_{\ell'}) = \mathsf{Encode}(d_1, \ldots, d_\ell)$$
$$(d'_1, \ldots, d'_\ell, c'_1, \ldots, c'_{\ell'}) = \mathsf{Encode}(d'_1, \ldots, d'_\ell),$$

then there exists some index $i \in [\ell]$ where $d'_i < d_i$ or an index $i \in [\ell']$ where $c'_i < c_i$. Briefly explain why your attack from Part (b) no longer works in this case.

(d) Show that if f is an injective one-way function, then the scheme from Part (c) is a secure one-time signature scheme. Specifically, let $L = \ell + \ell'$ and show that if $\mathcal{A}$ breaks one-time security of the scheme from Part (c) with advantage ε, then there is an efficient adversary $\mathcal{B}$ that inverts f with advantage $\varepsilon / (L(w-1))$.

Remark: This signature scheme is a core building block of the stateful hash-based signature scheme XMSS and the stateless scheme SLH-DSA, which was recently standardized by NIST as a post-quantum signature scheme.

Problem 4: Time Spent [1 point]. 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?