CS 388H: Cryptography **Fall 2026**

Homework 5: Signatures and Lattice-Based Cryptography

Due: December 1, 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: Schnorr's Protocol and Signatures [18 points]. Recall Schnorr's identification protocol over a group $\mathbb{G}$ of prime order p with generator g. The statement is $h = g^x \in \mathbb{G}$ and the prover knows $x \in \mathbb{Z}_p$. The prover samples $r \xleftarrow{R} \mathbb{Z}_p$ and sends $u \leftarrow g^r$. The verifier replies with a challenge $t \xleftarrow{R} \mathbb{Z}_p$, and the prover responds with $z \leftarrow r + xt \in \mathbb{Z}_p$. The verifier accepts if $g^z = u \cdot h^t$.

(a) Suppose that in Schnorr's protocol, the verifier instead samples its challenge $t \xleftarrow{R} S$ from a set $S \subseteq \mathbb{Z}_p$ where $|S| = \ell$. Show that there is an efficient prover that is given *only* the statement (g, h) (and not the discrete log x) and which convinces the verifier to accept with probability at least $1/\ell$.

Remark: This shows that to achieve negligible soundness error in Schnorr's protocol, the challenge must be sampled from a set of super-polynomial size. This is also why the challenge in Schnorr signatures can be shortened to 128 bits (as discussed in lecture), but not much further.

(b) Let $\mathbb{G}$ be a cyclic group of order $n = kq$, where $k > 1$ is a small constant (e.g., $k = 8$) and $q \gg k$ is prime. Let g be a generator of $\mathbb{G}$, and let $\hat{g} = g^k \in \mathbb{G}$. In this case, $\hat{g}$ generates a subgroup of $\mathbb{G}$ of prime order q. Consider Schnorr's protocol defined in the prime-order group generated by $\hat{g}$ (this coincides with the typical setting in cryptography where we work over a group of prime order). Namely, on input a statement $\hat{h}$, the prover's first message is $u = \hat{g}^r$ where $r \xleftarrow{R} \mathbb{Z}_q$, the verifier's challenge is $t \xleftarrow{R} \mathbb{Z}_q$, the prover's response is $z \in \mathbb{Z}_q$, and the verifier accepts if $\hat{g}^z = u \cdot \hat{h}^t$. Here, we interpret the challenge t as an integer in $\{0, \ldots, q-1\}$ when computing $\hat{h}^t$. Suppose $\hat{h} = g^x \in \mathbb{G}$ where $x \in \mathbb{Z}_n$ is *not* a multiple of k, so $\hat{h} \notin \langle \hat{g} \rangle$. Show that a (malicious) prover who knows x can convince the verifier to accept with probability at least $1/k - \mathsf{negl}$ (you can assume that the ratio k/q is negligible). Namely, even though $\hat{h}$ is not in the group generated by $\hat{g}$, the prover can nonetheless convince the verifier that it knows the discrete log of $\hat{h}$ base $\hat{g}$ (even though this discrete log does *not* exist).

Remark: This shows that Schnorr's protocol is no longer sound if the base $\hat{g}$ is not a generator of the full group $\mathbb{G}$ and the order of $\mathbb{G}$ has small factors. To prevent this attack and recover soundness, the verifier must *additionally* check that $\hat{h}$ is indeed in the group generated by $\hat{g}$. This setting is very common in practice (and should not be glossed over when designing cryptographic systems). For instance, if we consider groups over the integers, we typically work over a prime-order subgroup of

$\mathbb{Z}_p^*$ for a prime p (e.g., the subgroup of prime order q when $p = kq + 1$ for some integer k) or over an elliptic-curve group with order kq and small k (in this case, k is referred to as the "cofactor" of the curve).

(c) Recall the ECDSA signature scheme from lecture over a group $\mathbb{G}$ of prime order p with generator g. The verification key is $h = g^x$ and the signing key is $x \in \mathbb{Z}_p$. To sign a message m, the signer samples $\alpha \xleftarrow{R} \mathbb{Z}_p^*$, computes $r \leftarrow f(g^\alpha) \in \mathbb{Z}_p$ and $s \leftarrow (H(m) + rx)/\alpha \in \mathbb{Z}_p$, and outputs $\sigma = (r, s)$. If $r = 0$ or $s = 0$, the signer resamples α and tries again. Here, $f : \mathbb{G} \to \mathbb{Z}_p$ is a fixed public function and $H : \{0,1\}^* \to \mathbb{Z}_p$ is a hash function. We saw in lecture that in the Schnorr signature scheme, signing two different messages using the same randomness leaks the signing key. Show that this is also the case for ECDSA.

Remark: This attack was used to break the security of the Sony PlayStation 3. Sony used ECDSA with a *fixed* value of α to sign software for the PS3. This allowed attackers to extract Sony's ECDSA signing key and in turn, sign arbitrary firmware for the PS3. Thus, when using ECDSA, it is also critical to sample fresh randomness for every signature (or to derandomize the signing algorithm using a PRF, as we discussed for Schnorr signatures).

Problem 2: SIS and Inhomogeneous SIS [18 points]. Let n, m, q, β be lattice parameters where q is a prime with $q > \beta \geq 1$, and $m, \beta = \text{poly}(n)$. Recall the following two problems from lecture:

- $\mathsf{SIS}_{n,m,q,\beta}$: Given $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{Ax} = \mathbf{0} \bmod q$ and $\|\mathbf{x}\| \leq \beta$.
- $\mathsf{ISIS}_{n,m,q,\beta}$: Given $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$ and $\mathbf{y} \xleftarrow{R} \mathbb{Z}_q^n$, find a vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{Ax} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \leq \beta$.

Both problems are defined with respect to the ℓ_∞ norm: $\|\mathbf{x}\| = \max_{i \in [m]} |x_i|$. Show that the $\mathsf{SIS}_{n,m+1,q,\beta}$ problem is hard **if and only if** the $\mathsf{ISIS}_{n,m,q,\beta}$ problem is hard. In particular, give two explicit reductions and analyze their advantages. You can treat the lattice dimension n as the security parameter.

Problem 3: Regev Encryption [18 points]. In lecture, we described Regev encryption in the setting where the message is encoded in the *most significant bits* of the ciphertext. Here, we will consider a variant where the message is encoded in the *least significant bits* of the ciphertext. For simplicity, we will just consider the symmetric setting (but everything generalizes to the public-key setting in the manner described in lecture). Let the message space be $\mathbb{Z}_p$, n be the lattice dimension, q be the modulus, and χ be the error distribution. Suppose that $p < q$ and $\gcd(p, q) = 1$. Note that p and q need *not* be prime here.

- The secret key is a vector $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$.
- To encrypt a message $\mu \in \mathbb{Z}_p$, sample $\mathbf{a} \xleftarrow{R} \mathbb{Z}_q^n$ and $e \leftarrow \chi$, and output the ciphertext $\mathsf{ct} = (\mathbf{a}, \mathbf{s}^\intercal \mathbf{a} + pe + \mu)$. Here, we view μ as an integer in $\{0, \ldots, p-1\}$ and the arithmetic is over $\mathbb{Z}_q$.

(a) Given a ciphertext ct and the secret key $\mathbf{s}$, describe the decryption algorithm. Prove correctness of the encryption scheme with your choice of decryption algorithm. You may assume that $\Pr[e \leftarrow \chi : |e| < q/(2p) - 1] = 1$.

(b) Show that under the $\mathsf{LWE}_{n,m,q,\chi}$ assumption, the above encryption scheme has pseudorandom ciphertexts (and thus, is CPA-secure) against an adversary that makes at most m encryption queries. Namely, in the pseudorandom ciphertexts game that we described in lecture, the adversary can make

up to m encryption queries on (possibly-repeating) messages of its choosing; on each query $\mu_i \in \mathbb{Z}_p$, the adversary either receives $\mathsf{Encrypt}(\mathbf{s}, \mu_i)$ or it receives a random string $\mathsf{ct} = (\mathbf{a}, b)$ where $\mathbf{a} \xleftarrow{R} \mathbb{Z}_q^n$ and $b \xleftarrow{R} \mathbb{Z}_q$.

This version of Regev encryption can also be extended to obtain a fully homomorphic encryption 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 overall?