LECTURE NOTES

Factoring
$N = pq$
DLP
$g^x = h$
SVP
$\lambda_1(\Lambda) = \min_{v\in\Lambda\setminus\{0\}} \|v\|$
CVP
$\mathrm{dist}(t,\Lambda) = \min_{v\in\Lambda} \|t-v\|$
SIS
$A\mathbf{x} \equiv 0 \pmod{q},\ \|\mathbf{x}\|$ small
LWE
$As + e \equiv b \pmod{q}$
Codes
$He^T = s$
Isogeny
$\varphi : E_1 \to E_2$

Hard Problems in Cryptography

Mathematical Foundations for Post-Quantum Cryptography

Factorization · Discrete Logarithm · Lattices · Codes · Isogenies · Multivariate · Hash

Ji, Yonghyeon

Classical hardness assumptions and post-quantum reductions

Hard Problems in Cryptography

Mathematical Foundations for Classical and Post-Quantum Cryptography

Ji, Yonghyeon

February 20, 2026

Contents

Chapter 1

Preliminaries and Notation

1.1 Probability, Advantages, and Negligibility

A function $\mathrm{negl}(\lambda)$ is *negligible* if for every polynomial $p(\cdot)$, $\mathrm{negl}(\lambda) < 1/p(\lambda)$ for all sufficiently large λ.

For a distinguisher $\mathcal{A}$ attempting to distinguish distributions $\mathcal{D}_0, \mathcal{D}_1$,

$$\mathrm{Adv}_{\mathcal{A}}(\mathcal{D}_0, \mathcal{D}_1) \overset{\text{def}}{=} \left| \Pr[\mathcal{A}(x) = 1 \mid x \overset{\$}{\leftarrow} \mathcal{D}_0] - \Pr[\mathcal{A}(x) = 1 \mid x \overset{\$}{\leftarrow} \mathcal{D}_1] \right|.$$

1.2 Linear Algebra over Rings/Fields

For modulus $q \in \mathbb{N}$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$. We write vectors as column vectors by default; $A \in \mathbb{Z}_q^{m\times n}$ and $s \in \mathbb{Z}_q^n$ implies $As \in \mathbb{Z}_q^m$.

For codes, $\mathbb{F}_2$ is the binary field. A parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$ defines a linear code $C = \{c \in \mathbb{F}_2^n : Hc^\top = 0\}$.

1.3 Norms and "Shortness"

Common norms in lattices:

$$\|x\|_2 = \sqrt{\sum_i x_i^2}, \qquad \|x\|_\infty = \max_i |x_i|.$$

In ring-/module-structured settings, "shortness" often means coefficient vector is small under ℓ_2 or ℓ_∞.

1.4 Distributions for Errors

In LWE/NTRU, errors are usually sampled from:

- *Discrete Gaussian* D_σ over $\mathbb{Z}$ (or $\mathbb{Z}^m$) with parameter σ,
- *Centered binomial* (difference of two binomials), or

- bounded distributions like uniform on $\{-\eta, \ldots, \eta\}$.

Cryptographic security typically needs errors "small" compared to q but large enough to hide secrets statistically.

1.5 Search vs. decision vs. distinguishing.

Many hardness notions can be expressed as:

- *Search*: output a witness (e.g. a factor, a discrete log, an error vector).
- *Decision*: decide existence of a witness.
- *Distinguishing*: tell apart two distributions (e.g. LWE vs. uniform).

Chapter 2

Lattice-Based Hard Problems

2.1 Background: Lattices, Duality, and Problems

A (full-rank) lattice $\Lambda \subset \mathbb{R}^n$ is $\Lambda = \{Bz : z \in \mathbb{Z}^n\}$ for some basis matrix $B \in \mathbb{R}^{n \times n}$.

Classic algorithmic problems:

- **SVP (Shortest Vector Problem):** Find $0 \neq v \in \Lambda$ minimizing $\|v\|_2$.
- **GapSVP (Decision/SVP approximation):** Given (Λ, d) decide whether $\lambda_1(\Lambda) \leq d$ or $\lambda_1(\Lambda) > \gamma d$.
- **SIVP (Shortest Independent Vectors):** Find n linearly independent vectors of length $\leq \gamma \cdot \lambda_n(\Lambda)$.

The importance for cryptography: average-case problems (LWE/SIS) reduce from worst-case lattice problems (GapSVP/SIVP) under suitable parameters.

2.2 Learning With Errors (LWE)

2.2.1 Formal Statements

Definition 2.2.1 (Search-LWE)**.** Fix integers $n, m, q \in \mathbb{N}$ and an error distribution χ over $\mathbb{Z}$ (typically supported on small integers). Sample $A \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$ uniformly, secret $s \xleftarrow{\$} \mathbb{Z}_q^n$ (usually uniform), and error $e \xleftarrow{\$} \chi^m$. Given (A, b) where

$$b = As + e \bmod q \in \mathbb{Z}_q^m,$$

output the secret s (or equivalently recover e).

Definition 2.2.2 (Decision-LWE)**.** Under the same parameterization, consider two distributions over (A, b):

$$\mathcal{D}_0 : (A, As + e \bmod q), \qquad \mathcal{D}_1 : (A, u), \;\; u \xleftarrow{\$} \mathbb{Z}_q^m.$$

Given (A, b), output a bit indicating whether $(A, b) \xleftarrow{\$} \mathcal{D}_0$ or $(A, b) \xleftarrow{\$} \mathcal{D}_1$ with non-negligible advantage.

2.2.2 Geometric / Statistical Intuition

Each equation is:

$$\langle a_i, s\rangle + e_i \equiv b_i \pmod{q}.$$

If errors were 0, this is solving a linear system over $\mathbb{Z}_q$. Errors make it an instance of *noisy linear equations*, and (crucially) hide s.

A typical heuristic: if e is small in $\mathbb{Z}$ and q is large, the mapping $s \mapsto As + e$ looks like "random" without knowing s, but still allows decryption by rounding in cryptosystems.

2.2.3 Decision vs Search; Standard Relationships

Cryptographic constructions often assume decision-LWE hardness (for pseudorandomness) and search-LWE hardness (for extracting secrets). Under many standard parameter regimes, one can relate them:

Remark 2.2.3 (Search-to-decision (informal)). For prime q, there are classical reductions showing decision-LWE is no harder than search-LWE and vice versa (up to losses), under mild conditions. Intuitively, if you can recover s then you can distinguish; conversely, if you can distinguish, you can often recover s coordinate-by-coordinate using hybrid and rerandomization tricks.

2.2.4 Worst-Case to Average-Case Reductions (High-Level)

A landmark result (Regev-style) shows that (for appropriate α where errors have size about αq) decision-LWE is at least as hard as approximating worst-case lattice problems (GapSVP/SIVP) in dimension n within poly factors. The technical conditions tie α, q, and n.

Remark 2.2.4 (What you should remember). The security story is: *if LWE is easy on average, then certain canonical lattice problems are easy in the worst case*. This is why LWE is a central conservative assumption.

2.2.5 Parameter Regimes (Conceptual)

Let error magnitude scale be σ (e.g., standard deviation for discrete Gaussian). Often one defines $\alpha = \sigma/q$.

- **Correctness in encryption**: needs $\sigma \ll q$ so small errors can be rounded.
- **Security**: needs σ large enough that $As + e$ hides s; also m large enough to prevent solving.
- **Typical cryptosystems**: use $m \approx n \log q$ or m a constant multiple of n in module/ring variants.

2.2.6 Attack Taxonomy (What to Teach)

Best-known attacks broadly fall into:

- **Lattice reduction (primal):** view LWE as finding a close vector / BDD instance; build a lattice from A and b and run BKZ-type reduction; recover s by nearest-plane or enumeration.
- **Dual attack:** find a short vector y in the dual lattice such that $y^\top A \equiv 0 \bmod q$, then test $y^\top b$ for smallness vs uniform.

- **BKW / combinatorial:** reduce dimension via collision-finding on A rows; grows fast with q and noise but can matter for small moduli.
- **Arora–Ge (algebraic):** for very small error alphabets and special parameter settings, solve polynomial system.

2.2.7 Exercises (LWE)

Exercise 2.2.1 (Upper-undergrad: noiseless baseline)**.** Assume $e = 0$ and $m \geq n$. Show how to recover s efficiently from $(A, As \bmod q)$ when q is prime and A has full rank.

Exercise 2.2.2 (Masters: distinguishing via dual vector)**.** Let $y \in \mathbb{Z}^m$ satisfy $y^\top A \equiv 0 \pmod{q}$. Show that if $(A, b) \overset{\$}{\leftarrow} \mathcal{D}_0$ then

$$y^\top b \equiv y^\top e \pmod{q},$$

and argue heuristically why $y^\top b$ is statistically closer to small integers mod q than uniform if y is short.

Exercise 2.2.3 (PhD: hybrid for search-to-decision sketch)**.** Assume q prime. Outline a reduction strategy that recovers s_i (the i-th coordinate of s) using a decision oracle by embedding a guess into the distribution and using hybrids.

2.3 Short Integer Solution (SIS)

2.3.1 Formal Statement

Definition 2.3.1 (Search-SIS)**.** Let $q \in \mathbb{N}$, $n, m \in \mathbb{N}$, and bound $\beta \in \mathbb{N}$. Sample $A \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times m}$ uniformly. Find a nonzero vector $x \in \mathbb{Z}^m \setminus \{0\}$ such that

$$Ax \equiv 0 \pmod{q} \quad \text{and} \quad \|x\| \leq \beta,$$

where $\|\cdot\|$ is typically ℓ_2 or ℓ_∞.

2.3.2 Interpretation as Finding Short Relations

The condition $Ax \equiv 0 \pmod{q}$ means x is an integer relation among the columns of A modulo q. Without the shortness constraint, there are many solutions. The hardness is to find a *short* one.

2.3.3 Connection to Hash-and-Sign / Commitments

SIS underlies:

- lattice-based hash functions: mapping $x \mapsto Ax \bmod q$; collisions correspond to short x with $Ax \equiv 0$.
- commitments: binding reduces to SIS.
- signatures (e.g., GPV-style): produce short preimages under a public linear map.

2.3.4 Worst-Case Reductions (High-Level)

SIS is related to worst-case lattice problems as well: if SIS is easy for certain (n, m, q, β), then approximating certain lattice problems is easy in the worst case. The parameter tradeoff differs from LWE.

2.3.5 Attack Taxonomy

- **Lattice reduction:** interpret SIS as finding a short vector in a lattice of solutions; build lattice basis and run BKZ.
- **Combinatorial / meet-in-the-middle:** sometimes applicable for ℓ_∞ and special constraints (rare in standard parameters).

2.3.6 Exercises (SIS)

Exercise 2.3.1 (Upper-undergrad: pigeonhole existence)**.** Let $A \in \mathbb{Z}_q^{n\times m}$ and consider all $x \in \{0,1\}^m$. Show that if $2^m > q^n$, then there exist distinct $x \neq x'$ with $Ax \equiv Ax' \pmod{q}$. Deduce existence of a nonzero $\{-1,0,1\}^m$ solution to $A(x - x') \equiv 0$.

Exercise 2.3.2 (Masters: collision-resistance from SIS)**.** Define $H(x) = Ax \bmod q$ for short x in some domain. Formalize how a collision ($x \neq x'$) yields an SIS solution.

Exercise 2.3.3 (PhD: parameter reasoning)**.** For fixed n, q, explain qualitatively why increasing m makes SIS *easier* (more relations exist), but also allows setting smaller β while maintaining existence of solutions.

2.4 NTRU Search Problem

2.4.1 Ring Setting

Let $f(x)$ be a cyclotomic-like polynomial (e.g., $x^N + 1$ with N power of 2, or $x^N - 1$ for classical NTRU variants). Define

$$R = \mathbb{Z}[x]/(f(x)), \qquad R_q = R/qR \cong \mathbb{Z}_q[x]/(f(x)).$$

Elements are represented by degree-$< N$ polynomials; "small" typically refers to small coefficients.

2.4.2 Formal Problem (One Common Form)

Definition 2.4.1 (NTRU Search (informal canonical form))**.** Sample $f, g \in R$ from a "small" distribution such that f is invertible in R_q. Publish

$$h \equiv gf^{-1} \pmod{q} \in R_q.$$

Given h, recover a short pair (f, g) (or an equivalent short representation) satisfying $hf \equiv g \pmod{q}$ under the same smallness constraints.

2.4.3 Key Ambiguities in NTRU Statements

NTRU has many instantiations; the exact hardness depends on:

- ring choice ($x^N \pm 1$; N prime; etc.),
- modulus structure (prime q vs power-of-two),
- distribution of (f, g) (ternary, Gaussian, centered binomial),
- norm and acceptance region (e.g., ℓ_∞ bounds on coefficients),
- whether we are in *ring* vs *module* setting.

2.4.4 NTRU as a Lattice Problem

Given h, consider the *NTRU lattice*:

$$\Lambda_h = \left\{(u, v) \in R^2 : u - hv \equiv 0 \pmod{q}\right\}.$$

A secret key corresponds to a short vector $(g, f) \in \Lambda_h$. Thus, breaking NTRU is (at high level) a shortest-vector style task in a structured lattice.

2.4.5 Attack Taxonomy

- **Lattice reduction on NTRU lattice:** embed Λ_h into an integer lattice of dimension $2N$ and use BKZ; recover short (f, g).
- **Hybrid attacks:** partial guessing of coefficients + lattice reduction for the remaining.
- **Subfield / algebraic structure attacks:** exploit ring structure if parameters are ill-chosen (historically important lesson: structure can leak).

2.4.6 Exercises (NTRU)

Exercise 2.4.1 (Upper-undergrad: derive the NTRU relation)**.** Show that $h \equiv gf^{-1} \pmod{q}$ implies $hf \equiv g \pmod{q}$. Explain why small (f, g) is a "short relation" between 1 and h.

Exercise 2.4.2 (Masters: NTRU lattice membership)**.** Define Λ_h as above. Prove that $(g, f) \in \Lambda_h$. What other pairs are in Λ_h? Characterize them modulo q.

Exercise 2.4.3 (PhD: compare NTRU vs LWE intuition)**.** Give a conceptual comparison: NTRU keys correspond to short vectors in a structured lattice tied to one public ring element; LWE hides a secret with additive noise across many samples. Discuss how this affects the style of security reductions and the known attacks.

Chapter 3

Code-Based Hard Problems

3.1 Linear Codes and Syndromes

Let $C \subseteq \mathbb{F}_2^n$ be a linear $[n, k]$ code with parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$. For $e \in \mathbb{F}_2^n$, the *syndrome* is $s = He^\top \in \mathbb{F}_2^{n-k}$. Syndrome decoding asks: given H and s, find a low-weight e with syndrome s.

3.2 Syndrome Decoding (SD)

3.2.1 Formal Statements

Definition 3.2.1 (Search-SD)**.** Given $H \in \mathbb{F}_2^{(n-k)\times n}$, a syndrome $s \in \mathbb{F}_2^{n-k}$, and an integer weight w, find $e \in \mathbb{F}_2^n$ such that

$$He^\top = s \quad \text{and} \quad \mathrm{wt}(e) = w.$$

Definition 3.2.2 (Decisional SD (DSD))**.** Distinguish:

- $\mathcal{D}_0$: $e \xleftarrow{\$} \mathbb{F}_2^n$ uniform subject to $\mathrm{wt}(e) = w$, and $s = He^\top$.
- $\mathcal{D}_1$: $s \xleftarrow{\$} \mathbb{F}_2^{n-k}$ uniform.

Given (H, s) output whether s is a syndrome of a weight-w error vector with non-negligible advantage.

3.2.2 Why SD is Hard

For random H, the mapping $e \mapsto He^\top$ is linear and many-to-one. The hardness comes from the combinatorial explosion of possible e of weight w:

$$\#\{e \in \mathbb{F}_2^n : \mathrm{wt}(e) = w\} = \binom{n}{w}.$$

Brute force is exponential in n for typical w scaling.

3.2.3 Information Set Decoding (ISD) Family (High-Level)

The dominant attacks are *information set decoding* and its refinements (Prange, Stern, Dumer, BJMM, and modern variants). The meta-idea:

- guess an "information set" of coordinates where the error is assumed sparse/structured,
- reduce the decoding task to a smaller combinatorial search,
- repeat until success with certain probability.

Complexities are typically 2^{cn} with constant c depending on rate k/n and relative weight w/n.

3.2.4 Exercises (SD)

Exercise 3.2.1 (Upper-undergrad: syndrome as coset)**.** Fix H. Show that the set $\{e \in \mathbb{F}_2^n : He^\top = s\}$ is an affine subspace (a coset of $\ker(H)$). What is its size?

Exercise 3.2.2 (Masters: counting solutions)**.** Assume H is full rank. For random s, what is the expected number of solutions e of weight exactly w? Express it using $\binom{n}{w}$ and 2^{n-k} and justify the approximation.

Exercise 3.2.3 (PhD: ISD success probability sketch)**.** In Prange's algorithm, one chooses a set I of k positions and hopes the error is zero on I. Derive the success probability in terms of n, k, w and the expected work factor.

3.3 QC Syndrome Decoding (QCSD)

3.3.1 Quasi-Cyclic Structure

A binary quasi-cyclic (QC) code often uses a parity-check matrix built from circulant blocks. For block size p, a circulant matrix is determined by its first row; multiplication corresponds to polynomial multiplication modulo $x^p - 1$.

3.3.2 Problem Statement

Definition 3.3.1 (QCSD / DQCSD)**.** Same as SD/DSD, except H is drawn from a QC ensemble (block-circulant structure), and sometimes e is restricted to QC form. Given (H, s, w) find e with $He^\top = s$ and $\mathrm{wt}(e) = w$, or distinguish structured syndromes from uniform.

3.3.3 Security Subtleties

QC structure reduces public key sizes dramatically but introduces algebraic symmetry. Best practice is to choose parameters so that known structural attacks (e.g., exploiting cyclic shifts, folding, or module-based speedups) do not reduce security below target.

3.3.4 Exercises (QCSD)

Exercise 3.3.1 (Masters: circulant-as-polynomial)**.** Show how multiplying a circulant matrix by a vector corresponds to polynomial multiplication modulo $x^p - 1$.

Exercise 3.3.2 (PhD: symmetry and attack surface)**.** Explain how cyclic symmetry can introduce additional low-weight codewords or enable collision-style shortcuts. Give at least one concrete avenue (high-level) by which QC structure can be exploited.