Teaching Packet: Hard Problems in Cryptography (7 Weeks)

Contents

1 How to Use This Packet

- Each week contains **3 lectures** (A/B/C). For each lecture:
  - **Slides-outline**: bullet-point list suitable for beamer slides.
  - **Instructor notes**: a script-like narrative + emphasis points.
  - **Recitation worksheet**: student-facing problems for a 50–90 minute session.
- Each week ends with **homework** plus **solution sketches** (not fully worked, but enough to grade).
- Notation is consistent across topics; assumptions are made explicit.

2 Global Preliminaries (Week 0 / Lecture 0)

Slides-outline

Slide: Course framing

- "Hard problem" = conjectured infeasible for PPT adversary at chosen security parameter λ.
- Distinguish *mathematical* hardness vs *implementation* failures.
- Families: factoring, discrete log, lattices, codes, isogenies, MQ, hash.

Slide: Complexity language

- Negligible $\mathrm{negl}(\lambda)$; polynomial $\mathrm{poly}(\lambda)$; security parameter λ.
- Subexponential L-notation: $L_N[\alpha, c] = \exp((c + o(1))(\log N)^{\alpha}(\log\log N)^{1-\alpha})$.
- Search vs decision vs distinguishing formulations.

Instructor notes

Instructor notes. Set expectations: we care about *best known attacks*, not absolute impossibility. Emphasize that a scheme can be broken even if the underlying "family" remains plausible (e.g. SIDH/SIKE). Explain why mathematicians like formal problem definitions and reductions, while cryptanalysts speak in attack taxonomies.

Worksheet

Worksheet.

1. Give one example each of search, decision, distinguishing.
2. Show that if p, q are primes and you know $N = pq$ and $p + q$, then you can recover p, q.
3. (Short) Estimate collision probability after q hashes into n bits using the birthday heuristic.

3 Week 1: Integer Factorization (RSA/Rabin)

3.1 Lecture 1A: Definitions and Reductions

Slides-outline

Slide: Problem statement

- Factoring (search): given composite N, output nontrivial divisor.
- RSA distribution: $N = pq$ with p, q random λ-bit primes.

Slide: Reductions used in crypto

- Knowing $\varphi(N)$ factors semiprimes.
- Order-finding $\Rightarrow$ factoring (random a).
- Rabin inversion $\Rightarrow$ factoring.

Instructor notes

Instructor notes. Do not overclaim "RSA $\Leftrightarrow$ factoring"; explain the nuance: RSA inversion is *believed* equivalent to factoring but not proved in general. However, Rabin inversion is provably as hard as factoring for Blum integers. Use the $p + q$ trick to show $\varphi(N)$ is enough.

Worksheet

Worksheet.

1. Prove: if $N = pq$ and $\varphi(N)$ is known, then p, q can be recovered.
2. Show: if you can compute $\lambda(N)$ (Carmichael), you can factor $N = pq$.
3. For $N = 77$, compute $\varphi(N)$ and list $(\mathbb{Z}/N\mathbb{Z})^\times$ orders for $a \in \{2, 3, 5, 6\}$.

3.2 Lecture 1B: Classical Attacks (ECM, QS, GNFS)

Slides-outline

Slide: Landscape

- "Small factor" methods: trial division, Pollard ρ, Pollard $p - 1$, ECM.
- "Sieve" methods: QS ($L_N[1/2, 1]$), GNFS ($L_N[1/3, (64/9)^{1/3}]$).

Slide: ECM intuition

- Replace $a^M \bmod p$ smoothness with elliptic curve group order smoothness.
- Expected time depends on size of smallest prime factor.

Slide: QS/GNFS at 30,000 feet

- Collect relations $\Rightarrow$ sparse linear algebra over $\mathbb{F}_2$.
- Square root step produces congruence of squares.

Instructor notes

Instructor notes. Keep QS/GNFS black-box but conceptually correct: relations, smoothness probability, linear algebra in exponent vectors mod 2. For mathematicians: relate to ideal factorization language (NFS) without drowning in details.

Worksheet

Worksheet.

1. Explain why QS needs linear algebra over $\mathbb{F}_2$.
2. Run a toy QS by hand for $N = 77$: try $x^2 - N$ for several x and look for squares/smooth values.
3. Compare Pollard ρ expected time for a 20-bit factor vs a 40-bit factor (order-of-magnitude).

3.3 Lecture 1C: Quantum Factoring (Shor) at Concept Level

Slides-outline

Slide: Reduction

- Factoring $\to$ order-finding in $(\mathbb{Z}/N\mathbb{Z})^\times$.
- Order-finding via period finding for $f(x) = a^x \bmod N$.

Slide: QFT intuition

- Fourier sampling reveals period r with high probability.
- Classical post-processing: if r even, use $\gcd(a^{r/2} \pm 1, N)$.

Instructor notes

Instructor notes. Avoid full quantum circuit details; emphasize the mathematical structure: hidden periodicity and Fourier analysis on cyclic groups. Mention that asymptotically it is polynomial in $\log N$ but requires fault-tolerant qubits.

Worksheet

Worksheet.

1. Prove: if $r = \mathrm{ord}_N(a)$ is even and $a^{r/2} \not\equiv -1 \pmod{N}$, then $\gcd(a^{r/2} - 1, N)$ yields a nontrivial factor.
2. Compute order of $a = 2$ modulo $N = 15$ and recover factors using the above step.

3.4 Week 1 Homework + solution sketches

Homework.

1. (Reduction) Prove $\varphi(N)$ factors semiprimes; implement in pseudocode.
2. (Attack taxonomy) For each of Pollard ρ, ECM, QS, GNFS: state what property makes it effective and what input sizes it targets.
3. (Order-finding) Show how order-finding implies factoring for random a (state probability assumptions clearly).

Solution sketch. (1) Use $p + q = N - \varphi(N) + 1$ and solve quadratic. (2) Pollard ρ/ECM: small factors; QS: mid-size; GNFS: largest general. (3) Standard argument: random a has even order with decent probability; if $a^{r/2} \neq -1 \bmod N$ then gcd gives factor.

4 Week 2: Discrete Logarithms (Finite Fields & Elliptic Curves)

4.1 Lecture 2A: DLP/CDH/DDH and Generic Algorithms

Slides-outline

Slide: Definitions

- DLP: given g, h, find x with $g^x = h$ in cyclic group G of order n.
- CDH/DDH: compute g^{ab} / distinguish g^{ab} from random.

Slide: Generic algorithms

- Baby-step/giant-step: $\tilde{O}(\sqrt{n})$ time+memory.
- Pollard ρ: $\tilde{O}(\sqrt{n})$ time, low memory.
- Generic lower bound idea: need $\Omega(\sqrt{n})$ in black-box groups.

Instructor notes

Instructor notes. Drive home: in *generic* groups ECDLP is not easier than $\sqrt{n}$. Hence curves choose $n \approx 2^{256}$ for 128-bit classical security. Explain random-walk collision philosophy.

Worksheet

Worksheet.

1. Work baby-step/giant-step on $\mathbb{Z}_{29}^{\times}$ with generator $g = 2$, target $h = 18$.
2. Explain why Pollard ρ is a collision-finding algorithm on a pseudorandom map.

4.2 Lecture 2B: Pohlig–Hellman and Subgroup Attacks

Slides-outline

Slide: Pohlig–Hellman

- If $n = \prod p_i^{e_i}$ then DLP reduces to each prime power.
- Solve residues, combine via CRT.
- Implication: choose prime-order subgroup (or with one large prime factor).

Instructor notes

Instructor notes. Provide a worked example with n having small factors. Emphasize that many protocol failures come from wrong subgroup choice or missing validation.

Worksheet

Worksheet.

1. Do Pohlig–Hellman in a toy group where $n = 2^2 \cdot 3 \cdot 5$.
2. Explain what can go wrong in Diffie–Hellman if group membership is not validated.

4.3 Lecture 2C: Index Calculus vs ECDLP, Pairing Reductions, Shor

Slides-outline

Slide: Finite-field DLP

- Index calculus: factor base, relations, linear algebra, individual logs.
- Best-known in prime fields: NFS-DL ($L_p[1/3, (64/9)^{1/3}]$).

Slide: ECDLP

- Generic attacks dominate for well-chosen curves: $\tilde{O}(\sqrt{n})$.
- MOV/Frey–Rück: special curves reduce to finite-field DLP via pairings.

Slide: Quantum

- Shor solves DLP in abelian groups in $\mathrm{poly}(\log n)$ time.

Instructor notes

Instructor notes. Stress that "ECDLP is harder" is conditional: it avoids known index-calculus subexponential methods. But special curves (supersingular / small embedding degree) can invalidate this.

4.4 Week 2 Homework + solution sketches

Homework.

1. Prove correctness of Pohlig–Hellman and give runtime in terms of factorization of n.
2. Compare DLP hardness in $\mathbb{F}_p^\times$ vs elliptic curves of comparable size; justify using attack classes.
3. Show DDH $\Rightarrow$ IND-CPA security of ElGamal (standard reduction outline).

Solution sketch. (1) Use lifting to prime powers + CRT. (2) Finite fields admit index calculus; generic for EC. (3) Hybrid argument: replace g^{ab} with random if DDH hard.

5 Week 3: Lattices (SVP/CVP, SIS/LWE) and Cryptanalysis Toolkit

5.1 Lecture 3A: Geometry of Numbers Essentials

Slides-outline

Slide: Lattices

- $\mathcal{L}(B) = \{Bz : z \in \mathbb{Z}^d\}$, determinant/covolume, dual lattice.
- Successive minima $\lambda_1, \lambda_2, \ldots$.

Slide: Minkowski

- Statement: $\lambda_1(\mathcal{L}) \le \sqrt{d}\,\det(\mathcal{L})^{1/d}$.
- Interpret: short vectors exist but finding them is hard.

Instructor notes

Instructor notes. Give geometric intuition: fundamental parallelepiped volume; convex body argument. Make sure students can compute determinants and duals in low dimensions.

Worksheet

Worksheet.

1. For $B = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}$ compute $\det(\mathcal{L})$ and one nonzero short vector.
2. Compute the dual lattice basis $B^{-\top}$ and verify pairing integrality.

5.2 Lecture 3B: SVP/CVP, LLL/BKZ, Enumeration/Sieving

Slides-outline

Slide: Problems

- SVP/CVP and approximation γ-SVP/γ-CVP.
- Algorithm families: reduction (LLL/BKZ), enumeration, sieving.

Slide: LLL vs BKZ

- LLL: poly-time, exponential approximation factor.
- BKZ: parameter β improves quality; dominates real cryptanalysis.

Instructor notes

Instructor notes. Keep BKZ "concept-only": local SVP on blocks, iterative. If asked for numbers: mention that security estimates use BKZ blocksize β as main knob.

Worksheet

Worksheet.

1. Run (by hand) a single LLL size-reduction + swap step on a 2D basis.
2. Explain why enumeration complexity drops after basis reduction.

5.3 Lecture 3C: SIS/LWE + Attack Taxonomy (Primal/Dual/Hybrid/BKW)

Slides-outline

Slide: SIS

- Given $A \in \mathbb{Z}_q^{n\times m}$ find short nonzero x with $Ax \equiv 0 \pmod{q}$.

Slide: LWE

- Distinguish $(a, \langle a, s\rangle + e)$ from uniform; search-LWE recovers s.

Slide: Attacks

- Primal: embed to CVP/SVP, solve with BKZ+enum/sieve.
- Dual: find short dual vector to distinguish.
- Hybrid: guess some secret coordinates + reduce dimension.
- BKW: combinatorial sample combining; parameter-dependent.

Instructor notes

Instructor notes. This lecture is about how cryptanalysts reason: "dimension drives security". Explain qualitatively how q, noise α, and dimension interact in primal/dual attacks.

5.4 Week 3 Homework + solution sketches

Homework.

1. Prove $\det(\mathcal{L}^*) = 1/\det(\mathcal{L})$ for full-rank lattices.
2. In dimension 2, prove Minkowski's bound using area and convexity.
3. Give a one-page "attack selection guide" for LWE: when would you try primal vs dual vs hybrid vs BKW?

Solution sketch. (1) $\mathcal{L} = B\mathbb{Z}^d$, $\mathcal{L}^* = B^{-\top}\mathbb{Z}^d$, determinant transforms by $|\det(\cdot)|$. (2) Use symmetric convex body disk of area $> 4\det(\mathcal{L})$. (3) Primal favored at certain noise; dual when short dual vectors exist; hybrid when secret small/structured; BKW when many samples and moderate noise.

6 Week 4: Codes (Syndrome Decoding) and ISD Cryptanalysis

6.1 Lecture 4A: Codes, Syndromes, Decoding Basics

Slides-outline

Slide: Linear codes

- $[n, k]_q$ linear code; generator G; parity-check H.
- Hamming weight/distance; decoding as nearest codeword problem.

Slide: Syndrome

- For $r = c + e$, $s = Hr^\top = He^\top$ depends only on error.

Instructor notes

Instructor notes. Work a tiny $[7, 4]$ Hamming code example if time; otherwise keep conceptual. Make students comfortable with matrix equations over $\mathbb{F}_2$.

Worksheet

Worksheet.

1. Given H, compute syndrome of a received word and correct a single-bit error (toy).
2. Show that syndrome decoding is solving for a low-weight vector in an affine subspace.

6.2 Lecture 4B: Hard Problems (SD/MDP) and McEliece Context

Slides-outline

Slide: Syndrome Decoding (SD)

- Input: (H, s, t); output e with $He^\top = s$, $w_H(e) \leq t$.

Slide: McEliece

- Public code should look random; secret structure allows fast decoding.
- Attacker: generic SD (ISD) unless structure leaks.

6.3 Lecture 4C: ISD (Prange $\rightarrow$ Stern/Dumer/BJMM) + Quantum Notes

Slides-outline

Slide: Prange ISD

- Guess information set I of size k avoiding error positions.
- Success probability $\approx \binom{n-t}{k} / \binom{n}{k}$.

Slide: Modern ISD

- Stern/Dumer/BJMM: meet-in-the-middle improvements reduce exponent.
- Quantum: Grover speeds the guessing layers (model-dependent).

Instructor notes

Instructor notes. Derive Prange probability in class; it is very accessible to mathematicians. Explain that modern ISD refinements optimize constant factors/exponents via clever splitting.

6.4 Week 4 Homework + solution sketches

Homework.

1. Derive Prange expected work factor; plug in small toy parameters.
2. Implement Prange on random binary codes (tiny) and compare to brute force.
3. Explain what a "structural attack" means in code-based crypto and give one plausible distinguisher idea.

Solution sketch. (1) Inverse of success prob. (2) Empirical scaling matches combinatorial estimates. (3) Distinguisher examples: unusually low-weight dual codewords, rank properties, automorphism group size, etc.

7 Week 5: Isogenies (Elliptic Curves, Graphs, Attacks)

7.1 Lecture 5A: Elliptic Curve Essentials (finite fields)

Slides-outline

Slide: Elliptic curves

- $E/\mathbb{F}_q : y^2 = x^3 + ax + b$, $\Delta \neq 0$.
- Group law; torsion; Hasse bound (context).

Instructor notes

Instructor notes. Don't re-teach full EC theory; focus on what is needed: finite abelian group of points + morphisms. Optionally mention supersingular vs ordinary as a taxonomy.

7.2 Lecture 5B: Isogenies (kernels, degrees, evaluation)

Slides-outline

Slide: Isogeny definition

- Group homomorphism given by rational maps; finite kernel; degree.
- Separable isogeny determined by its kernel; Vélu gives explicit formula.

7.3 Lecture 5C: Hardness + Attacks (graph search, commutative actions, quantum)

Slides-outline

Slide: Hard problems

- Supersingular path-finding: find isogeny between E and E'.
- CSIDH-style: recover class-group action element (commutative hidden shift flavor).

Slide: Attacks

- Meet-in-the-middle / bidirectional search (Delfs–Galbraith style).
- Protocol-specific breaks (e.g. SIDH/SIKE) vs generic problem.
- Quantum: Kuperberg-type subexponential for commutative hidden shift settings.

7.4 Week 5 Homework + solution sketches

Homework.

1. Prove $\deg(\varphi \circ \psi) = \deg(\varphi)\deg(\psi)$.
2. Explain why kernels classify separable isogenies (state carefully; prove a special case).
3. Compare "path-finding" vs "hidden shift" formulations and their algorithmic consequences.

Solution sketch. (1) Degree of morphisms multiplies under composition. (2) In separable case, quotient by finite subgroup yields isogeny; Vélu constructs it. (3) Hidden shift allows Fourier methods (Kuperberg); generic path-finding is graph search.

8 Week 6: Multivariate (MQ) — Algebraic Attacks and Trapdoor Structure

8.1 Lecture 6A: MQ as Polynomial System Solving

Slides-outline

Slide: MQ

- Given quadratic $f_1, \ldots, f_m \in \mathbb{F}_q[x_1, \ldots, x_n]$, find $x \in \mathbb{F}_q^n$ with $f_i(x) = 0$.
- View as variety $V(I)$ for ideal $I = \langle f_1, \ldots, f_m \rangle$.

8.2 Lecture 6B: Gröbner Bases and Degree of Regularity

Slides-outline

Slide: Gröbner

- Term orders; leading term; elimination under lex.
- F4/F5 as efficient engines; complexity depends on degree of regularity.

8.3 Lecture 6C: XL/Hybrid/MinRank (Structured Attacks)

Slides-outline

Slide: Attack families

- XL/relinearization: multiply, linearize, solve linear system.
- Hybrid: guess k variables, solve remaining.
- MinRank/rank attacks exploit matrix structure of quadratic forms.

8.4 Week 6 Homework + solution sketches

Homework.

1. Convert a quadratic system over odd characteristic into matrix form; identify rank conditions.
2. Analyze hybrid complexity $q^k \cdot T(n-k)$; optimize k for a toy model $T(t) = q^{ct}$.
3. Solve a small MQ instance over $\mathbb{F}_2$ by linearization; compare to brute force.

Solution sketch. (1) Quadratic form $\leftrightarrow$ symmetric matrix after completing square; cross-terms map to off-diagonal. (2) Minimize exponent: $k + c(n-k) = cn + (1-c)k$ so choose $k = 0$ if $c < 1$, etc. (3) Linearization works if enough equations / low degree growth.

9 Week 7: Hash Functions — Games, Bounds, Constructions, Structural Attacks

9.1 Lecture 7A: Formal Games (CR/SPR/OW)

Slides-outline

Slide: Hash family

- $H : \{0,1\}^* \to \{0,1\}^n$, security notions as games.
- Collision resistance, second-preimage, preimage.

9.2 Lecture 7B: Generic Bounds (Birthday, Preimages) + Proofs

Slides-outline

Slide: Birthday

- Collision after $\approx 2^{n/2}$ queries.
- Approx formula: $1 - \exp(-q(q-1)/2^{n+1})$.

9.3 Lecture 7C: Merkle–Damgård, Length Extension, HMAC, Quantum

Slides-outline

Slide: Merkle–Damgård

- Iterated compression + padding.
- Length extension and why $H(k\|m)$ is a bad MAC.
- HMAC fixes it (double hash with keyed pads).
- Quantum: Grover preimages $\approx 2^{n/2}$.

9.4 Week 7 Homework + solution sketches

Homework.

1. Prove the birthday bound formula (use occupancy or Poisson approximation).
2. Demonstrate length extension in an idealized Merkle–Damgård model.
3. Given n-bit hash output, compute classical vs quantum work for preimages and collisions; infer recommended n for 128-bit post-quantum preimage security.

Solution sketch. (1) Probability no collision $\approx \prod_{i=0}^{q-1}(1 - i/2^n) \approx e^{-q(q-1)/2^{n+1}}$. (2) Internal chaining value after m lets extend with known padding and extra blocks. (3) Preimage: classical 2^n, quantum $2^{n/2}$, so for 128-bit PQ preimage choose $n \approx 256$.

10 Capstone (Optional): One Comparative Lecture + Exam-Style Questions

Slides-outline

Slide: Compare families

- Shor breaks factoring/DLP; Grover halves preimage exponent; others survive (no known poly-time).
- "Security knob": modulus size (factoring/DLP), group order (ECDLP), dimension (lattices), length/weight (codes), graph size/path length (isogenies), degree of regularity (MQ), output length (hash).

Exam-style questions

1. Explain why Pohlig–Hellman forces cryptographers to use prime-order subgroups.
2. Given an LWE instance, argue (qualitatively) how increasing q changes primal vs dual attack feasibility.
3. Compare birthday vs Grover and deduce hash output sizes for post-quantum targets.

11 How to Design Good Questions (Instructor Toolkit)

11.1 Learning objective → question template

For each topic, target a mix of:

1. **Definition checks** (precision): "State/derive the formal definition; identify inputs/outputs; specify distribution."
2. **Reduction problems** (mathematical thinking): "Show $A \leq B$ via explicit oracle reduction; track success probability."
3. **Algorithm traces** (mechanics): "Run the algorithm on a toy instance; show intermediate steps."
4. **Complexity reasoning** (asymptotics): "Explain why runtime is $\tilde{O}(\sqrt{n})$ / $L_N[\alpha, c]$ / $2^{\Theta(d)}$."
5. **Attack selection** (cryptanalytic judgment): "Given parameters/structure, which attack dominates and why?"
6. **Failure-mode questions** (engineering reality): "What breaks if validation/randomness is wrong? Provide counterexample."
7. **Proof-based extensions** (math depth): "Prove a standard lemma (Minkowski in 2D, Prange probability, birthday bound)."

11.2 Difficulty ladder (use for worksheets/homework/exams)

For each concept, create 4 tiers:

- **Tier 1 (warm-up):** recall/compute; single idea.
- **Tier 2 (core):** 2–3 steps; requires correct definitions.
- **Tier 3 (integration):** connects two concepts (e.g., DLP + subgroup structure; LWE + BKZ intuition).
- **Tier 4 (research-flavored):** open-ended but gradable: justify assumptions, compare attacks, critique parameter choices.

11.3 Common pitfalls to avoid

- Overly large toy numbers: keep hand-computable (e.g., primes < 50; lattice dimension 2 or 3; codes length ≤ 12).
- Vague prompts: force explicit input/output and probability space.
- "Prove hardness": instead ask to prove *reductions*, *bounds*, or *attack correctness*.
- Mixing security notions: be explicit about search vs decision vs distinguishing.

11.4 Grading rubrics (quick)

- **Definitions:** correct quantifiers, domains, modulo conventions.
- **Reductions:** explicit oracle calls; success probability; running time bound.
- **Algorithm traces:** correct intermediate computations; verify condition checks (gcd, smoothness, syndrome, etc.).
- **Attack selection:** justified by structure/parameters; not name-dropping.

12 Practice Problems by Topic (with short solution notes)

12.1 Week 1: Integer Factorization

Tier 1–2 (warm-up/core)

Exercise 12.1 (Factoring vs Euler totient). *Let $N = pq$ where p, q are distinct odd primes. Show that knowing $\varphi(N)$ allows recovery of p and q in time polynomial in $\log N$.*

Remark 12.1. ***Solution note.*** *Compute $S = p+q = N-\varphi(N)+1$ and solve $X^2-SX+N = 0$.*

Exercise 12.2 (Order-finding implies factoring). *Let $N = pq$ be an RSA modulus. Suppose an oracle returns $\mathrm{ord}_N(a)$ for any $a \in (\mathbb{Z}/N\mathbb{Z})^\times$. Give a randomized algorithm that factors N using the oracle and analyze its success probability.*

Remark 12.2. ***Solution note.*** *Pick random a; get $r = \mathrm{ord}_N(a)$. If r even and $a^{r/2} \not\equiv -1 \pmod N$ then $\gcd(a^{r/2} - 1, N)$ yields a factor. Bound success away from 0 under standard arguments.*

Exercise 12.3 (Pollard $p-1$ success condition). *State precisely the condition under which Pollard $p-1$ finds a factor $p \mid N$. Give a worked example with $N = 187 = 11 \cdot 17$ and a suitable smoothness bound.*

Remark 12.3. ***Solution note.*** *If $p-1$ is B-smooth and $M = \mathrm{lcm}(1, \ldots, B)$ then $a^M \equiv 1 \pmod p$ for many a, so $\gcd(a^M - 1, N)$ reveals p.*

Tier 3–4 (integration/research-flavored)

Exercise 12.4 (Why linear algebra appears in QS). *Explain why the Quadratic Sieve collects exponent vectors modulo 2 over a factor base. Derive the linear algebra condition that guarantees a congruence of squares.*

Remark 12.4. ***Solution note.*** *Smooth relations give $x_i^2 - N = \prod p_j^{e_{ij}}$; if $\sum_i e_{ij} \equiv 0 \pmod 2$ for all j, then $\prod_i (x_i^2 - N)$ is a square; hence $x^2 \equiv y^2 \pmod N$.*

Exercise 12.5 (Attack selection). *You are given a 2048-bit RSA modulus N and told it may have a 200-bit prime factor. Which attack do you try first and why? Contrast ECM vs GNFS.*

Remark 12.5. ***Solution note.*** *ECM targets small/medium prime factors and is far cheaper than GNFS if such a factor exists.*

12.2 Week 2: Discrete Logarithms (Finite Fields and Elliptic Curves)

Tier 1–2

Exercise 12.6 (Baby-step/giant-step by hand). *In $G = \mathbb{Z}_{29}^\times$, let $g = 2$ and $h = 18$. Compute x such that $2^x \equiv 18 \pmod{29}$ using baby-step/giant-step.*

Remark 12.6. ***Solution note.*** *Take $m = \lceil \sqrt{28} \rceil = 6$, build baby steps $g^0, \ldots, g^5$, and giant steps hg^{-6j} until collision.*

Exercise 12.7 (Pohlig–Hellman core step). *Let $G = \langle g \rangle$ have order $n = p^e$. Show how to recover $x \bmod p^e$ from an oracle that solves DLP modulo p repeatedly (lifting).*

Remark 12.7. ***Solution note.*** *Use base-p expansion $x = \sum_{i=0}^{e-1} x_i p^i$ and solve successive digits by powering to n/p.*

Tier 3–4

Exercise 12.8 (Why ECDLP avoids index calculus (concept)). *Give a precise statement of what "index calculus" needs (smoothness notion and factor base), and explain why a naive analogue fails in generic elliptic-curve groups.*

Remark 12.8. ***Solution note.*** *Finite fields admit unique factorization of ideals/elements and smoothness probabilities; generic EC groups do not provide comparable decomposition structure for random points.*

Exercise 12.9 (MOV condition). *State the condition (in terms of embedding degree) under which MOV/Frey–Rück reduces ECDLP to finite-field DLP. Why is this avoided in standard curve selection?*

Remark 12.9. ***Solution note.*** *If there exists small k with $n \mid (q^k - 1)$, pairings map to $\mathbb{F}_{q^k}^\times$, where index calculus applies.*

Exercise 12.10 (Subgroup-validation failure). *Construct an explicit example where a DH implementation that fails to validate subgroup membership leaks information about the secret exponent.*

Remark 12.10. ***Solution note.*** *Use small-subgroup confinement: attacker sends element of small-order subgroup; responses leak exponent mod that order.*

12.3 Week 3: Lattices (SVP/CVP, SIS/LWE)

Tier 1–2

Exercise 12.11 (Compute determinant and dual (2D)). *Let* $B = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}$. *Compute* $\det(\mathcal{L}(B))$ *and a basis of the dual lattice.*

Remark 12.11. ***Solution note.*** $\det = 6$. *Dual basis is $B^{-\top}$, scaled appropriately; verify inner products are integers.*

Exercise 12.12 (Minkowski in dimension 2). *Prove Minkowski's first theorem bound in $\mathbb{R}^2$ using an area argument.*

Remark 12.12. ***Solution note.*** *Use convex centrally symmetric body of area $> 4\det(\mathcal{L})$ implies nonzero lattice point.*

Tier 3–4

Exercise 12.13 (LWE distinguishing bias (dual attack intuition)). *Suppose you find $y \in \mathbb{Z}_q^m$ such that $y^T A \equiv 0 \pmod{q}$ and y is short. Show that $y^T b$ has a distributional bias when (A, b) is LWE vs uniform.*

Remark 12.13. ***Solution note.*** *If $b = As + e$, then $y^T b \equiv y^T e \pmod{q}$; short y keeps $y^T e$ small (non-uniform).*

Exercise 12.14 (Attack selection guide). *Given LWE parameters (n, q, α) (noise rate), explain when primal vs dual attacks are expected to dominate. Your answer should explicitly reference: dimension reduction quality, target vector norm, and sample count.*

Remark 12.14. ***Solution note.*** *Primal: embedding finds close vector; dual: find short dual; hybrid trades dimension; BKW if many samples and moderate noise.*

12.4 Week 4: Codes (Syndrome Decoding, ISD)

Tier 1–2

Exercise 12.15 (Syndrome depends only on error)**.** *Let H be a parity-check matrix and $r = c+e$ with $c \in C$. Show $Hr^\top = He^\top$.*

Remark 12.15. *Solution note.* $Hc^\top = 0$, *so* $Hr^\top = H(c+e)^\top = He^\top$.

Exercise 12.16 (Prange success probability)**.** *In binary SD, assume an error vector has weight t. If an algorithm guesses an information set I of size k uniformly among $\binom{n}{k}$ choices, derive the probability that I avoids all t error positions.*

Remark 12.16. *Solution note.* $\Pr[I \cap \mathrm{supp}(e) = \emptyset] = \binom{n-t}{k}/\binom{n}{k}$.

Tier 3–4

Exercise 12.17 (Affine-subspace viewpoint)**.** *Show that the solution set to $He^\top = s$ is an affine subspace of $\mathbb{F}_2^n$ of dimension k. Interpret SD as finding a low-weight element in that affine space.*

Remark 12.17. *Solution note.* *Fix one solution e_0; all solutions are $e_0+\ker(H)$;* $\dim\ker(H) = k$.

Exercise 12.18 (Structural vs generic attacks)**.** *Explain (with a concrete statistic) how one might distinguish a structured public code (e.g. with many low-weight dual codewords) from a uniformly random code of the same parameters.*

Remark 12.18. *Solution note.* *Compute weight distribution of dual, automorphism group size, rank properties, etc.*

12.5 Week 5: Isogenies

Tier 1–2

Exercise 12.19 (Degree multiplicativity)**.** *Let $\varphi : E_1 \to E_2$ and $\psi : E_2 \to E_3$ be isogenies. Prove* $\deg(\psi \circ \varphi) = \deg(\psi)\deg(\varphi)$.

Remark 12.19. *Solution note.* *Degree of morphisms multiplies under composition; can be shown via function field extensions.*

Exercise 12.20 (Kernel determines separable isogeny (special case))**.** *State and prove: for a finite subgroup $K \leq E(\overline{\mathbb{F}}_q)$ of order coprime to* $\mathrm{char}(\mathbb{F}_q)$*, there exists a separable isogeny with kernel K.*

Remark 12.20. *Solution note.* *Quotient curve E/K exists; Vélu gives explicit formulas.*

Tier 3–4

Exercise 12.21 (Graph search complexity heuristic)**.** *Model a supersingular ℓ-isogeny graph as a random d-regular graph on M vertices. Estimate the expected meet-in-the-middle time to find a path between two random vertices.*

Remark 12.21. *Solution note.* *Bidirectional BFS to depth $\approx \frac{1}{2}\log_d M$ visits $\approx d^{\ell/2} \approx \sqrt{M}$ states; refine to $\tilde{O}(M^{1/2})$ or $p^{1/4}$ depending on the parameterization used.*

12.6 Week 6: Multivariate (MQ)

Tier 1–2

Exercise 12.22 (Linearization)**.** *Given quadratic equations over $\mathbb{F}_2$ in variables $x_1, \ldots, x_n$, define new variables $y_{ij} = x_i x_j$ (for $i \le j$) and write the system as linear equations in the y_{ij}. When is this sufficient to solve the system?*

Remark 12.22. ***Solution note.*** *If enough independent equations exist and consistency constraints are manageable; otherwise many spurious solutions.*

Tier 3–4

Exercise 12.23 (Hybrid complexity optimization)**.** *Suppose solving MQ in t variables costs $T(t) = q^{ct}$ operations. If you guess k variables, derive total cost $q^k T(n-k)$ and find the optimal k.*

Remark 12.23. ***Solution note.*** *Exponent is $k + c(n-k) = cn + (1-c)k$; if $c < 1$, minimize at $k = 0$; if $c > 1$, at $k = n$ (toy model). Real models have non-linear $T(t)$ so optimization is nontrivial.*

Exercise 12.24 (Matrix form of quadratic maps (odd characteristic))**.** *Show that any quadratic polynomial $f(x) \in \mathbb{F}_q[x_1, \ldots, x_n]$ (odd q) can be written as $x^\top A x + b^\top x + c$ with A symmetric.*

Remark 12.24. ***Solution note.*** *Use $x_i x_j$ cross terms; symmetrize using $(A + A^\top)/2$ since 2 is invertible.*

12.7 Week 7: Hash Functions

Tier 1–2

Exercise 12.25 (Birthday bound derivation)**.** *Let H be a random function into $\{0,1\}^n$. After q queries, show*

$$\Pr[\textit{collision}] \approx 1 - \exp\left(-\frac{q(q-1)}{2^{n+1}}\right).$$

Remark 12.25. ***Solution note.*** *Probability of no collision $\approx \prod_{i=0}^{q-1}(1 - i/2^n)$ and use $\log(1-x) \approx -x$.*

Exercise 12.26 (Length extension (Merkle–Damgård))**.** *In an iterated hash $H(m) = f(\cdots f(IV, m_1), \ldots, m_t)$ with MD padding, explain how $H(m \| pad(m) \| m')$ can be computed from $H(m)$ and $|m|$ without knowing m.*

Remark 12.26. ***Solution note.*** *$H(m)$ is the internal chaining value after padding; reuse as IV for extra blocks.*

Tier 3–4

Exercise 12.27 (Post-quantum sizing)**.** *If Grover gives preimages in $\Theta(2^{n/2})$ quantum queries, what output length n is needed for $\approx$ 128-bit post-quantum preimage security? Compare to collision security.*

Remark 12.27. ***Solution note.*** *Need $2^{n/2} \approx 2^{128} \Rightarrow n \approx 256$ for PQ preimages; collisions require larger for PQ depending on collision algorithm model.*

13 Ready-to-Use Question Sets (by class type)

13.1 Quick in-class checks (5–10 minutes each)

1. (Factoring) State the exact condition that makes Pollard $p-1$ succeed.
2. (DLP) Why does Pohlig–Hellman force prime-order subgroups?
3. (Lattices) Define the dual lattice and compute it for a given basis.
4. (Codes) Derive Prange success probability.
5. (MQ) Explain relinearization in one paragraph.
6. (Hash) Derive birthday bound in two lines using $\log(1-x) \approx -x$.

13.2 Recitation set (60–90 minutes)

Pick 1–2 per topic:

1. Baby-step/giant-step on a small finite field.
2. One LLL step on a 2D lattice basis + interpret geometric meaning.
3. Prange ISD expected trials for a toy (n, k, t).
4. A small MQ system over $\mathbb{F}_2$ solved by linearization.
5. Birthday bound + compute q for 50% collision probability.

13.3 Exam-style integrators

1. Compare classical vs quantum asymptotics for factoring, DLP, hash preimages, and one PQ family (lattices/codes/isogenies/MQ).
2. Given a parameter set for an LWE-based KEM, explain qualitatively which attack is expected to dominate and what parameter changes would harden it.
3. Given a hash output length, infer collision vs preimage security in classical and quantum models and recommend a safe output length.