Weekly Worksheet Pack (UG / MS / PhD): Hard Problems in Cryptography

Contents

1 Week 1 Worksheet Pack: Integer Factorization (RSA/Rabin)

Instructions. Answer all questions. For Masters/PhD, justify steps cleanly and state any assumptions.
Time guidance. UG: 45–60 min MS: 60–90 min PhD: 90–120 min

Level: Upper-undergrad

Problem 1.1 (Warm-up: $\varphi(N)$ and factoring)**.** *Let $N = 221$. Compute $\varphi(N)$ by factoring N directly, then verify that knowing $\varphi(N)$ allows you to recover the primes.*

Solution. $221 = 13{\cdot}17$, so $\varphi(N) = (13-1)(17-1) = 12{\cdot}16 = 192$. Given N and $\varphi(N)$, compute $S = p+q = N-\varphi(N)+1 = 221-192+1 = 30$. Solve $X^2 - SX + N = 0$: $X^2 - 30X + 221 = 0$ has roots $13, 17$.

Problem 1.2 (Pollard ρ intuition)**.** *Explain (in words) why Pollard ρ tends to find a factor p of N in about $O(\sqrt{p})$ steps. (Hint: birthday paradox.)*

Solution. Iterating a pseudo-random map mod p produces a random walk in a set of size p. A collision appears after about $\sqrt{p}$ steps (birthday bound). Such a collision often yields $\gcd(x_i - x_j, N)$ revealing p.

Problem 1.3 (Order-finding to gcd trick)**.** *Let $N = 15$ and $a = 2$. Compute $\mathrm{ord}_N(a)$ and use $\gcd(a^{r/2} \pm 1, N)$ to factor N.*

Solution. Compute powers mod 15: $2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16 \equiv 1$, so $r = 4$. Then $a^{r/2} = 2^2 = 4$. $\gcd(4-1, 15) = \gcd(3, 15) = 3$, $\gcd(4+1, 15) = \gcd(5, 15) = 5$.

Level: Masters

Problem 1.4 (Formal reduction: $\varphi(N)$ factors semiprimes)**.** *Let $N = pq$ with distinct odd primes. Prove that from $(N, \varphi(N))$ one can recover p, q in time polynomial in $\log N$.*

Solution. $\varphi(N) = (p-1)(q-1) = pq-(p+q)+1 = N-(p+q)+1$, hence $p+q = N-\varphi(N)+1 =: S$. Then p, q are integer roots of $X^2 - SX + N = 0$; compute discriminant $\Delta = S^2 - 4N$ and take square root.

Problem 1.5 (Pollard $p-1$ correctness condition)**.** *State precisely the condition on a prime factor $p \mid N$ that makes Pollard $p-1$ succeed with $M = \mathrm{lcm}(1, \ldots, B)$. Give a proof sketch.*

Solution. If $p-1$ is B-smooth, then for any a with $p \nmid a$, Fermat implies $a^{p-1} \equiv 1 \pmod{p}$, and since $p-1 \mid M$, also $a^M \equiv 1 \pmod{p}$. If additionally $a^M \not\equiv 1 \pmod{q}$ for the other factor, then $\gcd(a^M - 1, N) = p$. Randomizing a gives decent chance to avoid $a^M \equiv 1 \pmod{q}$.

Problem 1.6 (Why linear algebra appears in QS (concept))**.** *Explain how smooth relations in QS give a linear system over $\mathbb{F}_2$ and why a nontrivial nullspace vector yields a congruence of squares.*

Solution. Relations: $x_i^2 - N = \prod_j p_j^{e_{ij}}$ over factor base primes p_j. Take exponent vectors $e_i = (e_{i1}, \ldots) \bmod 2$. If $\sum_{i \in S} e_i \equiv 0 \bmod 2$, then $\prod_{i \in S}(x_i^2 - N)$ is a square, so $\left(\prod x_i\right)^2 \equiv y^2 \pmod{N}$, giving $(X-Y)(X+Y) \equiv 0 \pmod{N}$ and typically a factor via gcd.

Level: PhD

Problem 1.7 (Order-finding implies factoring: success probability)**.** *Let $N = pq$ be an odd semiprime. Assume you can compute $r = \mathrm{ord}_N(a)$ for uniformly random $a \in (\mathbb{Z}/N\mathbb{Z})^\times$. Give a complete reduction factoring N and prove the success probability is bounded below by a constant (over random a), under standard group-structure arguments.*

Solution. Reduction: pick random a, compute r. If r odd, retry. If r even, set $x = a^{r/2} \bmod N$. If $x \equiv -1 \pmod N$, retry. Else output $\gcd(x-1, N)$ (or $\gcd(x+1, N)$). Success: in $(\mathbb{Z}/p\mathbb{Z})^\times \times (\mathbb{Z}/q\mathbb{Z})^\times$, the order has even probability $\geq 1/2$ (since each cyclic factor has half elements of even order). Condition $a^{r/2} \not\equiv -1$ fails for at most a small fraction (roughly $\leq 1/2$) of choices, giving constant overall success (often quoted $\geq 1/4$).

Problem 1.8 (ECM heuristic scaling)**.** *State the (standard) heuristic dependence of ECM on the size of the smallest prime factor p and explain why ECM is used before GNFS in practice.*

Solution. ECM runtime depends mainly on p, roughly $\exp(\sqrt{2 \log p \log \log p})$ (heuristic), independent of the cofactor size. GNFS depends on N and is far more expensive if p is medium/small. Hence ECM is an efficient pre-pass to peel off smaller factors.

Problem 1.9 (Rabin inversion $\Rightarrow$ factoring (outline))**.** *For Blum integers $N = pq$ with $p \equiv q \equiv 3 \pmod 4$, outline a reduction from an oracle that outputs a square root of a random quadratic residue mod N to factoring N.*

Solution. A random quadratic residue has four square roots mod N. If the oracle outputs a "different" root than a known one, their difference shares a nontrivial gcd with N. Sample x, set $y = x^2 \bmod N$, get root z. With good probability $z \not\equiv \pm x \pmod N$, then $\gcd(x - z, N)$ yields p or q.

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

Instructions. Answer all questions. For Masters/PhD, justify steps cleanly and state any assumptions.
Time guidance. UG: 45–60 min MS: 60–90 min PhD: 90–120 min

Level: Upper-undergrad

Problem 2.1 (Baby-step/giant-step (hand computation))**.** *In $G = \mathbb{Z}_{29}^\times$ let $g = 2$ and $h = 18$. Find x with $2^x \equiv 18 \pmod{29}$ using baby-step/giant-step.*

Solution. $|G| = 28$, take $m = \lceil \sqrt{28} \rceil = 6$. Baby steps: $2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32 \equiv 3$. Compute $2^{-6} \equiv (2^6)^{-1}$. $2^6 = 64 \equiv 6$, inverse of 6 mod 29 is 5 (since $6 \cdot 5 = 30 \equiv 1$). Giant steps: $h(2^{-6})^j = 18 \cdot 5^j$: $j = 0 : 18$ (not in list), $j = 1 : 18 \cdot 5 = 90 \equiv 3$ which matches baby step 2^5. So $x = 5 + 6 \cdot 1 = 11$.

Problem 2.2 (Why prime-order groups matter)**.** *Explain why using a group whose order has small prime factors can make discrete logs easier.*

Solution. Pohlig–Hellman reduces DLP in a group of order $n = \prod p_i^{e_i}$ to DLPs in the prime-power subgroups. If small primes divide n, those subproblems are easy, and CRT recombines them to solve the full DLP.

Problem 2.3 (Pollard ρ runtime intuition)**.** *Give a one-paragraph explanation for why Pollard ρ typically takes about $\sqrt{n}$ steps to solve DLP in a group of size n.*

Solution. Pollard ρ creates a pseudo-random walk in a state space of size about n; a collision appears after about $\sqrt{n}$ steps (birthday). A collision corresponds to two representations of the same group element, yielding a linear relation in the unknown exponent that can be solved.

Level: Masters

Problem 2.4 (Pohlig–Hellman correctness)**.** *Let $G = \langle g \rangle$ have order $n = \prod_i p_i^{e_i}$. Prove that solving DLP in each subgroup of order $p_i^{e_i}$ and recombining via CRT yields the DLP solution modulo n.*

Solution. Let x satisfy $g^x = h$. For each i, raise both sides to $n/p_i^{e_i}$ to project into the unique subgroup of order $p_i^{e_i}$. This yields $g_i^{x \bmod p_i^{e_i}} = h_i$. Recover $x_i := x \bmod p_i^{e_i}$, then use CRT to reconstruct unique $x \bmod n$.

Problem 2.5 (Index calculus prerequisites (concept))**.** *State what structural properties make index calculus possible in $\mathbb{F}_p^\times$ and why the analogous idea does not directly apply to generic elliptic-curve groups.*

Solution. Index calculus needs a factor base and a notion of smoothness with usable probability (unique factorization of elements/ideals). In generic EC groups, random points do not admit an efficient decomposition over a small "factor base" with comparable smoothness behavior.

Problem 2.6 (MOV/Frey–Rück statement)**.** *State the embedding-degree condition under which ECDLP can reduce to finite-field DLP via pairings, and explain the security implication.*

Solution. If a curve over $\mathbb{F}_q$ has subgroup of order n with small embedding degree k such that $n \mid (q^k - 1)$, a pairing maps the subgroup into $\mathbb{F}_{q^k}^\times$ so ECDLP reduces to DLP there, enabling subexponential attacks. Thus standard curves avoid small embedding degree unless pairings are intended.

Level: PhD

Problem 2.7 (Generic lower bound (statement + interpretation))**.** *State a rigorous generic-group lower bound for DLP (e.g., Shoup-type) and explain what it does* not *say about specific representations.*

Solution. In the generic group model, any algorithm solving DLP with non-negligible probability needs $\Omega(\sqrt{n})$ group operations. It does not rule out faster algorithms exploiting special representations/structure (e.g., finite fields admit index calculus).

Problem 2.8 (Small-subgroup confinement attack)**.** *Construct an explicit DH attack in a group where $|G|$ has a small factor ℓ if the implementation fails to validate subgroup membership. Show what is learned about the secret exponent.*

Solution. Attacker sends element u of order ℓ. Victim returns u^x. Since u^x depends only on $x \bmod \ell$, attacker learns $x \bmod \ell$ by table lookup. Repeating across several small factors and CRT recovers x.

Problem 2.9 (DDH vs CDH nuance)**.** *Give an example of a setting where CDH is believed hard but DDH is easy, and explain the mechanism.*

Solution. Pairing-friendly groups: a bilinear pairing $e : G \times G \to G_T$ allows distinguishing tuples (g, g^a, g^b, g^{ab}) from random by checking whether $e(g^a, g^b) = e(g, g^{ab})$; thus DDH easy, CDH still believed hard.

3 Week 3 Worksheet Pack: Lattices (SVP/CVP, SIS/LWE) and Cryptanalysis

Instructions. Answer all questions. For Masters/PhD, justify steps cleanly and state any assumptions.
Time guidance. UG: 45–60 min MS: 60–90 min PhD: 90–120 min

Level: Upper-undergrad

Problem 3.1 (Compute determinant and a short vector). *Let* $B = \begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}$. *Compute* $\det(\mathcal{L}(B))$ *and find a nonzero lattice vector of small Euclidean norm.*

Solution. $\det(\mathcal{L}) = |\det(B)| = 6$. Vectors are $z_1(2,0) + z_2(1,3) = (2z_1 + z_2, 3z_2)$. Try small integers: $z_2 = 0, z_1 = \pm 1$ gives $(\pm 2, 0)$ with norm 2 (very small).

Problem 3.2 (SVP vs CVP (definitions)). *State SVP and CVP in your own words, and give one reason CVP seems "harder".*

Solution. SVP: find shortest nonzero lattice vector. CVP: given target point, find lattice point closest to it. CVP generalizes nearest neighbor / decoding problems and includes SVP-like hardness; algorithmically it is at least as challenging.

Problem 3.3 (Toy LWE sample). *Let* $q = 11$, $n = 2$, *secret* $s = (3, 4)$. *For* $a = (2, 5)$ *and error* $e = 1$, *compute* $b = \langle a, s \rangle + e \pmod{q}$.

Solution. $\langle a, s \rangle = 2 \cdot 3 + 5 \cdot 4 = 6 + 20 = 26 \equiv 4 \pmod{11}$, so $b = 4 + 1 = 5 \pmod{11}$.

Level: Masters

Problem 3.4 (Dual lattice computation). *For full-rank* $B \in \mathbb{R}^{d \times d}$, *show* $\mathcal{L}(B)^* = \mathcal{L}(B^{-\top})$ *and* $\det(\mathcal{L}^*) = 1/\det(\mathcal{L})$.

Solution. $\mathcal{L}(B) = B\mathbb{Z}^d$. $y \in \mathcal{L}^*$ iff $y^\top Bz \in \mathbb{Z}$ for all z, i.e., $(B^\top y) \in \mathbb{Z}^d$. Thus $y = B^{-\top} w$ with $w \in \mathbb{Z}^d$. Determinant transforms by $|\det(B^{-\top})| = 1/|\det(B)|$.

Problem 3.5 (Minkowski in 2D). *Prove Minkowski's first theorem in dimension* 2 *(area argument).*

Solution. Let $\mathcal{L}$ have covolume $\det(\mathcal{L})$. Any centrally symmetric convex body $K \subset \mathbb{R}^2$ of area $> 4\det(\mathcal{L})$ contains a nonzero lattice point. Take K as disk of radius r where $\pi r^2 = 4\det(\mathcal{L}) + \varepsilon$; then $\lambda_1(\mathcal{L}) \le r$, yielding $\lambda_1 \le \sqrt{4\det/\pi}$ and a comparable form of Minkowski; relate to $\sqrt{2}\det^{1/2}$ constants.

Problem 3.6 (LWE dual bias (core idea)). *Assume* $A \in \mathbb{Z}_q^{m \times n}$, $b = As + e \pmod{q}$ *with small error vector* e. *If you find a short* $y \in \mathbb{Z}^m$ *with* $y^\top A \equiv 0 \pmod{q}$, *show* $y^\top b \equiv y^\top e \pmod{q}$ *and explain why that distinguishes from uniform.*

Solution. Multiply: $y^\top b \equiv y^\top As + y^\top e \equiv 0 + y^\top e \pmod{q}$. If y and e are short, $y^\top e$ is concentrated near 0 (mod q), unlike uniform.

Level: PhD

Problem 3.7 (Primal embedding sketch for LWE $\to$ BDD/CVP)**.** *Sketch how a collection of LWE samples can be embedded into a lattice so that recovering s corresponds to a bounded-distance decoding instance. Explicitly identify: lattice basis, target, and where the noise appears.*

Solution. Standard embedding: build $A \in \mathbb{Z}_q^{m\times n}$, $b \in \mathbb{Z}_q^m$ with $b = As + e$. Consider lattice $\Lambda = \{(x, Ax) \in \mathbb{Z}^n \times \mathbb{Z}^m : x \in \mathbb{Z}^n\} + q(\{0\}^n \times \mathbb{Z}^m)$ (various equivalent constructions). Target relates to $(0, b)$; difference to a lattice point encodes e. When e is small, this becomes BDD/CVP.

Problem 3.8 (Attack-selection reasoning)**.** *Give a principled, parameter-based argument (qualitative but precise) for when dual attacks beat primal attacks on LWE. Your argument must reference: (i) required BKZ blocksize for a target vector norm, (ii) sample count m, (iii) modulus q and noise rate.*

Solution. Dual attacks need a short vector in the dual lattice of samples; success depends on producing y with small norm so that $y^\top e$ is biased. More samples increase dimension of the dual lattice and can make shorter dual vectors exist (helping dual). Primal attacks aim to decode (recover s, e) via embedding; higher q and smaller noise can favor primal. Compare required BKZ blocksize: dual requires shortness relative to $q/\|e\|$ threshold; primal requires decoding radius relative to Gram-Schmidt lengths after reduction.

Problem 3.9 (SIS as a lattice problem)**.** *Show how SIS instances correspond to finding short vectors in a q-ary lattice and identify the lattice explicitly.*

Solution. Given $A \in \mathbb{Z}_q^{n\times m}$, define $\Lambda_q(A) = \{x \in \mathbb{Z}^m : Ax \equiv 0 \pmod q\}$, a full-rank lattice of determinant q^n (under rank conditions). SIS asks for a short nonzero vector in this lattice (SVP-like).

4 Week 4 Worksheet Pack: Codes (Syndrome Decoding) and ISD Attacks

Instructions. Answer all questions. For Masters/PhD, justify steps cleanly and state any assumptions.
Time guidance. UG: 45–60 min MS: 60–90 min PhD: 90–120 min

Level: Upper-undergrad

Problem 4.1 (Syndrome depends only on error)**.** *Let $C = \{c \in \mathbb{F}_2^n : Hc^\top = 0\}$ and $r = c + e$ with $c \in C$. Show that $Hr^\top = He^\top$.*

Solution. $Hr^\top = H(c + e)^\top = Hc^\top + He^\top = 0 + He^\top$.

Problem 4.2 (Toy decoding)**.** *Let* $H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$. *Compute the syndrome of $r = (1, 0, 1, 0, 0)$ and decide whether it is a valid codeword.*

Solution. Compute $s = Hr^\top$ over $\mathbb{F}_2$: Row1: $1 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 = 1 + 1 + 0 = 0$, Row2: $0 + 0 + 1 \cdot 1 = 1$, Row3: $0 + 0 = 0$. So $s = (0, 1, 0) \neq 0$, not a codeword.

Problem 4.3 (Prange probability (numerical))**.** *For $n = 10, k = 5, t = 2$, compute Prange's success probability $\binom{n-t}{k}/\binom{n}{k}$.*

Solution. $\binom{8}{5}/\binom{10}{5} = 56/252 = 2/9$.

Level: Masters

Problem 4.4 (Affine-subspace structure). *Fix $H \in \mathbb{F}_2^{(n-k)\times n}$ and syndrome s. Show the solution set $\{e : He^\top = s\}$ is an affine subspace of dimension k.*

Solution. If e_0 is one solution, then all solutions are $e_0 + \ker(H)$. $\mathrm{Rank}(H) = n - k$ (typical), so $\dim \ker(H) = n - (n-k) = k$.

Problem 4.5 (Derive Prange success probability). *Assume error e has weight t and information set I is a uniformly random k-subset of $\{1, \ldots, n\}$. Derive $\Pr[I \cap \mathrm{supp}(e) = \emptyset]$.*

Solution. Number of k-subsets avoiding the t error positions is $\binom{n-t}{k}$ out of $\binom{n}{k}$ total, so ratio.

Problem 4.6 (ISD viewpoint). *Explain in 5–8 lines how Prange reduces SD to repeatedly solving a linear system and why its cost is governed by the above probability.*

Solution. Choose I; if I avoids errors, the restriction to I gives correct information set and linear system yields candidate error. Success probability is probability I avoids support. Expected trials is inverse, so runtime scales accordingly.

Level: PhD

Problem 4.7 (ISD improvements (conceptual)). *Explain (conceptually) how Stern/Dumer/BJMM-type ISD improves over Prange. Your answer must identify at least one meet-in-the-middle split and what collision it searches for.*

Solution. They partition coordinates and seek partial sums with matching syndromes, using list merging to find low-weight combinations. Instead of guessing the entire information set free of errors, they allow some errors inside and correct via MITM, reducing exponent.

Problem 4.8 (Structural attacks vs generic ISD). *Give two concrete statistics that could distinguish a public code from a random code, enabling a structural attack. Explain why each statistic could plausibly help recover structure.*

Solution. Examples: unusually many low-weight codewords in dual; large automorphism group; rank properties in alternant/Goppa parity-check form; non-random weight distribution. Such features can leak algebraic structure to reconstruct secret decoding.

Problem 4.9 (Quantum note: Grover layering). *Explain how Grover-style amplitude amplification can speed up ISD-style algorithms at a high level, and why "halve the exponent" is an oversimplification.*

Solution. If algorithm is "search over guesses + classical verification," Grover can quadratically speed up the outer search. But ISD has internal list-merging costs and nonuniform success probabilities; optimal quantum speedup depends on balancing these subroutines.

5 Week 5 Worksheet Pack: Isogenies of Elliptic Curves (Graphs, Actions, Attacks)

Instructions. Answer all questions. For Masters/PhD, justify steps cleanly and state any assumptions.
Time guidance. UG: 45–60 min MS: 60–90 min PhD: 90–120 min

Level: Upper-undergrad

Problem 5.1 (Basic definition check). *Define an isogeny $\varphi : E_1 \to E_2$ of elliptic curves over a finite field. What are its key properties?*

Solution. A nonconstant morphism of curves defined over the field that is also a group homomorphism on points. It has finite kernel; degree is degree of the morphism; composition preserves homomorphism property.

Problem 5.2 (Degree multiplicativity (statement)). *State the multiplicativity of degrees under composition of isogenies.*

Solution. If $\varphi : E_1 \to E_2$ and $\psi : E_2 \to E_3$ are isogenies, then $\deg(\psi \circ \varphi) = \deg(\psi)\deg(\varphi)$.

Problem 5.3 (Graph model intuition). *Explain how an isogeny problem can be viewed as path-finding in a graph.*

Solution. Vertices represent isomorphism classes of curves; edges represent small-degree isogenies (e.g., ℓ-isogenies). Finding an isogeny between two curves corresponds to finding a path between their vertices.

Level: Masters

Problem 5.4 (Degree multiplicativity proof sketch). *Give a proof sketch of* $\deg(\psi \circ \varphi) = \deg(\psi)\deg(\varphi)$ *using either morphism degrees or function fields.*

Solution. Degree of morphism equals degree of induced extension of function fields. Extensions compose with multiplicative degrees, so degrees multiply under composition.

Problem 5.5 (Kernel determines separable isogeny (statement)). *State precisely (with conditions) the theorem that a finite subgroup K determines a separable isogeny with kernel K.*

Solution. If $\mathrm{char}(\mathbb{F}_q) \nmid |K|$, then there exists a separable isogeny $\varphi : E \to E/K$ with kernel K, unique up to isomorphism of the codomain; Vélu gives explicit formulas.

Problem 5.6 (Meet-in-the-middle estimate (toy)). *Suppose a graph has M vertices and is "random-like." Argue why bidirectional search suggests work about $M^{1/2}$ to connect random endpoints.*

Solution. Expanding BFS trees from both ends to depth where explored sets have size about $\sqrt{M}$ yields expected intersection; time proportional to explored set size.

Level: PhD

Problem 5.7 (Path-finding vs hidden shift). *Contrast (i) supersingular path-finding formulations and (ii) commutative class-group action (CSIDH-style) formulations. Explain which one admits Kuperberg-type hidden shift algorithms and why.*

Solution. Commutative class-group actions can be phrased as hidden shift in an abelian group, enabling Fourier sampling/Kuperberg. Generic supersingular path-finding is a noncommutative-looking graph problem without the same abelian HSP structure.

Problem 5.8 (Protocol-specific breaks (separating layers)). *Give a clear "layer separation" explanation: what it means for a specific protocol (e.g., SIDH/SIKE) to be broken without concluding that all isogeny-based primitives are broken.*

Solution. A break may exploit special auxiliary structure (torsion points, extra information leakage, malformed public keys) not present in other schemes. Thus the particular computational assumption instantiated by that protocol fails, while different isogeny assumptions (e.g., CSIDH/SQISign settings) may remain plausible.

Problem 5.9 (Graph heuristics caveat)**.** *List two reasons why modeling an isogeny graph as a random regular graph can mislead complexity estimates.*

Solution. Graphs may have additional algebraic structure, nonuniform mixing at relevant sizes, special subgraphs/cycles, and cost asymmetries for evaluating edges. Also memory constraints and representation issues can dominate asymptotics.

6 Week 6 Worksheet Pack: Multivariate (MQ): Solving Quadratic Systems

Instructions. Answer all questions. For Masters/PhD, justify steps cleanly and state any assumptions.
Time guidance. UG: 45–60 min MS: 60–90 min PhD: 90–120 min

Level: Upper-undergrad

Problem 6.1 (MQ definition + brute force)**.** *Define MQ over* $\mathbb{F}_2$*. For the system*

$$x_1x_2 + x_1 + 1 = 0, \qquad x_2 + 1 = 0,$$

find all solutions in $\mathbb{F}_2^2$*.*

Solution. Second equation gives $x_2 = 1$. Plug into first: $x_1 \cdot 1 + x_1 + 1 = (x_1 + x_1) + 1 = 1 \neq 0$. So no solutions.

Problem 6.2 (Linearization idea)**.** *Explain linearization: introduce new variables for monomials. What goes wrong if you ignore consistency constraints like* $y_{12} = x_1x_2$*?*

Solution. Linearization turns quadratic equations into linear ones in monomial-variables, but solutions in the expanded space may not correspond to actual assignments of x, creating spurious solutions unless constraints are enforced.

Problem 6.3 (Hybrid idea (concept))**.** *What does it mean to "guess* k *variables and solve the rest algebraically"? Why can this help?*

Solution. Brute force over q^k assignments reduces the remaining system size to $n-k$ variables. If algebraic solving grows superlinearly with n, this trade can reduce total work.

Level: Masters

Problem 6.4 (Quadratic forms as matrices (odd characteristic))**.** *Assume* $\mathrm{char}(\mathbb{F}_q) \neq 2$*. Show any quadratic polynomial can be written as* $f(x) = x^\top Ax + b^\top x + c$ *with* A *symmetric.*

Solution. Write f as sum of terms $a_{ij}x_ix_j$ and $a_ix_i + c$. For $i \neq j$, split coefficient across (i, j) and (j, i) and symmetrize: $x^\top \left(\frac{A+A^\top}{2}\right) x$ equals the same quadratic form since 2 invertible.

Problem 6.5 (XL (one page derivation))**.** *Describe XL: choose degree* D*, multiply equations by monomials to reach degree* $\leq D$*, then linearize. Explain what determines whether the resulting linear system is overdetermined.*

Solution. Number of generated equations depends on m times count of monomials up to $D-2$. Number of unknown monomials is count of monomials up to degree D. If generated equations exceed unknowns with sufficient rank, linear system can be solved.

Problem 6.6 (Hybrid complexity)**.** *Assume solving MQ in t variables costs $T(t)$ operations. Show the cost of guessing k variables is $q^k T(n-k)$.*

Solution. There are q^k assignments; each yields a reduced system of size $n-k$ solved in $T(n-k)$; multiply.

Level: PhD

Problem 6.7 (Gröbner basis complexity drivers (conceptual))**.** *Explain why the degree of regularity (or solving degree) is central in estimating the complexity of F4/F5 for semi-regular MQ systems.*

Solution. F4/F5 build Macaulay matrices up to a critical degree; matrix sizes (and thus time/memory) grow combinatorially with that degree. The solving degree approximates the maximal degree needed before elimination yields a univariate (or triangular) form.

Problem 6.8 (MinRank connection)**.** *Explain (conceptually) how certain structured MQ instances reduce to MinRank problems and why that can yield faster attacks.*

Solution. Quadratic maps can be represented as linear combinations of matrices; finding a solution or trapdoor can correspond to finding a low-rank combination. If the scheme introduces low-rank structure, MinRank solvers can exploit it subexponentially/polynomially for weak parameters.

Problem 6.9 (Design critique)**.** *Propose two parameter/structure pitfalls that could make an MQ signature scheme vulnerable, and explain the attack class each enables.*

Solution. Overdetermined systems enable linearization/XL; low-rank hidden structure enables MinRank/Kipnis–Shamir style attacks; poor masking enables rank/distinguisher attacks.

7 Week 7 Worksheet Pack: Hash Functions: Security Games, Generic Bounds, Constructions

Instructions. Answer all questions. For Masters/PhD, justify steps cleanly and state any assumptions.
Time guidance. UG: 45–60 min MS: 60–90 min PhD: 90–120 min

Level: Upper-undergrad

Problem 7.1 (Collision vs preimage)**.** *For an n-bit hash output, state the approximate work for (i) finding a collision, (ii) finding a preimage, using generic attacks.*

Solution. Collision: about $2^{n/2}$ (birthday). Preimage: about 2^n (brute force).

Problem 7.2 (Birthday estimate (numerical))**.** *For $n = 32$, about how many random hash evaluations are needed for a 50% chance of* some *collision?*

Solution. Roughly $1.2 \cdot 2^{n/2} \approx 1.2 \cdot 2^{16} \approx 7.9 \times 10^4$.

Problem 7.3 (Length extension (concept))**.** *Explain in words what a length extension attack is and name a standard construction that prevents it.*

Solution. For Merkle–Damgård hashes, knowing $H(m)$ and $|m|$ lets compute $H(m\|pad(m)\|m')$ without m. HMAC prevents this by hashing with inner/outer keyed pads.

Level: Masters

Problem 7.4 (Birthday bound derivation). *Let H be a random function to $\{0,1\}^n$. Show that after q queries,*

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

Solution. No-collision probability $\prod_{i=0}^{q-1}(1-i/2^n)$. Take logs and approximate $\log(1-x) \approx -x$ for small x, giving $\exp(-q(q-1)/2^{n+1})$.

Problem 7.5 (Formal game: collision resistance). *Write the formal collision resistance game for a family $\{H_\lambda\}$ with $H_\lambda : \{0,1\}^* \to \{0,1\}^{n(\lambda)}$.*

Solution. Adversary gets 1^λ and outputs (x, x'); wins if $x \neq x'$ and $H_\lambda(x) = H_\lambda(x')$. CR: win probability is negligible for all PPT adversaries.

Problem 7.6 (Post-quantum sizing). *Given Grover yields preimages in $\Theta(2^{n/2})$ quantum queries, what n gives about 128-bit post-quantum preimage security?*

Solution. Need $2^{n/2} \approx 2^{128}$, hence $n \approx 256$.

Level: PhD

Problem 7.7 (Merkle–Damgård length extension (formal)). *In the iterated model $h_{i+1} = f(h_i, m_i)$ with MD strengthening padding, show how an adversary given $H(m)$ and $|m|$ can compute $H(m\|pad(m)\|m')$ for chosen suffix m' without knowing m.*

Solution. $H(m)$ equals chaining value after processing m and its padding. Treat that as new IV and feed blocks of m' (with correct padding accounting for total length). This reproduces the internal state evolution.

Problem 7.8 (Security notion separation). *Give an example (conceptual) of a hash family that is preimage-resistant but not collision-resistant, and explain why this does not contradict definitions.*

Solution. Preimage resistance concerns inverting a random output; collision resistance concerns finding any pair with same output. A function can be hard to invert yet easy to collide (e.g., truncate outputs severely; collisions easy by birthday, inversion still exponential in output size).

Problem 7.9 (Quantum collisions (model awareness)). *State why "Grover halves the exponent" is insufficient for collision resistance, and mention a known quantum collision-finding speedup at a high level.*

Solution. Collision finding is not simple unstructured search for a single marked item; it is a structured "find any pair" problem. There exist quantum algorithms (e.g., Brassard-Høyer-Tapp-type) improving collision search over classical birthday in some oracle models.

Printing / Distribution Notes

- To produce **student handouts**: keep `\def\showsolutions{0}`.
- To produce **instructor key**: set `\def\showsolutions{1}` (solutions appear in blue).
- You can split into separate PDFs per week by copying each Week section into its own file.

8 Quiz Bank (Tagged)

8.1 How to Use

- Each item is tagged by: **Topic**, **Level** (UG/MS/PhD), **Type** (Def/Alg/Red/Comp/Jdg/Sec), and **Points**.
- Suggested quiz format: 8–12 minutes, 10–15 points total.
- For each week, select 3–5 items: 1 Def, 1 Alg/Comp, 1 Red/Jdg, optional PhD extension.

8.2 Week 1 Quiz Bank: Integer Factorization

[Factoring — UG — Def — 2 pts] *State the factoring problem precisely*
State the integer factorization (search) problem as a function/relation problem, specifying input and required output.

Solution. Input $N \in \mathbb{Z}_{\geq 2}$ composite. Output d with $1 < d < N$ and $d \mid N$ (or full prime factorization).

[Factoring — UG — Comp — 3 pts] *Compute φ and verify relation*
Let $N = 91$. Factor N and compute $\varphi(N)$.

Solution. $91 = 7 \cdot 13$. $\varphi(N) = 6 \cdot 12 = 72$.

[Factoring — UG — Alg — 3 pts] *Order/gcd trick*
For $N = 21$ and $a = 2$, compute $\mathrm{ord}_N(a)$ and use $\gcd(a^{r/2} \pm 1, N)$ to factor N (if possible).

Solution. Compute powers mod 21: $2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32 \equiv 11, 2^6 \equiv 1$ so $r = 6$. Then $a^{r/2} = 2^3 = 8$. $\gcd(8-1, 21) = \gcd(7, 21) = 7$, $\gcd(8+1, 21) = \gcd(9, 21) = 3$.

[Factoring — MS — Red — 4 pts] *Show $\varphi(N)$ factors semiprimes*
Assume $N = pq$ with distinct primes. Given $(N, \varphi(N))$, derive p, q explicitly and justify correctness.

Solution. Let $S = p+q = N-\varphi(N)+1$. Solve $X^2 - SX + N = 0$; discriminant $\Delta = S^2 - 4N = (p-q)^2$.

[Factoring — PhD — Jdg — 5 pts] *Attack selection*
You suspect an RSA modulus N has a 160-bit prime factor. Choose between ECM and GNFS first and justify using asymptotic/heuristic scaling.

Solution. ECM depends primarily on the smallest prime factor; far cheaper than GNFS if a 160-bit factor exists. Run ECM first.

8.3 Week 2 Quiz Bank: Discrete Logarithms

[DLP — UG — Def — 2 pts] *Formal DLP definition*
Define DLP in a cyclic group $(G, \cdot)$ of order n.

Solution. Given generator g and $h \in G$, find $x \in \mathbb{Z}_n$ s.t. $g^x = h$.

[DLP — UG — Alg — 4 pts] *BSGS computation*
In $\mathbb{Z}_{23}^{\times}$, let $g = 5$, $h = 4$. Use baby-step/giant-step with $m = \lceil \sqrt{22} \rceil$ to find x with $5^x \equiv 4 \pmod{23}$.

Solution. $m = 5$. Baby: $5^0 = 1, 5, 2, 10, 4$. Hit: $h = 4$ equals baby step at exponent 4. So $x = 4$.

[DLP — MS — Red — 4 pts] *Pohlig–Hellman core idea*
Explain why DLP in a group of order $n = \prod p_i^{e_i}$ reduces to DLP modulo each $p_i^{e_i}$, and how CRT recombines.

Solution. Project by exponentiation $n/p_i^{e_i}$ into subgroup of order $p_i^{e_i}$, recover residues, combine by CRT.

[DLP — PhD — Sec — 5 pts] *Small subgroup confinement*

Describe an explicit DH attack when subgroup membership is not validated and $|G|$ has a small factor ℓ. What leaks about secret x?

Solution. Send element of order ℓ; response reveals $x \bmod \ell$; repeat and CRT.

8.4 Week 3 Quiz Bank: Lattices (SVP/CVP, SIS/LWE)

[Lattices — UG — Comp — 3 pts] *Determinant*

Given $B = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$, compute $\det(\mathcal{L}(B))$.

Solution. $\det(B) = 3 \cdot 2 - 1 \cdot 1 = 5$ so $\det(\mathcal{L}) = 5$.

[Lattices — UG — Def — 2 pts] *SVP vs CVP*

State SVP and CVP (search versions) in one sentence each.

Solution. SVP: shortest nonzero lattice vector. CVP: lattice vector closest to a target point.

[LWE — MS — Sec — 4 pts] *Dual bias identity*

Assume LWE samples $b = As + e \bmod q$. If $y^\top A \equiv 0 \bmod q$, show $y^\top b \equiv y^\top e \bmod q$ and explain bias.

Solution. Multiply: $y^\top b = y^\top As + y^\top e \equiv y^\top e$. If y, e short, distribution non-uniform.

[SIS — PhD — Red — 5 pts] *SIS as short vector in q-ary lattice*

Define $\Lambda_q(A) = \{x \in \mathbb{Z}^m : Ax \equiv 0 \bmod q\}$. Explain why SIS asks for a short nonzero vector in $\Lambda_q(A)$ and state determinant (under full-rank assumptions).

Solution. SIS exactly searches short $x \neq 0$ with congruence; $\Lambda_q(A)$ has determinant q^n if A has rank n mod q.

8.5 Week 4 Quiz Bank: Codes (Syndrome Decoding)

[Codes — UG — Comp — 3 pts] *Syndrome computation*

Over $\mathbb{F}_2$, let $H = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$ and $r = (1, 0, 1, 1)$. Compute syndrome $s = Hr^\top$.

Solution. Row1: $1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 = 1 + 0 + 0 + 1 = 0$. Row2: $0 + 0 \cdot ? + 1 \cdot 1 + 0 = 1$. So $s = (0, 1)$.

[Codes — MS — Red — 4 pts] *Prange probability*

Derive Prange success probability $\binom{n-t}{k} / \binom{n}{k}$ for weight-t error.

Solution. Count k-subsets avoiding t error positions.

[Codes — PhD — Jdg — 5 pts] *Structural vs generic*

Give one concrete statistic to distinguish a structured public code from random and explain how that could lead to a key-recovery (high level).

Solution. E.g., abnormal number of low-weight dual codewords; can indicate alternant/Goppa structure enabling reconstruction of hidden support/permutation.

8.6 Week 5 Quiz Bank: Isogenies

[Isogenies — UG — Def — 2 pts] *Define isogeny*

Define an isogeny of elliptic curves over $\mathbb{F}_q$ and mention kernel finiteness.

Solution. Nonconstant morphism of curves over $\mathbb{F}_q$ that is a group homomorphism; kernel finite; degree defined.

[Isogenies — MS — Red — 4 pts] *Degree multiplicativity*

Prove/sketch why $\deg(\psi \circ \varphi) = \deg(\psi) \deg(\varphi)$.

Solution. Use function field extensions (degrees multiply under composition).

[Isogenies — PhD — Jdg — 5 pts] *Modeling caveat*

Give two reasons why "random regular graph" heuristics can misestimate isogeny attack costs.

Solution. Graph has algebraic structure/nonuniform costs; representation/memory constraints dominate; mixing not immediate; edge evaluation asymmetry.

8.7 Week 6 Quiz Bank: Multivariate (MQ)

[MQ — UG — Comp — 3 pts] *Solve small MQ over* $\mathbb{F}_2$

Solve over $\mathbb{F}_2$: $x_1x_2 + x_2 = 0$, $x_1 + 1 = 0$.

Solution. $x_1 = 1$. First becomes $1 \cdot x_2 + x_2 = 0$ always, so solutions: $(1, 0)$ and $(1, 1)$.

[MQ — MS — Alg — 4 pts] *Linearization count*

For n variables, how many monomials of degree ≤ 2 exist (over any field)? Use this to estimate the number of variables after quadratic linearization.

Solution. $1+n+\binom{n}{2}+n = 1+n+\binom{n}{2}+n = 1+2n+\binom{n}{2}$ (quadratics include x_i^2 and $x_ix_j, i < j$).

[MQ — PhD — Jdg — 5 pts] *Hybrid optimization*

Given cost model $q^kT(n-k)$, explain what information about $T(\cdot)$ you need to choose optimal k in practice.

Solution. Need empirical/theoretical scaling of Groebner/XL (degree of regularity) vs dimension and sparsity; memory constraints; parallelism.

8.8 Week 7 Quiz Bank: Hash

[Hash — UG — Comp — 3 pts] *Birthday scaling*

For $n = 40$, estimate the number of random hashes needed for collision probability about 50%.

Solution. About $1.2 \cdot 2^{n/2} \approx 1.2 \cdot 2^{20} \approx 1.26 \times 10^6$.

[Hash — MS — Def — 3 pts] *Formal CR game*

Write the collision resistance game for $H_\lambda : \{0,1\}^* \to \{0,1\}^{n(\lambda)}$.

Solution. Adversary outputs (x, x'); wins if $x \neq x'$ and hashes equal; CR means win prob negligible.

[Hash — PhD — Sec — 5 pts] *Length extension formalism*

State precisely what length extension means for Merkle–Damgård and why HMAC avoids it.

Solution. Given $H(m)$ and $|m|$ compute $H(m\|pad(m)\|m')$ by continuing compression; HMAC rekeys inner/outer so chaining value is not a valid public IV for extension.

9 TA Grading Checklists (Per Week, Per Level)

9.1 How to Grade Fast and Consistently

- Each problem has a **3-part rubric**: (i) correctness, (ii) reasoning/justification, (iii) presentation (definitions/quantifiers).
- Award **partial credit** for correct setup even if arithmetic slips.
- For PhD items, require explicit assumptions (e.g., "assume A full rank mod q," "assume random oracle model," etc.).

9.2 Week 1 Checklist: Factoring

UG

- **Full credit:** Computes factorization/totient/order correctly; uses gcd trick correctly.
- **Partial credit:** Correct method with minor arithmetic errors; correct gcd setup but wrong final gcd.
- **No credit:** No clear method; confuses φ with $N-1$; uses invalid modular steps.

MS

- **Full credit:** Shows $S = p + q = N - \varphi(N) + 1$ and solves quadratic; states polynomial time in $\log N$.
- **Partial credit:** Gets S right but muddles quadratic; missing justification about integer roots.
- **No credit:** Does not connect φ to $p + q$; no recovery method.

PhD

- **Full credit:** Provides full reduction + success probability argument (constant bound); distinguishes failure cases (r odd, $a^{r/2} \equiv -1$).
- **Partial credit:** Correct reduction but weak probability discussion; cites constant success without justification.
- **No credit:** Incorrect reduction; claims always succeeds; ignores failure cases.

9.3 Week 2 Checklist: DLP

UG

- **Full credit:** Correct BSGS/Pollard intuition; understands prime-order reason.
- **Partial credit:** Method correct but arithmetic mistakes; vague explanation but directionally right.
- **No credit:** Confuses DLP with factoring; wrong group order logic.

MS

- **Full credit:** Correct Pohlig–Hellman projection and CRT recombination; clean modular reasoning.
- **Partial credit:** Understands reduction but missing CRT detail or prime-power lifting step.
- **No credit:** No reduction logic; incorrect subgroup projection.

PhD

- **Full credit:** States a correct generic-group lower bound and limitations; constructs valid small-subgroup confinement leakage argument.
- **Partial credit:** Correct concept but missing rigorous statement/assumptions; incomplete leakage derivation.
- **No credit:** Overclaims (e.g., "ECDLP always subexponential"); no explicit attack mechanism.

9.4 Week 3 Checklist: Lattices

UG

- **Full credit:** Correct determinant and basic lattice vector computations; correct SVP/CVP definitions.
- **Partial credit:** Correct formula but arithmetic slip; definitions mostly correct.
- **No credit:** Determinant wrong due to misunderstanding; SVP/CVP swapped or undefined.

MS

- **Full credit:** Correct dual lattice derivation $\mathcal{L}^* = \mathcal{L}(B^{-\top})$ and determinant inversion; correct dual-bias identity for LWE.
- **Partial credit:** Right idea but missing integrality argument; weak explanation of bias.
- **No credit:** Incorrect dual characterization; confuses primal/dual; wrong modular reasoning.

PhD

- **Full credit:** Clear primal embedding sketch (lattice, target, noise); coherent attack-selection reasoning referencing BKZ blocksize/sample count/noise.
- **Partial credit:** Correct high-level but vague on construction; mixes primal/dual conditions.
- **No credit:** No meaningful embedding; incorrect claims ("quantum solves SVP in poly time").

9.5 Week 4 Checklist: Codes

UG

- **Full credit:** Correct syndrome computation and explanation that syndrome depends only on error; computes Prange probability numerically.
- **Partial credit:** Minor $\mathbb{F}_2$ arithmetic errors; probability formula right but numeric slip.
- **No credit:** Treats operations over $\mathbb{Z}$ not $\mathbb{F}_2$; wrong probability model.

MS

- **Full credit:** Derives affine-subspace structure; derives Prange success probability cleanly.
- **Partial credit:** Correct structure but missing dimension argument; probability derivation missing combinatorial count.
- **No credit:** Confuses kernel/image; does not derive probability.

PhD

- **Full credit:** Explains MITM split in Stern/Dumer/BJMM; gives plausible distinguisher statistics and why they help; quantum note is nuanced.
- **Partial credit:** Names algorithms but little mechanism; quantum "halves exponent" stated without caveats.
- **No credit:** No mechanism; incorrect or irrelevant statistics; overclaims.

9.6 Week 5 Checklist: Isogenies

UG

- **Full credit:** Correct definition + graph/path intuition; states degree multiplicativity.
- **Partial credit:** Definition mostly correct but misses morphism/homomorphism requirement.
- **No credit:** Confuses isogeny with isomorphism; no kernel notion.

MS

- **Full credit:** Gives correct proof sketch of degree multiplicativity via function fields; states kernel theorem with conditions.
- **Partial credit:** Idea correct but missing separability/char condition; incomplete proof.
- **No credit:** Incorrect proof; wrong statement (e.g., kernel infinite).

PhD

- **Full credit:** Correctly contrasts path-finding vs hidden shift; articulates modeling caveats beyond slogans.
- **Partial credit:** Correct but superficial; only one caveat.
- **No credit:** Claims "isogeny broken" globally; cannot distinguish protocol-specific from general assumptions.

9.7 Week 6 Checklist: MQ

UG

- **Full credit:** Correct brute-force solving over $\mathbb{F}_2$; explains linearization spurious solutions.
- **Partial credit:** Correct solution set but weak explanation.
- **No credit:** Arithmetic over wrong field; confuses equations with inequalities.

MS

- **Full credit:** Correct matrix form of quadratics in odd characteristic; correct monomial counts/XL conditions.
- **Partial credit:** Mostly correct but misses symmetrization/2 invertible point; count off by small term.
- **No credit:** Wrong monomial counts; matrix representation incorrect.

PhD

- **Full credit:** Explains degree of regularity role; gives coherent MinRank connection and design pitfalls with attack classes.
- **Partial credit:** Right buzzwords but no causal linkage; only one pitfall.
- **No credit:** Overclaims polynomial-time for generic MQ; confuses Groebner with Gaussian elimination.

9.8 Week 7 Checklist: Hash

UG

- **Full credit:** Correct collision vs preimage scaling; correct 50% collision estimate order.
- **Partial credit:** Right scaling but numeric slip.
- **No credit:** Swaps collision and preimage complexities.

MS

- **Full credit:** Correct CR game; correct birthday derivation outline.
- **Partial credit:** Correct game but quantifiers sloppy; derivation missing approximation step.
- **No credit:** Game incorrect (e.g., allows $x = x'$); wrong probability model.

PhD

- **Full credit:** Formal length extension explanation and HMAC mitigation; distinguishes collision vs preimage vs SPR; quantum collision caveat.
- **Partial credit:** Correct but missing formal chaining-value argument; vague on mitigation.
- **No credit:** Claims HMAC still length-extendable; confuses RO model with concrete hashes.