# Lecture Notes on Core Hard Problems in Cryptography

## Contents

# 1 Global Preliminaries

## 1.1 Security parameter, PPT, negligible functions

**Definition 1.1** (Negligible function)**.** *A function $\mu : \mathbb{Z}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is* negligible *if for every $c > 0$ there exists $\lambda_0$ such that for all $\lambda \geq \lambda_0$, $\mu(\lambda) < \lambda^{-c}$. We write $\mu(\lambda) = \mathrm{negl}(\lambda)$.*

**Definition 1.2** (PPT algorithm)**.** *A probabilistic polynomial-time (PPT) algorithm is a randomized algorithm running in time* $\mathrm{poly}(\lambda)$ *on inputs whose sizes are bounded by* $\mathrm{poly}(\lambda)$*.*

## 1.2 Sampling notation and experiments

We write $x \xleftarrow{\$} S$ for uniform sampling from a finite set $S$. For a distribution $\mathcal{D}$, $x \leftarrow \mathcal{D}$ denotes sampling $x$ from $\mathcal{D}$.

## 1.3 Search vs. decision vs. distinguishing problems

- **Search:** output a witness (factor, discrete log, short vector, error vector, isogeny, solution).
- **Decision:** decide whether a witness exists (often NP-complete for codes, MQ; not meaningful for factoring).
- **Distinguishing:** distinguish two distributions (e.g. LWE vs. uniform).

## 1.4 Finite fields, groups, and encodings

All mathematical objects are assumed to have efficient encodings as bitstrings so they are valid algorithm inputs.

# 2 Integer Factorization Problem (IFP)

## 2.1 Formal problem statements

**Definition 2.1** (Integer factorization (search))**.** *Given $N \in \mathbb{Z}_{\geq 2}$, output $d$ with $1 < d < N$ and $d \mid N$ (equivalently output the prime factorization of $N$).*

**Definition 2.2** (RSA modulus distribution)**.** *Let $\mathcal{D}_\lambda$ output $N = pq$ where $p, q$ are independently sampled primes of $\lambda$ bits (often with constraints).*

**Definition 2.3** (Factoring hardness assumption (RSA))**.** *No PPT algorithm outputs a nontrivial factor of $N \leftarrow \mathcal{D}_\lambda$ with non-negligible probability in $\lambda$.*

## 2.2 Why factoring matters: RSA and Rabin

**RSA.** Pick $N = pq$, choose $e$ with $\gcd(e, \varphi(N)) = 1$, let $d = e^{-1} \bmod \varphi(N)$. Encryption: $c = m^e \bmod N$. Decryption: $m = c^d \bmod N$.

**Rabin.** Encryption: $c = m^2 \bmod N$. Decryption requires extracting square roots modulo $N$; security closely relates to factoring.

## 2.3 Core number-theoretic tools

- Modular arithmetic; Euler $\varphi(N)$; Carmichael $\lambda(N)$.
- Order of an element modulo $N$.
- Quadratic residues, Jacobi symbol (for Rabin-type discussions).

## 2.4 Classical attacks (algorithmic overview)

### 2.4.1 Small/special factor methods

**Trial division.** Complexity $\tilde{O}(\sqrt{N})$ worst-case, practical only for small primes.

**Pollard $p-1$.** If $p-1$ is $B$-smooth, choose $M = \mathrm{lcm}(1, \ldots, B)$ and compute $\gcd(a^M - 1, N)$.

**Pollard $\rho$.** Random walk modulo $p$ yields expected time $O(\sqrt{p})$ to find a factor $p$.

**ECM.** Random elliptic curves mod $N$; if group order mod $p$ is smooth, factor emerges. Excellent at finding medium-size factors; often used as a preprocessing step.

### 2.4.2 Sieve methods

**Quadratic Sieve (QS).** Find many relations $x^2 \equiv y^2 \pmod{N}$ via smooth values of $x^2 - N$. Asymptotic: $L_N[1/2, 1]$.

**GNFS.** Asymptotically fastest for general $N$: $L_N\left[1/3, \sqrt[3]{64/9}\right]$. High-level phases: polynomial selection, sieving (relation collection), linear algebra, square root.

## 2.5 Quantum attack: Shor

Shor reduces factoring to order-finding: given $a \bmod N$, find $\mathrm{ord}_N(a)$ via quantum Fourier transform, then derive a factor from $\gcd(a^{r/2} \pm 1, N)$ when $r$ even and $a^{r/2} \not\equiv -1$.

### 2.6 Exercises

1. Prove that knowing $\varphi(N)$ for $N = pq$ allows recovery of $p, q$.
2. Implement Pollard $\rho$ and test on semiprimes with a small factor.
3. Outline why QS produces a congruence of squares.

## 3 Discrete Logarithm Problem (DLP)

### 3.1 Formal definitions

**Definition 3.1** (DLP (search))**.** *Let $G = \langle g \rangle$ be cyclic of order $n$. Given $g$ and $h \in G$, find $x \in \mathbb{Z}_n$ such that $g^x = h$.*

**Definition 3.2** (CDH and DDH (context))**.** *Given $(g, g^a, g^b)$, the* Computational Diffie–Hellman (CDH) *problem is to compute $g^{ab}$. The* Decisional Diffie–Hellman (DDH) *problem is to distinguish $(g, g^a, g^b, g^{ab})$ from $(g, g^a, g^b, g^c)$.*

### 3.2 Generic group algorithms

**Baby-step/giant-step.** Time and memory $\tilde{O}(\sqrt{n})$.

**Pollard $\rho$.** Expected $\tilde{O}(\sqrt{n})$ time, small memory.

**Pohlig–Hellman.** If $n = \prod_i p_i^{e_i}$, reduce DLP to DLPs modulo each $p_i^{e_i}$, then CRT. Hence cryptographic groups choose $n$ having a large prime factor.

### 3.3 Finite-field DLP: index calculus

In $\mathbb{F}_p^\times$ and extension fields, index calculus expresses elements via a factor base, collects relations, solves a linear system, then computes logs of targets. The best-known for prime fields is NFS-DL with $L_p\left[1/3, \sqrt[3]{64/9}\right]$.

### 3.4 Elliptic-curve DLP (ECDLP)

For generic elliptic curves, no subexponential index calculus is known in full generality; best-known attacks are generic $\tilde{O}(\sqrt{n})$.

**MOV / Frey–Rück.** Certain special curves admit embedding of ECDLP into finite-field DLP via pairings; avoided in standard curve selection.

### 3.5 Quantum

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

### 3.6 Exercises

1. Prove correctness of Pohlig–Hellman using CRT.
2. Show Pollard $\rho$ is a collision-finding method on a pseudorandom walk.

# 4 Lattices: SVP/CVP, SIS, LWE

## 4.1 Lattices and basic geometry

**Definition 4.1** (Lattice). *A full-rank lattice $\mathcal{L} \subset \mathbb{R}^d$ is $\mathcal{L}(B) = \{Bz : z \in \mathbb{Z}^d\}$ for an invertible $B \in \mathbb{R}^{d \times d}$.*

**Definition 4.2** (Determinant). $\det(\mathcal{L}) := |\det(B)|$, *independent of the basis $B$.*

**Definition 4.3** (Successive minima). $\lambda_1(\mathcal{L}) := \min\{\|v\|_2 : v \in \mathcal{L} \setminus \{0\}\}$.

## 4.2 SVP and CVP

**Definition 4.4** (SVP and $\gamma$-SVP). *Given a basis $B$ of $\mathcal{L}$, find $v \in \mathcal{L} \setminus \{0\}$ minimizing $\|v\|_2$ (SVP), or find $v$ with $\|v\|_2 \le \gamma\lambda_1(\mathcal{L})$ ($\gamma$-SVP).*

**Definition 4.5** (CVP and $\gamma$-CVP). *Given basis $B$ and target $t \in \mathbb{R}^d$, find $v \in \mathcal{L}$ minimizing $\|t - v\|_2$ (CVP), or within factor $\gamma$ ($\gamma$-CVP).*

## 4.3 Algorithmic toolkit: reduction + enumeration + sieving

**LLL.** Polynomial-time basis reduction giving approximation factors exponential in $d$.

**BKZ.** Blockwise reduction parameterized by blocksize $\beta$; the main engine in practical lattice cryptanalysis.

**Enumeration.** Searches for short vectors given a reduced basis; exponential in $d$ with strong dependence on reduction quality.

**Sieving.** Heuristic nearest-neighbor style methods to find very short vectors; typically exponential time and memory.

## 4.4 SIS and LWE (cryptographic problems)

**Definition 4.6** (SIS (short integer solution)). *Fix $n, m, q$ and a norm bound $\beta$. Given $A \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, find a nonzero $x \in \mathbb{Z}^m$ such that*

$$Ax \equiv 0 \pmod{q} \quad \text{and} \quad \|x\| \le \beta.$$

**Definition 4.7** (LWE (decision form)). *Fix $n, q$ and error distribution $\chi$ on $\mathbb{Z}_q$. Given samples $(a_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, distinguish:*

$$b_i = \langle a_i, s\rangle + e_i \ (\mathrm{mod} q), \quad s \xleftarrow{\$} \mathbb{Z}_q^n, \ e_i \xleftarrow{\$} \chi$$

*from uniform $(a_i, b_i)$.*

## 4.5 Attack taxonomy for LWE (how cryptanalysts think)

**Primal attacks.** Embed the LWE instance into a lattice so that the secret/error corresponds to a short or close vector; solve via BKZ+enumeration/sieving.

**Dual attacks.** Find a short dual vector $y$ such that $y^T b$ reveals bias distinguishing LWE from uniform.

**Hybrid attacks.** Guess part of the secret to reduce dimension, then primal/dual.

**Combinatorial (BKW).** Trade dimension for noise growth via sample combining; effective in some parameter regimes.

### 4.6 Quantum

No known polynomial-time quantum algorithm for worst-case lattice problems; quantum can accelerate some search/sieving subroutines.

### 4.7 Exercises

1. Prove $\det(\mathcal{L})$ is basis-independent.
2. For a 2D lattice, compute $\lambda_1$ by inspection and compare with LLL output.
3. Derive the dual lattice $\mathcal{L}^* = \{y \in \mathbb{R}^d : \langle y, x\rangle \in \mathbb{Z}\ \forall x \in \mathcal{L}\}$ and basic properties.

## 5 Codes: Syndrome Decoding and Related Problems

### 5.1 Linear codes

**Definition 5.1** (Linear code)**.** *A linear $[n,k]_q$ code is a $k$-dimensional $\mathbb{F}_q$-subspace $C \subseteq \mathbb{F}_q^n$. A parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$ satisfies $C = \{c \in \mathbb{F}_q^n : Hc^\top = 0\}$.*

**Definition 5.2** (Hamming weight and distance)**.** *For $x \in \mathbb{F}_q^n$, $w_H(x) = |\{i : x_i \neq 0\}|$ and $d_H(x,y) = w_H(x-y)$.*

### 5.2 Hard problems

**Definition 5.3** (Syndrome Decoding (SD))**.** *Given $(H, s, t)$ with $H \in \mathbb{F}_q^{(n-k)\times n}$, $s \in \mathbb{F}_q^{n-k}$, and integer $t$, find $e \in \mathbb{F}_q^n$ such that*

$$He^\top = s, \qquad w_H(e) \leq t.$$

**Definition 5.4** (Minimum Distance Problem (MDP))**.** *Given a linear code $C$ and integer $t$, decide whether there exists $c \in C \setminus \{0\}$ with $w_H(c) \leq t$.*

### 5.3 McEliece-type cryptography (context)

Public key is a disguised generator matrix of a structured code (e.g. Goppa), chosen so legitimate decoding is efficient while the public code looks random.

### 5.4 Attacks: Information Set Decoding (ISD)

**Prange.** Guess an information set $I$ of size $k$ that avoids error positions; attempt solve. Time $\approx \binom{n}{t}/\binom{n-k}{t}$ (heuristic).

**Stern/Dumer/BJMM.** Use meet-in-the-middle and combinatorial improvements; modern best practical methods are ISD variants.

**Structural attacks.** If the public code is distinguishable from random, recover structure and decode efficiently.

### 5.5 Quantum

Grover can speed up the guessing layers in ISD (model-dependent exponent reductions).

### 5.6 Exercises

1. Derive the Prange success probability for an information set.
2. Show how SD can be written as finding a low-weight vector in an affine space.

## 6 Isogenies of Elliptic Curves (Supersingular/CSIDH-style)

### 6.1 Elliptic curves and isogenies

**Definition 6.1** (Elliptic curve over $\mathbb{F}_q$)**.** *(Characteristic $\neq 2, 3$.) $E/\mathbb{F}_q$ given by $y^2 = x^3 + ax + b$ with discriminant $\Delta \neq 0$. The set $E(\mathbb{F}_q)$ forms a finite abelian group.*

**Definition 6.2** (Isogeny)**.** *An isogeny $\varphi : E_1 \to E_2$ over $\mathbb{F}_q$ is a nonconstant morphism defined over $\mathbb{F}_q$ that is also a group homomorphism. Its kernel is finite and* $\deg(\varphi)$ *is its morphism degree.*

### 6.2 Canonical computational problems

**Definition 6.3** (Supersingular isogeny (search))**.** *Work over $\mathbb{F}_{p^2}$. Given supersingular $E, E'/\mathbb{F}_{p^2}$, find an explicit isogeny $\varphi : E \to E'$ of prescribed smooth degree (often $\ell^r$), represented so $\varphi$ can be evaluated.*

**Definition 6.4** (Class group action / CSIDH-type problem (high level))**.** *Given a commutative group action of an ideal class group on a set of curves, the hard problem is to recover the acting group element (ideal class) mapping $E$ to $E'$ (hidden shift-like).*

### 6.3 Attacks (high-level)

**Graph/path-finding.** Model the supersingular isogeny graph: vertices are curves, edges are $\ell$-isogenies. Generic attacks are essentially path-finding with meet-in-the-middle.

**Meet-in-the-middle (Delfs–Galbraith style).** Bidirectional search reduces naive exponent (heuristically $\tilde{O}(p^{1/4})$ for some variants).

**Protocol-specific cryptanalysis.** Some isogeny key exchanges (notably SIDH/SIKE) were broken by exploiting additional structure. This is not a break of "all isogeny problems" but invalidates those particular schemes/assumptions.

### 6.4 Quantum

In commutative settings (hidden shift–like), Kuperberg-type algorithms give subexponential time. No known general polynomial-time algorithm for supersingular path-finding.

### 6.5 Exercises

1. Prove $\deg(\varphi \circ \psi) = \deg(\varphi)\deg(\psi)$.
2. Show that kernels of separable isogenies correspond to finite subgroups.

# 7 Multivariate Cryptography: MQ and Structured Variants

## 7.1 MQ problem

**Definition 7.1** (MQ (Multivariate Quadratic) search)**.** *Let $\mathbb{F}_q$ be a finite field and $f_1, \ldots, f_m \in \mathbb{F}_q[x_1, \ldots, x_n]$ with $\deg(f_i) \leq 2$. Find $x \in \mathbb{F}_q^n$ such that*

$$f_i(x) = 0 \quad \forall i \in \{1, \ldots, m\}.$$

## 7.2 Why MQ appears in signatures

Many multivariate signature schemes publish a public map $P : \mathbb{F}_q^n \to \mathbb{F}_q^m$ that is quadratic and designed to be easy to invert with a trapdoor (e.g. Oil-and-Vinegar families) but hard to invert without it.

## 7.3 Algebraic attacks

**Gröbner bases (F4/F5).** Compute a Gröbner basis for the ideal $\langle f_1, \ldots, f_m \rangle$. Complexity depends on the *degree of regularity* and sparsity; often dominates best-known attacks.

**XL and relinearization.** Multiply equations by monomials to increase degree, then linearize monomials as variables.

**Hybrid.** Guess some variables to reduce $n$, then apply Gröbner/XL.

**Rank/MinRank and structural attacks.** Many structured MQ schemes admit reductions to MinRank or exploit low-rank structure (Kipnis–Shamir-type ideas in some settings).

## 7.4 Quantum

Grover can accelerate the guessing part in hybrid methods; core algebraic steps see limited generic quantum speedups.

## 7.5 Exercises

1. Write a small MQ instance over $\mathbb{F}_2$ and solve by brute force; compare with linearization.
2. Show how a quadratic form corresponds to a symmetric matrix (in odd characteristic).

# 8 Hash Functions: Definitions, Bounds, and Attacks

## 8.1 Formal security notions

**Definition 8.1** (Hash family)**.** *A hash family $\{H_\lambda\}$ is a collection of efficient functions $H_\lambda : \{0,1\}^* \to \{0,1\}^{n(\lambda)}$.*

**Definition 8.2** (Collision resistance (CR))**.** *$\{H_\lambda\}$ is collision resistant if for every PPT adversary $\mathcal{A}$,*

$$\mathbb{P}\left[(x, x') \leftarrow \mathcal{A}(1^\lambda) : x \neq x' \ \wedge \ H_\lambda(x) = H_\lambda(x')\right] = \mathrm{negl}(\lambda).$$

**Definition 8.3** (Second-preimage resistance (SPR))**.** *For every PPT $\mathcal{A}$ and a specified input distribution $\mathcal{D}$,*

$$\mathbb{P}\left[x \leftarrow \mathcal{D};\ x' \leftarrow \mathcal{A}(1^\lambda, x) : x' \neq x \wedge H_\lambda(x') = H_\lambda(x)\right] = \mathrm{negl}(\lambda).$$

**Definition 8.4** (Preimage resistance (OW))**.** *For every PPT $\mathcal{A}$,*

$$\mathbb{P}\left[y \xleftarrow{\$} \{0,1\}^{n(\lambda)};\ x \leftarrow \mathcal{A}(1^\lambda, y) : H_\lambda(x) = y\right] = \mathrm{negl}(\lambda).$$

### 8.2 Generic bounds (information-theoretic / black-box)

**Birthday bound.** For random $H : \{0,1\}^* \to \{0,1\}^n$, a collision appears after about $2^{n/2}$ queries.

**Preimages.** Expected $2^n$ queries classically for random preimage search.

### 8.3 Design-level attacks (structural)

**Differential cryptanalysis.** Track differences through compression functions to produce collisions faster than birthday (design-specific).

**Chosen-prefix collisions.** Construct collisions for two chosen distinct prefixes; crucial for real-world certificate forgeries in weakened hashes.

**Length extension.** Merkle–Damgård hashes satisfy $H(m\|pad(m)\|m')$ computable from $H(m)$ and $|m|$. Thus MAC constructions of the form $H(k\|m)$ are insecure; HMAC avoids this.

### 8.4 Quantum

Grover gives preimages in about $2^{n/2}$ quantum queries. (Quantum collision-finding can beat $2^{n/2}$ in some models; details depend on the oracle model.)

### 8.5 Exercises

1. Prove the birthday bound estimate for collisions.
2. Show length extension for Merkle–Damgård at the level of the iteration structure.

## 9 Recommended Reading (non-exhaustive)

- J. Silverman, *A Friendly Introduction to Number Theory* (background).
- H. Cohen, *A Course in Computational Algebraic Number Theory* (factoring and NFS context).
- D. Boneh and V. Shoup, *A Graduate Course in Applied Cryptography* (DLP, RSA, hash, etc.).
- O. Goldreich, *Foundations of Cryptography* (formal security notions).
- D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems* / lattice-crypto surveys.
- R. Lidl and H. Niederreiter, *Finite Fields* (finite field DLP context).
- F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.*
- L. C. Washington, *Elliptic Curves: Number Theory and Cryptography.*
- Cox, Little, O'Shea, *Ideals, Varieties, and Algorithms* (Gröbner basics).