Mathematical Foundations

CS 276: Introduction to Cryptography

Sanjam Garg

January 20, 2026

Overview

Why Mathematical Foundations?

- Modern cryptography requires **formal definitions** of security
- Two key assumptions:
    1. Attackers run in **polynomial time** (not unreasonably large)
    2. Security can be broken with **very small** (negligible) probability
- Without these: must use information-theoretic cryptography
    - Example: One-time pad requires keys as long as messages
    - Often impractical for real applications

Probabilistic Polynomial Time (PPT)

Definition

A probabilistic Turing Machine M is **PPT** if $\exists c \in \mathbb{Z}^+$ such that $\forall x \in \{0,1\}^*$, $M(x)$ halts in $|x|^c$ steps.

- Probabilistic = can make random choices during execution
- Polynomial time = runtime bounded by polynomial in input length
- Any deterministic polynomial-time algorithm is trivially PPT
- Probabilistic algorithms can sometimes be more efficient

Non-uniform PPT

Definition

A **non-uniform PPT** machine is a sequence $\{M_1, M_2, \cdots\}$ such that $\exists c \in \mathbb{Z}^+$ where $\forall x \in \{0,1\}^*$, $|M_{|x|}|$ is of size $\leq |x|^c$ and $M_{|x|}(x)$ halts in $|x|^c$ steps.

- Different machine for each input length
- Each machine can have "advice" specific to that length
- Models adversaries with precomputed information
- Stronger model than uniform PPT
- More convenient for security reductions

Characterizing Probabilities

We need to formalize what "very small" means:

- **Noticeable**: Larger than some inverse-polynomial
- **Negligible**: Smaller than any inverse-polynomial

These concepts are crucial for defining security:

- Security can fail with **negligible** probability
- Attacks must succeed with **non-negligible** probability

Noticeable Functions

Definition

A function $\mu : \mathbb{Z}^+ \to [0, 1]$ is **noticeable** if

$$\exists c \in \mathbb{Z}^+, n_0 \in \mathbb{Z}^+ \text{ such that } \forall n \geq n_0, \ \mu(n) > n^{-c}$$

Noticeable Functions

Definition

A function $\mu : \mathbb{Z}^+ \to [0,1]$ is **noticeable** if

$$\exists c \in \mathbb{Z}^+, n_0 \in \mathbb{Z}^+ \text{ such that } \forall n \geq n_0,\ \mu(n) > n^{-c}$$

Example 1

$\mu(n) = n^{-3}$ is noticeable.

- Satisfied for $c = 4$ and $n_0 = 1$
- For all $n \geq 1$: $n^{-3} > n^{-4}$

Negligible Functions

Definition

A function $\mu : \mathbb{Z}^+ \to [0,1]$ is **negligible** if

$$\forall c \in \mathbb{Z}^+ \; \exists n_0 \in \mathbb{Z}^+ \text{ such that } \forall n \geq n_0,\ \mu(n) < n^{-c}$$

Negligible Functions

Definition

A function $\mu : \mathbb{Z}^+ \to [0, 1]$ is **negligible** if

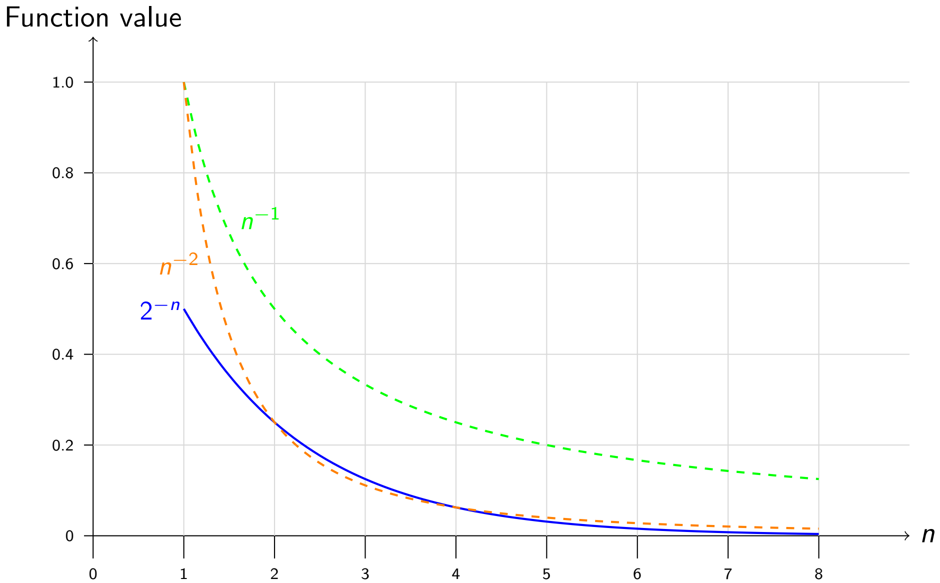
$$\forall c \in \mathbb{Z}^+ \; \exists n_0 \in \mathbb{Z}^+ \text{ such that } \forall n \geq n_0, \; \mu(n) < n^{-c}$$

Example 2

$\mu(n) = 2^{-n}$ is negligible.

- For any c, eventually $2^{-n} < n^{-c}$
- Exponential decay beats any polynomial bound

Visualizing Negligible vs Noticeable



The Gap: Neither Negligible Nor Noticeable

Key Insight

A function that is **not negligible** is **not necessarily noticeable**!

The Gap: Neither Negligible Nor Noticeable

Key Insight

A function that is **not negligible** is **not necessarily noticeable**!

Example 3

$$\mu(n) = \begin{cases} 2^{-n} & \text{if } n \bmod 2 = 0 \\ n^{-3} & \text{if } n \bmod 2 \neq 0 \end{cases}$$

- Not negligible: $\mu(n) = n^{-3} > n^{-4}$ for odd n
- Not noticeable: $\mu(n) = 2^{-n} < n^{-c}$ for large even n
- Function oscillates between the two behaviors

Properties of Negligible Functions

Theorem 4

If $\mu(n)$ and $\nu(n)$ are negligible, then:

1. $\mu(n) + \nu(n)$ *is negligible*
2. $\mu(n) \cdot \nu(n)$ *is negligible*
3. $\text{poly}(\mu(n))$ *is negligible (for any polynomial)*

Properties of Negligible Functions

Theorem 4

If $\mu(n)$ and $\nu(n)$ are negligible, then:

1. $\mu(n) + \nu(n)$ *is negligible*
2. $\mu(n) \cdot \nu(n)$ *is negligible*
3. $\text{poly}(\mu(n))$ *is negligible (for any polynomial)*

Proof idea.

- For sum: Given c, find n_0 where both $\mu(n), \nu(n) < n^{-c-1}$
- Then $\mu(n) + \nu(n) < 2n^{-c-1} \leq n^{-c}$ for $n \geq 2$
- Similar arguments for product and polynomial

$\square$

Practical Intuition

- **Negligible probability** $=$ essentially impossible in practice
- Example: $\mu(n) = 2^{-128}$
  - Smaller than probability of being struck by lightning twice ($\approx 10^{-12}$)
  - For $n = 64$: $2^{-64} \approx 5 \times 10^{-20}$ (astronomically small)
- We can safely ignore negligible probabilities in security analysis
- This is why cryptographic schemes are considered "secure" even with negligible failure probability

Why Hardness Assumptions?

- We need **concrete problems** that are believed to be hard
- These form the foundation for practical cryptography
- Three main families:
  1. Discrete-Log family
  2. Factoring
  3. Lattice problems (covered later)
- Studied for decades, widely believed to be hard
- Proving hardness would resolve major open problems (e.g., $P \neq NP$)

Group Ensembles

Definition

A **group ensemble** is a set of finite cyclic groups $\mathcal{G} = \{\mathbb{G}_n\}_{n \in \mathbb{N}}$ where:

- Group operations computable in polynomial time (in n)
- Generator g of $\mathbb{G}_n$ computable in polynomial time

Example 5

$\mathbb{Z}_p^*$ for prime p (multiplicative group modulo p)

Discrete-Log Problem

Setup

- Group $\mathbb{G}$ of prime order
- Generator g of $\mathbb{G}$
- Given g^x for random x, find x

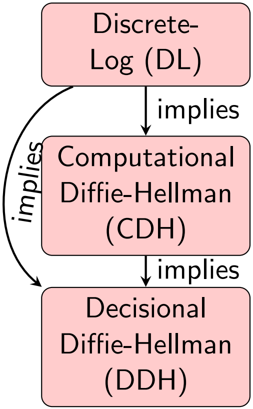
Discrete-Log Assumption

For group ensemble $\mathcal{G} = \{\mathbb{G}_n\}_{n \in \mathbb{N}}$, for every non-uniform PPT algorithm $\mathcal{A}$:

$$\mu_{\mathcal{A}}(n) := \Pr_{x \leftarrow |G_n|}[\mathcal{A}(g, g^x) = x]$$

is negligible.

Diffie-Hellman Problems



- DL is the **weakest** assumption
- Breaking DL $\Rightarrow$ breaking CDH $\Rightarrow$ breaking DDH
- Reverse implications not known to hold

Computational Diffie-Hellman (CDH)

Problem

Given g, g^x, g^y for random x, y, compute g^{xy}

CDH Assumption

For group ensemble $\mathcal{G}$, for every non-uniform PPT algorithm $\mathcal{A}$:

$$\mu_{\mathcal{A}}(n) := \Pr_{x,y \leftarrow |G_n|}[\mathcal{A}(g, g^x, g^y) = g^{xy}]$$

is negligible.

- Clearly implied by DL assumption
- If you can compute discrete logs, you can compute g^{xy}

Decisional Diffie-Hellman (DDH)

Problem

Distinguish between:

- (g, g^x, g^y, g^{xy}) for random x, y
- (g, g^x, g^y, g^z) for random x, y, z

DDH Assumption

For group ensemble $\mathcal{G}$, for every non-uniform PPT algorithm $\mathcal{A}$:

$$\mu_{\mathcal{A}}(n) = |\Pr[\mathcal{A}(g, g^x, g^y, g^{xy}) = 1] - \Pr[\mathcal{A}(g, g^x, g^y, g^z) = 1]|$$

is negligible.

Groups for Diffie-Hellman

1. $\mathbb{Z}_p^*$ (multiplicative group mod prime p)
   - CDH assumption believed to hold
   - DDH assumption does NOT hold!
   - Can use Legendre symbol to distinguish

Groups for Diffie-Hellman

1. $\mathbb{Z}_p^*$ (multiplicative group mod prime p)
    - CDH assumption believed to hold
    - DDH assumption does NOT hold!
    - Can use Legendre symbol to distinguish
2. **Order-q subgroup of** $\mathbb{Z}_p^*$ where $p = 2q + 1$ (both prime)
    - DDH assumption believed to hold

Groups for Diffie-Hellman

1. $\mathbb{Z}_p^*$ (multiplicative group mod prime p)
   - CDH assumption believed to hold
   - DDH assumption does NOT hold!
   - Can use Legendre symbol to distinguish
2. **Order-q subgroup of** $\mathbb{Z}_p^*$ where $p = 2q + 1$ (both prime)
   - DDH assumption believed to hold
3. $\mathbb{QR}_N$ (quadratic residues mod $N = pq$)
   - DDH assumption believed to hold
   - Special property: CDH $\Rightarrow$ Factoring (we'll prove this!)

Groups for Diffie-Hellman

1. $\mathbb{Z}_p^*$ (multiplicative group mod prime p)
   - CDH assumption believed to hold
   - DDH assumption does NOT hold!
   - Can use Legendre symbol to distinguish
2. **Order-q subgroup of** $\mathbb{Z}_p^*$ where $p = 2q + 1$ (both prime)
   - DDH assumption believed to hold
3. $\mathbb{QR}_N$ (quadratic residues mod $N = pq$)
   - DDH assumption believed to hold
   - Special property: CDH $\Rightarrow$ Factoring (we'll prove this!)
4. **Elliptic Curve groups**
   - DL, CDH, and DDH assumptions all believed to hold
   - Examples: secp256k1 (Bitcoin/Ethereum), Curve25519 (modern protocols)

Key Result: CDH $\Rightarrow$ Factoring

Theorem 6

If CDH is hard in $\mathbb{QR}_N$ (where $N = pq$), then factoring N is hard.

Reduction Strategy

Given a CDH solver $\mathcal{A}$, construct a factoring algorithm $\mathcal{B}$:

1. Use $\mathcal{A}$ to find square roots
2. Use square roots to factor N (with probability $\frac{1}{2}$)

The Reduction: Setup

Algorithm $\mathcal{B}$ on input N:

1. Sample random $r \in \mathbb{Z}_N^*$, compute $v := r^2 \bmod N$ (so $v \in \mathbb{QR}_N$)
2. Compute $g := v^2 \bmod N$ (generator of $\mathbb{QR}_N$ with high probability)
3. Sample $x, y \leftarrow [N]$ uniformly
   - Statistically close to sampling from $[\phi(N)]$ since $|N - \phi(N)| = O(\sqrt{N})$
4. Let $u := \mathcal{A}(g, g^x \cdot v, g^y \cdot v)$
5. Compute $w := \frac{u}{g^{xy} \cdot v^{x+y}}$
6. If $w^2 \equiv v \pmod{N}$ and $w \neq \pm r$, then:
   - Factor N using $\gcd(N, w - r)$ or $\gcd(N, w + r)$
7. Otherwise, output $\perp$

Why This Works

Key Observation

If $\mathcal{A}$ correctly computes CDH, then:

$$u = g^{(x+2^{-1})(y+2^{-1})} = v^{2xy+x+y+2^{-1}}$$

Therefore:

$$w = \frac{u}{g^{xy} \cdot v^{x+y}} = \frac{v^{2xy+x+y+2^{-1}}}{v^{2xy+x+y}} = v^{2^{-1}}$$

So w is a square root of v: $w^2 = (v^{2^{-1}})^2 = v$

Completing the Factorization

Square Roots in $\mathbb{Z}_N^*$

In $\mathbb{Z}_N^*$ where $N = pq$, each quadratic residue has exactly **4 square roots**:

- Two are $\pm r$ (the "easy" roots we already know)
- Two are $\pm w$ where $w \neq \pm r$ (computed via CDH oracle)

Factoring from Square Roots

If $w^2 \equiv v \pmod{N}$ but $w \neq \pm r \pmod{N}$, then:

$$w^2 - r^2 = (w - r)(w + r) \equiv 0 \pmod{N}$$

Since $N = pq$ and neither factor is 0 mod N:

$$\gcd(N, w - r) \text{ and } \gcd(N, w + r) \text{ give the factors}$$

With probability $\frac{1}{2}$, we get a useful square root ($w \neq \pm r$)!

Key Takeaways

1. **PPT adversaries**: Polynomial-time attackers with randomness
2. **Negligible functions**: Smaller than any inverse-polynomial
   - Essential for formalizing "very small" failure probabilities
3. **Discrete-Log family**: DL $\Rightarrow$ CDH $\Rightarrow$ DDH
   - Foundation for many cryptographic schemes
4. **CDH** $\Rightarrow$ **Factoring** in $\mathbb{QR}_N$
   - Shows connection between different hardness assumptions

Next Steps

- These foundations enable formal security definitions
- We'll use these concepts throughout the course
- Next: Applying these to construct cryptographic primitives
- Questions?