Foundations of Cryptography (CS60088)

Spring 2026

Tutorial 5: Number Theory and Public-Key Encryption

1. Let $N \in \mathbb{N}$ and suppose $ab \equiv c \pmod{N}$ with $\gcd(b, N) = d$. Prove the following:
   (a) $d \mid c$.
   (b) $a \cdot (b/d) \equiv (c/d) \pmod{N/d}$.
   (c) $\gcd(b/d, N/d) = 1$.

2. Let G be a cyclic group of prime order q with generator g.
   (a) Show that for any $h \in G$, h is a generator of G if and only if $h = g^k$ for some k with $\gcd(k, q) = 1$.
   (b) Give an efficient algorithm to test whether a given element $h \in G$ is a generator.
   (c) Describe how to compute g^x efficiently for large x.
   (d) Show how to compute the inverse of an element $g^x \in G$ efficiently.

3. Let G be a cyclic group.
   (a) Prove that if the Computational Diffie-Hellman (CDH) problem is hard relative to G, then the discrete logarithm (DL) problem is also hard relative to G.
   (b) Prove that if the Decisional Diffie-Hellman (DDH) problem is hard relative to G, then the CDH problem is also hard relative to G.

4. Assume a public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Show that for any public key pk and ciphertext c generated as
$$c \leftarrow \mathsf{Enc}_{pk}(m),$$
an unbounded adversary can recover m with probability 1. In other words, prove that perfectly-secret public-key encryption is impossible.

5. Consider the following public-key encryption scheme. The public key is (G, q, g, h) and the private key is x, where $h = g^x$.
   - To encrypt a bit b:
     - If $b = 0$, choose $y \leftarrow \mathbb{Z}_q$ uniformly and compute $c_1 = g^y$, $c_2 = h^y$. Output (c_1, c_2).
     - If $b = 1$, choose $y, z \leftarrow \mathbb{Z}_q$ independently and compute $c_1 = g^y$, $c_2 = g^z$. Output (c_1, c_2).

   (a) Show that decryption can be performed efficiently given x.

(b) Prove that this scheme is CPA-secure assuming the hardness of the Decisional Diffie-Hellman (DDH) problem in G.

6. Prove formally that the ElGamal encryption scheme is not secure against chosen-ciphertext attacks (CCA). Construct an explicit adversary that breaks CCA security.