Cryptography ENEE/CMSC/MATH 456: Homework 5

Due by 2pm on 4/24/2024.

1. Prove formally that the hardness of the CDH problem relative to G implies the hardness of the discrete logarithm problem relative to G.

2. Determine the points on the elliptic curve $E : y^2 = x^3 + 2x + 1$ over Z_{11}. How many points are on this curve?

3. Can the following problem be solved in polynomial time? Given a prime p, a value $x \in Z^*_{p-1}$ and $y := g^x \mod p$ (where g is a uniform value in Z^*_p), find g, i.e., compute $y^{1/x} \mod p$. If your answer is "yes," give a polynomial-time algorithm. If your answer is "no," show a reduction to one of the assumptions introduced in this chapter.

4. Describe in detail a man-in-the-middle attack on the Diffie-Hellman key-exchange protocol whereby the adversary ends up sharing a key k_A with Alice and a different key k_B with Bob, and Alice and Bob cannot detect that anything has gone wrong.

   What happens if Alice and Bob try to detect the presence of a man-in-the-middle adversary by sending each other (encrypted) questions that only the other party would know how to answer?

5. Consider the subgroup of Z^*_{23} consisting of quadratic residues modulo 23. This group consists of the following elements: $\{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$. We choose $g = 2$ to be the generator of the subgroup. Let $(23, 11, 2, x = 4)$ be the secret key for ElGamal. Find the corresponding public key. Then encrypt the message $m = 3$, using randomness $r = 6$, obtaining some ciphertext c. Decrypt c to recover m. Do the computations by hand and show your work.

   Hint: To speed up your computations, use the fact that $7^3 \equiv -2 \mod 23$, $4^{-1} = 6 \mod 23$.

6. Consider the following key-exchange protocol:

   Common input: The security parameter 1^n.
   (a) Alice runs $\mathcal{G}(1^n)$ to obtain (G, q, g).
   (b) Alice chooses $x_1, x_2 \leftarrow Z_q$ and sends $\alpha = x_1 + x_2$ to Bob.
   (c) Bob chooses $x_3 \leftarrow Z_q$ and sends $h_2 = g^{x_3}$ to Alice.
   (d) Alice sends $h_3 = g^{x_2 \cdot x_3}$ to Bob.
   (e) Alice outputs $h_2^{x_1}$. Bob outputs $(g^\alpha)^{x_3} \cdot (h_3)^{-1}$.

   Show that Alice and Bob output the same key. Analyze the security of the scheme (i.e., either prove its security or show a concrete attack).

7. Show that any 2-round key-exchange protocol (that is, where each party sends a single message) can be converted into a CPA-secure public-key encryption scheme.

8. Consider the following variant of El Gamal encryption. Let $p = 2q + 1$, let G be the group of squares modulo p, and let g be a generator of G. The private key is (G, g, q, x) and the public key is $G, g, q, h)$, where $h = g^x$ and $x \in Z_q$ is chosen uniformly. To encrypt a message $m \in Z_q$, choose a uniform $r \in Z_q$, compute $c_1 := g^r modp$ and $c_2 := h^r + mmodp$, and let the ciphertext be $\langle c_1, c_2 \rangle$. Is this scheme CPA-secure? Prove your answer.

9. Recall that the DDH assumption is false in the group $\mathbb{G} := \mathbb{Z}_p^*$, of order $q = p - 1$, where p is prime. This is due to the fact that the Legendre symbol allows one to check whether or not $x \in \mathbb{Z}_p^*$ is a quadratic residue–i.e. a perfect square.

   (a) What information is leaked about the message $m \in \mathbb{G}$ when ElGamal encryption is instantiated with the group $\mathbb{G} := \mathbb{Z}_p^*$? Explain your answer.
   Hint: Consider using the Legendre symbol to compute whether $h = g^x$, g^y, and $h^y \cdot m$ are quadratic residues. What can be deduced about m? When is this information leaked?

   (b) Why does this problem go away when we instantiate El Gamal Encryption with the group $\mathbb{G}'$ of order q' that contains only the quadratic residues in $\mathbb{G}$ where $p = 2q' + 1$ is a strong prime?

10. Assume the Schnorr identification scheme is run in the group Z_p^*, where p is a sufficiently large prime. Recall that in this case, one can efficiently compute the Legendre symbol of $y = g^x, g^k$. Explain how a verifier can use this information to cause the distribution of s to not be uniform random. In particular, if x is odd, the verifier can cause s to always be even. Explain why this would mean that the simulation strategy we gave in class for Schnorr's algorithm would fail.