Homework 5

Posted: Wednesday, November 8, 2023 – 11:59pm
Due: Monday, November 20, 2023 – 11:59pm

Task 1 – Group Generators (8 points)

a) **[2 points]** Let $m \geq 2$ be an integer, and $a \in \mathbb{Z}_m^*$. Show that for every $b \in \mathbb{Z}_m$, there always exists a solution to the modular equation $ax \equiv b \pmod{m}$.

b) **[2 points]** Let $m \geq 2$ be an integer, and $a \in \mathbb{Z}_m \setminus \mathbb{Z}_m^*$. Show that there exists $b \in \mathbb{Z}_m$ such that the modular equation $ax \equiv b \pmod{m}$ has *no solution*.

c) **[2 points]** Let $\mathbb{G} = \langle g \rangle$ be a cyclic group with generator g and order $|\mathbb{G}| = m \geq 2$. Show that there are exactly $\varphi(m)$ generators in $\mathbb{G}$.

d) **[2 points]** Which elements of $\mathbb{G} = \langle g \rangle$ are generators if the group order m is additionally prime?

Task 2 – Group Order Factorization & Discrete Logarithms (15 points)

Let $\mathbb{G} = \langle g \rangle$ be a cyclic group with generator g and order $|\mathbb{G}| = m$. Further, let $n \geq 1$ be an integer that divides m, i.e., $m = \alpha \cdot n$ for some integer $\alpha \geq 1$.

a) **[2 points]** Show that $\langle g^\alpha \rangle = \{1, g^\alpha, g^{2\alpha}, \ldots\}$ has order n.

b) **[2 points]** Let $X = g^x \in \mathbb{G}$. What is the discrete logarithm of X^α to the base g^α? (Note that this has to be a number between 0 and $n-1$.)

c) **[5 points]** Show that the DDH assumption cannot be true in $\mathbb{Z}_p^*$ for a prime p by giving an efficient distinguisher running in time polynomial in $\log p$.

d) **[6 points]** Assume that $m = p_1 p_2 \cdots p_{44}$, where $p_1 = 2$, $p_2 = 3$, ..., $p_{44} = 193$ are the 44 smallest primes. Note that $m \approx 2^{256}$.

Show how in this case the discrete logarithm x of $X = g^x$ can be recovered with a relatively small number of group operations. Your solution should use a few thousand group exponentiations.

Hint: You can use the so-called Chinese Remainder Theorem: If for some $x \in \mathbb{Z}_m$ and $m = p_1 p_2 \cdots p_k$, where $p_1, \ldots, p_k$ are distinct primes, we know that $x \bmod p_i = x_i$ for all $i = 1, \ldots, k$, then we can efficiently reconstruct x from $x_1, \ldots, x_k$. (You do not need to consider the complexity of this reconstruction, or how this is done, as part of your response. Just invoke this fact as a black box.)

Task 3 – A Little Number Theory (5 points)

Assume that you are given an algorithm which on input $N = PQ$, for distinct unknown primes P and Q, outputs $\varphi(N)$. Explain how you can use this algorithm as a black box to efficiently factor N in time polynomial in $\log N$.

Task 4 – Collision-Resistance from Discrete Logarithms (8 points)

In this task, we want to show that collision-resistant hash functions can be built in a cyclic group $\mathbb{G} = \langle g \rangle$ of *prime order* p where the discrete logarithm assumption holds. To this end, we consider the hash function $\mathsf{H} : \mathbb{G} \times \mathbb{Z}_p^2 \to \mathbb{G}$ such that

$$\mathsf{H}(S, (x_1, x_2)) = g^{x_1} S^{x_2} \ .$$

Note that here the seed is a group element $S \in \mathbb{G}$, the inputs are pairs of integers mod p, and the digests are group elements. The function is hence compressing, as there are $|\mathbb{Z}_p^2| = p^2$ inputs, and only $|\mathbb{G}| = p$ outputs.

a) [6 points] Show that if we are given a collision $(x_1, x_2) \neq (x_1', x_2')$ such that

$$\mathsf{H}(S, (x_1, x_2)) = \mathsf{H}(S, (x_1', x_2')) \ ,$$

then we can compute the discrete logarithm of S with respect to g, i.e., we can find x such that $g^x = S$. Explain where you used the fact that the order is prime.

Note: We only ask you to argue the above, and not give to a full reduction.

b) [2 points] Show that a malicious entity picking the seed $S = g^x$ for a *known* x can always find a collision for H with respect to seed S. In other words, a maliciously chosen seed S enables a backdoor for this hash function.

Task 5 – Security of Key-Agreement Protocols (9 points)

Consider the following key-agreement protocol:

- Alice chooses $k, r \leftarrow \{0,1\}^n$ at random, and sends $s = k \oplus r$ to Bob.
- Bob chooses $t \leftarrow \{0,1\}^n$ at random and sends $u = s \oplus t$ to Alice.
- Alice computes $w = u \oplus r$ and sends w to Bob.
- Alice outputs k and Bob computes $w \oplus t$.

a) [2 points] Show that Alice and Bob output the same key.

b) [7 points] In an execution of the above protocol, we let S, U, W and K be the random variables corresponding to the values of s, u, w and k. Also, let K' be a uniform n-bit string, drawn independently of S, U, W. For the protocol to be secure, (S, U, W, K) and (S, U, W, K') need to be computationally indistinguishable. Show that this is however not the case, i.e., give an efficient distinguisher D such that

$$\left|\Pr[D(S, U, W, K) \Rightarrow 1] - \Pr[D(S, U, W, K') \Rightarrow 1]\right|$$

is large.