Introduction to cryptology
TD#5

2022-W12, ...

Exercise 1: Discrete logarithms *(Mix exams '18 & '19)*

In the following questions, $\mathbb{G}$ is a finite cyclic group of prime order p (meaning that it contains p elements), g denotes one of its generators, and $h \neq g$ another element of $\mathbb{G}$.

Q. 1:

1. Give an example of a finite cyclic group, and specify its order and whether it is prime.
2. Under which condition is h a generator of $\mathbb{G}$?
3. Give the definition of the discrete logarithm of h with respect to g.
4. Is the map $[\![0, p-1]\!] \to \mathbb{G}$, $x \mapsto g^x$ injective? What if we take $x \in [\![0, p]\!]$ instead?
5. Give an algorithm that computes the inverse of an element in $\mathbb{G}$.
6. Is the map $\mathbb{G} \to \mathbb{G}$, $x \mapsto hx$ a permutation for all h? If not, under which condition on h is it one?

The *discrete logarithm* (DLOG) assumption for $\mathbb{G}$ states that given g, $h = g^a$, with $a \twoheadleftarrow [\![0, p-1]\!]$, it is hard to compute the discrete logarithm of h in base g. An adversary is said to *break* DLOG if she/he is able to perform this computation.

Q. 2: The *computational Diffie-Hellman* (CDH) assumption states that given g, g^a, g^b, with $a \twoheadleftarrow [\![0, p-1]\!]$, $b \twoheadleftarrow [\![0, p-1]\!]$, it is hard to find g^{ab}. An adversary is said to *break* CDH if she/he is able to find g^{ab}.

1. Show that if one can compute discrete logarithms in base g with cost L and an exponentiation in an arbitrary base with cost E, then one can break CDH with cost $\leq L + E$.

Q. 3: We define the *decisional Diffie-Hellman problem* (DDH) as follows: an adversary is given one of the two triples (g^a, g^b, g^{ab}), with $a, b \twoheadleftarrow [\![0, p-1]\!]$ or (g^a, g^b, g^c), with $a, b, c \twoheadleftarrow [\![0, p-1]\!]$, each with probability 0.5. The adversary wins if it correctly guesses which triple it was given. The DDH *assumption* then states that it is hard to win the DDH game with a significant advantage over a random choice.

1. Show that if one can break CDH with cost C, one can break DDH with advantage ≈ 1 with cost C.

Q. 4: An assumption A is said to be *stronger* than an assumption B if breaking B implies breaking A with a similar cost, but breaking A does not necessarily imply breaking B with a similar cost.

1. Order the DDH, CDH and DLOG assumptions from weakest to strongest.

Exercise 2: Interactive proof of identity

Let $\mathbb{G} = \langle g \rangle$ be a finite group of prime order p where the discrete logarithm problem is hard. A *prover* wants to prove to a verifier that s/he knows a number x s.t. $X = g^x$, with $X \in \mathbb{G}$. S/he suggests the following protocol for a *verifier* to check this assertion:

1. The prover picks $r \twoheadleftarrow [\![0, p-1]\!]$ and sends $R = g^r$ to the verifier
2. The verifier picks a *challenge* $c \twoheadleftarrow [\![0, p-1]\!]$ and sends it to the prover
3. The prover computes $a = r + cx \mod p$ and sends it to the verifier
4. The verifier computes g^a and accepts the proof iff. it is equal to RX^c

Q. 1: Show that if the prover indeed knows x, the verifier always accepts the proof.

Q. 2: Why is it important for an honest prover to pick a random r? What would happen if r was easy to predict (say with probability larger than 2^{-40})?

Q. 3: When running the protocol twice, why is it important for the two random numbers r and r' to be distinct?

Q. 4: Show that by picking R and c him/herself, a challenger is able to create a fake run of the protocol that is indistiguishable from a real one. (Hint: try to first pick c and a and compute an R that makes the proof valid.)

Remark: This last property of the above protocol has interesting consequences: it ensures that the prover does not reveal any information about the secret x. The same secret may then be used in many proofs without decreasing the security.

Q. 5: Despite the previous remark, why is there still a limit on the number of times a single secret may be used?

Exercise 3: Random Self-Reducibility of the DLP

In this short exercise, we will see that in prime-order groups, the ability to solve the discrete logarithm problem *on "average"* allows to solve the problem on any instance with a similar cost. This shows that the worst-case complexity of the problem is not more than the one of average cases (where an average case is defined to be a random problem instance)..

Let $\mathbb{G} = \langle g \rangle$ be a finite group of prime order p.

Q. 1⋆: Show how one can construct such a group $\mathbb{G}$ from the multiplicative group $\mathbb{F}_{2p+1}^{\times}$ of the field with $2p+1$ elements, p prime. More precisely, give an efficient (possibly randomised) algorithm that takes p as input and returns a generator of a subgroup of order p of $\mathbb{F}_{2p+1}^{\times}$.

Q. 2: Let $h = g^a$ be an element whose discrete logarithm we wish to compute. Show that if one knows $r \in [\![1, p-1]\!]$, this is equivalent to computing the discrete logarithm of g^{ar}. How would you need to adapt the statement if $\mathbb{G}$ were not of prime order?

Q. 3: Let $a \in [\![1, p-1]\!]$, explain why if $r \twoheadleftarrow [\![0, p-1]\!]$, then $\Pr[g^{ar} = X] = 1/p$ for all $X \in \mathbb{G}$. How would you need to adapt the statement if $\mathbb{G}$ were not of prime order?

Q. 4: Assuming you know an efficient deterministic algorithm to compute the discrete logarithm of a fraction of 2^{-10} of the elements of $\mathbb{G}$, give an efficient *randomized* Las-Vegas algorithm that computes the discrete logarithm of any element of $\mathbb{G}$.