

FUNDAMENTOS Y APLICACIONES DE BLOCKCHAINS

Homework 4

Depto. de Computación, UBA, 2do. Cuatrimestre 2025

1/11/25

Student: Juan DElia

Due: 13/11/25, 17:00 hs

Instructions

- Upload your solution to Campus; make sure it's only one file, and clearly write your name on the first page. Name the file '<your last name>_HW4.pdf.'

If you are proficient with \LaTeX , you may also typeset your submission and submit in PDF format. To do so, uncomment the "`%\begin{solution}`" and "`%\end{solution}`" lines and write your solution between those two command lines.

- Your solutions will be graded on *correctness* and *clarity*. You should only submit work that you believe to be correct.
- You may collaborate with others on this problem set. However, you must **write up your own solutions** and **list your collaborators and any external sources (including ChatGPT and similar generative AI chatbots)** for each problem. Be ready to explain your solutions orally to a member of the course staff if asked.

This homework contains 4 questions, for a total of 40 points.

1. **Bitcoin backbone: Chains of variable difficulty.**

(a) (5 points) Describe Bahack's "difficulty raising" attack.

Solution: El ataque consta de lo siguiente:

- Supone que en una ronda r todos los participantes tienen una cadena de longitud λm
- El atacante construye una época entera por su cuenta con Timestamps falsos, lo que resulta en una dificultad muy alta para la siguiente época.
- Setear un T' chico, tal que si computa el primer bloque de la siguiente época rápido, obtiene la cadena más pesada

Este ataque funciona con probabilidad constante.

(b) (5 points) What aspect of Bitcoin's target recalculation function makes the attack ineffective? Elaborate.

Solution: La función de recalcular define límites en la dificultad que se puede setear en cada época. Esto hace que el atacante no pueda setear un T arbitrario, si no que se va acotar. Esto basta para que el atacante no pueda definir una cadena tan pesada como para que sea la que se adopta

2. Proofs of Stake (PoS).

- (a) (5 points) Consider the *long range attack* on a PoS blockchain in a permissionless environment. Assume that an honest-stake majority always holds, and that all the parties are aware of the global clock. Describe the scenarios wherein the honest parties lose their advantages, and explain why freshly joining parties cannot distinguish an honest chain from other chains by the longest-chain selection rule.
- (b) (5 points) A PoS protocol performs leader election based on the stakes each party owns. Recall that the initial stake distribution is hard-coded in the genesis block. Does a PoS protocol assume a PKI? Further, the stakes in a party's account have monetary values. Where can the initial stakes come from?

3. Verifiable Random Functions (VRFs).

- (a) (4 points) Enumerate the security properties a VRF should satisfy.
- (b) (6 points) We stated in class that given a hash function, modeled as a *random oracle*, and an unforgeable signature scheme, VRFs are readily realizable. Show that that's indeed the case by proposing a construction and arguing its security—i.e., it achieves the desired security properties. Follow the VRF terminology we used in class.

4. **The *Ouroboros* protocol.**

- (a) (5 points) We saw in class that in the Ouroboros protocol the slot leader election process is abstracted out and modeled as an “ideal functionality.” Describe one realistic approach to elect the slot leader, and explain why there are \perp symbols in the characteristic strings.
- (b) (5 points) Describe the implementation of the *dynamic* stake setting. Why can't the parties use the most recent stake distribution (i.e., the stake distribution at the end of the previous epoch)?

