Implement a distributed consensus algorithm in SPDZ

The goal of this exercise is to implement a distributed average consensus scheme using SPDZ.

You can install SPDZ following these guidelines:

1) Install Ubuntu on your PC (e.g., v 20.04). Alternatively, install WSL in Windows (it is good to use the VS code extension for this).
2) Check the requirements:
   a. https://github.com/data61/MP-SPDZ#requirements
3) Download the distribution of MP-SPDZ here:
   a. https://github.com/data61/MP-SPDZ/releases
   b. I downloaded the mp-spdz-0.2.8.tar.xz
   c. Unpack the distribution in your working directory
4) In the terminal run:
   ```
   Scripts/tldr.sh
   ```
5) Run the tutorial (it uses two players with a input string [1 2 3 4])
   ```
   ./compile.py tutorial
   echo 1 2 3 4 > Player-Data/Input-P0-0
   echo 1 2 3 4 > Player-Data/Input-P1-0
   Scripts/mascot.sh tutorial
   ```
6) Documentation with more info: https://mp-spdz.readthedocs.io/en/latest/index.html

Consider the following distributed consensus problem:

$$\min_{x_i \in \mathbb{R}} \frac{1}{2} \sum_{i=1}^{n} x_i^T q_i x_i$$

where $n = 3$, and $x_1^0 = 100,\ x_2^0 = 10, x_3^0 = 30,\ q_1 = 2, q_2 = 4, q_3 = 1.$

Rewrite the problem as follows:

$$\min_{x_i \in \mathbb{R}} \frac{1}{2} \sum_{i=1}^{n} x_i^T q_i x_i$$
$$\text{subject to } x_i = \bar{x}$$

You can consider the x_i as local variables and $\bar{x}$ as a global variable. You can solve the problem using the alternating direction method of multipliers (ADMM). ADMM iteratively solves the following problem ($\rho = 1$ and `iter_max` = 10):

for k = 0, 1, 2, …, iter_max

for i = 1, 2, 3

$$x_i^{k+1} = \text{argmin}_{x_i}(x_i^T q_i x_i + \frac{\rho}{2} \| x_i - \bar{x}^k + u_i^k \|^2)$$

end

$$\bar{x}^{k+1} = \frac{1}{n}\sum_{i=1}^{n} x_i^{k+1}$$

for i = 1, 2, 3

$$u_i^{k+1} = u_i^k + x_i^{k+1} - \bar{x}^{k+1}$$

end

end

[3 pts] Solve the problem above in plaintext. Then,

- Report the values of each x_i and final $\bar{x}$ in one plot.
- Analyze the computation time of each iterate of the algorithm and the overall solving time.
- Attach the code to your submission with a README file.

[5 pts] Consider now that the values of x_i are private and encrypted. Use MP-SPDZ to solve the problem. **Hint:** Create your program under Source (e.g., `myprogramme.mpc`). The default scripts in the Scripts folder are set for 2 players only. You will have to create a new script to handle 3 players (e.g., you can start from `mascot.sh` and `run-common.sh`). Remember to run the command

```
chmod +x script-name.sh
```

to give permission to execute such a script.

- Show that you can you obtain the same values x_i and $\bar{x}$.
- Analyze the computation time of each iterate of the algorithm and the overall solving time. Do you see a difference compared to the previous case?
- Attach the code to your submission with a README file.

[2 pts] Consider now that both x_i and q_i are private and encrypted. Use MP-SPDZ to solve the problem.

- Show that you can you obtain the same values x_i and $\bar{x}$.
- Analyze the computation time of each iterate of the algorithm and the overall solving time. Do you see a difference compared to the previous case?
- Attach the code to your submission with a README file.

Submit the report as a PDF file with only the requested explanations to answer the questions above. Also provide the code with a readme file to run it.