I used the same n values with the previous project.

n=100, 70, 50, 20, 3
The accuracy Paillier: 100, 100, 100, 100, 100
The accuracy Shamir: None, None, None, 100, 100
The accuracy Differential Privacy: 99.04, 100, 99.14, 78.16, 100
None means that it is not available for accuracy check.
The accuracy shows the percentage of accuracy of each average compared to No Privacy Protection.

As you see above, once n exceeds 20, Shamir does not work and gives negative for average. However, Shamir gives you very accurate averages as long as n does not exceed 20, since there is no inherent noise added to the secret when sharing or reconstructing it. The accuracy of the reconstructed secret is high as long as the threshold number of shares is available.
In Pailler, addition and multiplication of encrypted values correspond to addition and multiplication, and this means that the results are consistent with the operations on the plaintext data when you perform operations on encrypted data.

The accuracy of Differential Privacy is the worst among the three, because the Differential Privacy is the only one which adds noise, and the more noise added, the more privacy is preserved, but the less accurate the result becomes.

The order of runtime from fast to slow is No privacy protection, Differential Privacy, Shamir, Paillier.
Usually, their runtime increases when n increases, but differential privacy runs in 0.0001 sec when n=100, but runs in 0.018 when n=70. I think the reason is that the amount of noise added can vary depending on the specific query and privacy parameters. This randomness can lead to variations in the runtime of individual queries, especially if the amount of noise needed is high.