Just a couple notes on my solution:

* As to my testing in general, you may find some dependencies represented as functions rather than objects as awkward. As I tend to maximize the use of pure functions, I find this style where relevant requires less code, and once I got the hang of it, just as easy to understand.

Where I used

Func<LotteryNumber> as a dependency, I could have just as easily used a class LotterNumberGenerator : I GenerateLotteryNumbers along with IGenerateLotteryNumbers.GetLotteryNumber.

I am quite happy to adopt the more traditional approach if the team wants.

* By "interface" in point 3, I assumed you meant it in the generic OO way (e.g. an class' public interface) as opposed to an actual C# interface construct. I extracted one from LotteryNumberService in case you did, but it's not needed by the test client.
* When you write "you must expost an API to enable remote querying of the lottery number generator service" I could not determine if you wanted an API in the strict sense of the word, or as-in a WebApi service for example. Being short on time, and not wanting to over-engineer I opted for the strict interpretation, which is implemented in the publicly accessible LotteryNumberService class.
* Re: Bonus 3b, I assumed the text file to be written if the IWriteFileToDisk.Write() method was called. Since there is no client for that code except the tests I didn't have the spare time to implment a concrete version that actuallywrites the file accompanied by an integration test. But the separation of concerns is there.
* I realize that tests like TheNumberShouldBeBetweenOneAndFortyNine and TheNumberShouldBeRandom are less than ideal, but I don't know of any better way to test something indeterministic like randomness. I made the tests deterministic using a seed value.
* While I think I have managed to isolate dependencies with a DI approach, I haven't mocked/faked every response. I realize the issues with ripple effects in test suites when you do this kind of thing on a large scale. I only did it with the random number generator to save the time of making a long static list, and I always used a constant seed to make it deterministic.