NYU, cs6903, Project 2, Type 3 (Designing a cryptography solution to allow computation over your outsourced encrypted data via homomorphic encryption):

The project consists of design, software implementation and demonstration of your method to allow computation over outsourced encrypted data via homomorphic encryption. Assume Alice wants to outsource her dataset (e.g., numeric datasets, and/or text, image, audio, video files with keywords) to a cloud server Carol. Then, assume Alice desires confidentiality for her data and thus stores in encrypted form, using a (partially or fully) homomorphic encryption scheme. Later, Alice may query Carol for a function to be computed over the stored encrypted data, as follows: (1) Alice specifies the function to Carol; (2) Carol evaluates this function over the encrypted data and returns a ciphertext to Alice; (3) Alice obtains the output of this function evaluation by decrypting the ciphertext obtained by Carol. Your project should consist of the following steps:

1) Choose Alice's dataset; suggestions include the following:

1. A database table with attributes and relative numeric values,
2. A list of text, image, audio, and/or video files, tagged with keywords, or
3. your favorite dataset (if approved by the instructor).

2) Choose Alice's desired class of query functions; suggestions include the following:

1. Statistical queries (Average, min, max, median, etc.)
2. Number of dataset entries matching a given keyword (chosen by Alice)
3. Position (within dataset) of dataset entries matching a given keyword (chosen by Alice)
4. All dataset entries matching a given keyword (chosen by Alice)
5. Any of queries 2,3,4 where matching is intended in the sense of string equality, or equality except for a few characters, or equality except for some wildcards
6. (Equality to a keyword kw1) OR/AND (Equality to a keyword kw2)
7. Natural generalizations of the formula in item 6
8. your favorite class of queries (if approved by the instructor).

2) Design a method, based on a careful choice of an homomorphic encryption scheme, for Alice to store an encrypted version of her dataset with Carol, and later query a function to be computed with Carol's help. A list of homomorphic encryption schemes and libraries implementing them can be found in https://en.wikipedia.org/wiki/Homomorphic_encryption

3) Analyze the performance of this method as input parameters (such as the length of the dataset, the allowed query functions) vary across some suitably large range

4a) Compare the performance of this method with the analogue method that does not use encryption, thus performing computation over plaintext data; or, alternatively

4b) Compare the performance of this method when two different homomorphic encryption schemes (for the same function) are used

Before starting implementation, you have the (strongly encouraged) option to check your design with the instructor, who will not go over too many details of your design but will tell you if he can see any major design flaws or omissions

5) Prepare a project presentation file (using, for instance, Microsoft Powerpoint). You must include in your presentation a detailed description of above steps 1-4 (only one between 4a and 4b), together with a demonstration (using videos or screenshots) of how your method works You will use this presentation file for your in-person or recorded 10/15-min workshop presentation.

Your submission will be judged based on the following project grading criteria:

1. Dataset/query function choice (i.e., how interesting are the dataset and function class, etc.)
2. Design validity (i.e., if you chose appropriate cryptographic primitives, if the schemes instantiating the primitives and their key length parameters are valid choices in terms of security and efficiency)
3. Comparison insights (i.e., if your comparison between 2 methods is insightful)
4. Implementation validity (i.e., if your software, after inspection of the presentation demonstration and some amount of testing, seems to satisfy correctness; if your software is easy to use / run, has a well-written readme file, etc.)
5. Demonstration/presentation quality (i.e., if the presentation is well written and insightful, if the demonstration is clear and insightful, etc.).