### Online based solutions

[5] – Could not find a free copy of this. Do you have one for me?

[6] – Already written

In [9], one of the earlier papers presented on encrypted searching, they point out that since the advent of cloud storage, many individuals and organizations would like exploit these services for their remarkable utility. However, while they may trust the cloud storage provider (CSP) to provide reliable storage, they do not necessarily trust them to secure their data confidentiality needs. Many organizations have very sensitive data and would like to safeguard it from being exposed.

The obvious solution is for the client to encrypt the data and upload the encrypted content to the CSP. However, if the client is on a device with limited resources, e.g., limited bandwidth, then this may not be reasonable solution, especially as the volume of information they need secure access grows in size. Rather, they need some way to allow the CSP to search their data, without knowing the contents of the data they are searching through (data confidentiality), nor the contents of the search query issued by the client (query privacy).

They propose a provably secure mechanism in which this is possible, with the limitation that only simple keyword string matching is possible. Typically, especially for large data repositories, a simple keyword search will not be able to adequately represent the information need of the user; in addition, it will fail on typographical errors, common word variations (“computes” instead of “compute”), and so on. The end result is it will return too many non-relevant documents and it may miss too many relevant documents.

In [1], they propose a system which permits provably secure conjunctive queries for certain capabilities, where a capability is defined as being able to determine if a pre-specified set of fields (like the “From” field in an email) match certain values, like whether the “From” field is equal to “Bob”.

By secure, they mean that, given a document with a certain conjunctive search (on a set of fields in the capabilities) is true, and another document in which the same conjunction search is also true, an attacker will not be able to distinguish these two documents appreciably better than a random guess.

Their work is important in that they allow for provably secure conjunctive search, which is an improvement over keyword searching discussed in [9], but their solution is still rather limited. Notably, they do not allow disjunctive queries, they still only perform exact string matching (instead of approximate string matching), and their solution only works on pre-specified “fields”. Indeed, this too can be a fairly serious breach of confidentiality, e.g., some “fields” may expose a lot of meta-information about the contents of the document.

In [?] – “Fuzzy Keyword Search Over Encrypted Data” – they propose a mechanism to address the limitation in which only exact matches on keywords are allowed. For instance, their solution adequately handles typographical errors. To accomplish this, for each keyword, they also add all of the k-edit error variations of it, where an error is an insertion, deletion, or substitution of a character. For example, to allow up to a 1-edit error on the keyword “age”, expand it to {age, \*age, a\*ge, ag\*e, age\*, \*ge, a\*e, ag\*}, where the \* represents any character, including null. So, when a user enters a query for age, it will also match any string in the encrypted document which matches any of the 1-edit error patterns.

This represents an improvement over previous methods with respect to enabling approximate matching, but there remain many limitations. They do not consider the importance of keyword proximity, e.g., searching for “hello world” will match on any document in which both the keywords “hello” and “world” exist regardless of how far apart they are. It is clear that a document which contains the exact phrase, “hello world”, is more relevant than another document that has “hello” and “world” on separate pages. Additionally, depending on the degree of approximation allowed, the index for allowing approximate matching can become quite large.