**SYSTEM ANALYSIS**

**EXISTING SYSTEM:**

* With the increasing popularity of cloud-based data services, data owners are highly motivated to store their huge amount of potentially sensitive personal data files on remote servers in encrypted form.
* Clients later can query over the encrypted database to retrieve files while protecting privacy of both the queries and the database, by allowing some reasonable leakage information.
* To this end, the notion of searchable symmetric encryption (SSE) was proposed.
* The above dynamic SSE schemes cannot achieve forward privacy, and they also leak a lot of additional information during the updates.
* In a recent work, by leveraging such update leakage, the authors showed some devastating file injection attacks which can be run on almost all the existing SSE schemes.

**DISADAVANTAGES**

* Encrypted data make effective data retrieval a very challenging task.
* Security problem
* From the perspective of search functionality, one common limitation of the SSE solutions is that they only support single-keyword search.

**PROPOSED SYSTEM:**

* We propose bucket-encrypting index structure with random generator, named BEIS. Based on the basic index structure, we encrypt data identifier vectors (DIVs) representing the collection of entire data files, using random numbers.
* BEIS allows practically efficient multi-keyword query with privacy preservation.
* We also propose bucket-encrypting index structure with homomorphic generator. To save bandwidth during the query process, strategically makes use of additively homomorphic encryption to encrypt DIVs.
* Finally, our formal security analysis shows that both our constructions enjoy security against adaptive chosen-keyword attacks (i.e., CKA-security) and also forward privacy.
* To demonstrate the practicality of our solutions, we conduct experimental evaluations using large representative datasets.

**ADAVANTAGES**

* To save bandwidth.
* Improved Security.
* Moreover, BEIS can support effective and secure batch data processing (including updates) due to the cipher text packing.
* Our constructions carefully make a trade-off between query efficiency and privacy, achieving flexible query functionalities.