Secure Multidimensional Range Queries over Outsourced Data

Bijit Hore, Sharad Mehrotra, Mustafa Canim and Murat Kantarcioglu

Abhishek Srivastava
Student ID: 861307778

March 7, 2017
CS 236, Winter 2017

The problem:
In this paper we tackle the same issue of supporting queries on the *database as service* model where client store all it data on the server and it tries to prevent the data by storing encrypted data, but supports ranged queries on single dimension rather than on multiple dimensions.

The contribution:
The authors present their approach as to use *bucketization* technique to generate *secure indexing tag* of data. Authors main objective is to use above mentioned procedure to support multidimensional queries on multidimensional data. There is a trade-off involved using this approach between computational cost and disclosure risk and it should be treated as a optimization problem.

The method:
The server stores all the data in encrypted form because it is considered as untrusty. The client side has two components: *query translator* and *query post-processor*. *query translator* translate the actual query into encrypted format so it can be queried upon the encrypted data which will return a subset of it. *query post-processor* is applied on the returned subset data which involves decrypting it and removing false positive results from it. Most important part of this process occur at server end which involves executing encrypted query on stored data.

On the server end multidimensional data is stored in indexed manner using bucketization. The multidimensional ranged query contains two similar dimensional points. Using these points a set of points is selected which satisfies the query predicate and then return this result to the client.

In bucketization method encrypted data should be divided into buckets such that bucket labels should not give any information related to data values stored in them. Due to this property result from encrypted query give false positive with correct answers. Bucketization consists of two phases: first create buckets with size which optimizes performance, **greedy partitioning algorithm** is used in this paper and second, re-distributing content in controlled manner into multiple composite buckets and then making content more diverse in each bucket by manipulating entropy and variance of distributed data which decreases disclosure risks, **Controlled diffusion** is used in this paper for this purpose.

Comments:
The paper presents a way to perform a multidimensional encrypted query on stored encrypted multidimensional data on the server and provide optimal balance between cost and risk .

However, there are some drawbacks of proposed algorithm:

- Greedy method to evaluate partitioning bucket size may not always result into best performance.
- How bucketization will effect the performance in case of dynamic data addition and update.