A MovingObject Index for Efficient Query Processing with PeerWise Location Privacy
by D. Lin, Ch. S. Jensen, R. Zhang, L. Xiao, and J. Lu

1 Summary

In the paper, authors consider a problem of preserving security (misnamed as “privacy”) of moving users (peers) in spatio-temporal settings. The main focus is on peer-wise location security, i.e., preventing peers from discovering location of other peers, unless defined and accepted security policies allow that. Specifically, different peer users are allowed to see the location of a user when the user is within a specified spatio-temporal range. Two types of queries are supported – range and $k$ nearest neighbors.

To support such queries authors introduced a new indexing technique called PEB-tree, which bases on $B^x$-tree indexes and indirectly on $B^+$-trees. The PEB-tree uses as an index key new values that express location, security policy, and time of peers. With such keys, all peers, which are close to each other, and can “see” each other are grouped close to each other on a hard drive (similar values of the index key). Range and $k$ nearest neighbor query algorithms exploit the PEB-tree to simultaneously filter candidate users according to both security policies and spatial proximity.

Authors evaluate their approach, also using their own cost function. PEB-tree is more efficient than a spatial index, which is caused by considering security policies.

2 Strong Points

- Privacy policies are well defined as triples: role, location, time
- PEB-tree bases on $B^+$-tree, which promises easy integration into existing commercial database systems
- Rather than modifying DBMS authors introduced new way index key. Its usage in the index improves performance of query executions.
- All technicalities are well described.
- Paper is well written, with enough examples explaining difficulties.

3 Weak Points

- Authors are focused only on improving performance of peer-wise security in filter-based solution. They do not consider other solutions to the
- No comparison of PEB-tree and multi-index approaches, i.e., settings, where time, location, and security policies have different indexes each.
- Missing security analysis, which would specify security breaches from subqueries of the $k$ nearest neighbor query.
- Source of the Figure 1 is not clear. Very similar (the same ?) figure can be found in Wikipedia for “Bx-tree”, with a creation time in 2008. Wikipedia author allowed to distribute that picture with proper attribution.

4 Questions and Discussion Points

Authors have not considered another baseline scenario, which seems to be quite obvious. Considering encoded security policies as an additional attribute in the database, and using any well know index, should be a good reference point for experiments and any analytical comparison with the PEB-tree approach. In addition, authors have could consider approximating restrictiveness of a query predicate (including predicate on policies), and using such values to execute the query in a proper order, i.e., starting from the most restrictive attribute.

An alternative approach to optimization of queries considered in the paper is to optimize query execution, in general. Knowing security restrictions, location, and time of query predicates a DBMS can build efficient query execution plan. Comparing such approach with PEB-tree would be valuable for this research, as it would be position efficiency of PEB-tree among other methods.