

# Privacy Preserving Mobile Services: An Access Control System for Moving Objects and Customer Profiles

Mahmoud Youssef

Nabil R. Adam

Vijayalakshmi Atluri

MSIS Department and CIMIC - Rutgers University

{youssefm, adam, atluri}@cimic.rutgers.edu

## 1 Introduction

A key challenge for Mobile services (M-Services) is to offer personalized contents while preserving the privacy of customers. Personalization involves the collection of different types of information which raises privacy concerns. This information includes customer profiles and preferences, location information, and click stream. Studies have shown that most customers would not opt-in to personalized services unless they are assured that their information will not be shared. Nonetheless, other studies have found evidence that customers are willing to trade-off their information for convenience. In this work, we present an access control model that enables customers to specify spatio-temporal access policies on their profile and location information. We also present an enforcement mechanism that utilizes a new variation of the trie structure that we term the Adaptive Search Multi-way trie (ASM-trie).

Mobile customers can be *continuously* moving which creates an update problem to traditional databases. The Moving Objects Databases (MOD) and the Moving Objects Spatio-Temporal model (MOST) [SWCD97] have been proposed to address this problem. In the MOST, location information is represented as a linear function of time. In addition, the MOST supports several types of queries that can usually be reduced to a *spatial window* with a *time interval* query. It also supports *predictive queries* which include time interval about the near future. The MOST has been adopted by most of the indexing techniques in the literature. We also adopt the MOST in our access control system.

The task of designing access control system for customer information in m-services is a challenging undertaking due to several reasons: (1) the spatio-temporal nature of the information to be protected as well as the constraints; (2) the model size which is related to the number of customers and merchants; and (3) the desired granular representation of the model components. In addition to these challenges, the enforcement of access control should not adversely impact the response time of the query.

## 2 The proposed solution

First, we propose to maintain all profiles in a secure centralized database. Since customers trust the Location Service (LS) for their location information, it is then prudent and economical that the LS maintains that database. In addition, it becomes possible to apply access control on the centralized database.

### 2.1 The access control model

The components of the access control model are mapped as follows: *subjects* are the merchants; *objects* are customer information (profile and/or location); and the only *access mode* is *read*. It is important to notice that the merchant does not actually read customer information; rather the permission is whether the customer permits receiving offers from that merchant at this location and time. Moreover, since *read* is the only access mode, it is removed from the model. Consequently, an access rule in the proposed model consists of two parts: an authorization triple  $\langle s, o, +/- \rangle$ , and a spatio-temporal constraint (*stc*) where  $s$  is a subject,  $o$  is an object, and  $+/-$  is a flag (that is, grant/deny). The spatio-temporal constraint is a geographic location (e.g., a county) and a time interval (e.g., working hours) to which the authorization is applicable.

The proposed model cannot be classified as a closed or an open system. Rather, each customer starts from a closed system where she has one rule that assigns negative access to all merchants on all objects at all locations during all times. Later, the customer can change the flag of this rule or add other rules that override it partially.

We adopt a hierarchical representation for subjects, objects, location, and time. The root of each hierarchy is *All members*, e.g., “All Industries”, and the leaves are the individual members at their most specific representation, e.g., a merchant. Thus, the nodes along the path from a leaf to the root represent a member at its different granularities.

In the integrated model (Figure 1), each object, a customer ID + a subset of location and profile, has its own industry hierarchy; every node in that hierarchy has its own location hierarchy; and similarly, each node in the

location hierarchy has its own time hierarchy. However, with exception to roots, nodes are only created when they are specified in an access rule. To the contrary, the roots are created while inserting the closed system rule. In addition to the granular representation, this approach provides a smaller model size.

Conflict among the rules specified on different nodes on the same path is resolved using *inheritance with override* scheme. In that scheme, if no rules are specified for a given member, then it is assumed to inherit the permissions from the immediate ancestor that already exists. Otherwise, an access rule specified on a member overrides all the access rules specified on its ancestors. As a result, the evaluation mechanism needs to search only for the most specific representations of the components.

## 2.2 Evaluation of access rules

Evaluation of access rules involves composing search keys and matching them to the access rules database. Each customer in the query results produces several search keys due to the different representation and to the different locations/times that intersect the query predicate. The evaluation process undergoes two stages; first, for each individual combination of location and time leaves, a flag is obtained by applying inheritance and overriding. In the second stage, permissions from different location and time combinations are defused using the *least privilege* principle.

The possible combinations of location and time values are calculated as the intersection of the customer linear motion function (from the MOST) and the query spatial window when both are projected on a line map, e.g. the census TIGER. For each leaf location, we calculate the entry and exit times. If both times are in the same leaf, they are merged as one time leaf. Finally, possible combinations are the cross product of locations and times.

In the first stage, we search for an existing access rule that overrides all the other existing rules. Therefore, we start traversing the hierarchies using leaf value of each component. Upon failure, the search retreats to the next representation. If the search keeps failing, it will end up finding the closed system access rule. We term this search procedure the *adaptive search* (Figure 1).

## 2.3 The enforcement mechanism

The evaluation and enforcement mechanism consists of three basic components: a *spatial module*, the *ASM-trie*, and an *encoder*. We first encode access rules into a string database and insert them in the memory as an ASM-trie. Upon receiving a query result, the access control is enforced as follows: the spatial module composes the possible combination of search keys for each customer in the result, the keys are then encoded, and finally, the ASM-trie is searched for those keys. A decision is made to whether to leave a customer in the query result by

defusing all the flags for that customer.

**2.3.1 The ASM-trie.** The proposed trie is based on a 27-character radix: the 26 letters from A to Z and an end-of-string symbol. In the ASM-trie, the node (Figure 2) includes a pointer to its parent for backward traversal, and a Boolean variable to indicate whether this level supports variable search. The leaves of the ASM-trie, which have a different node structure, are the flags of the access rules.

We conducted an evaluation study of the performance of the ASM-trie, versus a regular trie and a DBMS based search. Figure 3 shows the results on a *logarithmic scale*.

## References

[SWCD97] P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao, "Modeling and Querying Moving Objects", In *Proceedings of the 13th IEEE ICDE*. Birmingham, UK, 1997.

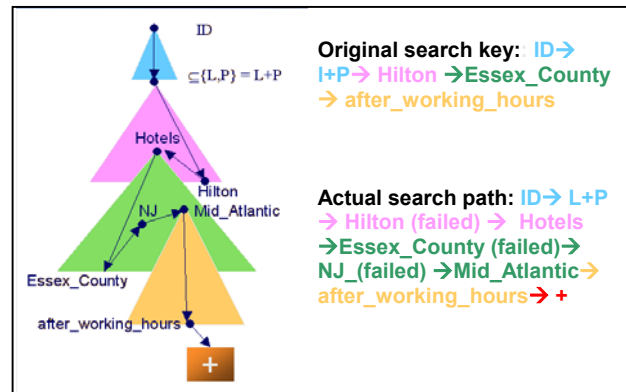


Figure 1: Adaptive search in model hierarchies

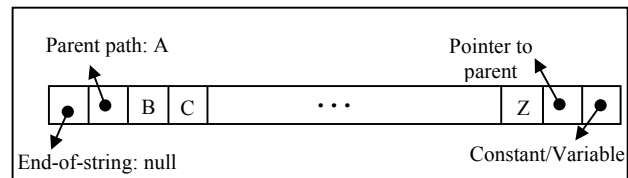


Figure 2: An ASM-trie node

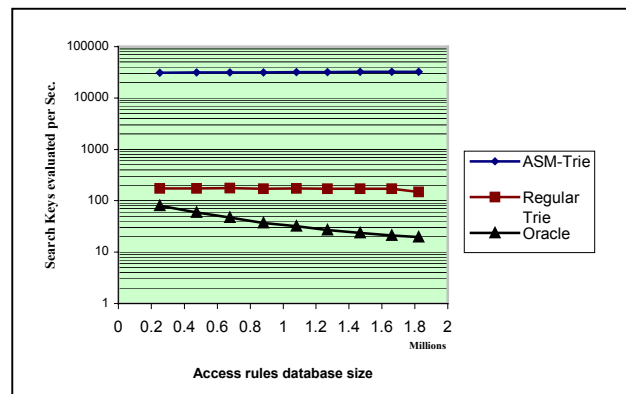


Figure 3: Search performance among ASM-trie, Regular trie, and Oracle