PeRA: Individual Privacy Control in Intelligent Transportation Systems

Martin Kost, Raffael Dzikowski, Johann-Christoph Freytag

DBIS Group
Humboldt-Universität zu Berlin
[kost|dzikowsk|freytag]@informatik.hu-berlin.de

Abstract: In the domain of Intelligent Transportation Systems (ITS) manufacturers and service providers start to implement and deploy plenty of (new) applications running on a vehicle. These applications involve the user and external services. Therefore, we must incorporate mechanisms providing the individual for controlling his/her privacy. Existing approaches only consider to control the event of data access using a central instance. In contrast, we consider to implement individual privacy requirements for the complete data flow of distributed systems. The Privacy-enforcing Runtime Architecture (PeRA) provides a holistic privacy protection approach, which implements user-defined privacy policies. A data-centric protection chain ensures that ITS components process data according to attached privacy policies. PeRA instances constitute a distributed privacy middleware, which evaluates privacy policies to mediate data access by applications. The PeRA architecture includes an integrity protection layer to create a distributed policy enforcement perimeter between ITS nodes, which prevents the circumvention of policies. We implemented the PeRA architecture as a proof-of-concept prototype.

1 Introduction

Designing and implementing co-operative mobile systems that comply with current and future privacy regulations is a great challenge today. Evolving Intelligent Transportation Systems (ITS) provide cooperative applications which implement an improved functionality such as enhanced travel services, driving support, and transportation optimization. These applications exchange information about participating individuals (e. g., vehicle owners and drivers); thus, impacting the privacy of persons. Uncontrolled information flows potentially allow for privacy infringements (e. g., generating movement profiles).

For identifying possible privacy threats and appropriate protection requirements, we analyzed and applied domain independent privacy principles on the ITS domain; especially the functional requirements and processes [Die12]. Thereby, one of the challenges which we address is to prevent an attacker from circumventing the defined privacy constraints within a distributed system. For instance, if we apply a policy enforcement mechanism we have to guarantee that the policies of the individuals as well as the application code will not be manipulated. Additionally, we have to guarantee the privacy-compliant execution

of applications which consists of (standard and user-defined) operations. For addressing these challenges our approach requires privacy policies resulting from a comprehensive privacy analysis. Based on existing solutions such as *Hippocratic databases* [AKSX02], we designed and implemented the privacy middleware *Privacy-enforcing Runtime Architecture* (PeRA) which realizes the identified requirements.

In the following, we introduce the concepts of our ITS privacy middleware PeRA and describe our demonstration scenario. Our demo setting consists of the ITS nodes (1) motorcar, (2) truck, (3) Road Side Unit (RSU), and (4) traffic control center. We implemented and distributed the ITS applications (a) Moving Map, (b) Intersection Collision Detection, (c) Traffic Status, and (d) Fleet Management on these nodes. Running the scenarios we will demonstrate—by visualizing the resulting effects—how an individual may configure policies in order to control the processing of his/her data within this distributed system.

2 PeRA Concepts

Most technical proposals for privacy preservation in ITS only support single applications. We developed a policy-based privacy enforcement architecture which provides an application independent privacy middleware for ITS. Moreover, current solutions for protecting privacy implement mechanisms to control the event of accessing data from a central instance such as a database management system. In contrast, our architecture PeRA controls data processing for the complete data flow; i.e., we include events such as data communication and data processing by different applications or remote nodes.

PeRA provides a policy enforcement perimeter that realizes a data-centric approach for privacy protection. Subjects get control of their data by declaring/configuring privacy policies which restrict how applications may process their data. For instance, the following policy statement specifies a context—which is defined by a set of constraints—together with operations which are permitted within this context, and a reference to the new policy.

```
Policy-ID="Example-Privacy-Policy-1"
Context(node-type="traffic control center" and ...) {
 Permit access On location From db.vehicle With
    PostCondition(anonymity-value="10")
 Post-Policy="Example-Post-Policy-A"}
```

All data is combined with an immutable set of privacy policies upon creation; e. g., we couple all GPS data of a vehicle which is sent to the traffic control center and stored in its local database *db* with the previous described policy *Example-Privacy-Policy-1*.

Mandatory privacy control (MPC) components ensure that applications only perform policy compliant operations on the data. To prevent data processors from circumventing the MPC, we introduce the MPC integrity protection (MIP) layer [KWD+11]. The MIP layer stores data securely and encrypts data for information exchange between PeRA instances. It monitors the integrity of MPC components and only grants data access if all MPC components are in a trusted state. The MPC components mediate all data access and processing. An application poses an operation request as a query to the Privacy Control Monitor (PCM). The PCM evaluates the privacy policies of affected data items. Based on the eval-

uation result, a query is rejected or executed. Also, the MPC may perform additional data transformations on the data to meet the privacy requirements specified in attached policies. For instance, a result set might be perturbed to ensure a certain anonymity set size. We guarantee for all data that leaves the PCM to be policy compliant. However, once outside the control of the PeRA, we cannot guarantee policy enforcement anymore. Therefore, external applications may not gain data access on the required level of detail. Thus, we integrated an application sandbox, the Controlled Application Environment [Die12] (CAE). Application parts running inside the CAE are heavily restricted in their communication and resource access capabilities. Controlled applications may have detailed data access, which is mediated and controlled by the CAE and PCM.

3 ITS Demonstration Scenario

We use a real world ITS scenario to demonstrate the privacy protection capabilities of the PeRA prototype. The current scenario, depicted in Figure 1, is an extension of a previous demonstration [KWD+11]. Each of the nodes (vehicles, RSU, and the traffic control center) runs a separate PeRA instance.

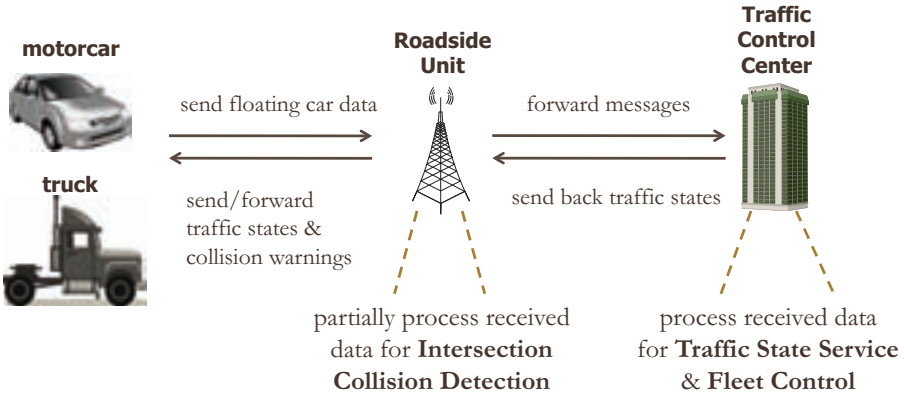

Figure 1: A Visualization of the ITS Demonstration Scenario.

In our setup, each vehicle runs a *moving map* application which displays the current location of the vehicle on a map. Furthermore, the vehicles send floating car data (FCD) records (of pre-recorded location tracks) to the RSU. These FCD records contain information about location, time, traffic status (light, normal, dense), license plate, and cargo. The RSU mediates the communication between vehicles and the traffic control center and provides an intersection collision detection (ICD) functionality. Thereby, the *ICD* application sends collision warnings to impacting vehicles and the *moving map* application of the RSU displays vehicles in its range. In the backend, the traffic control center provides traffic information/management services. A *vehicle tracking* application requests single FCD records to provide real-time tracking of single vehicles on a map. Fleet management

and freight tracking are typical purposes for such an application. Furthermore, a *traffic status* application provides a map showing the real-time traffic situation in a road network.

For all personal information the drivers may dynamically configure their privacy preferences. We provide an abstract user interface with a slider (for selecting a privacy protection level), check boxes, and textual descriptions in order to simplify the policy specification. The interface provides the following options: (1) permit all data processing; (2) permit only the specified data flow of the scenario; (3) no data processing is permitted; (4) individual privacy configuration: a) the single applications at the different nodes are permitted or not, b) full details for fleet management on/off. User settings are subsequently translated into corresponding privacy policies which the PeRA instance then permanently attaches to new/imported data. Thus, the user determines how applications can use the submitted FCD records. PeRA instances evaluate the attached privacy policies and permit an execution of policy compliant operations on the data only. In general, policies describe the permitted operations on certain data items, possibly including additional constraints, such as a certain degree of obfuscation.

The demo system offers two views on the showcased scenario to foster better understanding of the prototype's policy enforcement concepts and effects as well as architecture internals. The *application views* visualize what data is available to specific applications with different purposes. The *information flow view* shows the processing flow inside a PeRA instance and effects of policy-based decisions. Given a request, we construct the corresponding data flow graph enhanced with selected privacy properties.

4 Conclusions

In this paper, we address the challenge of designing and implementing ITS applications for distributed systems in a privacy protecting manner. The described PeRA concepts solve this issue by providing a privacy middleware that gives the individuals the control about his/her data. Our implemented prototype ensures privacy policy compliant data processing throughout an Intelligent Transportation System. Using an ITS scenario we show how PeRA supports a wide range of ITS applications and illustrate the fine-grained control mechanisms of our privacy policy enforcement.

References

[AKSX02] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Hippocratic Databases. In *Proceedings of the 28th VLDB Conference*, Hong Kong, China, 2002.

[Die12] Dietzel, S. et al. CANE: A Controlled Application Environment for Privacy Protection in ITS. In *12th Int. Conf. on ITS Telecommunications (ITST 2012)*. IEEE, 2012.

[KWD+11] M. Kost, B. Wiedersheim, S. Dietzel, F. Schaub, and T. Bachmor. PRECIOSA PeRA: Practical Enforcement of Privacy Policies in Intelligent Transportation Systems. In *Proc. of the Demo. Session at the Fourth ACM Conf. on Wireless Network Sec.*, 2011.