Data Protection and Data Sharing in Telematics

Sastry Duri, Jeffrey Elliot, Marco Gruteser, Xuan Liu, Paul Moskowitz, Ronald Perez, Moninder Singh and Jung-Mu Tang
IBM T.J. Watson Research Center, Hawthorne, NY

Abstract. Automotive telematics may be defined as the information-intensive applications enabled for vehicles by a combination of telecommunications and computing technology. Telematics by its nature requires the capture, storage, and exchange of sensor data to obtain remote services. Such data likely include personal, sensitive information, which require proper handling to protect the driver's privacy. Some existing approaches focus on protecting privacy through anonymous interactions or by stopping information flow altogether. We complement these by concentrating instead on giving different stakeholders control over data sharing and use. In this paper, we identify several data protection challenges specifically related to the automotive telematics domain, and propose a general data protection framework to address some of those challenges. The framework enables data aggregation before data is released to service providers, which minimizes the disclosure of privacy sensitive information. We have implemented the core component, the privacy engine, to help users manage their privacy policies and to authorize data requests based on policy matching. The policy manager provides a flexible privacy policy model that allows data subjects to express rich constraint-based policies, including event-based, and spatio-temporal constraints. Thus, the policy engine can decide on a large number of requests without user assistance and causes no interruptions while driving. A performance study indicates that the overhead is stable with an increasing number of data subjects.

Keywords: data protection architecture, automotive telematics, privacy policies

1. Introduction

Information is the new currency of the global economy. We expect the right information at the right time to support decision-making or to provide a service. As we march towards realizing Mark Weiser's vision of ubiquitous computing, a number of our living spaces will be embedded with computational capabilities. Automobiles are one instance of such spaces, where information is created or exchanged. The automobile is, in effect, a computing platform to which mobile commerce services may be delivered. Services available today and projected for the near future include navigation information, emergency roadside assistance, location-based services, delivery of digital information such as e-mail, entertainment, diagnostics and prognostics, and pay-for-use rental and insurance. These applications are enabled by the collection and use of data which may include information on the location of a vehicle as a

function of time, emergency situations including accidents and personal health emergencies, diagnostic data on the many systems within the vehicle, services and entertainment that are selected by the vehicle occupants, the demographics of the driver and passengers, and the behavior of the vehicle driver.

The growth of e-commerce on the World Wide Web has been limited by the reluctance of consumers to release personal information. In Building Consumer Trust in Online Environments [14] the authors find that "Fully 94 percent of Web users have declined to provide personal information to Web sites at one time or another when asked and 40 percent who have provided demographic data have gone to the trouble of fabricating it." If potential automotive telematics users share the concerns of Web users, then a large segment of the potential telematics market, perhaps as much as fifty percent may be lost.

In addition, service providers will depend on the integrity of collected information. Without data protection, end users or consumers may substitute false data or hack into in-vehicle applications to achieve a financial gain. Thus, data must be protected, so that users are assured of their privacy and that the data meets service provider's integrity requirements.

In this paper, we propose a framework that provides data confidentiality and integrity for automotive telematics services, while still enabling the sharing of data with service providers. We identify the need to support heterogeneous in-car platforms, to verify the integrity of sensor information, and to grant or deny requests for data without user assistance as key challenges in the telematics domain. The proposed architecture addresses these through a flexible data aggregation framework and can exploit a trusted hardware base. Expressive privacy policies govern the release of data. Privacy policies can include various context-dependent constraints and allow a hierarchical organization of requesters and data items. We discuss the implementation of our policy engine and present performance measurements.

The remainder of this paper is organized as follows. The next section describes related work and provides background in privacy and data protection. Section 3 characterizes the automotive telematics application domain with an insurance scenario. Section 4 then identifies data protection challenges that are specific to the telematics domain. Next, section 5 describes the proposed data protection architecture and section 6 explains the design of our policy engine. Finally, section 7 evaluates the applicability of our privacy policies and the performance overhead.

2. Background and Related Work

In a general sense, privacy may be defined as the ability of individuals to decide when, what, and how information about them is disclosed to others. Privacy principles [11, 21, 17] demand that systems minimize personal data collection. Before personal data can be collected, consent from the users needs to be obtained by notifying them about the nature and purpose of their data-collection and offering policy choices. Furthermore, it requires the application of privacy preferences, either through technology, business practices, laws, or some combination thereof, in the use and further dissemination of the disclosed information.

Anonymity mechanisms effectively avoid the collection of personal data. Gruteser and Grunwald [13] describe the identification risks inherent in location information, which would be released when using a location-based service. They also present algorithms that perturb location data in an automotive telematics context to achieve a specified degree of anonymity. However, at least for some services, anonymity conflicts with the service provider's need for billing.

Several approaches to handling privacy preferences during personal information exchanges that are not based on anonymity have been proposed (the interested reader is referred to Bohrer and colleagues [5, 4] for a more detailed discussion of these methods). For example, AT&T Privacy Minder [3], provides Web-privacy enforcing agents that enable individuals to formally express their privacy preferences in P3P (Platform for Privacy Preferences) [26], and automatically match them to the privacy policy of any Web sites visited by the individual. Standards have also been developed that promote the exchange of data through non-Web messaging systems. The Customer Profile Exchange Specification or CPExchange [7] is a standard that defines how a P3P policy can be associated with personal data in an XML message. Concepts for personal location policies based on the premise that individuals should be able to adjust the accuracy of observed information for specific purposes are presented in [24]. The author defines the notion of observation to consist of identity, location, time of observation, and speed, and presents a language for formulating personal location policies. The IBM Privacy Services (IPS) system, based on the IBM Enterprise Privacy Architecture [15, 16], provides individuals with the means to specify relatively complex privacy policies over their data and the means for automatic and manual authorization for release of this data by matching the individual's privacy policies with those of data-requesters. We extend IPS with context-sensitive constraints in privacy policies,

and provide support for publish/subscribe functionality specifically to address the telematics application domain.

Physically and logically secure systems would resist most physical and logical attacks (e.g., physical penetration, voltage or temperature attacks, power analysis, monitoring of electromagnetic emissions), and sensing and responding to all others before a system compromise (e.g., by rendering sensitive data inaccessible). However, such systems do not currently exist commercially. What does exist are secure coprocessors: physically and logically secure subsystems that operate in conjunction with a local host system, employing cryptographic acceleration hardware, and providing a secure execution environment for the programs that are to be run [23, 8]. The Trusted Computing Platform Alliance (TCPA) specifies a device and architecture that provides for some protected data storage, as well as metrics that measure the integrity of the software stack from system boot time, along with the ability to attest to the state of the software stack [25]. However, aside from specific operations, TCPA does not allow for protection of general computation within the device itself. TCPA's advantage over low-end secure tokens (e.g., smart cards) is that it is integrated with the rest of the system, and it is designed to be the root of trust in the system (the use of tokens in conjunction with a TCPA compliant system is complementary). If used properly, and in conjunction with a trusted operating system, TCPA does have the potential to provide the necessary platform for our proposed framework, both for the in-vehicle client systems as well as the service and solutions providers' servers.

Platform security deals with building a secure platform from the ground-up and involves extending a chain of trust, which is rooted in power-on hardware to subsequent layers using cryptographic verification techniques. The power-on software layer cryptographically authenticates/verifies a minimal set of commands and data that enable the configuration and update of the subsequent software layer. Once the platform system software has been securely configured/updated, the power-on software layer authenticates the operating system before each execution/instantiation [2]. The operating system must provide certain security features, such as access control, in order to support overall system and data protection and must be capable of virtual machine application authentication, and execution [19, 18]. A method to generate secure audit logs that are suitable for verification by third parties without actually knowing the contents of the log entries is described in [22]. Such a logging mechanism should be an essential component of every data sharing and data protection system.

Mobile Agents enable running computations close to the data sources [12]. Issues and requirements for mobile agent security are dis-

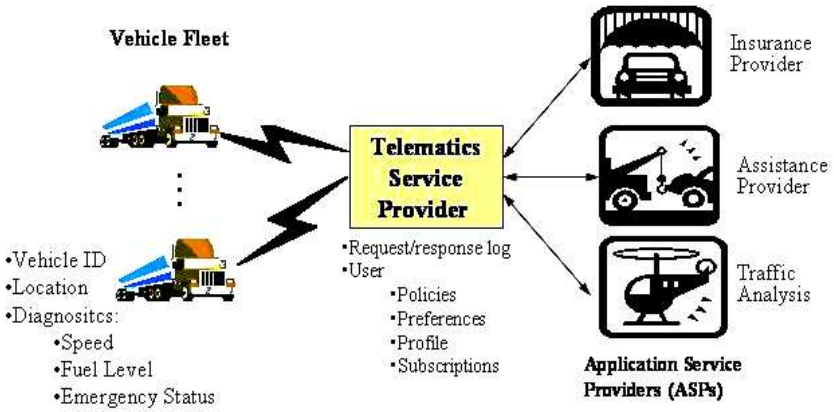

Figure 1. Typical automotive telematics applications.

cussed in [9, 20]. Wilhelm [27] uses mobile agents, and tamper resistant hardware to protect privacy. Parties requesting access to private information receive mobile agents that encapsulate private data and access control policies. The agents are executed in a protected environment to access data. The mobile agent notifies the owner whenever personal information is created, or accessed.

In our framework, agents are deployed by parties requesting access to private data into a trusted system holding private data. This enables the interested party to use private data without the data leaving the system trusted by the owner. The tamper resistant hardware, when used, protects both the data and computations involving private data from both physical and logical attacks, and thus promotes trust by both data owner and data user.

3. Automotive Telematics Applications

Figure 1 shows an overview of a typical set of automotive telematics applications. Vehicles are equipped with a wireless communication device, a variety of sensors, and a computer. The computer has a display, sufficient memory, storage, and processing to run complex embedded applications and middleware. The vehicle computer interfaces to the

vehicle bus and other sensors and collects engine performance data and safety information. In a typical scenario, vehicle location data is obtained from a Global Positioning System (GPS) sensor.

Vehicle users subscribe to the telematics service provider (TSP) to get a variety of services from application service providers (ASP) which include pay-for-use insurance, traffic analysis, and emergency assistance as shown in Fig. 1. Each vehicle transmits the data collected from sensors to the TSP. The TSP shares the data with ASPs according to the user service agreements. In addition to vehicle sensor data, the TSP stores user preferences and user subscriptions to services.

3.1. Pay-for-Use Insurance Scenario

The following scenario, taken from the user's point of view, illustrates how a customer may choose among a set of privacy policies, how the data may be aggregated by a TSP, and how the data is used to calculate the customer's bill.

3.1.1. *Enrollment*

Jane is a working professional who uses her automobile to commute a short twenty miles to work and for local shopping. She uses a rental car for company business trips. Thus, she is interested in the new pay-for-use (PFU) program that is offered by her insurance company, Giant Inc. The description of the program that she received in the mail indicates that she can enroll by calling an 800 number or by using the company's web site. Jane chooses the web site.

Jane enters the URL of the site on her laptop at home and quickly sees the page for the Giant PFU program. The page explains that PFU subscribers will be charged only when they use their car. Rates will be based upon miles driven and whether the driving is done in an urban area or a suburban area such as the one in which Jane lives. The page also explains that there can be several privacy polices available. Each privacy policy offers a different degree of data sharing and premium discount.

Policy 1 This policy provides the greatest degree of personal protection. Only Jane's cumulative data, not detailed location data, will be available to the insurance company without Jane's explicit consent.

Policy 2 This policy allows Giant full access to Jane's anonymized driving data; that is, all personal identification information has been stripped from the data and the data is not accurate enough to reidentify Jane. Only summary reports of total cumulative mileage

are sent to Giant with Jane's ID attached. It also allows Giant to sell anonymous data to third parties. This policy is offered at a five percent discount with respect to policy 1.

Policy 3 This policy allows Giant full access to Jane's driving and personal information to enable Giant to provide Jane with special offers. But it does not allow sharing that data with third parties. This policy is offered at a ten percent discount with respect to policy 1.

Policy 4 This policy allows Giant and third parties full access to Jane's driving data and personal information. This policy is offered at a fifteen percent discount with respect to policy 1.

Jane chooses Policy 2. She does not mind having her anonymous driving data used by Giant and third parties. The enrollment web page asks Jane to enter her insurance ID number to confirm her choice. Jane installs necessary software in her car and is ready to go.

3.1.2. *Driving - Data Aggregation*

That evening when Jane starts her car, she is pleased to see a message appear on the navigation screen: "PFU system now running - press # 1 for charges incurred this month." Jane presses # 1, only to see the message "Cumulative Charges for January 2003 - $0.00." Of course, she has yet to drive any distance. She tries # 1 again after returning home. This time the screen reads "Cumulative Charges for January 2003 - $1.00." Jane does a quick calculation; at 5 cents per mile, her yearly insurance bill for the 15,000 miles that she normally drives will be only $750. This represents a savings of more than $250 per year over her previous insurance rates.

As Jane drives, her data is accumulated at the CarAid center in a trusted computing system that is not directly controlled by Giant. CarAid is a telematics service provider that delivers a variety of services to Jane's vehicles: emergency assistance, navigation, concierge services. Monthly reports on total mileage for urban and suburban areas where Jane has driven are sent by CarAid to the Giant billing computer. Policy 2 also allows anonymous location information to be divulged to Giant and third parties—provided that the information is so unspecific that Jane cannot be identified. The Giant billing computer calculates charges based upon cumulative mileage and sends bills to Jane. Jane is pleased to see that the charges in the bills correspond to the charges that she has been informed of by her in-car device.

4. Challenges

Mobile Commerce in the automotive telematics domain creates several challenges that have not been addressed by privacy architectures for ubiquitous computing and location-based applications.

Data Integrity and Authenticity Mobile commerce applications such as pay-for-use insurance directly affect users' finances; thus, some customers may be tempted to cheat by providing false data. The service providers' data integrity and authenticity concerns are amplified by the requirement to collect data in a vehicle that is under complete control of the customer. Unlike transactions that create data through direct user interaction with institutions such as banks, book stores, or online web sites the private data used in automotive telematics transactions is created outside user interaction with the service provider. As a result, the architecture should provide means to ascertain the integrity of the data—in addition to the customer's privacy. We discuss a number of mechanisms differing in cost and complexity to address this issue.

Flexibility and Heterogeneity The average lifetime of an automobile is much longer than that of typical IT hardware. To capture a large share of the market, telematics service providers need to offer flexible and cost-effective options to service older vehicle models. This flexibility should be reflected in the architecture. Furthermore, security features are typically chosen based on return-on-investment assessments which compare the value of the protected information and the probability for losses with the cost of providing a security mechanism. For future telematics applications these parameters are unknown. Therefore, the architecture should offer different levels of data protection and a migration path between them, so that consumers and businesses can decide which level is most appropriate.

Limited User Interaction Automatically evaluating the appropriateness of data access (from a privacy perspective) is notoriously difficult. Therefore, most systems present a request to the data subject and ask for approval if a request is not covered by the user's preferences already. In a telematics domain this option is unavailable, because the driver usually cannot be interrupted for the mundane task of approving a data request.

5. Architecture

The architecture addresses these challenges through the following key concepts. First, the architecture supports three levels of data protection that can support a wide range of heterogeneous in-car telematics platforms. Second, the software architecture allows a trustworthy telematics service provider (TSP) to assume functions of less capable in-car clients. Specifically, data aggregation before data is released to service providers can minimize the disclosure of privacy sensitive information. Therefore, flexible data aggregation can deploy aggregation modules across the TSP and in-car clients.

In addition, highly flexible privacy policies govern access to the data subject's information. This enables data subjects to specify suitable privacy policies in advance; thus, interruptions while driving can be limited. Section 6 describes policies and their evaluation in more detail.

5.1. Classification of Data Protection Capabilities

Application service providers need to support heterogeneous in-car computing platforms. We distinguish three classes of in-car telematics platforms and describe the level of protection available in each class.

5.1.1. *Thin Client*

A thin client simply collects data from available sensors such as GPS or sensors on the car bus and transmits it to the telematics service provider. At most, applications can specify data collection parameters, such as the types of data and sampling frequencies.

For such clients the (trustworthy) telematics service provider regulates ASP access to collected data according to the data subject's privacy policies. Role based access control (RBAC) is used to control access to resources.[1] Traditional RBAC systems are not adequate for data sharing needs. Once data is sent to a service provider it can be used for any purpose, retained indefinitely, and can be shared with others without the knowledge or permission of the originator of the data. Therefore, it is necessary to take these aspects into account while granting access to data. The Platform for Privacy Preferences (P3P) specification addresses these concerns by mandating that each data request specify purpose, retention, and recipients of data. Use of P3P

[1] In simple terms, a RBAC [6, 10] consists of the following entities: resources, operations, users, roles, and permissions. A resource is associated with a set of operations. Permission defines a set of operations that can be invoked. Permissions are assigned to roles. Users are assigned to roles. Users acquire permissions assigned to a role as a result of membership in it. Users can belong to multiple roles, and permissions can be assigned to multiple roles. Roles can have zero or more users.

policies to obtain data creates an explicit obligation on the part of requestor to adhere to that policy. Actual implementations may extend these basic features by supporting the ability to define hierarchies of roles, and resources (i.e., data items). Further, constraints can be attached to policies to define conditions in which a given policy is applicable.

5.1.2. *Aggregating Clients*

Aggregating Clients offer a sophisticated computing platform that enables service providers to install custom software modules inside the car. We assume that this platform comprises a car PC with a Java runtime environment.

These systems add in-car data aggregation capabilities over the thin client platform. This leads to enhanced privacy in the case of applications that make use of secondary (less private) data derived from the private data. Consider an insurance application which computes premiums based on the actual mileage driven on city roads and highway roads. The mileage data can be computed using data consisting of periodically collected GPS coordinates and odometer readings. Therefore, by sharing aggregate mileage data, user's privacy is better protected.

Both parties, service providers and users, need to establish trust in the programs that compute secondary data. Thus, they must be implemented (or at least audited) by a trusted third party.

5.1.3. *Secure Clients*

A secure client offers additional hardware security features. A chip token could securely store the vehicles private encryption keys. At the highest security level, the system would be able to authenticate the complete software stack and provide a physically tamper-proof execution environment and sensor system. It should be built from the ground-up using secure operating systems [19, 18], secure coprocessors [23] or TCPA compliant system [25].

This platform would offer a high level of protection against adversaries who are in physical possession of the car, such as the owner or a car mechanic. They would find it difficult to retrieve keys for forging identities, and to destroy or modify collected sensor information.

5.2. Flexible Data Aggregation

The architecture supports the construction of applications as a set of collaborating components. Data acquisition components collect information from available sensors on the Car Bus. Aggregation components can further process this information—for example, with feature extrac-

tion or data fusion algorithms. Finally, applications can act dependent on the received data.

The Data Protection Manager offers a generic framework that allows service providers or trusted third-parties to transparently deploy these components throughout the system. Deployment options for components include the in-car, the telematics service provider, and the application service providers computing system.

Components interact through a blackboard-style communication system which allows different entities to interact anonymously through the data they exchange with the blackboard. Blackboard-based architectures provide a suitable paradigm for composing sensor-based applications; they are a common choice for building ubiquitous computing smart spaces. However, blackboards exhibit another key advantage for our privacy protection framework. Every data access passes through the central Data Protection Manager. This provides a locus for enforcing that data accesses comply with the privacy policies. The DPM provides an interface for information producers such as sensors or aggregation applications to publish data on the blackboard. Information consumers access this data from the blackboard through periodic queries or through a subscription/notification mechanism. The blackboard paradigm extends across the network. That is, applications at the TSP or ASP can submit queries to or receive notifications from the in-car blackboard mechanism, for example.

Figure 2 illustrates how data/messages are communicated among applications in this framework. The GPS sensor (not shown in the figure) periodically publishes location data items in the Data Protection Manager. The Mileage Calculator can subscribe to the GPS data and compute the total mileage driven on different types of roads with the help of a road map. The results are again published in the Data Protection Manager, where a Diagnostic Aggregate subscribes to the data. In addition, a Risk Analysis application (not shown in the figure) running on the insurance server remotely subscribes to the aggregated and classified mileage data.

The flow associated with the get data request is as follows:

1. The requester is authenticated by the administrative domain in which DPM is deployed.

2. The Request Processor forwards the request to Privacy Manager to determine whether the requester has permission to access the requested data. The Privacy Manager compares the privacy policy accompanying the data request with user's privacy policies and annotates individual data items in the request with grant/deny decisions.

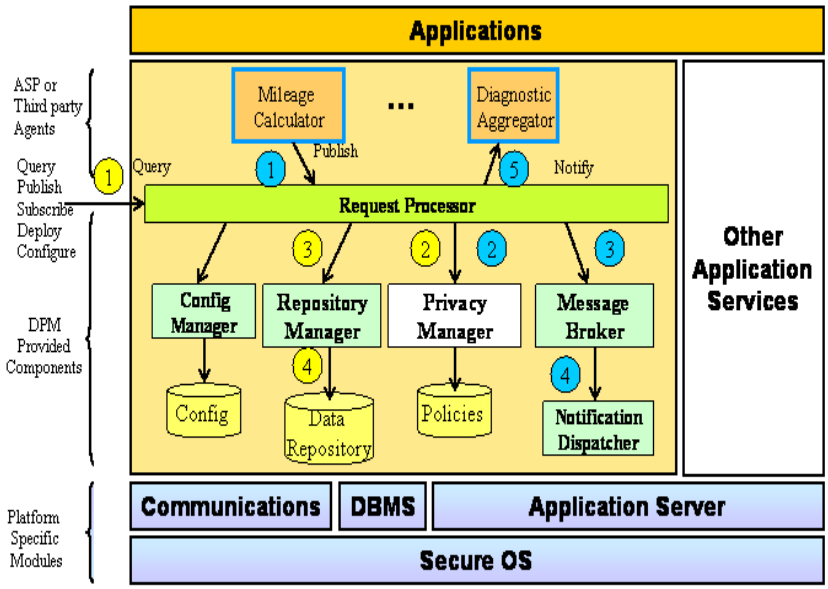

Figure 2. Generic Data Protection Architecture

3. The Request Processor forwards the annotated request to the Repository Manager which obtains the granted data and composes the response.

The flow associated with publish request from the Mileage Calculator is very similar:

1. As request is originating from within DPM, no separate authentication is necessary. The publish request is received by the Request Processor which then forwards the request to Privacy Manager to determine whether the Mileage Calculator is authorized to publish the data.

2. The Privacy Manager checks the user's privacy policies to determine whether the Mileage Calculator could publish the data.

3. The Request Processor forwards the annotated request to the Message Broker which sends out notifications to data subscribers.

5.3. Exploiting Trusted Hardware for Mutual Protection

We expect that aggregation modules will be provided by application service providers or third-party implementers. The user's and ASP's

trust in these components with regard to privacy and data integrity can be increased through the following mechanisms. First certificates can assure the origin of an aggregation component. Second, through the classloader mechanism the Java runtime environment can be configured to execute the component in an individual sandbox that isolates it from other components and all communication channels, except from the blackboard mechanisms that allows communication governed by privacy policies. Finally, openly available source code would allow independent expert inspections, for example to check for covert channels or inaccurate data aggregation.

A secure hardware platform could further strengthen the system against malicious manipulations by adversaries with direct access to the system. For example, a truck driver might be tempted to download and install a software package that misrepresents his driving speed when authorities decide to monitor speeds electronically.[2] Hardware and operating systems that verify the integrity of the complete software stack during the boot process are developed by the TCPA initiative [25]. The runtime environment can then in turn verify the integrity of data collection and aggregation components.

6. Policy Engine

One of the telematics challenges described in Section 4 is to minimize interruptions to the driver. While other privacy approaches assume that an individual can review and decide on some data requests, this framework has to protect privacy without user interactions. To accomplish this task, we need to provide a means for users to specify their privacy preferences in advance. The privacy manager can later authorize requests by matching requesters' privacy policies with users' privacy preferences. To minimize user interactions, the privacy preferences specified by users should cover as many cases as possible, which requires a flexible privacy model that is expressive enough to handle various constraints, general rules, and exceptions.

The key features of the policy engine include expressive policies supporting context-sensitive constraints, a flexible compositional hierarchy, and dynamic grouping. We will briefly explain these features in the remainder of this section.

[2] European trucks are already equipped with tachographs, whose data can be reviewed during traffic stops. These recorders are subject to many manipulations by drivers willing to compromise on safety [1].

6.1. Policies

We developed a privacy policy model that extends the traditional role based access model with usage control; that is, users' privacy preferences can not only specify who can access what data, but also for what purpose, under what constraints, for how long the data can be retained, and to whom it can be distributed. In our privacy model, a rule capturing a user privacy preference comprises the following dimensions. First, the *data subject* refers to the individual or group, whose data this policy protects. Second, the *data view* specifies the set of data items this policy protects. The data view can be an abstract data category (e.g., driving history), a set of data types (e.g., location or contact info), or an object instance value (e.g., my cellar phone number). Third, a *data user* specifies a single requester or a class of requesters to whom this rule applies. In the example scenario the requester would be the insurance company. Fourth, the *data action* defines the modes of data access for the requester (create, modify, query, etc.). Fifth, the *data usage* specifies the privacy usage control, including purpose, recipient, and retention. It extends the P3P usage statement for the telematics domain. Sixth, the *constraint* can specify an additional criterion that needs to be satisfied. In the telematics domain, the constraint could be an event-based constraint (e.g., airbag deployed), a spatial constraint (e.g., within Acadia National Park), or temporal constraints (e.g., between 9am and 6pm). Seventh, the *decision* specifies what authorization decision should be made if this rule applies. It can take the value of allow, deny, notify, or get guardian's consent. Finally, *precedence* corresponds to rule level or priority. It is designed to support two aspects: to allow users to express exceptions that override general privacy rules and to accommodate service providers' need to comply with law enforcement data requests. The following example privacy policy illustrates the rule dimensions:

> Roadside Assistance provider (data user) is allowed to (decision) access (data action) Joe's (data subject) location (data view) for providing roadside assistance (data usage/purpose) when Joe's run out of fuel (constraints)

6.2. Flexible Compositional Hierarchy

Hierarchies are a frequently used concept when modeling the real world. Hierarchies are supported for the data subject, the data user, and the data view dimensions. For example, the data user and data subject hierarchies can correspond to organizational hierarchies, and the data view hierarchies can represent categories or both aggregate data en-

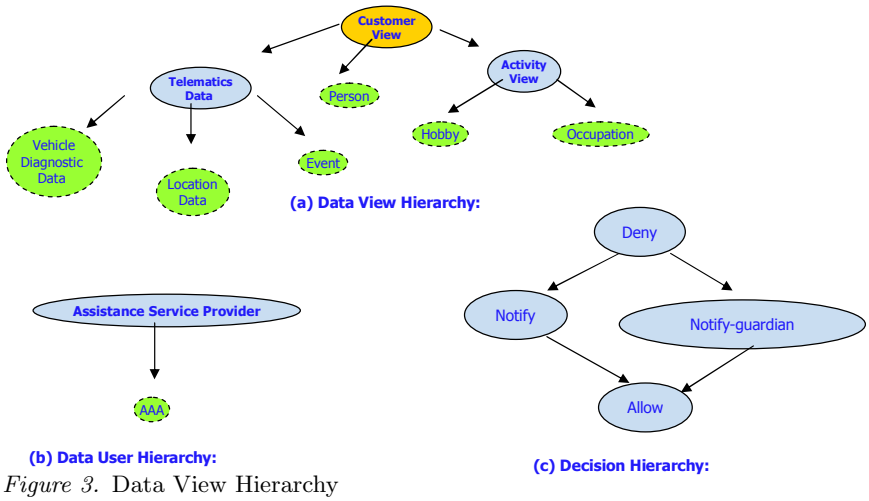

Figure 3. Data View Hierarchy

tities and classification groups. Privacy policies can be specified for aggregate/abstract entities in the higher levels of the hierarchy or on more specific entities in the hierarchy. This creates several advantages. First, it separates the logical privacy policy from concrete deployment information. For example, a privacy officer in an enterprise can specify the privacy policy on a customer data view, and the real enforcement will happen at deployment time when the concrete data items are linked to an abstract view. Second, hierarchies provide a flexible way to specify privacy policy exceptions, which helps resolving possible conflicts. Finally, they improve scalability and management of policies by enabling reuse of groups and data views.

Part (a) in Fig. 3 is an example data view hierarchy, where the customer view comprises the contact info view, the activity view, and the person class. A general company's privacy policy can be defined on the customer view by the companys privacy officer.

6.3. Dynamic Grouping

The proposed privacy policy model also supports the definition of implicit data subject groups, implicit data user groups, and implicit data views. The term *dynamic grouping* refers to dynamically computing the membership in the implicit groups and views at runtime. For example, this enables the following policy: "The location of drivers within 0.5 miles of the power plant will be released to the security office." In this example, "drivers within 0.5 miles of the power plant" may be expressed through a dynamically evaluated group. Furthermore, dynamic grouping allows new users, new data subjects, and new data items that are added to the system to be automatically covered by policy rules. However, dynamic definitions can decrease the performance of the system because the conditions need to be evaluated over the detailed information of a large number of potential members.

6.4. Specificity Checking in the Authorization Engine

The authorization of a request is based on matching the data subject's privacy preferences with the requester's privacy policy. Each request must include the data user, the data subject, the requested data set, the action to be performed on the data items, and the privacy usage controls that the requestor agrees to comply with.

The authorization engine evaluates requests through the following steps.

1. Find all rules R1 that apply to the given data requestor, data subject, and action
2. For each data item in the input data set, choose the most applicable rules from R1 based on rule precedence and /or rule specificity.
3. For each resulting rule in R2 from step 2, match the privacy usage control with request (privacy usage).
4. For each rules from step 3, evaluate constraints
5. For all rules obtained from step 4, make authorization decision by conflict resolution.

The authorization engine employs both rule precedence and rule specificity to determine the applicable rule. Rule evaluation proceeds from rules with higher precedence (priority) to lower precedence. Thus, precedence provides a simple mechanism to specify rules that override others. Only for conflicting rules with the same precedence level, the authorization engine uses specificity checking.

Rule specificity is defined through the combination of specificity on the data view, the data subject, and the data user dimension. A dimension in a rule is less specific if it uses a value closer to the root of the hierarchy. A rule A is more specific than a rule B, if and only if every dimension in rule A is not less specific than the corresponding dimensions in rule B, and at least one dimension in rule A is more specific than the corresponding dimension in rule B.

7. Evaluation

From our experience prototyping sample telematics applications, we found the policies to be sufficiently expressive for most applications. However, the following situations require extensions or different approaches.

Privacy policies typically assume exclusive individual ownership and work towards protecting the private data. We find certain situations in telematics where data ownership is not so clearly definable. Different interpretations of data ownership—shared ownership, split ownership, or exclusive ownership—could have different implications for privacy protections and need to be explored. For example, consider a user driving a rental car. The ownership of data collected from the vehicle could be assigned to either renter, rental fleet owner, or the vehicle manufacturer, depending on the application and data use.

In hindsight, flexible privacy policies should be part of a telematics privacy solution, but are not applicable to all applications. For example, consider emergency applications, which are especially popular, since driving is an inherently dangerous activity. Even though modern cars contain crash sensors that could be used to signal the privacy manager an emergency situation, not every emergency involves a crash. Clearly, a privacy manager that overzealously protects critical information from emergency service providers is undesirable. A potential solution would be to allow emergency service providers unconditional access to location information, but with a strong auditing system in place to identify individuals who abuse their power.

7.1. Authorization Overhead

The authorization engine supports an expressive policy model in which for each request a variety of constraints need to be evaluated. These evaluations need to execute on the limited resources of an in-car telematics system; even when executed on the TSP server, the evaluation function needs to scale to a large number of clients. The following performance measurements shed light on the computational costs associated with these flexible privacy policies.

Using realistic policies from our example scenario, we investigate the effect of three factors on the authorization delay imposed on each request by the policy engine. First, the number of data subjects describes how many vehicles are supported by a TSP. Second, the number of policies specifies how many different rules one data subject has defined for one requester. This effectively tests the policy resolution mechanism. Finally, the number of data items controls how many different data fields a policy governs.

Table I reports the mean authorization delay and the standard deviation from five repeated measurements for every configuration. Each authorization takes approximately 50ms with only a slight increase for larger numbers of data subjects. Additional tests revealed that most of this time is spent in an XML conversion routine (policy and request

Table I. Mean authorization time and standard deviation for an increasing number of data subjects.

Number of Data Subjects	50	100	200	500
Mean Authorization Time (ms)	51.0	51.4	54.0	58.0
Stdev	9.9	12.7	12.0	11.6

formats are based on XML). The process of evaluating a policy is independent from the number of additional policies from other data subjects in the system; thus, only the performance of the database query for the applicable policy is affected. We suspect that other threads in the Java runtime environment (e.g., the garbage collector) cause the variation in execution times.

Figure 4 depicts the effect of the other two factors—that is, the number of policies per data subject and the number of data items controlled by a policy—on the authorization delay. The time to authorize a request increases about linearly with the number of data items and approximately triples for 13 items (compared to 1 item). In addition, a higher number of policies per data subject also significantly increases the authorization time. This overhead is due to specificity checking; the authorization engine needs to compare all policies to select the most applicable one.

8. Conclusions

In this paper, we have identified several challenges that are special in the telematics domain. To address those challenges, we proposed a general Data Protection Framework to enable data sharing in the automotive telematics domain. The key idea is to allow data subjects to have full control over the release of their personal data. We have implemented the core component, the policy engine, to manage users' privacy policies and to authorize the data requests based on policy matching. The policy engine provides a flexible privacy policy model that allows data subjects to specify policies subject to various constraints, including event-based, spatial, and temporal constraints. This flexible policy model will help reduce the need for user interaction while enabling privacy protection. The performance study shows that the authorization overhead keeps stable with increasing numbers of data subjects in the system; it however increases with the number of policies each data subject has. This may be improved with a caching

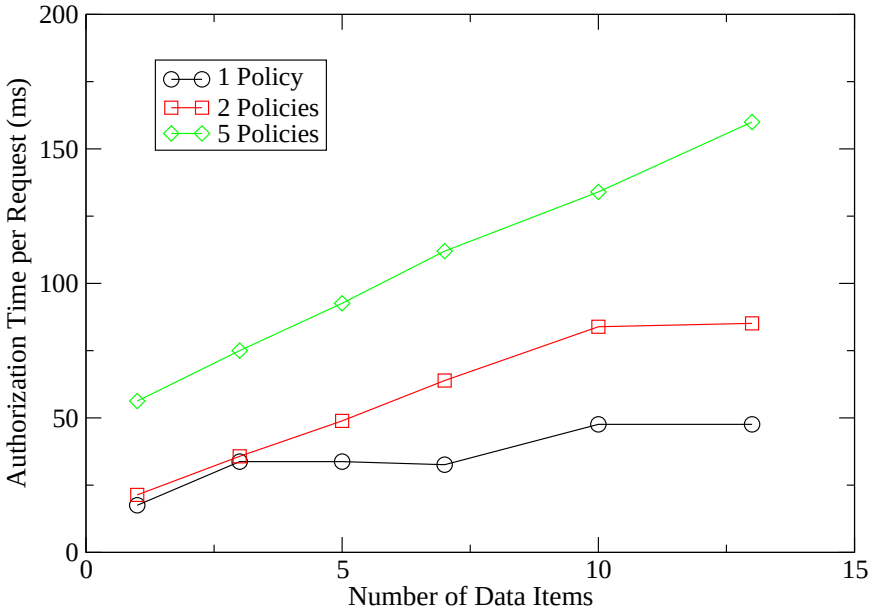

Figure 4. Dependency of mean authorization time on the number of conflicting policies and the number of retrieved data items.

mechanism. We plan to explore such optimization techniques in future work.

Acknowledgements

The authors thank their colleagues: Charles Tressor for challenging us to explore issues of privacy and security for automotive telematics; Marisa Viveros, David Wood, and Paul B. Chou for their support and encouragement.

References

1. Anderson, R.: 2001, *Security Engineering: A Guide to Building Dependable Distributed Systems*, Chapt. Taxi meters, tachographs, and truck speed limiters. Wiley.
2. Arbaugh, W., D. Farber, and J. Smith: 1997, 'A Secure and Reliable Bootstrap Architecture'. In: *IEEE Symposium on Security and Privacy*. pp. 65–71.
3. AT&T: 2002, 'Privacy Minder'. `http://www.research.att.com/projects/p3p/pm`.
4. Bohrer, K., D. Kesdogan, X. Liu, M. Podlaseck, E. Schonberg, M. Singh, and S. Spraragen: 2001a, 'How to Go Shopping On the World Wide Web Without

Having Your Privacy Violated'. In: *4th International Conference on Electronic Commerce Research*.
5. Bohrer, K., X. Liu, D. Kesdogan, E. Schonberg, M. Singh, and S. Spraragen: 2001b, 'Personal Information Management and Distribution'. In: *4th International Conference on Electronic Commerce Research*.
6. Covington, M., W. Long, S. Srinivasan, A. Dey, M. Ahamad, and G. Abowd: 2001, 'Securing Context-Aware Applications Using Environment Roles'. In: *6th ACM Symposium on Access Control Models and Technologies (SACMAT 2001)*.
7. CPExchange: 2002, 'Global standards for privacy-enabled customer data exchange'. `http://www.cpexchange.org/standard/`.
8. Dyer, J., R. Perez, R. Sailer, and L. V. Doorn: 2001, 'Personal Firewalls and Intrusion Detection Systems'. In: *2nd Australian Information Warfare and Security Conference*.
9. Farmer, W., J. Guttman, and V. Swarup: 1996, 'Security for Mobile Agents: Issues and Requirements'. In: *19th National Information Systems Security Conference*, Vol. 2. pp. 591–597.
10. Ferraiolo, D., R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli: 2001, 'Proposed NIST Standard for Role-Based Access Control'. *ACM Transactions on Information and System Security* **4**(3), 224–274.
11. Fischer-Hubner, S. (ed.): 2001, *IT-Security and Privacy - Design and Use of Privacy-Enhancing Security Mechanisms*, LNCS. Springer.
12. Gray, R., G. Cybenko, D. Kotz, and D.Rus: 2001, *Handbook of Agent Technology*, Chapt. Mobile agents: Motivations and State of the Art. AAAI/MIT Press.
13. Gruteser, M. and D. Grunwald: 2003, 'Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking'. In: *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*.
14. Hoffman, D., T. Novak, and M. Peralta: 1999, 'Building Consumer Trust Online'. *Communications of the ACM* **42**(4), 80–85.
15. IBM: 2002, 'Enterprise Privacy Architecture (EPA)'. `http://www.ibm.com/services/security/epa.html`.
16. Karjoth, G., M. Schunter, and M. Waidner: 2002, 'Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data'. In: *2nd Workshop on Privacy Enhancing Technologies*.
17. Langheinrich, M.: 2001, 'Privacy by Design – Principles of Privacy-Aware Ubiquitous Systems'. In: G. Abowd, B. Brumitt, and S. Shafer (eds.): *Ubicomp 2001 Proceedings*, Vol. 2201 of *Lecture Notes in Computer Science*. pp. 273–291.
18. Linux, B.: 2002. `http://www.bastille-linux.org`.
19. NSA: 2002, 'Security-Enhanced Linux'. `http://www.nsa.gov/selinux`.
20. Opplinger, R.: 1999, 'Security issues related to mobile code and agent-based systems'. *Computer Communications* **22**(12), 1165–1170.
21. Pfitzmann, A. and M. Koehntopp: 2000, 'Anonymity, unobservability, and pseudonymity – a proposal for terminology'. In: *Workshop on Design Issues in Anonymity and Unobservability*.
22. Schneier, B. and J. Kelsey: 1999, 'Secure audit logs to support computer forensics'. *ACM Transactions on Information and System Security (TISSEC)* **2**(2), 159–176.
23. Smith, S. and S. Weingart: 1999, 'Building a High-Performance, Programmable Secure Coprocessor'. In: *Computer Networks (Special Issue on Computer Network Security)*. pp. 831–860.

24. Snekkenes, E.: 2001, 'Concepts for personal location privacy policies'. In: *Proceedings of the 3rd ACM conference on Electronic Commerce*. pp. 48–57.
25. TCPA: 2002, 'The Trusted Computing Platform Alliance'. `http://www.trustedcomputing.org/tcpaasp4/index.asp`.
26. W3C: 2002, 'The Platform for Privacy Preferences 1.0 (P3P1.0)'. `http://www.w3.org/TR/P3P`.
27. Wilhelm, U.: 1999, 'A Technical Approach to Privacy based on Mobile Agents Protected by Tamper Resistant Hardware'. Ph.D. thesis, cole Polytechnique Fdrale de Lausanne, Switzerland.