Ensuring Situation-Aware Privacy for Connected Vehicles

Yunxuan Li
yunxuan.li@ipvs.uni-stuttgart.de
University of Stuttgart, IPVS / AS
Stuttgart, Germany

Pascal Hirmer
pascal.hirmer@ipvs.uni-stuttgart.de
University of Stuttgart, IPVS / AS
Stuttgart, Germany

Christoph Stach
christoph.stach@ipvs.uni-stuttgart.de
University of Stuttgart, IPVS / AS
Stuttgart, Germany

Bernhard Mitschang
bernhard.mitschang@ipvs.uni-stuttgart.de
University of Stuttgart, IPVS / AS
Stuttgart, Germany

ABSTRACT

As technology advances in new sensors and software, modern vehicles become increasingly intelligent. To date, connected vehicles can collect, process, and share data with other entities in connected vehicle environments. However, in terms of data collection and exchange, privacy becomes a central issue. It is challenging to preserve privacy in connected vehicle environments when the privacy demands of drivers could change from situation to situation even for the same service. In this paper, we analyze the requirements for a privacy-preserving system in connected vehicle environments with a focus on situation-awareness and safety aspects. Based on the analysis, we propose a novel situation-aware privacy-preserving framework for connected vehicles. Our framework supports individual privacy protections for specific end-point services and situation-aware privacy protections for different circumstances.

CCS CONCEPTS

• **Security and privacy → Privacy protections**; **Domain-specific security and privacy architectures**; *Privacy-preserving protocols*; • **Social and professional topics → Privacy policies**.

KEYWORDS

Connected Vehicle, Privacy-Preserving, Situation-Awareness

ACM Reference Format:
Yunxuan Li, Pascal Hirmer, Christoph Stach, and Bernhard Mitschang. 2022. Ensuring Situation-Aware Privacy for Connected Vehicles. In *Proceedings of the 12th International Conference on the Internet of Things (IoT '22), November 7–10, 2022, Delft, Netherlands.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3567445.3569163

1 INTRODUCTION

The booming development of the Internet of Things (IoT) has also put the Internet of Vehicles (IoV) in the spotlight. As one of the key members of IoT, IoV is enabled by the rapid growth of Connected Vehicles (CVs). According to Coppola and Morisio [2], CVs are vehicles that are equipped with modern applications (apps) and are capable of accessing the internet, collecting and processing real-time data from multiple sources, and interacting with their external environments. With these capabilities, CVs are capable of sharing and exchanging their data with other CVs as well as other traffic participants. These data can be beneficial in achieving autonomous driving or creating safer road environments.

However, data collection and exchange in connected vehicle environments (CVEs) pose privacy challenges. The collected data include location or speed data of the underlying CV as well as audio-visual data of the driver. These data usually contain a considerable amount of sensitive information that can be used to identify the underlying CV or even construct a detailed profile of the driver. Furthermore, the privacy demands of drivers are not always static. It is common that the driver's privacy demands change from app to app or even from situation to situation. For example, drivers may agree to share their source location data with a third-party app when they are not near their homes. However, to protect their home locations from being exposed, they might change their privacy demands to use perturbed location data for sharing, when near their homes.

In our paper, we therefore analyze the requirements for a privacy-preserving system in CVEs from the privacy and safety perspective in Section 2 and propose a novel situation-aware privacy-preserving framework for connected vehicles (SAPP4CV) in Section 3. The SAPP4CV framework allows drivers to create privacy policies for individual services and specific situations. While driving, the created privacy policies will be automatically executed when the situation occurs. In Section 4, we review the related work, and we conclude the paper in Section 5.

2 PRIVACY ANALYSIS

From a driver's perspective, we assume that end-point services in CVEs would always act in a greedy manner. It is expected that these services would try to retrieve every possible source data from a CV, even though they are not necessary for their service functionalities. It can also be assumed that these services would attempt to derive hidden information from the data retrieved. On the other hand, drivers' general demand is to continue utilizing as many functions provided by services (e.g., navigation) as possible. Therefore, it is not enough to acquire the driver's consent for data sharing or collection. The protection of CV data themselves is also necessary.

Primarily, drivers have different privacy demands on protecting CV data for individual apps or specific data sections (e.g., speed, location). For example, a driver may share the source location and

IoT '22, November 7–10, 2022, Delft, Netherlands

ACM ISBN 978-1-4503-9665-3/22/11.
https://doi.org/10.1145/3567445.3569163

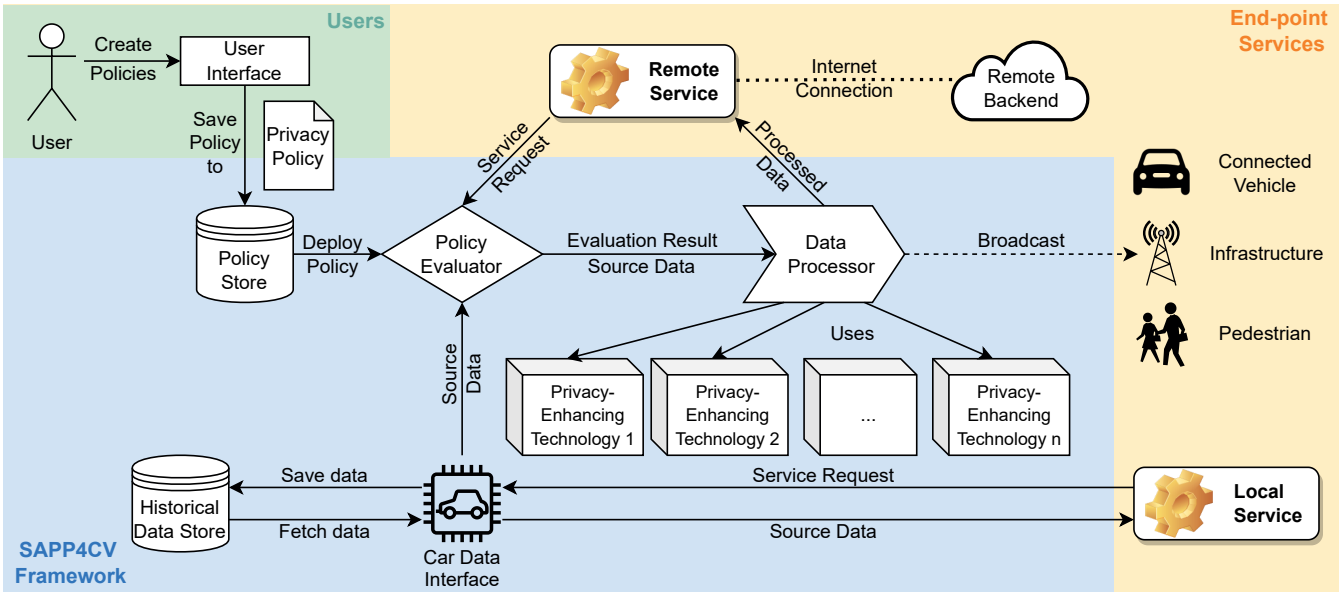

Figure 1: System Architecture of SAPP4CV Framework

speed data with the trusted navigation app, however, the same driver may only share the approximate location data with an untrusted restaurant recommendation app. Moreover, privacy demands for the same app may also vary from situation to situation. As mentioned in Section 1, drivers could have different privacy demands for the same app and the same data section in different situations to protect their exact home location from unwanted disclosures.

From the safety aspect, any inaccurate or corrupted data transmission for safety services (e.g., collision-avoidance) may lead to harm [6] and privacy protections should not be exploited to encourage dangerous driving behaviors. For example, drivers may attempt to hide their speeding behavior from every possible party using privacy demands. To this end, all legal and safety requirements in CVEs regulated by law or traffic regulations should be transformed into policies by qualified law experts. Should expert-defined legal or safety requirements conflict with driver-defined privacy demands, the expert-defined policy should always be preferred. Thus, although the driver could create privacy policies to hide law-breaking or dangerous driving behaviors from every party, these policies will be replaced by expert-defined policies.

Based on the discussion above, we summarize three requirements for a privacy-preserving system in CVEs:

- **R_1 Individuality**: drivers can define individual privacy demands for different end-point services and specific data sections of an end-point service.
- **R_2 Situation-awareness**: drivers can define different privacy demands for end-point services in various situations. While the CV is in motion, only the privacy demands that match the current situation should be executed.
- **R_3 Safety first**: safety and legal requirements that are regulated by law or traffic regulations are transformed into policies by law experts and are always guaranteed.

3 SAPP4CV FRAMEWORK

To ensure safety and provide maximum privacy protection in CVEs, we propose the SAPP4CV framework. With our framework being positioned between the source car data stream and different end-point services, the services can no longer access the source data directly. Instead, services will receive potentially perturbed data that have been processed by our framework based on the specific privacy demands of drivers. In most cases, only the data that are necessary for the desired service functionality are forwarded.

The framework requires an authentication mechanism as the prerequisite to ensure the legitimacy of end-point services. This mechanism could be managed by the government authority or an industry association. After carefully inspecting the service's functionalities, the authentication mechanism issues a signed certificate to the service provider to prove the legitimacy of the service. This certificate also contains the service metadata, such as the access purpose, the service locality (*local* or *remote*), and essential or optional data sections for the core or additional functions.

3.1 Architecture

As depicted in Figure 1, there are two parties that interact with our framework: users (green block) and end-point services (yellow block). Users can be further divided into three user groups: *system admins*, *law experts*, and *general users* who are usually drivers of CVs. The *system admins* are responsible for configuring the system, and the *law experts* are accountable for defining `LAW` and `SAFETY` policies that represent the legal or safety requirements in CVEs. Both user groups should be maintained by the same government authority or the industry association that manages the authentication mechanism. In addition, they cannot access CVs or drivers' data directly. Any change they made to the system can only be deployed in CVs through system updates.

As introduced by Plappert et al. [5], end-point services can be divided into *local* services whose computations are all done locally, and *remote* services that require data transmissions to components external to the CV for its computations. Another widely used communication method in CVEs, broadcasting, is also considered in our framework. Broadcasting can be seen as a special case of the *remote* service, as broadcast data also leaves the CV.

The SAPP4CV Framework itself (blue block) consists of a *User Interface (UI)* where *general users* can interact with our framework and a *CarData Interface (CDI)* where the source data stream originates. It also contains two data stores: a *Policy Store (PS)*, where expert- and driver-defined privacy policies are held, and a *Historical Data Store (HDS)*, where unmodified source data are stored for a certain period (e.g., a week). When the defined period is passed, the newly arriving data will overwrite the old data.

The core components of the framework are the *Policy Evaluator (PE)* and *Data Processor (DP)*. The PE component is responsible for evaluating the source data and the metadata of the requesting service against all deployed privacy policies. Based on the evaluation result, DP removes the undesired sensitive information from the source data using different Privacy-Enhancing Technologies (PETs). The *system admins* maintain the list of all available PETs for the framework. Generally, our framework has no restrictions regarding the selection criteria of PETs. However, PET's properties, such as the privacy protection degree, run-time, and resource cost, should be considered. To use selected PETs in a modular manner, certain adaptations might also be necessary.

3.2 Privacy Policy

Each driver has their individual privacy demands. However, all informal privacy demands must be transformed into formalized, machine-readable privacy policies to be deployable in the PE. Listing 1 depicts a sample privacy policy in JavaScript Object Notation (JSON). The privacy policy P_{home} is defined to protect the driver's exact home location from being disclosed to *AppX*.

As shown in Listing 1, each privacy policy contains a unique `policyID` that can be used to identify the policy and five other fields: `meta`, `priority`, `conditions`, `situation`, and `actions`. The `meta` field records general information about a policy, such as the policy's *name*, *lifetime* (i.e., the effective period of the policy), and *creator*.

Following the `meta` field is the `priority` value of the policy. This value indicates the precedence of the policy. The smaller the priority value is, the higher the policy priority. In our framework, the highest two priority levels (value 0 and 1) are preserved for expert-defined `LAW` and `SAFETY` policies, respectively. Any driver-defined policies can only have a priority value starting from 2.

The `conditions` field records the general constraints of the underlying policy, whereas the `situation` field consists of a set of constraints that describe a particular situation. The connections between constraints within the `situation` field are conjunction (`AND`). It can be extended to include other logic connectors, such as disjunction (`OR`). The `situation` field of P_{home} illustrates the scenario where the driver is within a 2km range of the home location (specified by the latitude and longitude). Note that this field is the key to situation-aware privacy protection ($\mathbf{R_2}$), as it enables drivers to specify various situations when creating privacy policies.

Listing 1: Sample Privacy Policy P_{home}.

```
{
  "policyID": "p-a93fa6c5566c",
  "meta": {
    "name": "location policy 1",
    "lifetime": "ALWAYS",
    "creator": "user-pjeXDbRozh",
    ... },
  "priority": 2,
  "conditions": null,
  "situation": {
    "location": {
      "latitude": 48.745340,
      "longitude": 9.106757},
    "range": "2km",
  },
  "actions": [
  {
    "targetServices": [ "AppX" ],
    "targetDataSections":
      [ "LOCATION" ],
    "PET": "N_MIX",
    "parameters": [ "500", "4" ]
  }]
}
```

policy metadata

situation definition

desired privacy-preserving actions

The last field, the `actions` field, contains a set of privacy *action*s that should be executed when the policy is evaluated to *true*. Each *action* is represented by four attributes. The first two attributes indicate the targeted service and its corresponding data section(s). These two attributes act as sub-constraints within each *action* and support drivers to define individual privacy demands ($\mathbf{R_1}$) for specific end-point services or data sections. The other two attributes, *PET* and *parameters*, are used to specify the concrete PET as well as its parameter(s). Altogether, P_{home} describes that when the driver is near home, the PET *N-MIX* by Wightman et al. [7], which randomizes the location data within a 500m range should apply to the location data before being shared with *AppX* to protect the exact home location from being disclosed.

3.3 Data Flow

The data flow in our framework can be divided into three phases: (i) policy creation, (ii) in motion, and (iii) service request.

3.3.1 Policy Creation. Different kinds of privacy demands could be created with high flexibility through the UI depicted in Figure 1. When drivers indicate in the UI that they have finished the policy creation phase, the UI transforms the informal privacy demands into formalized privacy policies. After that, the potential conflicts between policies with the same priority are examined. If any conflict is detected, drivers are informed and guided to resolve the conflict. When there is no conflict remaining, the created policies are stored in the PS and then deployed in the PE.

3.3.2 In Motion. When the CV starts moving, a constant source data stream is forwarded to the framework through the CDI. For

every source data point our framework receives, a copy of it is saved in the HDS in CV. The HDS is needed if an end-point service requires historical data or in situations, such as post-accident, where unmodified source data have to be presented to law enforcement officers or investigation parties. For this purpose, HDS should guarantee that the data saved in it is not modifiable. As the source data usually contain sensitive information, the HDS should also be implemented in a way that only eligible parties (e.g., law enforcement officers) and our framework can access it directly.

3.3.3 Service Request. The data processing for service requests from different kinds of end-point services (*local*, *remote*, and broadcasting) is slightly different. For both *local* and *remote* services, our framework only accepts service requests from services with valid certificates. Per definition, no data leaves the CV during the computation of *local* services. Thus, our framework provides them with direct data access to requested source data upon service request. Note that the data access is only granted for the necessary data sections stated in the service metadata. Other data sections and policies stored in PS are inaccessible to the requesting *local* service.

When the service request from a legitimate *remote* service arrives, the service metadata included in the certificate is forwarded to the PE, as well as the requested source data. The PE evaluates the requested data together with the service metadata against all deployed privacy policies. During the evaluation, only the policies that match the current service request and situation are considered. If multiple policies match the current situation, only the policy that has the highest priority will be evaluated to *true*. With the priority ordering introduced in Section 3.2, the framework ensures that any driver-defined policies that are incompliant with law-enforced legal or safety requirements will be ignored and replaced by expert-defined policies during the evaluation. Thus, the safety first requirement ($\mathbf{R_3}$) is always guaranteed.

The result of the policy evaluation contains a set of *action*s from policies that are evaluated to *true*. This result is forwarded to the DP together with the requested data. Assuming that service *AppX* is requesting the data point D_{req} := <*ID*: 59*af*36, *emission*:96.83, *speed*:51.16, *location*:(48.746, 9.112)> and P_{home} is evaluated to *true* in the PE. Consequently, the privacy *action*s from P_{home} and D_{req} are forward to the DP. In the DP, the desired PET *N-MIX* is applied to the location data section (represented by latitude and longitude) of D_{req}. This results in a perturbed data point, D_{ptb} := <*ID*:59*af*36, *emission*: 96.83, *speed*:51.16, *location*:(**48.748, 9.118**)>. In the metadata, it is specified that the essential data sections for *AppX* are *ID* and *location*. Thus, other data sections are filtered out by the DP, which results in a processed data point D_{pcd} := <*ID*:59*af*36, *location*:(48.748, 9.118)>. Finally, the processed data D_{pcd} is transmitted back to the requesting service.

The data processing for broadcasting is basically the same as for *remote* services, where the service metadata used in PE is substituted with the broadcast channel information, as the receivers of broadcast data are usually unknown before data are shared.

4 RELATED WORK

Duri et al. [3] propose a Data Protection framework that allows users to specify relatively complex privacy policies regarding what data can be shared with what parties. The authorization for data release is conducted by matching the individual's privacy policies with the data requests. In the same vein, Plappert et al. [5] propose a privacy-aware data access system for automotive applications that enables users to control third-party access to their personal data through privacy policies and privacy-enhancing technologies. This system allows users to set allowances for particular services, specific functions, or individual data types. However, both studies have paid no attention to situation-aware privacy protection.

Ghane et al. [4] propose another system called Differentially Private Data Stream (DPDS), designed for a distributed edge computing system where the edge controller is untrusted. In the DPDS system, vehicles subscribed to the same controller form a group. The group leader is responsible for collecting data from other vehicles, aggregating the data, and forwarding the aggregated data to the controller. However, the DPDS system leaves all three requirements described in Section 2 untouched.

The Block4Forensic framework introduced by Cebe et al. [1] provides a privacy-aware framework for post-accident analysis. This approach periodically records the hash of vehicle data on the Blockchain. Should a car accident occur, the source data, which match the published hash, are disclosed to investigation parties. Due to its focus on post-accident analysis, this paper does not consider individual and situation-aware privacy protections or privacy protection for other scenarios.

5 CONCLUSION

Preserving privacy under situation awareness and safety concerns in CVEs is challenging. To cope with this challenge, we propose the SAPP4CV framework. The privacy policy format introduced in the framework allows drivers to create individual privacy policies for specific end-point services or data sections ($\mathbf{R_1}$) and situational privacy policies for various situations ($\mathbf{R_2}$). Our framework also values the safety of all traffic participants ($\mathbf{R_3}$). This is assured by only allowing qualified law experts to create `LAW` and `SAFETY` policies and grant them higher priority.

REFERENCES

[1] Mumin Cebe, Enes Erdin, Kemal Akkaya, Hidayet Aksu, and Selcuk Uluagac. 2018. Block4Forensic: An Integrated Lightweight Blockchain Framework for Forensics Applications of Connected Vehicles. *IEEE Communications Magazine* 56, 10 (2018), 50–57.

[2] Riccardo Coppola and Maurizio Morisio. 2016. Connected Car: Technologies, Issues, Future Trends. *Comput. Surveys* 49, 3 (2016), 46:1–46:36.

[3] Sastry Duri, Marco Gruteser, Xuan Liu, Paul Moskowitz, Ronald Perez, Moninder Singh, and Jung-Mu Tang. 2002. Framework for Security and Privacy in Automotive Telematics. In *Proceedings of the 2nd International Workshop on Mobile Commerce (WMC '02)*. 25–32.

[4] Soheila Ghane, Alireza Jolfaei, Lars Kulik, Kotagiri Ramamohanarao, and Deepak Puthal. 2021. Preserving Privacy in the Internet of Connected Vehicles. *IEEE Transactions on Intelligent Transportation Systems* 22, 8 (2021), 5018–5027.

[5] Christian Plappert, Daniel Zelle, Christoph Krauß, Benjamin Lange, S Mauthöfer, Jonas Walter, Bettina Abendroth, Rasmus Robrahn, Thilo von Pape, and Hendrik Decke. 2017. A Privacy-aware Data Access System for Automotive Applications. In *Proceedings of the 15th ESCAR Embedded Security in Cars Conference (escar Europe '17)*. 1–11.

[6] Joshua E. Siegel, Dylan C. Erb, and Sanjay E. Sarma. 2018. A Survey of the Connected Vehicle Landscape—Architectures, Enabling Technologies, Applications, and Development Areas. *IEEE Transactions on Intelligent Transportation Systems* 19, 8 (2018), 2391–2406.

[7] Pedro Wightman, Winston Coronell, Daladier Jabba, Miguel Jimeno, and Miguel Labrador. 2011. Evaluation of Location Obfuscation techniques for privacy in location based information systems. In *Proceedings of the 2011 IEEE Third Latin-American Conference on Communications (LatinCOM '11)*. 1–6.