An Authorization Framework for Cooperative Intelligent Transport Systems

Sowmya Ravidas*, Priyanka Karkhanis*, Yanja Dajsuren, and Nicola Zannone

Eindhoven University of Technology,
Eindhoven, The Netherlands
{s.ravidas, p.d.karkhanis, y.dajsuren, n.zannone}@tue.nl

Abstract. Cooperative Intelligent Transport Systems (C-ITS) aims to enhance the existing transportation infrastructure through the use of sensing capabilities and advanced communication technologies. While improving the safety, efficiency and comfort of driving, C-ITS introduces several security and privacy challenges. Among them, a main challenge is the protection of sensitive information and resources gathered and exchanged within C-ITS. Although several authorization frameworks have been proposed over the years, they are unsuitable to deal with the demands of C-ITS. In this paper, we present an authorization framework that addresses the challenges characterizing the C-ITS domain. Our framework leverages principles of both policy-based and token-based architectures to deal with the dynamicity of C-ITS while reducing the overhead introduced by the authorization process. We demonstrate our framework using typical use case scenarios from the C-ITS domain on location tracking.

1 Introduction

Intelligent Transport Systems (ITS) are "*systems in which information and communication technologies are applied in the field of road transport, including infrastructure, vehicles and users, in traffic management and mobility management, as well as for interfaces with other modes of transport*" [4]. Cooperative Intelligent Transport Systems (C-ITS) aims to improve the quality of ITS through the use of sensing capabilities and advanced information and communication technologies [13].

Significant developments have taken place over the past few years in the C-ITS domain. Several initiatives and projects have been established all over the world (e.g., DITCM [34], CONVERGE [1], US-ITS [2]) to enable the development of a cooperative architecture to support the communication of vehicles with the transport infrastructure, service providers and other vehicles. While these initiatives provide a foundation for the design and development of C-ITS, their results can be deployed at a large scale only if the developed infrastructure and services meet the requirements posed by the C-ITS domain, including scalability, performance and security.

Security is particularly challenging to achieve within C-ITS as it encompasses multiple systems, such as automotive systems, road infrastructure, services and applications, and requires addressing attackers with various motivations and levels of skills, and diversity of threats and countermeasures [22]. Among the several security concerns, the

[*] Equal contribution to this manuscript.

protection of information gathered and shared within C-ITS is of utmost importance to enable its deployment at a large scale. Typically, sensitive data are protected through the adoption of authorization mechanisms that guarantee that only authorized parties can gain access to the data. While several authorization frameworks have been proposed to address authorization concerns in several application domains, there has been very little attention towards authorization in the C-ITS domain. Given the critical and dynamic nature of C-ITS, authorization mechanisms should not affect the functioning and performance of the system as well as provide fine-grained protection of sensitive information and resources. Specifically, an authorization framework for C-ITS should allow the specification and evaluation of context-aware policies to deal with the dynamicity of C-ITS, minimizing the overhead of the authorization process, and guaranteeing its reliability [30].

In this paper, we present an authorization framework that addresses the unique challenges of the C-ITS domain. The design of our framework leverages principles of both policy-based [26] and token-based [3,11] architectures to deal with the dynamicity of C-ITS while minimizing the overhead introduced by the authorization process. Specifically, we decouple the evaluation of policies from their enforcement. Our solution encompasses a policy-based authorization server that is used off-line to generate tokens encoding user permissions based on the policies provided by the resource owner. Tokens are then locally validated by the resource server at request time to determine whether access should be granted. While this decoupling allows minimizing the overhead introduced by the authorization process at request time, relying only an off-line policy evaluation does not make it possible to account for access constraints based on the run-time environment. To this end, we devise authorization tokens that encompass constraints to be verified at request time. For the design of our authorization framework, we leverage a C-ITS reference architecture as a baseline. The adoption of a C-ITS reference architecture helps identifying the C-ITS systems involved in the authorization process and, thus, facilitate the realization and integration of our authorization framework within existing C-ITS deployment sites.

The paper is organized as follows. Section 2 provides background on the C-ITS domain and authorization. Section 3 discusses related work. Section 4 introduces the C-ITS reference architecture used in this work and Section 5 proposes an authorization framework conforming to such an architecture. Section 6 presents an application of the proposed authorization framework in the context of location tracking services. Finally, Section 7 discusses design choices, and Section 8 concludes the paper and presents directions for future work.

2 Background

Cooperative Intelligent Transport System (C-ITS) is emerging to improve the quality of existing ITS infrastructure by making transportation more safe and economical by combining data from vehicles and other sensors [13]. In particular, C-ITS applies information and communication technologies to the field of road transportation, including infrastructures, vehicles and users, for efficient traffic management and mobility management. While bringing several advantages to individuals, industry and society, the

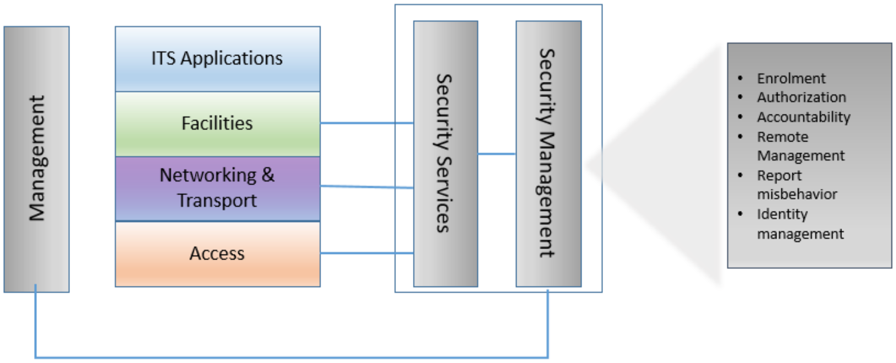


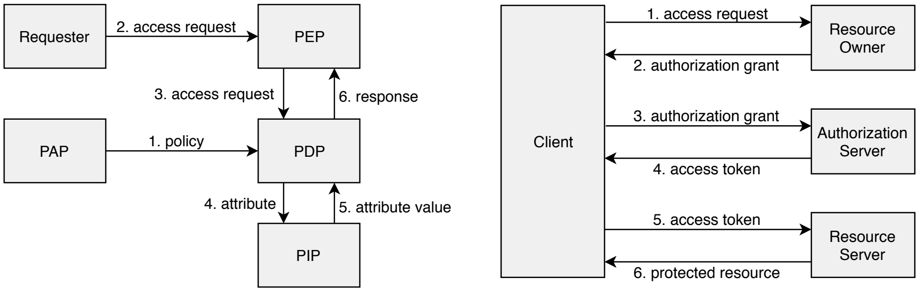
Fig. 1: ETSI security services within C-ITS

deployment of C-ITS also opens new challenges. To be adopted at a large scale, C-ITS should facilitate the addition and management of a large range of heterogeneous devices, allow for transferring data at high rate and provide real-time response.

On top of these issues, security is a critical success factor for the adoption of C-ITS. This has spurred several efforts from both industry and academia to enable and improve security within the C-ITS domain. Standardization bodies such as the European Telecommunication Standards Institute (ETSI), have defined guidelines towards the design and development of secure services for C-ITS [5]. Specifically, ETSI has identified key secure functionalities to be provided by C-ITS, including identification, authentication, authorization and enrolment. These functionalities are positioned within security management services, as illustrated in Figure 1.

Authorization services (the focus of this work) aim at the protection of sensitive information exchanged within C-ITS (e.g., location data). A typical solution to protect sensitive information and resources is through the adoption of access control solutions that guarantee that only authorized parties can gain access. Access is regulated using policies that specify which actions an entity can perform on a certain object. In the remainder of the section, we provide an overview of the reference architectures commonly adopted for the design of authorization frameworks and discuss the main challenges to be addressed in the design of an authorization framework tailored to C-ITS.

Authorization Reference Architectures: Several architectures have been proposed for the design of authorization mechanisms. Two widely adopted architectures are the *policy-based* and *token-based* architectures. Policy-based architectures can be exemplified by the reference architecture proposed by XACML [26], the de facto standard for the specification and enforcement of attribute-based access control policies. This architecture comprises four main components: *Policy Enforcement Point* (PEP), which provides an interface with the system and is responsible for enforcing access decisions; *Policy Decision Point* (PDP), which evaluates access requests against access control policies and determines whether access should be granted or denied; *Policy Administration Point* (PAP), which acts as a policy repository and offers facilities for policy management; *Policy Information Point* (PIP), which denotes the source of information (e.g., context



(a) Policy-based Architecture

(b) Token-based Architecture

Fig. 2: Authorization Reference Architectures

information) needed for policy evaluation. Figure 2a shows the interaction between these components. The PAP makes the policies available to the PDP (1). Upon receiving an access request (2), the PEP forwards the request to the PDP (3), which evaluates the request against the policies fetched from the PAP. If additional information is required for policy evaluation, the PDP queries the PIP (4,5). The PDP evaluates the request against the policies and returns a response specifying the access decision to the PEP (6), which enforces the decision.

Authorization mechanisms adopting a policy-based architecture typically provide a single, centralized point for the evaluation and enforcement of access control policies [18]. This solution may not be suitable when resources are distributed across different nodes, which is a typical situation in C-ITS. The last years have seen the emergence of token-based architectures as an alternative to policy-based architectures to deal with the needs of open and decentralized systems. Various standards have defined reference token-based architectures and authorization protocols [3,11]. Although these architectures and protocols vary in the way tokens are generated and in the flow of the authorization process, they share the same underlying principles. As an example, Figure 2b presents the OAuth protocol [11], in which a client application first obtains a token encoding its permissions from an authorization server and subsequently uses it to access a given resource.

Challenges: An authorization framework should not affect the functioning of C-ITS. Given the constraints imposed by these systems, the design of an authorization framework for C-ITS presents a number of challenges. These challenges will be used to identify the main concepts and design principles that should be considered when developing an authorization framework for C-ITS.

- *Dynamicity:* C-ITS are complex and dynamic systems in which an increasing number of entities (e.g., vehicles, RSUs) are connected and in which network topology and connectivity changes over time. To handle the dynamic nature of C-ITS, an authorization framework should support the specification and evaluation of context-aware access control policies that impose conditions on the C-ITS ecosystem such as access time and location.
- *Management:* The dynamicity of the C-ITS ecosystem can also affect policy management. In such systems, the resources of an entities can be stored and managed by different administrative domains interacting together. Therefore, an authorization

framework should be able to support the management of access control policies for devices and resources across multiple domains.

- *Automation:* A main characteristic of C-ITS is collaboration, which is achieved through interactions between entities involved in the C-ITS (e.g., vehicles, RSUs). These interactions involve the sharing of real-time safety critical information. Hence, C-ITS systems require a high level of automation, possibly without any user involvement. This need for automation also reflects in the authorization process.
- *Performance:* C-ITS are critical systems in which delays can have serious consequences and even result in human loss. Therefore, services deployed within the C-ITS should not introduce latency both in terms of computation and communication. This constraint extends to the authorization process. In particular, the authorization process should not inhibit performance with significant overhead, which violates the timing constraints imposed by the C-ITS.
- *Reliability:* The critical nature of C-ITS also poses high demands for business continuity, even in cases of system failures. On the other hand, the highly sensitive information gathered and exchanged within the C-ITS requires protection and its disclosure to unauthorized parties should be prevented. Meeting these (apparently conflicting) demands requires the authorization process to be reliable. Even in cases where failures or loss of connectivity occur, the authorization framework must still be operational.

3 Related Work

Several security services for ITS have been proposed in the literature [6,22,30,31,36,37]. However, they typically focus either on authentication alone or on the protection of communication using cryptographic techniques. To the best of our knowledge, only a few authorization frameworks have been proposed in the ITS context. Salonikias et al. [33] propose a policy-based authorization mechanism tailored to vehicular infrastructures based on fog computing. This mechanism comprises multiple PDPs and PEPs located at the edge, while a single PAP (which also encompasses a PIP) deployed in the cloud is responsible to maintain and propagate access control policies to the PDPs. Gupta et al. [15] present an authorization framework for Internet of Vehicles. This framework proposes the deployment of authorization components (PEP, PDP, PAP, PIP) at different layers – object, virtual object and cloud level layer – to deal with different types of interactions. Dorri et al. [12] propose an authorization framework for vehicular networks based on blockchain. In this framework, interactions between vehicles are stored in the blockchain as transactions, which are verified by powerful nodes acting as miners. Albouq et al. [8] propose a policy-based framework for ITS infrastructures based on fog computing. Service providers deploy their services in fog nodes and vehicles can connect to these nodes through RSUs acting as edge network units. RSUs rely on the publish-subscribe paradigm to enable vehicles to subscribe to the services deployed in fog nodes. In this respect, the authorization framework resides within RSUs to control which services can be published whereas vehicles can subscribe to any (allowed) service. Riabi et al. [32] propose the use of a distributed hash table (DHT) to handle authorization within ITS. Resources are stored in fog nodes and each fog node maintains a DHT specifying the mapping between fog nodes and the Access Control List (ACL) maintained by them.

	Ref. Arch.	**Dynamicity**	**Management**	**Automation**	**Performance**	**Reliability**
Salonikias et al. [33]	policy-based	●	●	●	◑	◑
Gupta et al. [15]	policy-based	◑	●	●	◑	◑
Dorri et al. [12]	blockchain	○	○	●	○	◑
Albouq et al. [8]	policy-based	◑	●	●	◑	◑
Riabi et al. [32]	policy-based	○	◑	●	◑	○

Table 1: Analysis of existing authorization frameworks for ITS. Symbol ● denotes that a challenge is *addressed*, ◑ that it is *partially addressed*, and ○ that its is *not addressed*

Upon receiving an access request for a given resources, fog nodes use the DHT to identify the node handling the requested resource and forward the request to such a node, which makes an access decision by evaluating the request against its ACL.

Discussion: Despite the number of authorization mechanisms for ITS proposed by both industry and academia, existing authorization mechanisms are inadequate to deal with the open and dynamic nature of C-ITS systems. Table 1 presents an analysis of existing authorization frameworks for ITS with respect to the challenges discussed in Section 2.

An authorization framework for C-ITS should cope with the dynamic nature of ITS. While some frameworks (e.g., [15,33]) support the definition of context constraints in policies and their evaluation, many frameworks (e.g., [8,12,32]) do not, thereby not addressing this challenge. Nonetheless, most frameworks [8,15,33] provide a single point for policy administration, thus facilitating policy administration. An exception is the frameworks in [12], in which policies reside within vehicles. It is worth noting that the framework in [32] allows resource owners to deploy their policies to a single fog node and uses a DHT to identify which nodes should evaluate a request for a given resource. However, the DHT stored in each node has to be updated whenever an ACL is modified. The automation of the authorization process is satisfied by all frameworks as they do not require user involvement in the authorization process.

To be effective in C-ITS, an authorization framework should not introduce significant overhead and latency and, in general, should not affect the overall performance of the C-ITS [30]. None of the existing frameworks fully satisfies this requirement. Existing authorization frameworks typically perform policy evaluation upon receiving an access request, thus delaying service provision. In addition, some frameworks require additional communication to retrieve the context information needed for policy evaluation [33], or rely on technology that is computational expensive like blockchain [12]. Other frameworks [8,15,32] adopt a centralized architecture where all authorization components reside within the cloud, a fog node or the vehicle. However, assuming that all (context) information needed for policy evaluation is available from a single source limits the constraints on the context that can be verified.

Existing authorization frameworks partially address the reliability of the authorization process by placing authorization components within the cloud [15,33], which typically provides recovery measures to ensure business continuity. In addition, Salonikias and colleagues envision redundancy for those components deployed at the edge. Similarly, the frameworks in [12] and in [8] can theoretically ensure the reliability of authorization components by replicating them in blockchain nodes and RSUs, respectively. However, for both frameworks, scenarios of node failure or loss of connectivity are not analyzed. In [32], the request can be sent to any fog node, which forwards it to the node that has the requested resource. However, ACLs are not replicated among nodes, leading to reliability issues in case of connectivity loss or node failure.

In summary, existing authorization frameworks fail to fully address all challenges posed by C-ITS. A main drawback is given by latency due to the choice of a policy-based architecture for their design. In this work, we present an authorization framework for C-ITS that adopt principles underlying the token-based architecture as a baseline for its design (Section 5). This architecture provides a foundation to deal with the dynamicity and performance constraints typical of C-ITS scenarios.

4 C-ITS Reference Architecture

For the design of our authorization framework for C-ITS and its realization and integration in existing C-ITS sites, we adopt a C-ITS reference architecture as a baseline for our design. A reference architecture is typically used to facilitate communication and cooperation between different stakeholders during the design and development of complex systems. A reference architecture for the C-ITS domain addresses not only demands in the software/system engineering field, but also in traffic engineering, civil engineering, information technology, etc. Moreover, its design should account not only for new systems but also taking into account the infrastructure and systems already in place.

In the recent years, several C-ITS reference architectures have been proposed to address the interdisciplinary concerns and to enable the large scale deployment of region or nation wide C-ITS services. In this work, we adopt the C-ITS reference architecture proposed in the C-MobILE project (`http://c-mobile-project.eu`) as a baseline for the design of our authorization framework. The C-MobILE reference architecture provides a baseline for the design of a C-ITS infrastructure mainly targeting traffic related concerns [9,20]. The C-MobILE reference architecture is based on the generalization of existing C-ITS architectures while addressing the main concerns of the C-ITS stakeholders.

The C-MobILE reference architecture categorizes C-ITS systems into five main types based on the functionalities they provide, as illustrated in Figure 3. Below we present a brief description of the main systems and refer to [9] for details:

- *Support system* consists of systems supporting the governance and management of C-ITS services. Support systems influence all other systems of the C-ITS.
- *Central system* comprises systems that support connected vehicles and roadside units by capturing data from vehicles and roadside units, and providing such data to C-ITS applications. Central systems can be aggregated together or can be geographically or functionally distributed.
- *Roadside system* consists of systems forming the physical road infrastructure such as roadside units, traffic light controllers, and cameras.
- *Vehicle system* comprises systems integrated within vehicles such as a Vehicle On-Board Unit (V-OBU).
- *Traveler/VRU system* consists of personal devices, typically a smart phone or personal navigation device used by a traveler or Vulnerable Road User (VRU).

Security services are provided by the support system. Below we describe its sub-systems.

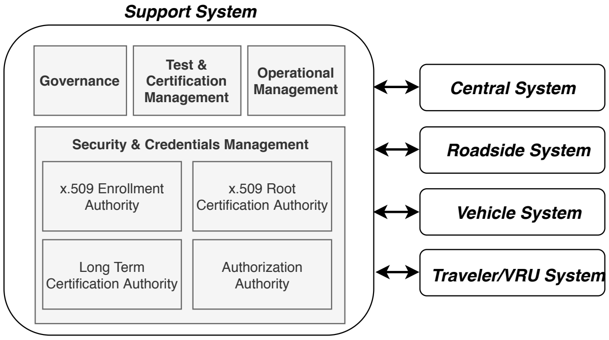

Support System
Governance
Test & Certification Management
Operational Management
Security & Credentials Management
x.509 Enrollment Authority
x.509 Root Certification Authority
Long Term Certification Authority
Authorization Authority
Central System
Roadside System
Vehicle System
Traveler/VRU System

Fig. 3: C-ITS Reference Architecture

- *Governance* comprises systems and entities that are responsible for the functioning and security of the C-ITS.
- *Operational Management* comprises systems enabling operational processes such as fault, performance and configuration management of C-ITS systems.
- *Test and Certification Management* supports the registration and management of tested and certified communication systems for ITS (safety) applications.
- *Security and Credentials Management* provides a high-level representation of the systems that enable trusted communications between mobile devices, roadside devices and centres, and protect data from unauthorized access. A sub-systems is the *Authorization Authority*, which is in charge of issuing authorization tickets to ITS entities.

In the next section, we present the design of our authorization framework and show how its components are mapped to the systems of the C-ITS reference architecture. This mapping will help understand the external interfaces, high level functional capabilities of the authorization components within the C-ITS architecture.

5 Authorization Framework

Existing authorization frameworks are usually based on either a policy-based or a token-based architecture. As discussed in Section 2, policy-based frameworks often introduce delays that cannot be tolerated by the C-ITS. While existing token-based frameworks address this issue by limiting the operations to be performed at run-time to token validation, they usually require user involvement to determine whether access should be granted, thus providing no automation of the authorization process. In this work, we propose a *hybrid* authorization framework that leverages the advantages of both these architectures. In particular, we divide the authorization process into two main stages: an *off-line* process in which tokens are automatically generated based on policies (without any user involvement) and a *run-time* process in which tokens are validated. Such an approach provides the flexibility and performance necessary to deal with the dynamic and critical nature of C-ITS while providing a high degree of automation. In the remainder of the section, we present the main components of the framework along with the authorization process.

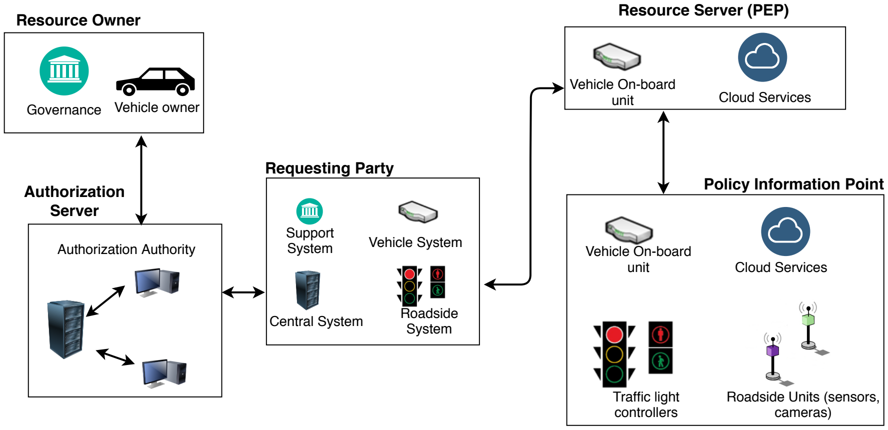


Fig. 4: Authorization framework and mapping of its components to C-ITS systems

Authorization Components: The authorization framework encompasses the following entities and components:

- *Resource Owner* is the user or legal entity that controls a given resource.
- *Resource Server* is the component hosting the resource on behalf of resource owner.
- *Authorization Server* is the component that protects resources hosted on a resource server on behalf of the resource owner. The authorization server generates tokens based on access requirements specified by the resource owner, thus acting as the PDP.
- *Policy Information Point* denotes the source of context information.
- *Requesting Party* is a user or a legal entity that uses a client application to access resources.

Figure 4 shows these components along with their interactions. It also provides their mapping to the systems of the C-ITS reference architecture in Figure 3. This mapping identifies which C-ITS systems can play a role in the authorization process.

Authorization process: The authorization process supported by our hybrid authorization framework is performed in two stages. First, the authorization server evaluates the policies off-line and generates an authorization token asserting the permissions of the requesting party. Then, when requesting access to a resource, the requesting party provides the token along with request to the resource server, which validates the token and verifies additional constraints on the context (if any). The procedures for off-line token generation and run-time token validation are represented in Figure 5.

We assume that the resource owner stores her resources in a resource server. Moreover, she has provided the authorization server with access control policies defining who can access her resources. It is worth noting tha resources can be under the control of multiple entities or negotiation between entities may be necessary to determine how resources can be used and with whom they can be shared. In this settings, data sharing agreements [19,25] should be established between the involved parties to determine provisions concerning access and dissemination. How data sharing agreements and collaborative policies [10,23] can be defined is out of the scope of this work and here we

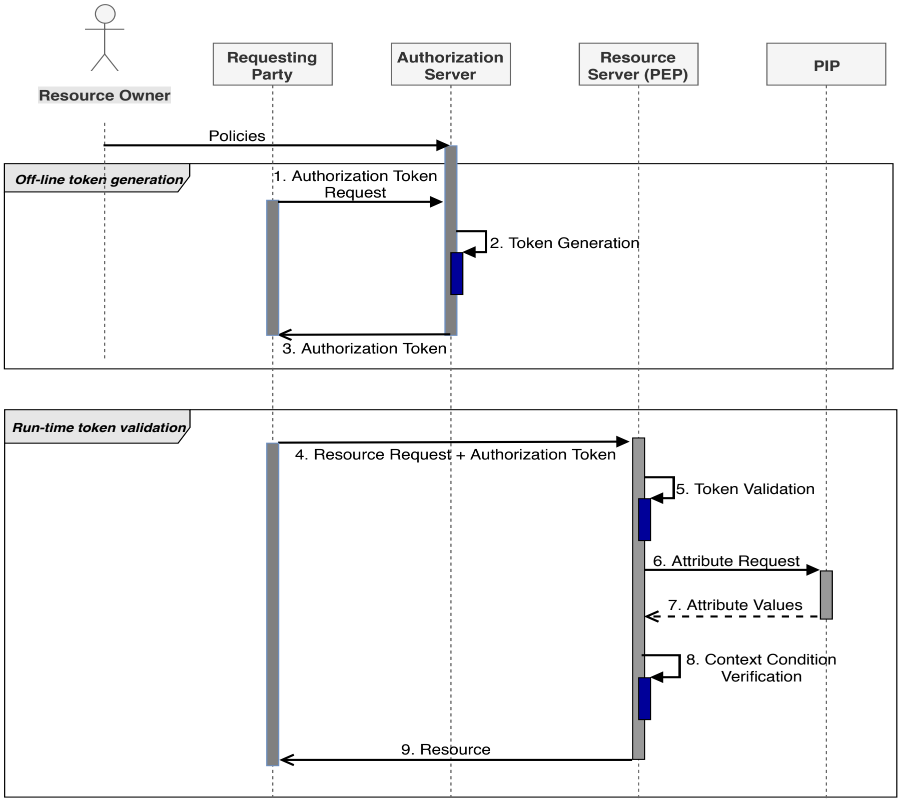


Fig. 5: Authorization process

simply assume that the policies to be enforced are provided to the authorization server. Interested readers can refer to [28] for a thorough discussion on this issue.

In this work, we consider policies specified in attribute-based access control (ABAC) as this paradigm provides a means for the specification of fine-grained access control policies. In ABAC, access requests and policies are defined in terms of attribute name-value pairs. Policies have a *target*, which defines the applicability of the policy by specifying to which requests the policy applies, and an *effect*, which specifies whether the subject has the permission to perform the specified action on the resource (permit) or not (deny). Figure 6a shows an example policy expressed in (a compact representation of) the XACML policy language [26]. This example policy is used to regulate the access to the location information of a given vehicle and consists of two rules combined using permit-override combining algorithm. The first rule states that subjects working in a certain insurance company are allowed to perform a GET operation to retrieve location information, whereas the second rule is used to restrict the access to the traffic authority operating in the region in which the vehicle is passing through. Note that policies, being evaluated off-line, can only be used to verify constraints on static properties of the subjects and resources. To account for context-depended properties

```
policy {
  “Combining Algorithm“: permit-overrides
  “target”: {
     “resource”: vehicle7282:location
  }
  “rule“:{
     “target”: {
       “subject_organization“: MyInsurance
       “action”: GET
     }
     “effect”: permit
     “constraint”:Driver=A120223
  }
  “rule“:{
     “target”: {
       “subject_role“: traffic authority
       “action”: GET
     }
     “effect”: permit
     “constraint”: resource_location ∈ subject_region
  }
}
```

(a) Policy

```
request: {
  subject_id: AI094520
  subject_organization: MyInsurance
  resource_id: vehicle7282:location
}
```

(b) Request

```
token{
  “permissions”: [
     {
        “resource_id”: vehicle7282:location
        “subject_id”: AI094520
        "scopes": [
           "GET"
           “constraint”: Driver=A120223
        ],
     }
  ]
  “created_at”: 2019-03-10T11:55:00
  “expires_at”: 2019-04-10T11:54:59
}
```

(c) Authorization Token

Fig. 6: Example of a policy, a request and a token

(e.g., location, current time), policies also include *constraints* that are returned along with the authorization token and verified at run-time time (see below).[1] For instance, in our example, the constraint of the first rule allows the insurance company to retrieve location information of the vehicle only when a given individual is driving the vehicle.

Off-line token generation: The authorization process starts with the off-line generation of an authorization token (top of Figure 5). The requesting party requests the authorization token from the authorization server to access a resource (1). The authorization server determines the permissions of the requesting party on the resource on the basis of the policies provided by the resource owner and generates an authorization token listing all permissions the requesting party has over the resource (2). The token is then sent to the requesting party (3). Figure 6c shows the authorization token generated by evaluating the access request in Figure 6b against the policy in Figure 6a. The token contains the *permissions* of the requesting party on the resource along with the validity period of the token. It is worth noting that the *constraint* specified in the policy is passed, together with the permissions, to the authorization token in order to prevent application overprivilege [17,35]. This constraint is then verified at run-time to determine whether access should be granted.

Run-time token validation: When the requesting party wants to access a resource, she sends a request to the PEP located in the resource server along with the authorization token (4). The resource server verifies whether the token is valid (5). In addition, the resource server verifies the constraints provided in the token. If additional information is needed to verify the constraints on the context, the resource server retrieves it from

[1] Constraints can be specified in XACML using element `<Obligations>`. In XACML, obligations are returned along with the access decision (either *permit* or *deny*) to enrich the decision.

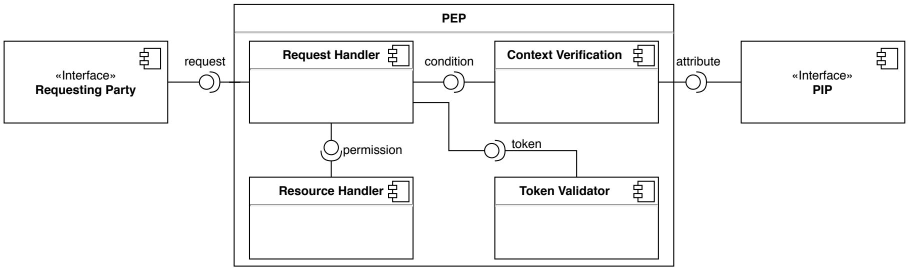


Fig. 7: Component Diagram of PEP

the appropriate sources (PIPs), which can be located within the resource server or in a different component (6 and 7). For example, the resource server could be vehicle which may have to retrieve context information from the nearby RSUs. If the verification of the constraint (8) is successful, the resource is disclose to the requesting party (9). Figure 7 provides a detailed view of the components and interfaces involved in the run-time token validation process.

Note that authorization tokens needs to be protected against tampering or relay attacks. How tokens can be protected against those attacks is out of the scope of this work and we refer to [11] for approaches commonly used to secure authorization tokens.

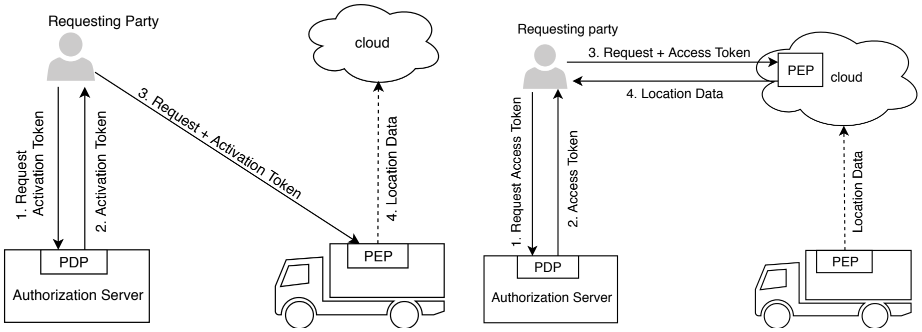
6 Application to Location Tracking Services

This section presents typical C-ITS use case scenarios and discusses how our authorization framework can be deployed to deal with such scenarios.

6.1 Location Tracking Services

Location information is an enabler for several services in the C-ITS domain [5]. For instance, location information can be used to increase vehicular safety (such as notification of nearby accident), tracking of stolen vehicle, pay-per-drive insurance, car sharing, toll payment, etc. To enable the retrieval of location information from a vehicle, the vehicle owner typically has to activate the forwarding of location information within the vehicle, including setting the time interval data are transmitted. Since this might generate a large amount of data, it is not ideal to forward the data to the requester directly. To this end, in our scenarios, we envision that data are transmitted to one of the C-ITS central systems (e.g., a Data Provider Back Office in the cloud), from which data can be retrieved when needed. Below we present typical use case scenarios relying on location tracking. These scenarios are an adaptation of the ones defined in the ETSI standard [5].

Scenario 1: Pay per drive insurance. Consider two sibyls, Alice and Bob, who co-own a car. They want to insure their car, but they would like different types of insurance. While Bob prefers a fixed premium, Alice wants a pay-per-drive insurance where the premium of the insurance policy is based on the kilometers traveled. In order to calculate



(a) Activation of Location Data Forwarding

(b) Retrieval of Location Data

Fig. 8: Deployment of the authorization framework for location tracking

the premium, the insurance company should be able to retrieve the location information from the vehicle when it is driven by Alice.

Scenario 2: Stolen Car. Alice and Bob's car was stolen and, thus, the two sibyls alert the police. Assuming that Bob has previously activated the forwarding of location information from the vehicle to the cloud, he can retrieve the exact location of his car in real time. Bob shares this information with the police to assist them in retrieving the car.

While enabling a variety of services, location information is sensitive and, thus, should be protected from unauthorized accesses. Next, we present how the authorization framework in Section 5 can be used to enable the selective sharing of location information.

6.2 Authorization Framework for Location Tracking Services

The first step for adapting our authorization framework to the scenarios above is to identify the C-ITS systems to which its components are deployed. In the scenarios, the owner of the vehicle represents the resource owner as he is the entity to whom information refers and, thus, he has the control on how the information is processed and to whom it can be disclosed [14]. The authorization server is handled by the Authorization Authority within the support system (cf. Fig. 3). The insurance company (scenario 1) and the police (scenario 2), which can be seen as two instances of the service provider back office (SP-BO) within the central system, are the requesting parties.

The scenarios involve two main phases: a first phase in which the forwarding of location information is activated and a second phase in which the information is retrieved from the Data Provider Back Office (DP-BO) that the vehicle owner used to store its data. Accordingly, the C-ITS system acting as the resource server varies in the two steps; in the first step the V-OBU acts as the resource server whereas in the second step the DP-BO acts as the resource server. We also distinguish two types of authorization tokens based on their purpose, namely *activation tokens* and *access tokens*. Activation tokens are used to enable the forwarding of location data from vehicle to the cloud. Access tokens are used to enable the retrieval of location information from the cloud.

Forwarding of Location Information: Alice wants to activate the gathering of the location of her vehicle, e.g., to enable vehicle tracking as demanded by her insurance company. To this end, she enables the forwarding of location information from the vehicle to the cloud. Figure 8a depicts the forwarding activation process. The requesting party (acting on behalf of Alice) requests an activation token to the authorization server (1). The authorization server provides Alice with an activation token listing her permissions on the vehicle (2). These steps are performed during the off-line phase. At run-time, the requesting party provides the activation token to the V-OBU (3). The PEP in the vehicle validates the token as well as verifies the constraints on the context (if any). Upon successful validation, the vehicle starts forwarding location information to the cloud (4).

Retrieval of Location Information: Suppose that Alice has specified a policy that allow the insurance company to access location information of her car but only under the condition that she is driving (see Fig. 6a). To comply with Alice's access requirements, the resource server (i.e., the cloud) has to verify this constraint at run-time before disclosing location data. This means that the resource server might have to retrieve additional information from the vehicle or road-side units in order to evaluate such constraints. Figure 8b depicts the information retrieval process. In the off-line phase, the requesting party (acting on behalf of the insurance company) requests an access token to the authorization server (1). The authorization server verifies the permissions of the insurance company and provides it with an access token (2). When requesting access to the location information of Alice's vehicle, the insurance company attaches the access token to the request (3). The resource server validates the token and verifies the constraints on the context conditions (i.e., whether Alice is driving). Upon successful validation, the location information is disclosed to the insurance company (4).

7 Discussion

This section discusses the feasibility of our framework and provides a qualitative analysis of the main design choices with respect to the challenges presented in Section 2. These choices encompass the use of a hybrid authorization framework, the use of a centralized authorization server and the handling of contextual information.

We have adopted a hybrid authorization framework that combines principles of both policy-based and token-based frameworks. As discussed previously, policy-based frameworks perform policy evaluation at request time, introducing delay in service provisions. This, however, might be problematic in critical systems as C-ITS. In our design of the authorization framework, we leverage a token-based architecture where a token is generated off-line and then validated (along with the constraints on the context) at run-time, when access to a resource is requested. This allows performing policy evaluation off-line, thus reducing overhead and latency [24]. However, differently from existing token-based frameworks like OAuth [11], which require the resource owner to authorize an application the first time it requires access to a resource, we automate the generation of tokens by exploiting the use of policies. Although there have already been efforts to integrate the use of policies in the token-based architecture [3], existing framework usually do not support the verification of context conditions, making them unsuitable to deal with the dynamicity of C-ITS. It is worth noting that token validation along with verification of context conditions does not introduce a significant overhead as this operation is significantly less expensive than policy evaluation and token generation [16].

Our framework employs a centralized component for token generation (i.e., the authorization server). This provides resource owners with a single point for policy administration where they can efficiently manage their policies [7,18]. The use of a centralized authorization server can also bring other advantages compared to deploying the policy decision point into (multiple) edge nodes (e.g. [32]) or within vehicles (e.g. [12]). For example, it allows exploiting the benefits of cloud computing in terms of scalability and reliability. It is worth noting that, in C-ITS, entities can rely on several resource servers to store and manage their data and resources. Therefore, an approach based on sticky policies [29], in which the resource server is required to attach policies to the data, is not particularly suitable as an entity would be required to configure their policies in each resource server in which her resources are stored.

In C-ITS, the information needed to verify context conditions may have to be retrieved from different sources, e.g. vehicles, road-side units or cloud. Thus, assuming that the resource server is the only source of context information as in [8,15] restricts the types of context conditions that can be verified, thus limiting the level of granularity for access control. However, retrieving context information from different sources can have an impact on latency as it requires additional interactions between parties. Hence, one has to make a trade-off between the expressiveness of context conditions and the latency introduced by the retrieval of the information necessary for their verification.

Unlike other authorization frameworks, our framework has been designed to address the challenges characterizing the C-ITS domain. In this work, we have looked into these challenges from a design perspective. However, in practical deployments, other factors such as communication protocols (CoAP, MQTT) [21,27], data format (JSON, XACML), handling of token refreshing and revocation, should be taken into account. Nevertheless, we believe that our hybrid authorization framework makes a step forward to the development of practical authorization mechanisms tailored to C-ITS. Moreover, the adoption of a C-ITS reference architecture as a baseline for our framework facilitates its integration and realization in existing C-ITS deployment sites.

8 Conclusions and Future Work

In this paper, we have designed an authorization framework tailored to the C-ITS domain. Our framework leverages principles of both policy-based and token-based architectures to minimize the overhead introduced by the authorization process while providing fine-grained protection. We have adopted the C-MobILE reference architecture as a baseline for the design of our hybrid authorization framework. This will help identifying the C-ITS systems involved in the authorization process for a specific application scenario and, thus, realizing the framework at various C-ITS deployment sites. We have also provided a qualitative analysis of our framework by demonstrating its application to typical C-ITS scenarios and showing how it addresses the challenges characterizing the C-ITS domain.

In the future, we plan to implement, integrate and validate our authorization framework within existing C-ITS deployment sites. To this end, we will further refine the design of our framework by investigating communication and implementation aspects.

Acknowledgements. This work is funded by the Horizon 2020 C-MobILE project (723311) and the ITEA3 project APPSTACLE (15017).

References

1. CONVERGE. `https://converge-online.de`, accessed: 2019-6-25
2. US-ITS. `https://local.iteris.com/arc-it`, accessed: 2019-6-25
3. User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization. `https://kantarainitiative.org/file-downloads/rec-oauth-uma-grant-2-0-pdf/`, accessed: 2019-6-25
4. Directive 2010/40/EU of the European Parliament and of the Council. Official Journal of the European Union **50**, 207 (2010)
5. Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management. ETSI TS 102 940, ETSI (2018)
6. Abrougui, K., Boukerche, A.: Efficient group-based authentication protocol for location-based service discovery in intelligent transportation systems. Sec Commun Netw **6**(4), 473–484 (2013)
7. Ahmad, T., Morelli, U., Ranise, S., Zannone, N.: A Lazy Approach to Access Control as a Service (ACaaS) for IoT: An AWS Case Study. In: Proceedings of Symposium on Access Control Models and Technologies. pp. 235–246. ACM (2018)
8. Albouq, S.S., Fredericks, E.M.: Securing communication between service providers and road side units in a connected vehicle infrastructure. In: Proceedings of International Symposium on Network Computing and Applications. pp. 1–5. IEEE (2017)
9. Dajsuren, Y., Karkhanis, P., Kadiogullary, D., Fuenfrocken, M.: C-MobILE D3.1 Reference Architecture. Tech. rep. (2017), `http://c-mobile-project.eu/library/`
10. Damen, S., den Hartog, J., Zannone, N.: Collac: Collaborative access control. In: Proceedings of International Conference on Collaboration Technologies and Systems. pp. 142–149. IEEE (2014)
11. Denniss, W., Bradley, J.: OAuth 2.0 for Native Apps. RFC 8252, IETF (2017), `https://tools.ietf.org/html/rfc6749`
12. Dorri, A., Steger, M., Kanhere, S.S., Jurdak, R.: Blockchain: A distributed solution to automotive security and privacy. IEEE Communications Magazine **55**(12), 119–125 (2017)
13. Festag, A.: Cooperative intelligent transport systems standards in Europe. IEEE Communications Magazine **52**(12), 166–172 (2014)
14. Guarda, P., Zannone, N.: Towards the development of privacy-aware systems. Information & Software Technology **51**(2), 337–350 (2009)
15. Gupta, M., Sandhu, R.: Authorization Framework for Secure Cloud Assisted Connected Cars and Vehicular Internet of Things. In: Proceedings of Symposium on Access Control Models and Technologies. pp. 193–204. ACM (2018)
16. Hernández-Ramos, J.L., Jara, A.J., Marin, L., Skarmeta, A.F.: Distributed Capability-based Access Control for the Internet of Things. Journal of Internet Services and Information Security **3**(3/4), 1–16 (2013)
17. Jia, Y.J., Chen, Q.A., Wang, S., Rahmati, A., Fernandes, E., Mao, Z.M., Prakash, A.: ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms. In: Proceedings of Network and Distributed System Security Symposium (2017)

18. Kaluvuri, S.P., Egner, A.I., den Hartog, J., Zannone, N.: SAFAX - an extensible authorization service for cloud environments. Front. ICT **2015** (2015)
19. Karafili, E., Lupu, E.C.: Enabling data sharing in contextual environments: Policy representation and analysis. In: Proceedings of Symposium on Access Control Models and Technologies. pp. 231–238. ACM (2017)
20. Karkhanis, P., van den Brand, M., Rajkarnikar, S.: Defining the C-ITS reference architecture. In: Proceedings of International Conference on Software Architecture Companion. pp. 148–151. IEEE (2018)
21. Laaroussi, Z., Morabito, R., Taleb, T.: Service Provisioning in Vehicular Networks through Edge and Cloud: an Empirical Analysis. In: Proceedings of Conference on Standards for Communications and Networking. IEEE (2018)
22. Le, V.H., den Hartog, J., Zannone, N.: Security and privacy for innovative automotive applications: A survey. Computer Communications **132**, 17–41 (2018)
23. Mahmudlu, R., den Hartog, J., Zannone, N.: Data governance and transparency for collaborative systems. In: Data and Applications Security and Privacy XXX. LNCS, vol. 9766, pp. 199–216. Springer (2016)
24. Martinez, J.A., Ruiz, P.M., Marin, R.: Impact of the pre-authentication performance in vehicular networks. In: Proceedings of Vehicular Technology Conference-Fall. IEEE (2010)
25. Matteucci, I., Petrocchi, M., Sbodio, M.L.: CNL4DSA: a controlled natural language for data sharing agreements. In: Proceedings of Symposium on Applied Computing. pp. 616–620. ACM (2010)
26. OASIS: eXtensible Access Control Markup Language (XACML) v. 3.0. OASIS Standard (2013)
27. Ojanperä, T., Mäkelä, J., Mämmelä, O., Majanen, M., Martikainen, O.: Use Cases and Communications Architecture for 5G-Enabled Road Safety Services. In: Proceedings of European Conference on Networks and Communications. pp. 335–340. IEEE (2018)
28. Paci, F., Squicciarini, A.C., Zannone, N.: Survey on access control for community-centered collaborative systems. ACM Comput. Surv. **51**(1), 6:1–6:38 (2018)
29. Pearson, S., Casassa-Mont, M.: Sticky policies: An approach for managing privacy across multiple parties. Computer **44**(9), 60–68 (2011)
30. Ravidas, S., Lekidis, A., Paci, F., Zannone, N.: Access control in internet-of-things: A survey. Journal of Network and Computer Applications **144**, 79–101 (2019)
31. Raya, M., Papadimitratos, P., Hubaux, J.P.: Securing Vehicular Communications. IEEE Wireless Communications **13**(5), 8–15 (2006)
32. Riabi, I., Saidane, L.A., Ayed, H.K.B.: A proposal of a distributed access control over Fog computing: The ITS use case. In: Proceedings of International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks. IEEE (2017)
33. Salonikias, S., Mavridis, I., Gritzalis, D.: Access control issues in utilizing fog computing for transport infrastructure. In: Critical Information Infrastructures Security. pp. 15–26. LNCS 9578, Springer (2015)
34. van Sambeek, M., Ophelders, F., Bijlsma, T., Turetken, O., Eshuis, R., Traganos, K., Grefen, P.: Towards an architecture for cooperative-intelligent transport system (C-ITS) applications in the Netherlands. Tech. rep., DITCM Innovations (2015)
35. Schuster, R., Shmatikov, V., Tromer, E.: Situational Access Control in the Internet of Things. In: Proceedings of Conference on Computer and Communications Security. pp. 1056–1073. ACM (2018)
36. Sha, K., Xi, Y., Shi, W., Schwiebert, L., Zhang, T.: Adaptive privacy-preserving authentication in vehicular networks. In: Proceedings of International Conference on Communications and Networking in China. pp. 1–8. IEEE (2006)

37. Sucasas, V., Mantas, G., Saghezchi, F.B., Radwan, A., Rodriguez, J.: An autonomous privacy-preserving authentication scheme for intelligent transportation systems. Computers & Security **60**, 193–205 (2016)