Accountable and privacy-aware flexible car sharing and rental services

Francesco Pollicino
Department of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
francesco.pollicino@unimore.it

Luca Ferretti
Department of Physics, Informatics and Mathematics
University of Modena and Reggio Emilia
Modena, Italy
luca.ferretti@unimore.it

Dario Stabili
Department of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
dario.stabili@unimore.it

Mirco Marchetti
Department of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
mirco.marchetti@unimore.it

***Abstract*—The transportation sector is undergoing rapid changes to reduce pollution and increase life quality in urban areas. One of the most effective approaches is flexible car rental and sharing to reduce traffic congestion and parking space issues. In this paper, we envision a flexible car sharing framework where vehicle owners want to make their vehicles available for flexible rental to other users. The owners delegate the management of their vehicles to intermediate services under certain policies, such as municipalities or authorized services, which manage the due infrastructure and services that can be accessed by users. We investigate the design of an accountable solution that allow vehicles owners, who want to share their vehicles securely under certain usage policies, to control that delegated services and users comply with the policies. While monitoring users behavior, our approach also takes care of users privacy, preventing tracking or profiling procedures by other parties. Existing approaches put high trust assumptions on users and third parties, do not consider users' privacy requirements, or have limitations in terms of flexibility or applicability. We propose an accountable protocol that extends standard delegated authorizations and integrate it with Security Credential Management Systems (SCMS), while considering the requirements and constraints of vehicular networks. We show that the proposed approach represents a practical approach to guarantee accountability in realistic scenarios with acceptable overhead.**

***Index Terms*—V2X, access control, authorization, accountability, authentication**

I. Introduction

Since decades, large cities have become overcrowded, and vehicles and traffic congestion are destined to grow even further due to ongoing urbanization trends, also causing issues related to shortage of parking places and even worse increasing local pollution and reducing the quality of the life. Shared mobility systems were introduced to help solve transportation problems in urban areas, and it has many benefits for the end-users such as the lack of responsibilities of private vehicle ownership, maintenance, and insurance costs. Rather than owning one or more vehicles, the user can access a fleet of shared vehicles on an as-required basis. They also represent a flexible solution for commuters by offering them alternatives in addition to public transport and they help reducing the amount of intra- and inter- cities traffic raised by personal vehicles. In particular, personal car-sharing systems, where any vehicle owner can decide to rent its own vehicle to other people, represent a novel business model and an opportunity for developing innovative markets [1], [2]. The flexibility of these service also introduces challenges in terms of usability and requires developing novel infrastructures and services that traditional systems cannot satisfy [1]. Sharing and rental services based on smart vehicles, apps and cloud services have become widespread and offer a way to build a modern personal car-sharing system. However, they introduce risks in terms of security and privacy [3].

In this paper, we design a novel architecture for flexible car sharing and rental services which allows people to rent their own vehicle to other people through intermediate brokers. These brokers must operate as authorization services and enforce the requirements defined by vehicle owners when allowing users to access the vehicles. Our proposal tackles security and privacy challenges related to the possibility that the involved entities do not always behave honestly and that vehicle users could be traced for commercial purposes. Our proposal represents a solution for building an accountable delegated authorization protocol that allows vehicle owners to expose potential misbehavior of the user and guarantee the privacy of the users with respect both the vehicle owner and the authorization service.

In existing systems, users must submit their information such as their identity and driving license to the service provider when they request a car-sharing service. The service provider verifies that the customer has the right and ability to drive as a valid driver. After that, the user can utilize the car-sharing service through a service provider.

The rest of the paper is organized as follows. Section II outlines the related work. Section III introduce the base

knowledge required for the understanding of the paper, Section IV define the considered scenario, Section V describes the proposed architecture, and the details of the proposed protocol are detailed in Section VI. Conclusions follow in Section VII.

II. Related work

Car rental systems may be deployed by using two type of approaches depending on the characteristics and requirements of the involved scenarios: station-based and free-floating [4]. In station-based systems, the rental service reserves stalls for pick-up and return, and ensures the availability of the car when it is needed. This approach is aimed at users who decide to replace ownership of a car with the use of other means (public transport, car sharing, car rental, taxi), and for those who make sporadic use of the car and rely on systematic travel. It is therefore connoted as strongly complementary to public transport and allows a gradual expansion of the service, on a geographical basis. Furthermore, it has the advantages of being also suitable for medium-sized city centers. In free-floating systems, cars can only be picked up if available at the time of use, the car can be released in any place within a predefined urban perimeter, and the withdrawal takes place from the point of release of the previous user. It is aimed at users who make trips which can be carried out with other means of transport. For this reason, it is an alternative to taxis or public transport or bicycles. It is therefore connoted as an addition to the public transport system. These two models are suitable for different types of mobility and can be considered complementary. Our proposal supports both type of model, and we do not introduce any constraints about the operational model.

Security and privacy of car rental sharing systems have been analyzed by the literature from multiple perspectives [3], [5], [5], [6]. The authors of [3] propose an analysis of security risks of keyless systems for accessing vehicles in car sharing services and propose a keyless sharing system (KSS). Our paper can leverage any authentication system for allowing users to access vehicles and thus can be considered as complementary to existing proposals in this context. Our proposal is more related to authorization systems for car sharing proposed in [6], [5] and [7]. In [6] the authors propose a secure free-floating car-sharing system that allows users to reserve the car-sharing service by authenticating himself by using standard access tokens by using a mobile device. However, the proposed scheme does not protect the privacy of the users nor enables auditability. In [5] the authors analyze cybersecurity issues for Shared electric and automated mobility (SEAM), including peer-to-peer car sharing, and present security solutions regarding authentication and authorization procedures. Our proposal differs from those papers because we use a third party that deals with managing the pseudonyms to guarantee the privacy of the end-users. The authors of [7] analyze users privacy and propose an authentication protocol for a car sharing service which addresses privacy-preserving by using a pseudonym. In particular, they use user-centric pseudonyms' management system based on Hierarchical Identity-Based Signature (HIBS). We differ from this proposal because we use the same security credential managed system (SCMS) used in V2X communication [8] that has the advantage of reusing protocols that are emerging as standard for vehicular infrastructures. Moreover, the security schemes used by SCMS systems are based on more established and lightweight cryptographic primitives that can be deployed in existing contexts.

Accountability for authorization protocols have also been proposed in the context of IoT for improving control of authorizations released for accessing distributed devices [9] and for improving security of supply chains in cyber-physical environments [10]. In particular, these proposals adopt transparency logs to enable monitoring of delegated parties and compliance to authorization policies. Our proposal represents a similar approach but does not introduce transparency logs to the authorization architecture. Integration of similar approaches is left as future work.

III. Background

A. Delegated authorizations in constrained environments

Delegated authorization frameworks allow to manage *authorizations* to access to *resources* in modern Web scenarios which include many stakeholders and require secure and scalable solutions.

Our proposal is modeled after the OAuth2 framework [11] which is the most popular standard in Web architectures, which includes four types of entities: a *resource owner* that is authoritative for the life-cycle of the *resources*, a *resources server* that hosts the *resources*, a *client* that needs to access the *resources*, an *authorization server* that releases *security tokens* that allow *clients* to access *resources* stored in *resource servers*. Moreover, we consider OAuth2 extensions for Access Constrained Environments (OAuth2-ACE) [12], which consider scenarios where *resource servers* might not be typical Web services that are assumed to be always connect to the Internet and available through standard HTTPs protocols. Instead, OAuth2-ACE supports *resource servers* with unstable or no Internet connection and that possibly use transport protocols with poor security guarantees.

B. ECQV implicit certificates

A certificate chain includes the public keys of the sender of the message and of all the intermediate CAs, hence the size of the certificate chain is proportional to the number of intermediate CAs. Verifying the certificate chain requires to verify the digital signatures attached by all the intermediate CAs up to the Root Certificate.

The implicit certificate scheme uses a more complex approach that leverages particular mathematical properties to bind identity information and public key *without explicitly* storing them. The most popular protocol for implicit certificate is ECQV [13]. Intuitively, an ECQV certificate does not include the public key of the sender, but allows a recipient to recompute it by using the certificate of the sender and the public key of the CA, thus saving network usage. Although implicit certificates seem very convenient thanks to their space efficiency, they have a few disadvantages that limit their adoption

in common Web communications. Network savings of implicit certificates can be considered negligible in most Web scenarios, because communications are mostly operated through channels exchanging large amounts of data and certificates are only used once during the secure channels handshakes. However, vehicular networks are deployed on possibly low-rate wireless networks and have tighter latency requirements. As a result, network savings of implicit certificates might be worth the more complex management procedure and reduced flexibility.

The ECQV operations framework includes four routines: *certificate sign request*, *certificate generation*, *certificate reception*, and *public key extraction*. These operations, combined with the *signature* and *verification* procedures of ECDSA, allow guaranteeing message integrity and authenticity. In the following we describe the flow of the operations from the generation of an ECQV certificate to the signature verification by referring to Figures 1 and 2.

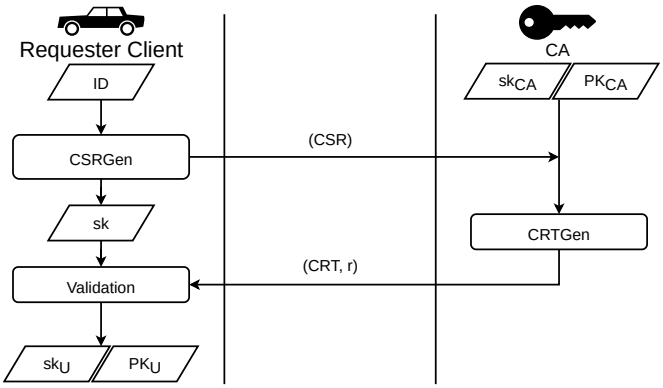

Fig. 1. Operation flow for the generation of the ECQV certificate

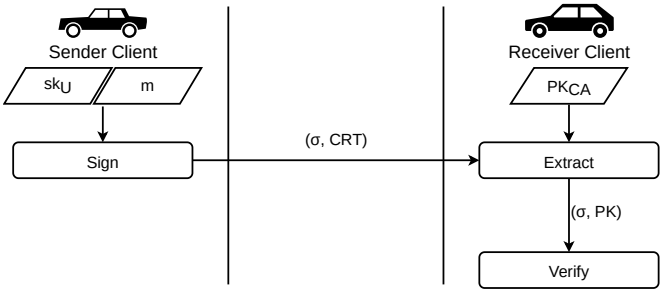

Fig. 2. Operations flow for the ECDSA signature based on ECQV certificates

A *requester* is a client that requests a valid certificate to the CA by generating a *Certificate Sign Request* (CSR) via the *CSR generation* function ($CSRGen$). The $CSRGen$ function requires the id of the entity requesting the certificate, and produces an output composed by the CSR (that includes id and PK, which is a intermediate public key of the requester) and the intermediate private key sk. The CSR is sent to the CA and, upon identity verification, produces the corresponding certificate CRT and a private key contribution r by using the *CRT generation* function ($CRTGen$). The $CRTGen$ function requires the CSR and the key pair of the CA sk_{CA}, PK_{CA} to produce the CRT. After the reception of the CRT from the CA, the requester validates the certificate with the *CRT Validation* function ($CRTRec$). Upon verification, the requester generates the final key pair (sk_u, PK_u) by using CRT and r. The private key sk_U is used generate the signature σ of the messages with the ECDSA *signing* function ($Sign$). The client uses the private key SK_U with the ECDSA *signing* function ($Sign$) to sign messages. After the signature $sigma$ is computed, the client sends the message m, the signature σ, and its own certificate, CRT by broadcasting it to the nearby clients.

C. Security Credential Management System (SCMS)

The Security Credential Management System (SCMS) is a public key infrastructure (PKI) designed to secure V2X messages with privacy as the highest priority. The SCMS structure must mitigate attacks on end-users privacy from SCMS insiders and outsiders. The POC has been officially adopted as a protocol by the United States Department of Transportation (USDOT) [14] and became an industry standard for all providers of PKI-based V2X security solutions, including AUTOCRYPT.

In this paper we exploit the architecture thoroughly described in [8], which is similar to the European C2X PKI described in [15], to build an authorization schema that guarantees the privacy of the users. In the following, we propose a description in terms of components, certificates types, and operations of the SCMS that are useful for a better comprehension of our scheme.

The SCMS includes a large set of components, but we focus only on those that are of use for the comprehension of our proposal, for a complete reference of all the components and actors please refer to the original paper [8].

- *enrollment certification authority (ECA)*: issue enrollment certificates that are used by the users to authenticate against the registration authority, there may be more ECA divided, for example, by geographical region;
- *registration authority (RA)*: creates individual requests for certificates to the PCA ensuring that revoked user can't require new certificates and that a user can't request more than one set of certificates for a given period,
- *pseudonym certification authority (PCA)*: issue short term pseudonym certificates to users;
- *intermediate CA*: shield the root CA from traffic and attacks;
- *root CA* is the root at the top of the certificate chains in the SCMS.

The SCMS supports different requirements for certificate management and different type of certificates, in the following, we show the three certificates types that are of interest for the contribution of the paper:

- *enrollment certificates* are provided during bootstrap and are used to request message signing and/or encryption certificates;

- *pseudonym certificates* provides pseudonymity, unlinkability, and efficient revocation of a multitude of certificates;
- *identification certificates* are classic certificate that identify and entity.

In particular, we focus on pseudonym certificates that are used to grant authorization when unlinkability is required. A pseudonym certificate does not include any real-world identifier and a given user can have a set of multiple simultaneous valid pseudonym certificates that cover an interlaced period.

The main operations of the SCMS that allow managing users' certificates from enrollment to (eventually) revocation are the following.

- *Bootstrapping*: provides all the information required to communicate with the SCMS. Bootstrapping consists of two operations: initialization and enrollment. With the first one, the user obtains certificates it needs to be able to trust received messages, with the second the user obtains an enrollment certificate that is used to sing messages to the SCMS.
- *Pseudonym certificate provisioning*: is designed to protect the privacy of the end-users and is the most complex process in the SCMS. The privacy mechanisms included in the SCMS allow to obscure the physical location of the user to the RA and the MA; hide certificates from the RA in a way that no one can correlate the public key seed with the resulting certificates; hide receiver and certificate linkage from PCA in way that the PCA can't correlate two certificate requests to the same user.
- *Linkage values*: are coupled to a set of pseudonyms and are used to revoke all the certificates with validity equal to or later than some time.
- *Misbehavior*: allows reporting, investigation, and eventually reaction to misbehaving users. The reaction includes revocation and blacklisting of all the certificates corresponding to the linkage values of the incriminated pseudonym certificate.

IV. Reference scenario

We consider a flexible car rental scenario where people can rent their vehicles to other people under certain constraints, such as adopting the vehicle only within a certain distance and within certain timings. To regulate access control procedures to their vehicles, owners subscribe to an intermediate broker to delegate it with the capability of allowing users to access vehicles through their smartphone (e.g. NFC technology in the vehicle door, or by using the GPS position of the user). The broker deploys the due infrastructures and services to handle users' rental requests and release authorization material to access vehicles. Each car is equipped with a smart start system that unlocks and starts the vehicle only if the user has obtained the due authorization, possibly after completing a payment procedure [16]. Moreover, the car is equipped with sensors that collect information about the vehicle state and position. The car is also connected to the Internet to send this information to an online service that can detect a potential misbehavior.

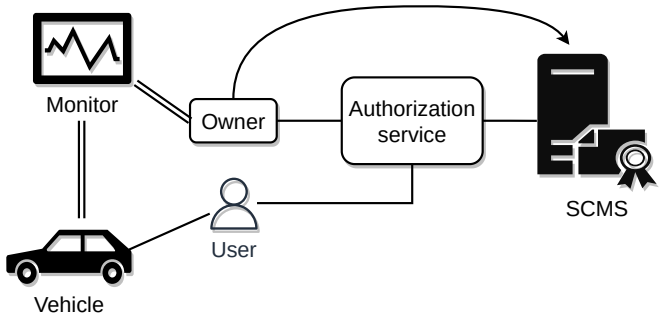

Fig. 3. Proposed architecture

We consider a rental procedure where a user wants to rent a specific vehicle. We model the procedure in three phases. First, the user sends a request to the broker to access the vehicle. Second, the broker validates the request and decides whether the user should be authorized to use the vehicle. Third, if the broker decides that the user is authorized, it releases authorization material that allows the user to access and start the vehicle. If the user is not authorized, the broker denies the user access by not releasing the authorization material.

The aim of our proposal is to design an architecture that allows vehicle owners to monitor the behavior of both brokers and users. Our proposal allows using the vehicle sensors to detect if a user violates the usage agreement established between the user and the broker. The bad behavior of a user allows the architecture to revoke all the authorizations released to the same user to use other vehicles. Moreover, our proposal also allows to detect if a broker authorizes users to use a vehicle under agreements that violate the usage constraints defined by vehicle owners. The bad behavior of a broker can be exposed publicly, thus affecting its reputation and acting as a deterrent. This is a very important contribution to build a flexible rental ecosystem, where many brokers can act in different geographical locations. Moreover, an important guarantee of our proposal is to protect the privacy of the users against tracking by brokers and owners.

We assume that owners and brokers have known identities and can establish secure and mutually authenticated connections by using standard Web protocols for secure communications and credential systems. We assume that users can adopt secure authentication protocols to access vehicles. We assume that vehicles can be provided with a small number of public keys used to authenticate access requests and support the related authentication protocols. We assume that per-owner public keys can be installed in the vehicle at purchase time and that they can updated via online update systems.

V. Proposed architecture

A. System model

We describe the proposed architecture by referring to Figure 3, which includes the *Security Credentials Management*

System (SCMS) (Section III-C) and five additional roles: *vehicle*, *owner*, *monitor*, *user*, and *authorization service*. We discuss each of the roles with regard to actors of delegated authorization frameworks (Section III-A).

The *vehicle* represents the resource that must be managed by the authorization architecture. The *vehicle* can be used by authorized *users* through a dedicated service via wireless connectivity (e.g., smart vehicle door with smart card [17]). Moreover, the vehicle is provided with sensors that are able to detect information about its state (e.g., GPS position) and is able to send this information to the vehicle *monitor*.

The vehicle *owner* acts as the authority of the resource and decides the *policies* to access the vehicle. It delegates the *authorization service* to decide which *users* can obtain the due authorization grants to access its *vehicle*. The *owner* can also interact with the vehicle *monitor* to control the state of the vehicle and to verify the compliance of the usage *policies*.

The vehicle *monitor* denotes a third party that collects the vehicle sensors data and allows the *owner* to access these data to validate delegation policies. Real world examples might include the vehicle manufacturer or an insurance company that already collects such data for other purposes (e.g., remote diagnostic, emergency assistance, incidents forensics).

The *user* denotes a person that requires access to a vehicle. To this aim, he submits a request to the *authorization service* by providing a *pseudonym certificate* and *usage policies*. If authorized, he receives the *authorization grant* that he can use to access the *vehicle*. A honest *user* must comply with the submitted *usage policies*, otherwise it could be reported by the *owner* of the vehicle and possibly revoked from the rental service.

The *authorization service* acts on behalf of the vehicle owner to control the access to the vehicle. It receives the *authorization request* from the *user* and releases the *authorization grant* if the request complies with the policies defined by the owner.

The *SCMS* acts as the identity provider for *users*, releases *pseudonym certificates* and manages the revocation of users. We do not modify its behavior with regard to existing standard SCMS architectures (see Section III-C).

Our proposal assumes that the *user* has already registered at the SMCS through the *bootstrapping* procedure and obtained an *enrollment certificate*, and that he can obtain multiple *pseudonym certificates* through the *pseudonym certificate provisioning* procedure. Moreover, we assume that the *owner* has already registered at the *authorization service*, and that he can delegate authorization management to it.

We describe the life-cycle of the architecture through three macro-phases:

- *authorization delegation*: the owner delegates the management of the authorization of a vehicle to the authorization service, specifying the policies that must be enforced;
- *authorization grant and vehicle access*: the user submits a request to the authorization service and release the authorization grant if the request complies with the policies defined by the owner
- *vehicle monitor and user revocation*: the owner interacts with the vehicle monitor to control the state of the vehicle and to verify the compliance of the usage policies, if the user misbehaves he can be reported to the SCMS that can eventually revoke him from the system.

We describe the details of each of the phases and of the management of the disputes in the following Section VI.

B. Threat model

- *Users* are untrustworthy or malicious, they can try to obtain illegitimate access to the vehicle or alter the information exchanged with the authorization service to lower the credibility of the system. The user can try to corrupt the information exchanged with the authorization service or the vehicle to gain some advantages (e.g. extend the rent period). The user can try to interfere with the vehicle to obtain information about the other user that rented the vehicle.
- *Owners* are semi-honest, they can try to learn and extract information about the users such as booking preferences, travel, and frequency. Owners have no interest to alter the protocol flow or the exchanged messages.
- *Monitors* are semi-honest, but they can try to learn and extract information about the users on behalf of the owners. Monitors have no interest to alter the protocol flow or the exchanged messages.
- *Vehicles* are untrusted but tamper-evident, attackers can try to interfere with the vehicle sensors to obtain advantages or learn about the behaviors of the other users. We assume that the vehicles are equipped with Trusted Platform Modules (TPMs) to safely store information.
- *Authorizations services* are semi-trusted could issue authorizations that do not comply with the policies defined by the resource owners, that is, they may illegitimately deny or release an authorization to a client that do not comply with the authorization policies defined by the resource owners.
- *SCMS* is a trusted third part.

VI. Details of the proposed protocol

We describe the details of the macro-phases of the protocol: *authorization delegation* (Sections VI-A), *authorization grant and vehicle access* (Section VI-B), *vehicle monitor and user revocation* (Section VI-C). We conclude by discussing how the proposal allows solving potential disputes among the parties (Section VI-D).

A. Authorization delegation

We describe authorization delegation by referring to Figure 4. The owner starts the procedure to delegate authorization management by sending an authorization policy attestation APA_o to the *monitor* that is built as follows:

$$AP_o = \langle id_o, id_v, T, AR \rangle \tag{1}$$

$$APA_o = \langle id_m, AP_o, \sigma_o\left(id_m, AP_o\right) \rangle \tag{2}$$

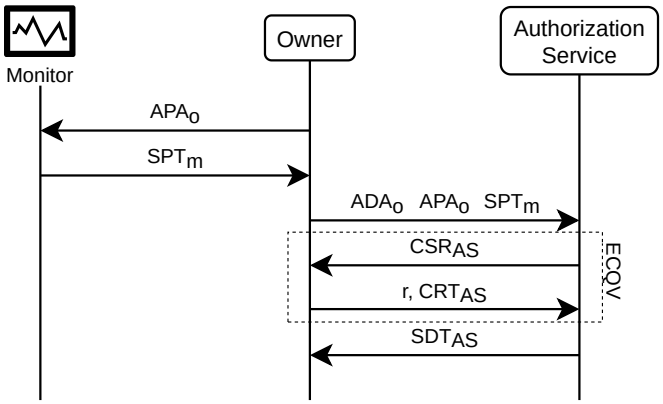

Fig. 4. Authorization delegation

where id_o is the unique identifier of the owner, id_v is the unique identifier of the vehicle, T is the time range that identifies the validity of the delegation, and AR is the authorization rule decided by the owner. The *monitor* validates the contents and the digital signature, and accepts the delegation procedure by answering with a *signature policy timestamp* SPT_m built as:

$$SPT_m = \langle \mathcal{H}(APA_o), \sigma_m(APA_o) \rangle \tag{3}$$

The owner builds an authorization delegation attestation ADA_o as follows:

$$ADA_o = \langle AP_o, \sigma_o(AP_o, PK_{AS}) \rangle \tag{4}$$

The owner sends ADA_o, APA_o and SPT_m to the AS, which verifies that SPT_m has been generated by using APA_o and is signed by the monitor. If the AS accepts the delegation, it starts the ECQV certificate generation procedure (see Section III-B):

- the AS generates the certificate sign request CSR_{AS} by using the owner identifier id_o, the vehicle identifier id_v and the time range T as the certificate metadata in the $csrgen$ routine:

$$CSR_{AS} = csrgen(id_o, id_v, T); \tag{5}$$

- the owner generates the implicit certificate CRT_{AS} by using the certificate generation routine:

$$(CRT_{AS}, r) = crtgen(sk_o, PK_o, CSR_{AS}), \tag{6}$$

  where r denotes the private key contribution;
- the AS generates the key pair (sk_{AS}, PK_{AS}) by using the certificate reception routine:

$$(sk_{AS}, PK_{AS}) = crtrec(PK_o, CRT_{AS}, r). \tag{7}$$

Finally, the AS sends the signed delegation timestamp SDT_{AS} built as follows:

$$SDT_{AS} = \langle \mathcal{H}(ADA_o), \sigma_A S(ADA_o) \rangle \tag{8}$$

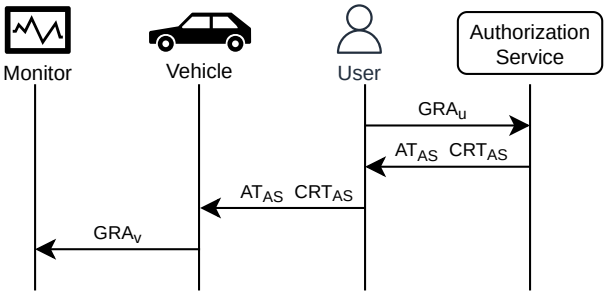

Fig. 5. Authorization grant and vehicle access

B. Authorization grant and vehicle access

We describe authorization grant and vehicle access by referring to Figure 5. The user requests access to a vehicle by sending a Grant Request Attestation GRA_u to the AS:

$$GRA_u = \langle pcrt_u, AP_u, \sigma_u(pcrt_u, AP_u) \rangle \tag{9}$$

where $pcrt_u$ is the pseudonym certificate chosen by the user to guarantee its anonymity, and AP_u is the authorization policy to which the user declares to comply when using the vehicle. First, the AS must verify the authenticity of the digital signature $\sigma_u(pcrt_u, AP_u)$ and the validity of the provided pseudonym certificate $pcrt_u$ against the public key of the SCMS PK_{scms} and the most updated version of the CRL_{scms}. Second, the AS verifies that AP_u complies with the authorization policy AP_o previously released by the owner (Equation (1)). If both verification procedures succeed, the AS returns its implicit certificate CRT_{AS} and the access token AT_{AS} built as follows:

$$\begin{aligned} AT_{AS} = \langle pcrt_u, AP_u.T, \mathcal{H}(GRA_u, APA_o), \\ \sigma_{AS}(pcrt_u, AP_u.T, \mathcal{H}(GRA_u, APA_o)) \rangle \end{aligned} \tag{10}$$

The user presents both CRT_{AS} and AT_{AS} to the vehicle. First, the vehicle uses the owner public key PK_o and CRT_{AS} as inputs of the ECQV *extract* routine (see Section III-B) to obtain the AS public key PK_{AS}. Second, it verifies the authenticity of AT_{AS} by using PK_{AS}. Third, it verifies the time range $AP_u.T$ is valid with regard to the current time known by the vehicle, and that it is included in the time range of the implicit certificate metadata, that is $CRT_{AS}.T$. Finally, the vehicle presents the GRA_v to the monitor:

$$GRA_v = \langle pcrt_u, AP_u.T, \sigma_v(pcrt_u, AP_u.T) \rangle \tag{11}$$

The monitor verifies that $AP_u.T$ is included in APA_o and, if it is not included, the monitor sends an alert to the owner.

C. Vehicle monitor and user revocation

We describe vehicle monitor and user revocation by referring to Figure 6. The *monitor* receives information about the state of the vehicle from its sensors in real-time, and verifies compliance with the authorization policies established by the *owner* and communicated to the monitor within the APA_o data structure (Eq.(2)). If the vehicle is not compliant with

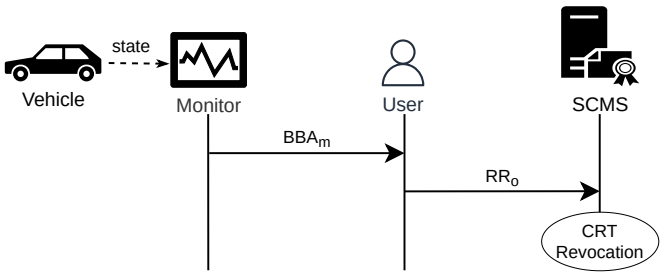

Fig. 6. Vehicle monitor and user revocation

the authorization policies, the *monitor* sends an alert BBA_m to the *owner*:

$$BBA_m = \langle id_v, id_m, pcrt_u, VS, \mathcal{H}(APA_o), \sigma_m(id_v, id_m, pcrt_u, VS, \mathcal{H}(APA_o))\rangle \quad (12)$$

Then the owner sends a revocation request RR_o to the SCMS, attaching BBA_m as a proof of the user misbehavior:

$$RR_o = \langle BBA_m, APA_o, \sigma_m(BBA_m, APA_o)\rangle \quad (13)$$

The SCMS verifies the validity of RR_o, revokes all the pseudonym certificates related to the user pseudonym $pcrt_u$, updates the CRL, and save the RR_o.

D. Disputes

The *user* starts the repudiation procedure at the SMCS by sending the access token AT_{AS} and the Grant Request Attestation GRA_u (Eq. (9)) received by the AS (Eq. (10)). The SCMS validates the access token AT_{AS} against the collected revocation requests RR_o (Eq. (13)) by operating the following steps:

- it verifies the digital signatures included in GRA_u and AT_{AS};
- it retrieves the revocation request RR_o that is associated with the user: $RR_o.BBA_m.pcrt_u \stackrel{?}{=} GRA_u.pcrt_u$;
- it verifies that the received AT_{AS} is correctly bound to GRA_u and to the policy APA_o included in RR_o. That is, it computes $\mathcal{H}(GRA_u, RR_o.APA_o)$ and verifies that it is equal to the corresponding value included in AT_{AS}.

Then, the SCMS decides the outcome of the dispute by analyzing the policies included in $RR_o.APA_o$ and $GRA_u.AP_u$, and the vehicle state information $RR_o.BBA_m.VS$:

- the user revocation is invalidated if the vehicle state $RR_o.BBA_m.VS$ complies with $GRA_u.AP_u$. This scenario implies that $GRA_u.AP_u$ violates $RR_o.APA_o$ and thus that the authorization service AS (which approved $GRA_u.AP_u$) misbehaved. Thus, the authorization service AS could also be reported and possibly sanctioned;
- the user revocation is confirmed if $GRA_u.AP_u$ does not violate $RR_o.APA_o$, because this implies that the usage of the vehicle by the user actually violated the policies decided by the owner.

VII. Conclusions

We proposed a system for building a flexible car rental service where people can make their vehicles available for rental through intermediate brokers. High levels of security as well as privacy are guaranteed by deploying accountable protocols to detect and prove misbehaving parties and pseudonym certificates based on Security Credential Management Systems that protect users against traceability. Moreover, the proposal shows a novel use of implicit certificates to build space-efficient attested authorization material to reduce network overhead and fits even deployments within constrained networks. The designed system could open the way to novel flexible rental paradigms within emerging smart cities.

References

[1] S. A. Shaheen, M. A. Mallery, and K. J. Kingsley, "Personal vehicle sharing services in north america," *Research in Transportation Business & Management*, 2012.

[2] S. A. Shaheen and A. P. Cohen, "Carsharing and personal vehicle services: worldwide market developments and emerging trends," *International journal of sustainable transportation*, 2013.

[3] I. Symeonidis, M. A. Mustafa, and B. Preneel, "Keyless car sharing system: A security and privacy analysis," in *IEEE Int'l Smart Cities Conf*, 2016.

[4] F. Ciari, B. Bock, and M. Balmer, "Modeling station-based and free-floating carsharing demand: Test case study for berlin," *Transportation Research Record*, 2014.

[5] B. Vaidya and H. T. Mouftah, "Security for shared electric and automated mobility services in smart cities," *IEEE Security & Privacy*, 2020.

[6] A. Dmitrienko and C. Plappert, "Secure free-floating car sharing for offline cars," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, 2017.

[7] M. Laurent, J. Leneutre, S. Chabridon, and I. Laaouane, "Authenticated and privacy-preserving consent management in the internet of things," *Procedia Computer Science*, 2019.

[8] B. Brecht and T. Hehn, "A security credential management system for v2x communications," in *Connected Vehicles*. Springer, 2019.

[9] L. Ferretti, F. Longo, M. Colajanni, G. Merlino, and N. Tapas, "Authorization transparency for accountable access to iot services," in *Proc. Third IEEE Int'l Cong. Internet of Things*, July 2019, pp. 91–99.

[10] L. Ferretti, F. Longo, G. Merlino, M. Colajanni, A. Puliafito, and N. Tapas, "Verifiable and auditable authorizations for smart industries and industrial internet-of-things," *Journal of Information Security and Applications*, vol. 59, p. 102848, 2021.

[11] IETF, "RFC 6749: The OAuth 2.0 Authorization Framework," Oct. 2012.

[12] L. Seitz, G. Selander, E. Wahlstroem, S. Erdtman, and H. Tschofenig, "Authentication and authorization for constrained environments (ace) using the oauth 2.0 framework (ace-oauth)," https://www.ietf.org/id/draft-ietf-ace-oscore-profile-02.txt, Internet-draft, Dec. Dec. 2018.

[13] Certicom Research, "Sec 4: Elliptic curve qu-vanstone implicit certificate scheme, standards for efficient cryptography group. version 1.0," SEC 4, 2013.

[14] U.S. Department of Transportation-National Highway Traffic Safety Administration. (Last visited Sept. 2021) Federal Motor Vehicle Safety Standards; V2V Communications. https://www.federalregister.gov/documents/2017/01/12/2016-31059/federal-motor-vehicle-safety-standards-v2v-communications.

[15] N. Bißmeyer, H. Stübing, E. Schoch, S. Götz, J. P. Stotz, and B. Lonc, "A generic public key infrastructure for securing car-to-x communication," in *18th ITS World Congress, Orlando, USA*, vol. 14, 2011.

[16] C. Busold, A. Taha, C. Wachsmann, A. Dmitrienko, H. Seudié, M. Sobhani, and A.-R. Sadeghi, "Smart keys for cyber-cars: Secure smartphone-based nfc-enabled car immobilizer," in *Proceedings of the third ACM conference on Data and application security and privacy*, 2013.

[17] J. Alsayaydeh, A. Khang, W. Indra, H. A. Zakir, V. Shkarupylo, S. Saravanan, and J. Pusppanathan, "Development of vehicle door security using smart tag and fingerprint system," *International Journal of Engineering and Advanced Technology*, 2019.