Key management system for private car-sharing scenarios

Ana C. Hernández*†, Jordi Castellà Roca*, and Alexandre Viejo*
* Universitat Rovira i Virgili
Dept. of Computer Science and Mathematics
CYBERCAT-Center for Cybersecurity Research of Catalonia
UNESCO Chair in Data Privacy
Av. Països Catalans 26, 43007, Tarragona, Catalonia, Spain
anacristina.hernandez@estudiants.urv.cat, {jordi.castella, alexandre.viejo}@urv.cat

† SEAT, S.A
Autovía A-2, Km. 585
08760 Martorell, Spain
ana-cristina.hernandez@seat.es

***Abstract*—Car Sharing is an innovative transport concept that has emerged to contribute to efficient use of vehicles in urban areas that nowadays suffer from traffic congestion or parking unavailability. Several companies such as vehicle manufacturers have identified the business opportunities related with the car sharing market and realized the potential of providing this service to end-users under a private sharing scenario. However, remaining technical challenges related with the secure management of digital permissions that grant access to vehicles must be addressed. In this paper, we propose a key management system that enables car-sharing use cases, based on a central key management server and a distributed management of the PKI infrastructure. Our system out-stands by providing accountability on the actors and being resilient to internal attackers that attempt to impersonate system entities, provide false credentials to honest users or illegally take over a vehicle.**

***Index Terms*—Car-sharing, key management, digital key, digital access.**

I. Introduction

Nowadays, 54% of the world's population live in urban areas, and by 2030 this number is expected to reach 60% [1]. As a direct result of this situation, in the last years, urban mobility has become one of the toughest challenges that cities face [2]. In particular, traffic congestion, parking unavailability and commuting times are some of the most relevant problems to be addressed [3].

In this scenario, effective resource management is mandatory, and reducing the number of cars in the city by efficiently utilizing the available cars has been acknowledged to be an adequate approach [4], [5]. In this way, *car sharing* is an innovative transportation concept that has emerged as a solution for highly dense metropolis. It can be defined as a class of Mobility Services based on modern technology that enables access to car-based transportation without the consumer owning the physical asset [6]. Users of car sharing are able to access a pool of vehicles at any time, addressing their mobility needs at the same time that the average time of utilization of cars is optimized. Moreover, it allows users to benefit from the flexibility of car-based transportation without supporting all of its costs.

Nowadays, car sharing models basically follow the vehicle-rent approach, where a company (e.g., Drive Now[1] or Car2Go[2], among several others) offers a pool of vehicles to be shared among users. Several companies have identified the business opportunities of such an emergent Car Sharing market, which has increased from 0.35 million in 2006 to 4.94 million in 2014 [7] and is expected to reach 35 million users by 2021 [8]. Nevertheless, *private use-cases of car sharing* cannot be neglected. Car manufacturers have realized the potential of offering car sharing services to vehicle owners, addressing a private scenario where several family members share access to the same resource.

To guarantee the success of the car sharing approach for private car owners, vehicle access systems need to evolve from the traditional physical key-based access to allow the use of digital keys that can be remotely provisioned to the user and validated by the vehicle. Passing physical keys is a traditional way of car sharing, however, it is low efficient, inflexible and not consequent with the service values.

The inclusion of several communication interfaces like NFC, Bluetooth or GSM in vehicles and the increasing adoption in smartphones/wearables has enabled communication channels that can be used to allow shared access, playing an important role in the evolution of car sharing. However, the remaining challenge that car sharing systems face today concerns the security issues inherent to the management of digital keys that grant access permissions to vehicles. In this way, mechanisms to generate, transmit and revoke digital keys must be proposed to securely handle such sensitive resources in front of capable adversaries.

A. Related Work

Provision of digital access to vehicles based in cryptographic tokens has been addressed by Busol et al. in [9]. Vehicle owners are granted with access by means of an access token generated by the key server, based on pre-shared keys between the key server and the vehicle. On the other hand, delegated users must provide an additional token, generated

[1]https://www.drive-now.com/en
[2]https://www.car2go.com

978-1-5386-6358-5/18/$31.00 ©2018 IEEE

by the owner. Security of the delegation token relies on the owner keys not being compromised.

Token-based access has been also presented in [10], where authors have proposed a 2-factor authentication solution where users obtain and handle authentication factors separately, first through user registration and afterwards during vehicle booking. The first authentication factor consists of a user key that is used while authenticating against any vehicle. The second authentication factor is dependent on booking details.

Authors in [11] provide users with access tokens relying on shared secret in the vehicle. Besides providing a framework to delegate access to vehicles, this specific contribution addresses accountability of car-sharing systems in addition to the privacy requirements first introduced in [12].

Aforementioned solutions use symmetric key cryptography as a basis for security. Consequently, in a car-sharing environment where secrets must be shared with multiple vehicles or with multiple users, this approach could lead to increased damage when one the secrets gets compromised.

Our work is closely related to the work in [13]. Authors have proposed a PKI-based car-sharing scheme that does not rely on pre-shared secrets. However, our work provides convenient enhancements that avoid relying on large data strings or printed bar-codes to provide users with an ownership proof that might lead to fraud in case of unappropriated handling.

B. Contribution and plan of this paper

In order to fill the gap found in the current literature on this topic, in this paper we propose a framework to securely manage shared access to vehicles. We provide mechanisms to remotely generate and share access permissions in the private car-sharing use case.

The main contributions of this paper are the following:

- We define a system architecture to provide private car sharing functions based on distributed and scalable PKI management, where every entity is provided with differentiated key-pairs.
- We propose a key management system with the necessary functions for private car-sharing use cases. We address generation and sharing of access permissions, as well as ownership changes.
- We provide a security analysis on the proposed solution.

The remainder of this paper is organized as follows: Section II provides an overview on the system requirements. The model architecture is presented in Section III. Specification of protocols is provided in Section IV while the security analysis of the solution is discussed in Section V. Finally, conclusions and future work are proposed in SectionVI.

II. System Requirements

A. Functional Requirements

- *Delegation of keys and permissions*: the system must allow the generation of digital keys from owner to users. A digital key might be defined as the credential and the associated set of transactions that provide a user with access to a shared vehicle.
- *Offline Authentication:* bilateral authentication between the vehicle and the user should be possible without accessing the cloud.
- *System scalability:* the system must be able to scale and handle user requests quickly and efficiently.
- *User-oriented:* user-related processes must be easy to follow.

B. Security Requirements

- *Entity Authentication*: all actors involved in the system must be able to verify the identity of communicating parties.
- *Key authenticity*: only authorized entities can generate digital keys, they cannot be forged or falsified.
- *Key integrity*: digital keys cannot be manipulated by malicious users intending to extend their privileges or the service conditions.
- *Authorization*: vehicle resources are handled only by entities that hold the corresponding rights.
- *Secure channels*: authentic and confidential communication channels are established between authorized parties in order to protect sensitive data from being disclosed or modified.
- *Non-repudiation and accountability*: entities cannot deny their participation in a process. Transactions are traceable and imputable to an entity.

III. System Architecture

The system model is depicted in Figure 1 and consists of the following entities:

- *Key Management Server (KMS)* is a server or a compound of servers that act as central point of the system architecture and assist most of the interaction between the entities. It aims to:
  - Provide support in user-related processes (enrollment, registration of vehicle owner, transfer of ownership).
  - Validate and assist user requests such as the generation of digital keys to other users.

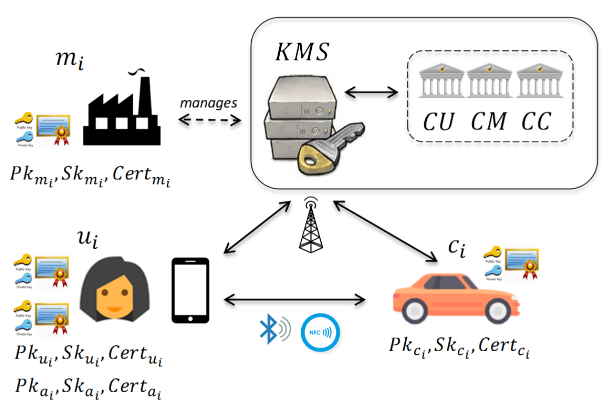

Figure 1. System architecture.

- *Car Manufacturer* M runs the *KMS*. Every instance (i.e. productive center) of M is denoted by m_i and possesses a key pair $\{Pk_{m_i}, Sk_{m_i}\}$ and a certificate $Cert_{m_i}$ issued by *CM*.
- *The car* c_i is equipped with a control module that manages access. It possesses a long-range communication interface (such as LTE) for the interaction with the *KMS* as well as short-range communication interfaces (such as BLE, NFC, WiFi) for the communication with smart devices. Every car c_i possesses a unique Vehicle Identification Number (VIN), a key pair $\{Pk_{c_i},\ Sk_{c_i}\}$ and a certificate $Cert_{c_i}$ issued by *CC*.
- *User* u_i enrolls the platform to use sharing system. Every user u_i holds unique credentials, including a key pair $\{Pk_{u_i},\ Sk_{u_i}\}$ and a certificate $Cert_{u_i}$ issued by *CU*. Moreover, access credentials are provided when the user gets registered as a vehicle owner or when he obtains access permission from another owner. Access credentials consist of a key pair $\{Pk_{a_i},\ Sk_{a_i}\}$ and a certificate $Cert_{a_i}$ issued by *CU* and associated to a car c_i. The extensions contained in the $Cert_{a_i}$ allow the system to identify u_i as an owner or as an authorized user allowed to access the vehicle.
- *The App* is a software intended to be run in smart devices. It allows users to interact with the system entities.
- *Three Certification Authorities* named CU, CC and CM. Certificate management tasks are distributed in three different Certification Authorities (CA), in order to decrease the complexity of certificate management functions and guarantee the scalability of the system.
  1) The *User Certification Authority - CU* attest the identity of users and their access permissions.
  2) The *Car Certification Authority - CC* issues the certificates of every vehicle.
  3) The *Car manufacturer Certification Authority - CM* issues and manages the certificates of the car manufacturing center.

  Every CA is able to issue a certificate status response ϵ_{a_x}, which is provided after a secure connection is established, according to OCSP Stapling specification [14].

  The structure of every CA is shown in 2. The depth of the tree is denoted by l, while the width of the tree at leaf level is denoted by s. The top of the CA is kept offline, while only leaf nodes are connected to the network. Therefore, if a leaf node gets compromised, the remaining nodes are not affected.

IV. Protocol specification

A set protocols for key management in a vehicle sharing platform are presented in this section.

First, the vehicle is set-up with a register that declares the manufacturer as the current vehicle owner. When the vehicle is sold (from manufacturer to 1st owner, or from 1st to 2nd owner), the digital owner of the vehicle changes and this information is updated in the car. Aforementioned transactions are registered in the property chain $\{\delta_0^{'},\ \cdots,\ \delta_n^{'}\}$, a linked set

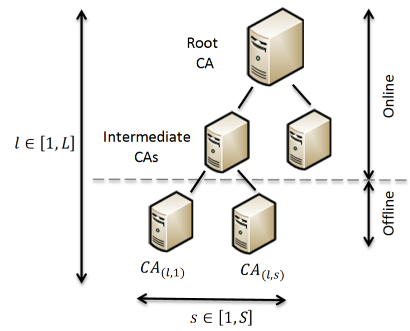

Figure 2. Structure of Certification Authorities CU, CC and CM.

of registers that contains information about the current vehicle owner.

When the owner shares her vehicle with another user, an access chain $\{\rho_0^{'},\ \cdots,\ \rho_n^{'}\}$ is created. It allows to validate user authorization over a vehicle and the conditions that limit the permission. Both chains contain authentic registers that provide integrity to system communication and bind system transactions to a given entity.

In order to identify the operations performed by an entity in each one of the chains, a set of flags is defined in Table I. Table II comprises the symbols used in current section.

A. System set-up

In order to perform cryptographic operations, the vehicle must be provided with specific cryptographic material and data. It is assumed that his process is executed in a secure environment such as a factory, during vehicle assembly.

The car manufacturer provides each vehicle c_i with (i) asymmetric key pair Pk_{c_i}, Sk_{c_i}, (ii) vehicle certificate $Cert_{c_i}$, (iii) a repository with available Certification Authorities $CU_{l,s}$,

Table I
System Flags

Flag	Description	Flag	Description
F_0	Provide initial data.	F_1	Start owner transfer.
F_2	Accept vehicle ownership.	F_3	End owner transfer.
F_4	Authorize new owner.	F_5	Update owner in vehicle.
F_6	Start access authorization.	F_7	Accept access authorization.
F_8	End access authorization.		

Table II
System symbols

Symbol	Description
I	Personal data of user.
β	OTP identification token.
α	Validation Code.
η_i	Phone number of user u_i.
P_i	Profile of user u_i.
$\{Pk_{x_i}, Sk_{x_i}\}$	Asymmetric key pair of x_i.
CSR_{x_i}	Certificate Signing Request of x_i.
$Cert_{x_i}$	Certificate of x_i.
ϵ_{x_i}	Certificate Status of $Cert_{x_i}$
$CRL_{CA_{l,s}}$	Certificate Revocation List of $CA_{l,s}$.

$CM_{l,s}$, $CC_{l,s}$ and (iv) the initial ownership transaction $\delta_0^{'}$= {γ, δ_0, $Cert_{m_i}$}. Where δ_0 = $Sk_{m_i}(\gamma)$, and γ={F_0, t, $Cert_{m_i}$}. This register states the vehicle manufacturer as the current owner of the vehicle

B. User Enrollment

Users must enroll the application and obtain personal credentials to later authenticate against the server.

- The user u_i performs the following steps:
  1) Introduces personal information *I*={*Name, Surname, Address,* η_i*, ID*} in the App.
  2) Generates {Pk_{u_i}, Sk_{u_i}} and CSR_{u_i}.
  3) Establishes a secure communication channel with $CU_{l,s}$ and sends {*I*, CSR_{u_i}}.
- The Certification Authority $CU_{l,s}$ performs the following steps:
  1) Verifies CSR_{u_i}.
  2) Generates a code α valid during a certain time interval and a token β.
  3) Sends β via App and α to u_i using a second communication channel (e.g. SMS using η_i, post using address, etc.).
- The user u_i establishes a secure communication channel with $CU_{l,s}$ after receiving α and before expiration of ζ, and sends α and β.
- The Certification Authority $CU_{l,s}$ verifies α and β; if both are valid issues $Cert_{u_i}$ and sends it to u_i.
- The user u_i receives and verifies $Cert_{u_i}$, stores properly Pk_{u_i}, Sk_{u_i} and $Cert_{u_i}$.

C. Owner Registration

When the vehicle c_i is purchased, the owner u_i must obtain the digital credentials that confer total permissions over the vehicle from the car manufacturer m_i. This requires the owner to be enrolled in the platform (See section IV-B).

- The car maker m_i performs the following steps:
  1) Obtains I and $Cert_{u_i}$ by means of η_i.
  2) Creates an authorization Ψ_i to obtain a new key pair for u_i, where Ψ_i={F_1, c_i, $Cert_{u_i}$, P_i}. The profile P_i contains the set of attributes assigned to the vehicle owner u_i.
  3) Sends $\delta_1^{'}$ to u_i through the *KSM*, where $\delta_1^{'}$= {Ψ_i, δ_1, $Cert_{m_i}$} and δ_1= $Sk_{m_i}(\Psi_i)$.
- The user u_i executes the following steps after she has received $\delta_1^{'}$:
  1) Verifies that $\delta_1^{'}$ is valid.
  2) Generates a new key pair {Pk_{a_i}, Sk_{a_i}} and computes CSR_{a_i}.
  3) Computes $\delta_2^{'}$={CSR_{a_i}, δ_2, $Cert_{u_i}$}, where δ_2= $Sk_{u_i}(\delta_1^{'}, CSR_{a_i})$; sends $\delta_2^{'}$ and $\delta_1^{'}$.
- The Certification Authority $CU_{l,s}$ performs the following steps:
  1) Verifies $\delta_2^{'}$ using $Cert_{u_i}$, $\delta_1^{'}$ using $Cert_{m_i}$ and CSR_{a_i}.
  2) Verifies that the certificate $Cert_{u_i}$ included in $\delta_1^{'}$ is the same that has been used to sign δ_2.
  3) If previous verifications are successful, then the $CU_{l,s}$ issues a new certificate $Cert_{a_i}$ with the extensions *Owner* and c_i, and sends it to u_i.
- The user u_i performs the following steps:
  1) Verifies the signature and extensions of $Cert_{a_i}$ (including c_i and *Owner*) using $\delta_1^{'}$.
  2) Computes $\delta_3^{'}$ and sends it to m_i, where $\delta_3^{'}$= {F_2, δ_3, $Cert_{a_i}$}, and δ_3=$Sk_{a_i}(\delta_2^{'}, F_2)$.
- The car manufacturer m_i performs the following steps:
  1) Verifies $Cert_{a_i}$ and the extensions c_i and *Owner* using $\delta_1^{'}$. Moreover, verifies the chain {$\delta_1^{'}$, $\delta_2^{'}$, $\delta_3^{'}$} and the flag F_2.
  2) Computes $\delta_4^{'}$ and sends to u_i, where $\delta_4^{'}$= {F_3, δ_4, $Cert_{m_i}$}, and δ_4=$Sk_{m_i}(\delta_3^{'}, F_3)$.
- The user u_i verifies and stores the chain {$\delta_1^{'}$, $\delta_2^{'}$, $\delta_3^{'}$, $\delta_4^{'}$}.

D. Change of vehicle owner

When the current owner u_i sells the vehicle c_i to u_j the new proprietor must obtain digital credentials that grant total permissions over the vehicle. This process requires u_j to be enrolled and that the current owner u_i authorizes the emission of credentials for the new proprietor.

- The owner u_i performs the following steps:
  1) Obtains the phone-number η_j.
  2) Obtains $Cert_{u_j}$ using η_j and creates a transfer authorization Ψ_j, where Ψ_j={F_1, c_i, $Cert_{u_j}$, P_j}.
  3) Sends $\delta_z^{'}$ to u_j through the *KSM*, where $\delta_z^{'}$= {Ψ_j, δ_z, $Cert_{a_i}$} and δ_z= $Sk_{a_i}(\delta_{z-1}^{'}, \Psi_j)$. The register $\delta_{z-1}^{'}$ corresponds to the last element of the ownership chain. By using $Cert_{a_i}$, it is possible to associate the user u_i with the vehicle c_i.
- The user u_j executes the following steps after she has received $\delta_z^{'}$:
  1) Verifies that the digital signature δ_z is valid.
  2) Verifies that the profile P_j contains the properties, permissions and vehicle functionality of previous owner u_i.
  3) Generates {Pk_{a_j}, Sk_{a_j}} and CSR_{a_j}.
  4) Computes $\delta_{z+1}^{'}$={CSR_{a_j}, δ_{z+1}, $Cert_{u_j}$}, where δ_{z+1}= $Sk_{u_j}(\delta_z^{'}, CSR_{a_j})$.
  5) Sends $\delta_{z+1}^{'}$ and $\delta_z^{'}$ to m_i.
- The car maker m_i does the following steps:
  1) Verifies $\delta_z^{'}$ using $Cert_{a_i}$ and $\delta_{z+1}^{'}$ using $Cert_{u_j}$.
  2) Verifies that u_i is the owner of c_i by means of $Cert_{a_i}$; verifies that P_j is valid according to P_i. The profile of the new owner must contain the same attributes of the previous one.
  3) Creates a new certificate authorization $\Psi_j^{'}$={F_4}.
  4) Sends { $\delta_z^{'}$, $\delta_{z+1}^{'}$, $\delta_{z+2}^{'}$ } to $CU_{l,s}$ and u_j, where $\delta_{z+2}^{'}$= {$\Psi_j^{'}$, δ_{z+2}, $Cert_{m_i}$} and δ_{z+2}= $Sk_{m_i}(\delta_{z+1}^{'}, \Psi_j^{'})$.

- The Certification Authority $CU_{l,s}$ performs the following steps:
  1) Verifies $\delta^{'}_{z+1}$ using $Cert_{u_j}$, $\delta^{'}_{z+2}$ using $Cert_{m_i}$ and CSR_{a_j}.
  2) Verifies that $Cert_{u_j}$ (included in $\delta^{'}_z$) was used to sign $\delta^{'}_{z+1}$.
  3) Revokes $Cert_{a_i}$, issues $Cert_{a_j}$ with the extensions *Owner* and c_i, and sends it to u_j.
- The user u_j performs the following steps:
  1) Verifies the digital signature and extensions of the certificate $Cert_{a_j}$, including c_i and *Owner*, using $\delta^{'}_z$.
  2) Computes $\delta^{'}_{z+3}$ and sends it to m_i, where $\delta^{'}_{z+3}$= $\{F_2$, δ_{z+3}, $Cert_{a_j}\}$, and δ_{z+3}=$Sk_{a_j}(\delta^{'}_{z+2}$, $F_2)$.
- The car manufacturer m_i performs the following steps:
  1) Verifies the chain $\{\delta^{'}_z$, $\delta^{'}_{z+1}$, $\delta^{'}_{z+2}$, $\delta^{'}_{z+3}\}$, $Cert_{a_j}$, its extensions c_i, *Owner* and the flag F_2 using $\delta^{'}_z$.
  2) Computes $\delta^{'}_{z+4}$, where $\delta^{'}_{z+4}$= $\{F_3$, δ_{z+4}, $Cert_{m_i}\}$, and δ_{z+4}=$Sk_{m_i}(\delta^{'}_{z+3}$, $F_3)$.
  3) Sends $\delta^{'}_{z+4}$ to u_i and u_j.
- The user u_j[3] verifies and stores the chain $\{\delta^{'}_z$, $\delta^{'}_{z+1}$, $\delta^{'}_{z+2}$, $\delta^{'}_{z+3}\}$.

E. Update the car's owner in the vehicle

When there is a new proprietor, the ownership chain in the vehicle must be updated. This occurs after performing the steps in IV-C or IV-D. In general, if the vehicle is on-line, communication is directly established between the *KMS* and the vehicle c_i. Otherwise, the user u_j relays the messages to c_i.

- When there is a connection between the car and the cloud (*KMS*), M_i performs the following steps[4]:
  1) Establishes a secure connection with the car c_i and sends the chain $\{\delta^{'}_z, \ldots, \delta^{'}_{z+k}\}$.
- The car c_i does the following steps:
  1) Verifies $\delta^{'}_z$ using $\delta^{'}_{z-1}$. Notice that the vehicle possesses an ownership chain ending in δ_{z-1}, which corresponds to the last element of the property chain.
  2) Verifies the rest of the chain $\{\delta^{'}_z, \ldots, \delta^{'}_{z+k}\}$ and the status of the certificates.
  3) If the previous verifications are correct, the owner of c_i is updated to u_j and the new chain $\{\delta^{'}_z, \ldots, \delta^{'}_{z+k}\}$ is stored in the car.
  4) Computes $\delta^{'}_{z+k+1}$ and sends it to u_j and m_i[5], where $\delta^{'}_{z+k+1}$= $\{F_5$, δ_{z+k+1}, $Cert_{c_i}\}$, and δ_{z+k+1}=$Sk_{c_i}(\delta^{'}_{z+k}$, $F_5)$.
- The new owner u_j receives $\delta^{'}_{z+k+1}$ and verifies the received chain. Finally, m_i verifies the information and updates the chain.

[3] u_i does the same operations

[4] In case the vehicle is offline, the steps are performed by u_j.

[5] If there is no connection with the cloud $\delta^{'}_{z+k+1}$ is sent to u_j and then forwarded to m_i.

F. Vehicle Sharing with terms of service

The vehicle owner u_j authorizes to share the car under certain terms of service with the user u_x. In order to obtain the access credentials, the user u_x must hold the personal credentials obtained in Section IV-B.

- When the vehicle owner u_j intends to share the vehicle c_i with the user u_x, she performs the following steps:
  1) Obtains the phone-number η_x and therefore $Cert_{u_x}$.
  2) Creates an authorization χ to share c_i with u_x, where χ=$\{F_6$, c_i, $Cert_{u_x}$, $P_x\}$. u_j defines in P_x a set of attributes, including the duration of this permission, which will define the validity time of the access certificate. Moreover, P_x might comprise (i) limitations in the actions allowed, such as: locking, unlocking or starting the vehicle; (ii) activation or deactivation of the navigation/driving assistance systems; or (iii) limitations on the driving area or the maximum speed. Implementation of such features is out of the scope of this work.
  3) Sends $\rho^{'}_0$ to u_x through the *KMS*, where $\rho^{'}_0$= $\{\chi$, ρ_0, $Cert_{a_j}\}$ and ρ_0= $Sk_{a_j}(\chi)$.
- The user u_x executes the following steps after receiving $\rho^{'}_0$:
  1) Verifies that the digital signature ρ_0 is valid, and validates that P_x corresponds with the agreement between u_j and u_x.
  2) Generates $\{Pk_{a_x}$, $Sk_{a_x}\}$ and CSR_{a_x}.
  3) Computes $\rho^{'}_1$ and sends it together with $\rho^{'}_0$ to $CU_{l,s}$, where $\rho^{'}_1$=$\{CSR_{a_x}$, ρ_1, $Cert_{u_x}\}$, and ρ_1= $Sk_{u_x}(\rho^{'}_0$,$CSR_{a_x})$.
- The Certification Authority $CU_{l,s}$ performs the following steps:
  1) Verifies $\rho^{'}_1$, $\rho^{'}_0$, the status of $Cert_{a_j}$ and validates that the current owner of c_i, indicated by a_j is u_j.
  2) Verifies CSR_{a_x} and validates that $Cert_{u_x}$ (included in $\rho^{'}_0$) was used to sign ρ_1.
  3) Issues $Cert_{a_x}$, with extensions *Access*, c_i and sends it to u_x.
- The user u_x performs the following steps:
  1) Verifies $Cert_{a_x}$ and its extensions c_i and *Access*, using $\rho^{'}_0$.
  2) Computes $\rho^{'}_2$ and sends it to u_j, where $\rho^{'}_2$= $\{F_7$, ρ_2, $Cert_{a_x}\}$, and ρ_2=$Sk_{a_x}(\rho^{'}_1$, $F_7)$.
- The car's owner u_j performs the following steps:
  1) Verifies $Cert_{a_x}$ and its extensions c_i and *Access*, using $\rho^{'}_0$.
  2) Verifies the chain $\{\rho^{'}_0$, $\rho^{'}_1$, $\rho^{'}_2\}$ and the flag F_7.
  3) Stores $Cert_{a_x}$, computes $\rho^{'}_3$ and sends it to u_x, where $\rho^{'}_3$= $\{F_8$, ρ_3, $Cert_{a_i}\}$, and ρ_3=$Sk_{a_j}(\rho^{'}_2$, $F_8)$.
- The user u_x verifies and stores the chain $\{\rho^{'}_0$, $\rho^{'}_1$, $\rho^{'}_2$, $\rho^{'}_3\}$.

G. Vehicle use according to the terms of service

In order to use the vehicle c_i according to the terms of service, the user u_x must hold $Cert_{a_x}$ associated to the vehicle and authorized by the owner u_j. In addition, the certificate status response ϵ_{a_x} is used to verify that the certificate has not been revoked when the user intends to access the vehicle.

The user u_x must be able to validate the identity of c_i contained in $Cert_{c_i}$ by retrieving $CRL_{CC_{l,s}}$ and verify that $Cert_{c_i}$ has not been revoked.

- Seamlessly, the App running in the mobile phone of u_x performs the following steps in a frequent basis:
    1) Obtains the *certificate status response* ϵ_{a_x} corresponding to $Cert_{a_x}$.
    2) Obtains the $CRL_{CC_{l,s}}$ required to verify the $Cert_{c_i}$ of c_i issued by $CC_{l,s}$.

  Only when u_x is in close proximity to c_i, the mobile application awakes and allows the user to interact with the vehicle. The minimum proximity range depends on the communication technology used between u_x and c_i. Distance computations can be performed by measuring round-trip or through distance bounding protocols [15], [16]. Once the user is in communication range, the application requests confirmation to establish a secure connection with c_i. This action requires that u_x introduces a PIN, fingerprint, pattern, etc. A secure connection with the car c_i is established after confirmation.
- After the secure connection is established and c_i has received ϵ_{a_x} from u_x, c_i does the following steps:
    1) Verifies that the certificate $Cert_{a_x}$ has not expired and has not been revoked.
    2) Verifies whether the certificate has the extension *Owner* or *Access*.
    3) If the certificate has *Access* extension:
        a) Checks if there exists a chain $\{\rho_0', \rho_1', \rho_2', \rho_3'\}$ where ρ_2' contains $Cert_{a_x}$.
        b) If the previous verification fails, sends a request to obtain the chain $\{\rho_0', \rho_1', \rho_2', \rho_3'\}$.
        c) Verifies and stores the chain.
        d) Verifies that $Cert_{a_x}$ has the extension c_i.
    4) If the certificate has *Owner* extension:
        a) Verifies that $Cert_{a_x}$ has the extension c_i.
    5) If previous verifications are successful, provides access and sets up the vehicle functions according to P_x included in ρ_0'.

V. Security Analysis

We next detail the adversary model and the possible attacks the proposed scheme has to be robust against. Those attacks are related to the security requirements that must be fulfilled by our scheme and that were introduced in Section II.

A. Adversary model

Our attacker model considers both internal and external adversaries, focusing on internal attackers who can be a *vehicle owner* or a *vehicle user*. External adversaries comprise any external entity which is not registered in the system. In any case, we assume that adversaries' computational power do not permit them to break current computationally secure cryptosystems.

Regarding the other entities of the proposed system, the *Car Manufacturer* and the *Certification Authorities* are fully trusted; therefore, they are not considered in this section.

B. System's behavior against the considered attacks

We next introduce the relevant attacks and how the proposed solution deals with them.

1) Impersonating a Certification Authority: The communication between users and the different CAs involved in the proposed protocol are secured by means of TLS/SSL, in which certificates X.509 are used to authenticate the CAs and prevent this attack from succeeding as long as the corresponding cryptographic material and the whole TLS/SSL process remains secure.

2) Impersonate a legitimate system's user (i.e., car owner or car user): An adversary A who wants to impersonate a certain user U of the system should interfere in the "User Enrollment" step and link her public key PK_A with the identity of the target user I_U, discarding the legitimate public key from U in the CSR, and retrieving in the subsequent response a valid certificate issued by the CA.

The proposed system provides two countermeasures: i) communications between users and CAs are protected by means of TLS/SSL tunnels that prevent man-in-the-middle attacks; and ii) the issuing CA performs a two-factor authentication in which the user must provide a one-time password received through a second channel (i.e., a SMS). Consequently, in order to succeed in this attack an adversary ought to be capable of breaking the security of TLS/SSL or she should have control of the second channel used in this process. In the rest of the steps of the proposed protocol, users are protected from impersonation by means of the certificate received in the "User Enrollment" process. As long as the cryptographic material of the users remains safe, adversaries will not be able to perform this attack.

3) Illegally obtain valid credentials to own a vehicle: An adversary who wants to obtain the credentials for owning someone else's car should perform her attack during the protocols "Owner Registration", "Change of Vehicle Owner" or "Update the Car's Owner". In all the cases, in order to succeed in this attack she should be able to introduce her certificate and key pair in the CSR sent to the CA. In case the CA issues a certificate that validated the adversary's cryptographic material, then she will get the ownership on the vehicle. The CSR includes a data structure containing the certificate of the next owner and it is digitally signed by the current owner (i.e., the car maker or the current user who owns the vehicle). As a result, the adversary should substitute the certificate of the legitimate owner with her own certificate, and she should be capable of generating a new valid digital signature with the secret key of the current owner. As long as the cryptographic material of the current owner remains safe, the adversary cannot perform this attack.

4) Illegally obtain valid credentials to use a vehicle: An adversary who wants to obtain the credentials for using a car or its services should perform her attack during the protocol “Vehicle Sharing”. As in the former attack, in order to succeed in this attack the adversary should be able to introduce her certificate and key pair in the CSR sent to the CA; needing the owner of the vehicle to digitally sign her cryptographic material. As long as the secret key of the current owner remains safe, the adversary cannot perform this attack.

5) Provide fake credentials to honest users: An adversary may try to sell fake credentials to honest users getting economical reward through this process. In case of fake transfer of ownership, the protocol “Change of vehicle owner” includes a step in which the car manufacturer is directly contacted by the user who receives the credentials and verifies the identity of the owner of the vehicle that initiates the transfer by means of her cryptographic material. In this step, the car maker validates whether the user who is trying to transfer these credentials is the real owner or not. As long as the cryptographic material of the real owner remains safe, the adversary cannot perform this attack. The same conclusion is valid when the adversary attempts to transfer fake user access permissions (related to the protocol “Vehicle Sharing”); however, in this case, the verification of the identity of the vehicle’s owner is performed by the CA.

6) Use a certain service of a vehicle without owning the right credentials: An adversary may try this attack following two different approaches: i) she may have a set of legitimate but outdated or revoked credentials that gave her access to a certain service in the past; and ii) she may try to generate fake credentials to gain access to a certain service. Vehicles perform access control by verifying the certificate and extension issued by the CA. Specifically, the vehicle checks whether this certificate states the use of the requested service and it also verifies whether it has been revoked or it has expired. Consequently, in the first case, the adversary should be able to modify the expired certificate issued by the CA in order to seem updated; this would require the knowledge of the secret key used by the CA. The same knowledge is required to change the status of a revoked certificate. Regarding the second case, the adversary should generate a fake authorization that contains the specific service to be used, which should be signed by the vehicle owner; therefore, as long as the cryptographic material of the vehicle owner remains safe, the adversary cannot perform this attack.

VI. Conclusions

This work proposed a system to provide key management functions suitable for private car sharing scenarios where a vehicle owner is able to provide users with access permissions (or digital keys) over its vehicle. Our model is based on a PKI structure with OCSP stapling, and a central key management server that supports most of the interactions between the system users and the vehicle. The model is characterized by providing unique credentials to each involved actor and avoids the use of same keys to access different resources. Moreover, management of the PKI is distributed in three CA structures in order to achieve proper load distribution and system scalability. A set of protocols that enables car-sharing services was proposed. The solution is able to achieve authenticity and integrity of the interaction between actors, provided by means of transaction chains as well as accountability on user’s transactions. Moreover, it provides mechanisms to deal with external and internal attackers that try to impersonate system entities, illegally obtain permissions over a vehicle or try to commit fraud by providing fake permissions over vehicles out of their ownership. Future research lines will be devoted to propose further key management operations including revocation of permissions, renovation of cryptographic material or processes to recover credentials of users that have lost their mobile devices.

References

[1] E. . S. Affairs, “The world’s cities in 2016: Data booklet,” tech. rep., 2016.

[2] F. Van Auenhove, O. Korniichuk, L. Dauby, and J. Pourbaix, “Future of urban mobility 2.0,” tech. rep., 2014.

[3] K. Zavitsas, I. Kaparias, and M. Bell, “Transport problems in cities,” Coordination Of Network Descriptors for Urban Intelligence Transport Systems (CONDUITS) Deliberable 1.1, Imperial College London, September 2010. 7th Framework Programme.

[4] L. Fulton, J. Mason, and D. Meroux, “Three Revolutions in Urban TRANSPORTATION,” Supported by Climate Works Foundation, William and Flora Hewlett Foundation, Barr Foundation, UCDAVIS SUSTAINABLE TRANSPORTATION ENERGY PATHWAYS of the Institute of Transportation Studies, Institute for Transportation & Development Policy, May 2017.

[5] ITF, “Shared mobility: Innovation for liveable cities,” International Transport Forum Policy Papers 21, OECD Publishing, 2016.

[6] S. L. Vine, A. Zolfaghari, and J. Polak, “Carsharing: Evolution, challenges and opportunities,” tech. rep., Centre for Transport Studies, Imperial College London, 2014.

[7] Frost and Sullivan, “Strategic insight of the global carsharing market,” tech. rep., 2014.

[8] J. Bert, B. Collie, M. Gerrits, and G. Xu, “What’s ahead for car sharing. the new mobility and its impact on vehicle sales.” 2016.

[9] 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*, CODASPY ’13, (New York, NY, USA), pp. 233–242, ACM, 2013.

[10] 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*, CODASPY ’17, (New York, NY, USA), pp. 349–360, ACM, 2017.

[11] I. Symeonidis, A. Aly, M. A. Mustafa, B. Mennink, S. Dhooghe, and B. Preneel, *SePCAR: A Secure and Privacy-Enhancing Protocol for Car Access Provision*, pp. 475–493. Cham: Springer International Publishing, 2017.

[12] I. Symeonidis, M. A. Mustafa, and B. Preneel, “Keyless car sharing system: A security and privacy analysis,” in *2016 IEEE International Smart Cities Conference (ISC2)*, pp. 1–7, Sept 2016.

[13] M. Elkins, “A peer-to-peer approach to digital key sharing for vehicle access control.” Conference lecture presented at Embedded Security in Cars (escar) Conference, 2017.

[14] Y. N. Pettersen, “The Transport Layer Security (TLS) Multiple Certificate Status Request Extension.” RFC 6961, June 2013.

[15] S. Brands and D. Chaum, *Distance-Bounding Protocols*, pp. 344–359. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994.

[16] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji, “Detecting relay attacks with timing-based protocols,” in *Proceedings of the 2Nd ACM Symposium on Information, Computer and Communications Security*, ASIACCS ’07, (New York, NY, USA), pp. 204–213, ACM, 2007.