Trustworthy Parking Communities: Helping Your Neighbor to Find a Space

Julian Timpner, *Student Member, IEEE*, Dominik Schürmann, *Student Member, IEEE*, and Lars Wolf, *Member, IEEE*

Abstract—Cooperation between vehicles facilitates traffic management, road safety and infotainment applications. Cooperation, however, requires trust in the validity of the received information. In this paper, we tackle the challenge of securely exchanging parking spot availability information. Trust is crucial in order to support the decision of whether the querying vehicle should rely on the received information about free parking spots close to its destination and thus ignore other potentially free spots on the way. Therefore, we propose Parking Communities, which provide a distributed and dynamic means to establish trusted groups of vehicles helping each other to securely find parking in their respective community area. Our approach is based on high-performance state-of-the-art encryption and signature algorithms as well as a well-understood mathematical trust rating model. This approach allows end-to-end encrypted request-response communications in combination with geocast and can be used as an overlay to existing vehicular networking technologies. We provide a comprehensive comparison with other security architectures and simulation results showing the feasibility of our approach.

Index Terms—VANET, vehicular networks, parking search, trust management, reputation, security, identity management

1 Introduction

Modern vehicles are equipped with an array of sensor systems and assistance functions, which can greatly enhance driving comfort and safety. However, in order to maximize their effect, these disparate systems need to cooperate with each other. Hence, vehicles do not have to rely on on-board sensors only, but can acquire further information from other systems, both mobile and fixed, in their environment. As an example, consider a scenario where a driver on his way home from work is interested in a free parking spot on his downtown home street. The vehicle thus uses a geocast (a specialized form of multicast, in which destination nodes are addressed by their geographic location instead of by their IDs) to send a corresponding query into the destination area. Here, vehicles use their sensor systems to gather information about their surroundings, such as distance to the closest objects (e.g., cars), and respond to the query originator. Thus, the vehicle can advise the driver where to find parking, preferably close to his home location.

In the example, trust is crucial in order to support the decision of whether the query originator should rely on the received information about free parking spots close to his destination and thus ignore other potentially free spots on the way. This bears the risk of learning that there is no available spot at all in the destination area, and the previously ignored spots might be taken by then. Conversely, trust alleviates prioritizing incoming queries and can provide an incentive to help other vehicles, such that they will also be provided with inquired information, in a tit-for-tat manner. Moreover, attackers are likely to try to gain an advantage, e.g., by providing false data to keep parking spots to themselves or by intercepting parking spot availability information in order to reach free spots earlier than competing drivers. Unfortunately, there is no easy way to decide which vehicles to trust, or more specifically, to what extent. Even if a trusted third party (TTP) exists, for instance in form of a certificate authority (CA) providing pseudonym certificates [1], it cannot necessarily verify the trustworthiness of vehicle responses. In order to do so, it would require trusted sensors at each parking spot throughout the city, which is expensive [2] and requires infrastructure networking support.

We thus propose, design, implement, and evaluate the concept of Parking Communities, which, in the style of good neighborly help, provide a distributed and dynamic means to establish trusted groups of vehicles helping each other to find parking in their respective community area. Our approach is based on high-performance state-of-the-art encryption and signature algorithms, in particular elliptic curve cryptography (ECC), as well as a well-understood mathematical trust rating model.

1.1 Contributions

In this paper, we present the design, implementation and evaluation of Parking Communities, a novel trust management for vehicular parking applications without reliance on a central TTP or road-side units (RSU). Its novel features include a distributed trust model for parking applications as well as encrypted and signed request-response communication in combination with geocast. It thereby achieves protection against impersonation, Sybil attacks, interception and tampering despite its distributed design. Further, it can be used as an overlay to existing vehicular networking technologies [1], [3], thus benefiting from established security

- *The authors are with the Institute of Operating Systems and Computer Networks, Technische Universität Braunschweig, 38106 Braunschweig, Germany. E-mail: {timpner, schuermann, wolf}@ibr.cs.tu-bs.de.*

Manuscript received 29 June 2014; revised 16 Apr. 2015; accepted 21 Apr. 2015. Date of publication 28 Apr. 2015; date of current version 20 Jan. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TDSC.2015.2427838

mechanisms, e.g., pseudonym certificates for anonymity and location privacy. We give a detailed analysis of attack scenarios and describe our implementation of the proposed security architecture in IBR-DTN [4], an open source RFC 5050 [5] implementation. We further provide a comprehensive evaluation in terms of a comparative analysis with other key and trust management protocols and simulation results.

1.2 Outline

The remainder of this paper is structured as follows. Section 2 discusses related work in the field of key and trust management in vehicular networks. The proposed Parking Community concept is introduced in Section 3. Attack scenarios on Parking Communities and their mitigations are presented in Section 4. Section 5 describes a prototypical implementation in an overlay network based on IBR-DTN. We analyze the protocol in comparison to existing solutions in Section 6, which can also serve for balancing the implementation tradeoffs of Parking Communities. We provide simulation results in Section 7. The paper concludes in Section 8.

2 Related Work

This section provides a short introduction to cryptographic fundamentals, such as ECC. Related work on vehicular key and trust management is discussed. A detailed comparison of how our key and trust management relates to existing ones can be found in Section 6.

2.1 ECC Fundamentals

ECC is a recognized cipher for vehicular networks and is already employed by the IEEE 1609.2 [3] and ETSI (TS 103 097) standards. From a theoretical perspective, ECC is based on the difficulty to solve the elliptic curve discrete logarithm problem (ECDLP) [6]. Modern representatives of ECC signature algorithms are the elliptic curve digital signature algorithm (ECDSA) [7] and edwards-curve digital signature algorithm (EdDSA) [8]. In most cases, ECC is not directly used to encrypt messages; rather, the peers agree on a session key using key agreement protocols, such as Diffie-Hellman (DH) [9].

2.2 Key Agreement Fundamentals

In addition to the DH key agreement based on the discrete logarithm problem, there also exist ECC variants, which require a smaller key size resulting in less energy, memory, and bandwidth consumption. DH-based key agreement protocols are designed for synchronous communications as opposed to the asynchronous elliptic curve integrated encryption scheme (ECIES). Since end-to-end connectivity cannot be guaranteed in vehicular networks and the number of roundtrips should thus be minimized, the asynchronous ECIES is more feasible in this context.

2.3 Trust in Vehicular Networks

There is an urgent need to assess the quality of information received in vehicular networks, lest a node reports false or inaccurate information to gain an advantage, e.g., allegedly congested roads in the hope that other vehicles avoid them and thus clear the path. Hence, the notion of trust among nodes is an important issue. Trust allows vehicles to detect dishonest and malicious data and to give incentives for honest and altruistic behavior.

There is a rich literature on trust models, which is why we do not aim to provide a comprehensive summary here, but instead refer the interested reader to the excellent surveys on trust management in vehicular networks [10], [11]. In this paper, we focus on self-organizing trust models which do not rely on an online connection to a security infrastructure in order to retrieve trust ratings (though a *key* management infrastructure can be used to achieve accountability, as described in Section 2.4). Instead, nodes form trust relationships directly with each other. These models can be classified into entity-oriented, data-oriented, and hybrid trust models. Entity-oriented trust models [12] focus on modeling the trustworthiness of nodes, but typically do not evaluate the trustworthiness of the data itself. This issue is addressed by data-oriented models. Raya et al. [13], for instance, use several decision logics, such as Bayesian inference and Dempster-Shafer theory to determine the level of trust that can be put in the received data. Vinel et al. [14] evaluated the effects on the decision delay when deploying a majority consensus algorithm to decide upon safety messages. They were able to show that a majority consensus works in practice, while decision delays should not exceed 6 seconds. A drawback of these approaches and, typically, of data-oriented models in general, is that only ephemeral trust in data is established, but no long-term trust relationships between nodes are formed. Hybrid trust models combine both aforementioned approaches and model the trustworthiness of nodes and use the result to evaluate the reliability of received data. Patwardhan et al. [15], for instance, determine a node's reputation by validating its data, which is similar to the approach in Parking Communities. Yet, the authors assume that certain nodes are preauthenticated and thus provide inherently trustworthy data. Parking Communities differ in that they do not assume any inherently trusted nodes. Instead, trust is only established by actually and physically validating received data. Similar to our approach, Park et al. [16] propose to make use of vehicles' daily commute routine to build up long-term reputation. The proposed system, however, relies heavily on support from roadside infrastructure, which we consider impractical.

To the best of our knowledge, we are the first to investigate a hybrid trust model with physical verification and no additional infrastructure support in the context of parking detection applications to build trusted communities.

2.4 Key Management

To allow for long-term reputation, accountability in form of non-repudiable key-identity bindings is vital. Common key management standards for vehicular communication are based on traditional Public Key Infrastructures (PKIs), subdivided into CA regions and extended with pseudonym certificates [1], [3], [17], [18]. RSU are introduced as additional infrastructure for communication between vehicles and central services, such as pseudonym CAs. Key pairs are usually generated on the nodes themselves, and the binding of a key pair to a node's identity is verified by a CA. Certificates serve as a proof of this binding and can be verified by any node in the network. IEEE 1609.2 [3], for instance, defines the format of security messages and uses anonymous public keys to sign and verify messages and short-lived anonymous certificates to automatically revoke keys. Studer et al. [17] improves upon the IEEE standard and provides temporary anonymous certified keys and automatic key change when entering a new region.

An alternative to PKIs are key management techniques based on identity-based cryptography (IBC), as proposed by several authors [19], [20], [21], [22], [23], [24]. In IBC, public keys are derived from IDs, while all key pairs are generated and stored by a central trusted authority. Using a secret only known to this authority, key pairs are generated using a cryptographic pairing scheme, such as Weil Pairing [25], resulting in node IDs. Using the pairing scheme and public parameters, nodes in the network are able to directly derive public keys from the ID. It provides certificateless cryptography and requires no retrieval of public keys as PKI schemes do.

There is a typical tradeoff between PKIs and IBC—pseudonym certificates achieve a limited form of anonymity, while IBC has the advantage of binding keys to identities without certificates. In Parking Communities, we operate on a more abstract level and can thus use either system, allowing us to make the most appropriate choice per use case. Each parking community member regularly collects its fellow members' public keys (as described in Section 3), independent from whether these derive from pseudonym certificates or IBC IDs. Furthermore, if needed for encryption or signature verification, public keys can be queried from a TTP in the PKI scenario or derived from IBC IDs.

3 Parking Communities

The motivation for Parking Communities is the interest to learn about free parking spots before reaching a destination area. We consider a typical working day with people parking their vehicle on their home street by night, at a primary work place by day, and visit different areas mostly in the evening [26]. A driver on his way home from work, for instance, sends a corresponding query via geocast into the destination area. Vehicles driving through or parking in this area can use their sensor systems to gather information about their surroundings [27], such as distance to the closest objects (e.g., other parked cars), and respond to the query originator. In this scenario, each vehicle requires an estimate of the trustworthiness of its communication partners in order to prioritize incoming queries or to determine a response's validity. To this end, drivers (to be more precise, their vehicles) regularly visiting the same area, such as neighbors or co-workers, dynamically create trusted Parking Communities to cooperate in exchanging parking spot information. By establishing trust anchors, signed and encrypted communication with previously encountered vehicles is facilitated. Thus, message interception and tampering is mitigated. Through a sophisticated mathematical rating model, vehicles dynamically establish an estimate of other vehicles' trustworthiness, without the need of a central TTP or RSU.

In this section, we present the conceptual design of the Parking Community protocol.

3.1 Creating a Community

A vehicle uses a new public/private key pair $\langle pk, sk\rangle$ (obtained via IBC or PKI) exclusively for each community c. Further, c includes a trust anchor τ, consisting of a set of areas $\mathcal{A}$ mapped to a set $ID_c \subset \mathcal{ID}$ of IDs encountered in these areas, i.e., vehicles that are part of the community c. Moreover, c comprises a mapping σ of each vehicle v's ID $id_v \in ID_c$ to two counting variables r_v and s_v. Formally, c is defined by the tuple

$$c = \langle\langle pk, sk\rangle, \tau, \sigma\rangle, \text{ with} \tag{1}$$

$$\tau : \mathcal{A} \rightarrow \mathcal{ID}, \tag{2}$$

$$\sigma : ID_c \rightarrow \{r, s\}. \tag{3}$$

In vehicular networks, there is no need to use human-readable IDs because networks are created ad hocly without human interaction, which allows us to generate them randomly. Because of this, we propose encoding pk_c directly as a vehicle's community ID, id_c. Thus, knowledge of id_c enables encrypted message exchange without prior key retrieval from TTPs.

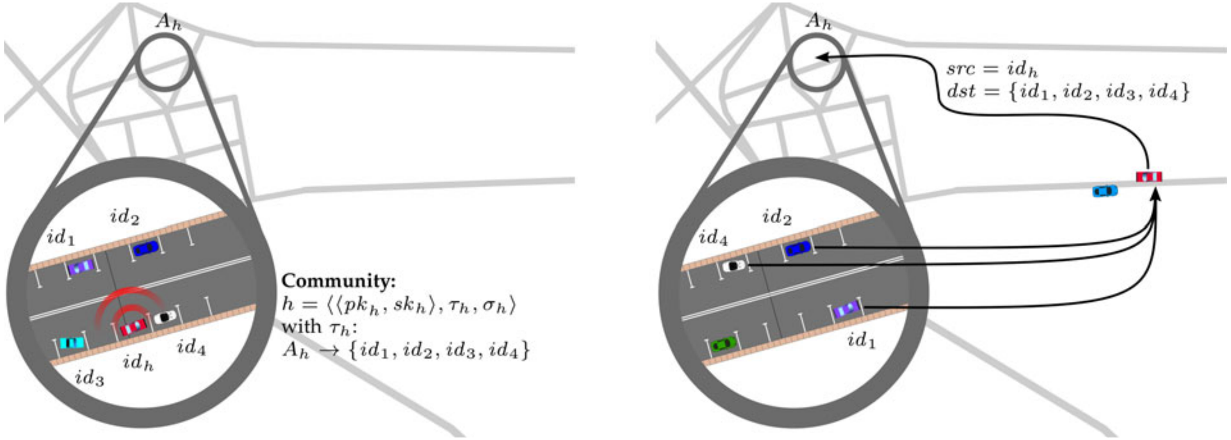
Referring to the running example, suppose a driver returning home at night and parking on his home street. After the engine is turned off, a new home community h with $id_h = pk_h$ is generated for the home parking area, if it does not exist yet. Else, the existing home community is selected based on location information. For communications with the community, id_h is actively used as source address src. For privacy reasons, a more sophisticated scheme is required in practice, which we describe in Section 4.4.

As depicted in Fig. 1a, IDs (i.e., public keys) of vehicles in the home area A_h are collected in the set ID_h via neighbor discovery while parking. To prevent Sybil attacks, position announcements of vehicles can be verified with a high probability as shown by previous work [28]. $A_h \rightarrow ID_h$ is added as a mapping to the trust anchor τ_h. Vehicle with id_h adding vehicle id_1 to its Parking Community does not require that the vehicle with id_1 adds id_h (cf. Fig. 1a). Thus, Parking Communities are not reciprocative and typical secure group management primitives such as join and leave are not required. Vehicles are only responsible for their own sets of communities. This reduces the communication overhead as no messages for group management are required. The mapping σ is initialized with $r = s = 0$ for each $id \in ID_h$.

As the engine is started again, e.g., when the driver leaves for work, the ID collection for this community is stopped. While at work, a corresponding Parking Community is created or updated with vehicle IDs via neighbor discovery. Of course, additional Parking Communities are created based on driver habits, e.g., for locations visited regularly such as shopping malls and friends' houses.

3.2 Querying

When driving back home, the set ID_h of previously collected IDs for A_h is looked up from τ_h. A query for available parking spots is cryptographically signed with h's private key sk_h. An ephemeral symmetric key is generated randomly and asymmetrically encrypted with the respective public key decoded from each $id \in ID_h$ as depicted in Fig. 1b. Conclusively, the query is sent via geocast into the home location A_h. The message contains (a) the


A_h
id_1
id_2
id_4
id_h
id_3
Community:
$h = \langle\langle pk_h, sk_h\rangle, \tau_h, \sigma_h\rangle$
with τ_h:
$A_h \to \{id_1, id_2, id_3, id_4\}$
A_h
$src = id_h$
$dst = \{id_1, id_2, id_3, id_4\}$
id_2
id_4
id_1

(a) Collecting IDs via neighbor discovery with physical verification and establishing a trust anchor

(b) Encrypted and signed query/response for a free spot via geocast

Fig. 1. Creating and querying a parking community.

symmetrically encrypted payload, and (b) the symmetric key encrypted for each vehicle in the corresponding community h, which comes with reasonable overhead compared to the overall message size which is dominated by the payload.

3.3 Responding

Each vehicle v with $id_v \in ID_h$ that is located in A_h (in Fig. 1b this includes the vehicles with IDs id_1, id_2, id_4, while id_3 has not arrived yet) can decrypt the query and verify its source because v also collected the ID of the querying vehicle in Step 3.1, when the community was created or updated. By means of this authentication, incoming queries can also be prioritized, as is further described in Section 3.5. Receiving vehicles encrypt their responses using the source ID src of the message, which corresponds to the public key. The response consists of an estimate e:

$$e = \begin{cases} 1 & \text{if a space is available} \\ -1 & \text{if no space is available.} \end{cases} \tag{4}$$

For the sake of simplicity, we do not further elaborate on how exactly vehicles come up with this estimate, but assume that each vehicle is able to use on-board sensor systems (e.g., ultra sonic, cameras) to determine which parking spots are available while driving through the home area A_h and while parking there, as was demonstrated in previous work [27]. Based on these data, as well as the time passed since the data was recorded, and other parameters, each vehicle estimates the likelihood of available parking spots in A_h that is finally mapped to a binary estimate e as shown in Equation (4). If no clear estimate is possible, we assume that the corresponding vehicle does not respond to the query at all in order to not provide potentially false data and to not risk deteriorating its rating (as described in Section 3.4).

3.4 Rating

The query originator finally receives the responses from an arbitrary number of community vehicles, depending on how many of them are located in the destination area and have chosen to respond with an estimate.

For each community vehicle v, the originator keeps a count of how many estimates e_v (see Section 3.3) turned out to be correct and incorrect, which we refer to as r_v and s_v, respectively. These values are used to calculate a reputation rating $Rep_v(r_v, s_v)$, based on the beta probability density function which can be used to represent probability distributions of binary events such as the estimation process $e_v \in \{-1; +1\}$ described in Section 3.3. The mathematical background of the beta function is analyzed in many text books on probability theory [29]. We therefore only present results based on the beta reputation system [30], which provides us with a mathematically sound and well-understood indication of how a particular vehicle is expected to behave in the future, that is in our case, to correctly or incorrectly announce a free parking spot. To this end, the probability expectation value $E(p)$ of the beta reputation function $\varphi(p|r, s)$ is a very suitable representation for this indicator, as argued by Josang and Ismail [30]. This gives us a reputation rating in the range $[0, 1]$ where the value 0.5 represents a neutral rating. Formally, the reputation rating $Rep_v(r_v, s_v)$ for vehicle v is thus defined as

$$\begin{aligned} Rep_v(r_v, s_v) &= E(\varphi(p|r_v, s_v)) \\ &= \frac{r_v + 1}{r_v + s_v + 2}, \end{aligned} \tag{5}$$

with φ being the beta reputation function [30]

$$\varphi(p|r, s) = \frac{\Gamma(r + s + 2)}{\Gamma(r + 1)\Gamma(s + 1)} p^r (1 - p)^s, \tag{6}$$

where $0 \leq p \leq 1, 0 \leq r, 0 \leq s$ and Γ being the gamma function.

After a timeout, the querying vehicle weighs all n received responses e_i with the corresponding vehicle i's reputation rating Rep_i to determine a consensus ω about the likelihood of a free parking spot in the destination area.

$$\omega = \frac{\Sigma_i^n (Rep_i(r_i, s_i) \cdot e_i)}{n} \tag{7}$$

If the outcome ω is below the threshold $\omega_{thresh} = 0$, the driver is advised to not rely on finding parking in his home area, but instead take the first free spot that he considers close enough, for example.

If the driver decides to drive to the home area (most likely if $\omega \geq \omega_{thresh}$), the vehicle scans the street for available spots itself and thus compares the actual situation with the received estimates, updating each r_v and s_v accordingly and providing feedback for the next calculation of the reputation rating.

3.5 Prioritization

Prioritization of incoming queries is done by responding vehicles solely based on their community information. Two different levels are possible: (a) member and (b) non-member prioritization.

a) Receiving vehicles can prioritize incoming queries based on the reputation rating of the originator, who signed the query. The reputation rating thereby directly correlates to a priority level—reputable vehicles are thus more likely to receive a response than those with a lower reputation. Consequently, it is in the vehicle's own interest to obtain a high reputation rating, such that it will also be provided with inquired information. This incentivizes frequent and honest responses and discourages dishonest and uncertain estimates in a tit-for-tat manner.

b) Vehicles receiving a query will typically favor community members over non-member requests and thus save resources, e.g., computing power. No reputation rating is available for non-members and thus the lowest priority level is assigned. Different advanced priority and resource management schemes can be considered to save energy or other resources, in particular while vehicles are parking. One option is a modification of the leaky bucket algorithm [31], for instance, with two buckets of, say, energy supply, one for members of a particular Parking Community and another for unknown requesters. Since this is not the focus of this paper, though, we do not elaborate on resource management.

3.6 Robustness

If vehicle density is sparse, there might not be sufficient vehicles in a destination area to get a response to a parking query. This is particularly true if the query is encrypted for the community and can thus only be responded to by community members, which excludes potential non-member communication partners. In a sparse network, this restriction could be relaxed such that queries are only signed by the originator, but not encrypted. Consequently, members as well as non-members are able to respond to the query, thus increasing the robustness of the protocol because a higher number of communication partners is available. Signing but not encrypting queries also allows vehicles to query for parking spots in irregularly or newly visited locations where they are not part of a community (and cannot predict which vehicles are currently located in that area). Since an originator does not have a reputation rating Rep_i for non-members, though, their responses are only taken into consideration in our protocol if the originator does not receive any responses from members, lest Sybil attacks become possible. Existing communities are not influenced and thus not put at risk by non-member responses.

From a receiving vehicle point of view, members and non-members will prioritize queries differently as explained in Section 3.5, but in either case the responses can be encrypted using the public key of the originator (which can be obtained as explained in Section 2.4), thus providing confidentiality of the parking availability data.

4 Attack Scenarios

In this section we first introduce the main security challenges for creating Parking Communities based on trust establishment and then analyze common attack scenarios.

Our scheme should work as an overlay on existing vehicular network protocols and without reliance on a central TTP. When a consensus for free parking spots is established, the scheme needs to account for impersonation and Sybil attacks to prevent impersonated answers and forged identities to reach a majority. Already generated key pairs used in the underlying network protocol can directly be utilized as unique identifiers. This prevents impersonation attacks, as it is not feasible to generate a private key, e.g., for signing messages, to a given public key, i.e., a given ID. In the case of ECC, public keys are short and can easily be encoded as identifiers (cf. Section 5.3). Sybil attacks, however, are harder to account for when establishing a consensus without a TTPs. We therefore propose a trust on first use (TOFU) model to verify the existence of an actual vehicle for each identity used for answering parking spot queries through physical encounters [28].

Our attack model is as follows: As little information as possible should be transmitted in the open, protecting the driver's anonymity against passive adversaries. Collecting physically encountered vehicle IDs makes it difficult to perform global Sybil attacks. Considering active attackers, capable of executing man-in-the-middle (MitM) and constrained targeted Sybil attacks, access to resources must be regulated. It should be prevented that information about vacant parking spaces is intercepted by a third party along the communication path. Conversely, vehicles (especially while parking) must be able to prioritize incoming queries in order to prevent Denial of Service (DoS) attacks, where malicious vehicles deplete resources by generating queries with multiple fake IDs (Sybil attack). Attacks and their mitigations are further discussed in the following sections.

4.1 Impersonation and Sybil Attacks

In all scenarios, our key management prevents impersonation attacks, where a vehicle impersonates another vehicle by adopting its ID during an ongoing communication. Because we require all messages to be signed, a message's signature always corresponds to the public key pk_s encoded in the message's src. An attacker would need to generate sk_s corresponding to an existing pk_s. This requires to randomly generate key pairs until a collision with the existing public key is found. In case of an ECC based protocol, the success probability is 2^{256} and the attack is thus considered infeasible. This is true if the difficulty of ECDLP holds and

ECDSA as well as its implementation has no critical flaws (e.g., insufficient entropy). When a Parking Community is created, context information such as the origin of a communication signal [28] allows a collecting vehicle to differentiate between physical vehicles. Thus, an attacker needs to be physically present when the victim is parking and is constrained in how many vehicles can be forged for a Sybil attack due to the difficulty of forging communication signals originating from different locations.

4.2 Interception of Parking Spot Availability

In Parking Communities, vehicles cooperate in order to gain an informational advantage. The information of available resources, namely ’parking spots’, is to be protected against passive adversaries as it could be used for reaching available spaces earlier than the original requester, without being part of the community. By encrypting query responses (confidentiality), intercepted information is of no value for eavesdropping adversaries.

4.3 Denial of Service

An attacker could try to exhaust available resource of a parking vehicle by querying many times for available parking spots. While the main purpose of the proposed Parking Communities is to provide a way to reach a consensus regarding specific parking locations, we introduced the idea of limiting computing resources for incoming queries. As described in Section 3.5 (b), vehicles can decide to only answer queries originating from reputable members of their own Parking Community. This works as a self-protecting feature in case of a Denial of Service attack.

4.4 Location Tracking

Existing privacy threats have been thoroughly investigated before [32], as have challenge-response protocols been proposed to prevent the exposure of context information. Global passive adversaries, on the one hand, can always track vehicles using RSUs, independent of whether IDs are changed regularly or not. Simply because of wireless emissions originating from vehicles, transmitted messages can be tracked from source to destination. It has been shown that such an attacker can correlate beacon messages to specific vehicles with a probability of nearly 100 percent [33]. On the other hand, local adversaries that physically follow a tracked vehicle cannot be protected against via any digital privacy mechanism either.

Yet, there is a wide spectrum in between these two extreme cases of attackers. Therefore, pseudonym certificates, e.g. [1], are deployed to cover the identity of vehicles. In addition to changing pseudonyms regularly, Sampigethaya et al. [34] have shown that a silent period between pseudonym changes is necessary. However, the concept of distributed communities requires vehicles to be uniquely identifiable by their peers.

We therefore propose using a key derivation function (KDF) allowing vehicles to change pseudonyms regularly but in a deterministic and reproducible way for members of the Parking Community (and only for them). During neighbor discovery (see Section 3.1), a common secret is shared besides the ID. This secret as well as the last valid pseudonym ID are input parameters to the KDF, which computes a new ID. This is done by both the vehicle changing its pseudonym and by all community members that have collected its ID and secret. Generally, each vehicle starts with a dedicated pseudonym per area, which is also only used for communication with the community. For other purposes, such as safety messages (e.g., CAM/DENM [1]), other pseudonyms according to the underlying security architecture are used and changed frequently [32]. The dedicated pseudonym per community area is typically only used once per day (e.g., when driving home), and can thusly be changed in intervals of one day using the KDF as described above. Consequently, Parking Communities also provide a means for anonymity and location privacy.

4.5 Accountability

Independent from using PKI or IBC as the underlying key management, we assume that a central trusted authority provides a means to unambiguously verify a vehicle’s public key.

5 Implementation

As described above, Parking Communities can be implemented on top of existing networking stacks, thus benefiting from standardization and security efforts already in place. To show the feasibility of our approach, we have implemented a prototype for the underlying security architecture by extending IBR-DTN,[1] a high-performance [35] Bundle Protocol [5] implementation in C++, to provide integration of ECDSA and ECIES, key management for ECC keys, encoding public keys as IDs, and our trust rating model. Since delay-tolerant networking (DTN) is an overlay network, we can transparently exchange the underlying networking stack, such as TCP/IP, IEEE 802.15.4, or IEEE 802.11p and its higher layer standard IEEE 1609. In DTN terminology, an ID is called endpoint identifier (EID), and messages are called bundles. This section describes the implementation details and cryptographic algorithms used for the Parking Community prototype.

5.1 Crypto Libraries

IBR-DTN uses OpenSSL,[2] which provides support for ECDSA, but no ECC encryption schemes, e.g., ECIES, out of the box. Furthermore, OpenSSL’s ECDSA implementation has been attacked via a side-channel [36]. Matured cryptographic libraries are Botan[3] and Crypto++.[4] Crypto++ has a long development history and is thus available on almost all Unix-like systems and Windows. While Botan only provides ECDSA, Crypto++ provides a wide range of functionality, among others the ECC-based algorithms ECDSA, ECNR, ECIES, ECDH, and ECMQV. For using recently proposed curves like Curve25519 [37], its authors provide a library called NaCl.[5] However, as described in Section 5.2.1, an integration of ECC into the Bundle Security Protocol

[1] http://www.ibr.cs.tu-bs.de/projects/ibr-dtn
[2] http://www.openssl.org
[3] http://botan.randombit.net
[4] http://www.cryptopp.com
[5] http://nacl.cr.yp.to

requires an asynchronous ECC encryption scheme and access to underlying cryptographic primitives. NaCl only provides synchronous DH key agreement and high-level access. Conclusively, we chose Crypto++ for our implementation.

The DTN daemon has been configured to reject bundles not cryptographically signed and has been extended to support and manage communities via an API.

5.2 Encryption and Signature Algorithm

This section introduces our extensions to the Bundle Security Protocol and discusses the security background of the used algorithms.

5.2.1 Extending the Bundle Security Protocol

The *Bundle Security Protocol Specification* (RFC 6257) [18] defines RSA-based cipher suites in conjunction with the AES block-cipher using galois/counter mode (GCM) for fast symmetric encryption of payload. Since modern ECC implementations are much faster than RSA implementations [7] and allow for shorter but equally secure key lengths,[6] we use ECC. We chose the widely used signature scheme ECDSA and the encryption scheme ECIES for payload integrity blocks (PIBs) and payload confidentiality blocks (PCBs), respectively. In traditional public key cryptosystems, the cryptographic principle of key separation is applied, i.e., generating different key pairs for signing and encrypting [38]. This was mainly motivated by the properties of the RSA trapdoor function. Degabriele et al. [39], however, have proven that ECDSA and ECIES can be securely combined using the same key pair. Breaking the key separation principle allows us to generate one key pair only. Thus, only one public key needs to be encoded as an EID, resulting in short EIDs.

5.2.2 Elliptic Curve Cryptography

We chose the curve ‘secp256k1’ [40], since it has a sufficiently long security history and is provided by nearly all cryptographic libraries available. It is also used in conjunction with ECDSA to sign Bitcoin transactions [41]. Bitcoin has undergone a comprehensive five-year analysis since its beginning and has shown no major weaknesses. In contrast to curves like NIST’s P-256, ‘secp256k1’ is not based on hashing unexplained seeds and is thus considered “somewhat rigid” [42].

In recent years, there have been advances in cryptanalysis of curves based on non-prime fields, e.g., $\mathbb{F}_{2^n}$, while the “overall security picture [has been] unchanged for prime-field ECC” [37], [43]. ‘secp256k1’ is a generalization of the Koblitz curve but associated to a prime field $\mathbb{F}_p$ with $p = 2^{256} - 2^{32} - 977$. It has two known primary weaknesses: Due to its structure, it has an efficiently computable endomorphism, which also leads to speed ups in Pollard’s rho algorithm [44]. The other weakness is its twist security [45]. Conversely, carefully implemented, problems due to twist security can be avoided. Besides those weaknesses, ’secp256k1’ is mathematically sound and it has shown no major drawbacks in the past [42].

5.3 Key Management

In DTNs, nodes are identified by an EID, which is formed by a uniform resource identifier (URI) [46], whereas the precise structure leaves room for adapting it for specific network structures. URIs offer a variable length and a standardized syntax, which can also be used to define groups of related nodes.

In Parking Communities, each vehicle v has a set of IDs, or EIDs, i.e., $EID_v \subset \mathcal{EID}$, with $\mathcal{EID}$ being the set of all valid endpoint identifiers. Each community’s $eid_c \in EID_v$ is derived from its public key pk according to the following form:

$$eid_c := \text{‘sec://’} \parallel base64url(pk). \tag{8}$$

Here, $base64url()$ corresponds to URL-safe Base64 encoding [47]. We introduced a new URI scheme ’sec’ to indicate that the following scheme-specific part (SSP) consists of the encoded public key instead of the typical node part and optional client/application specific parts. In our scheme, the SSP consists at minimum of the bytes consumed by the encoded public key. An ECC public key is $32\,\mathrm{b}$ long. Base64 uses four characters to represent $3\,\mathrm{b}$, always resulting in a multiple of 4; thus the length of n bytes encoded in Base64 is defined by

$$len_{ssp}(n) = \left\lceil \frac{n}{3} \right\rceil \cdot 4. \tag{9}$$

Conclusively, the SSP consumes $44\,\mathrm{b}$ without the application/client specific part. This is well below the maximum length of $1,023\,\mathrm{b}$ as defined by RFC 4648 [47].

6 Discussion

In this section, we provide a comparison of key and trust management schemes from the literature. Parking Communities can be implemented on top of different key management approaches, thus the following description can be used as a guideline for choosing the most appropriate architecture per use case. Moreover, existing trust management approaches are compared to Parking Communities. In particular, traditional certificate-based PKI, IBC, and incentive-based schemes are elaborated on. In the following sections we compare selected aspects of these architectures and summarize the results in Table 1.

PKI-based and IBC architectures have been introduced in Section 2.4. IBC schemes are subdivided into flat and hierarchical ones. Hierarchical Identity-Based Cryptography (HIBC) schemes are organized by tree-based hierarchy structures to distribute trust among intermediate authorities, e.g., affiliated to geographical regions for example [24], instead of having one central point of failure.

Incentive schemes, designed to protect against selfish behavior, are classified into barter-based, credit-based and reputation-based schemes [48]. As credit- and reputation-based schemes often engage with each other (e.g., [49]), they are treated as one category. However, a subdivision between schemes requiring a TTP acting as a virtual bank and self-organizing ones has been investigated. These schemes introduce credits, similar to virtual currencies, traded between nodes to pay for forwarding/routing of bundles. Reputation-based schemes are similar, while also

[6] http://www.keylength.com

TABLE 1
Comparison of Key and Trust Management Approaches

		Key Management			Credit/Reputation	
Property	Parking Com.	PKI[a]	IBC[b]	HIBC[c]	Bank[d]	SO[e]
No TTP Required	✓	✘	✘	✘	✘	✓(setup)
Revocation/Expiry	✓	✓	✓(expiry)	✓(expiry)	–	–
Anonymity	–[g]	✓/✘[f]	✘	✓(limited)	✘	✘
Confidentiality	✓/✘	✓	✓	✓	–	–
Integrity and Authenticity	✓	✓	✓	✓	✓	✓
Forward Secrecy	–[g]	✓	✓(limited)	✓(limited)	–	–
No Physical Encounters Required	✘	✓	✓	✓	✓	✓
Required Network Connectivity	sparse	high	medium	medium	medium	sparse
Protocol Complexity	medium	low	low	low	medium	high
No Single Point of Failure	✓	✘	✘	✓	✘	✓
Protects against Impersonation	✓	✓	✓	✓	–	–
Protects against Sybil Attacks	✓/✘	✓	✓	✓	–	–
Protects against Selfish Behavior	✓	✘	✘	✘	✓	✓

[a] *PKI schemes with traditional (X.509) or pseudonym certificates [1]*
[b] *IBC schemes: [20]*
[c] *HIBC schemes: [19], [21], [22], [24]*
[d] *Credit schemes, virtual bank: [49], [50], [51]*
[e] *Credit schemes, self organizing: [52]*
[f] *✓(limited): pseudonym certificates [1]; ✘: X.509 certificates*
[g] *Depending on underlying key management*
✓/✘ Only true for specific scenarios/proposed protocols
–Not part of this scheme's objectives.

providing protection against adversaries with high computational power.

6.1 Trusted Third Parties

Most schemes' authentication is based on one or more centralized TTPs. They are required for the initial authentication of new nodes and bootstrapping of trust. Traditional PKIs are organized hierarchically but without any restrictions with regard to which identities they are allowed to issue certificates. Thus, one compromised intermediate authority can compromise the whole network. Additionally, message exchange requires retrieval of public keys from TTPs before encryption/verification is possible. In IBCs, derivation of public keys from IDs allows encryption/verification without retrieving keys from TTPs in advance [20]. PKI certificates are issued using certificate signing requests (CSR), whereas key pairs were generated solely by the node itself; IBC schemes issue IDs by generating and storing key pairs. Thus, compromising an IBC infrastructure has much broader consequences to a network. Credit-based schemes require TTPs for reputation dissemination or a credit clearance process. Wei et al. distribute this task to a self-organizing network, leaving only the initial bootstrapping of nodes to an offline TTP [52].

6.2 Revocation

Revocation of certificates is typically achieved by distributing revocation lists, which can cause a significant overhead and poses a problem in sparse and intermittent networks. IBC schemes propose to encode an expiry date into the IDs themselves. While no direct revocation is possible, using short expiry dates, nodes are required to renew their ID regularly by contacting the IBC TTP over a secure channel. In parking communities, revocation of a public key is achieved by its owner digitally signing a revocation message and distributing it in the community, ensuring that nobody but the possessor of the private key can inject such a message.

6.3 Anonymity

Anonymity as a property is difficult to measure in real-world applications. To complicate data aggregation by attackers with limited capabilities, such as malicious vehicles recording metadata of forwarded bundles, pseudonyms are required. In vehicular protocols, such as proposed by the car 2 car communication consortium (C2C-CC), vehicles are issued a limited amount of pseudonym certificates by a central TTP. Vehicles iterate over this set until it has been depleted allowing a certain degree of pseudonymity [1]. As we have shown in Section 5, Parking Communities can be implemented on top of different networking stacks, including recent C2C-CC standards. Therefore, its underlying certificate infrastructure can be used to allow for a certain level of pseudonymity. As defined in our attack model in Section 4.4, Parking Communities require vehicles to recognize their peers for which we have provided a secure KDF-based solution. Consequently, the same level of anonymity (and location privacy) as in the underlying technology is achieved.

6.4 Trust Management

To establish trust, Parking Communities introduce trust anchors based on physical encounters to distinguish surrounding vehicles [28], preventing certain attacks as described in Section 4. In typical PKI or IBC schemes, central entities decide which nodes can be trusted, in case of PKI by providing lookup services. While IBC already provides an advantage over the traditional PKI system, as no

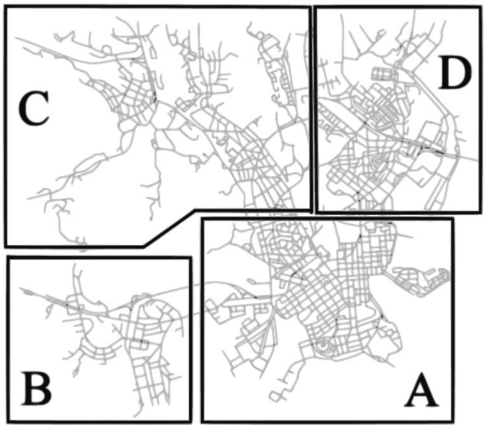


Fig. 2. Map of Helsinki with artificial districts [26].

public key lookup needs to be performed before transmissions, it still requires connectivity in regular intervals to extend the validity of IDs, though. A significant advantage of Parking Communities is that they require only sparse network connectivity because no lookup or renewal using central services is required.

6.5 Summary

Table 1 summarizes the aspects of the examined key and trust management schemes most relevant to vehicular networks. Some of these aspects have been discussed in the previous sections.

Similar to self-organizing credit-based schemes, our scheme does not require a security infrastructure to retrieve trust ratings. However, existing key management solutions, such as PKI or IBC, can be used to establish accountability. HIBCs improve over IBCs by hierarchical organization, but still leave a single root TTP. This is suitable for military scenarios, but has been proven ineffective against global, active adversaries.

While public key protocols with TTPs provide perfect protection against impersonation and Sybil attacks, our scheme additionally offers protection against impersonation attacks despite its distributed design. By means of the proposed trust anchor concept, it is also able to mitigate Sybil attacks, as discussed in Section 4.

We argue that the advantages of IBC in comparison to traditional PKI are minimal because both infrastructures need to somehow authenticate nodes on deployment. This is a major challenge, as a secure key-identity binding is crucial for any authenticated scenario. Establishing key-identity bindings with IBC leaves the key-escrow problem unsolved. Incentive schemes introduce high protocol complexity and more infrastructure [50] to allow distributed agreements in disruptive networks. Similar to Parking Communities, they allow prioritization based on incentives like virtual currencies or reputation and thus protect against selfish behavior.

Conclusively, this comparison illustrates the difficulty of balancing the trade-offs between centralized and decentralized key and trust management schemes. Parking Communities are a lightweight approach that integrates aspects from the wide range of existing architectures creating a novel approach for highly decentralized scenarios.

7 Simulation

We use The ONE [53] to simulate Parking Communities in a working day scenario [26] in the city of Helsinki, Finland. The model presents the everyday life of people going to work in the morning, spending their day at work, and commute back home at night. Our goal is to evaluate the development of reputation ratings over time and to show the general feasibility of our approach, i.e., if a car encounters sufficient other cars in order to create a sufficiently large community to get replies to its queries.

7.1 Setup

In the Working Day Movement model, over $1,000$ nodes move on a map of the Helsinki area with the size of roughly $7{,}000$ x $8{,}500\,\mathrm{m}^2$. The nodes and their home zones are assigned to four main and three overlapping artificial districts, as depicted in Fig. 2 and further described by Ekman et al. [26]. Each node has its own home zone, which typically overlaps with other zones depending on the node density per district. Twenty-five percent of all nodes are either malicious nodes or benign nodes with potentially false sensor information, i.e., they may report false positives. For the sake of readability, we subsume both groups under the term malicious nodes because it is irrelevant why false information is reported.

In contrast to the original movement model, we assume that all nodes are regular vehicles, instead of also including busses and taxis. We used a warmup period of a full day (as opposed to half a day), due to the periodic nature of the proposed protocol as well as of the mobility model. We set the transmit range of all nodes to $100\,\mathrm{m}$ and the home zone radii to $300\,\mathrm{m}$. Hence, vehicles always park within a radius of $300\,\mathrm{m}$ to their home zone center, with a random offset, and create a community by collecting vehicle IDs in their communication range. Every morning, each vehicle leaves for work at a specified time, and stays there for $8\,\mathrm{h}$, before it either commutes back home or follows an evening activity first. Halfway home, though, each vehicle geocasts a query into the home zone according to Section 3. It then waits for responses from its community members. In our simulations, the probability of a free spot in the home zone (the *ground truth*) is 0.5. Honest nodes receiving the query always respond with the ground truth, while malicious nodes lie with a probability of $\psi = 0.5$, i.e., respond with the opposite of the ground truth. The querying vehicles then receive the responses and calculate a weighted consensus ω. In the home zone, they compare the responses with the ground truth and update the reputation ratings accordingly.

The simulation runs for $700{,}000\,\mathrm{s}$, which corresponds to eight full days. We repeat the simulation 10 times.

7.2 Results

Fig. 3 shows the number of members per Parking Community per simulation day, averaged over all 10 simulations runs. It is observable that after five days 50 percent of all communities have at least two to four members, with another 25 percent having between four and 20 members. These values further increase during the following days, as vehicles park at random locations in their home zones, thus meeting new vehicles. For the simulated scenario, the

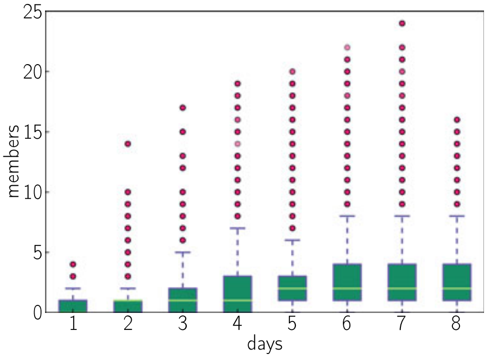


Fig. 3. Parking Community sizes.

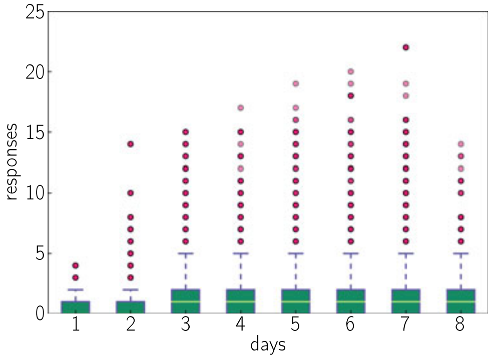


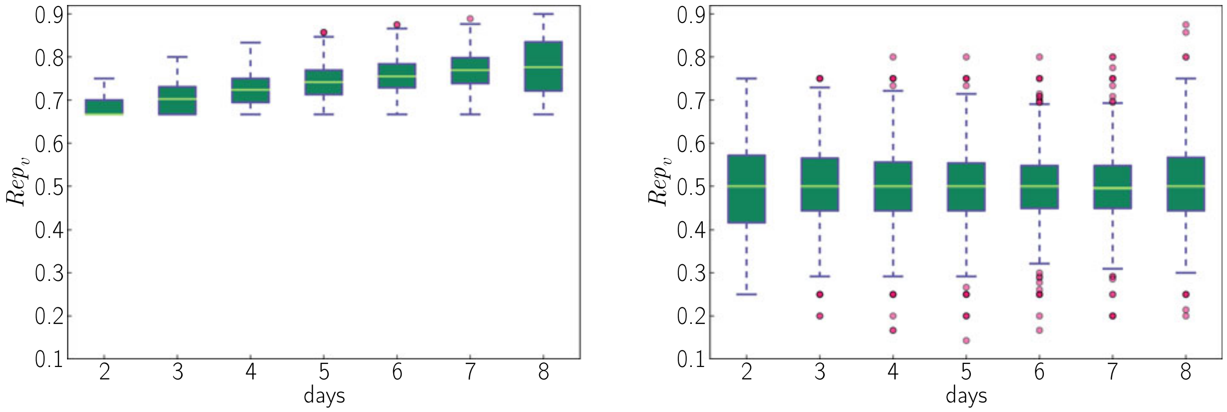
Fig. 4. Number of responses received per day.

community sizes basically stabilize around days 6 and 7. In sum, at least 75 percent of all vehicles have between 3 and more than 20 vehicle IDs collected after a few days. Due to the specific geography of Helsinki, with some remote and isolated areas (e.g., on islands only connected by a bridge to the mainland), some vehicles can only create very small communities, while vehicles in densely populated areas, such as District A in Fig. 2, have quite large communities after a short period of time.

Fig. 4 now correlates the community sizes with the number of successful query/response exchanges. It can be observed that from day 3 on, vehicles receive two responses on average. Remarkably, 25 percent of vehicles received significantly more responses, up to 15. The maximum number of responses further increases to up to 23 which is almost the maximum Parking Community size. In this particular case, this indicates that the querying vehicle was (a) part of a large community, and (b) was returning home as one of the latest out of his peers, such that almost every other node was already located in the home zone and thus able to respond to the query. As described above, vehicles in densely populated areas (and thus with a large community size) have a significant advantage over remote areas. In downtown areas these vehicles receive sufficiently many responses to make a meaningful contribution to the parking search.

We further evaluate how reputation ratings develop over time, in particular by comparing honest and malicious nodes in Fig. 5. As we have a decentralized model, in which no single entity is in charge of keeping track of a vehicle's reputation rating, but each community member establishes its own rating per peer, we average the reputation rating for each vehicle over all other nodes that have it in their respective communities.

All nodes start with a reputation value of 0.5, which represents a neutral rating. As the reputation $Rep(r, s)$ depends on the physical verification of received responses, Fig. 5 omits the simulated day 1, since only after the vehicles parks in the home zone, the respective values r, s can be updated, while the reputation is already updated halfway home when a consensus ω is calculated. As can be seen in Fig. 5a, honest nodes' reputation continually increases over the simulation time, but has already reached an average of 0.7 on day 2. A peculiar observation is that on day 8, the box (i.e., the interquartile range) is larger than on the previous days, indicating a larger variance. This is because some vehicles have not yet reached their home area before the simulation ends, which does not affect the general validity of the observations. In comparison, Fig. 5b shows the reputation ratings for malicious nodes. At first sight, it may seem curious that malicious nodes' reputation remains at 0.5 on average, with some outliers being at par with honest nodes' reputation. However, this is clearly expected as we have modeled the behavior of malicious nodes to arbitrarily lie or tell the truth. Hence, vehicles cannot identify and downrate malicious nodes, but have to remain neutral, which is reflected in the simulation results. Yet, as we have shown above, honest vehicles are uprated quite quickly in comparison, such that a weighted consensus ω is nevertheless a meaningful



(a) Reputation ratings for honest nodes per day

(b) Reputation ratings for malicious nodes per day

Fig. 5. Development of reputation ratings averaged over nodes and 10 simulations runs.

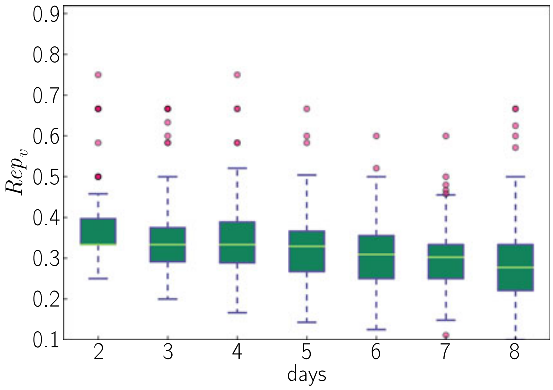


Fig. 6. Reputation for malicious nodes, $\psi = 0.85$.

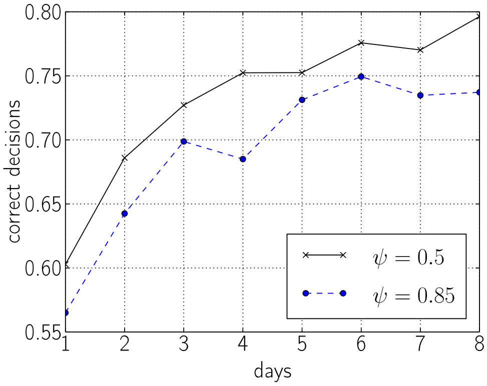


Fig. 7. Rate of correct decisions over time.

criterion. To provide further evidence, though, Fig. 6 shows the reputation ratings for malicious nodes with $\psi = 0.85$, instead of $\psi = 0.5$ (while keeping constant all other parameters). It can be clearly observed that malicious vehicles can clearly be identified and are downrated significantly (and continually) from day 2 on. On day 7, for instance, the average rating is 0.3, with 75 percent of all (malicious) nodes having a lower rating than 0.35.

Finally, we evaluate how often vehicles make the right decision about relying on available parking spots in their home area, as described in Section 3.4. A decision is correct, if (a) a spot is free and $\omega \geq \omega_{thresh} = 0$ or (b) no spot is available and $\omega < \omega_{thresh} = 0$.

Fig. 7 shows the relative frequency of correct decisions per simulated day for different probabilities ψ of lying. As expected, the rate of correct decisions increases over time because the reliability of reputation ratings increases as well. For $\psi = 0.5$, the correct decision rate is already higher than 0.75 after day 4 and keeps rising. It takes longer to reach the same values for $\psi = 0.85$ as the system has to cope with liars that are more chronic. In sum, though, good values are achieved after only a few days (remember that, in our simulations, the system is used once per day when driving home), showing the feasibility of the approach.

8 Conclusion

In this paper, Parking Communities have been presented. They provide a novel trust management for vehicular parking applications without reliance on a central TTP for retrieving trust ratings. For this purpose, vehicles create communities, trusted groups helping their members to find parking in their respective community area. Trust anchors enable signed and encrypted request-response communication in disrupted environments. As our approach can be used as an overlay to existing vehicular networking technologies, it can directly benefit from established security mechanisms, e.g., pseudonym certificates. Our approach is based on high-performance state-of-the-art encryption and signature algorithms, in particular ECC, as well as a well-understood mathematical trust rating model. Attack scenarios and their mitigations are discussed. Without requiring a TTP, our scheme provides protection against impersonation and Sybil attacks utilizing trust anchors and physical verification. The underlying security architecture of Parking Communities has been implemented in the open-source IBR-DTN, which is publicly available. We provide a comprehensive comparison with existing key and trust management schemes for vehicular networks, as well as simulations showing the concept's feasibility.

8.1 Future Work

We plan to design fine-grained access control mechanisms to improve resource management and prioritization of incoming queries, e.g., based on energy/response budgets or additional properties verifiable by trusted third parties, such as certificates of disability. In order to further increase the frequency of correct decisions, vehicles with high mutual trust could exchange and merge their sets of communities. The expected results are an increase in the size and number of communities as well as more robust reputation ratings.

Acknowledgments

This project has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement Number [269916, V-Charge]. J. Timpner is the corresponding author.

References

[1] Car 2 Car Commun. Consortium, "Manifesto: Overview of the C2C-CC system, V1. 1," Tech. Rep., Aug. 2007.

[2] Federal Highway Administration, *Advanced Parking Management Systems: A Cross-cutting Study: Taking the Stress Out of Parking*. Intelligent Transportation Systems, U.S. Department of Transportation, 2007.

[3] *Wireless Access in Vehicular Environments (WAVE)—Security Services for Applications and Management Messages*, IEEE Std 1609.2-2006, Jul. 2006.

[4] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "IBR-DTN: A lightweight, modular and highly portable bundle protocol implementation," *Electron. Commun. Eur. Assoc. Softw. Sci. Technol.*, vol. 37, pp. 1–11, Jan. 2011.

[5] K. Scott and S. Burleigh, "Bundle protocol specification," RFC 5050, IETF, Nov. 2007.

[6] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[7] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. New York, NY, USA: Springer, 2010.

[8] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. (2012). High-speed high-security signatures. *J. Cryptographic Eng.* [Online]. *2(2)*, pp. 77–89. Available: http://cr.yp.to/papers.html#ed25519

[9] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[10] J. Zhang, "A survey on trust management for VANETs," in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2011, pp. 105–112.

[11] P. Wex, J. Breuer, A. Held, T. Leinmuller, and L. Delgrossi, "Trust issues for vehicular ad hoc networks," in *Proc. Veh. Technol. Conf. Spring*, May 2008, pp. 2800–2804.

[12] M. Gerlach, "Trust for vehicular applications," in *Proc. 8th Int. Symp. Auton. Decentralized Syst.*, Mar. 2007, pp. 295–304.

[13] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1238–1246.

[14] A. Vinel, C. Campolo, J. Petit, and Y. Koucheryavy, "Trustworthy broadcasting in ieee 802.11p/wave vehicular networks: Delay analysis," *IEEE Commun. Lett.*, vol. 15, no. 9, pp. 1010–1012, Sep. 2011.

[15] A. Patwardhan, A. Joshi, T. Finin, and Y. Yesha, "A data intensive reputation management scheme for vehicular ad hoc networks," in *Proc. 3rd Annu. Int. Conf. Mobile Ubiq. Syst. Workshops*, Jul. 2006, pp. 1–8.

[16] S. Park, B. Aslam, and C. C. Zou, "Long-term reputation system for vehicular networking based on vehicle's daily commute routine," in *Proc. Consumer Commun. Netw. Conf.*, 2011, pp. 436–441.

[17] A. Studer, E. Shi, F. Bai, and A. Perrig, "TACKing together efficient authentication, revocation, and privacy in VANETs," in *Proc. IEEE Sensor, Mesh, Ad Hoc Commun. Netw.*, Jun. 2009, pp. 1–9.

[18] S. Symington, S. Farrell, H. Weiss, and P. Lovell, "Bundle security protocol specification," RFC 6257, IETF, May 2011.

[19] A. Seth and S. Keshav, "Practical security for disconnected nodes," in *Proc. 1st IEEE ICNP Workshop Secure Netw. Protocols*, 2005, pp. 31–36.

[20] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo, "Applicability of identity-based cryptography for disruption-tolerant networking," in *Proc. 1st Int. Workshop Mobile Opportunistic Netw.*, 2007, pp. 52–56.

[21] A. Kate, G. Zaverucha, and U. Hengartner, "Anonymity and security in delay tolerant networks," in *Proc. 3rd Int. Conf. Security Privacy Commun. Netw.*, 2007, pp. 504–513.

[22] R. Patra, S. Surana, and S. Nedevschi, "Hierarchical identity based cryptography for end-to-end security in DTNs," in *Proc. 4th Int. Conf. Intell. Comput. Commun. Process.*, Aug. 2008, pp. 223–230.

[23] W. L. Van Besien, "Dynamic, non-interactive key management for the bundle protocol," in *Proc. 5th ACM Workshop Challenged Netw.*, 2010, pp. 75–78.

[24] M.-R. Fida, M. Ali, A. Adnan, and A. Arsalaan, "Region-based security architecture for DTN," in *Proc. 8th Int. Conf. Inf. Technol.: New Gen.*, 2011, pp. 387–392.

[25] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Adv. Cryptology*, 2001, pp. 213–229.

[26] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working day movement model," in *Proc. 1st ACM SIGMOBILE Workshop MobilityModels*, 2008, pp. 33–40.

[27] P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Mühlfellner, S. Wonneberger, J. Timpner, S. Rottmann, B. Li, B. Schmidt, T. N. Nguyen, E. Cardarelli, S. Cattani, S. Brüning, S. Horstmann, M. Stellmacher, H. Mielenz, K. Köser, M. Beermann, C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, R. Iser, R. Triebel, I. Posner, P. Newman, L. Wolf, M. Pollefeys, S. Brosig, J. Effertz, C. Pradalier, and R. Siegwart, "Toward automated driving in cities using close-to-market sensors: An overview of the V-charge project," in *Proc. Intell. Veh. Symp.*, Jun. 2013, pp. 809–816.

[28] M. Fiore, C. Ettore Casetti, C. Chiasserini, and P. Papadimitratos, "Discovery and verification of neighbor positions in mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 289–303, Feb. 2013.

[29] G. Casella and R. L. Berger, *Statistical Inference*, series Duxbury advanced series. Pacific Grove, CA, USA: Duxbury Press, 1990.

[30] A. Jøsang and R. Ismail, "The beta reputation system," in *Proc. 15th Bled Electron. Commerce Conf.*, 2002, p. 41.

[31] J. Turner, "New directions in communications(or which way to the information age?)," *IEEE Commun. Mag.*, vol. CM-24, no. 10, pp. 8–15, Oct. 1986.

[32] J. Freudiger, M. Jadliwala, J.-P. Hubaux, V. Niemi, and P. Ginzboorg, "Privacy of community pseudonyms in wireless peer-to-peer networks," *Mobile Netw. Appl.*, vol. 18, no. 3, pp. 413–428, 2013.

[33] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in inter-vehicular networks: Why simple pseudonym change is not enough," in *Proc. Wireless On-Demand Netw. Syst. Serv.*, 2010, pp. 176–183.

[34] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "Caravan: Providing location privacy for vanet," DTIC, Tech. Rep., 2005.

[35] W.-B. Pöttner, J. Morgenroth, S. Schildt, and L. C. Wolf, "An empirical performance comparison of DTN bundle protocol implementations," in *Proc. Workshop Challenged Netw.*, Las Vegas, NV, USA, 2011, pp. 61–64.

[36] D. Brumley and D. Boneh, "Remote timing attacks are practical," *Comput. Netw.*, vol. 48, no. 5, pp. 701–716, 2005.

[37] D. J. Bernstein, "Curve25519: New Diffie-Hellman speed records," in *Proc. 9th Int. Conf. Public Key Cryptography*, 2006, pp. 207–228.

[38] S. Haber and B. Pinkas, "Securely combining public-key cryptosystems," in *Proc. Comput. Commun. Security*, 2001, pp. 215–224.

[39] J. P. Degabriele, A. Lehmann, K. G. Paterson, N. P. Smart, and M. Strefler, "On the joint security of encryption and signature in EMV," Cryptology ePrint Archive, Report 2011/615, 2011.

[40] S. SEC. (2000). 2: Recommended elliptic curve domain parameters [Online]. Available: http://www.secg.org

[41] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Consulted*, vol. 1, p. 2012, 2008.

[42] D. J. Bernstein and T. Lange. SafeCurves: Choosing safe curves for elliptic-curve cryptography [Online]. Available: http://safecurves.cr.yp.to

[43] D. J. Bernstein and T. Lange. (2013, May). Security dangers of the NIST curves. [Pres] [Online]. Available: http://www.hyperelliptic.org/tanja/vortraege/20130531.pdf

[44] J. M. Pollard, "A Monte Carlo method for factorization," *BIT Numerical Math.*, vol. 15, no. 3, pp. 331–334, 1975.

[45] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," Cryptology ePrint Archive, Report 2013/734, 2013.

[46] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture," RFC 4838, IETF, Apr. 2007.

[47] S. Josefsson, "The base16, base32, and base64 data encodings," RFC 4648, IETF, Oct. 2006.

[48] J. Miao, O. Hasan, S. Mokhtar, L. Brunie, and K. Yim, "An analysis of strategies for preventing selfish behavior in mobile delay tolerant networks," in *Proc. Innovative Mobile Internet Services Ubiquitous Comput.*, 2012, pp. 208–215.

[49] R. Lu, X. Lin, H. Zhu, X. Shen, and B. Preiss, "Pi: A practical incentive protocol for delay tolerant networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 1483–1493, Apr. 2010.

[50] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "SMART: A secure multilayer credit-based incentive scheme for delay-tolerant networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 4628–4639, Oct. 2009.

[51] C. Gong, W. Bo, and Z. Faru, "SIS: Secure incentive scheme for delay tolerant networks," in *Proc. Symp. Distrib. Comput. Appl. Bus., Eng. Sci.*, 2012, pp. 310–313.

[52] L. Wei, H. Zhu, Z. Cao, and X. Shen, "MobiID: A user-centric and social-aware reputation based incentive scheme for delay/disruption tolerant networks," in *Ad-hoc, Mobile, and Wireless Networks*, series Lecture Notes in Computer Science, H. Frey, X. Li, and S. Ruehrup, Eds. New York, NY, USA: Springer, 2011, vol. 6811, pp. 177–190.

[53] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. 2nd Int. Conf. Simul. Tools Techn.*, 2009, p. 55.

Julian Timpner received the BS and MS degrees in computer science in 2009 and 2012, respectively, from Technische Universität Braunschweig. He is currently working toward the PhD degree at the Institute of Operating Systems and Computer Networks, Technische Universität Braunschweig, where he has been a research fellow since 2012. In 2010, he was a visiting student at the University of California, San Diego. His research interests include vehicular networks and e-mobility. He is a student member of the IEEE.

Dominik Schürmann received the BS and MS degrees in 2010 and 2014, respectively, from Technische Universität Braunschweig. He is currently working toward the PhD degree at the Institute of Operating Systems and Computer Networks, Technische Universität Braunschweig, where he has been a research fellow since 2014. His research interests include unobtrusive security in distributed systems and cryptographic algorithms in general. He is a student member of the IEEE.

Lars Wolf received the diploma degree in 1991 and the doctoral degree in 1995, both in computer science. From 1991 to 1996, he worked at IBM's European Networking Center, until he joined the Technische Universität Darmstadt as an assistant professor. He joined Universität Karlsruhe (TH) in 1999, where he was an associated professor in the Computer Science Department and alternate director of the computer center. Since spring 2002, he is a full professor for computer science at the Technische Universität Braunschweig, where he is head of the Institute of Operating Systems and Computer Networks. His current research interests include wireless and mobile networking in general, sensor networks, vehicular networks, delay-tolerant networks, and network & system support for mobile systems. He is a member of the IEEE.