Get in Line: Ongoing Co-Presence Verification of a Vehicle Formation Based on Driving Trajectories

Published in:
3rd IEEE European Symposium on Security and Privacy (Euro S&amp;P'18)

Accepted/In press: 26/04/2018

Document Version
Early version, also known as pre-print

Please cite the original version:
Get in Line: Ongoing Co-Presence Verification of a Vehicle Formation Based on Driving Trajectories

Christian Vaas\(^*\), Mika Juuti\(^\dagger\), N. Asokan\(^\ddagger\) and Ivan Martinovic\(^*\)

*Department of Computer Science
University of Oxford, Oxford, United Kingdom
\{christian.vaas, ivan.martinovic\}@cs.ox.ac.uk

\dagger Department of Computer Science
Aalto University, Helsinki, Finland
mika.juuti@aalto.fi

\ddagger Department of Computer Science
Aalto University and University of Helsinki, Helsinki, Finland
asokan@acm.org

Abstract—Intelligent transportation systems and the advent of smart cities have created a renewed research interest in vehicular networks (VANET). These ad-hoc networks are the key technology for new collaborative approaches to increase the efficiency and safety of our roads. In effect, city-scale field trials are being conducted by major high-tech companies to explore the capabilities and limitations of vehicle-to-infrastructure and vehicle-to-vehicle communication. Initial advances have led to safety enhancing applications like electronic emergency brake light, cooperative collision avoidance and cooperative adaptive cruise control. In IEEE standard 1609.2, security measures to guarantee the integrity and authenticity of VANET messages are specified. However, physical properties like spatial proximity and driving direction are not considered. These become notably important when vehicles make decisions that concern the safety of users for example to avoid a collision.

We propose a novel approach to verify the ongoing co-presence of two vehicles. Our method is based on the observation that the trajectory through a road network can be used to uniquely define a vehicle’s location as well as its driving direction. Our system provides a protocol to authenticate VANET messages for a group of vehicles driving in succession and to de-authenticate vehicles that have left the formation.

To demonstrate the feasibility of trajectories as proof for co-presence, we implemented a smartphone application and conducted driving experiments under real-world conditions. We analyze the road network of several major cities from different continents to show the generalizability of our approach. Additionally, we systematically evaluate the security properties of our system by performing city-scale simulations under realistic conditions.

1. Introduction

Although government support for public and mass transportation has increased, automobiles still remain the dominant mode of transportation. Despite drawbacks in safety [1], efficiency [2], and sustainability [3], their flexibility and availability is unmatched by any other means of transportation [4]. Combined with the emergence of smart cities, the overwhelming use of automobile transport has created a renewed interest in vehicular network (VANET) research. These wireless networks use dedicated short range communication (DSRC), based on the transmission standard IEEE 802.11p [5], to allow the communication between vehicles’ on board units (OBU) as well as road side units (RSU) stationed along the roadway.

As automobiles present drawbacks compared to public and mass transportation, efforts to address those have spawned new approaches such as vehicle platoons and cooperative adaptive cruise control (CACC). In 2016, the European Union organized the European platooning challenge with participants like Daimler, MAN and Volvo to promote these efforts. A platoon is a group of vehicles that consists of a platoon leader and several member vehicles driving in formation. This facilitates a decrease in fuel consumption, reduces cost and environmental pollution [6], [7]. In the future, an autonomous platoon leader may coordinate an entire group of vehicles, meaning platoon members may allow the leader to steer for them [8]. Moreover, platoons also promise a more efficient utilization of the road infrastructure and a reduction of congestion [9]. Field trials for highway based platoons have returned positive results emphasizing their efficacy [10]. Encouraged by this and the continued increase in urbanization, current mobility research tries to transfer these advantages into the urban scenario [11].

CACC and other VANET based safety applications such
as electronic emergency brake light (EEBL), local danger warning (LDW), and cooperative collision avoidance (CCA) all rely on continuous message exchange between OBUs and RSUs. As shown in recent research, these location based applications are subject to attacks that can reduce their efficiency and even cause fatalities [12], [13], [14], [15]. For example by inserting non-existing vehicles through false location claims, CCA systems can be fooled to induce rear-end collisions.

Under the IEEE 1609.2 standard [16], security mechanisms to guarantee the integrity and authenticity of VANET messages have been proposed. The standard suggests that a public key infrastructure (PKI) is needed which includes certificate authorities (CA) that issue certificates at vehicle registration and bind them to an individual’s identity. Although this provides cryptographic guarantees like integrity and authenticity, the security standard lacks the verification of physical properties. Location information is particularly important for the aforementioned safety and efficiency applications to determine co-presence [17]. Thus, there is a need to verify this property between the sender and receiver to ensure the reliability of VANET messages. In effect, this would allow the receiver to only accept messages originating from vehicles that are actually driving along the same route at the same time for a sustained period of time.

Several systems have been proposed to address co-presence verification. One approach compares the road surface as a context parameter to verify physical proximity of vehicles in a formation [18]. However, it is difficult to assess the entropy of this property and provide an adequate measure of security. It has also been suggested to use distance bounding for proximity verification [19] which limits a sender’s position to a perimeter, however, this method does not identify whether the sender is in front or behind the receiver. Moreover, a wireless implementation of this protocol is sensitive to interference between the two parties which poses a major challenge in an urban environment with a multitude of vehicles [20]. Other approaches that use specialized antennas allow to place a sender in front or behind the receiver [21], [22] but these methods do not take into account the road layout; thus, the receiver cannot be certain whether the sender is on the same or a parallel road. A combination of the latter two methods, was proven efficient in highway simulations, but has only limited applicability in urban environments [23].

With the proliferation of VANETs, protecting the privacy of an individual’s location becomes increasingly important [24]. Even though a driver’s identity is not directly exposed in vehicular communication, it was demonstrated that linking multiple messages to the same sender can allow an attacker to infer an individual’s home and work location, as well as revealing the driver’s identity [14], [15], [25]. Applications like CACC require exactly this linkability to track the movements of co-present vehicles. Therefore, we need a technique that allows vehicles to reliably authenticate messages while unlinkability mechanisms like pseudonym schemes remain intact. These methods have been proposed to ensure the user’s identity is kept private.

In this paper, we propose a system to enable the safe application of VANET messages in an urban environment. We present a method for ongoing co-presence verification in a privacy preserving way. It enables vehicles to discriminate between messages from vehicles traveling in the same direction alongside them and messages from vehicles that are stationary or moving in a different direction. Additionally, the protocol presented enables the authentication of VANET messages between vehicles while the user’s identity remains protected. We implemented a smartphone application to evaluate the performance and feasibility of our system using real-world experiments. The security of the protocol is conceptually analyzed and tested through realistic traffic simulations.

Contributions.

- We present a novel approach to verify the ongoing co-presence of vehicles in an urban environment. We exploit the characteristics of a trajectory through a road network and require a vehicle to share the same route as a leading vehicle in order to become a verified following vehicle. Co-present vehicles gain knowledge of verified neighbors, and thus the capability to authenticate their VANET messages. This allows authenticity checks for safety critical applications.
- We design a protocol that verifies the ongoing co-presence of two vehicles in a privacy preserving way in line with recent advances to protect drivers’ identity and location privacy. Our approach operates transparent to pseudonym schemes and hence cannot be used by an attacker to make message linkable to the same sender.
- We implement our system as an Android application to evaluate its performance in experiments involving two cars. Furthermore, we conceptually analyze and test its security properties through realistic traffic simulations. We use false accept and false reject rates to summarize our findings.

This paper is structured as follows. First, an application scenario motivates the approach followed by the assumptions on system and threat model in Section 2. Additionally, we identify the requirements to support pseudonym schemes as a mechanism to preserve a driver’s privacy. Next, the architecture is described in Section 3 and the results of our evaluation are discussed in Section 4. Thereafter, related work on VANET security and co-presence verification is reviewed in Section 5. To conclude, Section 6 summarizes our findings and states future work.

2. Assumptions and Goals

This section introduces the assumptions of our approach and its general concept. We start with the application scenario to illustrate the different verification stages of a vehicle. We follow with the system and threat model and identify important considerations to ensure the driver’s privacy in VANETs. Lastly, we identify and discuss the requirements for our system.
2.1. Application Scenario

We propose a method to verify the ongoing co-presence of vehicles. Similar to platoon systems, we exploit vehicles driving in succession and demonstrate that their trajectory through a road network can be used to verify their ongoing co-presence. Based on this method, a leader vehicle (LV) can promote unverified candidate vehicles (CV) to verified following vehicles (FV).

The application scenario for our system is illustrated in Figure 1. The vehicle formation depicted consists of the leader vehicle $LV$ and several following vehicles $FV$, with their route indicated as a dashed line. A candidate vehicle $CV$ which took a different route, highlighted with a dotted line, eventually merges with the formation’s route. Until this point, $CV$’s ongoing co-presence is not verified, hence it does not belong to the set of verified following vehicles.

The system is triggered by the $CV$ sensing the formation’s proximity through beacon messages that are periodically sent by the $LV$. These messages include the leader’s location to indicate its presence. When $CV$ meets $LV$, the two parties compare their upcoming trajectory without disclosing them to protect their location privacy. From this, $CV$ and $LV$ learn that they are going to share a section of their journeys. Thus, at time $t_0$, the candidate gets in line with the last verified vehicle and the ongoing co-presence verification process is started.

At time $t_1$, the $CV$ has driven in line with the formation and continuously compared its past trajectory. This is used as proof for ongoing co-presence and since the $CV$ traversed the same route as the other vehicles, it gets promoted to a verified $FV$. Both vehicles continue exchanging beacon messages to mutually verify their ongoing co-presence.

As a member of the formation, a $FV$ is granted access to their secure communication channel. It is used to confirm electronic signature information, i.e. their ECC public keys, allowing their OBUs to authenticate incoming VANET messages. This enables them to distinguish between messages from members of the formation and messages from stationary vehicles or vehicles driving in a different direction. Authenticating these messages is transparent to mechanisms like CACC and other safety related applications, providing an additional layer of security. Further, we envision that the communication among $FVs$ can be used to implement a more interwoven CACC which not only transfers acceleration and braking behavior but may in the future even delegate steering to an autonomous $LV$.

For convenience on longer journeys, members could also establish a communication channel among their passengers. Eventually, $FVs$ may reach their destination or divert from the formation’s route. Such a diversion has to trigger the de-authentication of the $FV$ by the $LV$ since they are no longer co-present. Figure 1 shows a deviation between a $FV$ and the vehicle formation at time $t_2$ which is automatically detected by the $LV$ who terminates $FV$’s session, demoting it to a $CV$.

2.2. Driver Privacy Considerations

VANET communication is commonly categorized into vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) messages. With increasing receiver diversity these communications are grouped under the term V2X. The underlying communication standard IEEE 802.11p and security standard IEEE 1609.2 allow message exchange for applications with real-time requirements, such as safety and efficiency mechanisms.
Given the human involvement in V2X systems, driving safety is of paramount importance and requires the communication protocol to implement sufficient security measures. It has been shown that adversaries can cause accidents, leading to physical harm or even death of passengers if no security is in place [12, 13, 23]. For example, an attacker can generate misleading messages to disable CCA [26] or other safety applications like the EEBL and LDW. Therefore, a mechanism to ensure integrity, authenticity, and non-repudiation of V2X messages is necessary. The proposal of using asymmetric cryptography [27] has received the highest attention in the standardization committees. It is proposed that the required certificates are issued by a CA upon vehicle registration. Based on Elliptic Curve Digital Signature Algorithm (ECDSA) signatures, messages are authenticated and their respective public keys validated using a PKI maintained through RSUs.

Since V2X messages are signed with a vehicle’s private key, both actions and locations can be linked to its electronic identity. Further, the vehicle registration process allows authorities to map this identity to an individual which is needed for law enforcement purposes. For privacy protection, several approaches suggest a separation of the CA and identity resolution authority [28, 29]. However, identifying the driver and their home address can also be achieved by knowing the start and end location of trips [30]. Similarly, the continuous monitoring of messages allows location tracking by linking messages to the same sender [31].

Adversaries who exploit this can be governmental institutions or larger organizations that are involved in providing or maintaining the road infrastructure [32]. A pool of collected messages gives them enough information to compromise location privacy and hence the driver’s identity. It can be assumed that in an urban environment, the maintenance of the road infrastructure is done by one organization which makes such attacks possible.

Location privacy can be achieved by providing anonymity to vehicles while communicating [24]. Under the requirements of authentication and non-repudiation, pseudonym schemes are the most promising approach. In these schemes, identifying material, such as MAC, IP addresses, and other intrinsic communication stack data, is removed from messages [33]. Further, a vehicle is provided with multiple instead of only one elliptic curve cryptography (ECC) identity which can be used interchangeably to sign VANET messages. The system presented in this paper implements this privacy requirement and only provides linkability within a vehicle formation while keeping messages unlinkable for an outside observer.

2.3. System Model

Our system model provides the assumptions that apply to all network participants. Then, we define five entities relevant to our system: a vehicle formation as a sequence of vehicles, leader, member, candidate vehicles, and driving trajectories.

**Network entities.** The vehicular network we assume is made up of OBU’s and RSUs. Each is in possession of at least one ECC key pair and a certificate from a trusted CA. The certificate allows an entity to verify the validity of public keys. Message integrity can be checked by a recipient through its ECDSA signature. Each ECC key pair can be linked to a unique identifier, however we assume that a pseudonym scheme is deployed to protect the location privacy of individuals. Thus, each vehicle holds a set of key pairs and several identifiers that can be used to sign messages. We do not limit the selection of the pseudonym scheme and assume that a vehicle can arbitrarily choose to change the identity it uses. Every vehicle holds a copy of a map for its geographic location which can for example be obtained from a publicly available source such as the OpenStreetMap (OSM) project [34].

**Vehicle formation.** The vehicle formation is a sequence of vehicles led by a LV. Each vehicle might have a different origin and destination, and therefore join and leave the formation at arbitrary times. However, at minimum they have a mutual intermediate section of their route. To become a verified member of the formation, a vehicle is required to follow the group and provide evidence for this maintained co-presence. A vehicle formation also provides a cryptographically protected communication channel. Members propagate their pseudonym changes over this channel, allowing them to skip the re-authentication of these new identities which can immediately be used to authenticate VANET messages. In addition to providing co-presence verification, this method also reduces false positives of Sybil attack detection systems [35].

**Leader vehicle.** The LV is the first vehicle in the sequence. It is the founder of the formation and announces its leadership offer periodically through beacon messages that include its location. We note that as soon as a vehicle receives a status beacon indicating another vehicle’s proximity, the first vehicle can become a LV. A LV provides its following vehicles with the information to authenticate the ongoing co-presence property of VANET messages (for example CACC, EEBL, or CCA messages). To guarantee the safety and privacy of all vehicles inside the formation, the LV verifies their ongoing co-presence. Through continuous verification, if it detects when a FV leaves, revokes its access to the internal communication, and informs the remaining members to update their set of verified identities.

**Following vehicle.** A FV is a vehicle that intends to share parts of its future route with the formation and has passed the ongoing co-presence verification. It has access to the internal communication using a session established with the LV. It leverages the information disseminated through the vehicle formation’s communication to keep an up-to-date list of verified identities. For instance, this enables the identification of relevant safety messages. The ongoing co-presence between a FV and its LV is continuously verified through beacon messages and trajectory comparison.

**Candidate vehicle.** A CV is an unverified vehicle that is not associated with a leader vehicle. To join a vehicle
formation, a CV compares its intended trajectory with the planned route of the LV. If their routes overlap, the CV can decide to start the co-presence verification process by continuously comparing its past trajectory with the LV. After confirming their co-presence, the CV becomes a FV. When the candidate reaches its destination, or the routes do not concur anymore, the vehicle leaves the formation and continues independently.

**Trajectory.** The intended trajectory of a vehicle is its future route through the road network, can be directly derived from navigation information. While this trajectory is used to probe whether two vehicles intend to travel the same route, their past trajectory is used as proof for ongoing co-presence. The past trajectory is the route a vehicle took through a road network. It is described by a sequence of turns and straight movement. The trajectory of a vehicle can be acquired using inertial measurement units (IMU) like gyroscopes to determine turning angles. Depending on the length and the number of turns, the trajectory can uniquely describe a route on the map and implicitly a location. Location ambiguities are defined as collisions and occur when a trajectory matches to more than one location.

### 2.4. Threat Model

For our threat model, we assume attackers possess ECC credentials to produce valid VANET messages but are limited to polynomial time and thus cannot compromise asymmetric keys or symmetric cryptography. We recognize that an attacker can either pose as a CV or as a LV by disseminating beacon messages announcing themselves as a leader. In both cases, we assume a stationary attacker or an attacker who is driving in a different direction passed by the victim. The adversary is considered to be within wireless transmission range for the duration of the attack. The correct behavior of computing devices is ensured using mechanisms such as remote attestation or trusted computing. For the evaluation of our approach, we categorize adversaries with respect to the following attack goals.

**Identity and location privacy.** This attacker aims to compromise the identity and upcoming location privacy of a driver. Therefore, he uses the current location and heading of its victim to provide a valid trajectory for future route comparison. He accomplishes this, if the forged trajectory matches the victim’s trajectory. Then, the victim recognizes him as co-present which allows the attacker to participate in the ongoing co-presence verification phase.

**Trusted following vehicle.** To become a FV, the attacker has to provide the victim with a matching past trajectory. He can use the location and heading and attempt to infer the victim’s movement. An attacker who becomes a verified FV gains knowledge of the identities of other formation members, allowing him to link their pseudonyms and thus defeat the pseudonym scheme. Furthermore, the attacker can introduce malicious identities that are then trusted by all FVs. The following vehicles will falsely authenticate messages signed with these identities assuming the co-presence of their senders. As a result, an attacker can send messages with falsified location claims to interfere with safety critical systems [12], [13]. Additionally, he can use the past trajectory to infer the victim’s home or work location [14], [15] as well as identify drivers through their commuting habits [25].

Once verified, FVs are considered trusted. Attacks from the inside of the vehicle formation can be mitigated using misbehavior and intrusion detection; for these, we refer the reader to [12] as we consider them out of scope for this paper. Further, we do not consider attacks against our system in which a verified FV colludes with an attacker by relaying trajectory information or disclosing its cryptographic key. A FV who would do so would not only jeopardize its own safety, but also already have access to the internal communication.

### 2.5. Design Goals

In this section we set the design goals relevant to the implementation of the method described in our application scenario. The considerations related to the security, privacy and efficiency of our approach will be revisited in Section 4 throughout the security analysis and performance evaluation.

**Co-presence verification.** Only vehicles that share the intended trajectory of the LV and are able to provide continuous proof of ongoing co-presence must be recognized as a verified FV. An attacker who supplies a fraudulent trajectory must be detected and rejected by the co-presence verification protocol. The system needs to ensure that the trajectory required to join the vehicle formation uniquely describes the current location, and is also resilient against inference by an attacker. Trajectory generation should not solely rely on external signals, for example GPS, as an attacker could spoof these signals [36] and make vehicles believe they are driving in formation.

The leader vehicle must establish a secure communication channel that is used to transmit the identities of FVs. It should not be possible for an attacker to read or insert messages. The credentials provided by the LV are tied to a vehicle’s identity to fulfill non-repudiation and allow selective exclusion of participants.

After a vehicle becomes a FV, they mutually and continuously verify their ongoing co-presence using beacon messages. The protocol design has to ensure that only the legitimate vehicles can provide this proof. In the case where a following vehicle leaves the formation, its credentials and identities have to be revoked from the set of trusted vehicles. Equally, if a leader vehicle abandons its group, all FVs have to recognize such an event and invalidate their sessions. It is not sufficient for a member vehicle to indicate its departure from the formation as an attacker could try to pretend he is co-present for longer than he actually is by suppressing such messages.

**Identity and location privacy.** The leakage of location information can reveal an individual’s habits, activities, and even lead to physical harm [37]. As the location of a vehicle
can be inferred based on the route it has traversed [15, 38], taking part in our protocol must not reveal a vehicle’s intended or past trajectories. Furthermore, parameters exchanged during the protocol should not enable a new attack pathway allowing tracking or localization, and compromise the victim’s privacy. Thus, the protocol must work even without linkable VANET messages, in particular when beacon messages are periodically exchanged as this would allow an attacker to link the pseudonyms in use.

**System Efficiency.** Without human interaction, the system should only be triggered if two vehicles plan to share the next section of their journey. If they start the verification process, the number of messages required for its completion, and their computational complexity, has to be kept to a minimum to maximize the benefits of becoming a FV. The duration a CV has to follow the formation until its co-presence has been verified should be minimized. The overhead added by the ongoing co-presence verification through beacon messages should be negligible in comparison to the VANET communication of other applications, such as status beacons sent at 10 Hz.

### 3. System Architecture

Our system consists of two phases: (1) destination matching and (2) ongoing co-presence verification. The data flow between the different components is shown in Figure 2. In this section, we first describe the generation and comparison of intended and past trajectories which does not yet consider privacy preservation. Then, we present our co-presence verification protocol allowing us to perform these operations without compromising either of the parties’ identity or location privacy. Finally, the protocol is designed to work independently from the participants’ identities, to enable the use of a pseudonym scheme while providing linkability of messages within the vehicle formation.

#### 3.1. Intended Trajectory

Given a map of the road network and a vehicle’s current location, the intended trajectory is that section of its overall journey that has not yet been traversed. Assuming the driver is going to a specific destination, the journey can be determined using an integrated navigation system, for example based on OSM data. The comparison of two intended trajectories allows the participants to determine whether their future routes overlap for at least the next $d_{in}$ meters. As we will show in the evaluation section, an intended trajectory can be used to uniquely identify its starting location and thus assert spatial proximity.

**Representation.** Road networks are commonly represented as a graph, $G$, consisting of vertices (intersections and bends), $V$, and edges (roads), $E$. As shown in Equation 1 we represent an intended trajectory, $T_{in}$, through a road network as the sequence of vertices it passes for a given length $d_{in}$ starting at $v_0$.

**Generation.** Figure 3 shows a road network based on OSM data. To describe the intended trajectory of the vehicles, $LV$ and $CV$, located on the straight segment leading to $v_0$, and the future route represented by the red dashed line, the graph is traversed to obtain the sequence of vertices. The trajectory considered terminates after the vehicle’s destination or $d_{in}$ is reached.

$$V = \text{latitude, longitude}, E = (v_i, v_j), G = (V, E)$$
$$v_0 = \text{start}$$

$T_{in} = (v_0, v_1, ..., v_n); \forall v \mid \text{dist}(v_0, v) \leq d_{in}$

#### 3.2. Past Trajectory

The past trajectory of a vehicle is used as proof for ongoing co-presence. Due to the inaccuracy of GPS coordinates [39], and their susceptibility to spoofing attacks [40], we use gyroscopes to identify a vehicle’s turns and generate trajectory sequences. This allows us to capture variations in driving speed as well as obstacles such as parking cars and buses. Given that IMUs are on-board elements, they are resilient against most physical attacks as they do not acquire external signals and readings can be processed directly on the device. During ongoing co-presence verification, the two vehicles periodically challenge each other to compare the latest segment of their past trajectory.

**Representation.** We represent the trajectory of a vehicle as a sequence, $Seq_{pa}$, of movement segments. These segments are constructed as a tuple that includes a qualifier ($p$) and a quantifier ($q$). As shown in Equation 2, qualifiers distinguish between an inter-turn travel duration ($D$), as well as left ($L$) and right ($R$) turns. We use degrees to quantify angles and seconds to quantify travel duration. The elements in the sequence are in descending order by time of occurrence. We define $T_{pa} = (p, q)_0$ as the latest segment in the sequence.

$$p \in \{L, R, D\}$$

$$q = \begin{cases} q^{\circ} \in [1, 180], & \text{if } p \in \{L, R\}; \\ q^{s} \in \mathbb{N}, & \text{if } p \in \{D\}. \end{cases}$$

$Seq_{pa} = ((p, q)_0, (p, q)_1, (p, q)_2, ...)$

$T_{pa} = (p, q)_0$

#### Comparison.** The data underlying a sequence of vertices $T_{in}$ is not subject to measurement errors, hence the comparison of two intended trajectories can be computed as an exact match of the sequences. In order to guarantee that the trajectories only match if at least $d_{in}$ meters overlap, a cryptographic hash function, $H()$, is used to compute the comparison as $H(T_{in}^{LV}) = H(T_{in}^{CV})$.
Figure 2: Map data and IMU measurements are used to generate intended and past trajectories. Their privacy preserving comparison allows two vehicles to first determine whether their future routes are going to overlap for the length $d_{in}$ and to establish a symmetric key for ongoing co-presence verification. After a CV has been verified over a distance $d_{pa}$, and thus been promoted to a $FV$, their ongoing co-presence is continuously verified.

Figure 3: Map segment with an overlay of the road network. The $LV$ and $CV$ are currently driving towards $v_0$. Vertices and edges in red outline their future route used for intended trajectory comparison.

For turn generation, we use angular velocity $(rad/s)$ which is provided by the gyroscopes along three orthogonal axes. Since a vehicle drives parallel to the ground, the axis pointing towards the sky is relevant to detect its turns. With only this axis as input, we use a peak detection algorithm [41] to identify spikes in the readings. We choose this approach over a continuous integration of the angular velocity as it does not suffer from cumulative errors inherent to inertial sensors due to thermal and mechanical disturbances. On the downside, this approach cannot recognize very slow turns. However, as we will show in our evaluation section, this does not appear to pose a problem when used in vehicles in an urban environment. The resulting peaks are then segmented and their turning angle quantified through integration. Figure 4 shows the classification result with green and red triangles, denoting right and left turns, and peak boundaries indicated with black lines. Orange areas provide the inter-turn travel duration which is obtained by computing the time elapsed between peak boundaries.

Comparison. To verify the ongoing co-presence of two vehicles, the latest segment of their past trajectories have to match. Hence, the comparison of these is computed as an exact match $T_{LV} = T_{CV}$. However, due to variations in IMU measurements, this matching is affected by two types of errors.

An order preserving error occurs when a past segment is of the correct quality but the quantities do not match. Such errors can be caused by sensor inaccuracies that lead to different turning angles, or by temporal variations like an obstacle on the road delaying one vehicle but not the other.

An order disturbing error changes the sequence number
of a segment and thus the qualifiers of the two segments do not match. The reason for such an error is a direction change that has not been identified by the peak detection algorithm. This occurs when the two vehicles turn at different speeds.

These errors can lead to a false reject of a verification attempt. We will investigate the impact and frequency of such errors through experiments in our evaluation in Section 4.

3.3. Co-Presence Verification Protocol

The protocol design reflects the mutual authentication of the LV and the CV followed by the ongoing co-presence verification of FVs. Each operation is performed symmetrically and abort decisions can be made by both parties. To mitigate offline guessing attacks, the protocol requires the other party’s involvement for each comparison attempt. The procedure itself is divided into two phases to accommodate the different stages of the CV: Phase 1 matches intended trajectories of length \( d_{in} \) and establishes a symmetric key between the LV and CV. In phase 2, the initial co-presence of the two vehicles is verified with the past trajectory over the distance \( d_{pa} \). If successful, the symmetric key is confirmed and the CV is promoted to a FV after which ongoing co-presence is periodically verified. In case a FV leaves the transmission range of its LV or cannot provide the correct past trajectory, it gets de-authenticated and demoted to a CV. Thresholds for \( d_{in} \) and \( d_{pa} \) are used by all participants and their impact is presented in our evaluation section.

We assume every vehicle has a set of pseudonyms, ECC public and private key pairs \((K_A^+, K_A^-)\), where \( A \) denotes the associated identity. During phase 1, we omit signature verification steps and assume that OBUs verify all messages using a PKI. Furthermore, to preserve the functionality of the pseudonym scheme, our protocol does not require a linkable stream of ECDSA signed messages that would allow an attacker to link the pseudonyms of one sender. In phase 2, which requires the continuous exchange of beacon messages, we use symmetric cryptography based message authentication to prevent an attacker from linking messages [43], while providing internal linkability to the formation using the key established in phase 1.

**Phase 1: Intended trajectory matching.** As the formation coordinator, the LV announces the group’s presence periodically as a beacon message that includes location data. The first phase is triggered when a CV receives this beacon and finds that the vehicle formation is in proximity. Before ongoing co-presence verification is initiated, both parties determine whether they are going to share the next \( d_{in} \) meters of their intended trajectory. After this phase, both parties were able to either establish a symmetric key in case of a match or abort the protocol. The trajectories themselves are not disclosed to protect the vehicles’ future location. To provide a private comparison, our design like [43] assumes the following for LV and CV:

- A priori generated RSA key pairs with private exponents \( d_{LV}, d_{CV} \), public exponents \( e_{LV}, e_{CV} \) and modules \( n_{LV}, n_{CV} \).
- A full-domain hash \( H() \mid H : 0, 1^* \rightarrow \mathbb{Z}_n^* \) and cryptographic hashes \( H'(()) \) and \( H''() \).
- A generator \( R \leftarrow \mathbb{Z}_n^* \).

The different steps of this phase are outlined in Listing 1. Curly brackets denote transmitted messages and their signatures. In step 1, LV and CV exchange the public parts of their RSA key pairs and agree on a \( d_{in} \). In step 2, both vehicles hash and blind their inputs to \( Y_{LV} \) and \( Y_{CV} \). Upon receiving these elements in step 3, each party signs them, to obtain \( Y'_{LV} \) and \( Y'_{CV} \). The hashed and signed elements \( K_LV \) and \( K_{CV} \) are used to compute the tags \( T_{LV} \) and \( T_{CV} \) as well as to encrypt a nonce with a symmetric cipher like for example AES. In step 4, both compare their respective tags and, in case of a match, each party can recover the respective keys by reversing the blinding operation and decrypt the nonces. Finally, the symmetric key \( SK_{CV,LV} \) is derived from a hash over the nonces, nonce_{LV} and nonce_{CV}. The correctness of this protocol is analogue to the proof in [44] for private set intersection with data transfer.

**Step 1**

\[
LV \rightarrow CV: \{n_{LV}, e_{LV}, d_{in}\}_{K_LV} \\
CV \rightarrow LV: \{n_{CV}, e_{CV}, d_{in}\}_{K_{CV}}
\]

**Step 2**

\[
LV \rightarrow CV: \{Y_{LV}\}_{K_{CV}} = H(T_{in}) \cdot (R_{LV})^{e_{LV}} \mod n_{LV} \\
CV \rightarrow LV: \{Y_{CV}\}_{K_{LV}} = H(T_{in}) \cdot (R_{CV})^{e_{CV}} \mod n_{CV}
\]

**Step 3**

\[
LV: K_{LV} = H(T_{in})^{d_{LV}} \mod n_{LV} \\
LV \rightarrow CV: \{Y'_{CV}\}_{K_{LV}} = (Y_{CV})^{d_{LV}} \mod n_{LV} \\
\{T_{LV}\}_{K_{LV}} = H'(K_{LV}) \\
\{S_{LV}\}_{K_{LV}} = AES(nonce_{LV}) \cdot H''(K_{LV})
\]

**Step 4**

\[
CV: K_{LV} = H''(Y'_{CV}) \mid H'(Y'_{CV}) = T_{LV} \wedge T_{LV} = T_{LV}' \\
nonce_{LV} = AES^{-1}(S_{LV}, K_{LV}) \\
SK_{CV,LV} = H''(nonce_{LV}, nonce_{CV})
\]

\[
LV: K_{CV} = H''(Y'_{LV}) \mid H'(Y'_{LV}) = T_{CV} \wedge T_{CV} = T_{CV}' \\
nonce_{CV} = AES^{-1}(S_{CV}, K_{CV}) \\
SK_{CV,LV} = H''(nonce_{LV}, nonce_{CV})
\]

Listing 1: Phase 1 of the protocol privately compares the intended trajectories \( T_{CV} \) and \( T_{LV} \) for a given length \( d_{in} \). Participants only learn about an exact match, which is used to derive a symmetric key \( SK_{CV,LV} \).

After the execution of this phase, the vehicles can use the established symmetric key to sign messages intended for members of the formation, using a keyed-message authentication code (HMAC). A HMAC reduces the overhead of
message authentication significantly while not requiring the use of pseudonyms [43]. Messages sent in phase 2 will use this signature mechanism, as shown in Listing 2 instead of ECDSA with their ECC credentials.

\[
CV \rightarrow \ast: \{msg\}_{HMAC_{SK}}
\]
\[
LV \rightarrow \ast: \{msg\}_{HMAC_{SK}}
\]

Listing 2: VANET messages intended for formation members can be signed and authenticated using a HMAC and the symmetric key \(SK\). This is used in phase 2.

**Phase 2: Ongoing co-presence verification.** Phase 2 of our protocol uses trajectory beacon messages to verify ongoing co-presence of two vehicles. A similar approach based on VANET status beacons has been proposed by Studer [23]. However, their approach is not suitable for urban environments and also relies on majority vote which makes it susceptible to collaborating attackers. Furthermore, it requires re-authentication whenever a vehicle changes identity and can hence not be combined with pseudonym schemes without a significant loss in efficiency.

Instead of using status beacons, we use messages that contain a vehicle’s latest trajectory segment \(T_{pa}\). To verify the co-presence of two vehicles, they have to mutually compare their \(T_{pa}\) without disclosing the correct value to the other party. We deploy a lightweight version of phase 1 that reuses the parameters transmitted in step 1, skips the symmetric key derivation, and uses the last trajectory segment, \(T_{pa}\), instead of \(T_{in}\) as the payload. The \(LV\) and \(CV\) send mutual challenges whenever the qualifier of their \(T_{pa}\) changes. Each vehicle compares the message content with its latest trajectory segment. The session is terminated if there is a mismatch or the vehicles are out of range, otherwise the verification continues. Once the \(CV\) has successfully been verified for a distance, \(d_{pa}\), it is promoted to a verified \(FV\). Afterwards, the verification continues to detect when one of the vehicles is no longer co-present.

The presented two-phase protocol matches the intended trajectory of two vehicles in a privacy preserving way. After establishing a symmetric key, both vehicles verify their ongoing co-presence based on their past trajectory without disclosing it to a third party or a vehicle that offers a fraudulent trajectory. By replacing ECDSA with a HMAC signature mechanism, the identity of participants is protected without additional burden on the pseudonym scheme. The symmetric key can also be used to establish a secured communication channel to propagate pseudonym changes to co-present vehicles without infringing location privacy. We evaluate the values for \(d_{in}\) and \(d_{pa}\) and discuss their impact, in the security analysis of our evaluation in Section 4.2.

**4. Evaluation**

The evaluation of our approach is conducted in three steps: (1) map analysis, (2) security analysis, and (3) real-world experiments. Using map data, we first show the generalizability of our system and quantify an attacker’s search space for intended trajectories. Then, we use vehicle simulation as well as real-world experiments to analyze the resilience against attacks on the two phases of our protocol and assess the performance of our system. In each section, the respective results are presented and discussed. Finally, we conclude with a summary of our findings.

**4.1. Map Analysis**

OSM data [34] is used for the map analysis of this system to study the topology of road networks across different cities. This allows us to investigate city specific characteristics and assess the generalizability of our approach. An adapted Manhattan mobility model is then presented and subsequently used for trajectory generation and simulation in the security analysis section. Next, the impact of the number of turns and overall length on trajectory uniqueness is evaluated. The clearer a trajectory can be described, the fewer false positives, i.e. trajectories that match in shape but not in location are accepted in phase 2 of our protocol. Throughout our analysis, we call such trajectories collisions.

**Road network topology.** OpenStreetMap data provides a road network as a graph of vertices and edges. Vertices are annotated with GPS information and occur at intersections or bends in the road. Edges connect two vertices and specify a road type, for instance a cycling lane, sidewalk or normal road. We only consider roads accessible by car and assume
that roads can be traveled in both directions. Using the GPS information, the length of road segments and turning angles are computed as displayed in Figure 6. We define turning angles as heading change, in this example a turn of $43^\circ$ occurs for a vehicle turning around $V_2$ from $V_1$ towards $V_3$.

To characterize the layout of road networks, we compute histograms for a total of four large, medium, and small sized cities. These were chosen from North America, Europe, and Asia to eliminate country specific architectural differences. Since roads are bidirectional, turns around a vertex are symmetric and thus considering only one turning direction is sufficient. In reality, one way streets may prohibit certain turns, however their number is negligible in comparison to the total number of roads in a city. Our results shown in Figure 6 are obtained through graph traversal and GPS based angle computation. They display the absolute number of turns for each angle between 0 and 180 degrees in all four cities with their respective area provided in $km^2$.

In general, the distributions expose a similar shape with a large number of small turns and a low number of large turns. A peak around the $90^\circ$ mark can be observed in all four cases which seems natural, particularly in North America due to the grid like road structure. However, the figures of the other cities suggest that the rectangular shape of buildings causes a similar effect in Europe and Asia. The larger number of smaller turns can be explained by straight sections and longer curves, separated into several segments. These figures indicate that the turns in a city are independent of its road layout and areal size. As a result, they suggest that our approach is transferable between different cities. We use these four cities throughout our security analysis to confirm the generalizability of our system.

Mobility model. For the subsequent trajectory feature analysis and the simulation in the next section we adapt the Manhattan mobility model [45] for arbitrary turning angles as shown in Figure 7. Straight movements are equally likely as taking a turn, and turns to the left are as likely as turns to the right. In the case, where a vehicle reaches an impasse, it is allowed to turn around. We consider angles ranging between $+30^\circ$ and $-30^\circ$ as straight movement, turns between $+30^\circ$ and $+150^\circ$, and $-30^\circ$ and $-150^\circ$ as left and right turns respectively. Turns larger than $150^\circ$ are considered backward movement.

Trajectory feature impact. To evaluate the feasibility of turns and straight segments as trajectory features, we test two sets of trajectories per city by computing the average number of collisions within their respective road network. For the first experiment, their overall length is fixed to 50, 100, 200, 300, 400, 500, 1000, 1500 and 2000 meters. The results presented in Figure 8a indicate that with an increase in number of turns, the number of collisions decreases in all four cities. It suggests that combining multiple turns improves the uniqueness of a trajectory. Additionally, this parameter appears to be independent of the city size with City C, the second to smallest, starting off at around 3000 collisions and City B, the largest, at around 1000 collisions. For trajectories with 9 turns, City C is still above the other cities with 60 versus 20 collisions on average. Trajectories in City D that include 9 turns were found to have the least number of collisions. Given that City D has a smaller absolute number of turns, it is natural that its trajectories are more unique than when drawn from a larger city with a greater number of turns.

Next, the impact of a trajectory’s overall length is evaluated by drawing 9000 trajectories of 9 different lengths from each road network. Figure 8b shows the results as average number of collisions per trajectory. The results suggest that this number depends on two factors: the area of the city and the length of the trajectory. With increasing area, a city has a larger number of roads which inherently allows for more collisions. However, if the length increases the exact combinations of turn and straight segments become rarer and the number of collisions decreases. Overall, these findings confirm that a trajectory through a road network can be described with turns and straight segments.

4.2. Security Analysis

Security analyses of the two phases of our protocol are provided in this section. We simulate vehicle movement on the road networks for cities A, B, C and D based on the aforementioned mobility model. We first investigate the effect of values for $d_{in}$ on the attack success rate against phase 1. To evaluate phase 2, we compare two methods: status beacon based and trajectory beacon based. We show that our approach improves over [23] in urban environments. After presenting the results for all four cities, we conclude...
An attacker of this phase knows the victim’s current location, heading and speed using VANET status beacons. He uses this information to generate a fraudulent trajectory that is most likely to match the victim’s data. A successful attack, \textit{i.e.}, a false accept of an intended trajectory, compromises the future location privacy of the victim and allows the attacker to proceed with an attack on phase 2 of the protocol.

We conduct simulations on the OSM data for all four cities. Starting from a given vertex, the victim’s intended trajectory is chosen based on our mobility model. Afterwards, 100 attackers are deployed at the same vertex which could be malicious vehicles or RSUs. Each attacker tries to infer the correct trajectory by traversing the graph and applying the probabilities from the mobility model at each intersection. We simulate this attack for each vertex in the graph that has at least two adjacent vertices, \textit{i.e.} is not an impasse, and the resulting false accept rates with respect to $d_{in}$ are displayed in Figure 9. The curve for each city starts at the average edge length of the respective road network. The chance of correctly predicting the first intersection depends on the number of outgoing edges, \textit{i.e.} how interconnected the roads are. For cities B and D, the figures suggest that there are predominantly T-junctions that lead to the attacker having close to 50% chance for successfully predicting $T_{in}$ for small $d_{in}$. In general, the FAR drops for all four cities and reaches 5% for City D at 250 meters, cities A and C around 350 meters and City B at 450 meters. The slower decrease for City B suggests that its road network is less interconnected than for the other three cities and an attacker has less choices over the same distance.

To summarize, by choosing a large enough value for $d_{in}$, only participants that actually intend to take the same route manage to proceed to the next phase of the protocol. For a small $d_{in}$, an attacker gets admitted to phase 2 more easily and obtains the next $d_{in}$ meters of the victim’s route.

\textbf{Phase 2: Ongoing co-presence verification.} The second phase of the protocol verifies the ongoing co-presence between two vehicles that have successfully passed phase 1. Both periodically exchange beacon messages and compare the latest segment of their past trajectory. A new CV has to successfully prove its ongoing co-presence for $d_{pa}$ meters until it is promoted to a verified FV. The LV and all FVs continue this verification periodically. In case a trajectory segment does not match or one of the parties is unable to provide a beacon, \textit{i.e.} is out of range, they are de-authenticated. We assume that an attacker targeting this phase does not know the victim’s past trajectory as he is otherwise by definition not violating the co-presence property. This leaves two types of adversaries for the following analysis: an attacker who passed phase 1 and is stationary or driving in a different direction within transmission range, and a malicious or misconfigured FV that does not notify the LV about its departure from the formation. A successful attack on this phase allows a malicious CV to become a verified FV, as well as a malicious FV to retain access to

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure8}
\caption{Trajectory feature impact: (a) Average number of collisions per trajectory for each city from 9000 randomly chosen trajectories with three, six, and nine turns per trajectory. (b) Average number of collisions per trajectory for each city from 9000 randomly chosen trajectories of 9 different lengths.}
\end{figure}

with the cumulative false accept rate (FAR) which describes an attacker successfully passing both phases of our protocol.

As indicated by the previous section, an attacker who is aware of the victim’s location has a significantly higher likelihood to be successful. Therefore, we only consider malicious entities that are close to the victim, such as stationary vehicles or vehicles driving in the opposite direction. We note that we do not distinguish between a LV and the CV in the role of the victim in phase 1, since the attacks are analogue for both cases. Victims in phase 2 can also either be the LV or a FV, as a leader can abandon its vehicle formation and a following vehicle can diverge, both cases have to be detected by the other party.

\textbf{Phase 1: Intended trajectory matching.} The first phase of the protocol matches the destination of the LV and the CV for the next $d_{in}$ meters of their intended trajectory. The two vehicles privately compare this trajectory, and only commit to the next phase of the protocol in case of an exact match. An attacker of this phase knows the victim’s current location, heading and speed using VANET status beacons. He uses this information to generate a fraudulent trajectory that is most likely to match the victim’s data. A successful attack, \textit{i.e.}, a false accept of an intended trajectory, compromises the future location privacy of the victim and allows the attacker to proceed with an attack on phase 2 of the protocol.
For the following analysis, we compare two approaches. One that is based on status beacons [23], and our system based on past trajectory beacons. Both assume a limited range of 250 meters for DSRC transmissions, a beaconing frequency of 1 Hz, and normal driving speeds within city limits. We model the attacker as a form of Bayesian Stackelberg game [46] with two participants: the victim and the adversary. The simulation is performed in turns. During each turn, the victim moves, followed by the attacker, and both are governed by the mobility model. After a turn is completed, their trajectory segments are matched and the transmission range is checked. A false accept occurs if the attacker satisfies the range check and successfully matches the segment of the victim.

We start by comparing both: status beacons and past trajectory beacons with simulations in City D to show their difference in performance. The result in Figure 10 shows the false accept rate with respect to the distance traveled by the victim. As can be seen, the first de-authentication occurs earlier for trajectory beacons as they can already mismatch even if both vehicles are still within transmission range. For both approaches, the FAR declines with increasing distance and falls below 5% after 290 meters for the past trajectory beacons and after 1500 meters for the status beacon based method. As the past trajectory beacon method showed improved performance compared to status beacons in City D, we simulated this system on the remaining cities to verify these results. As displayed in Figure 11, a similar behavior is observed for all four cities. The different duration until the first de-authentication occurs can be explained by different average edge lengths across cities. Only after the first edge
Figure 13: Experimental route with an overall length of 5.5 km, selected for the analysis of phase 2. An impasse at A serves as start and endpoint.

<table>
<thead>
<tr>
<th>Number of runs</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average run length</td>
<td>5256 m</td>
</tr>
<tr>
<td>FRR for 30% FAR</td>
<td>7%</td>
</tr>
<tr>
<td>FRR for 5% FAR</td>
<td>10%</td>
</tr>
</tbody>
</table>

TABLE 1: Summary of the parameters and results of our experiments.

is traversed, the attacker has a chance to predict the wrong trajectory as we assume the attacker knows heading and location of its victim at the start.

Overall, to evaluate the security of our system, we combine the FARs of the two phases by multiplying their probabilities. The respective parameters $d_{in}$ and $d_{pa}$ are added to show the overall distance traveled, since phase 2 can only begin after the vehicles drove for $d_{in}$ meters. The combined FARs are shown in Figure 12. City D exhibits the fastest decline and reaches a FAR of 5% after 270 meters while City B and C fall below this mark at around 410 meters. In an urban environment, such distances respectively take 20 and 30 seconds to travel, after which a CV can be promoted to a FV. Additionally, we note that the performance of this method for City B is almost equal to that for City C when considering the combined probabilities, yet City B shows the worst FAR of all cities when phase 1 is considered separately. Lastly, we emphasize that the combined probability is reflective of an attacker successfully passing through both phases of the protocol. Yet, Figure 11 remains applicable in the case of a misconfigured FV that does not notify the LV of its departure from the formation.

4.3. Experiments

The efficacy of phase 2 depends on the accuracy of the past trajectory generation algorithm. In this section, we describe real-world experiments conducted to investigate the false reject rate (FRR), i.e. mismatches during the verification of two co-present vehicles. To acquire the gyroscope readings of two vehicles, we implemented an Android application that transforms those measurements into trajectory sequences, $T_{pa}$. For each experiment run we obtained two sequences which are subsequently compared, yielding the error rate.

For our experiment runs, we selected the route outlined in Figure 13 with an overall length of 5.5 km. We rented the two cars shown in Figure 14a and placed smartphones in the center console of each vehicle as displayed in Figure 14b. We repeated this experiment for a total of nine iterations throughout an entire day featuring rush hour as well as midday traffic conditions. An impasse at location A is used as start and endpoint to allow adjustments between runs.

Mismatches directly affect phase 2 of the protocol since a vehicle that provides a segment that does not exactly match is de-authenticated. An example of gyroscope measurements for LV and CV from one of the experiment runs is given in Figure 15. As depicted, displacement or differences in amplitude can introduce errors in the past trajectory sequence, causing a false reject. The FRR for phase 2 depends on the required distance, $d_{pa}$, before a CV is promoted to a FV. Larger values have a smaller FAR, however they expose a higher chance to be affected by a mismatch while smaller $d_{pa}$ are more robust and hence provide a lower FRR. This correlation between FAR and FRR is shown in Table 1. We present the values for FARs of 30% ($d_{pa} = 120m$) and 5% ($d_{pa} = 250m$) which result in FRRs of 7% and 10%, respectively. These numbers apply when we solely consider phase 2. As shown in the previous section, the combined FAR can be lowered depending on the value chosen for $d_{in}$ and hence improve the overall FAR while still maintaining a low false reject rate.

5. Related work

VANET security and message authentication are becoming increasingly important as these are integrated into modern road networks. At the same time, the vehicle platoon is only one of the emerging approaches that rely on verified location claims to ensure its safe operation. In this section, we first review work related to attacks on VANETs and then present defense mechanisms from recent research.

Attacks on VANETs. The security standard IEEE 1609.2 does not foresee the verification of physical properties such as co-presence. Hence, one of the major attack vectors are
impersonation attacks with falsified location claims. In these attacks, messages from non-existing vehicle are inserted into the traffic to disrupt safety applications, for example EEBL, CCA, and CACC. As described by Bißmeyer et al. [13], such ghost vehicles can be generated by an attacker who disseminates forged messages via a malicious RSU.

Attacks on CACC in particular can be used to target a larger group of vehicles simultaneously. In their study on vehicular platoon misbehavior, DeBruhl et al. [12] identify several attack strategies: mis-report, collision induction, and reduced headway. Generally, these attacks aim to degrade the performance of a platoon or reduce its string stability. In fact, small fluctuations in acceleration get amplified along a platoon with low string stability, and can ultimately lead to crashes and fatalities.

With the roll-out of RSUs and the monopolization of road network providers, location privacy becomes increasingly important. In their survey, Peti et al. [24] categorize different attackers according to their capabilities of tracking, locating and identifying individuals based on VANET messages. They present pseudonym schemes as a technique to protect drivers’ privacy, highlighting common attack vectors and performance shortcomings.

**Co-presence verification.** Defense mechanisms against impersonation attacks, in particular against falsifying location information, can be grouped under the term co-presence verification. It was first introduced as sustained co-presence by Miettinen et al. [17] as the continuous comparison of context parameters for device pairing. Its goal is to verify that two devices are within close proximity. They achieve this through periodic comparison of ambient luminosity. In the context of VANETs, co-presence can be used to assert that messages do not contain falsified location information.

In their work, Studer et al. [23] provide convoy member and vehicle sequence authentication by combining beacon messages with distance bounding. Their approach relies on majority voting to determine the validity of a location claimed in a VANET message. Each vehicle stores the identity used in beacon messages and only considers messages of vehicles that recently provided such a message. However, their approach has limited applicability since it does not support pseudonym schemes as each identity change would require re-authentication of the vehicle. It also faces limitations in urban environments.

The research by Han et al. [18] also aims to authenticate messages through physical context verification. In their report, they use the surface of the road as an external factor. Road conditions like bumps and potholes are measured by an accelerometer and used to derive a secret key, shared between co-present vehicles. Unfortunately, in their evaluation the entropy of the key material and the effect of different road surfaces remains unexplored making the system’s security claims unclear.

The authors of [8] and [21] rely on directional antennas to verify the position of a sender relative to the receiver. Their approach determines whether the sender is in front or behind the verifier. However, this method does not provide the receiver with the traveling direction of the sender and does not reveal its proximity without distance bounding. On the contrary, Song et al. [19] propose a combination of distance bounding, plausibility checks and ellipse-based location estimation. However, their method does not allow the receiver to determine the direction of the sender and thus limits the proximity verification to a circle around a vehicle. Distance bounding in general is a very sensitive and computationally heavy method that is not suitable for continuous verification.

Lastly, Juuti et al. [40] record trajectories as a second factor for authentication to ensure co-presence. A smartphone is used to recognize the location where it is allowed to grant access to credentials based on the bearer’s trajectory. After the co-presence is verified, it participates in a proximity based authentication scheme. In contrast to our approach, they store authorized trajectories to identify a location instead of verifying sustained co-presence.

### 6. Conclusion

We propose a novel approach for ongoing co-presence verification of vehicle formations in a privacy preserving way. Our two-phase protocol first allows vehicles to compare their future routes and then verifies their ongoing co-presence. After a candidate has successfully been promoted to a following vehicle, the group can authenticate VANET messages based on the trusted identities of formation members. This enables vehicles to defend against false location claims by an attacker who is stationary or driving in a different direction. Our approach is complimentary to existing safety applications such as EEBL, CCA, LDW, and CACC as well as transparent to privacy protection methods like pseudonym schemes. Through real-world experiments and realistic city-scale simulations, we showed the feasibility of our approach to verify ongoing co-presence with acceptable false reject rates (10%) while maintaining low false accept rates (5%) after only 20 seconds of driving.

In the future, we plan to investigate the propagation of DSRC messages in urban environments, and how their
shorter transmission ranges affect the status beacon and therefore our trajectory beacon based method. Moreover, we will experiment with multiple vehicles to explore options for following vehicles to transition from one leader to a new leader vehicle without having to re-authenticate.

References


