COLAW: Cooperative Location Proof Architecture for VANETs based on Witnessing

Philippos Barabas
philippos.barabas@tum.de
Technical University of Munich, Germany

Emanuel Regnath
emanuel.regnath@tum.de
Technical University of Munich, Germany

Sebastian Steinhorst
sebastian.steinhorst@tum.de
Technical University of Munich, Germany

***Abstract*—Vehicular applications heavily rely on location information to improve road safety and efficiency as well as to provide a personalized driving experience through a variety of location-based services. To determine their position, vehicles depend on different technologies like GPS, which might be unreliable or vulnerable to interference or spoofing. In the safety-critical vehicular world, a secure mechanism must be in place which guarantees the accuracy and trustworthiness of location information to the service that requires it.**

In this work we propose *COLAW*, a COoperative Location proof Architecture based on Witnessing that leverages the distributed nature of vehicular ad-hoc networks to create verifiable and secure location proofs. The evaluation of COLAW shows that it is possible for a group of neighboring vehicles to generate secure location proofs for each other with a significantly lower message overhead than previously proposed approaches and that the protocol's performance can be further improved, by taking certain environmental parameters and road conditions into consideration.

***Index Terms*—Cooperative Driving, Intelligent Transport, VANET, Location Proofs**

I. Introduction

For years, Location Based Services (LBSs) have played a major role in everyone's life. Navigation, shipment tracking and video games are just few use cases where users – in many cases unknowingly – share and retrieve location information. These added-value services are not only limited to smartphones and computers but start appearing in many types of devices, from smart TVs and smart watches to basically everything that has a Global Navigation Satellite System (GNSS) receiver and a connection to the Internet. Their growth is further accelerated with the development of 5G, Automated Driving, Smart Cities and the IoT, and is expected to reach an annual revenue of 195 billion EUR in 2025 [1].

In the road segment, which is expected to generate 50% of that revenue, GNSS-enabled In-Vehicle Systems already support a multitude of applications. Their increasing ubiquity as well as the introduction of platforms such as *Android Automotive* that ease application development will accelerate the integration of second- and third-party applications and transform the automotive industry similar to how the mobile industry was transformed in the recent years.

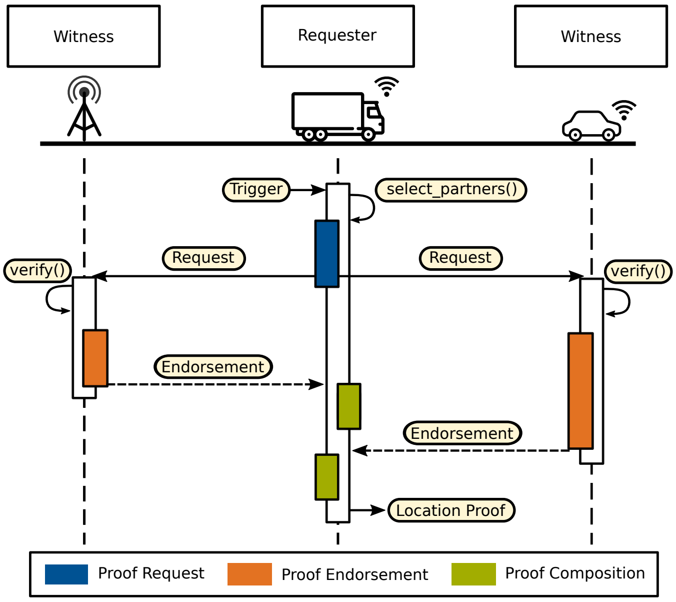

Figure 1: Conceptual overview of the COLAW system. The requester collects location endorsements from witnesses to create a location proof.

A. Contribution

Is this work we will introduce *COLAW*, a lightweight location proof architecture that leverages the ad-hoc connectivity of vehicles to generate secure location proofs from signatures collected by neighboring witnesses, which can be submitted at any given point in time to a verifier to prove the vehicle's position. Location proof architectures have been proposed for smartphones and mobile environments, but to the best of our knowledge, this is the first location proof architecture fully leveraging the benefits of the distributed nature of Vehicular Ad Hoc Networks (VANETs).

We will develop COLAW according to the latest European Telecommunications Standards Institute's standards on Intelligent Transport Systems (ETSI ITS) and conduct an analytical as well as experimental evaluation and comparison according to well-defined metrics and show that existing and future vehicular applications can significantly benefit from our proposed location proof architecture.

In particular, we

- propose a location proof architecture which leverages the

Symbol	Description
id_i	Pseudonymous identifier of vehicle i
lc_i	Location claim of vehicle i
$hlc_i = h(lc_i)$	Hashed location claim using hash function $h(\cdot)$
(key_i, key_i^{-1})	Public/Private cryptographic key pair of vehicle i
$\sigma_i(msg, key_i^{-1})$	Digital signature of message msg with private key key_i^{-1}
$\sigma_{j,i}(hlc_i, key_j^{-1})$	Digital signature of vehicle i's hashed location claim with vehicle j's private key
$cert_i$	Digital Certificate of vehicle i

Table I: Symbol nomenclature.

Function	Description
$select\_partners()$	Returns a list of ids from the LDM
$latest\_lc(id_i)$	Returns the latest location claim from the LDM for vehicle i
$endorse(id_i, hlc_i)$	Returns true if the LDM contains a lc for vehicle i whose hash corresponds to hlc_i. Returns false otherwise
$sign(msg, key_i^{-1})$	Returns the signature of message msg with private key key_i^{-1}
$verify(\sigma_i)$	Verifies signature σ_i. Returns true if successful, false otherwise.

Table II: Function description.

distributed nature of VANETs to generate digital evidence about the vehicle's presence. The algorithm is developed according to the latest ETSI ITS standards and does not rely on any additional hardware or trusted infrastructure to ease integration (Section II).

- conduct a theoretical analysis of the protocol to evaluate its cost and compare it with existing approaches for mobile environments (Section IV-A).
- examine the security and privacy implications of our architecture by examining whether common attacks against location proof architectures are prevented or not (Section IV-B).
- implement and test the algorithm in a simulated environment to analyze how different driving conditions and environmental parameters affect the protocol's performance (Section IV-C).

Furthermore, Section III discusses the related work and our conclusions are drawn in Section V.

B. Background

It is important to point out that COLAW is not a *location verification* protocol but a *location proof architecture*, also called *spatial-temporal attestation service*. Location verification protocols use different approaches to decide whether a location claimed by a user should be trusted or not, whereas location proof architectures aim at generating a digital evidence about the user's location that can be submitted to a verifier as a certificate of presence at a certain location. Location proof architectures often use location verification methods to assess the truthfulness of a location claim before generating a location proof.

Because of the safety implications that bogus information dissemination has in vehicular environments, protective measures should be taken on vehicular and system level to detect malicious or faulty nodes. Usually, they are distinguished between *node-centric detection mechanisms*, where information about the node is monitored to determine the correctness of a message, and *data-centric detection mechanisms*, where the information received is analyzed either locally or through neighbor cooperation. Periodically-received status information in the form of Cooperative Awareness Messages (CAMs) as well as event-triggered Decentralized Environmental Notification Messages are verified on reception and stored locally on the vehicle's Local Dynamic Map (LDM). COLAW, like other services, can access this information to provide further functionality.

II. COLAW: The Protocol

Vehicles are periodically broadcasting important status information to the surrounding vehicles. These awareness messages are processed and stored by the receiving vehicles and help them create a detailed view of their surrounding environment. Being aware of neighboring vehicles and knowing where they are headed, is an important enabler for cooperative vehicular applications. Imagine the following use case from the logistics industry represented in Figure 1. Conventional fleet management software does not verify the location that is reported by the trucks, which degrades the reliability of the service that they provide.

Trucks (i.e. the requester) implementing COLAW can generate location proofs en route by sending out a *Location Proof Request (LPR)* asking witnessing vehicles to endorse its location claim. Witnesses that receive the LPR and are willing to help out (*accepting witnesses*) make an assessment about the truck's position based on the beacons that they have received and send back a *Location Proof Endorsement (LPE)* message which includes their digital signature. After receiving all LPE messages, the truck generates a *Location Proof (LP)* that includes the location claim and the collected endorsements. Finally, instead of submitting the raw location claim, the truck submits the generated LP to any service (i.e. verifier) requesting a location update, such as the fleet management software in our case. Table I and Table II provide an overview of the symbols and functions that will be used in the following four sections that discuss the different phases of COLAW in more detail.

A. Phase 1 - Proof Request

The *Proof Request* phase includes the steps that the requester executes whenever a location proof is requested and can be seen in Algorithm 1. The more endorsements the requester manages to collect the stronger the generated location proof

Algorithm 1: Phase 1 - Proof Request (Requester).

```
/* Triggered by the middleware */
Output: $LPR_R$ message
1 $LPR_R \leftarrow LPR(h(latest\_lc(id_R)))$;
2 $PL_R \leftarrow select\_partners()$;
3 foreach $id \in PL_R$ do
4     $hlc_{id} \leftarrow h(latest\_lc(id))$;
5     $\sigma_{id,R} \leftarrow sign(hlc_{id}, key_R^{-1})$;
6     $LPR_R.append(id, hlc_{id}, \sigma_{id,R})$;
7 end
8 $\sigma_{msg} \leftarrow sign(LPR_R, key_R^{-1})$;
9 $broadcast(LPR_R, \sigma_{msg})$;
```

Algorithm 2: Phase 2 - Proof Endorsement (Accepting Witnesses).

```
Input  : $LPR_R$ message
Output : $LPE_{W_i}$ message
1 $id_R \leftarrow LPR_R.id_R$;
2 $hlc_R \leftarrow LPR_R.hlc_R$;
3 if $verify(LPR_R.\sigma_{msg})$ AND $id_{W_i} \in LPR_R.id()$
    AND $verify(LPR_R.\sigma_{W_i,R})$ AND
    $endorse(id_R, hlc_R)$ then
4     $LPE_{W_i} \leftarrow LPE(h(LPR_R))$;
5     $\sigma_{R,W_i} \leftarrow sign(hlc_R, key_{W_i}^{-1})$;
6     $LPE_{W_i}.append(id_R, hlc_R, \sigma_{R,W_i})$;
7     foreach $id, hlc_{id} \in LPR_R.id(), LPR_R.hlc()$ do
8         if $endorse(id, hlc_{id})$ then
9             $\sigma_{id,W_i} \leftarrow sign(hlc_{id}, key_{W_i}^{-1})$;
10            $LPE_{W_i}.ENL.append(id, hlc_{id}, \sigma_{id,W_i})$;
11        end
12    end
13    $\sigma_{msg} \leftarrow sign(LPE_{W_i}, key_{W_i}^{-1})$;
14    $broadcast(LPE_{W_i}, \sigma_{msg})$;
15 else
16    $discard(LPR_R)$;
17 end
```

will be. As a result, some kind of incentivization mechanism must be installed for witnesses to take part at the location proof generation round. This incentive is provided by the requester. When sending out the LPR message, the requester endorses the locations of the witnesses, giving them the chance to generate a LP on their own. This procedure requires the requester to implement a certain selection mechanism that selects the most suitable witnesses (line 2). Facilities such as the LDM that store received beacons from surrounding vehicles can play a significant role in finding the witnesses that are most likely to respond. After the selection process, the requester has generated a partner list PL_R, that consists of the pseudonyms of the witnesses. For each selected witness, the requester extracts the latest awareness beacon that is stored, hashes it and signs the resulting hash with its private key before adding it to the LPR message (lines 3-7). As will be discussed in Section IV-B, hashes are used to protect against semi-honest nodes. Finally, the message is signed and broadcast (lines 8-9).

B. Phase 2 - Proof Endorsement

During the *Proof Endorsement* phase, witnesses have the chance to submit their endorsements by providing their digital signature and the steps to do so are shown in Algorithm 2. On reception of an LPR message, the witness verifies the message signature σ_{msg}. If successful, the witness decides whether to respond to the request or not. In the previous section, we mentioned that the requester is incentivizing the witnesses to endorse the transmitted request by providing a list of endorsements. As a result, the decision whether a witness will respond depends on whether the requester has attached an endorsement for the witness or not. In case the witness' pseudonymous id is listed in the LPR message, the witness is considered an *accepting witness* and continues by verifying the rest of the signatures attached to the message ($\sigma_{W_i,R}$). If all signature checks are successful, the witness starts comparing the hashed location claims that are included in the LPR message with the actual location claims that are stored in the witness' LDM.

By leveraging location verification mechanisms vehicles only store location claims in their LDM which are classified as plausible. This means that, if the LDM of the witness contains a location claim for a pseudonymous id whose hash correspond to the received hashed location claim the witness will endorse that id. This procedure is summarized in the $endorse(id_i, hlc_i))$ function which returns $true$ if successful and $false$ otherwise. The witness endorses the hashed location claim of the requester and if successful does the same for the rest of the witnesses. All the above mentioned checks are summarized in line 3 of Algorithm 2.

Having the witnesses endorse each other too, optimizes the protocol for all participants. Not only the requester can collect endorsements from the selected witnesses but every accepting witness is rewarded for participating in the location proof generation by receiving endorsements not only from the requester (*requester incentive*) but also from the other accepting witnesses (*witness incentive*). Each participant is thus able to construct its own location proof as we will see in the following section. For every verified hashed location claim the witness generates a signature and attaches it to the LPE message (lines 7-12). After generating all possible signatures, the LPE includes at most the same amount of signatures as the LPR message. However, this ideal scenario cannot always be achieved, as we will see in Section IV-C.

C. Phase 3 - Proof Composition

During the *Proof Composition* phase, requester and accepting witnesses independently generate their location proof. They listen for incoming LPE messages and on reception verify the message's signature σ_{msg}. If successful, the recipient

<table>
<tr><td colspan="3">$cert_R$ (125 bytes)</td></tr>
<tr><td colspan="3">σ_{msg} (64 bytes)</td></tr>
<tr><td colspan="3">Default Message Information $(1 + 4 + 24$ bytes)</td></tr>
<tr><td>Message Type</td><td>Pseudonymous id_R</td><td>Location Claim lc_R</td></tr>
<tr><td colspan="3">Endorsement List $(N * (4 + 64)$ bytes)</td></tr>
<tr><td>Pseudonymous id_{W_1}</td><td colspan="2">Signature $\sigma_{R,W_1}(hlc_R, key_{W_1}^{-1})$</td></tr>
<tr><td>$\vdots$</td><td colspan="2">$\vdots$</td></tr>
<tr><td>id_{W_N}</td><td colspan="2">$\sigma_{R,W_N}(hlc_R, key_{W_N}^{-1})$</td></tr>
</table>

Table III: Data structure of the requester's location proof with collected endorsements from N witnesses.

checks if the received LPE belongs to the correct location proof generation round by checking the hashed LPR which is attached. If yes, then the rest of the signatures are verified and if successful, the recipient extracts the signature that endorses its location claim.

Since the recipient is aware of the number of participants, the total number of expected signatures is known. In the event that not all signatures are collected, a timer will signal the end of the location proof generation round. With all the collected signatures, requester and accepting witnesses both compose their final LP, which can then be submitted to a verifier. Table III shows how the location proof could look like for the requester.

D. Verification

Each vehicle taking part in a location proof generation round finishes the round with a number of signatures that can be used to generate a location proof for that vehicle. Location-based services act as verifiers that receive the generated location proof and, once verified, provide their service to the end-user. Besides the compulsory checks such as signature verification, every verifier can conduct other optional checks to increase the plausibility of the location proof. Threshold-based checks rely on calculating a predefined metric value from certain properties of the location proof. In case the result of the calculated metric is greater than a certain threshold, the location proof can count as plausible. In case the threshold is not reached, the location proof might be discarded and the submitter might be asked to submit a *stronger* location proof, with a higher plausibility value. The number of collected signatures as well as the diversity of these signatures are promising candidates for these metrics. Although the number of available signatures to collect strongly depends on environment and road considerations as we will see in following sections, both the number and the diversity of the signatures can decrease the chance for colluding vehicles to generate a location proof that will be accepted by the verifier.

Another common approach to incentivize vehicles to behave honestly is to attach a trust value to each vehicle, which will be adjusted based on the number of successfully submitted location proofs. Vehicles with a low trust value might be deprived of the rights to using the service, motivating them to behave honestly. Furthermore, trace analysis of the submitted location proofs over time can also be used to detect anomalies. Although a substantial amount of information is necessary, crosschecking information from multiple sources is a common approach for detecting cheating attempts.

	MO	**CO** [bytes]	**PGT** [s]
Multi-Party [2]	$2N^2 + 6N$	$320N^2 - 192N$	$1.43N^2 + 6.74N$
VeriPlace* [3]	$8N - 8$	$1113N - 1113$	$17.18N - 17.18$
APPLAUS [4]	$0.5N^2 + 0.5N - 1$	$64N^2 - 56N - 8$	$0.08N^2 + 3.73N - 3.81$
Vouch+ [5]	$2N^2$	$169N^2 + 93N$	$2.23N^2 + 8.12N$
PROPS [6]	$4N^2 - 4N$	$4182N^2 - 4182N$	$2980N^2 - 2980N$
COLAW	N	$100N^2 + 126N$	$1.13N^2 + 4.17N$

Table IV: Message and Communication Overhead (MO, CO) and Proof Generation Time (PGT) for N witnesses (* : infrastructure-based). Equations are derived from the paper descriptions and step-by-step calculations can be found in our public repository [7].

Although the list is not exhaustive and more sophisticated detection techniques can be used, ultimately it comes down to the LBS itself to decide which plausibility checks will be implemented based on its business model.

III. Related Work

With the rise of LBSs in mobile devices such as smartphones and the increasing value that they provide, an increase in attempts to spoof the smartphone's location to exploit the LBS has been noticed. For this reason, intensive research has been conducted around methods to verify a device's location.

Dupin et al. [2] developed a location proof system which ensures location and identity privacy of both the requester and the witnesses. Their protocol is based on **Multi-Party** computations and *group-signatures* to guarantee that. By running a min/max computation protocol between requester and all accepting witnesses, the requester's coordinates are determined, which are signed by all witnesses using the group signature scheme. Although the protocol achieves a high degree of privacy, there are currently no vehicular standards for using group signatures which makes vehicular integration questionable.

Luo and Hengartner proposed **VeriPlace** [3], a location proof architecture where user information and location information is managed by distinct trusted third party entities. Users can acquire an intermediate and a final location proof by communicating with them. In order to detect possible cheaters, the users have to submit their location proofs frequently enough for the cheating authority to detect it. Although VeriPlace's key design components are user privacy and cheat detection it relies on three trusted third parties to achieve that.

Zhu and Cao designed **APPLAUS** [4], a privacy-preserving location proof updating system in which co-located devices can leverage their short-range Bluetooth connections to mutually generate location proofs that can be uploaded to a location server. Similar to our approach, they use periodically changing pseudonyms to ensure source location privacy of the devices. Nevertheless, the use of short-range Bluetooth connections

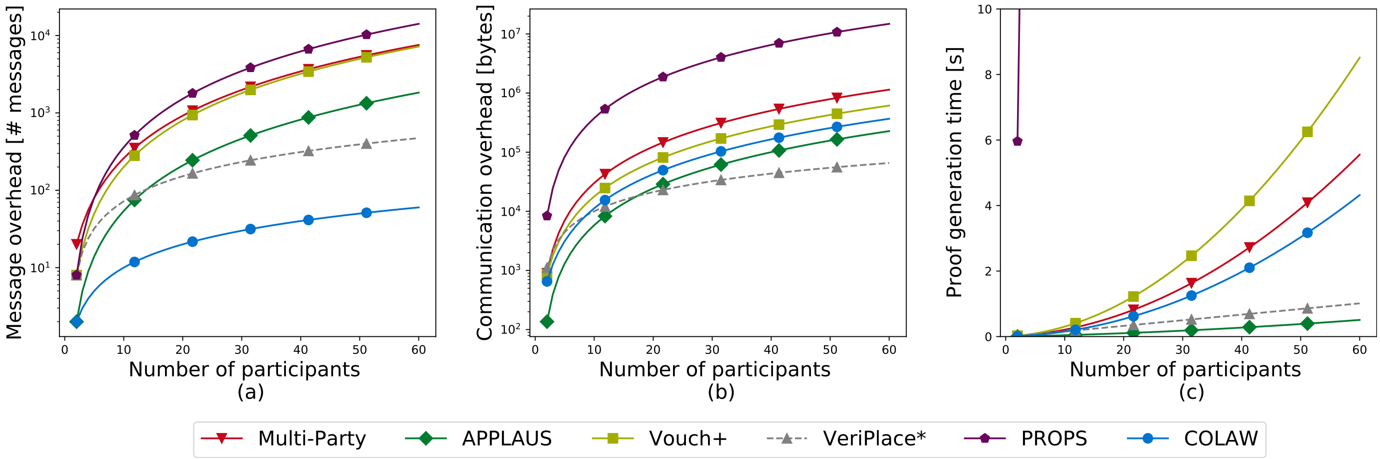

Figure 2: Plots comparing (a) Message Overhead, (b) Communication Overhead and (c) Proof Generation Time as a function of the number of participants. (* : infrastructure-based).

infers that the performance of the protocol will be decreased in vehicular environments especially during non-rush hours where vehicle density is low and vehicles might have to communicate over longer distances.

Boeira et al. introduced *Vouch* [8] and its successor, **Vouch+** [5], a secure proof-of-location scheme tailored for VANETs. Although Vouch relies on trusted infrastructure, the decentralized scheme of Vouch+ allows any road participant to act as a *Proof Provider*. Once the proof is generated and transmitted, a 2-hop trust chain between *Verifier*, *Prover* and *Proof Provider* is established. Location verification is based on a plausibility model that takes the high mobility of VANETs into consideration. Both protocols are tailored for use in vehicular environments but having each vehicle provide location proofs for one vehicle at a time increases the message and communication overhead significantly.

Gambs et al. propose **PROPS** [6], an infrastructure independent location proof system in which witnesses provide *Location Proof Shares* to provers. The location privacy of the witnesses is ensured through group signatures, the authenticity of the prover is achieved through zero-knowledge proofs and the distance between provers and witnesses is verified through a proximity testing procedure based on distance-bounding protocols. The use of zero-knowledge proofs guarantees a high level of location privacy and distance-bounding protocols can provide a solution for terrorist attacks but the high processing and communication costs are not acceptable in safety-critical vehicular environments.

The importance of location information has attracted the attention of numerous start-ups, which try to solve the problem of location verification by establishing a chain of trust based on blockchain technology. Through consensus mechanisms, blockchains establish a unanimous state of the network, making it seemingly impossible for participants to upload bogus information or modify existing data without the network's consent. XYO [9] and FOAM [10] have both experimented with different ways of submitting and retrieving location information in a verifiable way. Although they claim that their approaches are functional, besides their Whitepapers, no other information is publicly available, which is why they are not further discussed in this work.

In this section an overview was provided about current developments and research in location proof architectures. Although some of the approaches were specifically tailored for use in a vehicular environment, most of the proposed schemes still focus on mobile phone usage. Leveraging the distributed and broadcasting nature of VANETs can be used to generate location proofs in a more efficient and secure manner.

IV. Evaluation

A. Analytical Evaluation

We define the following three cost metrics that will be used to compare COLAW to the other presented approaches:

- **Message Overhead (MO)**: The number of messages that have to be exchanged between vehicles in order to generate a location proof.
- **Communication Overhead (CO)**: The number of bytes that have to be exchanged between vehicles in order to generate a location proof.
- **Proof Generation Time (PGT)**: The time it takes between sending out a location proof request until the requester retrieves all requested signatures.

Although many of the approaches use different communication technologies than COLAW, we assume that all protocols use the same wireless protocols and cryptographic algorithms to be able to compare them with each other. Maximum network load is generated when all N users in an area try to collect $N-1$ signatures from neighboring witnesses. Table IV compares COLAW to similar approaches and Figure 2 visualizes these

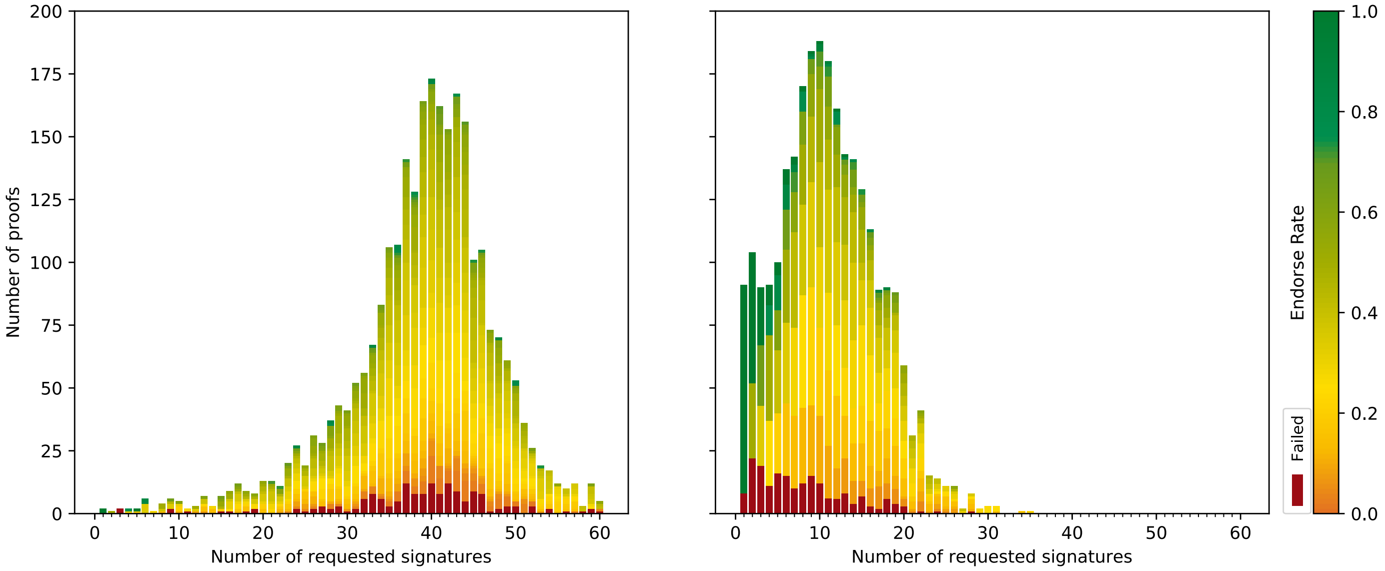

Figure 3: Number of proofs as a function of the number of requested signatures for the reference (left) and the urban scenario (right).

results. Taking full advantage of the distributed nature of vehicular communications and by incentivizing vehicles to endorse each other already in the *Proof Request* phase, the message overhead of generating N distinct location proofs can be reduced by at least an order of magnitude. Furthermore, only infrastructure-based approaches like VeriPlace manage to achieve a noticeably lower communication overhead with a large number of participants because the trusted infrastructure is not generating location proofs. Although trusted infrastructure can decrease network overhead, their availability cannot be guaranteed. Finally, looking at the calculated proof generation times we can see that COLAW performs averagely. At around 40 vehicles, COLAW requires about 2s to generate one location proof for each vehicle. This is mainly, due to the increased number of signature generations and verifications that are necessary for COLAW and need to be attached to every exchanged message according to the ETSI ITS standards. Besides the infrastructure-based VeriPlace solution, only APPLAUS, which was not designed for safety-critical environments and thus deviates from these requirements, manages to achieve a lower proof generation time. In highly dynamic vehicular environments, where inter-contact times between vehicles might be short, the increasing proof generation time has to be taken into consideration. In subsection IV-C it will be shown that by dynamically adjusting the maximum number of participants that can take part during one generation round or with more strict partner selection mechanisms COLAW's proof generation time can be both controlled and improved.

B. Security and Privacy

In order to evaluate COLAW's security we adopted the threat model introduced in [11], which distinguishes between *malicious adversaries* that arbitrarily deviate from the protocol's specification and *semi-honest adversaries* that try to infer additional information about the system and violate the participant's privacy. In the following, an overview of some of the most common attacks against location proof architectures and how they can be prevented, will be given. *False proof* and *Collusion* attacks are considered to be conducted by *malicious* adversaries, while *Location* and *Identity* privacy violation attacks are considered to be conducted by *semi-honest* adversaries.

False proofs: The requester might try to generate a fake proof for an arbitrary location. To do so, the requester would have to include the hash of a fake location claim into the LPR message. Accepting witnesses willing to endorse the requester will look in their LDM for a location claim which matches the one in the LPR message. In case the requester did not sent out the fake location claim, they will not find the entry and thus will not endorse the requester. Even if the requester did send out the fake location claim, the witnesses will not endorse the request because the location verification methods that each witness uses to evaluate the plausibility of received location claims will detect the bogus location claim before storing it in the LDM and will discard it. In any case the requester will not receive any endorsements and will not be able to generate a location proof.

Collusion attack: Collusion attacks, where two requesters collude with each other, are known as *terrorist fraud attacks* and are extremely difficult to prevent. This attack involves a requester A at position P_A, a requester B at position P_B and a witness W in the vicinity of B. A and B collude to generate a location proof for A and W at location P_B. Although A and B are colluding, B has no access to the private keys of A, which means that B acts as a proxy between A and W. For such a collusion to be successful, not only LPR and LPE messages would have to be forwarded but also all beacons that B generates will have to be signed by A first so that W generates LPE messages that endorse A. Although

Parameter	Value
Scenario	Bologna Pasubio
Simulation time	3600s
Radio Communication	IEEE 802.11p @ $5.89\,\text{GHz}$
Transmission power	20mW
Path loss model	Free space ($\alpha = 2.0$)
Obstacle shadowing	$\beta = 9dB, \gamma = 0.4dB/m$

Table V: Simulation parameters for Veins.

extremely difficult, the attack is still possible. The only direct countermeasure against this attack is the assumption that the underlying location verification mechanism on *W* will be able to detect such kind of attack and not insert the bogus beacons in the vehicle's LDM. Distance-bounding protocols are known to prevent terrorist attacks, which means that in case this kind of attack is considered highly probable, the protocols must be implemented on vehicular level to prevent bogus location claims to be inserted into the LDM.

Location privacy: Vehicles might try to infer additional information about the location of surrounding vehicles. By including only hashed location claims in the messages exchanged and because it is practically infeasible to recreate the original message when only the hash is available, it is also practically impossible for any vehicle that has not received the location claim and has stored it in its LDM to figure out the exact location of the vehicle listed in any of the messages. Even if the location claim is available in the local LDM, COLAW does not sacrifice any additional location privacy than is already sacrificed by the underlying beaconing service.

Identity privacy: Road participants might try to infer additional information about the identity of the surrounding vehicles. This is possible to a certain degree for witnesses and eavesdroppers. Due to imperfect witness selection from the requester, witnesses, and eavesdroppers might find out that a vehicle that they cannot *see* is in the requester's vicinity. Nevertheless, because of periodically changing pseudonyms no more information about the real identity of the vehicle can be inferred. The described attack cannot be conducted by the requester himself because witnesses will only include information about surrounding vehicles in their LPE message that were already endorsed by the requester in the LPR message. In any case, we argue that this partial loss of identity privacy can be tolerated and could even be minimized by introducing more advanced witness selection mechanisms.

C. Experimental Evaluation

We have implemented COLAW as an ITS-G5 service running on top of the Artery middleware [12] to be compliant with the European communication standards. Artery runs on top of Veins [13], a popular open source framework for running vehicular network simulations. The code of our simulation is publicly available at [7] for review and further research. We generated a random road traffic scenario with the parameters seen in Table V based on the small-scale scenario of the city of Bologna [14]. Each vehicle started a new location proof generation round every one minute and the maximum number of partners that a requester could endorse was limited to 60. Each vehicle was broadcasting an awareness beacon every 100 ms and received beacons were stored in the LDM for 1.1s.

Scenario	APS	AER	TSR
Reference	11.82	30.72%	93.93%
Without Partner Selection	11.31	29.37%	93.11%
Urban	3.42	36.15%	92.7%
No Rush-Hour	3.36	38.91%	94.85%

Table VI: Average proof size (APS), average endorse rate (AER) and total success rate (TSR) for the discussed scenarios.

To evaluate our protocol under different scenarios we define the following metrics:

- **Average Proof Size (APS)**: The average number of signatures collected by the requesters during one run of COLAW.
- **Average Endorse Rate (AER)**: The average number of signatures that are collected by a requester compared to the number of signatures that were requested.
- **Total Success Rate (TSR)**: The percentage of successful attempts to generate a location proof. Proofs count as successful once there is at least one signature collected by the requester.

We wanted to examine how certain road conditions like the vehicle's density will affect our protocol's performance. During rush-hours (6-8 am and 4-6 pm), where the vehicle density is higher, an increased number of location proofs is expected to be generated. We modeled this parameter by adjusting the vehicle insertion rate of our simulation. During rush-hours the insertion rate was set to 0.25 Hz and during no rush-hours to 0.08 Hz.

Furthermore, we wanted to examine how environmental parameters can affect the communication between the participants and as a result COLAW's performance. For example, in urban cities the existence of buildings causes distortions that weaken or even block the vehicle's radio transmissions. We expected that in urban scenarios vehicles will have a harder time detecting and endorsing each other. To examine this scenario we added large buildings on every street corner of the simulated city.

Finally, we wanted to examine how different approaches for the partner selection ($select\_partners()$ in Algorithm 1) might improve the overall protocol performance. Received CAM messages from neighboring vehicles are stored in the LDM together with an expiry timestamp. By introducing an expiry timestamp threshold for the neighbor selection, vehicles can select their neighbors based on the most recent encounter.

Reference Scenario: The parameters of our reference scenario include a rural driving environment, rush-hour vehicle density and threshold-based partner selection. We used this scenario as a reference to conduct our parameter analysis.

Urban Scenario: As expected, the number of requested and collected signatures significantly decreased once the buildings were introduced. Figure 3 shows that in rural environments 40 signatures were requested on average while no more than 20 signatures were requested in the urban scenario. Nevertheless, Table VI shows that the total success rate (indicated in Figure 3 by the number of *failed proofs*) in both scenarios did not change significantly. On the other hand, the average endorse rate (indicated in Figure 3 by the *color bar*) was slightly increased resulting in a *greener* figure. From the average number of requested signatures and the calculated average endorse rate we got the average proof size which, as expected, decreased significantly from 11.82 to 3.42.

No Rush-Hour Scenario: The behavior that was noticed when examining the No Rush-Hour scenario was similar to the urban scenario. While the total success rate stayed relatively similar, the average proof size dropped significantly and the average endorse rate increased. This behavior was expected since during no rush-hours vehicle encounters are scarce.

Without Partner Selection: Although the trivial threshold-based partner selection does not take the movement and distance of neighbors into consideration, removing it resulted in a slight decrease in the average proof size from 11.82 to 11.31. This is the case because of the relatively high beaconing frequency (10 Hz) compared to the low proof generation frequency (1/min). When a vehicle is accessing its LDM, the nearest neighbors are already stored and out-of-range neighbors have already been discarded. The introduction of a threshold-based partner selection filters out only the partners that move out of the requester's range but have not yet been discarded from the LDM.

During our experimental evaluation we conducted a first parameter analysis of the potential parameters that could affect COLAW's performance. In all scenarios the total success rate of COLAW stayed above 92% proving that vehicles were highly successful in generating location proofs. The average endorse rate is between 30-40%, indicating that more signatures were requested than were actually collected. This is mainly due to the basic partner selection mechanism that was used during the simulations and is expected to be improved with more sophisticated mechanisms in the future.

V. Conclusion

In this work we presented COLAW, a cooperative location proof architecture based on witnessing, that fully leverages the distributed nature of vehicular ad hoc networks. We motivated the usage and implementation of such a protocol in current and future vehicular applications and used a realistic use case from the logistics industry to demonstrate its necessity. We evaluated COLAW based on a number of cost and performance metrics and extensively discussed the security and privacy aspects of our protocol. Compared to existing location proof architectures, COLAW causes the least message overhead and only few solutions from the world of smartphones achieve a lower communication overhead and proof generation time, which are not applicable in the vehicular environment. Furthermore, we proved that COLAW's success rate is not affected by road conditions and environmental parameters. In future version of COLAW more sophisticated selection mechanisms are expected to further improve the average endorse rate and as a result the overall performance of our protocol.

We believe that the integration of COLAW in future deployment phases of C-ITS can increase the reliability and quality of service of all services that rely on location information and can be further extended to verify other information as well. The safety-critical and highly dynamic vehicular environments rely on the trustworthiness of such information and verification architectures like COLAW must be in place to pave the way for an accident-free road transport with optimal traffic flow, which is the ultimate objective of cooperative and automated driving.

Acknowledgement

With the support of the Technische Universität München – Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement n° 291763.

References

[1] European Global Navigation Satellite Systems Agency, "GNSS market report," https://www.gsa.europa.eu/market/market-report, 2017.

[2] A. Dupin, J.-M. Robert, and C. Bidan, "Location-proof system based on secure multi-party computations," in *Provable Security*. Springer International Publishing, 2018, pp. 22–39.

[3] W. Luo and U. Hengartner, "Veriplace: A privacy-aware location proof architecture," in *GIS '10 Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Jan. 2010, pp. 23–32.

[4] Z. Zhu and G. Cao, "APPLAUS: A privacy-preserving location proof updating system for location-based services," in *2011 Proceedings IEEE INFOCOM*, Apr. 2011, pp. 1889–1897.

[5] F. Boeira, M. Asplund, and M. Barcellos, "Decentralized proof of location in vehicular ad hoc networks," *Computer Communications*, Aug. 2019.

[6] S. Gambs, M. Traoré, M. Roy, and M.-O. Killijian, "PROPS: A privacy-preserving location proof system," *Proceedings of the IEEE Symposium on Reliable Distributed Systems*, vol. 2014, Oct. 2014.

[7] TUM-ESI, "COLAW Simulation Code and Supplemental Material," GitHub Repository, 2020, https://github.com/tum-esi/COLAW.

[8] F. Boeira, M. Asplund, and M. Barcellos, "Vouch: A secure proof-of-location scheme for VANETs," in *MSWIM '18 Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Oct. 2018, pp. 241–248.

[9] S. Scheper, M. Levin, and A. Trouw, "The XY oracle network: The proof-of-origin based cryptographic location network," https://docs.xyo.network/XYO-White-Paper.pdf, Jan. 2018, accessed: 2019-12.

[10] Foamspace Corp, "FOAM whitepaper," https://foam.space/publicAssets/FOAM_Whitepaper.pdf, Jan. 2018, accessed: 2019-12.

[11] C. Hazay and Y. Lindell, "Efficient secure two-party protocols," *Information Security and Cryptography*, 2010.

[12] R. Riebl, H. Günther, C. Facchi, and L. Wolf, "Artery: Extending veins for VANET applications," in *International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, Jun. 2015.

[13] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, p. 3–15, 2011.

[14] L. Bieker, D. Krajzewicz, A. P. Morra, C. Michelacci, and F. Cartolano, "Traffic simulation for all: a real world traffic scenario from the city of bologna," in *SUMO 2014*, May 2014. [Online]. Available: https://elib.dlr.de/89354/