LIDAR: Lidar Information based Dynamic V2V Authentication for Roadside Infrastructure-less Vehicular Networks

Kiho Lim
Department of Computer Science
University of South Dakota
Vermillion SD, USA
kiho.lim@usd.edu

Kastuv M. Tuladhar
Department of Computer Science
University of South Dakota
Vermillion SD, USA
kastuv.tuladhar@coyotes.usd.edu

***Abstract*—Advancement of autonomous driving and vehicular ad-hoc networks (VANETs) have demanded many applications requiring vehicle-to-vehicle (V2V) communications as vehicles cooperatively share their traffic information (collected by sensors) with each other to improve driving safety, traffic efficiency and convenience. In order to secure V2V communications, authentication has been carried out in the presence of the central trusted authority and infrastructures. However, the scenarios where the infrastructures are not available have not been addressed well. In this paper, we propose LIDAR, a lidar information based authentication mechanism for V2V communication, to authenticate vehicles locally without involvement of a trusted authority and infrastructures. Our protocol utilizes the sensors installed in vehicles to verify the shared surrounding objects. The proposed scheme is robust against possible security threats such as man-in-the-middle attack. An extensive simulation is also conducted in an autonomous driving environment to evaluate our scheme.**

Key words: V2V, VANET, authentication, sensor, Lidar, infrastructure-less.

I. Introduction

In recent years, vehicular technology has taken its shape as smart vehicle systems and pilot assisted self-driving vehicles. Various technology companies, automotive manufacturers & supplier and academic organizations have been conducting the research to develop secure, robust and responsive transportation systems. To improve the traffic safety and enable the new services for the Intelligent Transportation Systems (ITS), Vehicle-to-Vehicle (V2V) communications have drawn the significant attention [1]–[3]. Through V2V communications, vehicles can disseminate emergencies/alerts messages, and exchange the information related to weather, navigations and other value added services. Although V2V is intended to provide and improve the driving safety, it also opens up the opportunity for attackers to launch various attacks depends on their intention i.e. either benefit themselves or maliciously harm the victims. For example, attackers can transmit the false information to find the best routes by diverting the neighboring cars or pretend to be another or multiple vehicles by using false identities [4]. Further, the safety of autonomous cars is even more crucial as it requires almost no human inputs and it heavily relies on sensor information and on-board computing systems. Attackers, in this case, can attempt to forge the sensor information of vehicles to circumvent the accident liability or corrupt the data that can lead towards the property damage or even life threat [5]. Thus, securing the inter-vehicular communication is one of the highest priorities in vehicular networks.

Prior to any communication between vehicles, both parties should be able to authenticate each other properly. The IEEE Standard 1609.2 for Wireless Access in Vehicular Environments (WAVE) [6] and Dedicated Short-Range Communications (DSRC) [7] assume that Public Key Infrastructure (PKI) is used to authenticate vehicles with certificated issued by a trusted third party such as a Trusted Authority (TA) or a Certificate Authority (CA). Thus, the TA or CA has to be involved in the process of certificate verification to authenticate vehicles [8]. However, due to the lengthy certificates, it can lead to undesirable communication and computation overheads [9]. Also, the centralized trust management system for authentication is not a fault-tolerant approach so it demands an alternative or supplementary approach for V2V authentication. In addition, it requires infrastructures to be connected with the TA. Hence, the authentication issue has become a big question when infrastructures are not readily available and such issues has not been addressed well.

In efforts to improve the road safety and driving efficiency, modern vehicles have advanced driver-assistance systems (ADAS) [10], that senses the driving environment and warn the drivers if any immediate threats are found to minimize the human error. The advancement in the sensing technology and sensor fusion is leading the vehicles towards connected vehicles (CV) and fully autonomous vehicles. Typically, the automotive sensors such as ultrasonic sensors, millimeter wave radars, lidar, and cameras [11] are used for modern automated vehicles and autonomous driving. Lidar (Light detection and ranging) sensors play an important role in the ADAS as it is useful in distance ranging, obstacle detection/avoidance, and path planning. Also, the information gathered by lidar or sensor fusion information can also be used for additional features.

To address the aforementioned issues, we leverage the lidar information for V2V authentication without the involvement of a third party trusted authority. Our proposed scheme utilizes the existing sensors on vehicles without additional hardware cost and the locality information of vehicles are identified by

978-1-5386-5553-5/19/$31.00 ©2019 IEEE

comparing the *similarity* of the *locality information* of the surroundings generated by sensor data. As the authentication process can be performed locally, our proposed scheme does not require the use of PKI certificates and help of infrastructure, hence it would not face its drawbacks.

The rest of the paper is organized as follows. Section II provides the background of our work and section III describes our system model and assumptions. Section IV presents the LIDAR protocol in detail, and our implementation and analysis are presented in Section V. In Section VI, we present further discussion and our future work. Finally, we conclude in Section VII.

II. Related Work

Various authentication schemes have been studied in the literature to provide authentication in vehicular networks. Raya and Hubaux [4] has proposed a PKI based anonymous certificate authentication scheme to preserve the user privacy. The PKI based certificates are quite lengthy and the TA can be overwhelmed by managing them. Also, privacy is still a concern under this approach because it deals with true identities of vehicles. Huang et al. [12] proposed pseudonyms certificate to authenticate vehicles using identity-based encryption (IBE). Similarly, in [13], IBE with pseudonyms using proxy mobile IPv6 for better mobility, is proposed to authenticate vehicles. Both approaches solved the security issues to some extent but they require infrastructures and authority to maintain the database for pseudonyms identity and true identity.

Yanbing et al. [14] proposed the dual authentication scheme where a vehicle generates an anonymous temporary encryption key and the TA verifies the anonymous key. The vehicle initiates the authentication in the first phase but it still relies on the centralized authentication afterward. To authenticate the vehicle in a decentralized group, Shao et al. [15] proposed an anonymous authentication scheme that uses the group signature scheme. However, the above-mentioned schemes need the infrastructures to exchange the key. To authenticate the vehicle beyond the range of the infrastructure, Lim et al. [16] proposed the trajectory-based pre-key exchange scheme that utilizes the trajectory information of vehicles and authenticate before communicating another infrastructure, however, the range of authentication for such vehicles is very limited. All of the above-mentioned approaches require the infrastructure at some point and require a third party authority such as the TA to centrally manage and maintain the certificates. Thus, an efficient decentralized authentication mechanism is necessary until the supportive infrastructure is fully deployed in a VANET architecture or such mechanism can be used as a supplementary in remote/rural areas where the infrastructure is not available.

With the emergence of autonomous vehicles, various sensors have been installed in the modern vehicles. Many researchers studied to utilize sensor information to assist the driving safety and efficiency. In [17], sensor data with camera images was proposed for object detection, recognition, and mapping. Similarly, in [18], the sensor data has been utilized to improve the visibility and the situational awareness of vehicles. In this paper, we leverage the available on-board sensors and sensor fused information to authenticate vehicles in an ad-hoc manner without requiring additional hardware.

III. System Model

In this section, we describe our system model and the assumptions for the proposed protocol and the attacks of interest. In our approach, the locality information of vehicle is used for authentication. In order to generate the locality information, the following sensors are used:

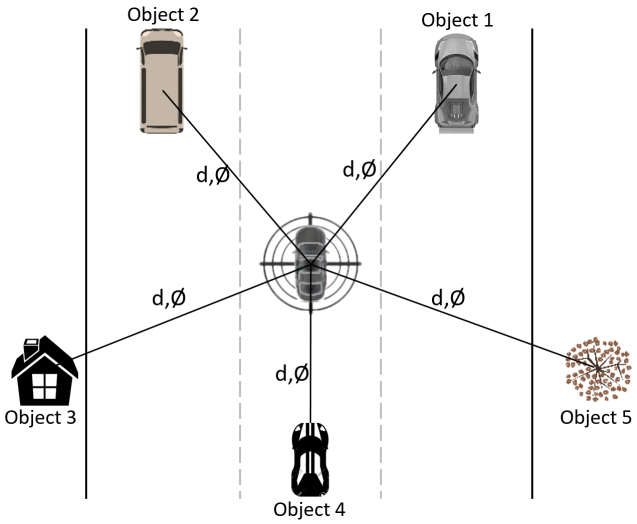


Figure 1: Overview of the System Model

- **Lidar sensor:** Lidar [19] senses an object through the active perception by casting a ray directed towards it. Each point are computed by the laser added for each channel which is distributed to maintain the desired field of view (FOV). A rotating lidar implemented with ray-casting is mounted at the top of the vehicle and it generates the frame of the casted points around its surrounding. The frame of the surrounding generated by the lidar is in a raw form and it is processed by On-Board-Unit (OBU) of the vehicle. The processed data can find the object around the vehicle with the distance & angle, and the object type can be determined using the camera sensor if needed.

In addition, the location provided by the GPS can be fabricated by the attacker but the relative distance of the vehicle can be calculated with the lidar data, thus the location spoofing attacks can be prevented.

- **Camera sensor:** The object detected by the lidar itself is hard to provide the comprehensive information about the object type, thus, the cameras installed in the modern vehicles are utilized to identify the object type. A fused data of the lidar and cameras can provide the holistic view of the surrounding, object types, its distance and angles [20].

The distance and the angle between two vehicles or vehicle/object can be calculated based on the center point of the object/vehicle. Due to the characteristics of the sensors, depending upon the sensor direction, the sensor date might not create the 360° view of the object. So we assume that Machine Learning (ML) algorithms such as YOLOv3 [21] are used to identify the object type and shape. The center point of an object is approximated from the object type and shape, and the vehicles can be authenticated within a threshold error boundary.

The overview of the system model is described in Figure 1. The vehicle equipped with a lidar scans the periphery areas and detects the surrounding objects near the vehicle. Also, object type determined using the sensor data and the center point is calculated. The vehicle then computes the distance (d) and angle (ϕ) towards the surrounding objects and can exchange this information over the DSRC wireless channel [7] to authenticate with other vehicles.

In this paper, we are interested in the attacks which are co-located with the legitimate entities i.e. the neighboring vehicles traveling along the road. We mainly focus on attackers impersonating legitimate entities by launching man-in-the-middle (MITM) attacks.

To summarize, our main contributions are highlighted as follows:

- A decentralized trust management system that can authenticate without the presence of central trusted third party such as the TA.
- An authentication mechanism that can perform in an infrastructure-less VANET environment.
- Utilization of available hardware i.e. lidar, cameras and the on-board sensor units.
- A secure and robust protocol against the possible attacks.

IV. LIDAR PROTOCOL

Phase 1: Initialization		
1.	A $\rightarrow$ All	Broadcasts $BEACON_A$ message $BEACON_A = ID_A, Loc, Ts$
2.	B	Checks against "paired" list; Aborts if found.
3.	B $\rightarrow$ A	requests for pairing
4.	A	Checks against "paired" list; Aborts if found.
Phase 2: Lidar Map Generation		
5.	A, B	$MapGen(Ts)$ Obtain map_A and map_B, respectively.
Phase 3: Key Agreement		
6.	A $\rightarrow$ B	$C_A = H(g^a \| \| ID_A \| \| map_A \| \| Ts)$
7.	B $\rightarrow$ A	$C_B = H(g^b \| \| ID_B \| \| map_B \| \| Ts)$
8.	A $\rightarrow$ B	$D_A = g^a \| \| ID_A \| \| map_A \| \| Ts$
9.	B	if $C_A! = H(D_A)$ then aborts.
10.	B	$Obj\_Verification(map_B, map_A, ID_A, Loc_A)$ If verification returns FALSE, then aborts.
11.	B $\rightarrow$ A	$D_B = g^b \| \| ID_B \| \| map_B \| \| Ts$
12.	A	if $C_B! = H(D_B)$ then aborts.
13.	A	$Obj\_Verification(map_A, map_B, ID_B, Loc_B)$ If verification returns FALSE, then aborts.
14.	A	Nonce $n_A \xleftarrow{R} \{1,0\}^\eta$
15.	A $\rightarrow$ B	$n_A \| \| HMAC(K', n_A)$
16.	B	Nonce $n_B \xleftarrow{R} \{1,0\}^\eta$
17.	B $\rightarrow$ A	$n_B \| \| HMAC(K, n_A \| \| n_B)$
18.	A	$HMAC(K, n_A \| \| n_B) \stackrel{?}{=} HMAC(K', n_A \| \| n_B)$
19.	A $\rightarrow$ B	$HMAC(K', n_B)$
20.	B	$HMAC_(K', n_B) \stackrel{?}{=} HMAC(K, n_B)$ If fails, then aborts.

Figure 2: LIDAR Protocol

The proposed protocol consists of three phases - 1) initialization; 2) lidar map generation; 3) key agreement. The LIDAR protocol is described in Figure 2. We discuss the details of our protocol as follows:

A. Initialization

As a vehicle starts broadcasting a periodic beacon message to attempt V2V communication, the LIDAR protocol is initiated. The beacon message contains vehicle ID, its GPS location, and a time stamp. Once the beacon message $BEACON_A$ from V_A is received by V_B, V_B first check its paired list to find if it has previously paired. If it's not found, then V_B sends a request for pairing to V_A. Similarly, V_A also checks its paired list and proceed to next phase if V_B is not found in its paired list.

B. Lidar Map Generation

After the initialization phase between V_A and V_B, both vehicles generate their own lidar maps using their sensor data. Note that it is assumed that their clocks are synchronized. The generated lidar map includes all surrounding objects detected through lidar data associated with camera images, with distance d and angle ϕ to each object. Note that the generated lidar map is used to compare with sensor data sent from another vehicles who wish to be authenticated.

C. Key Agreement

The key establishment process is based on the Diffie-Hellman key agreement with lidar information. In the key agreement phase, V_A and V_B exchange their commitments C_A and C_B, which bind the elements of the Diffie-Hellman key agreement $\{g^a, g^b, p\}$ with the lidar map map_A and map_B, respectively, and then disclose their committed information by sending their decommitments D_A and D_B. Upon receiving the decommitments, each vehicle computes the hash of decommitement to perform verifying the commitments. If the hash of the decommitment does not match with the commitment, then the protocol is aborted.

As a part of the verification process, each car confirms the presence of the target vehicle and its surrounding objects. The pseudocode for the object verification process is provided in Algorithm 1. For this process, the first step is to verify the target vehicle. It is obvious that if the target vehicle cannot be verified, it is not necessary to confirm the surrounding objects for efficiency. V_B first computes distance d_{BA} between V_B and V_A, angle ϕ_{BA} from V_B to V_A, and the object type $type$ of V_A using its own sensor data map_B, and compares them with the sensor data sent from V_A. Note that map'_A is generated by V_B for the comparison so that the both map_A and map'_A have the same base of V_A. For example, map'_A is generated from map_B by adding the distance (x, y) from V_B to V_A.

To verify the target vehicle, V_B checks if the distance d_{AB} and the angle ϕ_{AB} claimed by V_A are within the error threshold boundary (ϵ_d and ϵ_ϕ). It is also worth noting that π angle is added in the line 7 to compare the relative angles between V_A and V_B. For simplicity, we assume that the angle starts with zero at East direction and increases in a counterclockwise direction, so 180° needs to be added to get the view of V_A from V_B. Subsequently, the object type of

Algorithm 1 Pesudocode of Object Verification

1: **procedure** OBJ_VERFICATION($map_B, map_A, ID_A, Loc_A$)
2:
3: ▷ Target vehicle verification Process
4: $d_{BA} \leftarrow computeDistance(map_B, ID_A, Loc_A)$
5: $\phi_{BA} \leftarrow computeAngle(map_B, ID_A, Loc_A)$
6: $type_A \leftarrow computeObjectType(map_B, ID_A, Loc_A)$
7: **if** $((d_{BA} - \epsilon_d \leq d_{AB} \leq d_{BA} + \epsilon_d)$ & & $(\phi_{BA} - \epsilon_\phi \leq \phi_{AB} + \pi \leq \phi_{BA} + \epsilon_\phi)$ & & $(type_A = \text{"}car\text{"}))$ **then return** $continue$
8: **else**
9: **return** $FALSE$
10: **end if**
11:
12: ▷ Compute ObjectList A'_{Li} from local sensor data
13: $A'_{Li} \leftarrow computeObjectList(map_B)$
14: ▷ Compute ObjectList A_{Li} from map_A
15: $A_{Li} \leftarrow computerObjectList(map_A)$
16:
17: ▷ Compare A'_{Li} with A_{Li}
18: **for** each $L'_i \in L$, $r\phi \leq L_i.\phi' \leq l\phi$ **do**
19: **if** $((L_{i\cdot d} - \epsilon_d \leq L_i.'_d \leq L_{i\cdot d} + \epsilon_d)$ & & $(L_{i\cdot\phi} - \epsilon_\phi \leq L_i.\phi' \leq L_{i\cdot\phi} + \epsilon_\phi)$ & & $(L_i.'_{type} = L_{i\cdot type}))$ **then**
20: $continue$
21: **else**
22: **return** $FALSE$
23: **end if**
24: **end for**
25:
26: ▷ Successfully verified
27:
28: **return** $TRUE$
29: **end procedure**

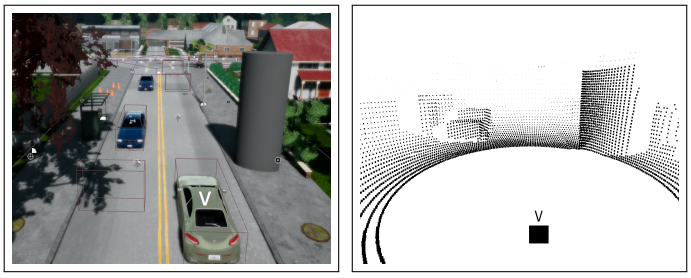
V_A is checked and then it is verified as a target vehicle. The process of target vehicle verification has been illustrated from line 3 to line 10. Now, V_B continues to verify V_A's surrounding objects. V_B leverages its own sensor data map_B generated from $MapGen(Ts)$ to compute the object list A'_{Li}, which includes all surrounding objects of V_A detected by V_B, with distance d, angle ϕ, and object type $type$. Similarly, another object list A_{Li} of V_A is also computed from the sensor data map_A sent from V_A. Computing two object lists from two different sensor data sources is illustrated from line 12 to 15. Note that now V_B has A_{Li} and A'_{Li} which are computed considering the relative distance and angle between the two vehicles, and the objects are sorted in the order of distance from V_A and the size of object.

Finally, the V_B compares its own object list A'_{Li} associated with its distance ($L_i.'_d$), angle ($L_i.'_\phi$) & object type $L_i.'_{type}$, with the acquired distance ($L_{i\cdot d}$), the angle ($L_{i\cdot\phi}$), and the object type $L_{i\cdot type}$ from V_A. For each object in A'_{Li}, if the distance and the angle from V_A are within the error boundary (ϵ_d and ϵ_ϕ), and the object type is matched, then the surrounding objects of the target vehicle are successfully verified. The process of object comparison is illustrated from line 17 to 24, and this process is repeated by V_A to verify each other. Note that some objects might not be detected due to obstacles or inclement weather, so in this case, a similarity score can be used. This will be further discussed in Section VI.

After the object verification process succeeds, the both vehicles now can compute a shared symmetric key and the generated keys need to be confirmed by sending nonces. V_A randomly generates η bit nonce n_A and sends the nonce n_A with its keyed-hash message authentication code (HMAC) to V_B. Here the HMAC is computed over the nonce n_A using the symmetric key $K' = (g^b)^a$ by V_B. upon receiving, V_B verifies the HMAC using its symmetric key $K = (g^a)^b$ and sends V_A another η bit nonce n_B along with a HMAC computed over n_A and n_B using the symmetric key K. If the HMAC received by V_B is verified, then V_A sends the last HMAC to V_B. Once the last HMAC is successfully verified, the shared symmetric key between V_A and V_B are confirmed and it can be used for their further communication.

V. IMPLEMENTATION & EVALUATION

In this section, we present the implementation of the LIDAR protocol and evaluate the proposed protocol.



(a) Target vehicle (b) Front view of lidar image

Figure 3: Simulation Environment

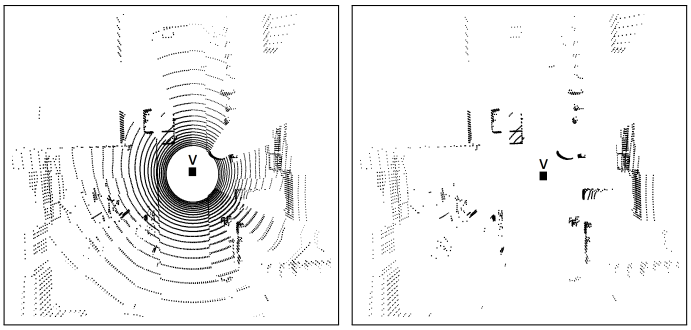
To evaluate our proposed protocol, we have simulated the LIDAR in auto-pilot mode using CARLA, an autonomous urban driving simulator [22]. The objects and environment settings are configured using Unreal Engine 4.18 [23]. For lidar sensor, we used the ray casting rotating lidar sensor supported by the CARLA, which performs similar to Velodyne models HDL-32E or VLP-16 [22]. The lidar generates the data in the point cloud format which can be viewed using a mesh processing tool such as meshlab [24]. The lidar setup and the simulation parameters considered are given in the Table I.

The simulation environment is shown in Figure 3a & 3b. Figure 3a shows CARLA simulation that includes the surrounding objects near the vehicles such as pole, other vehicles, bus-stop, houses, etc. The lidar sensor is placed on top of the vehicle V where the surrounding objects are visible and the front view of the lidar sensor is shown in Figure 3b.

Table I: Lidar sensor setup and simulation parameters

Parameters	Value
Number of channels	32
Range	100 m
Points per seconds	100000
Rotation Frequency	10
FOV Limit	40°
Vehicle running mode	$auto-pilot$

Lidar transmits the pulses of laser light and measures the object on the basis of time elapsed by the reflected pulse waves. As it scans around its periphery, data processing is required to filter out unnecessary data. The top view of the raw data captured by the lidar is shown in Figure 4a. In order to identify

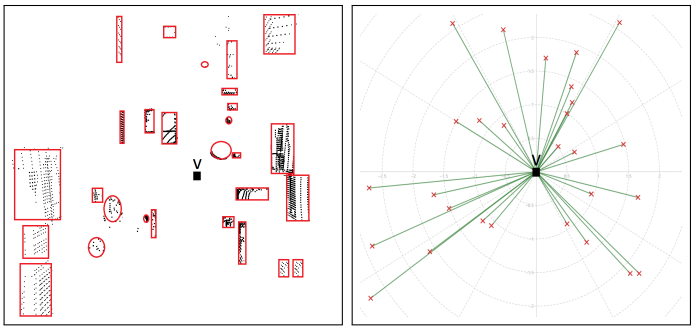


(a) Raw Data (b) Processed Data

Figure 4: Data Processing

the surrounding objects, unnecessary points are filtered out. The point cloud after the raw data is processed are shown in Figure 4b, which now the surrounding objects can be easily identifiable.

The processed data have the information of the surrounding object but it cannot characterize the object type without a proper object recognition mechanism. To find out the object type in the LIDAR, we project the 3D frame into 2D space in an XY plane where the location of the object is approximated as a center point of the object. To classify the object types, a ML algorithm is used and its result is shown in Figure 5a. The center point of the object can be computed after the object type is identified. The vehicle can then calculate the locality information after acquiring the central point of the surrounding objects. The grid view of the vehicle with the central point of the objects is shown in the Figure 5b.



(a) Object Detection (b) Grid View Lidar

Figure 5: Object Identification

Our LIDAR protocol authenticates the vehicle on the basis of the locality information. The locality information contains the surrounding object type and its distance & angle associated with the vehicle. We compute the distance and the angle by leveraging the center point of the object corresponding to the vehicle. For example, if the vehicle is in (x_1, y_1) and the center of the object is in (x_2, y_2) positions, then the distance and angle can be calculated.

The locality information is sorted with the least distance, since the lidar can identify the nearest object with higher accuracy. In addition, the size of the objects is also considered to increase the accuracy. The vehicle needs to validate the information of the target vehicle and if it validates, it can share the information of the surrounding object to authenticate. Thus, the message format of the locality information contains object number, its distance, angle and the object type are placed denoted as $No.$, dis, ang and $type$ in Figure 6.

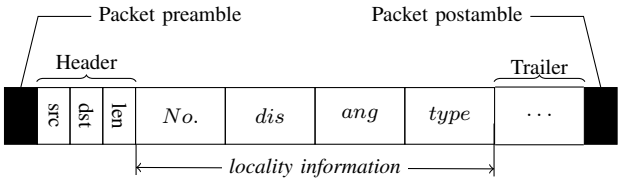


Figure 6: Message format of locality information

A. Evaluation

To evaluate the aforementioned locality information – distance, angle and the object type, we find the center point of the object using the object recognition and computed the distance. Here, we also consider the size of the objects to filter out too small objects, which may not be detected. After sorting it with the least distance, we also evaluate the angles. In this observation, we considered the 10 nearby objects of the vehicle. Similarly, the distance and the angles to the same objects were also calculated from the target vehicle. To compare the locality information, the relative distance and angle of the objects from the target vehicle are also calculated. The relative distance is calculated as such the co-ordinate of the vehicle is subtracted from the co-ordinate of the target vehicle. Similarly, the relative angle is computed using the relative location.

The computation errors of the distance (ϵ_d) and angle (ϵ_ϕ) are also calculated, and the results are shown in Table II, with the statistical parameters: maximum (Max_{ϵ_d}, Max_{ϵ_ϕ}), minimum error (Min_{ϵ_d} & Min_{ϵ_ϕ}), average error (μ_{ϵ_d} & μ_{ϵ_ϕ}), standard deviation (σ_{ϵ_d} & σ_{ϵ_ϕ}), and the standard error (ψ_{ϵ_d} & ψ_{ϵ_ϕ}). In addition, the LIDAR protocol has generated small error values of (ϵ_d) & (ϵ_ϕ) as expected because the center point of object is approximated. Thus, the threshold of the error can be determined using this approach as an appropriate error boundary.

Table II: Computation error of distance (ϵ_d) and angle (ϵ_ϕ)

	MAX	MIN	AVG	STD DEV	STD ERROR
$\epsilon_d(m)$	1.32	0.01	0.57	0.5	0.19
$\epsilon_\phi(^\circ)$	2.81	0.39	1.5	0.86	0.32

B. Security Analysis

To analyze the security of the LIDAR protocol, consider a scenario where an attacker tries to launch a man-in-the-middle-attack against V_A and V_B. Suppose the attacker V_M tries to impersonate V_A to V_B. The V_A transmits its commitment and decommitment (C_A and D_A) to V_B. The attacker can prevent V_B from receiving the D_A by jamming and send its own decommitment $D_{A'} = g^{a'}||ID_A||map_A\prime||Ts$. However, V_A's DH parameters g^a along with $ID_A||map_A||Ts$ are bounded with the commitment C_A by a hash function, such as SHA-3 algorithm. Also, the locality information $map_A\prime$ cannot be

verified because the distance d and the angle ϕ at the time Ts are the unique information. Hence, the man-in-the-middle-attack cannot be successful under the LIDAR protocol.

VI. Discussion & Future Work

There are a few important points that are worth to discuss. The lidar sensor may not be able to detect some objects if it is completely shadowed by the front object due to the nature of the lidar. (i.e., if the object falls in the line of sight from the vehicle to objects.) To reduce possible error during the verification, a similarity score can be used. Also such objects can be identified and filtered out from the object list. Besides, we assume the sensor fusion to be applied when it requires. Sensor fusion provides the extra information and could be the huge benefit for analyzing the surrounding environment of vehicles. Further, we also assumed the application of the machine learning algorithm in the projected 3D to 2D space but only in XY plane. The algorithm can compare the object even more accurately if the object is recognized in the 3D model. In addition, the proper threshold of the error needs to be determined considering the factors mentioned in the above scenarios. This paper mainly focuses on the feasibility to authenticate vehicles by leveraging the on board-sensors installed on vehicle, and we will consider the above-mentioned constraints as our future work.

VII. Conclusion

Authentication in V2V communication has been relied on the trusted authority and it requires infrastructures to convey the message back and forth, and assist verification process, however, such solutions still have some issues as a third party trusted authority has multiple points of failures and infrastructures increases overhead. Also, infrastructures for V2V communication are not readily available yet, even though self-driving vehicles are cruising on the roads. Therefore, and an alternative solution is necessary to support the areas where the infrastructure is not available. To address this concern, we propose a lidar information based authentication for infrastructure-less vehicular networks, called LIDAR. The LIDAR leverages available hardware to authenticate vehicles based on the shared surrounding information. A secure V2V symmetric key is established after verifying the common object types, its distance and angle. The scheme is robust against the man-in-the-middle-attack because the hash value of the commitment including the sensor information is verified during the key agreement process. We implemented and analyzed our protocol in an autonomous urban driving model. The experiment results show that our scheme provide a dynamic decentralized authentication for V2V communication with a very small error in verifying surrounding objects.

References

[1] T. Litman, *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute Victoria, Canada, 2017.

[2] M. M. Waldrop *et al.*, "No drivers required," *Nature*, vol. 518, no. 7537, p. 20, 2015.

[3] H. Hartenstein and K. Laberteaux, *VANET: vehicular applications and inter-networking technologies*. John Wiley & Sons, 2009, vol. 1.

[4] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of computer security*, vol. 15, no. 1, pp. 39–68, 2007.

[5] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, 2016.

[6] IEEE Standard 1609.2, "1609.2-2016 - IEEE standard for wireless access in vehicular environments–security services for applications and management messages," *IEEE Xplore*, pp. 1–240, 2016.

[7] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[8] K. Lim and D. Manivannan, "An efficient protocol for authenticated and secure message delivery in vehicular ad hoc networks," *Vehicular Communications*, vol. 4, pp. 30–37, 2016.

[9] S. Tangade, S. S. Manvi, and P. Lorenz, "Decentralized and scalable privacy-preserving authentication scheme in vanets," *IEEE Transactions on Vehicular Technology*, 2018.

[10] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations," *Vehicle System Dynamics*, vol. 44, no. 7, pp. 569–590, 2006.

[11] S. E. Reutebuch, H.-E. Andersen, and R. J. McGaughey, "Light detection and ranging (LIDAR): an emerging tool for multiple resource inventory," *Journal of Forestry*, vol. 103, no. 6, pp. 286–292, 2005.

[12] D. Huang, S. Misra, M. Verma, and G. Xue, "PACP: An efficient pseudonymous authentication-based conditional privacy protocol for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 736–746, 2011.

[13] T. Gao, X. Deng, N. Guo, and X. Wang, "An anonymous authentication scheme based on PMIPv6 for VANETs," *IEEE Access*, vol. 6, pp. 14 686–14 698, 2018.

[14] Y. Liu, Y. Wang, and G. Chang, "Efficient privacy-preserving dual authentication and key agreement scheme for secure v2v communications in an IoV paradigm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2740–2749, 2017.

[15] J. Shao, X. Lin, R. Lu, and C. Zuo, "A threshold anonymous authentication protocol for VANETs," *IEEE Transactions on vehicular technology*, vol. 65, no. 3, pp. 1711–1720, 2016.

[16] K. Lim and K. M. Tuladhar, "Trajectory based pre-key exchange scheme for seamless vehicular networks connectivity," in *Proceedings of the Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*. IEEE, 2018, pp. 1–5.

[17] Y. Maalej, S. Sorour, A. Abdel-Rahim, and M. Guizani, "VANETs meet autonomous vehicles: A multimodal 3d environment learning approach," in *Proceedings of the GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[18] H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, "Cooperative BSM architecture to improve transportation safety in vanets," in *Proceedings of the Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International*. IEEE, 2017, pp. 1016–1022.

[19] B. Schwarz, "Lidar: Mapping the world in 3d," *Nature Photonics*, vol. 4, no. 7, p. 429, 2010.

[20] M. Ruta, F. Scioscia, F. Gramegna, S. Ieva, E. Di Sciascio, and R. P. De Vera, "A knowledge fusion approach for context awareness in vehicular networks," *IEEE Internet of Things Journal*, 2018.

[21] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[23] "Epic Games: Unreal engine 4.18," in *Online]. https://www.unrealengine.com*, 2017.

[24] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: an open-source mesh processing tool." in *Proceedings of Eurographics Italian chapter Conference*, vol. 2008, 2008, pp. 129–136.