Authenticating Drone-Assisted Internet of Vehicles Using Elliptic Curve Cryptography and Blockchain

Mohamed A. El-Zawawy, Alessandro Brighente, *Member, IEEE,* and Mauro Conti, *Fellow, IEEE,*

***Abstract*—The inclusion of drones in Internet of Vehicles (IoV) is a current trend that presents significant trade-offs. On the one hand, Unmanned Aerial Vehicles (UAVs) provide advantages such as enabling ground communications also when physical obstacles limit the connectivity. On the other hand, they increase the attack surface. For instance, physical attacks on drones provide the attacker with credentials that can be used to inject bogus information into the IoV network, thus jeopardizing not only security but also users' safety. In this scenario, authentication plays a fundamental role to guarantee security. It is however fundamental to develop authentication protocols that can, at the same time, protect ground users' data and prevent attacks to drones. However, currently available authentication schemes cannot guarantee security in case of attacks to drones.**

In this paper, we propose a Blockchain-supported authentication protocol for Drone-assisted IoV using Elliptic curve cryptography (BDIVE). Compared to existing authentication protocols, we extend the threat model from an honest-but-curious drone to active attacks against drones. BDIVE provides both energy-efficiency, traceability, and accountability thanks to the use of blockchain at the Trusted Authority (TA). Using Burrow–Abadi–Needham (BAN) logic, we analyze and prove the security of mutual authentication in BDIVE. We also prove the security of BDIVE against several attacks by implementing it in AVISPA. To assess its scalability and energy efficiency, we implement BDIVE using Omnetpp with its Castalia simulator. The comparison of BDIVE with currently existing authentication protocols, shows that it reduces the energy consumption up to 70% and the computational cost up to 68%, while providing resistance to previously unconsidered attack vectors.

***Index Terms*—Internet of Vehicles, Drones, Blockchain, Authentication**

I. Introduction

The Internet of Vehicles (IoV) may benefit from the use of Unmanned Aerial Vehicles (UAVs) (or drones) for multiple purposes, including packet routing, task offloading, extended communication range, and improved quality of service [1]. Thanks to their flying capabilities, UAVs are not constrained by the physical structure of the roads and are able to freely move to support the network where needed [1]. Therefore, they provide a significant advantage compared to other network entities that only provide localized services, such as edge servers or Road Side Units (RSUs). Besides increased connectivity and network performance (e.g., in terms of delay and reliability) [2], UAVs may also provide other advantages to the IoV. For instance, thanks to drones it is possible to reduce the number of deployed infrastructural nodes (therefore lowering the deployment costs) while at the same time guaranteeing the absence of link quality degradation due to obstacles [1]. On the other hand, the introduction of a larger number of connected devices imposes additional security challenges. In the case of non-secure communications, a malicious user can capture and modify the information transmitted among the different network entities having multiple consequences. Information exchanged in the IoV may be related to sensitive users' data, which may lead to their tracking or profiling. Furthermore, an attacker may be able to compromise the safety of the road by modifying or injecting messages that may cause inconsistencies and possibly car crashes. Therefore, it is fundamental to secure UAV-supported IoV against cybersecurity attacks. To this aim, authentication plays a fundamental role to allow only authorized entities to deliver messages in the IoV.

Currently available state-of-the-art authentication solutions for IoV either do not account for the presence of drones [3], [4], [5], [6] or do not consider the fundamental needs of this type of network. First, they do not provide an energy demand analysis [7]. In fact, drones are battery-limited devices whose lifespan depends on the complexity of the operations they should execute. Authentication is a necessary step, which however should not consume a large amount of energy. Second, they consider a simple attacker model. Indeed, an honest but curious drone is a fair adversary model, however, it is not sufficient to guarantee the security of the network. Drones can in fact be stolen by an adversary that can hence actively attack the network. Lastly, they consider a centralized identity and session manager [7]. Keeping track of identities and sessions is a fundamental requirement to provide accountability in an IoV network, with applications ranging from insurance management to traffic monitoring. However, a single entity managing all this information might be corrupted hence jeopardizing the security of the overall network.

In this paper, we propose a Blockchain-supported authentication protocol for Drone-assisted IoV using Elliptic curve cryptography (BDIVE), a novel authentication scheme solving the aforementioned issues. We particularly focus on the Vehicle-to-Vehicle (V2V) communication as it has been recognized as one of the two communication links where cyberattacks are most likely to occur in IoV [8]. By employing elliptic curve cryptography, we provide an effective and energy-efficient solution. Compared to the existing literature on the subject, we consider a stronger attacker model moving from a passive-but-curious drone to active attacks

M. A. El-Zawawy is with the Department of Mathematics, Faculty of Science, Cairo University, Giza 12613, Egypt (e-mail: maelzawawy@cu.edu.eg).

M. Conti and A. Brighente are with the Department of Mathematics and HIT Research Center, University of Padova, Italy (e-mail: alessandro.brighente@unipd.it; mauro.conti@unipd.it).

M. Conti is also affiliated with TU Delft.

Corresponding author: M. A. El-Zawawy.

Manuscript received April 19, 2005; revised August 26, 2015.

against it considering e.g., that a UAV might be stolen and compromised. Furthermore, we strengthen the security of the overall network by using blockchain technology at the Trusted Authority (TA) side We use transactions to store authentication sessions among the different entities, collecting also meta-information related to the context (e.g., location). This allows for traceability and accountability of the exchanged messages, and also mitigates possible attacks against the data stored at the TA's server.

Our contributions can be summarized as follows.

- We propose BDIVE, a novel elliptic curve cryptography based authentication scheme for drone-assisted V2V communications in IoV. BDIVE not only guarantees mutual authentication and key agreement, but also allows for the dynamic addition of vehicles.
- Compared to the state-of-the-art, we strengthen the security of the overall IoV network with respect to two different perspectives: i) we consider active attacks towards drones; ii) we provide traceability, accountability, and protection against attacks aiming at modifying the information stored at the TA by using blockchain technology.
- We prove the security of BDIVE via both BAN Logic and via implementation thorough AVISPA tool and compare its security features to those of other available state-of-the-art authentication protocols.
- Via numerical evaluation, we compare the communication and energy costs of BDIVE with that of other available state-of-the-art authentication schemes. Our results show that BDIVE, compared to other authentication schemes, reduces the energy consumption up to 70% and the computational cost up to 68%.

The rest of the paper is organized as follows. In Section II, we describe the related works. In Section III, we present the system and threat models. In Section IV, we present BDIVE and provide all its details. In Section V, we show the security of BDIVE using BAN logic. In Section VI, we compare BDIVE with other relevant state-of-the-art schemes in terms of security, communication cost, and energy cost. We then derive the conclusions and discuss possible future works in Section VII.

II. Related Work

In this section we present the related literature on the use of drones in vehicular networks (Section II-A), authentication in IoV and Vehicular Ad Hoc Network (VANET) (Section II-B), and blockchain for vehicular networks (Section II-C).

A. Use of Drones in Vehicular Networks

Drones have been first introduced to provide longer range and continuous network communication among vehicles and from vehicle to infrastructure [9]. In fact, in VANET, vehicles communicate through dedicated short-range communications, whose connection coverage is limited in space. Therefore, drones are used as relaying entities, amplifying and forwarding the signal. Later, given the increasing processing capabilities of drones, they have also been considered for task offloading at the network's edge [10] and for collecting data that can be used for federated learning [11]. Drones can also help sustain high-traffic loads and may be used to extend the network infrastructure considering the possibilities given by drone-to-drone communications [12]. Drones have not only been deployed to help vehicles, but also to increase the communication with users and guarantee a certain quality of service [13]. In fact, they can be exploited to securely collect information regarding the road status to facilitate autonomous driving or to provide users with feedback on safety-critical situations [14].

B. Authentication in IoV

The use of drones in IoV is a relatively novel paradigm, which has been mostly considered to enhance communication capabilities. The literature on authentication protocols for drone-assisted IoV is scarce, and most of the authentication protocols for IoV do not consider the presence of drones. For instance, Vasudev et al. [3] considered the need for lightweight operations in authentication schemes in order for them to be suitable for IoV networks. Indeed, guaranteeing the communication security shall not come at the cost of high communication delays, as this may have an impact on the road safety in the IoV. To deal with the lightweight requirement of authentication in IoV, Chuang et al. [4] proposed an authentication scheme based on the concept of trust to minimize the number of required message exchanges. Chen et al. [5] deal with the scalability problem of IoV by proposing an iterative method for identity verification, while Aman et al. [6] propose a three-layered infrastructure to reduce the number of messages on the communication channel. A different approach has been proposed by Song et al. [15], where the complexity of key management in the IoV context is reduced thanks to the application of fog computing, where specific vehicles act as fog heads and are hence responsible for managing a subset of vehicles. However, none of these proposals consider the existence of drones in the network, therefore they cannot be easily extended to provide secure authentication also in the case where drones are used to support IoV. Zhang et al. [16], propose the use of drones to authenticate communication in VANET. However, the authors assume that the drone is an honest but curious device, without including the possibility of the device being stolen. As this represents a concrete threat in drone-supported IoV, we instead include it in the design of BDIVE. Furthermore, Zhang et al. did not account for the energy efficiency of the protocol, therefore not providing guarantees on the feasibility of its implementation on battery-limited drones. BDIVE is instead highly energy-efficient, thus increasing the drone's lifetime.

C. Use of Blockchain in VANET and IoV

Blockchain has been proposed to serve different scopes in the context of vehicular networks [17]. Jiang et al. [18] propose to divide the whole IoV network into smaller sub-networks, each characterized by a certain sub-set of vehicles and RSUs as blockchain nodes. The authors show how blockchain can be used to improve the performance of IoV. Kang et al. [19]

proposed a mechanism based on the vehicle's reputation to provide secure solutions against collusion. The social aspect of IoV has been discussed by Butt et al. [20], where the authors proposed a blockchain solution to deal with users' sensitive data. A blockchain solution to block malicious users from accessing the IoV network has been proposed by Amad et al. [6], where user authentication is based on physical unclonable functions and smart contract. The closest solution to our proposal has been proposed by Zhang et al. [7], where the blockchain is used to enable the message propagation among vehicles. In [21], Vangala et al. propose a blockchain-based scheme for Internet of Things (IoT) devices, where devices are distributed in zones. This work shares some similarities with our scheme. However, in our scheme we explicitly consider zones to select the cryptographic material to use when communicating with zone-specific deployed vehicles. Furthermore, we use blockchain technology to secure the information at the TA side besides providing traceability and accountability of the collected data.

III. System and Threat Model

In this section, we first introduce the system model in Section III-A. Then, we present our considered threat model in Section III-B.

A. System Model

We consider the system depicted in Figure 1. Our proposed model has the following components.

- Vehicle: is a mobile object (moving at high speed) that is equipped with a device for Dedicated Short-Range Communication (DSRC). A vehicle can request communication from an Assisting Drone (AD) or a RSU. We assume that in a pre-registration action, each vehicle is assigned a smart card loaded with an ID and password.
- Assisting Drone: is an Unmanned Aircraft System (UAS) that is equipped with Inertial Measurement Unit (IMU), battery, flight controller, a communication module, and rotors. Each drone collaborates with a pair of Road Side Units (RSUs) to manage vehicle communications in a specific section of the road.
- Road Side Unit: is a server on the roadside. It is equipped with DSRC and other necessary network-enabling components. RSUs are placed at regular intervals along the road. Hence RSUs are partitioning the road into *regions*, where we define each region as the road space delimited by two consecutive RSUs. Each region is managed by an AD and a pair of RSUs. Hence RSUs reply to vehicles' communication requests. We assume that each RSU manages only one zone of the road.
- Trusted Authority: this is a fully trusted component that registers ADs, Vehicles (VHs), and RSUs. TA also handles the authentication processes among the different components of the system.
- Blockchain cloud sever: it represents a network of P2P entities that manage the information collected by the TA. The P2P cloud server uses blockchain technology to store transactions related to the key establishment procedures among the different entities, providing traceability, accountability and mitigating possible attacks to TA's database. We will later provide the details on its implementation.

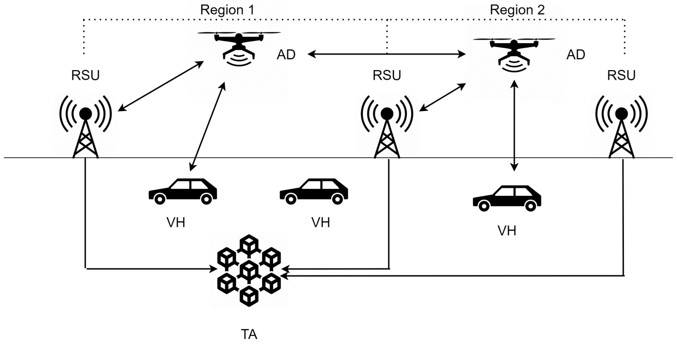


Fig. 1: Proposed Network Model for Drone Assisted V2V Communication.

We assume that RSUs are located far apart, and hence unable to appropriately support the huge number of requests coming from VHs. Therefore, each pair of consecutive RSUs is backed up by an AD that can freely fly in a zone to enhance its communication capacity. All VHs, ADs, and RSUs must register with TA to participate in V2V communications. Hence, TA stores the data related to all system entities and controls all the system's communication. The objective of our paper is to build a security protocol that guarantees validation and authentication of all system components.

The registration of VHs, drones, and RSUs with TA is done using secure channels. After different authentications among system entities during the communication phase, the entities generate communication session keys. A pair of RSUs and their drone communicate after exchanging keys through TA. The VHs communication in the same zone is done after exchanging session keys through the zone drone. The communication between vehicles in different zones is done after exchanging session keys between drones and RSUs through TA.

As common in the literature, we assume that parameter initialization and offline registration at the TA cannot be compromised [3], [4], [5]. However, we assume that an attacker might try to modify the information on already registered entities stored in TA's database. For instance, an attacker may try to delete entries from the TA database or to update entries on behalf of a victim user via impersonation attack. We also assume that VHs and drones that are not registered with the TA can not participate in the V2V communication. Moreover, registered VHs and drones do not share their credentials outside the protocol boundaries.

B. Threat Model

In this paper, we adopt both the Dolev-Yao (DY) [22] and the Canetti-Krawczyk (CK)-adversary models [23]. We adopt these models because they represent standard models and capture basic capabilities that an adversary may have.

Furthermore, they are universally accepted to assess the cryptanalytic security of authentication protocols. DY assumes that an adversary controls open-channel messages in wireless networks. Therefore, an adversary can delete, inject, eavesdrop, and modify valid messages transmitted in wireless public channels. According to this model, it is also possible for the adversary to steal smart cards (e.g., by physically capturing the drone) and obtain their confidential information via power analysis attacks [24], [25]. DY assumes that only one value can be guessed by an adversary in a polynomial time. This assumption is in line with the fact that it is anyway not feasible (computationally) to guess multiple values at once. To further assess the security of BDIVE, we consider also the CK model, where the adversary can compromise secret credentials and session states.

The impact of attacks to IoV is strictly related to road safety. In fact, an attacker able to inject or modify road management information may cause crashes among multiple vehicles. This impacts both the infrastructure and drivers' safety. Therefore, it is fundamental to ensure the legitimacy and integrity of the exchanged information. We assume the following extra adversary capabilities. The adversary can launch ephemeral secret leakage, anonymity, and untraceability attacks. The adversary can also launch impersonation, physical system entity capture, privileged insider, and DoS attacks. These capabilities do not conflict with our assumed standard modes. Furthermore, some of these capabilities may be implicitly included in DY and CK. However, we prefer to explicitly list them [26].Lastly, we assume that an internal attacker (e.g., a vehicle) may try to propagate false information to the TA database. This type of attacker represents a critical point, as it posses valid cryptographic material and cannot be blocked by means of access control.

IV. BDIVE: New Authentication Protocol for Drone Assisted Vehicle Communication

In this section, we present BDIVE, our new scheme for authenticated key management in drone-assisted V2V communications. Based on the model of Figure 1, BDIVE effectively overcomes the limitations and security issues of existing authentication protocols. As common in the related literature [27], [28], we assume that each network system entity is synchronized with its device clock.

The design of BDIVE takes into consideration the usability of the underlying system. This is an important aspect as it facilities the application of BDIVE in practice. Furthermore, from a safety point-of-view, it is essential in the IoV that BDIVE works most of the times. For such motivations, rather than creating three-party session keys, BDIVE creates two kinds of two-party session keys: (drone, road side unit) and (vehicle, road side unit). Hence in case of unavailability of drones, the two-party (vehicle, road side unit) session keys can be used as a backup plan to keep the system running. Hence our protocol maintains high system usability.

The notations that we use to present and analyze BDIVE are presented in Table I. Section IV-A provides a high-level overview of the protocol phases.

A. Phases of BDIVE

BDIVE is divided into the following phases.

1) *System initialization:* the protocol fixes all the parameters of the network system entities.
2) *Registration:* the TA registers all other parties of the system.
3) *Login and Authentication:* The protocol performs two types of authentications. The first one is AD-RSU authentication, in which the protocol authenticates drones and roadside units and generates secret session keys necessary for communications among these system entities. The other type is Vehicle (VH)-AD authentication, in which the protocol creates secret session keys necessary for communications among VHs and AD. This involves performing the necessary authentication steps.
4) *V2V communication:* the protocol implements V2V communications exploiting the aforementioned secured channels.
5) *Dynamic vehicle addition:* the protocol, via this phase, allows to seamlessly deploy new vehicles to the system.
6) *Blockchain management:* this phase takes care of all necessary blockchain actions including creating blocks and verifying and inserting them into the Blockchain (BC).

We describe in detail the protocol phases in the following sections.

TABLE I: Notation used in this paper.

Notation	Meaning
$\mathcal{H}(\cdot)$	Collision-resistant one-way cryptographic hash function.
q	a prime number.
$\{q_1, q_2\}$	A subset of $Z_q = \{0, 1, \ldots, q-1\}$.
$\mathcal{E}(q_1, q_2)$	A non-singular elliptic curve.
z	a zero for $\mathcal{E}(q_1, q_2)$.
b	is a base point in $\mathcal{E}(q_1, q_2)$ whose order O_b satisfies $O_b b = z$.
n_v	Random secret from Z_q^* for a vehicle v.
$p\mathcal{I}_v$	An auxiliary ID for v.
$p\mathcal{P}_v$	An auxiliary password for v.
C_v	A credential of v.
pK_d	Public key for d.
TA	Trusted Authority.
r	Road Side Unit.
AD	Assisting Drone.
CSN	Cloud Server Network.
v	Vehicle.
d	Drone.
ECC	Elliptic Curve Cryptography.
PBFT	Practical Byzantine Fault Tolerance.
K_{TA}	The master key of TA.
VANET	Vehicular Ad Hoc Network.
OBU	On-Board Unit.
DSRC	Dedicated Short-RangeCommunication.
BC	Blockchain.

B. System Initialization

During initialization, the TA fixes the parameters to use in the following phases. The TA also publishes some of the fixed parameters, some of which are related to our use of Elliptic Curve Cryptography (ECC) [21] in BDIVE. The fixed parameters are as follows.

1) A "collision-resistant one-way cryptographic hash function" denoted by $\mathcal{H}(\cdot)$.
2) A tuple $< q_1, q_2, z, b, k >$ such that:
   - $\{q_1, q_2\} \subseteq Z_q = \{0, 1, \ldots, q-1\}$, where q is a prime number.
   - $4q_1^3 + 27q_2^2 \neq 0 (mod\ q)$.
   - $\mathcal{E}(q_1, q_2) : b^2 = a^3 + q_1 a + q_2 (mod\ q)$ is a non-singular elliptic curve over the Galois field $GF(q)$.
   - z is a zero for $\mathcal{E}(q_1, q_2)$.
   - $b \in \mathcal{E}(q_1, q_2)$ is a base point whose order O_b satisfies $O_b b = z$.
   - $K_{TA} \in Z_q^*$. This is a secret key for TA.
3) A Practical Byzantine Fault Tolerance (PBFT) algorithm. This is needed for the consensus process used in TA's blockchain.

Finally, TA publishes to all parties the parameters $\mathcal{H}(\cdot), \mathcal{E}(q_1, q_2)$, and b during the registration time.

C. System Entities Registration

In this phase which is a one-time process, the TA offline registers all the system entities, i.e., VHs, Assisting Drones (ADs), RSUs, and Cloud Server Network (CSN). Algorithm 1 shows the registration steps for all entities. We describe all the involved steps in the following sections.

1) Drone Registration: A drone d registration is achieved via the following steps. TA fixes a real ID and a random secret $n_d \in Z_q^*$ in lines 7 and 8 of the algorithm. The TA also includes the current time stamp T_d. Next, the TA computes a public key pK_d as $bH(n_d \oplus \mathcal{I}_d)$ in line 10. An auxiliary ID $p\mathcal{I}_d$ is computed as $H(\mathcal{I}_d \parallel K_{TA} \parallel n_d)$ in line 11. Then TA calculates the drone credentials C_d in line 12 as $H(p\mathcal{I}_d \parallel T_d \parallel K_{TA} \parallel n_d)$. Finally in line 13, TA preloads the drone d with the credentials $(\mathcal{I}_d, p\mathcal{I}_d, \mathcal{C}_d, n_d, pK_d, \mathcal{H}(\cdot), \mathcal{E}(q_1, q_2), b)$. Moreover, TA publishes the drone public key pK_d.

2) Roadside Unit Registration: TA performs the following steps to register a RSU r. In line 16, TA fixes a real ID $\mathcal{I}_r$ for r. TA then determines the drone d managing the zone that r is assigned to. Hence, d is the drone communicating with r. Then, a random secret $n_r \in Z_q^*$ is fixed in line 18 of the algorithm. The current time stamp is stored in T_r. Next a public key pK_r is calculated as $b\ .\ H(n_r \oplus \mathcal{I}_d \oplus \mathcal{I}_r)$ in line 20. An auxiliary ID $p\mathcal{I}_r$ is computed as $H(\mathcal{I}_r \parallel \mathcal{I}_d \parallel K_{TA} \parallel n_r)$ in line 21. Then, TA calculates the credential of r in line 22 as $H(p\mathcal{I}_r \parallel \mathcal{I}_d \parallel T_r \parallel K_{TA} \parallel n_r)$. Finally in lines 23 and 24, TA preloads the roadside unit with r with the credentials $(\mathcal{I}_d, p\mathcal{I}_d, \mathcal{C}_d, \mathcal{I}_r, p\mathcal{I}_r, \mathcal{C}_r, n_r, pK_r, \mathcal{H}(\cdot), \mathcal{E}(q_1, q_2), b)$. In addition, TA publishes the public key pK_r. It is worth noticing that r is preloaded with some credentials of its associated drone.

3) Vehicle Registration: TA registers a vehicle v via performing the following steps. In line 27, TA reads from a vehicle's smart card a real ID, $\mathcal{I}_v$ and a password $\mathcal{P}_v$ Then, a random secret $n_v \in Z_q^*$ is generated in line 28 of the algorithm. Next a public key pK_v is calculated as $b\ .\ H(n_v \oplus \mathcal{I}_v)$ in line 29. Auxiliary ID and password $p\mathcal{I}_v$ and $p\mathcal{P}_v$ are calculated in lines 30 and 31 as $H(\mathcal{I}_v \parallel K_{TA} \parallel n_v)$ and $H(\mathcal{P}_v \parallel \mathcal{I}_v \parallel K_{TA} \parallel n_v)$, respectively. The current time stamp is stored in T_v. Then, TA calculates the credential of v in line 33 as $H(p\mathcal{I}_v \parallel p\mathcal{P}_v \parallel T_v \parallel K_{TA} \parallel n_v)$. TA publishes the public key pK_v. Finally in lines 34 and 35, TA preloads v with the credentials $(\mathcal{I}_v, p\mathcal{I}_v, \mathcal{P}_v, p\mathcal{P}_v, \mathcal{C}_v, n_v, pK_v, \mathcal{H}(\cdot), \mathcal{E}(q_1, q_2), b)$.

Algorithm 1 Registration Details

```
Input: All the system model entities.
  Steps:
1: Call $TA_1$(D: system drones);
2: Call $TA_2$(R: system road side units);
3: Call $TA_3$(V: system vehicles);
4: Call $TA_4((c, \mathcal{I}_c)$ : cloud server);
5: procedure $TA_1$(D: SYSTEM DRONES)
6:     for each drone $d \in D$ do
7:         Fix $\mathcal{I}_d$;
8:         Pick $n_d \in Z_q^*$;
9:         $T_d \leftarrow Time\_Stamp()$;
10:        $pK_d \leftarrow b\ .\ H(n_d \oplus \mathcal{I}_d)$; (Public key)
11:        $p\mathcal{I}_d \leftarrow H(\mathcal{I}_d \parallel K_{TA} \parallel n_d)$; (Auxiliary ID)
12:        $\mathcal{C}_d \leftarrow H(p\mathcal{I}_d \parallel T_d \parallel K_{TA} \parallel n_d)$; (Drone credentials)
13:        $d \leftarrow (\mathcal{I}_d, p\mathcal{I}_d, \mathcal{C}_d, n_d, pK_d, \mathcal{H}(\cdot), \mathcal{E}(q_1, q_2), b)$
14: procedure $TA_2$(R: SYSTEM ROAD SIDE UNITS)
15:     for each road side unit $r \in R$ and a drone $d$ communicating with $r$ do
16:        Fix $\mathcal{I}_r$;
17:        Fix the drone $d$ communicating with $r$;
18:        Pick $n_r \in Z_q^*$;
19:        $T_r \leftarrow Time\_Stamp()$;
20:        $pK_r \leftarrow b\ .\ H(n_r \oplus \mathcal{I}_d \oplus \mathcal{I}_r)$;
21:        $p\mathcal{I}_r \leftarrow H(\mathcal{I}_r \parallel \mathcal{I}_d \parallel K_{TA} \parallel n_r)$;
22:        $\mathcal{C}_r \leftarrow H(p\mathcal{I}_r \parallel \mathcal{I}_d \parallel T_r \parallel K_{TA} \parallel n_r)$;
23:        $r \leftarrow (\mathcal{I}_d, p\mathcal{I}_d, \mathcal{C}_d)$;
24:        $r \leftarrow r + (\mathcal{I}_r, p\mathcal{I}_r, \mathcal{C}_r, n_r, pK_r, \mathcal{H}(\cdot), \mathcal{E}(q_1, q_2), b)$;
25: procedure $TA_3$(V: SYSTEM VEHICLES)
26:     for each vehicle $v \in V$ do
27:        Read, from $v$, $\mathcal{I}_v$ and $\mathcal{P}_v$;
28:        Pick $n_v \in Z_q^*$;
29:        $pK_v \leftarrow b\ .\ H(n_v \oplus \mathcal{I}_v)$;
30:        $p\mathcal{I}_v \leftarrow H(\mathcal{I}_v \parallel K_{TA} \parallel n_v)$;
31:        $p\mathcal{P}_v \leftarrow H(\mathcal{P}_v \parallel \mathcal{I}_v \parallel K_{TA} \parallel n_v)$;
32:        $T_v \leftarrow Time\_Stamp()$;
33:        $\mathcal{C}_v \leftarrow H(p\mathcal{I}_v \parallel p\mathcal{P}_v \parallel T_v \parallel K_{TA} \parallel n_v)$;
34:        $v \leftarrow (\mathcal{I}_v, p\mathcal{I}_v, \mathcal{P}_v, p\mathcal{P}_v)$
35:        $v \leftarrow v + (\mathcal{C}_v, n_v, pK_v, \mathcal{H}(\cdot), \mathcal{E}(q_1, q_2), b)$
36: procedure $TA_4((c, \mathcal{I}_c)$ : CLOUD SERVER)
37:     Pick $n_c \in Z_q^*$;
38:     $pK_c \leftarrow b\ .\ H(n_c \oplus \mathcal{I}_c)$;
39:     $p\mathcal{I}_c \leftarrow H(\mathcal{I}_c \parallel K_{TA} \parallel n_c)$;
40:     $c \leftarrow (\mathcal{I}_c, p\mathcal{I}_c, n_c, pK_c, \mathcal{H}(\cdot), \mathcal{E}(q_1, q_2), b)$
```

4) Cloud Server Registration: TA registers the cloud server c with ID $\mathcal{I}_c$ via performing the following steps. In line 37, TA generates a random secret $n_c \in Z_q^*$. Next a public key pK_c is calculated as $b\ .\ H(n_c \oplus \mathcal{I}_c)$ in line 38. An auxiliary

ID $p\mathcal{I}_c$ is calculated in lines 39 as $H(\mathcal{I}_c \parallel K_{TA} \parallel n_c)$. Finally, in lines 40 TA preloads c with the credentials $(\mathcal{I}_c, p\mathcal{I}_c, n_c, pK_c, \mathcal{H}(\cdot), \mathcal{E}(q_1, q_2), b)$ and publishes pK_c.

D. Login and Authentication

Before different system entities can communicate with one another, they must log in and authenticate. Upon successful login and authentication operations, they can establish secure mutual communications. Two types of authentications can take place in this phase. We provide the details on these types in the following sections.

1) AD-RSU authentication: The performers of this authentication type are the AD d and roadside unit r. We present the authentication details in Algorithms 2 and 3. This process involves generating secret session keys necessary for communications among these system entities. It is worth noticing that the TA is not involved in the process.

The AD starts this authentication process. This is done in the first step of Algorithm 2 which calls the method $AD_1()$ that is executed by the drone d. The logic of $AD_1()$ is as follows. Lines 6 and 7 generate a random secret $n_d^2 \in Z_q^*$ and read the current time stamp into T_d^2, respectively. In Line 8, the drone utilizes the generated parameters so far to create M_1 as $b\ .\ H(\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_d^2 \parallel T_d^2)$. The M_2 is built on M_1 via the calculation $b\ .\ H(M_1 \oplus\ (p\mathcal{I}_d \parallel\ \mathcal{C}_d \parallel\ n_d^2 \parallel\ T_d^2))$. Then, the credential S_1 is calculated via $H(\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_d^2 \parallel T_d^2) + H(M_1 \oplus\ (p\mathcal{I}_d \parallel\ \mathcal{C}_d \parallel\ n_d^2 \parallel\ T_d^2)) + H(n_d\ \oplus\ \mathcal{I}_d)(mod\ q)$. Finally, the method $AD_1()$ sends to r via a public channel the message $(\mathcal{I}_d, M_1, M_2, S_1, n_d^2, T_d^2)$.

In response to the message from d, r executes RSU_1. The entity r gets the current time stamp into T_r^2 in Line 14. Then, in Line 15, r checks the validity of the received time stamp using the condition $|T_r^2 - T_d^2| \leq \Delta T$, where ΔT is the allowed transmission delay. Only if the time stamp is valid, r checks the validity of the received message via the condition $b\ .\ S_1 == M_1 + M_2 + pK_d(mod\ q)$. If the message is not valid, then the algorithm terminates. Otherwise, r executes the following calculations. In Line 17, r generates a random secret $n_r^2 \in Z_q^*$. Then r extracts a part $(H(\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_d^2 \parallel T_d^2))$ of the received message M_1 and name it M_4. Then three message entities M_5, M_6, and M_7 are calculated to be used to create the session key as $b\ .\ H(p\mathcal{I}_d \parallel n_r^2 \parallel p\mathcal{I}_r \parallel T_r^2)$., $M_1\ \oplus\ H(p\mathcal{I}_r \parallel\ p\mathcal{I}_d \parallel\ \mathcal{C}_d \parallel\ n_r^2 \parallel\ T_r^2)$, and $M_2\ \oplus\ H(p\mathcal{I}_r \parallel\ p\mathcal{I}_d \parallel S_1 \parallel \mathcal{C}_r \parallel\ n_r^2 \parallel\ T_d^2)$ in Lines $19, 20$, and 21, respectively. In Line 22, r creates its version of the session key $SK_{R\to D}$ using $H((M_5\ \oplus\ M_7) \parallel H(p\mathcal{I}_r \parallel\ p\mathcal{I}_d \parallel\ \mathcal{C}_d \parallel\ n_r^2 \parallel\ T_r^2))$. The parameters needed to recreate the session key are wrapped in S_2 as $H(p\mathcal{I}_d \parallel\ n_r^2 \parallel\ p\mathcal{I}_r \parallel\ T_r^2) + H(p\mathcal{I}_r \parallel\ p\mathcal{I}_d \parallel S_1 \parallel \mathcal{C}_r \parallel\ n_r^2 \parallel\ T_d^2) + H(n_r\ \oplus\ \mathcal{I}_d\ \oplus\ \mathcal{I}_r)(mod\ q)$. Finally, r sends the message $(p\mathcal{I}_r, T_r^2, M_5, M_6, M_7, S_2)$ to the drone d on a public channel.

Upon receiving the message from r, d executes $AD_2()$ of Algorithm 3. The entity d gets the current time stamp into T_d^3 in Line 2. Then, it checks the validity of the received time stamp in Line 3 via the condition $|T_d^3 - T_r^2| \leq \Delta T$. If the time stamp is valid, d moves on and checks the validity of the received message, in Line 4, via the condition $b\ .\ S_2 == M_5 + b\ .\ (M_2 \oplus M_7) + pK_r(mod\ q)$. If the message is valid the algorithm executes the following calculations, otherwise it terminates. In Line 5, d extracts the value M_9 from the received message M_6 using its stored parameter M_1. In Line $6, d$ uses M_9 to create the drone version of the session key $SK_{D\to R}$ as $H((M_6\ \oplus\ M_7) \parallel M_9)$. This key is then wrapped in Line 7 as $SKV_{D\to R}$ using the calculations $H(T_d^3 \parallel SK_{D\to R})$. Finally, in Line $9, d$ sends back to r the message $(T_d^3, SKV_{D\to R})$ via a public channel. In response to

Algorithm 2 `(Drone, Roadside Unit) Key Agreement`

```
Input: A drone $d$ and a road side unit $r$.
  Steps:
 1: $\mathcal{I}_d, M_1, M_2, S_1, n_d^2, T_d^2 \leftarrow AD_1()$;
 2: $p\mathcal{I}_r, T_r^2, M_5, M_7, M_8, S_2 \leftarrow RSU_1(\mathcal{I}_d, M_1, M_2, S_1, n_d^2, T_d^2)$;
 3: $T_d^3, SKV_{R\to DR} \leftarrow AD_2(p\mathcal{I}_r, T_r^2, M_5, M_6, M_7, S_2)$;
 4: Call $RSU_2(T_d^3, SKV_{R\to DR})$;
 5: procedure $AD_1()$
 6:     Pick $n_d^2 \in Z_q^*$;
 7:     $T_d^2 \leftarrow Time\_Stamp()$;
 8:     $M_1 \leftarrow b\ .\ H(\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_d^2 \parallel T_d^2)$; (Message creation)
 9:     $M_2 \leftarrow b\ .\ H(M_1 \oplus\ (p\mathcal{I}_d \parallel\ \mathcal{C}_d \parallel\ n_d^2 \parallel\ T_d^2))$;
10:     $S_1 \leftarrow H(\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_d^2 \parallel T_d^2) + H(M_1 \oplus (p\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_d^2 \parallel T_d^2)) + H(n_d \oplus \mathcal{I}_d)(mod\ q)$; (Signature creation)
11:     $M_3 \leftarrow (\mathcal{I}_d, M_1, M_2, S_1, T_d^2)$;
12:     Send the message $M_3$, on a public channel, to $RSU_1()$ ;
13: procedure $RSU_1(\mathcal{I}_d, M_1, M_2, S_1, n_d^2, T_d^2)$
14:     $T_r^2 \leftarrow Time\_Stamp()$;
15:     if $|T_r^2 - T_d^2| \leq \Delta T$ then
16:         if $b\ .\ S_1 == M_1 + M_2 + pK_d(mod\ q)$ then
17:             Pick $n_r^2 \in Z_q^*$;
18:             $M_4 \leftarrow H(\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_d^2 \parallel T_d^2) = M_1/q(mod\ q)$;
19:             $M_5 \leftarrow b\ .\ H(p\mathcal{I}_d \parallel n_r^2 \parallel p\mathcal{I}_r \parallel T_r^2)$;
20:             $M_6 \leftarrow M_1 \oplus H(p\mathcal{I}_r \parallel p\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_r^2 \parallel T_r^2)$; (Message creation)
21:             $M_7 \leftarrow M_2 \oplus H(p\mathcal{I}_r \parallel p\mathcal{I}_d \parallel S_1 \parallel \mathcal{C}_r \parallel n_r^2 \parallel T_d^2)$;
22:             $SK_{R\to D} \leftarrow H((M_5 \oplus M_7) \parallel H(p\mathcal{I}_r \parallel p\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_r^2 \parallel T_r^2))$; (Session key creation)
23:             $S_2 \leftarrow H(p\mathcal{I}_d \parallel n_r^2 \parallel p\mathcal{I}_r \parallel T_r^2) + H(p\mathcal{I}_r \parallel p\mathcal{I}_d \parallel S_1 \parallel \mathcal{C}_r \parallel n_r^2 \parallel T_d^2) + H(n_r \oplus \mathcal{I}_d \oplus \mathcal{I}_r)(mod\ q)$;
24:             $M_8 \leftarrow (p\mathcal{I}_r, T_r^2, M_5, M_6, M_7, S_2)$;
25:             Send the message $M_8$, on a public channel, to $AD_2()$;
26:         else
27:             Reject and report security threat;
28:     else
29:         Reject and report security threat;
```

Algorithm 3 `Procedures necessary for Algorithm` 2

```
1:  procedure $AD_2(p\mathcal{I}_r, T_r^2, M_5, M_6, M_7, S_2)$
2:      $T_d^3 \leftarrow Time\_Stamp()$;
3:      if $|T_d^3 - T_r^2| \leq \Delta T$ then
4:          if $b \,.\, S_2 == M_5 + b \,.\, (M_2 \oplus M_7) + pK_r(mod\ q)$ then
5:              $M_9 \leftarrow M_1 \oplus M_6$;
6:              $SK_{D\rightarrow R} \leftarrow H((M_6 \oplus M_7) \| M_9)$; (Session key creation)
7:              $SKV_{D\rightarrow R} \leftarrow H(T_d^3 \| SK_{D\rightarrow R})$;
8:              $M_{10} \leftarrow (T_d^3, SKV_{D\rightarrow R})$;
9:              Send the message $M_{10}$, on a public channel, to $RSU_2()$;
10:         else
11:             Reject and report security threat;
12:     else
13:         Reject and report security threat;
14: procedure $RSU_2(T_d^3, SKV_{R\rightarrow DR})$
15:     $T_r^3 \leftarrow Time\_Stamp()$;
16:     if $|T_r^3 - T_d^3| \leq \Delta T$ then
17:         $SKV_{R\rightarrow D} \leftarrow H(T_d^3 \| SK_{R\rightarrow D})$;
18:         if $SKV_{R\rightarrow D} == SKV_{D\rightarrow R}$ then
19:             Start a session with the key $SK_{R\rightarrow D}$;
20:         else
21:             Reject and report security threat;
22:     else
23:         Reject and report security threat;
```

the message from d, r executes $RSU_2()$. The entity r gets the current time stamp into T_r^3 in Line 15. Then in Line 16, r checks the validity of the received time stamp using the condition $|T_r^3 - T_d^3| \leq \Delta T$. In case the time stamp is valid, r wraps its already created session key in the received time stamp to get $SKV_{R\rightarrow D}$ using $H(T_d^3 \| SK_{R\rightarrow D})$, in Line 17. Then, r checks in Line 18 if identical wrapped keys are obtained via the condition $SKV_{R\rightarrow D} == SKV_{D\rightarrow R}$. If so, a new session is successfully started with the key $SK_{R\rightarrow D}$.

2) VH-AD authentication: This authentication creates secret session keys necessary for communications between a vehicle v and an AD d. The authentication details are presented in Algorithms 4 and 5. The vehicle initiates the authentication process as shown in the first step of Algorithm 4 which calls the method $VH_1()$. The method $VH_1()$ sends the message $(p\mathcal{I}_v, M_{11}, M_{12}, S_3, T_v^2)$ to d via a public channel. To react to the message from v, d executes AD_3. Then d sends the message $(p\mathcal{I}_d, T_d^4, M_{15}, S_4, pK_d)$ to the drone v on a public channel.

Upon receiving the message from d, v executes $VH_2()$ of Algorithm 5. Then, in Line 8, v sends back to d the message (M_{17}, T_v^3), on a public channel. In response to the message from v, d executes $AD_4()$.

Algorithm 4 `(Vehicle, Drone) Key Agreement`

```
Input: A vehicle $v$ and a drone $d$.
  Steps:
1:  $p\mathcal{I}_v, M_{11}, M_{12}, S_3, T_v^2 \leftarrow VH_1()$;
2:  $p\mathcal{I}_d, T_d^4, M_{15}, S_4, pK_d \leftarrow AD_3(p\mathcal{I}_v, M_{11}, M_{12}, S_3, T_v^2)$;
3:  $M_{16}T_v^3 \leftarrow VH_2(p\mathcal{I}_d, T_d^4, M_{15}, S_4, pK_d)$;
4:  Call $AD_4(M_{16}T_v^3)$;
5:  procedure $VH_1()$
6:      Pick $n_v^2 \in Z_q^*$;
7:      $T_v^2 \leftarrow Time\_Stamp()$;
8:      $M_{11} \leftarrow H(p\mathcal{I}_v \| n_v^2 \| pK_v \| T_v^2)$; (Message creation)
9:      $M_{12} \leftarrow b \,.\, H(M_{11} \oplus (\mathcal{C}_v \| pK_v))$;
10:     $S_3 \leftarrow H(M_{11} \oplus (\mathcal{C}_v \| pK_v)) + H(pK_v \oplus (p\mathcal{I}_v \| pK_v \| T_v^2)) + H(n_v \oplus \mathcal{I}_v)(mod\ q)$; (Signature creation)
11:     $M_{13} \leftarrow (p\mathcal{I}_v, M_{11}, M_{12}, S_3, T_v^2)$;
12:     Send the message $M_{13}$, on a public channel, to $AD_3()$ ;
13: procedure $AD_3(p\mathcal{I}_v, M_{11}, M_{12}, S_3, T_v^2)$
14:     $T_d^4 \leftarrow Time\_Stamp()$;
15:     if $|T_d^4 - T_v^2| \leq \Delta T$ then
16:         if $b \,.\, S_3 == M_{12} + b \,.\, (H(pK_v \oplus (p\mathcal{I}_v \| pK_v \| T_v^2))) + pK_v(mod\ q)$ then
17:             Pick $n_d^3 \in Z_q^*$;
18:             $M_{14} \leftarrow H(p\mathcal{I}_d \| \mathcal{C}_d \| n_d^3 \| pK_d \| T_d^4)$;
19:             $M_{15} \leftarrow b \,.\, H(M_{14} \oplus pK_d)$;
20:             $SK_{D\rightarrow V} \leftarrow H(M_{14} \oplus pK_d) \,.\, M_{12}$; (Session key creation)
21:             $S_4 \leftarrow H(M_{14} \oplus pK_d) + H(p\mathcal{I}_v \| p\mathcal{I}_d \| pK_d \| SK_{D\rightarrow V} \| T_d^4) + H(n_d \oplus \mathcal{I}_d)(mod\ q)$;
22:             $M_{16} \leftarrow (p\mathcal{I}_d, T_d^4, M_{15}, S_4, pK_d)$;
23:             Send the message $M_{16}$, on a public channel, to $VH_2()$;
24:         else
25:             Reject and report security threat;
26:     else
27:         Reject and report security threat;
```

E. V2V communication

We consider the two possible scenarios for V2V communication: the first one treats V2V communication for vehicles in the same geographical zone, and the second one treats V2V communication for vehicles in different geographical zones. In these scenarios, we employ the established session keys, drones, and roadside units. The security of the communication is guaranteed by involving session keys of more than one party in the communication process. Therefore, having an adversary vehicle does not jeopardize the communication's security.

For vehicles in the same geographical zone, the communication is achieved as follows. Let us denote the region's AD and two vehicles in the region as D_r, V_1, and V_2, respectively. Suppose also that session keys $SKey_1$ and $SKey_2$ are the ones established between D_r and V_1 and V_2, respectively. When V_1

Algorithm 5 Procedures necessary for Algorithm 4

```
Input: A vehicle $v$ and a drone $d$.
    Steps:
procedure $VH_2(p\mathcal{I}_d, T_d^4, M_{15}, S_4, pK_d)$
    $T_v^3 \leftarrow \textit{Time\_Stamp}()$;
    if $|T_d^3 - T_r^2| \leq \Delta T$ then
        $SK_{V\rightarrow D} \leftarrow M_{15}\ .\ M_{11}$;
        if $b\ .\ S_4 == M_{15} + b\ .\ H(p\mathcal{I}_v \parallel p\mathcal{I}_d \parallel pK_d \parallel SK_{V\rightarrow D}) + pK_d(mod\ q)$ then
            $M_{17} \leftarrow H(SK_{V\rightarrow D} \parallel T_v^3)$;
            $M_{18} \leftarrow (M_{17}, T_v^3)$;
            Send the message $M_{18}$, on a public channel, to $AD_4()$;
        else
            Reject and report security threat;
    else
        Reject and report security threat;
procedure $AD_4(M_{17}T_v^3)$
    $T_d^5 \leftarrow \textit{Time\_Stamp}()$;
    if $|T_d^5 - T_v^3| \leq \Delta T$ then
        $M_{19} \leftarrow H(SK_{D\rightarrow V} \parallel T_v^3)$;
        if $M_{19} == M_{17}$ then
            Start a session with the key $SK_{D\rightarrow V}$;
        else
            Reject and report security threat;
    else
        Reject and report security threat;
```

requests communication with V_2, D_r encrypts $SKey_1$ with $SKey_2$ and sends the result to V_2. Hence, V_2 becomes able to securely communicating with V_1 via its session key.

For vehicles in different geographical zones, secure communication is achieved as follows. Suppose that vehicles V_1 and V_2 are in zones Z_1 and Z_2, respectively. Suppose that R_1 and R_2 are two roadside units managing Z_1 and Z_2, respectively. Suppose that D_1 and D_2 are the two managing drones of Z_1 and Z_2, respectively. When V_1 requests communication with V_2, the drones D_1 and D_2 exchange the session keys that they established with R_1 and R_2, respectively. The exchange is done via R_1 and R_2. This results in creating a communication network. Afterward, the vehicles encrypt their session keys using the session key between D_1 and D_2. Finally, the vehicles exchange the encrypted keys. It is worth noting that the zones Z_1 and Z_2 have other roadside units, R_1' and R_2'. These units can be realized as backup units for R_1 and R_2, and hence can provide a backup connection plan for vehicle communications in zones Z_1 and Z_2.

F. Dynamic Vehicle Addition

The steps executed by TA for deploying a new vehicle v^{new} to a running system is similar to that of registering a new vehicle. TA starts by reading from the smart card of v^{new} an ID, $\mathcal{I}^{v^{new}}$, and a password $\mathcal{P}^{v^{new}}$ and continues by generating a random secret $n^{v^{new}} \in Z_q^*$ which is used to calculate the key $pK^{v^{new}}$ as $bH(n^{v^{new}} \oplus \mathcal{I}^{v^{new}})$, an auxiliary ID $p\mathcal{I}^{v^{new}}$, and a password $p\mathcal{P}^{v^{new}}$, as $H(\mathcal{I}^{v^{new}} \parallel K_{TA} \parallel n^{v^{new}})$ and $H(\mathcal{P}^{v^{new}} \parallel \mathcal{I}^{v^{new}} \parallel K_{TA} \parallel n^{v^{new}})$, respectively. Then TA continues as in the registration phase.

G. Blockchain Management

Blockchain technology allows for the creation of a list of records over a secure decentralized network. Thanks to this technology, the TA can create a list of records of the key-establishment phases to improve the security of the overall network. Furthermore, as the information is decentralized, it removes the possibility of an attacker being able to corrupt or compromise the TA, as each transaction must be approved by a decentralized network. The only entity able to access the blockchain is the TA. We hence notice that a feasible alternative would be the implementation of a distributed database not relying on blockchain technology. However, blockchain is more suitable to our problem. Indeed, distributed databases do not generally implement solutions against Byzantine attacks and mostly focus on failures of nodes [29]. This may not be sufficient in a scenario where a malicious user injects false information, and may lead to successful false data injection. Furthermore, recording data in terms of transaction and securing them via cryptographic proof provides both security against manipulation, and traceability. Lastly, distributed databases generally rely on a central machine to take decisions on the system behavior [29]. This is different from a decentralized system, where each node independently acts based on local information. This removes the vulnerability related to a single point of failure.

The login and authentication phases of BDIVE allow for the creation of special key tracking transactions, that we denote as $\mathcal{KT}$ transactions. In this section, we give a precise definition of $\mathcal{KT}$ and illustrate how to utilize the blockchain technology to store these transactions in a Peer-to-Peer (P2P) Cloud Server (CS) network.

During the Login and Authentication phases illustrated above, different system entities establish session keys. This allows for mutual authentications among system entities using stored credentials and certificates and has to be completed before exchanging real-time sensitive data. Our proposed protocol, BDIVE, defines the following types of $\mathcal{KT}$ transactions issued by RSUs and drones.

1) $\mathcal{T}_{S1}^{C}$: this transaction contains meta-data related to the creation of a session key between an AD and a RSU. This includes the IDs of the involved system entities, timestamp of key creation, and can be complemented by other relevant information.
2) $\mathcal{T}_{S1}^{U}$: this transaction records meta-data related to a use case of the already established session keys among ADs and RSUs.
3) $\mathcal{T}_{S1}^{F}$: this transaction contains meta-data related to a failed creation of a session key between an AD and a RSU.
4) $\mathcal{T}_{S2}^{C}$: this transaction is similar to $\mathcal{T}_{S1}^{C}$, except that it focuses on session creation between an AD and a vehicle.

5) $\mathcal{T}_{S2}^{U}$: this transaction is similar to $\mathcal{T}_{S1}^{U}$, except that it focuses on session creation between an AD and a vehicle.
6) $\mathcal{T}_{S2}^{F}$: this transaction is similar to $\mathcal{T}_{S1}^{F}$, except that it focuses on session creation between an AD and a vehicle.

Suppose that an adversary $\mathcal{A}$ deploys a malicious vehicle or AD to the system. Suppose also that the malicious deployed entity (drone/vehicle) manages to create a valid session key or to get an already created one from another authorized entity. Then, thanks to the blockchain, all the activities of the malicious entity will be recorded using our $\mathcal{KT}$ transactions. Consequently, the TA can discover malicious activities by inspecting the $\mathcal{KT}$ transactions.

TABLE II: Block structure.

Block Header	Block Payload
$\mathcal{I}_T$	Encrypted $\mathcal{KT}$ transactions list
Timestamp	Block hash
Previous block hash	Signature on Block hash
Merkle tree root	Commit message pool
Previous public key	

The $\mathcal{KT}$ transactions related to system entities of different geographical areas shall be confidential and private. Therefore, it makes sense to insert these transactions into a private blockchain managed by the P2P CS network. It is worth noting that vehicles, RSUs, and drones have limited computational resources. Therefore, it is convenient to delegate the task of creating the blockchain transactions to the TA. Therefore, we suppose that TA collects information constituting $\mathcal{KT}$ transactions from different system entities and then builds the transactions and necessary calculations which include:

1) $\mathcal{I}_T$: a transaction ID.
2) $\mathcal{L}_E$: a list of IDs of system entities contributed to the information.
3) $\mathcal{E}_T$: an encryption form of the transaction using the public key of TA.
4) $\mathcal{H}_T$: a hash of the transaction.
5) $\mathcal{E}_A$: an elliptic curve digital signature algorithm that uses the private key TA on the transaction.

We recall that each transaction follows one of the $\mathcal{KT}$ transaction types. The TA sends the established transactions to a P2P CS network. This involves selecting a leader in the P2P cloud server using a leader selection algorithm [7]. Hence, for the received transactions, a cloud server begins a transaction pool. When the number of elements in the pool reaches a certain threshold, a new block is formed, following the block structure of Table II. The final step is to add the new block to the blockchain: a consensus algorithm is needed for block verification as part of the block addition process (done by CS to the already existing blockchain). For BDIVE, we apply the PBFT consensus algorithm [27]. We describe this process in detail in Algorithm 6.

We assume that the consensus algorithm is a voting distributed algorithm that is executed on distributed nodes of cloud servers. We also assume that the algorithm utilizes an incentive technique to increase the algorithm's scalability. Hence, in response to a valid vote reply, a participant cloud server gets an incentive that is augmented with every valid vote reply. Using the incentive values a subset of nodes can be selected for executing the PBFT consensus algorithm.

Algorithm 6 Block Verification and Addition Consensus

```
Input: A list $L_t$ of $\mathcal{KT}$ transactions, a list $L_c$ of $n_c$ cloud
servers of P2P nodes contributing to the process, and a
blockchain, $B$.
  Steps:
  $p \leftarrow \frac{3*N}{4}$;
  $validVote \leftarrow 0$;
  if $|L_t| \geq threshold$ then
      $lcs \leftarrow$ FixLeader($L_c$);
      $bl \leftarrow$ BlockCreation($L_t$);
      if BlockVerification($lcs, bl$) then
          for each $c \in L_c$ do
              if BlockVerification($c, bl$) then
                  Send(valid-vote);
                  $validVote \leftarrow validVote + 1$;
          if $validVote \geq p$ then
              AddBlock($B, bl$);
              Broadcast a commitment message;
```

V. Security Analysis

In this section, we analyze the security of our proposed protocol BDIVE in terms of potential threats. In particular, we prove in Section V-A the validity of mutual authentication in BDIVE using Burrow–Abadi–Needham (BAN) Logic [30]). We then prove in Section V-B the security resilience of BDIVE against several attacks.

A. Formal security analysis via BAN logic

In this section, we develop a logic for formal authentication of BDIVE. The logic relies on the well-established formal authentication logic, i.e., BAN logic [30] which is a group of inference rules to analyze and check the validity of information communicated over networks. In particular, the BAN logic allows us to assess whether the network is resilient towards alteration and sniffing of the exchanged information. The logic typically aims at proving certain authentication goals using assumptions, rules, and postulates. For BDIVE the BAN logic ensures that the different system entities share relevant session keys.

Our formalization uses the following notation.

1) E, E_1, E_2: system entities; S, S_1, S_2: statements; K: encryption key.
2) $E \vdash S$: E believes S.
3) $E \ll S$: E received S.
4) $E \gg S$: E sent S.
5) $E \propto S$: E controls S.
6) $\mathcal{F}(S)$: S is fresh.
7) $E_1 \leftrightarrow^{k} E_2$: E_1 and E_2 share K to communicate.
8) $E < K$: E has public key K.

9) $E < K^{-1}$: E has private key K.
10) $E_1 < S > E_2$: S is a secret shared only between E_1 and E_2.
11) $\{S\}_K$: S is encrypted with K.
12) $(S)_K$: S is hashed with K.
13) $S_1.S_2$: S_1 is combined with S_2.
14) (S_1, S_2): message composed of messages S_1 and S_2.
15) $SK_{E_1 \to E_2}$: Session key shared between system entities E_1 and E_2.

Our logic relies on the following BAN inference rules.

1) Message semantics (MS): $\dfrac{E_1 \vdash E_1 \leftrightarrow^k E_2 \quad E_1 \ll \{S\}_K}{E_1 \vdash E_2 \gg S}$;
   This rule reads as follows: if E_1 believes that it shares with E_2 a key K and it received message S encrypted with K, then E_1 believes that E_2 sent S.
2) Nonce freshness (NF): $\dfrac{E_1 \vdash \mathcal{F}(S) \quad E_1 \vdash E_2 \gg S}{E_1 \vdash E_2 \vdash S}$;
   This rule reads as follows: if E_1 believes that E_2 sent a fresh message S, then E_1 believes that E_2 believes S.
3) Message control (MC): $\dfrac{E_1 \vdash E_2 \propto S \quad E_1 \vdash E_2 \vdash S}{E_1 \vdash S}$;
   This rule reads as follows: if E_1 believes that E_2 controls and believes S, then E_1 believes S.
4) Message freshness (MF): $\dfrac{E \vdash \mathcal{F}(S_1)}{E \vdash \mathcal{F}(S_1, S_2)}$;
   This rule reads as follows: if E believes that S_1 is a fresh message, then E believes also that any message composed of S_1 and any other message S_2 is fresh too.
5) Belief (B): $\dfrac{E \vdash S_1 \quad E \vdash S_2}{E \vdash (S_1, S_2)}$;
   This rule reads as follows: if E believes the messages S_1 and S_2, then E believes also any message that is composed of S_1 and S_2.
6) Session key (SK): $\dfrac{E_1 \vdash \mathcal{F}(S) \quad E_1 \vdash E_2 \vdash S}{E_1 \vdash E_1 \leftrightarrow^k E_2}$;
   This rule reads as follows: if E_1 believes that E_2 believes a fresh message S, then E_1 believes that it shares a communication key K with E_2.

As per Algorithms 2, 3, 4, and 5, the goals that we would like to prove for BDIVE are the following.

- Goal 1: $d \vdash d \leftrightarrow^{SK_{D \to R}} r$; Goal 2: $d \vdash r \vdash d \leftrightarrow^{SK_{D \to R}} r$;
- Goal 3: $r \vdash r \leftrightarrow^{SK_{D \to R}} d$; Goal 4: $r \vdash d \vdash r \leftrightarrow^{SK_{D \to R}} d$;
- Goal 5: $v \vdash v \leftrightarrow^{SK_{V \to D}} d$; Goal 6: $v \vdash d \vdash v \leftrightarrow^{SK_{V \to D}} d$;
- Goal 7: $d \vdash d \leftrightarrow^{SK_{V \to D}} v$; Goal 8: $d \vdash v \vdash d \leftrightarrow^{SK_{V \to D}} v$;

We make the following assumptions about the initial states of BDIVE.

A1: $\mathcal{F}(n_d^2)$; A2: $\mathcal{F}(T_d^2)$; A3: $\mathcal{F}(T_r^2)$; A4:$\mathcal{F}(n_r^2)$;
A5: $\mathcal{F}(T_d^3)$; A6: $\mathcal{F}(n_v^2)$; A7: $\mathcal{F}(T_v^2)$; A8:$\mathcal{F}(n_d^3)$;
A9: $\mathcal{F}(T_d^4)$; A10: $\mathcal{F}(T_v^3)$; A11: $d \vdash d \leftrightarrow^{H(n_d \oplus \mathcal{I}_d)} r$;
A12: $r \vdash d \propto n_d^2$; $A13 : d \vdash r \propto n_r^2$;

We idealize the communicated messages among system entities as follows.

- $M_3 : \{\mathcal{I}_d, M_1, M_2, S_1, T_d^2 : \{n_d^2\}_{H(n_d \oplus \mathcal{I}_d)}\}$;
- $M_8 : \{p\mathcal{I}_r, T_r^2, M_5, M_6, M_7, S_2 : \{n_r^2\}_{H(n_d \oplus \mathcal{I}_d)}\}$;
- $M_{13} : \{p\mathcal{I}_v, M_{11}, M_{12}, S_3, T_v^2 : \{n_v^2\}_{H(n_d \oplus \mathcal{I}_d)}\}$;
- $M_{16} : \{p\mathcal{I}_d, T_r^4, M_{15}, S_4, p\mathcal{K}_d : \{n_d^3\}_{H(n_d \oplus \mathcal{I}_d)}\}$;

Idealizing M_{10} and M_{18} is not needed in the model proofs. This is because they are final messages (with session keys) and their content is not used to build further messages.

1) Proofs: The idea of the prove is to move gradually from assumptions to the requirements using the BAN inference rules. The proof of the goals $1-4$ stated above is as follows.

P1 By M_3, $r \ll \{\mathcal{I}_d, M_1, M_2, S_1, T_d^2 : \{n_d^2\}_{H(n_d \oplus \mathcal{I}_d)}\}$. The is so as r received this message.
P2 From MS rule, P1, and A11, we have: $r \vdash d \gg n_d^2$; The is so as r and d share a key that was used to encrypt M_3.
P3 From NF rule, P2, and A1, we have: $r \vdash d \vdash n_d^2$;
P4 From MC rule, P3, and A12, we have: $r \vdash n_d^2$;
P5 From SK rule, P3, and A1, we have: $d \vdash d \leftrightarrow^{SK_{D \to R}} r$. **This proves Goal 1**.
P6 From NF rule, P5, and A1, we have: $d \vdash r \vdash d \leftrightarrow^{SK_{D \to R}} r$. **This proves Goal 2**.
P7 By M_8, $d \ll \{p\mathcal{I}_r, T_r^2, M_5, M_6, M_7, S_2 : \{n_r^2\}_{H(n_d \oplus \mathcal{I}_d)}\}$.
P8 From MS rule, P7, and A11, we have: $d \vdash r \gg n_r^2$;
P9 From NF rule, P8, and A4, we have: $d \vdash r \vdash n_r^2$;
P10 From MC rule, P9, and A13, we have: $d \vdash n_r^2$;
P11 From SK rule, P9, and A4, we have: $r \vdash r \leftrightarrow^{SK_{D \to R}} d$. **This proves Goal 3**.
P12 From NF rule, P11, and A4, we have: $r \vdash d \vdash r \leftrightarrow^{SK_{D \to R}} d$. **This proves Goal 4**.

Similar reasoning proves the remaining goals.

B. Attacks Resilience

In this section, we show that our proposed protocol BDIVE is resilient to many popular attacks in IoV systems.

Theorem 1. *BDIVE is secure against Replay attack and man-in-the-middle attacks; and ESL (Ephemeral Secret Leakage) Attacks.*

Proof. 1) Suppose that an adversary, $\mathcal{A}$ obtains messages M_3, M_8, and M_{10} during the drone-RSU authentication, and M_{13}, M_{16}, and M_{18} during the vehicle-drone authentication phase. It is worth noting that timestamps and/or random secrets are included in these messages. Moreover, the timestamps are checked upon message reception. The failure of a timestamp check leads to message discarding. Hence it is not possible for $\mathcal{A}$ to replay previous messages. Therefore BDIVE is secure against replay attacks. Now we assume that $\mathcal{A}$ tampers the content of messages and resents them to legitimate entities. BDIVE uses four signatures, S_1, S_2, S_3, and S_4 based on $n_d^2, n_r^2, n_v^2, n_d^3$, respectively, which are private credentials to different system entities. Therefore tampered signatures can be instantly recognized by BDIVE at reception via signature verification. Hence BDIVE is secure against man-in-the-middle attacks.

2) In the drone-RSU authentication phase, a RSU, r, generates a shared session key with one of its corresponding drone, d. The key is calculated as $SK_{R\to D} = H((M_5 \oplus M_7) \parallel H(p\mathcal{I}_r \parallel p\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_r^2 \parallel T_r^2))$, where $M_7 = M_2 \oplus H(p\mathcal{I}_r \parallel p\mathcal{I}_d \parallel S_1 \parallel \mathcal{C}_r \parallel n_r^2 \parallel T_d^2)$ and $M_5 = b \,.\, H(p\mathcal{I}_d \parallel n_r^2 \parallel p\mathcal{I}_r \parallel T_r^2)$. The drone, d, also generates the shared session key as $SK_{D\to R} = H((M_6 \oplus M_7) \parallel M_9)$, where $M_9 = M_1 \oplus M_6$ and $M_6 = M_1 \oplus H(p\mathcal{I}_r \parallel p\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_r^2 \parallel T_r^2)$. The session key calculation relies on temporal secrets (n_r^2, T_r^2, and T_d^2) and permanent secrets ($p\mathcal{I}_r, p\mathcal{I}_d$, and $\mathcal{C}_d$). In the vehicle-drone authentication phase, a drone d generates a shared session key with a vehicle, v. The key is calculated as $SK_{D\to V} = H(M_{14} \oplus pK_d) \,.\, M_{12}$, where $M_{12} = b \,.\, H(M_{11} \oplus (\mathcal{C}_v \parallel pK_v))$ and $M_{14} = H(p\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_d^3 \parallel pK_d \parallel T_d^4)$. The vehicle v generates the shared session key as $SK_{V\to D} = M_{15} \,.\, M_{11}$, where $M_{15} = b \,.\, H(M_{14} \oplus pK_d)$ and $M_{11} = H(p\mathcal{I}_v \parallel n_v^2 \parallel pK_v \parallel T_v^2)$. The session key calculation relies on temporal secrets (n_v^2, T_v^2, and T_d^4) and permanent secrets ($p\mathcal{I}_d, p\mathcal{I}_v$, and $\mathcal{C}_d$). There are two possible scenarios for consideration here concerning the the adversary:
   - The adversary compromise only the temporal secrets ($n_r^2, T_r^2, n_v^2, T_v^2, T_d^4$, and T_d^2). In this, the adversary can not move on and compromise the session keys without compromising the permanent secrets ($p\mathcal{I}_r, p\mathcal{I}_d, p\mathcal{I}_v$, and $\mathcal{C}_d$).
   - The adversary compromise only the permanent secrets ($p\mathcal{I}_r, p\mathcal{I}_d, p\mathcal{I}_v$, and $\mathcal{C}_d$). In this, the adversary can not move on and compromise the session keys without compromising the temporal secrets ($n_r^2, T_r^2, n_v^2, T_v^2, T_d^4$, and T_d^2).

   Hence according to assumptions of the CK-adversary model, BDIVE is secure against ESL attacks.

□

Theorem 2. *BDIVE is resilient against privileged insider, physical system entity capture, impersonation, DoS (Denial-of-Service), anonymity, and untraceability attacks.*

Proof.

1) The TA registers drones, roadside units, vehicles, and cloud servers. This involves calculating and equipping these parties with their secret credentials. Therefore, no secrets are sent over a public channel. Hence, a privileged insider can not gain knowledge of secret credentials and hence BDIVE is secure against privileged insider attacks. Suppose now that an adversary $\mathcal{A}$ captures a system entity such as a vehicle. Suppose also $\mathcal{A}$ extracts the vehicle credentials by applying power analysis techniques [25]. As a result, the credentials $(\mathcal{I}_v, p\mathcal{I}_v, \mathcal{P}_v, p\mathcal{P}_v, \mathcal{C}_v, n_v, pK_v, \mathcal{H}(\cdot), \mathcal{E}(q_1, q_2), b)$ can be exposures. However, leaking these credentials does not threaten the security of the vehicle's communication with other secure system parties. This is so as the leaked credentials are unique to the compromised vehicle. Similarly, secure system entities can communicate securely with other secure parties although $\mathcal{A}$ knows the vehicle credentials. Therefore, BDIVE is resilient against physical system entity capture attacks.
2) Suppose that an adversary, $\mathcal{A}$ obtains communicated messages M_3, M_8, and M_{10} drone-RSU authentication, and M_{13}, M_{16}, and M_{18} during vehicle-drone authentication phase. We discuss the following impersonation cases.
   - Suppose $\mathcal{A}$ tries to attack the drone via impersonating the RSU and fabricating $M_8 = (p\mathcal{I}_r, T_r^2, M_5, M_6, M_7, S_2)$. This requires $\mathcal{A}$ to generate $M_6 = M_1 \oplus H(p\mathcal{I}_r \parallel p\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_r^2 \parallel T_r^2)$. However M_6 calculations require the timestamp T_r^2, random secret n_r^2, and permanent $p\mathcal{I}_r, p\mathcal{I}_d$, and $\mathcal{C}_d$. Therefore,BDIVE is secure RSU impersonation attacks.
   - Suppose $\mathcal{A}$ tries to attack the vehicle via impersonating the drone and fabricating $M_{16} = (p\mathcal{I}_d, T_d^4, M_{15}, S_4, pK_d)$. This requires $\mathcal{A}$ to generate $M_{15} = b \,.\, H(M_{14} \oplus pK_d)$ which in turn requires calculating $M_{14} = M_{14} \leftarrow H(p\mathcal{I}_d \parallel \mathcal{C}_d \parallel n_d^3 \parallel pK_d \parallel T_d^4)$. However M_{14} calculations require the time timestamp T_d^4, random secret n_d^3, and permanent $p\mathcal{I}_d$ and $\mathcal{C}_d$. Therefore,BDIVE is secure against drone impersonation attacks.

   All in all, BDIVEis resilient against all types of impersonation attacks.
3) It is possible to detect many messages from an adversary $\mathcal{A}$ The usage is due to the insertion of current timestamps of these messages. Hence the adversary can not abuse the resources of system entities because these resources are mainly consumed by lightweight cryptographic like ECC and hash computations. Therefore BDIVE is secure against DoS attacks. Recall that communicated messages are M_3, M_8, and M_{10} in drone-RSU authentication, and M_{13}, M_{16}, and M_{18} during vehicle-drone authentication phase. These messages rely on temporal secrets auxiliary identities, not real ones, of system entities. Furthermore, the messages are made unique via the used random secrets. Therefore $\mathcal{A}$ can not recognize or trace parties communicating in successive sessions. Hence BDIVE maintains anonymity and untraceability features and resists their attacks.

□

VI. Evaluation

In this section, we assess the performance of BDIVE via extensive experiments and compare BDIVE to related state-of-the-art protocols [4], [31], [3], [32]. We perform the experiments on a Dell (Vostro) Intel(R) Core(TM) i7-3612 QM CPU @ 2.10 GHz, 8.00 GB RAM on Windows 10 (64-bits) OS. We make the result files obtained via the different simulations tools used to evaluate BDIVE available in a repository[1] .

Considering both active and passive attacker models, we utilized AVISPA [33] to formally prove the resilience of

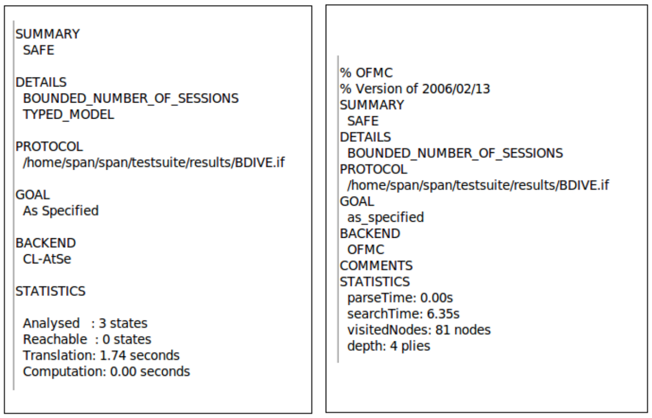
[1] https://github.com/maelzawawy/BDIVE

BDIVE against man-in-the-middle and replay attacks. This is presented in Section VI-A. Moreover, we compared BDIVE against state-of-the-art protocols based on different performance metrics critical in the IoV context. These metrics include communication cost, energy consumption, computation costs, and security and functionality features. The comparison results prove the better performance of BDIVE compared to the related protocols. These are presented in Section VI-B. To prove its practicality, we also implement BDIVE using the well-known tool for networking simulation, Omnetpp[2] with its Castalia[3] simulator. The details of the Omnetpp implementation are presented in Section VI-C.

A. FORMAL SECURITY VERIFICATION

One of the common tools for simulating and checking the safety of security protocols is AVISPA [33]. AVISPA relies on High-Level Protocol Specification Language (HLPSL) and evaluates a protocol as inconclusive, safe, or unsafe. HLPSL is based on temporal logic and its code is built on roles for state transition. Moreover, each HLPSL program has session and environment roles. AVISPA considers the DY threat model [22]. Therefore, AVISPA checks protocols against man-in-the-middle and replay attacks. This includes active and passive adversary communication.

AVISPA offers four different backends: SAT-based Model Checker (SATMC), On The Fly Model Checker (OFMC), Constraint Logic-based Attack Searcher (CL-AtSe), and Tree Automate based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). We develop the registration and authentication phases of BDIVE in AVISPA using a role in the code for each system entity. Afterward, we completed the code simulation using SPAN[4] (Security Protocol ANimator for AVISPA). As the bitwise XOR operation is not included in SATMC and TA4SP backends, we completed our simulation using OFMC and CL-AtSe backends. Figure 2 shows the simulation results which confirm the safety of BDIVE against both men-in-the-middle and replay attacks. The files of simulation results are available in a repository[5].



```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/BDIVE.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 3 states
  Reachable  : 0 states
  Translation: 1.74 seconds
  Computation: 0.00 seconds
```

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/BDIVE.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 6.35s
  visitedNodes: 81 nodes
  depth: 4 plies
```

Fig. 2: Results of Simulating BDIVE in AVISPA.

B. Functionality Features and Costs Comparison

In this section, we compare our proposed protocol against existing state-of-the-art IoV and VANET authentication protocols [4], [31], [3], [32]. The comparison considers computational and communication costs and security features, including guessing, man-in-the-middle, replay, privileged-insider, ESL, traceability, and anonymity attacks. Table III shows that existing protocols neither support drone assistance nor satisfy-basic security requirements. On the other hand, BDIVE fills these gaps providing a secure solution for today's and future IoV networks. Attributes 3 and 4 of the table are illustrated in Section IV-E.

TABLE III: Comparing functionality characteristics of BDIVE against the state-of-the art protocols.

#	Functionality Attribute	[4]	[31]	[3]	[32]	**BDIVE**
1	Supporting drone assistance.	×	×	×	×	✓
2	Supporting Blockchain usage.	×	×	×	×	✓
3	Providing backup connection plan for vehicle communication.	×	×	×	×	✓
4	Providing backup entities for roadside units.	×	×	×	×	✓
5	Traceability and anonymity.	×	×	✓	✓	✓
6	Resilience to privilege-insider and ESL attacks.	×	×	×	✓	✓
7	Resilient to replay and guessing attacks.	✓	✓	✓	✓	✓
8	Resilient man-in-the-middle attacks.	✓	×	✓	✓	✓

The symbol ✓ (×) denotes that the corresponding protocol supports (does not support) the corresponding attribute.

TABLE IV: Comparing computational cost of BDIVE against the state-of-the art protocols per session key.

Participant	[4]	[31]	[32]	BDIVE
Vehicle	$9 \times t_h$	$7 \times t_h$	$9 \times t_h$	$7 \times t_h$
TA	$0 \times t_h$	$9 \times t_h$	$11 \times t_h$	$0 \times t_h$
Roadside unit	$11 \times t_h$	$4 \times t_h$	$5 \times t_h$	$10 \times t_h$
Comparable Total [ms]	$20 \times t_h$ ≈ 3.64	$20 \times t_h$ ≈ 3.64	$25 \times t_h$ ≈ 4.55	$17 \times t_h$ ≈ 3.094
Drone	$-$	$-$	$-$	$\leq 8 \times t_h$

The symbol $-$ denotes that the corresponding protocol does not support the corresponding participant.

Following the state-of-the-art protocols, we consider the size of the hash output (SHA-256 hashing), the random nonce, the timestamp, and the identity as 256, 160, 32, and 160 bits, respectively. According to Table V, BDIVE consumes a communication cost of 2560 bits for communicating three messages, whereas the related protocols we compare to, Chuang et al. [4], Mohit et al. [31], and Lee et al. [32], consumes 2304 bits (three messages), 2528 bits (four messages), and 2592 bits (four messages), respectively. This proves that the communication costs of BDIVE are comparable to the related state-of-the-art protocols.

To compute the computation costs, we consider the login and authentication phases. We rely on the execution times

[2] https://omnetpp.org/

[3] https://github.com/boulis/Castalia

[4] http://www.avispa-project.org/

[5] https://github.com/maelzawawy/BDIVE

TABLE V: Comparing communication cost of BDIVE against the state-of-the art protocols per session key.

#	Parameter	[4]	[31]	[32]	**BDIVE**
1	No. of messages	3	4	4	**3**
2	No. of bits	2304	2528	2592	**2560**

reported in [32] for one-way hash map $H(\cdot)$, considering the SHA-256. The time is denoted by t_h. This time, for a vehicle, is 0.309 ms, whereas for TA and roadside unit is 0.055 ms. We use the average (i.e., 0.182 ms) of these timings in our calculations. While a vehicle in BDIVE executes seven hash maps, TA executes zero hash functions. While a drone in BDIVE executes eight hash maps, a roadside unit executes ten hash functions. The total computational cost for BDIVE is $17t_h$. Table IV presents the computational costs per system entity for both BDIVE and related existing authentication protocols. We note that [31] and [32] are better than BDIVE in terms of computational cost at the road-side units. However, BDIVE is better than or comparable to [31] and [32] in terms of computational cost at vehicles. Considering the mobility nature and limited computational resources of vehicles, and stationary nature and reasonable computational resources of road-side units, BDIVE is hence advantageous over state of the art protocols.

The energy consumption of BDIVE depends on the speed and size of the messages communicated by the protocol, especially messages involving vehicles and drones. It is worth noting that the communicated messages move in the physical protocol layer. The vehicle DSRC (Dedicated Short-Range Communication) executes these data transmissions according to IEEE 802.11p. The IEEE standards [32] for vehicle networks specify channel bandwidth, frequency, transmit power, and data rate ($\mathcal{R}_d$) as 10 MHz, 5.8 GHz, 25 dBm, and 6 Mbps, respectively. We consider two parameters to measure and compare BDIVE energy consumption: $\mathcal{E}_k$ and $\mathcal{E}_l$ [32]. While the first parameter measures the energy consumed during key generation, the second parameter measures the energy consumed during the login and authentication phases. The total energy and its aforementioned components are calculated as

$$\mathcal{E} = \mathcal{E}_k + \mathcal{E}_l, \quad \mathcal{E}_k = \mathcal{P}_c \times \mathcal{P}_{cpu}, \text{ and } \quad \mathcal{E}_l = \frac{\mathcal{M}_s \times \mathcal{P}_{cpu}}{\mathcal{R}_d}, \tag{1}$$

where $\mathcal{P}_c, \mathcal{P}_{cpu}$, and $\mathcal{M}_s$ are the total computational cost, the CPU maximum power, and the message size, respectively. $\mathcal{P}_{cpu}$ is 10.88 W for wireless networks [32].

TABLE VI: Comparing energy consumption cost of BDIVE against the state-of-the art protocols.

#	Parameter	[4]	[31]	[32]	**BDIVE**
1	$\mathcal{E}_k$ [mJ]	39.603	39.603	49.504	**33.662**
2	$\mathcal{E}_l$ [mJ]	4.177	4.584	4.70	**4.642**
3	$\mathcal{E}$ [mJ]	43.78	44.187	54.204	**38.304**

Table VI compares the energy consumption of BDIVE to that of related state-of-the-art protocols. It is clear that, concerning energy consumption cost, BDIVE is more efficient than related state-of-the-art protocols.

TABLE VII: Simulation configurations.

Network parameter	Value
Operating system	windows 10
Simulator	Omnetpp & Castalia
Network Simulation area	a road of $50m \times 200m$
No. of simulation scenarios	6
No. of system entities	70 and 100
No. of vehicles	66 and 96
No. of roadside units	2
Communication	802.15.4 MAC
Mobility model	LineMobilityManager
Vehicle mobility	15 mph and 25 mph
Drone mobility	20 mph and 30 mph
Routing Protocol	MultipathRingsRouting
Channel bandwidth	20 MHz
Noise Bandwidth	194 MHz
Data rate	250 KBPS
Modulation Type	PSK
Noise Floor	-100 DBM

C. Practical Perspective

To confirm the practical feasibility of BDIVE, we implement it using the well-known tool for networking simulation Omnetpp with its Castalia simulator. In a repository[6], we make available our Castalia output files. These files can be used as input for the command CastaliaResults to obtain information about the configuration that produced the results and the creation date. Our experiments are built on the IoV characteristics reviewed in Section III.

The configuration of our experiments is shown in Table VII. The network simulation area considers a road with width 50 m and a length 200 m. We deploy two RSUs: one at the beginning and the other at the end of the road. VHs are initially distributed as matrix on the road. We run the six network simulation scenarios listed in Table VIII and detailed as follows.

S1: The scenario has 70 system entities: 1 TA, 1 AD, 2 RSUs, and 66 VHs. Vehicle and AD speeds are 15 MPH and 20 MPH, respectively.

S2: Similar to the first scenario S1, except that the drone speed is increased to 30 MPH.

S3: Similar to the second scenario S2, except that the vehicle speed is increased to 25 MPH.

S4: The scenario has 100 system entities: 1 TA, 1 AD, 2 RSUs, and 96 VHs. Vehicle and drone speeds are 15 MPH and 20 MPH, respectively.

S5: Similar to the fourth scenario S4, except that the drone speed is increased to 30 MPH.

S6: Similar to the fifth scenario S5, except that the vehicle speed is increased to 25 MPH.

We evaluate BDIVE in terms of average time needed for creating a session (S1T), the average time needed for creating two sessions (S2T) of BDIVE average energy consumed per system entity (CE), the average number of transmitted packets (ATX), and the average number of received packets (ARX). The results of the experiments are shown in Table VIII. Figure 3 shows S1T and S2T. The figure shows that the extra time needed to construct the second session after constructing the first one decreases with the increase in the speed of

[6] https://github.com/maelzawawy/BDIVE

TABLE VIII: Scenarios used for testing practical perspectives of BDIVE with their results.

SID	SE#	S1T	S2T	CE	ATX pkts	ARX pkts
S1	70	6.9	62.71	33.99	1133.5	1143.6
S2	70	17.27	82.271	33.99	1260.129	1269.371
S3	70	5.75	82.271	33.997	228.129	235.786
S4	100	7.47	40	33.993	740.98	752.11
S5	100	28	43.61	33.995	521.98	531.23
S6	100	140	166	33.994	625.46	633.47

SID: Scenarios ID, SE#: No. of system entities,S1T: Average time needed for creating a session, S2T: Average time needed for creating two sessions, CE: Average energy consumed per system entity [mJ], ATX: Average number of transmitted packets, ARX: Average number of received packets.

vehicles and drones. This extra time is smaller for scenarios S5 and S6 than for the remaining scenario. This is so because for these two scenarios the number and mobility of vehicles is larger than that of other scenarios. This improves the chance of message arrival. The average energy consumed per node is almost 33.99 mJ. The reported times and consumed energy prove the practicality of BDIVE. This is also confirmed by the small differences between the average number of transmitted and received packets per node reported in Table VIII. While Tables III, IV, and V show a detailed comparisons of BDIVE against state-of-the-art protocols, the objective of Figure 3 is to discuss and prove the practicality and scalability of BDIVE.

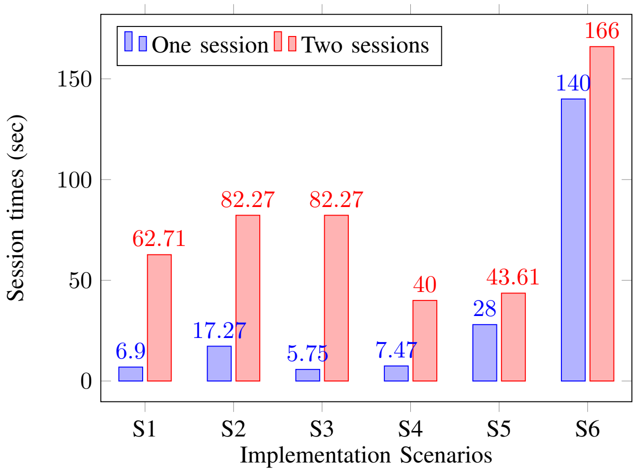


Fig. 3: Times needed to construct one session and two sessions in BDIVE implementation scenarios.

VII. Conclusion and Future Work

In this paper we proposed BDIVE a novel drone-assisted authentication scheme with a three-fold improvement: i) it provides higher security guarantees compared to other state-of-the-art authentication schemes, ii) it reduces the communication and energy costs, thus being a viable solution for resource constrained devices, and iii) it increases the TA security thanks to the use of blockchain. We proved these three claims by thoroughly evaluation both the security and the capabilities of BDIVE comparing it with other state-of-the-art protocols.

In future works we plan to extend our protocol to consider the different type of communication links characterizing IoV networks, hence authenticating all the entities involved in the Vehicle-to-Everything (V2X) paradigm. This represents a significant challenge due to the high number of involved entities and hence high number of exchanged messages which may impact on the communication's latency.

References

[1] J. Hu, C. Chen, L. Cai, M. R. Khosravi, Q. Pei, and S. Wan, "Uav-assisted vehicular edge computing for the 6g internet of vehicles: architecture, intelligence, and challenges," *IEEE Communications Standards Magazine*, vol. 5, no. 2, pp. 12–18, 2021.

[2] G. Xu and Z. Song, "Performance analysis of a uav-assisted rf/fso relaying systems for internet of vehicles," *IEEE Internet of Things Journal*, 2021.

[3] H. Vasudev, V. Deshpande, D. Das, and S. K. Das, "A lightweight mutual authentication protocol for v2v communication in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6709–6717, 2020.

[4] M.-C. Chuang and J.-F. Lee, "Team: Trust-extended authentication mechanism for vehicular ad hoc networks," *IEEE systems journal*, vol. 8, no. 3, pp. 749–758, 2013.

[5] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for internet of vehicles," *Ieee Access*, vol. 7, pp. 12 047–12 057, 2019.

[6] M. N. Aman, U. Javaid, and B. Sikdar, "A privacy-preserving and scalable authentication protocol for the internet of vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1123–1139, 2020.

[7] H. Zhang, J. Wang, and Y. Ding, "Blockchain-based decentralized and secure keyless signature scheme for smart grid," *Energy*, vol. 180, pp. 955–967, 2019.

[8] B. Ji, X. Zhang, S. Mumtaz, C. Han, C. Li, H. Wen, and D. Wang, "Survey on the internet of vehicles: Network architectures and applications," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 34–41, 2020.

[9] M. Khabbaz, J. Antoun, and C. Assi, "Modeling and performance analysis of uav-assisted vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8384–8396, 2019.

[10] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in mec-and uav-assisted vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 131–141, 2020.

[11] W. Y. B. Lim, J. Huang, Z. Xiong, J. Kang, D. Niyato, X.-S. Hua, C. Leung, and C. Miao, "Towards federated learning in uav-enabled internet of vehicles: A multi-dimensional contract-matching approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5140–5154, 2021.

[12] A. Raza, S. H. R. Bukhari, F. Aadil, and Z. Iqbal, "An uav-assisted vanet architecture for intelligent transportation system in smart cities," *International Journal of Distributed Sensor Networks*, vol. 17, no. 7, p. 15501477211031750, 2021.

[13] L. Wan, L. Sun, K. Liu, X. Wang, Q. Lin, and T. Zhu, "Autonomous vehicle source enumeration exploiting non-cooperative uav in software defined internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3603–3615, 2020.

[14] M. A. El-Zawawy, A. Brighente, and M. Conti, "Setcap: Service-based energy-efficient temporal credential authentication protocol for internet of drones," *Computer Networks*, p. 108804, 2022.

[15] L. Song, G. Sun, H. Yu, X. Du, and M. Guizani, "Fbia: A fog-based identity authentication scheme for privacy preservation in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5403–5415, 2020.

[16] J. Zhang, J. Cui, H. Zhong, I. Bolodurina, and L. Liu, "Intelligent drone-assisted anonymous authentication and key agreement for 5g/b5g vehicular ad-hoc networks," *IEEE Transactions on Network Science and Engineering*, 2020.

[17] M. B. Mollah, J. Zhao, D. Niyato, Y. L. Guan, C. Yuen, S. Sun, K.-Y. Lam, and L. H. Koh, "Blockchain for the internet of vehicles towards intelligent transportation systems: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4157–4185, 2020.

[18] T. Jiang, H. Fang, and H. Wang, "Blockchain-based internet of vehicles: Distributed network architecture and performance analysis," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4640–4649, 2018.

[19] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[20] T. A. Butt, R. Iqbal, K. Salah, M. Aloqaily, and Y. Jararweh, “Privacy management in social internet of vehicles: review, challenges and blockchain based solutions,” *IEEE Access*, vol. 7, pp. 79 694–79 713, 2019.

[21] A. Vangala, A. K. Sutrala, A. K. Das, and M. Jo, “Smart contract-based blockchain-envisioned authentication scheme for smart farming,” *IEEE Internet of Things Journal*, 2021.

[22] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[23] R. Canetti and H. Krawczyk, “Universally composable notions of key exchange and secure channels,” in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2002, pp. 337–351.

[24] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, “Examining smart-card security under the threat of power analysis attacks,” *IEEE transactions on computers*, vol. 51, no. 5, pp. 541–552, 2002.

[25] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Annual international cryptology conference*. Springer, 1999, pp. 388–397.

[26] D. Wang, D. He, P. Wang, and C.-H. Chu, “Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment,” *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2014.

[27] B. Bera, A. Vangala, A. K. Das, P. Lorenz, and M. K. Khan, “Private blockchain-envisioned drones-assisted authentication scheme in iot-enabled agricultural environment,” *Computer Standards & Interfaces*, vol. 80, p. 103567, 2022.

[28] Q. Jiang, S. Zeadally, J. Ma, and D. He, “Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks,” *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[29] P. Ruan, T. T. A. Dinh, D. Loghin, M. Zhang, G. Chen, Q. Lin, and B. C. Ooi, “Blockchains vs. distributed databases: dichotomy and fusion,” in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1504–1517.

[30] M. Abadi and M. R. Tuttle, “A logic of authentication,” in *ACM Transactions on Computer Systems*. Citeseer, 1990.

[31] P. Mohit, R. Amin, and G. Biswas, “Design of authentication protocol for wireless sensor network-based smart vehicular system,” *Vehicular Communications*, vol. 9, pp. 64–71, 2017.

[32] J. Lee, G. Kim, A. K. Das, and Y. Park, “Secure and efficient honey list-based authentication protocol for vehicular ad hoc networks,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2412–2425, 2021.

[33] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani *et al.*, “The avispa tool for the automated validation of internet security protocols and applications,” in *International conference on computer aided verification*. Springer, 2005, pp. 281–285.

Mohamed A. El-Zawawy is Full Professor of Computer Science at Faculty of Science, Cairo University Since 2022. He received a Ph.D. in Computer Science from the University of Birmingham in 2007, an M.Sc. in Computational Sciences in 2002 from Cairo University and a B.Sc. in Computer Science in 1999 from Cairo University. At Faculty of Science, Cairo University and during the periods 2007- 2014 and 2014- 2022, he held the positions of Assistant Professor and Associate Professor of Computer Science, respectively. During the year 2009, he held the position of an extra-ordinary senior research at the Institute of Cybernetics, Tallinn University of Technology, Estonia, and worked as a teaching assistant at Cairo University from 1999 to 2003 and later at Birmingham University from 2003 to 2007. He is mainly interested in security and privacy of Android and IoT. In this area, he published many papers in topmost international peer-reviewed journals.

Alessandro Brighente is assistant professor at the University of Padova. He received his Ph.D. degree in Information Engineering from the University of Padova in Feb. 2021. He was visiting researcher at Nokia Bell Labs, Stuttgart and University of Washington, Seattle in 2019 and 2022, respectively. He has been involved in European projects and industrial projects with the University of Padova. He served as TPC for several conferences, including Globecom and VTC. He is guest editor for IEEE Transactions on Industrial Informatics. His current research interests include security and privacy in cyber-physical systems, vehicular networks, blockchain, and physical layer security.

Mauro Conti is Full Professor at the University of Padua, Italy. He is also affiliated with TU Delft and University of Washington, Seattle. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor at the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of Security and Privacy. In this area, he published more than 450 papers in topmost international peer-reviewed journals and conferences. He is Editor-in-Chief for IEEE Transactions on Information Forensics and Security, Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and has been Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, CANS 2021, and General Chair for SecureComm 2012, SACMAT 2013, NSS 2021 and ACNS 2022. He is Fellow of the IEEE, Senior Member of the ACM, and Fellow of the Young Academy of Europe.