computers

MDPI

Article

DeliveryCoin: An IDS and Blockchain-Based Delivery Framework for Drone-Delivered Services

Mohamed Amine Ferrag [1] and Leandros Maglaras [2,*]

[1] Department of Computer Science, Guelma University, Guelma 24000, Algeria
[2] School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK
* Correspondence: leandros.maglaras@dmu.ac.uk

Received: 20 July 2019; Accepted: 5 August 2019 ; Published: 6 August 2019

Abstract: In this paper, we propose an intrusion detection system (IDS) and Blockchain-based delivery framework, called DeliveryCoin, for drone-delivered services. The DeliveryCoin framework consists of four phases, including system initialization phase, creating the block, updating the blockchain, and intrusion detection phase. To achieve privacy-preservation, the DeliveryCoin framework employs hash functions and short signatures without random oracles and the Strong Diffie–Hellman (SDH) assumption in bilinear groups. To achieve consensus inside the blockchain-based delivery platform, we introduce a UAV-aided forwarding mechanism, named pBFTF. We also propose an IDS system in each macro eNB (5G) for detecting self-driving network attacks as well as false transactions between self-driving nodes. Furthermore, extensive simulations are conducted, and results confirm the efficiency of our proposed DeliveryCoin framework in terms of latency of blockchain consensus and accuracy.

Keywords: blockchain; unmanned aerial vehicle; drone-delivered services; security; IDS

1. Introduction

Today we are witnessing the beginning of a new era in the automotive industry, that of highly automated driving. The BMW company [1] defined five levels of autonomous driving: (1) driver assistance, (2) partly automated driving, (3) highly automated driving, (4) fully automated driving, and (5) full automation (no driver!). These levels characterize the evolution of autonomous driving. Many leading companies are currently pioneering and working on self-driving car technology (e.g., Tesla, Google, Ford, Lyft and Volvo). Without humans behind the wheel, researchers in these companies try to solve some problems such as snow or bad weather, which could block the view of lane lines.

Based on the integration of 5G networks into the future smart city concept, the Internet of Vehicle (IoV) has emerged as a new research field of "vehicle-to-vehicle (V2V) communication" for the Internet of Things (IoT) [2]. The applications of IoV can be classified into two categories; (1) Safety and Management and (2) Business Oriented, as discussed by Kaiwartya et al. [3]. The Safety and Management category includes accident prevention, emergency call, real-time traffic information, parking helper, etc. The Business Oriented category includes car sharing, connected driving, car pooling, etc. In our work, we consider 5G enabled IoV for buying and delivering packages using two types of autonomous vehicles, including unmanned aerial vehicles (UAVs) and self-driving cars, as presented in Figure 1.

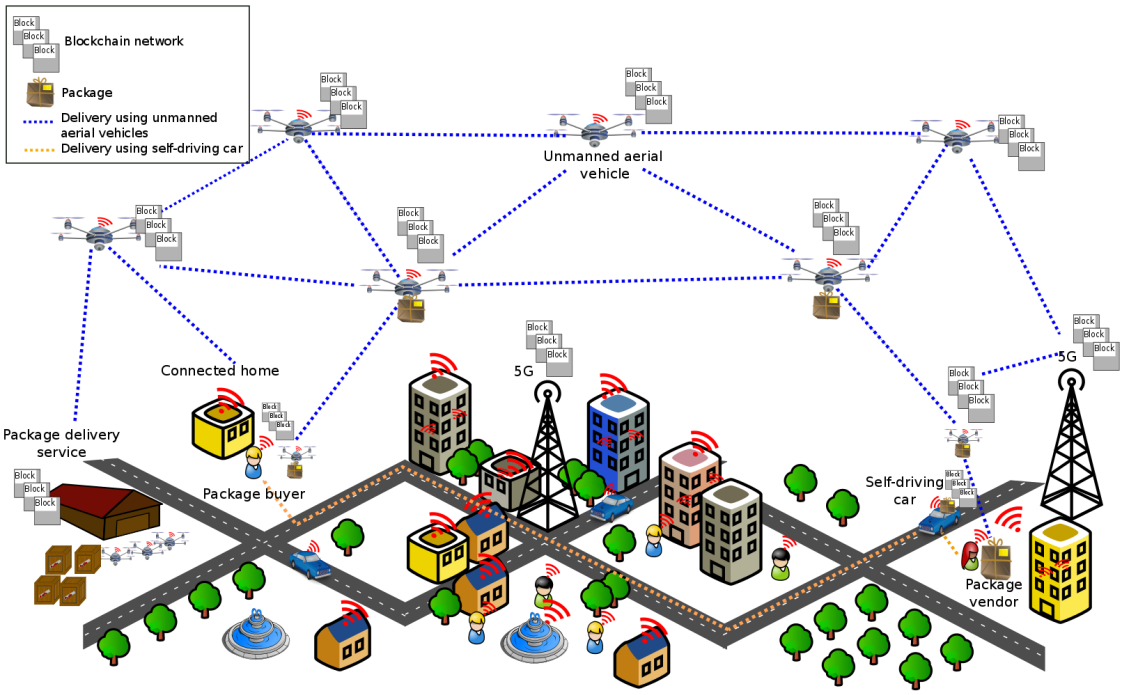


Figure 1. 5G enabled Internet of Vehicles for buying and delivering packages.

The world's leading logistics companies (e.g., DHL, UPS, Amazon) have recently launched a new service, called the UAV-based Delivery System, which aims at modernizing the delivery systems in the near-future. The UAV-based delivery system can be used for urban first and last mile delivery, rural delivery, surveillance of logistic infrastructures, and use for intra-logistics [4]. The main problem in the development of drone-delivered services is not located at the physical support but mainly in reassuring both security and privacy. Based on locations, identities, and profiles, an adversary can launch active or passive attacks (e.g., via a false data injection attack) in order to steal the data routing over flying UAVs, vehicles, and roadside infrastructure. Hence, in order to protect the drone-delivered services within a smart city environment, the privacy-preserving scheme should preserve the privacy of sensitive information (e.g., location) from vehicles and other drones [5].

The blockchain technology was born when Satoshi Nakamoto proposed the cryptocurrency Bitcoin (BTC) [6]. The blockchain technology can be effectively applied in almost all domains of the Internet of Things [7–12]. Yang et al. [8] proposed a trusted routing scheme, which is based on Blockchain and reinforcement learning for wireless sensor networks. The study uses four core technical elements, including distributed ledger, asymmetric encryption, consensus mechanism, and smart contract. To improve the trustworthiness of the routing information between the routing nodes, the study uses decentralized, tamper-proof and traceable characteristics of the blockchain transactions. Pieroni et al. [10] proposed a smart energy grid architecture, named Smarter City, which is based on blockchain technology. Derhab et al. [11] proposed software-defined wide-area network architecture, named SD-WAN, for industrial control systems. The SD-WAN architecture is based on two complementary components, including (1) an intrusion detection system, named RSL-KNN; and (2) a blockchain-based Integrity checking system, named BICS. The RSL-KNN system uses two machine learning approaches, namely, random subspace learning approach and K-Nearest Neighbor classifier, against forged command attacks. The BICS system analyzes the flow rules against the misrouting attack.

In this paper, we propose an intrusion detection system (IDS) and Blockchain-based delivery framework, entitled DeliveryCoin, for drone-delivered services. The main contributions of this work are summarized as follows:

- We propose a new Blockchain-based delivery framework for facilitating the package delivery service among self-driving nodes. To achieve privacy-preservation, the proposed scheme employs hash functions and short signatures without random oracles and the Strong Diffie–Hellman (SDH) assumption in bilinear groups.
- We introduce a UAV-aided forwarding mechanism, named pBFTF, that UAVs use in order to achieve consensus inside the blockchain-based delivery platform.
- We propose an IDS system in each macro eNB (5G) for detecting self-driving network attacks as well as false transactions between self-driving nodes. To the best of our knowledge, this is the first study that combines blockchain technology with an IDS system into one architecturally secure framework for an UAV-based delivery system.
- We provide various simulation results in terms of latency of blockchain consensus and accuracy.

The remainder of the paper is organized as follows. In Section 3, we present the threat model. In Section 4, we describe the proposed DeliveryCoin framework. The performance evaluation of the proposed DeliveryCoin framework is presented in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Recent studies have used blockchain technology to establish secure data sharing for vehicular networks. Cebe et al. [13] proposed an integrated lightweight blockchain framework, named Block4forensic, for connected vehicles. Based on all related parties such as drivers and car manufacturers, without requiring a trusted third party, the Block4forensic framework provides a lightweight privacy-aware blockchain. The Block4forensic framework uses three types of data, including event data, diagnosis data, and maintenance data. These forensic data types are used by four different types of nodes, namely, (1) Leader; (2) Validator; (3) Monitor units and (4) Client. These nodes apply permission blockchain technology and implement shared and fragmented ledgers. In addition, the Block4forensic framework preserves integrated membership management and privacy using pseudonym identities from the VPKI model suggested in IEEE 1609.2.

The framework is used by Kang et al. [14] as a fair metric for enhancing Proof-of-Stake (DPoS) schemes through a two-stage mechanism, including secure miner selection and reliable block verification. Specifically, this reputation management is proposed for miner selection, leading to selection of miner candidates with high reputation. This study indicated that the multiweight subjective logic model is particularly suitable for decreasing collusion between stakeholders. To achieve the reliability of confirming event occurrences for VANETs, Yang et al. [15] proposed a blockchain-based traffic event validation framework, named BTEV, which is based on the Proof-of-Event (PoE) consensus mechanism. In order to help identify the truth of events, the PoE mechanism is combined with fast event notification as well as trust verification of roadside units. In addition, the block producer in the PoE mechanism can be verified by other nodes.

Li et al. [16] proposed a network model for smart vehicles based on five entities, including trusted authority, trace manager, users, RSUs, and a cloud application server. To provide privacy-preserving in this network model, the study proposed a blockchain-based incentive announcement framework, named CreditCoin. To generate the signatures and to send announcements anonymously between smart vehicles, the CreditCoin framework applies an anonymous vehicular announcement aggregation protocol. The CreditCoin framework can achieve conditional privacy since malicious users' identities can be traced in anonymous announcements.

3. Threat Model

In our threat model, the macro eNBs (5G) are trustable and non-compromisable. However, we consider a global external attacker A against a blockchain-based delivery platform, which can perform the following six categories of attacks, including brute-force attacks, web attacks, Denial of Service (DoS) attacks, Distributed Denial of Service (DDoS) attacks, infiltration attacks, and Botnet

attacks. These attacks are simulated in the CSE-CIC-IDS2018 dataset [17]. As presented in Figure 2, we consider two networks, including (1) Attack-Network and (2) Victim-Network.The global external attacker A is located at Attack-Network while blockchain-based delivery platform is located at Victim-Network.

- *Brute-force attacks*: A brute force attack is an attempt to crack a password or username through a trial and error method, with dictionaries being the most basic tools. The use of both a Central Processing Unit (CPU) and Graphics Processing Unit (GPU) together increases the efficiency of brute force attacks. We assume that the adversary A lunches two types of brute-force attacks, including SSH-Bruteforce and FTP-BruteForce. The SSH-Bruteforce and FTP-BruteForce attacks use Secure Shell (SSH) and File Transfer Protocol (FTP) connections, and start by performing a series of tests to discover a valid blockchain id and password in order to take control of a legitimate blockchain node. Please note that there are other types of brute-force attacks, such as brute-force LDAP, brute-force SMB, brute-force of the password of encrypted ZIP, etc.
- *Web attacks*: A web attack is an attempt to manipulate web applications into altering Structured Query Language (SQL) commands and sending malformed requests in order to retrieve sensitive information. We assume that the adversary A lunches three types of web attacks against the blockchain-based delivery platform, including SQL Injection, Brute Force-Web, and Brute Force -XSS. Based on SQL queries (i.e., Select From Where), SQL Injection attacks create, read, update, alter or delete the block stored in the blockchain's SQL database. Web attacks can be launched by a PHP/MySQL web application, named DVWA (http://www.dvwa.co.uk/). A Cross-Site Scripting (XSS) attack injects malicious scripts into the blockchain's SQL database as well as the miner database.
- *DoS attacks*: A DoS attack attempts to make a network application unable to respond to requests from its users. We assume that the adversary A lunches four types of DoS attacks, including DoS attacks-Slowloris, DoS attacks-GoldenEye, DoS attacks-Hulk and DoS attacks-SlowHTTPTest. These attacks are used to overwhelm the blockchain servers by opening and maintaining many simultaneous HTTP connections.
- *DDoS attacks*: A DDoS attack attempts to make a distributed network unable to respond to requests from its users. We assume three types of DDoS attacks, including DDoS attack-LOIC-HTTP, DDoS attack-HOIC, and DDoS attack-LOIC-UDP. In order to overload the blockchain-based delivery platform, these attacks send a large sequence of UDP, TCP or HTTP requests to the distributed ledger.
- *Infiltration attacks*: An Infiltration attack is a malicious file (e.g., sent via an email) that attempts to enter and/or damage a user's device. We assume that the adversary A sends a malicious software via an email to blockchain nodes, which can be virus, worm, trojan horse, rootkit, adware, or spyware. Specifically, the malicious program is launched for the following three objectives: (1) delete blockchain file storage from the node, (2) degrade the performance of the mining system, and (3) block access of blockchain programs to the blockchain-based delivery platform.
- *Botnet attacks*: A botnet attack takes place when a network of devices is infected by a malicious software, in order to be remotely controlled by an adversary. We assume that the adversary A uses malicious software (e.g., Mirai IoT Botnet) for identifying and comproming connected objects and then running targeted DDoS attacks, in order to mine the cryptocurrency used by the blockchain-based delivery platform.

As discussed in our recent work [7], the blockchain network suffers from a nunber of vulnerabilities, such as private key leakage, double spending, 51% vulnerability, transaction privacy leakage, and selfish and reputation-based behaviors. The six categories of attacks considered in our threat model can exploit these vulnerabilities.

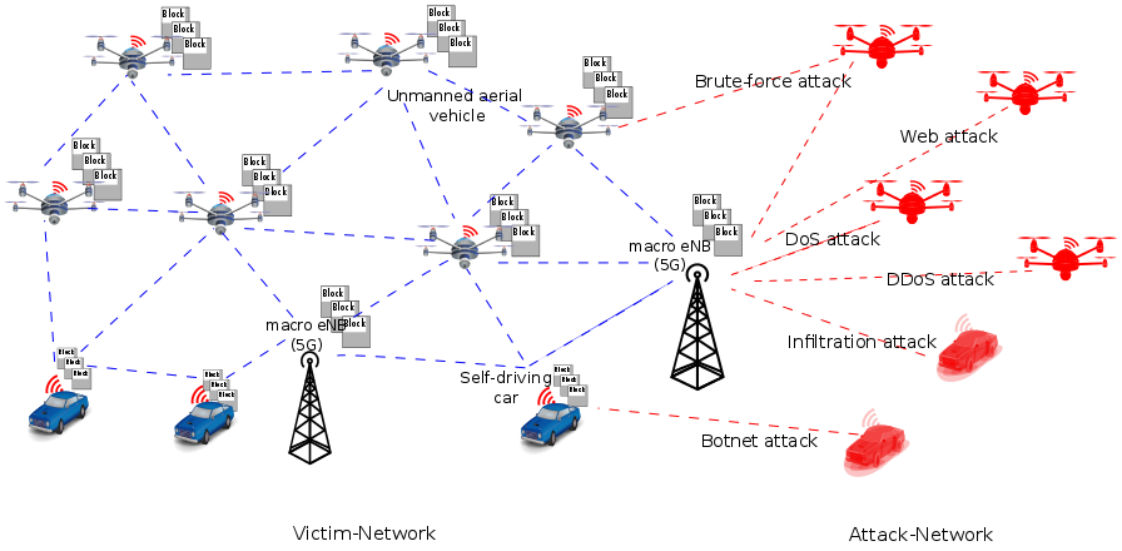


Figure 2. Threat model architecture.

4. The DeliveryCoin FrameWork

4.1. Network Model

In our network model, we consider five network entities: *Package buyer*, *Package vendor*, *Package delivery service*, *Autonomous vehicle, and Macro eNB (5G)*, which are described below.

- *Package buyer*: We assume two types of package buyers, including $Cust_{HAN}$ and $Store_{WAN}$, which are package network entities located in the Home Area Network (HAN) and the Wide Area Network (WAN), respectively. The customer $Cust_{HAN}$ and store center $Store_{WAN}$ plan to trade with package vendors by buying products on online shopping or over the Internet. These package network entities use a fully decentralized network, which does not depend on any central authority. The electronic payment is online and its unit of account is called DivCoin.
- *Package vendor*: We assume two types of package vendors, including Ven_{HAN} and Ven_{WAN}, which are package network entities located in the HAN network and the WAN network, respectively. These package network entities plan to sell products to package buyers.
- *Package delivery service (CDS)*: An entity which offers delivery services of package and documents based on the blockchain technology and machine learning approach. The blockchain is used as a database that handles the management of a certified and protected list of transactions between package vendors and buyers against falsification or modification. In addition, the blockchain [7] is a distributed digital ledger containing all package transactions in the autonomous Vehicle network. This distributed ledger is replicated and stored in different nodes, including $Cust_{HAN}$, $Store_{WAN}$, Ven_{HAN}, and Ven_{WAN}. A machine learning approach is used by an intrusion detection system (IDS) for detecting network attacks and false transactions.
- *Autonomous vehicle*: We assume two types of autonomous vehicles UAV_i, including unmanned aerial vehicles (UAVs) and self-driving cars. In order to provide cost-effective wireless connectivity for autonomous vehicles, the entities use two basic types of communication links, including the non-payload communications (CNPC) link and the data link [18]. As presented by Mozaffari et al. [19], the UAVs can be classified according to *altitude* (i.e., high altitude platform and low altitude platform) or *type* (i.e., fixed-wing and rotary-wing). We assume that users select the type of the autonomous vehicles according to the distance between package buyers and package vendors.
- *Macro eNB (5G)*: A terrestrial cellular network entity for supporting ground users as well as serve aerial users. We assume that this entity support drones in wireless networking applications such

as the concept of a 3D cellular network proposed by Mozaffari et al. [20], which incorporates both drone base stations and cellular-connected drone users. In addition, this entity provides the consensus process in the blockchain network and also detects network attacks and false transactions using a IDS system.

4.2. Description of DeliveryCoin Framework

The DeliveryCoin framework consists of four phases: system initialization phase, creating the block, updating the blockchain, and intrusion detection phase.

4.2.1. System Initialization Phase

Given the security parameter k and the bilinear groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ where $|\mathbb{G}_1| = |\mathbb{G}_2| = q$ for some prime q. A bilinear map is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the following three properties [21]: (1) $e(a^u, b^v) = e(a, b)^{uv}$ where $a \in \mathbb{G}_1$, $b \in \mathbb{G}_2$, and $u, v \in \mathbb{Z}$; (2) $e(g_1, g_2) \neq 1$ where g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$; (3) the group action in $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ can be computed efficiently. Then, the macro eNB (5G) pick a random generator $g_2 \in \mathbb{G}_2$, set $g_1 \in \varphi(g_2)$, and chooses a collision-resistant hash function $H : \{0,1\} \rightarrow \mathbb{Z}_q^*$. After these, the macro eNB (5G) sets the system public parameters $param = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g_1, g_2, H)$. When an entity node EN_x registers to the system where $x \in \{Cust_{HAN}, Store_{WAN}, Ven_{HAN}, Ven_{WAN}\}$, the macro eNB (5G) invokes Algorithm 1.

Algorithm 1 Registration Algorithm

Input: an entity node EN_x and system public parameters $param$
Output: sk_{EN_x} and pk_{EN_x}
1: Pick random $a, x, y, z \in \mathbb{Z}_q^*$;
2: Computes $u \leftarrow g_2^x \in \mathbb{G}_2$ and $v \leftarrow g_2^y \in \mathbb{G}_2$;
3: Computes $f \leftarrow e(g_1, g_2) \in \mathbb{G}_T$;
4: Computes the secret key is $sk_{EN_x} = (x, y, z, g_1^{\frac{1}{a+z}})$;
5: Computes the public key $pk_{EN_x} = (g_1, g_2, u, v, f)$;
6: **return** sk_{EN_x} and pk_{EN_x}

4.2.2. Creating the Block

When a package buyer node PB_x ($x = Cust_{HAN}$ or $Store_{WAN}$) plan to trade with package vendor node PV_x by buying products, they negotiate the price and quantity. Then, given a secret key $sk_{PV_i} = (x, y, z, g_1^{\frac{1}{a+z}})$ of the package vendor and a bloc $B_i \in \mathbb{Z}_q^*$, the package vendor PV_i pick a random $r \in \mathbb{Z}_q^*$ and computes $\sigma \leftarrow g_1^{1/(x+bloc+yr)}$, $\rho = (g_1{}^z, g_1^{\frac{1}{a+z}}, PV_i{}^{\frac{1}{z+H(T)}})$, which T_i is the time of creation of the block B_i. The signature of the block B_i is (σ, ρ, r).

4.2.3. Updating the Blockchain

Based on the Practical Byzantine Fault Tolerance (pBFT) algorithm [22], we adopt a UAV-aided forwarding mechanism, named pBFTF, where UAVs are used to achieve consensus in the blockchain-based delivery platform. Specifically, the pBFTF mechanism executes the following steps:

- *Step 1*. The package vendor PV_i sends his request to the macro eNB (5G).
- *Step 2*. After receiving the request at time T', the macro eNB (5G) invokes Algorithm 2.
- *Step 3*. The macro eNB (5G) creates a PRE-PREPARE message to to the other replicas, in order to propose the scheduling of the request in the blockchain network. This message contains a unique sequence number and a timestamp.
- *Step 4*. When a passing-by UAV node UAV_i is willing to help forwarding the message PREPARE, the macro eNB (5G) first investigates the destination location and computes the forwarding degree FD_i, which is the average time to reach the next-hop macro eNB (5G) node. Then, the macro eNB

(5G) node invokes Algorithm 3 to forward the message PRE-PREPARE-UAV to a proper next-hop macro eNB (5G) node. After these, the macro eNB (5G) creates a PRE-PREPARE-UAV message and sends it to UAV_i, as shown in Figure 3.

- *Step 5*. The UAV_i node forward the message PRE-PREPARE-UAV to next-hop macro eNB (5G) eNB_r. Then, the eNB_r invokes the first steps (i.e., *Step 1, Step 2, and Step 3*).
- *Step 6*. The correct nodes EN_x where $x \in \{Cust_{HAN}, Store_{WAN}, Ven_{HAN}, Ven_{WAN}\}$ respond with a PREPARE message, which is sent to all replicas. Please note that the macro eNB (5G) does not send the PREPARE message.
- *Step 7*. Once the correct nodes EN_x received $2f$ PREPARE message and the corresponding PRE-PREPARE, they agree on the order of the package vendor's request in the blockchain network. Then, the correct nodes EN_x send a message COMMIT.
- *Step 8*. Once a correct node EN_x received $2f + 1$ COMMIT message, it executes the order of block and responds to the package vendor PV_i with a REPLY message.

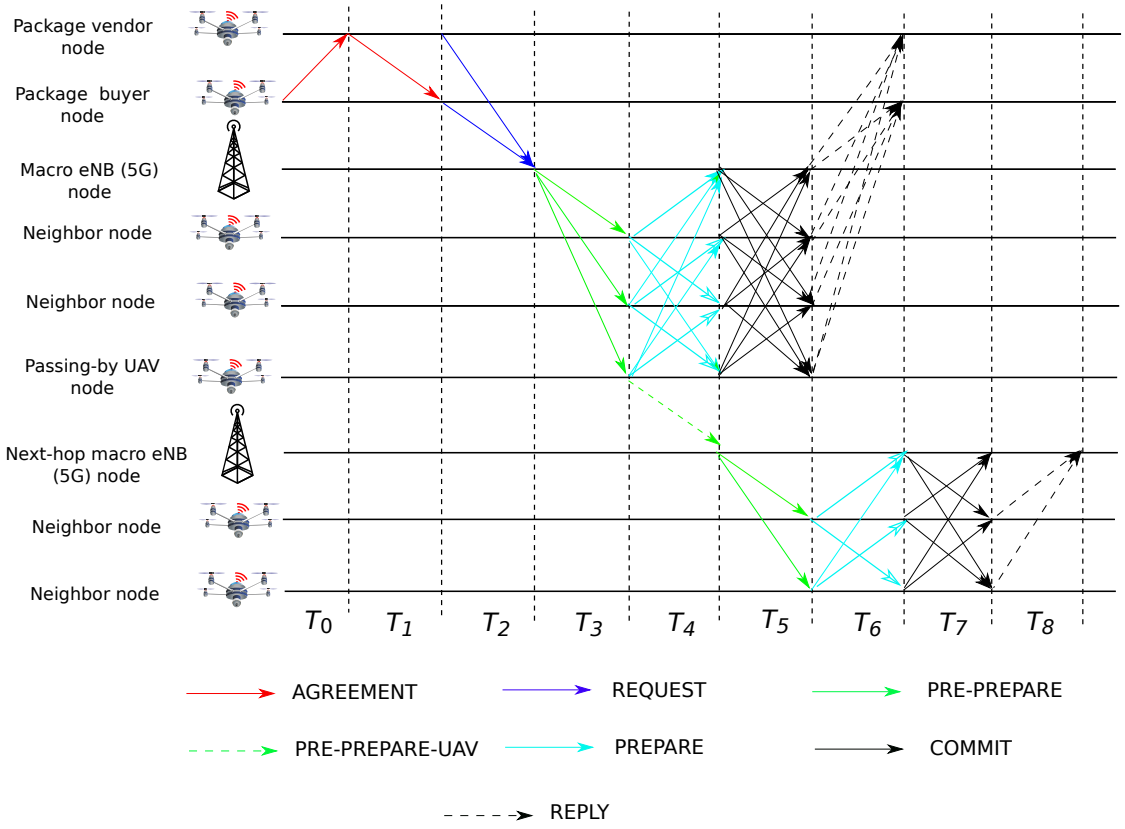


Figure 3. The consensus process for blockchain-based delivery platform, which the macro eNB (5G) node invokes UAV-aided forwarding algorithm to forward the message PRE-PREPARE-UAV to a proper next-hop macro eNB (5G) node. Let $Trans_{eNB}$, $Trans_{Next\_eNB}$, and $Trans_{UAV}$, where $Trans_{eNB} > Trans_{UAV}$ and $Trans_{eNB} = Trans_{Next\_eNB}$, be the transmission ranges of macro eNB (5G), next-hop macro eNB (5G), and UAV nodes, respectively. A passing-by UAV node UAV_i in $Trans_{eNB}$ is willing to help forwarding the message PREPARE to next-hop macro eNB (5G). When the UAV_i node leaves the transmission $Trans_{eNB}$ and enters in the transmission $Trans_{Next\_eNB}$, the UAV_i node sends the PRE-PREPARE-UAV message to next-hop macro eNB (5G).

Algorithm 2 Checking the validity of package vendor requests

Input: package buyer node PB_x, package vendor node PV_x, block B_i, time T_i' of creation of the block, and system public parameters $param$
Output: $Success$ or $Failure$

The macro eNB (5G) verify the signature (σ, ρ, r) of the block B_i using the public key (g_1, g_2, u, v, f). If the validity is true then $Validity\_PB_x = valid$ otherwise the result is $Validity\_PB_x = invalid$;
The macro eNB (5G) checks if the package buyer node PV_x has enough DivCoin to buy. If the validity is true then $Validity\_PV_x = valid$ otherwise the result is $Validity\_PV_x = invalid$;
if $T_i' - T_i \leq \triangle T$ **then**
    if $Validity\_PB_x = valid$ **then**
        if $Validity\_PV_x = valid$ **then**
            return Success;
        else
            The macro eNB (5G) sends a penalty to the PV_x node;
        end if
    else
        The macro eNB (5G) sends a penalty to the PB_x node;
        return Failure;
    end if
else
    return Failure;
end if

Algorithm 3 UAV-aided forwarding algorithm

Input: macro eNB (5G) sender eNB_s, next-hop macro eNB (5G) eNB_r, UAV_i, PRE-PREPARE-UAV, forwarding degree FD_i
Output: $Success$ or $Discarded$

When the macro eNB (5G) eNB_s try to forward PRE-PREPARE-UAV to the next-hop macro eNB (5G) eNB_r, the eNB_s set a holding time to wait eNB_r (T_h);
if no next-hop macro eNB (5G) eNB_r is available **then**
    return Discarded
else
    The macro eNB (5G) eNB_s chooses the UAV_i who has less the forwarding degree FD_i;
    The macro eNB (5G) eNB_s sends PRE-PREPARE-UAV to UAV_i;
    The UAV_i forward PRE-PREPARE-UAV to the next-hop macro eNB (5G) eNB_r;
    return Success;
end if

4.2.4. Intrusion Detection Phase

We propose an IDS system in each macro eNB (5G) for detecting network attacks and false transactions. As presented in Figure 4, the IDS system consists of five stages: (1) dataset stage, (2) pre-processing stage, (3) normalization stage, (4) training stage and (5) testing stage. The dataset stage consists of the selection of benchmark dataset which contains different attack scenarios. The pre-processing stage consists of apportioning the data set into training and test sets as well as labelling each row as an attack or benign. The normalization stage consists of normalizing the different features of the data set. The training stage consists of using machine learning classifiers (e.g., Random Forest, Learning Vector Quantization, Linear Regression, Naive Bayes, K-Nearest Neighbors, Support Vector Machines, Deep learning, ...etc.) to obtain a model. The test stage consists of processing each

row of the test data using the model obtained from the training stage in order to classify as Benign or a specific type of attack.

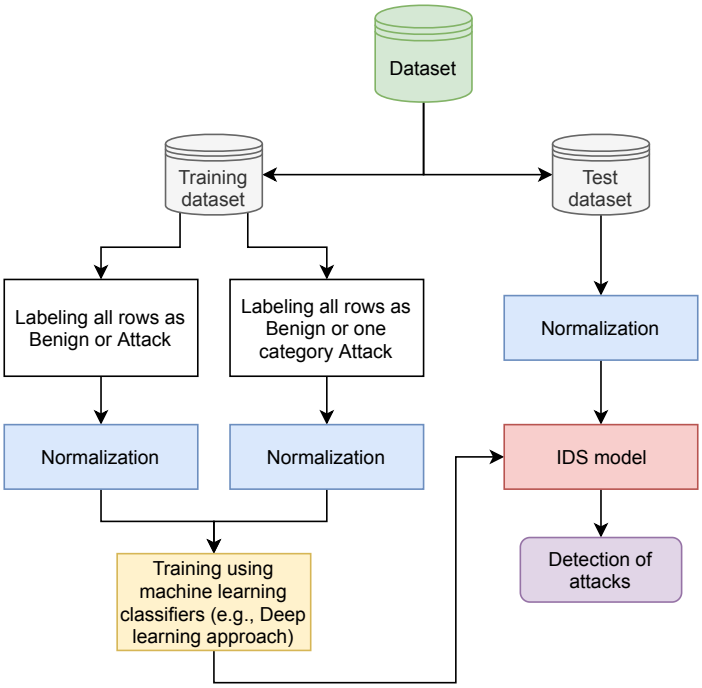


Figure 4. Intrusion detection phase in DeliveryCoin framework.

5. Performance Evaluation

To evaluate the performance of DeliveryCoin, we implement the framework and compare its performance using five performance metrics, including accuracy, training time, test time, communication overhead, and latency of blockchain consensus. The accuracy, training time and test time are used for evaluating the performance of the IDS system with different hardware accelerators and four different classification techniques, including support vector machine (SVM), recurrent neural network (RNN), convolutional neural network (CNN), and decision tree (DT). The communication overhead is used for evaluating the performance of communication costs compared to the currently popular Paillier Cryptosystem [23]. The latency of blockchain consensus is used for evaluating the performance overhead and compared to the Prime protocol [24] associated with the network delay between the package buyer and vendor nodes. To study the performance of the IDS system and blockchain platform, we use the Google Colaboratory and a custom simulator built in Java, respectively. The detailed parameter settings are summarized in Table 1.

The drone-delivered services uses different protocols, including (1) Information models and profiles, (2) application layer protocols, (3) transport layer protocols, and (4) media-specific protocols. Based on these communication protocols, we used and selected the most recent data sets that contain different attack scenarios against these communication protocols used by the drone-delivered services. Specifically, we used the CSE-CIC-IDS2018 dataset (https://registry.opendata.aws/cse-cic-ids2018/). This dataset contains six categories of attacks, including brute-force attacks, web attacks, Denial of Service (DoS) attacks, Distributed Denial of Service (DDoS) attacks, infiltration attacks, and Botnet attacks. We consider these attacks against a self-driving network. The CSE-CIC-IDS2018 dataset is an outcome of a collaborative project between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC) (https://www.unb.ca/cic/datasets/ids-2018.html). The simulations are performed on Google Colaboratory (https://colab.research.google.com) under python 3 using TensorFlow library and three types of hardware accelerators, including Central Processing Unit (CPU), Graphics Processing Unit (GPU), and Tensor Processing Unit (TPU). We used

five packages, NumPy, Pandas, Scikit-learn and Keras, and PyMongo. The PyMongo package is used for processing the CSV files into a NoSQL database.

Table 1. Simulation Settings in DeliveryCoin.

Parameter	Setting
Simulation area, duration	100,000 m * 150,000 m, 10 h
UAV nodes	
Number	{100, 150, 200, 250, 300, 350, 400, 450, 500}
Max speed	72 km/h
Battery life	31 min (3850 mAh)
Max Range	8 km/5 mi
Buffer size	100 MB
Holding time to wait next-hop macro eNB (5G)	3 min
eNB (5G) nodes	
Number	2
Frequency	30 GHz to 300 GHz range
IDS	
Dataset	CSE-CIC-IDS2018
Machine learning classifiers	SVM, RNN, CNN, DT
Hardware accelerators	CPU, GPU, TPU
Metrics	True Positive (TP), False Negative (FN), True Negative (TN), False Positive (FP)
Hyperparameters	
Hidden nodes	80
Learning rate	0.01
Number of epoch	100
Batch size	1000
Activation function	Sigmoid
Classification function	SoftMax

Table 2 presents the list of attack types in CSE-CIC-IDS2018 dataset, which contains 15,450,706 rows devised on 10 files, each row having 80 features. The contents of these files are described as follows:

- File 1 "Wednesday-14-02-2018": It contains benign traffic (667,626 rows) and two types of brute-force attacks, including SSH-Bruteforce (187,589 rows) and FTP-BruteForce (193,360 rows).
- File 2 "Thursday-15-02-2018": It contains benign traffic (996,077 rows) and two types of DoS attacks, including DoS attacks-Slowloris (10,990 rows) and DoS attacks-GoldenEye (41,508 rows).
- File 3 "Friday-16-02-2018": It contains benign traffic (442,020 rows) and two types of DoS attacks, including DoS attacks-Hulk (466,664 rows) and DoS attacks-SlowHTTPTest (139,890 rows).
- File 4 "Thursday-20-02-2018": It contains benign traffic (7,372,557 rows) and one type of DDoS attack, named DDOS attack-LOIC-HTTP (576,191 rows).
- File 5 "Wednesday-21-02-2018": It contains benign traffic (360,833 rows) and two types of DDoS attacks, including DDOS attack-HOIC (686,012 rows) and DDOS attack-LOIC-UDP (1730 rows).

- File 6 "Thursday-22-02-2018": It contains benign traffic (1,048,213 rows) and three types of web attacks, including SQL Injection (34 rows), Brute Force -Web (249 rows), and Brute Force -XSS (79 rows).
- File 7 "Friday-23-02-2018": It contains benign traffic (1,048,009 rows) and three types of web attacks, including SQL Injection (53 rows), Brute Force -Web (249 rows), and Brute Force -XSS (151 rows).
- File 8 "Wednesday-28-02-2018": It contains benign traffic (544,200 rows) and one type of infiltration attack, named Infiltration (68,871 rows).
- File 9 "Thursday-01-03-2018": It contains benign traffic (238,037 rows) and one type of infiltration attack, named Infiltration (93,063 rows).
- File 10 "Friday-02-03-2018": It contains benign traffic (762,384 rows) and one type of Botnet attack, named Bot (286,191 rows).

Table 2. Attack Types in CSE-CIC-IDS2018 dataset.

Category	Attack Type	Flow Count	Training	Test
Brute-force	SSH-Bruteforce	230	184	46
	FTP-BruteForce	611	489	122
Web attack	Brute Force -XSS	187,589	15,007	3752
	Brute Force -Web	193,360	15,469	3867
	SQL Injection	87	70	17
DoS attack	DoS attacks-Hulk	466,664	37,333	9333
	DoS attacks-SlowHTTPTest	139,890	111,912	27,978
	DoS attacks-Slowloris	10,990	8792	2198
	DoS attacks-GoldenEye	41,508	33,206	8302
DDoS attack	DDOS attack-HOIC	686,012	54,881	13,720
	DDOS attack-LOIC-UDP	1730	1384	346
	DDOS attack-LOIC-HTTP	576,191	46,095	11,524
Botnet	Bot	286,191	22,895	5724
Infilteration	Infilteration	161,934	12,955	3239
Benign	/	12,697,719	101,582	25,395
Total	/	15,450,706	462,254	115,563

To create a training and test subset, we import the 10 files into one JSON document using PyMongo 3.7.2. Then, we apportion the data into training and test sets, with an 80-20 split. Each value x_i of the feature j is normalized based on the following equation:

$$\overline{x_i(j)} = \frac{x_i(j) - min(x(j))}{max(x(j)) - min(x(j))}$$

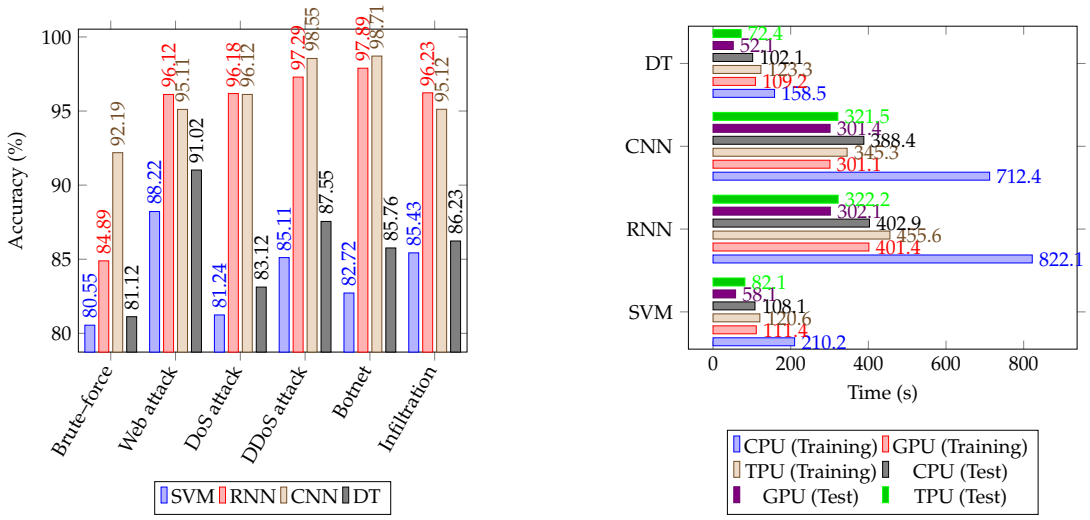
The most important performance indicator "accuracy" is used to represent the proportion of the total number of correct classifications.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Evaluation Results

Figure 5 shows the accuracy of the proposed DeliveryCoin with different hardware accelerators and four different classification techniques, including support vector machine (SVM), recurrent neural

network (RNN), convolutional neural network (CNN), and decision tree (DT). By comparing the results of the classifiers, we can see that overall CNN and RNN classifiers the effectiveness and accuracy of results are better than SVM and DT classifiers. The CNN classifier achieves the best effectiveness and accuracy in three types of attacks, including Brute force attack, DDoS attack, Botnet attack, which the achieved accuracy in these states being 92.19%, 98.55%, and 98.71%, respectively. The RNN classifier achieves the best effectiveness and accuracy in three types of attacks, including Web attack, DoS attack, and Infiltration attack, which the achieved accuracy in these states being 96.12%, 96.18%, and 96.23%, respectively. The results of the performance comparison in terms of training and test time, for the CSE-CIC-IDS2018 dataset, are shown in Figure 5b. From these results, we can see that the CNN classifier requires less training time as compared to the RNN classifier. In addition, the deep learning approaches (i.e., RNN and CNN) with graphics processing unit are recommended for use in DeliveryCoin framework where good accuracy and short training time are desired.

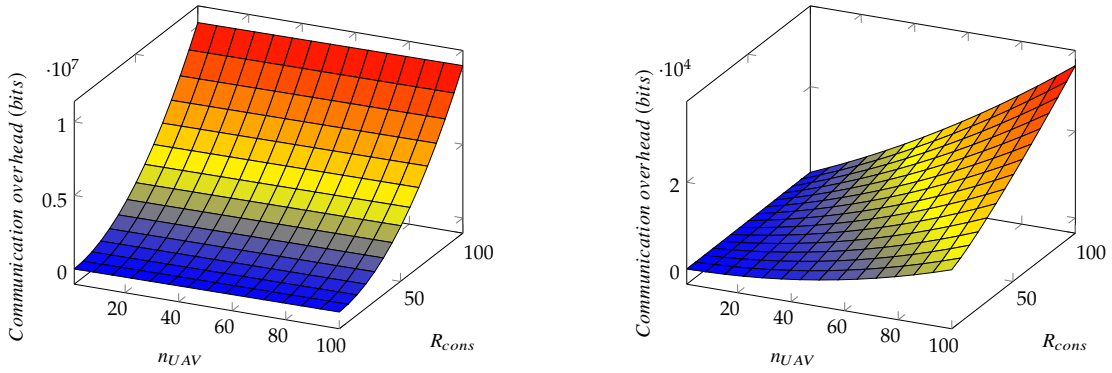


(**a**) Accuracy of the proposed DeliveryCoin (**b**) Time Comparison

Figure 5. Accuracy of the proposed DeliveryCoin with different hardware accelerators and four different classification techniques, including support vector machine (SVM), recurrent neural network (RNN), convolutional neural network (CNN), and decision tree (DT).

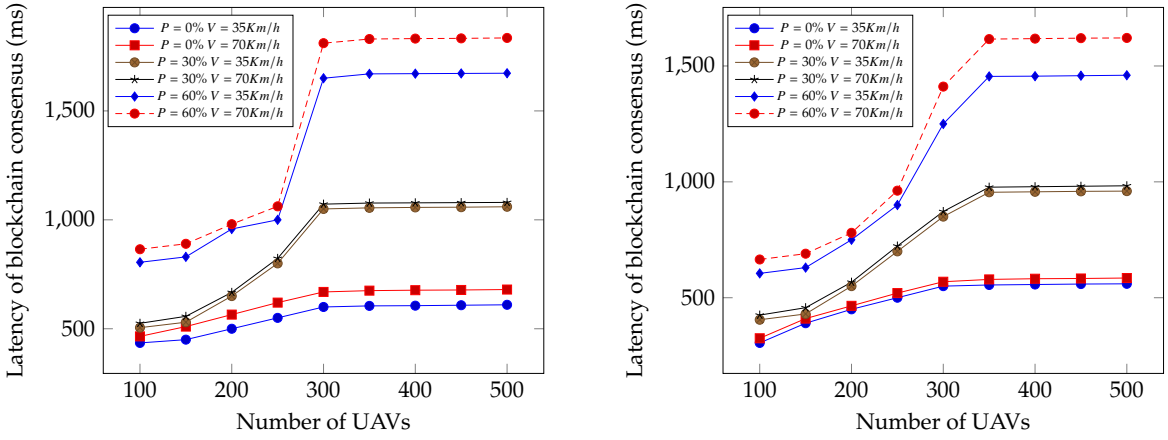
Figure 6 shows comparisons between the proposed DeliveryCoin and the Paillier Cryptosystem-based DeliveryCoin in term of communication overhead for varying number of number of UAVs n_{UAV} and different values for reaching a consensus R_{cons} in the blockchain-based delivery network. In contrast to the Paillier Cryptosystem-based DeliveryCoin, we can obviously observe that the proposed DeliveryCoin is very efficient in terms of communication cost.

Figure 7 shows comparisons between the proposed DeliveryCoin and the Prime-based DeliveryCoin [24] in term of latency of blockchain consensus for various number of UAVs N_{UAV} = {100, 150, 200, 250, 300, 350, 400, 450, 500}, probabilities of malicious UAV nodes P = {0%, 30%, 60%}, and velocities of UAV nodes V ={35 Km/h, 70 Km/h}. When P = 0%, the proposed DeliveryCoin has a lower latency of blockchain consensus than the Prime-based DeliveryCoin. As the probability P increases, malicious UAV nodes can add more false transactions in order to add more delay. In Figure 7a, when P = 60%, malicious UAV nodes under the Prime-based DeliveryCoin can add approximately 300 ms more delay than when P = 30%. In Figure 7b, when P = 60%, the malicious UAV nodes under the proposed DeliveryCoin can add approximately 200 ms more delay than when P = 30%. In addition, we can see that when the number of UAVs N_{UAV} increases, the latency of blockchain consensus begins to climb steeply due to fact that the macro eNB (5G) node invokes UAV-aided forwarding algorithm as well as updates queuing at UAV nodes more frequently.



(**a**) Paillier Cryptosystem-based DeliveryCoin (**b**) Proposed DeliveryCoin

Figure 6. Comparisons between the proposed DeliveryCoin and the Paillier Cryptosystem-based DeliveryCoin [23] in term of communication overhead varies with the number of UAVs n_{UAV} and reaching a consensus R_{cons} in the blockchain-based delivery network.



(**a**) Prime-based DeliveryCoin (**b**) Proposed DeliveryCoin

Figure 7. Comparisons between the proposed DeliveryCoin and the Prime-based DeliveryCoin [24] in term of latency of blockchain consensus varies with the number of UAVs N_{UAV} = {100, 150, 200, 250, 300, 350, 400, 450, 500}, probability of malicious UAV nodes P ={0%, 30%, 60%}, and velocity of UAV nodes V ={35 Km/h, 70 Km/h}.

Table 3 demonstrates a comparison of the proposed DeliveryCoin framework with other blockchain-based systems for vehicular networks. The proposed DeliveryCoin framework uses a UAV-aided forwarding mechanism, in which UAVs are used to achieve consensus inside the blockchain-based delivery platform. The proposed DeliveryCoin framework uses an IDS system in each macro eNB (5G) for detecting self-driving network attacks, with the CSE-CIC-IDS2018 dataset used in our simulations. In addition, DeliveryCoin is the only suitable method for UAV-based delivery system as compared to the other blockchain-based systems that were discussed in the related work section.

Table 3. Performance comparison with other blockchain-based systems for vehicular networks.

Framework	Year	IDS	Dataset	Consensus **	Suitable *
Cebe et al. [13]	2018	No	No	Byzantine agreement protocol	No
Li et al. [16]	2018	No	No	Byzantine agreement protocol	No
Kang et al. [14]	2019	No	No	Delegated Proof-of-Stake	No
Yang et al. [15]	2019	No	No	Proof-of-Event	No
Lei et al. [25]	2019	No	No	Proof of Work	No
Wang et al. [26]	2019	No	No	Proof of Reputation	No
Kaur et al. [27]	2019	No	No	Practical Byzantine Fault Tolerance	No
DeliveryCoin	/	Yes	Yes	UAV-aided forwarding algorithm	Yes

* Suitable for UAV-based delivery system; ** Achieving consensus between devices.

6. Conclusions

In this paper, we propose a new intrusion detection system (IDS) and Blockchain-based delivery framework, called DeliveryCoin, for drone-delivered services. The proposed DeliveryCoin framework combines hash functions and short signatures without random oracles and the Strong Diffie–Hellman (SDH) assumption in bilinear groups to achieve privacy-preservation. In addition, achieving consensus inside the blockchain-based delivery platform is performed over a UAV-aided forwarding mechanism. In order to detect self-driving network attacks as well as false transactions between self-driving nodes, an IDS system is integrated into DeliveryCoin in each macro eNB (5G). Furthermore, extensive simulations are conducted in order to evaluate the efficiency of DeliveryCoin. In future work, we will exploit security and efficiency issues of Edge computing in DeliveryCoin for drone-delivered services.

Author Contributions: Conceptualization, M.A.F. and L.M.; Methodology, M.A.F. and L.M.; Software, M.A.F. and L.M.; Validation, M.A.F. and L.M.; formal analysis, M.A.F. investigation, M.A.F. and L.M.; resources, M.A.F. and L.M.; data curation, M.A.F. and L.M.; writing—original draft preparation, M.A.F. and L.M.; writing—review and editing, L.M.; visualization, M.A.F. and L.M.; supervision, L.M.

Funding: This research received no external funding.

Conflicts of Interest: All authors declare no conflict of interest.

References

1. The Path to Autonomous Driving. Available online: https://www.bmw.com/en/automotive-life/autonomous-driving.html (accessed on 23 April 2019).
2. Amadeo, M.; Campolo, C.; Molinaro, A. Information-centric networking for connected vehicles: A survey and future perspectives. *IEEE Commun. Mag.* **2016**, *54*, 98–104. [CrossRef]
3. Kaiwartya, O.; Abdullah, A.H.; Cao, Y.; Altameem, A.; Prasad, M.; Lin, C.T.; Liu, X. Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects. *IEEE Access* **2016**, *4*, 5356–5373. [CrossRef]
4. Unmanned Aerial Vehicles Ready for Take-Off? Available online: https://www.logistics.dhl/global-en/home/insights-and-innovation/thought-leadership/trend-reports/unmanned-aerial-vehicles.html (accessed on 24 April 2019).
5. Menouar, H.; Guvenc, I.; Akkaya, K.; Uluagac, A.S.; Kadri, A.; Tuncer, A. UAV-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Commun. Mag.* **2017**, *55*, 22–28. [CrossRef]
6. Drożdż, S.; Minati, L.; Oświęcimka, P.; Stanuszek, M.; Wątorek, M. Signatures of the Crypto-Currency Market Decoupling from the Forex. *arXiv* **2019**, arXiv:1906.07834.
7. Ferrag, M.A.; Derdour, M.; Mukherjee, M.; Derhab, A.; Maglaras, L.; Janicke, H. Blockchain Technologies for the Internet of Things: Research Issues and Challenges. *IEEE Internet Things J.* **2019**, *6*, 2188–2204, doi:10.1109/JIOT.2018.2882794. [CrossRef]
8. Yang, J.; He, S.; Xu, Y.; Chen, L.; Ren, J. A Trusted Routing Scheme Using Blockchain and Reinforcement Learning for Wireless Sensor Networks. *Sensors* **2019**, *19*, 970. [CrossRef] [PubMed]

9. Ferrag, M.A.; Maglaras, L.; Janicke, H. Blockchain and its role in the internet of things. In *Strategic Innovative Marketing and Tourism*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1029–1038.
10. Pieroni, A.; Scarpato, N.; Di Nunzio, L.; Fallucchi, F.; Raso, M. Smarter city: Smart energy grid based on blockchain technology. *Int. J. Adv. Sci. Eng. Inf. Technol* **2018**, *8*, 298–306. [CrossRef]
11. Derhab, A.; Guerroumi, M.; Gumaei, A.; Maglaras, L.; Ferrag, M.A.; Mukherjee, M.; Khan, F.A. Blockchain and Random Subspace Learning-Based IDS for SDN-Enabled Industrial IoT Security. *Sensors* **2019**, *19*, 3119. [CrossRef] [PubMed]
12. Ferrag, M.A.; Maglaras, L. DeepCoin: A Novel Deep Learning and Blockchain-Based Energy Exchange Framework for Smart Grids. *IEEE Trans. Eng. Manag.* **2019**, 1–13. doi:10.1109/TEM.2019.2922936. [CrossRef]
13. Cebe, M.; Erdin, E.; Akkaya, K.; Aksu, H.; Uluagac, S. Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles. *IEEE Commun. Mag.* **2018**, *56*, 50–57. [CrossRef]
14. Kang, J.; Xiong, Z.; Niyato, D.; Ye, D.; Kim, D.I.; Zhao, J. Towards Secure Blockchain-enabled Internet of Vehicles: Optimizing Consensus Management Using Reputation and Contract Theory. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2906–2920. [CrossRef]
15. Yang, Y.T.; Chou, L.D.; Tseng, C.W.; Tseng, F.H.; Liu, C.C. Blockchain-Based Traffic Event Validation and Trust Verification for VANETs. *IEEE Access* **2019**, *7*, 30868–30877. [CrossRef]
16. Li, L.; Liu, J.; Cheng, L.; Qiu, S.; Wang, W.; Zhang, X.; Zhang, Z. CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2204–2220. [CrossRef]
17. CSE-CIC-IDS2018. Available online: https://www.unb.ca/cic/datasets/ids-2018.html (accessed on 30 May 2019).
18. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [CrossRef]
19. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.; Debbah, M. A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems. *IEEE Commun. Surv. Tutor.* **2019**, 1, doi:10.1109/COMST.2019.2902862. [CrossRef]
20. Mozaffari, M.; Kasgari, A.T.Z.; Saad, W.; Bennis, M.; Debbah, M. Beyond 5G with UAVs: Foundations of a 3D wireless cellular network. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 357–372. [CrossRef]
21. Boneh, D.; Boyen, X. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptol.* **2008**, *21*, 149–177. [CrossRef]
22. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, LA, USA, 22–25 February 1999; Volumn 99, pp. 173–186.
23. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.
24. Amir, Y.; Coan, B.; Kirsch, J.; Lane, J. Prime: Byzantine replication under attack. *IEEE Trans. Dependable Secure Comput.* **2011**, *8*, 564–577. [CrossRef]
25. Lei, A.; Cao, Y.; Bao, S.; Li, D.; Asuquo, P.; Cruickshank, H.; Sun, Z. A blockchain based certificate revocation scheme for vehicular communication systems. *Future Gener. Comput. Syst.* **2019**. [CrossRef]
26. Wang, Y.; Su, Z.; Zhang, N. BSIS: Blockchain based Secure Incentive Scheme for Energy Delivery in Vehicular Energy Network. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3620–3631. [CrossRef]
27. Kaur, K.; Garg, S.; Kaddoum, G.; Gagnon, F.; Ahmed, S.H. Blockchain-based Lightweight Authentication Mechanism for Vehicular Fog Infrastructure. *arXiv* **2019**, arXiv:1904.01168.