Article

COVID-19 Vehicle Based on an Efficient Mutual Authentication Scheme for 5G-Enabled Vehicular Fog Computing

Mahmood A. Al-Shareeda and Selvakumar Manickam *

National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, George Town 11800, Penang, Malaysia
* Correspondence: selva@usm.my

Abstract: The COVID-19 pandemic is currently having disastrous effects on every part of human life everywhere in the world. There have been terrible losses for the entire human race in all nations and areas. It is crucial to take good precautions and prevent COVID-19 because of its high infectiousness and fatality rate. One of the key spreading routes has been identified to be transportation systems. Therefore, improving infection tracking and healthcare monitoring for high-mobility transportation systems is impractical for pandemic control. In order to enhance driving enjoyment and road safety, 5G-enabled vehicular fog computing may gather and interpret pertinent vehicle data, which open the door to non-contact autonomous healthcare monitoring. Due to the urgent need to contain the automotive pandemic, this paper proposes a COVID-19 vehicle based on an efficient mutual authentication scheme for 5G-enabled vehicular fog computing. The proposed scheme consists of two different aspects of the special flag, SF = 0 and SF = 1, denoting normal and COVID-19 vehicles, respectively. The proposed scheme satisfies privacy and security requirements as well as achieves COVID-19 and healthcare solutions. Finally, the performance evaluation section shows that the proposed scheme is more efficient in terms of communication and computation costs as compared to most recent related works.

Keywords: COVID-19 vehicle; fog server; 5G-enabled vehicular network; authentication

1. Introduction

A major worldwide health crisis unlike any other in history is currently affecting everyone on Earth, causing chaos in people’s lives and spreading agony. The World Health Organization (WHO) has described the coronavirus disease COVID-19 as constituting fatal and highly contagious pandemic [1–4]. It has been responsible for thousands of deaths worldwide and is still wreaking havoc on communities [5–7]. The majority of countries have to date deployed tight detection and surveillance methods for public spaces such as airports and train stations, as well as means of public transit including airplanes, subways, and trains [8–10].

The special benefits of the fifth-generation 5G-enabled vehicular network could be taken into consideration with the pressing needs for pandemic control [11–14]. That is, the two key components of COVID-19-control—advanced healthcare monitoring and infection tracking—could be carried out by a 5G-enabled vehicular network without requiring direct human contact. As a result, the working staff is spared the perilous task of conducting surveillance at each inspection station. Instead, if required, each instance of fog computing might be given the role of the automatic checkpoint. In densely populated metropolitan cities and regions, the 5G-enabled vehicular network is regarded as a crucial element of the future vehicular ad hoc network (VANET) in order to deliver innovative vehicular data processing and traffic management analyses [15]. A major vehicular network infrastructure is composed of a 5G-base station, fog server, trusted authority (TA), and terminal vehicle equipped with the onboard unit (OBU) [16,17]. Vehicular networks provide for two different types of data transfer: vehicle-to-vehicle and vehicle-to-fog server through a 5G base station.
In order to address security challenges in the vehicular network, numerous studies using various safety procedures and cryptographic algorithms have recently been conducted. However, the criteria required for the 5G-enabled vehicular network’s actual implementation have not yet been fully taken into account. Vehicular networks’ prospective applications and useful uses should be investigated in light of the COVID-19 pandemic’s urgent global predicament. Measurements for vehicular networks and their related extensions should be developed now that the transportation system has emerged as one of the most hazardous scenarios for viral surveillance and infection tracking.

Due to the urgent need to contain the automotive pandemic, this paper will propose a COVID-19 vehicle-based efficient mutual authentication scheme for 5G-enabled vehicular fog computing. The proposed scheme consists of two different aspects according to a special flag (SF) value that involves a mutual authentication process. Based on the SF value, we consider the type of vehicle with SF = 0 and SF = 1 as the normal and COVID-19 vehicles, respectively. The main contributions of this paper are listed as follows:

- COVID-19 vehicle-based efficient mutual authentication scheme is proposed for 5G-enabled vehicular fog computing.
- A scheme does not only satisfy privacy and security requirements but also achieves COVID-19 and healthcare solutions.
- Performance evaluation section shows that the proposed scheme is more efficient in terms of communication and computation costs as compared to most recent related works.

The remainder of this paper is organized as follows: In Section 2, we review some mutual authentication schemes for vehicular networks. Section 3 describes the system model and mathematical methods of our proposal. Section 4 proposes a COVID-19 vehicle-based scheme for 5G-enabled vehicular fog computing. Section 5 evaluates security analysis and the comparison of proposal. Section 6 analyses the communication and computation overhead. Lastly, we provide the conclusion of this paper in Section 7.

2. Related Work

In this section, we review some mutual authentication schemes for vehicular networks as follows. Wang et al. [18] designed a mutual authentication system based on local identity by assigning unique long-term certification from TA to RSU and a vehicle in the enrolment process. Ming et al. [19] designed certificateless cryptography to provide communication security in a vehicular network. This scheme can simultaneously verify a large number of received messages by RSU. Al-Shareeda et al. [20] designed a privacy-preserving communication scheme by injecting fake messages to achieve unobservability requirements. This scheme has massive overhead in terms of communication and computation costs. For V2V communication, Ali and Li [21] proposed a signature based on an identity scheme using a secure hash function in high-density areas with high traffic. Zhang et al. [22] constructed a fuzzy logic mathematical approach to share security data among vehicles in a group based on a 5G-enabled model. Li et al. [23] presented a provable authentication scheme to provide both the security and privacy required for vehicular networks. Cui et al. [24] designed a content-sharing scheme for reliable communications in 5G-enabled vehicular networks. This scheme picked sophisticated proxy vehicles and requested them for content services. Alshudukhi et al. [25] constructed combined schemes between RSU-based and OBU-based schemes by exchanging temporary keys to sign messages and verify signatures during a short period. Al-Shareeda et al. [26] proposed a data-sharing scheme to secure 5G-enabled vehicular networks without using roadside units (RSUs). This scheme preloaded a large number of pseudonym IDs and relevant private keys to the registered vehicle during the key generation phase. However, this scheme has a massive overhead in verifying the messages exchanged among vehicles in terms of single verification and batch verification.

However, these mutual authentication schemes have massive performance overheads in terms of communication and computation costs. Additionally, none of these schemes
address the COVID-19 virus and healthcare solutions for vehicular networks to exchange messages with trusted parts. In order to cope with these issues, this paper will propose efficient mutual authentication to sign messages and verify the signature. Additionally, our proposal consists of two different aspects according to the special flag (SF) value that involves a mutual authentication process. Based on the SF value, we consider the type of vehicle to be a normal vehicle when SF = 0 and a COVID-19 vehicle when SF = 1.

3. Background

This section introduces the system model and mathematical method used in the proposed scheme as follows:

3.1. System Model

As shown in Figure 1, the system model of the proposed scheme consists of four main entities, namely the trusted authority (TA), fog server, 5G-base station (5G-BS), and onboard unit (OBU) that are equipped in each vehicle. The distribution of these entities is as follows.

- **Trusted Authority (TA):** It is the sole authority that is a trustworthy third party and can decipher an OBU’s identity from encryption. It is in charge of producing system parameters and has significant computing and storage capabilities.
- **Fog Server:** The fog server is regarded as a completely reliable entity that will assist TA in disclosing the names of the signers in our endeavour. In order to generate the pseudonym IDs of the vehicles over mutual authentication via 5G-BS, the fog server has its master key preloaded by TA. Our work relies on the public key of the fog server for the verification process.
- **5G-Base Station (5G-BS):** The 5G-BS is a reliable infrastructure that has been placed beside roads. Without any storage or computing, it functions as a bridge between entities.
- **Onboard Unit (OBU):** OBUs are installed in every car; they are secure and cannot be removed or interfered with. OBUs are wireless logical units that use the DRSC and 5G protocols to communicate with other OBUs and fog servers via 5G-BS, respectively. In this paper, there are two types of vehicles called normal vehicles and COVID-19 vehicles. In this paper, the terms of CVPS and NVPS will be used to refer to COVID-19 vehicles and normal vehicles, receptively.
3.2. Mathematical Methods Used

In this section, we describe the mathematical methods used in the proposed scheme as follows.

3.2.1. Elliptic Curve Cryptography (ECC)

Elliptic curve cryptography (ECC) was first conceived in 1958 by Miller [27]. Designing security procedures and digital signatures now frequently uses this kind of cryptographic technology. For more details about ECC, we recommend reading [20,28,29].

3.2.2. Hash Cryptographic Function

One of a number of hash functions created by the US national security agency (NSA), Secured Hash Function 512, is a component of the US Federal Information Processing Standard [30–32].

4. Proposed Scheme

In this section, we describe our proposed COVID-19 vehicle based on an efficient mutual authentication scheme for 5G-enabled vehicular fog computing in detail. In this paper, seven phases are included in our proposal, namely setup, enrolment, mutual authentication, updating private key, message signing, signature verification, and identity revocation phases. The seven phases in our proposal consist of two different aspects according to the special flag (SF) value that involves the mutual authentication process. Based on the SF value, we consider the type of vehicle to be a normal vehicle when SF = 0 and a COVID-19 vehicle when SF = 1. In the setup phase, the two types of SF values (e.g., 0 and 1) are used in this phase, in which TA is in charge of creating system parameters and broadcasting to all fog servers and vehicles. Furthermore, the two types of SF values are used in the setup phase. In the enrolment phase, TA is in charge of registering participating vehicles by applying both SF values to output two types of pseudonym IDs. Once the vehicle wishes to authenticate itself with the system, the vehicle will send its security parameters (e.g., password, public key, and SF value) to a nearby fog server through the wide communication range of 5G-BS. Based on the SF value, the vehicle then obtains a pseudonym ID to conceal its true identity. In the mutual authentication phase, the enrolled vehicle will request the private key that is also based on the SF value. If the SF value equals 0, approval is granted for both fog servers and vehicles to broadcast messages, while if SF value equals 1, approval is allowed for only fog servers. In the message signing phase, the vehicle will start to create and sign a message to generate a signature tuple. Therefore, the verifier checks the validity and authenticity of the message based on the signature in the signature verification phase. In updating the private key phase, once the timestamp of the private key is close to being expired, the normal vehicle requests that its private key is renewed, while the COVID-19 vehicle skips this phase, which as a result, enhances performance and satisfies the security and privacy properties for all COVID-19 vehicles. Finally, in the identity revocation phase, our proposal has the ability to trace and revoke any attacker (e.g., malicious normal or COVID-19 vehicles) that appears hostile to or pierced by the system. More precisely, our proposal expels and prevents the malicious vehicle from updating its private key in the previous phase. Figures 2 and 3 show normal vehicle and COVID-19 vehicle processes, respectively. These phases can be described in more detail as follows:
4.1. Setup Phase

The TA executes this phase to create the public parameters of the system. To maintain the security of the system in 5G-enabled vehicular fog computing, TA always frequently updates system parameters. The process of this phase is as follows:

- TA defines the equation of the elliptic curve $EC: y^2 = x^3 + ax + b \mod p$, where $a, b \in \mathbb{Z}_q^*$.  
- TA chooses the pairs of large primary numbers (e.g., $p$ and $q$) based on an additive group $G$.  
- TA selects a random number $s$ as the system’s private key and computes the concerned system’s public key $Pub = s \cdot P$.  
- TA picks three hash cryptographic functions (e.g., $h_1, h_2, h_3$) as $h_1 : G \rightarrow \mathbb{Z}_q^*$, $h_2 : \{0,1\}^* \times \{0,1\}^* \times G \rightarrow \mathbb{Z}_q^*$, $h_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.  
- TA securely saves the system’s private key $s$ to all fog servers.  
- Finally, TA broadcasts the system’s parameters $(p, q, Pub, P, h_1, h_2, h_3)$ to all fog servers through wire communication.

4.2. Enrolment Phase

Any new user who wishes to join 5G-enabled vehicular fog computing must first complete a number of legality-checking tasks before registering. This process is as follows:
The user submits a joining message including the vehicle’s true identity $VTID_{v_i}$, password (e.g., $PW$), and an SF value to TA through a secured channel. Where values of $SF = 0$ and $SF = 1$ indicate the normal vehicle and COVID-19 vehicle, respectively.

- In the case that a value of $SF = 0$, TA first verifies the vehicle’s true identity $VTID_{v_i}$ and then computes the normal vehicle’s pseudonym ID $NVPS = h_1(VTID_{v_i}||s)$. Finally, TA preloads personal information $(VTID_{v_i}, NVPS, PW)$ to the TPD of OBU for a normal registered vehicle.
- In the case of a value of $SF = 1$, TA first verifies the vehicle’s true identity $VTID_{v_i}$ and then computes the COVID-19 vehicle’s pseudonym ID $CVPS = h_1(VTID_{v_i}||s)$. Finally, TA preloads the personal information $(VTID_{v_i}, CVPS, PW)$ to the TPD of OBU for the COVID-19 registered vehicle.
- Finally, TA saves the sensitive data $(VTID_{v_i}, NVPS, PW)$ and $(VTID_{v_i}, CVPS, PW)$ into a normal vehicle registration list and COVID-19 vehicle registration list, respectively.

### 4.3. Mutual Authentication Phase

Once vehicle $v_i$ wishes to authenticate itself with the system, the following process should be executed:

- Vehicle $v_i$ picks random number $r \in Z_q^*$ and calculates the two pseudonym IDs $PID_i (PID_{i,1} \text{ and } PID_{i,2})$ (as Equations (1) and (2), respectively) based on the SF value to the normal vehicle and COVID-19 vehicle, respectively) in order to conceal the vehicle’s true identity.

  \[
  PID_i = (PID_{i,1}, PID_{i,2}) = (r \cdot P, NVPS \oplus h_1(r \cdot Pub))
  \]

  \[
  PID_i = (PID_{i,1}, PID_{i,2}) = (r \cdot P, CVPS \oplus h_1(r \cdot Pub))
  \]

- Vehicle $v_i$ generates signature $\delta_{auth}$ as Equations (3) and (4) according to the normal or COVID-19 vehicle, respectively.

  \[
  \delta_{auth} = h_2(NVPS||PID_i||T_1)
  \]

  \[
  \delta_{auth} = h_2(CVPS||PID_i||T_1)
  \]

- Vehicle $v_i$ transmits $(PID_i, T_1, \delta_{auth})$ to the nearest fog server $FS_j$ through the wide-range communication of 5G-BS.

- Once the fog server $FS_j$ receives $(PID_i, T_1, \delta_{auth})$ from the vehicle $v_i$, the $FS_j$ first checks the freshness of the timestamp as Equation (5). Where $T_\gamma$ is the predefined delay time and $T_i$ is the received time of $(PID_i, T_1, \delta_{auth})$.

  \[
  T_i > T_r - T_\gamma
  \]

- Once $T_1$ is valid, the fog server $FS_j$ computes $NVPS = PID_{i,2} \oplus h_1(sPID_{i,1})$ or $CVPS = PID_{i,2} \oplus h_1(sPID_{i,1})$ to check the validity of signatures as Equation (6) or Equation (7).

  \[
  \delta_{auth} \overset{?}{=} h_2(NVPS||PID_i||T_1)
  \]

  \[
  \delta_{auth} \overset{?}{=} h_2(CVPS||PID_i||T_1)
  \]

- If the above equation is false, the fog server $FS_j$ discards the message; otherwise, it sends $(NVPS, T_2)$ or $(CVPS, T_2)$ to TA based on the value of SF.

- Once TA receives the security parameters from fog server $FS_j$, TA checks the newness of timestamp $T_2$ and then verifies the match stored values $NVPS$ or $CVPS$ into the normal vehicle registration list and COVID-19 vehicle registration list, respectively.

- TA sends valid or not valid to fog server $FS_j$ according to the above verification.
4.4. Updating Private Key Phase

In this section, when lifetime $T_{SK}$ is close to expiring, the normal vehicle only executes this phase to update its private key $SK_N$, while the COVID-19 vehicle skips this phase. The normal vehicle will update the private key with the nearest fog server through wide-range communication by 5G-BS without being required to touch TA. This process is as follows:

- If the message is verified, fog server $FS_j$ will generate the private key $SK_N$ (as Equation (8)) for the normal vehicle that has $SF = 0$ or $SK_C$ (as Equation (9)) for the COVID-19 vehicle that has $SF = 1$, where $T_{SK}$ is a lifetime of the private key.

$$SK_N = s \cdot h_2(PID_{i,1}||PID_{i,2}||T_{SK})$$

$$SK_C = s \cdot h_2(PID_{i,1}||PID_{i,2})$$

- The fog server $FS_j$ encrypts the private key as Equation (10) or Equation (11) based on the type of vehicle.

$$SK_N^{Enc} = SK_N \oplus h_2(s \cdot PID_{i,1})$$

$$SK_C^{Enc} = SK_C \oplus h_2(s \cdot PID_{i,1})$$

- The fog server $FS_j$ sends $(SK_N^{Enc}, T_{SK}, T_3, \delta_{auth}^{fog})$ and $(SK_C^{Enc}, T_3, \delta_{auth}^{fog})$ to the normal vehicle and COVID-19 vehicle, receptively, where $\delta_{auth}^{fog}$ is calculated as Equations (12) and (13).

$$\delta_{auth}^{fog} = h_2(SK_N||T_3||T_{SK})$$

$$\delta_{auth}^{fog} = h_2(SK_C||T_3)$$

- In case of the COVID-19 vehicle (e.g., $SF = 1$), the vehicle $v_i$ decrypts the private key $SK_N = SK_N^{Enc} \oplus h_2(r \cdot Pub)$ and checks the signature $\delta_{auth}^{fog} = h_2(SK_N||T_3||T_{SK})$.
- In the case of the COVID-19 vehicle (e.g., $SF = 1$), the vehicle $v_i$ decrypts the private key $SK_C = SK_C^{Enc} \oplus h_2(r \cdot Pub)$ and checks signature $\delta_{auth}^{fog} = h_2(SK_C||T_3||T_{SK})$.

Note that the normal vehicle that has an $SF = 0$ will exchange data among the vehicles using its private key $SK_N$, while not required to broadcast messages (e.g., the velocity, location, speed, direction) to others for COVID-19 vehicles. Thus, the COVID-19 vehicle only sends a message (e.g., healthcare information) to the nearest fog server through the communication range of 5G-BS using its private key $SK_C$.

4.4. Updating Private Key Phase

In this section, when lifetime $T_{SK}$ is close to expiring, the normal vehicle only executes this phase to update its private key $SK_N$, while the COVID-19 vehicle skips this phase. The normal vehicle will update the private key with the nearest fog server through wide-range communication by 5G-BS without being required to touch TA. This process is as follows:

- Vehicle $v_i$ selects random number $r_{new} \in Z^*_p$ and calculates the two pseudonym IDs $PID_{i,1}^{new}$, $PID_{i,2}^{new}$ with Equation (14) in order to conceal the vehicle’s true identity.

$$PID_{i,1}^{new} = (PID_{i,1}^{new}, PID_{i,2}^{new})$$

$$= (r_{new}, P, NVPS \oplus h_1(r_{new} \cdot Pub))$$

- Vehicle $v_i$ sends $(PID_{i,1}^{new}, T_{SK}, T_4, \delta_{new})$ to nearest the fog server $FS_j$, where $\delta_{new}$ is calculated as Equation (15).

$$\delta_{new} = SK_N + r \cdot h_2(PID_{i,1}^{new}||PID_{i,2}^{new}||T_4)$$

- Once receiving $(PID_{i,1}^{new}, T_{SK}, T_4, \delta_{new})$ from the normal vehicle $v_i$, the fog server $FS_j$ firstly checks the newness of the timestamp $T_4$ as Equation (5). Additionally, the fog server $FS_j$ checks the expiration time of $T_{SK}$.
• The fog server $FS_j$ checks the authenticity and validity of signature $\delta_{\text{new}}$ as Equation (16).

$$
\delta_{\text{new}} \cdot P = (SK_N + r \cdot h_2(PID_{i,1}^{\text{new}} || PID_{i,2}^{\text{new}} || T_4)) \cdot P
$$

$$
= (s \cdot h_2(PID_{i,1}^{\text{new}} || PID_{i,2}^{\text{new}} || T_{SK}) + r \cdot h_2(PID_{i,1}^{\text{new}} || PID_{i,2}^{\text{new}} || T_4)) \cdot P
$$

$$
= (h_2(PID_{i,1}^{\text{new}} || PID_{i,2}^{\text{new}} || T_{SK}) \cdot s \cdot P + h_2(PID_{i,1}^{\text{new}} || PID_{i,2}^{\text{new}} || T_4) \cdot r \cdot P)
$$

$$
= h_2(PID_{i,1}^{\text{new}} || PID_{i,2}^{\text{new}} || T_{SK}) Pub + h_2(PID_{i,1}^{\text{new}} || PID_{i,2}^{\text{new}} || T_4) PID_{i,1}
$$

(16)

• If Equation (16) holds, the fog server $FS_j$ prepares a new private key $SK_N^{\text{new}}$ as Equation (17) with a new life-time ($T_{SK}^{\text{new}}$).

$$
SK_N^{\text{new}} = s \cdot h_2(PID_{i,1}^{\text{new}} || PID_{i,2}^{\text{new}} || T_{SK}^{\text{new}})
$$

(17)

• The fog server $FS_j$ encrypts new private key $SK_N^{\text{new}}$ as $SK_N^{\text{new.enc}} = SK_N^{\text{new}} \oplus h_1(s \cdot PID_{i,1}^{\text{new}})$ and computes $\delta_{\text{fog}}^{\text{new}}$ as Equation (18).

$$
\delta_{\text{fog}}^{\text{new}} = h_2(SK_N^{\text{new.enc}} || T_5 || T_{SK}^{\text{new}})
$$

(18)

• The fog server $FS_j$ sends $(SK_N^{\text{new.enc}}, T_{SK}^{\text{new}}, T_5, \delta_{\text{fog}}^{\text{new}})$ into a normal vehicle $v_i$.

• Once receiving the parameters $(SK_N^{\text{new.enc}}, T_{SK}^{\text{new}}, T_5, \delta_{\text{fog}}^{\text{new}})$, the vehicle $v_i$ decrypts $SK_N^{\text{new.enc}}$ and checks $\delta_{\text{fog}}^{\text{new}}$ to accept the new private key with its lifetime.

Note that the normal vehicle in our proposed travels from fog servers to others via different 5G-BS within the VANET system. This means that a vehicle has the ability to renew its pseudonym ID and private key without communicating with TA. As a result, our proposal avoids the single point of failure.

4.5. Message Signing Phase

Once vehicle $v_i$ wishes to broadcast the message in an open channel environment of 5G-enabled vehicular fog computing, the vehicle must run this phase as follows:

• Vehicle $v_i$ generates message $M_i$ regarding its road status and current freshness timestamp $T_i$.

• Vehicle $v_i$ prepares two pseudonym IDs $PID_i$ ($PID_{i,1}$ and $PID_{i,2}$) and concerned private key $SK_N$ which was obtained from fog sever.

• Vehicle $v_i$ computes a message signature $\delta_i$ as Equations (19) and (20) for normal vehicle and COVID-19 vehicle, respectively.

$$
\delta_i = SK_N + r \cdot h_3(PID_{i,1} || PID_{i,2} || M_i || T_i)
$$

(19)

$$
\delta_i = SK_C + r \cdot h_3(PID_{i,1} || PID_{i,2} || M_i || T_i)
$$

(20)

• Vehicle $v_i$ then computes $\sigma_i = \delta_i \cdot P$, which is applied to reduce the number of multiplication operations of ECC. As a result, reducing the overhead of the system from the verifier side in our proposal.

• Normal vehicle $v_i$ broadcasts message-tuple ($M_i, PID_{i,1}, PID_{i,2}, T_i, T_{SK}, \sigma_i$) to other normal vehicles or nearby fog servers.

Note that only a normal vehicle executes this phase to sign message and sends it to other vehicles or fog servers, while a COVID-19 vehicle only sends the security parameters with the message to nearby a fog sever through wide-range communication of 5G-BS.

4.6. Signature Verification Phase

Once the recipient (fog server or vehicle) receives message-tuple ($M_i, PID_{i,1}, PID_{i,2}, T_i, T_{SK}, \sigma_i$) from vehicle $v_i$, the verifier recipients should authenticate and validate the sent message before accepting message $M_i$ as follows:
• Checker tests the freshness of timestamp $T_i$ of message-tuple $(M_i, PID_{i,1}, PID_{i,2}, T_i, T_{SK}, \sigma_i)$ as shown in Equation (5) in order to detect replay attacks.

• Checker uses one of the following processes (single-signature verification or batch signature verification) in order to detect modification, forgery, or MITM attacks.

• Single signature verification process: checker tests whether Equation (21) holds or not.

$$\sigma_i = h_2(PID_{i,1}||PID_{i,2}||T_{SK}) \cdot Pub + h_3(PID_{i,1}||PID_{i,2}||M_i||T_i) \cdot PID_{i,1}$$  \quad (21)

• Batch signature verification process: checker tests whether Equation (22) holds or not.

$$\sum_{i=1}^{n} \sigma_i = \sum_{i=1}^{n} h_2(PID_{i,1}||PID_{i,2}||T_{SK}) \cdot Pub + \sum_{i=1}^{n} h_3(PID_{i,1}||PID_{i,2}||M_i||T_i) \cdot PID_{i,1}$$  \quad (22)

4.7. Identity Revocation Phase

In this phase, TA does not only trace the attacker (normal malicious vehicle or malicious COVID-19 vehicle), but also revokes the identity of the malicious vehicle from obtaining the VANET service. This process is as follows:

• Once reporting the forge message-tuple $(M_i, PID_{i,1}, PID_{i,2}, T_i, T_{SK}, \sigma_i)$, the fog server computes $NVPS$ for the normal vehicle or $CVPS$ for COVID-19 vehicle as Equation (23) or Equation (24), respectively.

$$NVPS = PID_{i,2} \oplus h_1(s \cdot PID_{i,1})$$  \quad (23)

$$CVPS = PID_{i,2} \oplus h_1(s \cdot PID_{i,1})$$  \quad (24)

• The fog server sends $NVPS$ or $CVPS$ to TA via a secure channel (wired technology).

• TA identifies $NVPS$ or $CVPS$ as matched and stored in a normal vehicle registration list and COVID-19 vehicle registration list, respectively, to revoke.

• TA deletes and revokes $NVPS$ or $CVPS$.

• TA sends the vehicle revocation list into all fog servers.

Note that when the timestamp $T_{SK}$ of private key $SK_N$ is close to expiring, the vehicle should request a nearby fog server to update the parameters. In the case that a normal vehicle is revoked, the fog server will discard the process since it was revoked and identified on the vehicle revocation list. At the same time, the fog server will discard the message that was sent from the revoked COVID-19 vehicle before it is accepted.

5. Security Analysis and Comparison

This section proves the security analysis and comparison of the proposed scheme as follows.

5.1. Security Analysis

Our proposed scheme should be satisfying security and privacy requirements in the following steps:

• Authentication and integrity: Before accepting a message, our proposal checks the signature that was attached to a message-tuple $(M_i, PID_{i,1}, PID_{i,2}, T_i, T_{SK}, \sigma_i)$. It then only accepts messages that calculate by evaluating Equations (21) and (22). Accordingly, the requirements of a authentication and integrity are applied in our proposal.

• Privacy-preserving: The proposed scheme generates two random numbers $s$ and $r$ as $Pub$ and $PID_{i,1} = r \cdot P$, respectively. Hence, any attacker attempting to obtain $NVPS/CVPS$ from a message-tuple $(M_i, PID_{i,1}, PID_{i,2}, T_i, T_{SK}, \sigma_i)$ will not be capable of doing so without these two numbers. Since $PID_{i,2} = NVPS \oplus h_1(r \cdot Pub)$ and $PID_{i,2} = CVPS \oplus h_1(r \cdot Pub)$, it becomes a difficult problems. Accordingly, the requirement of privacy-preserving is applied in our proposal.
Traceability and revocation: Any attacker that attempts to send forged messages or interfere with the operation of the system can be blocked and have their registration revoked by the TA by tracing the message’s source. The vehicle that receives the forged message transmits it to the TA, which performs the steps in Section 4.7. Accordingly, the requirements of traceability and revocation are applied in our proposal.

Replay attack: Since the timestamp $T_i$ is included in a message-tuple $(M_i, PID_{i1}, PID_{i2}, T_i, TSK, \sigma_i)$, the proposed scheme can avoid replay attacks using Equation (5). Accordingly, a replay attack is resisted in our proposal.

Forgery attacks: Since the signature tuple is validated by the TA using Equations (21) and (22), no attacker may falsify the identity of a legitimate vehicle sending a message. Accordingly, the forgery attack is resisted in our proposal.

Modify Attacks: Similar to the forgery attack, it needs an attacker to forge a signature tuple that is validated by computing (21) and (22). This is impossible to fake computationally. Hence, our technique is protected against this attack.

Man-in-the-middle attack: Because the vehicles communicate directly with one another and are shielded from interference, these types of attacks are not viable.

5.2. Security Comparison

This section evaluates and compares the proposed scheme and other related schemes in terms of privacy and security requirements. Table 1 lists the privacy and security comparison. As shown in Table 1, none of these schemes address the COVID-19 virus and solutions for vehicular networks to exchange messages with trusted parts. In order to cope with these issues, this paper proposes efficient mutual authentication that consists of two different aspects according to the special flag (SF) value that involves the mutual authentication process. Based on the SF value, we consider the type of vehicle to be a normal vehicle when SF = 0 and a COVID-19 vehicle when SF = 1.

Table 1. Comparison of Privacy and Security Requirements.

| Requirements                  | [22] | [23] | [24] | [25] | [26] | Our |
|-------------------------------|------|------|------|------|------|-----|
| Privacy-preserving Authentication | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| Integrity                     | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| Security attacks resistance   | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| Traceability                  | ✓    | ✓    | ✓    | ✓    | ✓    | ✓   |
| Revocability                  | x    | x    | x    | x    | x    | x   |
| Not RSU-based                 | ✓    | x    | x    | x    | x    | ✓   |
| COVID-19 solution             | x    | x    | x    | x    | x    | ✓   |
| Healthcare solutions          | x    | x    | x    | x    | x    | ✓   |

6. Performance Evaluation

6.1. Testbed Experiments Using MIRACL

This section presents the test-bed experiments to estimate the running time needed for various cryptographic operations used in the proposed scheme and existing related schemes utilizing the well-known “MIRACL Crypto SDK [33]” which is a C/C++ based library of programming software. For simplicity, the following notations used in this paper are:

- $T_{mul}$ denotes the estimated running time needed for ECC scalar multiplication operation;
- $T_{add}$ denotes the estimated running time needed for the ECC point addition operation $P + Q$;
- $T_h$ denotes the estimated running time needed for the secure cryptographic hash function.

Under this scenario, we have presented the computer setting as follows: “Model: Desktop, Processor: AMD Ryzen 7 5800H with Radeon Graphics, CPU Architecture:
64 bits, OS: Windows 11 Home Single Language with 16 GB memory”. There have been
1000 runs of each primitive. For each primitive, the average times in milliseconds are noted.
The experiment’s results based on MIRACL are displayed in Table 2.

### Table 2. Notation with its Costs of Running Time.

| Notation | Running Time |
|----------|--------------|
| \(T_{\text{mul}}\) | 0.6718 ms |
| \(T_{\text{add}}\) | 0.0031 ms |
| \(T_h\) | 0.0001 ms |

6.2. Computational Cost and Comparison

In this section, we estimate the computation cost of the operations used in single-
message signing, single-message verification, and batch messages verification for our
proposed and existing ECC-based related schemes. Table 3 lists a comparison of the
computational costs.

### Table 3. The Cost of Computation of Authentication Schemes.

| Scheme | MsgSign Phase | SigVerify Phase | BSigVerify Phase |
|--------|---------------|----------------|------------------|
| [22]   | \(2T_h + 2T_{\text{mul}} \approx 2 \times 0.0001 + 2 \times 0.6718 \approx 1.3456 \) ms | \(2T_{\text{add}} + 2T_h + 3T_{\text{mul}} \approx 2 \times 0.0031 + 2 \times 0.0001 + 3 \times 0.6718 \approx 2.0236 \) ms | \(nT_{\text{add}} + 2nT_h + (n + 2)T_{\text{mul}} \approx 1.3436 + 1.3487n \) ms |
| [23]   | \(2T_h + 2T_{\text{mul}} \approx 2 \times 0.0001 + 1 \times 0.6718 \approx 0.6729 \) ms | \(1T_{\text{add}} + 2T_h + 4T_{\text{mul}} \approx 1 \times 0.0031 + 2 \times 0.0001 + 3 \times 0.6718 \approx 0.6718 \) ms | \(nT_{\text{add}} + 2nT_h + (n + 2)T_{\text{mul}} \approx 1.3436 + 1.3487n \) ms |
| [24]   | \(3T_h + 3T_{\text{mul}} \approx 3 \times 0.0001 + 3 \times 0.6718 \approx 2.0157 \) ms | \(1T_{\text{add}} + 2T_h + 4T_{\text{mul}} \approx 0.0031 + 2 \times 0.0001 + 3 \times 0.6718 \approx 0.6718 \) ms | \(n - 1T_{\text{add}} + 2nT_h + (n + 2)T_{\text{mul}} \approx 1.3405 + 0.6782n \) ms |
| [25]   | \(2T_h + 2T_{\text{mul}} \approx 2 \times 0.0001 + 2 \times 0.6718 \approx 1.3438 \) ms | \(1T_{\text{add}} + 2T_h + 4T_{\text{mul}} \approx 0.0031 + 2 \times 0.0001 + 3 \times 0.6718 \approx 0.6718 \) ms | \(nT_{\text{add}} + 2nT_h + (n + 2)T_{\text{mul}} \approx 1.3436 + 0.6782n \) ms |
| [26]   | \(2T_h + 1T_{\text{mul}} \approx 2 \times 0.0001 + 0.6718 \approx 0.672 \) ms | \(1T_{\text{add}} + 2T_h + 4T_{\text{mul}} \approx 0.0031 + 2 \times 0.0001 + 4 \times 0.6718 \approx 2.0187 \) ms | \(nT_{\text{add}} + 2nT_h + (2n + 2)T_{\text{mul}} \approx 1.3436 + 1.3469n \) ms |
| Our    | \(1T_{\text{add}} + 1T_h + 1T_{\text{mul}} \approx 0.0031 + 0.0001 + 0.6718 \approx 0.675 \) ms | \(1T_{\text{add}} + 2T_h + 2T_{\text{mul}} \approx 0.0031 + 2 \times 0.0001 + 2 \times 0.6718 \approx 0.6718 \approx 1.3469 \) ms | \((n + 1)T_{\text{add}} + nT_h + 2T_{\text{mul}} \approx 1.3467 + 0.0032n \) ms |

The entity in the scheme of Zhang et al. [22] requires two general hash functions and
two operations of scalar multiplication for a single-message signing. Accordingly,
the whole computation overhead is \(2T_h + 2T_{\text{mul}} \approx 2 \times 0.0001 + 2 \times 0.6718 \approx 1.3456 \) ms,
while an entity needs two operations of point additions, two general hash functions,
and three operations of scalar multiplication for single-message verification. Accordingly, the
whole computation overhead is \(2T_{\text{add}} + 2T_h + 3T_{\text{mul}} \approx 2 \times 0.0031 + 2 \times 0.0001 + 3 \times 0.6718 \approx 2.0236 \) ms. Additionally, the entity needs \(n\) operations of point addition, \(2n\) general hash
functions, and \((2n + 2)\) operations of scalar multiplication for batch message verification. Accordingly, the whole computation overhead is \(nT_{\text{add}} + 2nT_h + (2n + 2)T_{\text{mul}} \approx 1.3436 + 1.3487n \) ms.

The entity in the scheme of Li et al. [23] requires two general hash functions and
one operation of scalar multiplication for single-message signing. Accordingly, the whole
computation overhead is \(2T_h + 2T_{\text{mul}} \approx 2 \times 0.0001 + 1 \times 0.6718 \approx 0.6729 \) ms, while the entity
needs one operation of point additions, two general hash functions, and four operations of
scalar multiplication for single-message verification. Accordingly, the whole computation

overhead is $1T_{add} + 2T_h + 4T_{mul} \approx 1 \times 0.0031 + 2 \times 0.6718 \approx 2.0236$ ms. Additionally, the entity needs $(n)$ operations of point addition, $2n$ general hash functions, and $(n + 2)$ operations of scalar multiplication for batch message verification. Accordingly, the whole computation overhead is $nT_{add} + 2nT_h + (n + 2)T_{mul} \approx 1.3436 + 1.3487n$ ms.

The entity in the scheme of Cui et al. [24] requires three general hash functions and three operations of scalar multiplication for a single-message signing. Accordingly, the whole computation overhead is $3T_h + 3T_{mul} \approx 3 \times 0.6718 \approx 2.0157$ ms, while the entity needs one operation of point addition, two general hash functions, and three operations of scalar multiplication for single-message verification. Accordingly, the whole computation overhead is $1T_{add} + 2T_h + 3T_{mul} \approx 0.0031 + 2 \times 0.6718 \approx 2.0187$ ms. Additionally, the entity needs $(n - 1)$ operations of point addition, $2n$ general hash functions, and $(n + 2)$ operations of scalar multiplication for batch message verification. Accordingly, the whole computation overhead is $n - 1T_{add} + 2nT_h + (n + 2)T_{mul} \approx 1.3405 + 0.6782n$ ms.

The entity in the scheme of Alshuduki et al. [25] requires two general hash functions and two operations of scalar multiplication for single-message signing. Accordingly, the whole computation overhead is $2T_h + 2T_{mul} \approx 2 \times 0.6718 \approx 1.3438$ ms, while the entity needs one operation of point addition, two general hash functions, and three operations of scalar multiplication for single-message verification. Accordingly, the whole computation overhead is $1T_{add} + 2T_h + 4T_{mul} \approx 0.0031 + 2 \times 0.6718 + 3 \times 0.6718 \approx 2.0187$ ms. Additionally, the entity needs $(n)$ operations of point addition, $2n$ general hash functions, and $(n + 2)$ operations of scalar multiplication for the batch message verification. Accordingly, the whole computation overhead is $nT_{add} + 2nT_h + (n + 2)T_{mul} \approx 1.3436 + 0.6782n$ ms.

The entity in the scheme of Al-Shareeda et al. [26] requires two general hash functions and one operation of scalar multiplication for a single-message signing. Accordingly, the whole computation overhead is $2T_h + T_{mul} \approx 2 \times 0.6718 \approx 1.3436$ ms, while the entity needs one operation of point addition, two general hash functions, and four operations of scalar multiplication for single-message verification. Accordingly, the whole computation overhead is $1T_{add} + 2T_h + 4T_{mul} \approx 0.0031 + 2 \times 0.6718+ 4 \times 0.6718 \approx 2.6905$ ms. Additionally, the entity needs $(n)$ operations of point addition, $2n$ general hash functions, and $(2n + 2)$ operations of scalar multiplication for batch message verification. Accordingly, the whole computation overhead is $nT_{add} + 2nT_h + (2n + 2)T_{mul} \approx 1.3436 + 1.3469n$ ms.

The entity in our proposed scheme requires one operation of point addition, one general hash function, and one operation of scalar multiplication for a single-message signing. Accordingly, the whole computation overhead is $1T_{add} + T_h + T_{mul} \approx 0.0031 + 0.6718 \approx 0.675$ ms, while the entity needs one operation of point addition, two general hash functions, and two operations of scalar multiplication for single-message verification. Accordingly, the whole computation overhead is $1T_{add} + 2T_h + 2T_{mul} \approx 0.0031 + 2 \times 0.6718 \approx 1.3469$ ms. Additionally, the entity needs $(n + 1)$ operations of point addition, $n$ general hash functions, and two operations of scalar multiplication for batch message verification. Accordingly, the whole computation overhead is $(n + 1)T_{add} + nT_h + 2T_{mul} \approx 1.3467 + 0.0032n$ ms.

6.3. Communication Cost and Comparison

In this section, we estimate the communication costs of the item size used as the final exchanged message among entities for our proposed and existing ECC-based related schemes. The major concentration is the communication overhead consisting of the pseudonym IDs, signatures, and timestamps for the message-signature tuples. Referring to [13,17,34], we suppose that the bit lengths for the timestamp, hash function, element in $Z_q^*$, and element in $G$ are 32, 160, 160, 320 bits, respectively. Table 4 lists a comparison of communication costs of the proposed scheme and related works.

In scheme of Zhang et al. [22], the entity broadcasts a message-tuple $\{ID_{i,j}, ID_{i,j}, M, T, \sigma\}$ to other vehicles, where $320 \times 2 + 32 + 160 \approx 832$. In the scheme of Li et al. [23], the entity broadcasts a message-tuple $\{M_i, PID_{i,j}, PK_{i,j}, R_{ij}, T_{i}, sig_{i}\}$ to other vehicles, where
In the scheme of Cui et al. [24], the entity broadcasts a message-tuple \(\{(DT_{ij}, D_j, PID_j, \delta_j, T_j)\}\) to other vehicles, where \(320 \times 3 + 160 + 32 \approx 1152\). In the scheme of Alshudukhi et al. [25], the entity broadcasts a \(\{(PsID^1_i, PsID^2_i, m_i, TS_i, \sigma_m)\}\) to other vehicles, where \(32 + 2 \times 160 + 1 \times 320 \approx 672\). In the scheme of Al-Shareeda et al. [26], the entity broadcasts a message-tuple \(\{(M_i, PK_i, PID_{ij}, D_i, T_i, \delta_i)\}\) to other vehicles, where \(2 \times 320 + 2 \times 160 + 32 \approx 992\) bits. In our proposal, the entity broadcasts a message-tuple \(\{(M_i, PID_{ij}, PID_{i,2}, T, T_{SK}, \sigma_i)\}\) to other vehicles, where \(2 \times 32 + 160 + 2 \times 320 \approx 864\) bits.

Hence, according to the above analysis, this paper proves that the communication costs of each element (as can be seen in Section 4.5) in message-tuple is lower compared with recent studies.

### Table 4. Comparison of Communication Costs.

| Scheme | Message-Signature Tuple | Size (bits) | \(n\) Size (bits) |
|--------|-------------------------|-------------|-------------------|
| [22]   | \(\{ID_{i,1}, ID_{i,2}, M, T, \sigma\}\) | \(320 \times 3 + 32 + 160 \approx 832\) | 832 \(n\) |
| [23]   | \(\{M_i, PID_{ij}, PK_{ij}, R_{ij}, sig_i\}\) | \(320 \times 3 + 160 + 32 \approx 1152\) | 1152 \(n\) |
| [24]   | \(\{DT_{ij}, D_j, PID_j, \delta_j, T_j\}\) | \(32 + 2 \times 160 + 2 \times 320 \approx 992\) | 992 \(n\) |
| [25]   | \(\{PsID^1_i, PsID^2_i, m_i, TS_i, \sigma_m\}\) | \(2 \times 320 + 2 \times 160 + 32 \approx 992\) | 992 \(n\) |
| [26]   | \(\{M_i, PK_{ij}, PID_{ij}, D_i, T_i, \delta_i\}\) | \(2 \times 320 + 2 \times 160 + 32 \approx 992\) | 992 \(n\) |
| Our    | \(\{M_i, PID_{ij}, PID_{i,2}, T, T_{SK}, \sigma_i\}\) | \(2 \times 32 + 160 + 2 \times 320 \approx 864\) | 864 \(n\) |

#### 6.4. Simulation Environment

This section implements the proposed scheme in a simulation environment in order to evaluate the performance.

As listed in Figure 4, the proposed scheme generates the traffic simulator and network simulator using SUMO [35] and OMNeT++ [36], receptively. Additionally, the proposed scheme applies tools and frameworks such as OpenStreetMap [37], GatcomSUMO [38], VEINS [39], FogNetSim++ [40], Simu5G [41], and MIRACL [33] to implement a simulation environment in an urban area for 5G-enabled vehicular fog computing. The simulation environment’s settings are listed in Table 5.

![Simulation Environment of Proposed Scheme](image)

Each cryptographic operation has its own distinct runtime, and that is what is used to calculate the total time. The overhead cost is the amount of time that has passed since the exit and the entrance (see Equation (25)).

\[
ET = \frac{1}{M} \sum_{i=1}^{n} M(T_{out}^i - T_{in}^i)
\]
where $T_{in}^i$ is the message entrance time $i$, $M$ is the message number, and $T_{out}^i$ is the message exit time $i$. Figure 5 depicts the average time to verify a single message between the proposed and existing schemes.

Table 5. Simulation Environment Parameters.

| Parameter          | Value                                      |
|--------------------|--------------------------------------------|
| Playground size    | $x = 3463$ m, $y = 4270$ m and $z = 50$ m |
| Maximum transmission| 20 mW                                      |
| Simulation time    | 200 s                                      |
| Bit rate           | 6 Mbps                                     |
| Physical layer     | IEEE 802.11p                               |
| Mac layer          | IEEE 1609.4                                |

Figure 5. Average Delay for Verifying a Single Message.

7. Conclusions

In this paper, the development of an efficient mutual authentication scheme for healthcare solutions in 5G-enabled vehicular fog computing places emphasis on controlling automotive pandemics in intelligent transportation systems. In the proposed scheme, there is no requirement to broadcast messages (e.g., velocity, location, speed, direction) to others for COVID-19 vehicles. Thus, the COVID-19 vehicle only sends a message (e.g., healthcare information) to the nearest fog server through the communication range of 5G-BS using its private key. Security analysis shows that the proposed scheme satisfies privacy and security requirements as well as achieves COVID-19 and healthcare solutions. The performance section evaluates whether the proposed scheme is more efficient in terms of communication and computation costs as compared to the most recent related works.

Author Contributions: Conceptualization, writing—review and editing, software, methodology, M.A.A.-S.; writing—original draft preparation, investigation, supervision, funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Vice Chancellor Initiative Allocation, Universiti Sains Malaysia, grant number 311/PNAV/4119101.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.
References

1. Velásquez, G. The World Health Organization Reforms in the Time of COVID-19. In *Vaccines, Medicines and COVID-19*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 93–108.

2. Al-Shareeda, M.A.; Manickam, S.; Saare, M.A.; Arjuman, N.C. Proposed security mechanism for preventing fake router advertisement attack in IPv6 link-local network. *Indones. J. Electr. Eng. Comput. Sci.* 2023, 29, 518–526. [CrossRef]

3. Zhang, F.; Karamagi, H.; Nsenga, N.; Nanyunja, M.; Karina, M.; Amanfo, S.; Chase-Topping, M.; Calder-Gerver, G.; McGibbon, M.; Huber, A.; et al. Predictors of COVID-19 epidemics in countries of the World Health Organization African Region. *Nat. Med.* 2021, 27, 2041–2047. [CrossRef] [PubMed]

4. Al-Shareeda, M.A.; Manickam, S.; Mohammed, B.A.; Al-Mekhlafi, Z.G.; Qtaish, A.; Alzahrani, A.J.; Alshammari, G.; Sallam, A.A.; Almekhlafi, K. Chebyshev Polynomial-Based Scheme for Resisting Side-Channel Attacks in 5G-Enabled Vehicular Networks. *Appl. Sci.* 2022, 12, 5939. [CrossRef]

5. Al-Shareeda, M.A.; Anbar, M.; Manickam, S.; Hasbullah, I.H. A Secure Pseudonym-Based Conditional Privacy-Preservation Authentication Scheme in Vehicular Ad Hoc Networks. *Sensors* 2022, 22, 1696. [CrossRef] [PubMed]

6. Leaby, A.K.; Yassin, A.; Hasson, M.; Rashid, A. Towards design strong emergency and COVID-19 authentication scheme in VANET. *Indones. J. Electr. Eng. Comput. Sci.* 2021, 21, 1808–1819. [CrossRef]

7. Al-Shareeda, M.A.; Manickam, S.; Mohammed, B.A.; Al-Mekhlafi, Z.G.; Qtaish, A.; Alzahrani, A.J.; Alshammari, G.; Sallam, A.A.; Almekhlafi, K. Cm-ppa: Chaotic map-based conditional privacy-preserving authentication scheme in 5g-enabled vehicular networks. *Sensors* 2022, 22, 5026. [CrossRef]

8. Al-Shareeda, M.A.; Manickam, S. MSR-DoS: Modular Square Root-based Scheme to Resist Denial of Service (DoS) Attacks in 5G-enabled Vehicular Networks *IEEE Access* 2022, 10, 120606–120615. [CrossRef]

9. Senapati, B.R.; Khilar, P.; Dash, T.; Swain, R.R. Vehicular Network based Emergency Data Transmission and Classification for Health Care System Using Support Vector Machine. 2022. Available online: https://assets.researchsquare.com/files/rs-1224943/v1_covered.pdf?c=1657042375 (accessed on 5 July 2022).

10. Al-Shareeda, M.A.; Manickam, S. Man-In-The-Middle Attacks in Mobile Ad Hoc Networks (MANETs): Analysis and Evaluation. *Symmetry* 2022, 14, 1543. [CrossRef]

11. Wang, L.; Alexander, C.A. Cyber security during the COVID-19 pandemic. *AIMS Electron. Electr. Eng.* 2021, 5, 146–157. [CrossRef]

12. Sun, X.; Wandelt, S.; Zheng, C.; Zhang, A. COVID-19 pandemic and air transportation: Successfully navigating the paper hurricane. *J. Air Transp. Manag.* 2021, 94, 102062. [CrossRef]

13. Alazzawi, M.A.; Al-behadili, H.A.; Srayyih Almalki, M.N.; Challoob, A.L.; Al-shareeda, M.A. ID-PPA: Robust identity-based privacy-preserving authentication scheme for vehicular ad hoc network. In Proceedings of the International Conference on Advances in Cyber Security, Penang, Malaysia, 8–9 December 2020; pp. 80–94.

14. Shen, J.; Duan, H.; Zhang, B.; Wang, J.; Ji, J.S.; Wang, J.; Pan, L.; Wang, X.; Zhao, K.; Ying, B.; et al. Prevention and control of COVID-19 in public transportation: Experience from China. *Environ. Pollut.* 2020, 266, 115291. [CrossRef] [PubMed]

15. Al-Shareeda, M.M.A.; Anbar, M.; Alazzawi, M.A.; Manickam, S.; Hasbullah, I.H. Security schemes based conditional privacy-preserving in vehicular ad hoc networks. *Indones. J. Electr. Eng. Comput. Sci.* 2021, 21, 479–488. [CrossRef]

16. Hamdi, M.M.; Mustafa, A.S.; Mahd, H.F.; Abood, M.S.; Kumar, C.; Al-shareeda, M.A. Performance Analysis of QoS in MANET based on IEEE 802.11 b. In Proceedings of the 2020 IEEE International Conference for Innovation in Technology (INOCON), Bengaluru, India, 6–8 November 2020; pp. 1–5.

17. Al-Shareeda, M.A.; Anbar, M.; Manickam, S.; Hasbullah, I.H.; Abdullah, N.; Hamdi, M.M.; Al-Hiti, A.S. NE-CPPA: A new and efficient conditional privacy-preserving authentication scheme for vehicular ad hoc networks (VANETs). *Appl. Math* 2020, 14, 1–10.

18. Wang, S.; Yao, N. LIAP: A local identity-based anonymous message authentication protocol in VANETs. *Comput. Commun.* 2017, 112, 154–164. [CrossRef]

19. Ming, Y.; Cheng, H. Efficient certificateless conditional privacy-preserving authentication scheme in VANETs. *Mob. Inf. Syst.* 2019, 2019. [CrossRef]

20. Al-Shareeda, M.A.; Anbar, M.; Manickam, S.; Yassin, A.A. Vypcs: Vanet-based privacy-preserving communication scheme. *IEEE Access* 2020, 8, 150914–150928. [CrossRef]

21. Ali, I.; Lawrence, T.; Li, F. An efficient identity-based signature scheme without bilinear pairing for vehicle-to-vehicle communication in VANETs. *J. Syst. Archit.* 2020, 103, 101692. [CrossRef]

22. Zhang, J.; Zhong, H.; Cui, J.; Tian, M.; Xu, Y.; Liu, L. Edge computing-based privacy-preserving authentication framework and protocol for 5g-enabled vehicular networks. *IEEE Trans. Veh. Technol.* 2020, 69, 7940–7954. [CrossRef]

23. Li, J.; Choo, K.K.R.; Zhang, W.; Kumari, S.; Rodrigues, I.J.; Khan, M.K.; Hogrefe, D. EPA-CPPA: An efficient, provably-secure and anonymous conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *Veh. Commun.* 2018, 13, 104–113. [CrossRef]

24. Cui, J.; Chen, J.; Zhong, H.; Zhang, J.; Liu, L. Reliable and Efficient Content Sharing for 5G-Enabled Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* 2020, 23, 1247–1259. [CrossRef]

25. Alshudukhi, J.S.; Al-Mekhlafi, Z.G.; Mohammed, B.A. A Lightweight Authentication With Privacy-Preserving Scheme for Vehicular Ad Hoc Networks Based on Elliptic Curve Cryptography. *IEEE Access* 2021, 9, 15633–15642. [CrossRef]
26. Al-Shareeda, M.A.; Manickam, S.; Mohammed, B.A.; Al-Mekhlafi, Z.G.; Qtaish, A.; Alzahrani, A.J.; Alshammari, G.; Sallam, A.A.; Almekhlafi, K. Provably Secure with Efficient Data Sharing Scheme for Fifth-Generation (5G)-Enabled Vehicular Networks without Road-Side Unit (RSU). *Sustainability* **2022**, *14*, 9961. [CrossRef]

27. Miller, V. Use of Elliptic Curves in Cryptography. In *Proceedings of the Conference on the Theory and Application of Cryptographic Techniques*, Linz, Austria, 1 January 2000; pp. 417–426.

28. Al-Shareeda, M.A.; Anbar, M.; Alazzawi, M.A.; Manickam, S.; Al-Hiti, A.S. LSWBVM: A lightweight security without using batch verification method scheme for a vehicle ad hoc network. *IEEE Access* **2020**, *8*, 170507–170518. [CrossRef]

29. Underwood, R.G. Elliptic Curves in Cryptography. In *Cryptography for Secure Encryption*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 271–296.

30. Long, S. A Comparative Analysis of the Application of Hashing Encryption Algorithms for MD5, SHA-1, and SHA-512. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2019; Volume 1314, p. 012210.

31. Ahmad, I.; Das, A.S. Hardware implementation analysis of SHA-256 and SHA-512 algorithms on FPGAs. *Comput. Electr. Eng.* **2005**, *31*, 345–360. [CrossRef]

32. NIST, U. Descriptions of SHA-256, SHA-384 and SHA-512. 2001. Available online: https://eips.ethereum.org/assets/eip-2680/sha256-384-512.pdf (accessed on 23 November 2022).

33. Scott, M. MIRACL-A Multiprecision Integer and Rational Arithmetic C/C++ Library. 2003. Available online: http://www.shamus.ie (accessed on 23 November 2022).

34. Al-Shareeda, M.A.; Anbar, M.; Hasbullah, I.H.; Manickam, S.; Hanshi, S.M. Efficient conditional privacy preservation with mutual authentication in vehicular ad hoc networks. *IEEE Access* **2020**, *8*, 144957–144968. [CrossRef]

35. Behrisch, M.; Bieker, L.; Erdmann, J.; Krajzewicz, D. SUMO–simulation of urban mobility: An overview. In *Proceedings of the SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, Barcelona, Spain, 23–29 October 2011.

36. Varga, A. Discrete event simulation system. In *Proceedings of the European Simulation Multi-conference (ESM’2001)*, Prague, Czech Republic, 6–9 June 2001; pp. 1–7.

37. Haklay, M.; Weber, P. Openstreetmap: User-generated street maps. *IEEE Pervasive Comput.* **2008**, *7*, 12–18. [CrossRef]

38. Abenza, P.P.G.; Malumbres, M.P.; Peral, P.P. 10 GatcomSUMO: A Graphical Tool for VANET Simulations Using SUMO and OMNeT+. SUMO 2017—Towards Simulation for Autonomous Mobility. 2017, p. 113. Available online: https://www.researchgate.net/publication/322750199_SUMO_2017_Towards_Simulation_for_Autonomous_Mobility (accessed on 23 November 2022).

39. Sommer, C.; German, R.; Dressler, F. Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Trans. Mob. Comput.* **2010**, *10*, 3–15. [CrossRef]

40. Qayyum, T.; Malik, A.W.; Khattak, M.A.K.; Khalid, O.; Khan, S.U. FogNetSim++: A toolkit for modeling and simulation of distributed fog environment. *IEEE Access* **2018**, *6*, 63570–63583. [CrossRef]

41. Nardini, G.; Sabella, D.; Sta, G.; Thakkar, P.; Virdis, A. Simu5G–An OMNeT++ library for end-to-end performance evaluation of 5G networks. *IEEE Access* **2020**, *8*, 181176–181191. [CrossRef]