mathematics

MDPI

Article

ANAA-Fog: A Novel Anonymous Authentication Scheme for 5G-Enabled Vehicular Fog Computing

Badiea Abdulkarem Mohammed [1], Mahmood A. Al-Shareeda [2,*], Selvakumar Manickam [2,*], Zeyad Ghaleb Al-Mekhlafi [1], Abdulaziz M. Alayba [1] and Amer A. Sallam [3]

[1] College of Computer Science and Engineering, University of Ha'il, Ha'il 81481, Saudi Arabia
[2] National Advanced IPv6 Centre (NAv6), Sains Malaysia University, Penang 11800, Malaysia
[3] Engineering and Information Technology College, Taiz University, Taiz 6803, Yemen
* Correspondence: alshareeda022@usm.my (M.A.A.); selva@usm.my (S.M.)

Abstract: Vehicular fog computing enabled by the Fifth Generation (5G) has been on the rise recently, providing real-time services among automobiles in the field of smart transportation by improving road traffic safety and enhancing driver comfort. Due to the public nature of wireless communication channels, in which communications are conveyed in plain text, protecting the privacy and security of 5G-enabled vehicular fog computing is of the utmost importance. Several existing works have proposed an anonymous authentication technique to address this issue. However, these techniques have massive performance efficiency issues with authenticating and validating the exchanged messages. To face this problem, we propose a novel anonymous authentication scheme named ANAA-Fog for 5G-enabled vehicular fog computing. Each participating vehicle's temporary secret key for verifying digital signatures is generated by a fog server under the proposed ANAA-Fog scheme. The signing step of the ANAA-Fog scheme is analyzed and proven secure with the use of the ProfVerif simulator. This research also satisfies privacy and security criteria, such as conditional privacy preservation, unlinkability, traceability, revocability, and resistance to security threats, as well as others (e.g., modify attacks, forgery attacks, replay attacks, and man-in-the-middle attacks). Finally, the result of the proposed ANAA-Fog scheme in terms of communication cost and single signature verification is 108 bytes and 2.0185 ms, respectively. Hence, the assessment metrics section demonstrates that our work incurs a little more cost in terms of communication and computing performance when compared to similar studies.

Keywords: fog computing; vehicular network; authentication and privacy; 5G technology; 5G-enabled vehicular fog computing

MSC: 94A60

Citation: Mohammed, B.A.; Al-Shareeda, M.A.; Manickam, S.; Al-Mekhlafi, Z.G.; Alayba, A.M.; Sallam, A.A. ANAA-Fog: A Novel Anonymous Authentication Scheme for 5G-Enabled Vehicular Fog Computing. *Mathematics* **2023**, *11*, 1446. https://doi.org/10.3390/math11061446

Academic Editors: Omprakash Kaiwartya, Houbing Song, Sushil Kumar, Ali Sadiq and Ahmad Fadhil Yusof

Received: 30 January 2023
Revised: 10 March 2023
Accepted: 14 March 2023
Published: 16 March 2023

1. Introduction

The World Health Organization (WHO) reports that annually 1.25 million individuals lose their lives in traffic-related incidents [1,2]. The fifth generation (5G) technology, vehicular networks, and fog computing have been deployed lately on a wide scale in several nations' transportation systems to improve driving safety and manage increasingly congested traffic scenarios. Vehicles equipped with a wireless device, known as onboard units (OBUs), are a type of intelligent transportation system (ITS) that collects, processes, and disseminates traffic data within the context of networked vehicles [3,4].

Participants in 5G-enabled vehicular fog computing communicate information about traffic conditions (such as road difficulties, congestion situations, and temperature conditions) and vehicle conditions (such as location, speed, traffic status, etc.) [5,6]. Emergency vehicles, such as traffic control centres, depend on these messages to make life-or-death decisions. Congestion and potential accidents will result if an attacker modifies or inserts

harmful messages into the network. It is crucial, then, that 5G-enabled vehicular fog computing pay close attention to privacy and security concerns [7,8].

Drivers' needs are changing as urban cars proliferate. Hence, the VANET under 5G mobile networks can meet current application requirements for capacity and coverage. Vehicular networks have many difficulties and opportunities. In general, 5G wireless networks have data transmission rates of 20 Gb/s and 100 Mb/s [9,10].

Using a fog server in place of an RSU is one way in which fog computing can help satisfy the need for adopting vehicular networks, as stated by the authors in [11]. The fog server is assumed to not be completely trustworthy despite its access to essential services, such as computation and storage. Meanwhile, writers mention [12] as evidence that fog computing can help meet the demand for rolling up 5G networks. Our study introduces a fog computing-based pseudonym authentication (FC-PA) method to reduce the load on 5G-enabled vehicle networks. For 5G-enabled vehicle networks, the authors in [13] propose a technique for removing pseudonyms via fog computing that is based on the Chebyshev polynomial.

A lot of authentication schemes fail to address privacy and security altogether. Additionally, performance efficiency is still more vulnerable. These techniques have massive performance efficiency issues withauthenticating and validating the exchanged messages. Therefore, this paper presents a novel anonymous authentication (ANAA-Fog) scheme to reduce the overhead of the system and achieve privacy and security requirements. The main contributions of this paper are listed as follows.

- A new ANAA-Fog scheme is proposed for 5G-enabled vehicular fog computing in which the trusted authority (TA) saves the master key in the fog server to generate the temporary secret key to each participating vehicle.
- The proposed ANAA-Fog scheme uses a fog server instead of RSU to generate and issue temporary keys for each vehicle located within the 5G-base station.
- By using a shared key, vehicle and fog servers can together achieve a mutual authentication process.
- Security analysis uses the ProfVerif simulator to prove the security of ANAA-Fog scheme formality. Additionally, this work satisfies authentication of the signer, integrity of the message, conditional privacy-preserving, unlinkability, traceability and revocability, and security attacks resistance in terms of modification, forgery, replay, and man-in-the-middle attacks.
- The efficiencies of our ANAA-Fog scheme in achieving privacy and security are dominated in terms of communication and computation overheads.

The rest of this paper is organized as follows: Section 3 provides the background. Section 2 reviews the existing schemes. The proposed ANAA-Fog scheme is explained in Section 4. Section 5 lists the numerical example of our approach. The security analysis of our work is shown in Section 6. Section 7 describes evaluation metrics. Lastly, the conclusions of this paper are provided in Section 8.

2. Related Work

2.1. Security and Privacy Research

Information shared by vehicles always involves driver safety; information requires validation and authentication before revealing the inside content. Zhong et al. [14] constructed a certificateless aggregate signature scheme with full aggregation to provide communication security in a vehicular network. Bayat et al. [15] constructed an anonymous authentication scheme based on the roadside unit (RSU) to authenticate vehicles during the joining process. Liu et al. [16] proposed a distributed computing based on a proxy-based authentication scheme to verify multiple messages with a verification function simultaneously. Asaar et al. [17] highlighted the limitation existing in the scheme of Liu et al. [16] that message authenticity is not satisfied, which is vulnerable to modification and forgery attacks. Li et al. [18] constructed a provable authentication scheme to provide both the privacy and security required in a vehicular network.

Recently, Zhang et al. [19] created a simple traffic route management system for fog-based VANETs. For this plan, automobiles will encrypt their travel plans with homomorphic encryption before transmitting them to a fog node. The fog node aggregates encrypted traffic data, which is then sent to the traffic management centre (TMC), where it is decrypted and used for traffic control without the TMC needing to know the specific routes taken by each vehicle. Cui et al. [20] created the Internet of Autonomous Vehicles (IoAV) paradigm to address the issues caused by these constraints. It is important to implement a trustworthy authentication mechanism that is applicable in IoAV to encourage safe remote control of the AV. We present a method for providing secure remote control features for AVs using authenticated key agreement (CMAKA) based on chaotic maps. Chen et al. [21] provided SAABS-CR, an efficient server-aided ABS that is also resistant to collusion and may be used for IoV. Server-assisted computing technology reduces the computational load on verifiers while remaining perfectly resistant to collusion attacks between signers and between the signer and the aided server.

2.2. Fog Computing Research

The traditional technology of cloud computing is not qualified for the case where an extension of information is generated. Xiao et al. [22] presented the concept of fog computing to the Internet of Things (IoT) area. This concept is acquired publicly and is growing and being applied in different service domains, including industrial IoT [23]. The fog computing of IoT indicates the producer and the consumer, i.e., some traditional cloud applications can be transferred to the fog server of the system, which can satisfy some valid effects, such as lower latency, better offloading, and so on. Zhang et al. [24] suggested an architecture vehicular edge computing framework based on cloud computing for offloading. In their work, a Stackelberg game model is used for optimizing resource allocation among vehicle fog/edge computing applications. Cui et al. [11] introduced the concept of fog computing to propose an anonymous authentication scheme for the vehicular network by using a fog server and group administrator. Tang et al. [25] presented the idea of resource pooling into vehicular fog computing (VFC) to jointly save computational applications in a community. Table 1 summarises related works in terms of the year, approach, and disadvantages.

Therefore, this paper introduces the concept of fog computing for 5G-enabled vehicular fog computing by proposing a novel anonymous authentication (ANAA-Fog) scheme to address security and privacy issues.

Table 1. Summarizing Related Work.

Paper Reference	Year	Approach	Disadvantages
[14]	2019	Bilinear Pairing Cryptography	Massive communication costs; requires bilinear pair, requires several scalar multiplication operations and requires several point addition operations
[15]	2019	Bilinear Pairing Cryptography	Massive computation and communication costs; requires map-to-point operations
[17]	2018	Elliptic Curve Cryptography	Several scalar multiplication operations
[18]	2018	Elliptic Curve Cryptography	Several scalar multiplication operations

3. Background

This section describes the design model, security objectives, and mathematical requirements of our work for 5G-enabled vehicular fog computing as follows.

3.1. Design Model

As shown in Figure 1, there are four main entities for our work, namely, one trusted authority (TA), some fog servers, some 5G-base stations (5G-BSs), and many vehicles

equipped with an onboard unit (OBU). The functional work of these entities is explained in the following steps.

- Trusted Authority (TA): The TA is fully trusted in the system and has powerful measurement and sufficient storage. The TA not only works to issue the cryptographic parameters, but also traces the malicious third party when the forged message is reported.
- Fog Server: The fog server is a reliable third party that assists the TA in revealing the signers' identities. Pseudonym IDs for vehicles are generated by mutual authentication via 5G-BS, with the master key preloaded on the fog server by the TA. The public key of the fog server is utilized in our work as the basis for verification.
- A 5G-Base Station (5G-BS): The 5G-BS is a reliable roadside infrastructure. It is a communication medium between entities without data storage or processing capabilities.
- Vehicle: Each vehicle has a wireless device, namely, an onboard unit (OBU), to exchange messages among entities. The OBU supports the 5G standard to save security parameters obtained from the fog server.

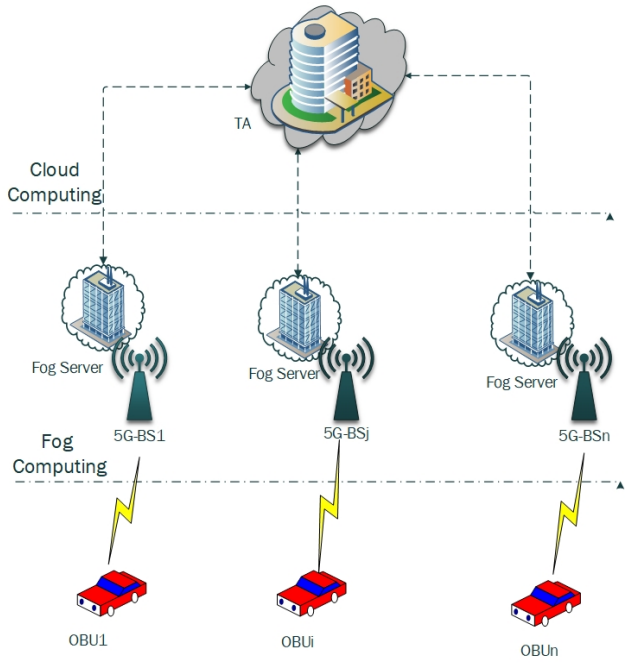

Figure 1. Design model of the proposed ANAA-Fog scheme.

3.2. Security Objectives

The following security objectives need to be met for 5G-enabled vehicular fog computing security.

- Authentication of Signer: To verify that the message is coming from trusted sources;
- Integrity of Message: Aiming to guarantee that the message is delivered unaltered;
- Conditional Privacy-Preserving: To make sure that no third party can reveal the true identity of the vehicle;
- Unlinkability: To ensure no third party can link two or more messages sent from the same signer;
- Traceability: If necessary, the TA can divulge the signer's identity to protect against internal attacks;
- Revocability: The TA can disable the signer's identification and revoke any further use of their signature if necessary.
- Security Attacks Resistance: To ensure that our work is resisting common security attacks, such as modification, forgery, man-in-the-middle, and replay attacks.

3.3. Mathematical Requirements

Presume that the item E/F_p stands for an ECC over a field of prime finite F_p such that p is several large primes. The curve ECC is determined as below.

$$y^2 = x^3 + ax + b \tag{1}$$

where a, b $\in F_p$, and $\delta = 4a^3 + 27b^2 \neq 0$ is the real-valued. The points on E/F_p with an extra point at infinity O form a cyclic additive group of ECC:

$$G = \{(x, y) : x, y \in F_p, E(x, y) = 0\} \cup \{O\} \tag{2}$$

G is the point addition '+' -based group of cyclic additive described as follows: Let $P, Q \in G, l$ be the connected line P and Q (tangent line to E/F_p if $P = Q$), and R be the third intersection point of l with E/F_p. Let l^- be the connected line R and O.

Then, $P + Q$ is the point such that l^- intersects E/F_p at R and O. A form of scalar multiplication based on E/F_p can be measured as follows:

$$tP = P + P + \ldots + P(ttimes) \tag{3}$$

where t $\in F_p$ and P $\in$ G.

- It is difficult to quantify $abP \in G$ when given P, aP, and $bP \in G$, which is the case for any a, b inZ_q^* in the Computational Diffie–Hellman (CDH) Problem.
- Calculating the value $0 \leq l \leq$ q $- 1$ on an elliptic circle complex (ECC) with P and Q of order q on ECC such that $Q = lP$ is known as the "Elliptic Curve Discrete Logarithm" (ECDL) Problem.

4. The Proposed Scheme

The proposed ANAA-Fog scheme consists of four phases: TA initialization, mutual authentication, vehicle signature, and message verification phases, as shown in Figure 2.

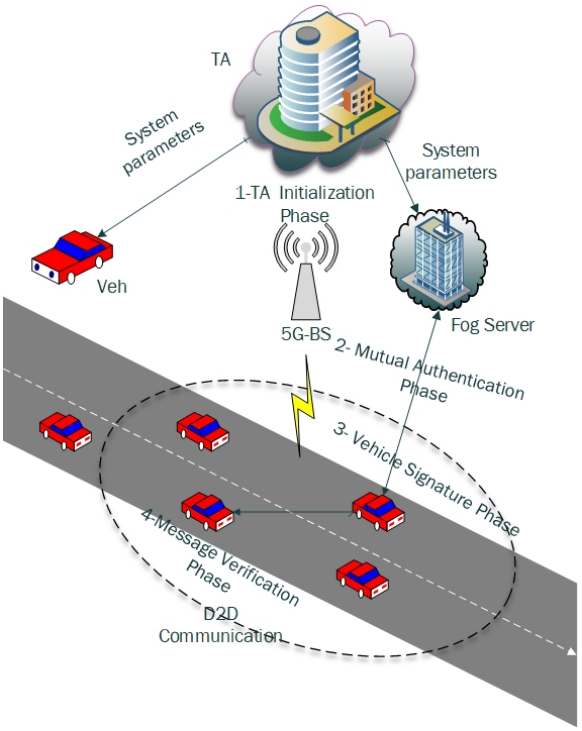

Figure 2. Overall flowchart of the proposed ANAA-Fog scheme.

4.1. TA Initialization Phase

In this phase, the TA executes system initialization as outlined in the following five steps.

- $Step_1$: Let G be an additive group with a generator P and p, q be large prime numbers. Let $E : y^2 = x^3 + ax + b \; mod \; p$ be an elliptic curve, where $a, b \in Z_q^*$.
- $Step_2$: The TA picks a secure message authentication code (MAC) function $MAC(\cdot)$ and three secure hash functions $H_1(\cdot), H_2(\cdot)$, and $H_3(\cdot)$ as $H_1 : G \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \times \{0,1\}^* \times G \rightarrow Z_q^*$, $H_3 : \{0,1\}^* \rightarrow Z_q^*$.
- $Step_3$: The TA picks the randomly chosen number $\xi_{ta} \in Z_q^*$ as the secret key and computes the corresponding public key $Pub_{ta} = \xi_{ta} \cdot P$.
- $Step_4$: The TA publishes and saves the system parameters $\psi = \{G, a, b, P, p, q, H_1, H_2, H_3, Pub_{ta}, MAC(\cdot)\}$ into OBUs and fog servers.
- $Step_5$: Finally, the TA sets the randomly chosen number $\xi_{fog_j} \in Z_q^*$ as the secret key for each fog server Fog_j and then saves both secret keys ξ_{ta} and ξ_{fog_j} on the fog server Fog_j.

4.2. Mutual Authentication Phase

The following nine stages detail how an OBU_i can access the temporary secret key of a fog server Fog_j while joining the 5G-BS coverage area.

- $Step_1$: OBU_i sets the randomly chosen number $\mu \in Z_q^*$ and generates its anonymous-ID (AID_i) as follows.

$$\begin{aligned} AID_i &= \langle AID_i^1, AID_i^2 \rangle \\ AID_i^1 &= \mu \cdot P \\ AID_i^2 &= ID_i \oplus H_1(\mu \cdot Pub_{ta}) \end{aligned} \tag{4}$$

  $Step_2$: Next, OBU_i sends its anonymous-ID (AID_i) to close Fog_j located on the area covered by 5G-BS.
- $Step_3$: While receiving (AID_i) from OBU_i, Fog_j reveals the real identity of OBU_i by using the TA's secret key as follows.

$$ID_i = AID_i^2 \oplus H_1(\xi_{ta} \cdot AID_i^1) \tag{5}$$

- $Step_4$: Next, Fog_j verifies legitimate ID_i by checking whether ID_i exists on the certificate revocation list (CRL). The TA periodically sends CRL to Fog_j to ensure that ID_i is not revoked.
- $Step_5$: Once ID_i is legitimate, Fog_j sets the randomly selected number $a \in Z_q^*$ and calculates $A = a \cdot P$, $R = a \cdot AID_i^1 = a \cdot \mu \cdot P$, $k_{ij} = H_1(R)$ as the same save key among OBU_i and Fog_j, where A helps OBU_i to generate the same save key among OBU_i and Fog_j.
- $Step_6$: Fog_j generates the new temporary secret key as $\xi_{Tkey_j} = H_1(\xi_{fog_j} || ts_{Tkey})$ and computes the corresponding public key of a temporary secret key as $Pub_{fog_j} = \xi_{Tkey_j} \cdot P$, where ts_{Tkey} is the valid timestamp. Note that Fog_j periodically broadcasts its public key (Pub_{fog_j}) with its timestamp ts_{Tkey} on its area covered by 5G-BS.
- $Step_7$: Next, Fog_j encrypts its new temporary secret key ξ_{Tkey_j} as $Enc_{fog_j} = MAC_{K_{ij}}(\xi_{Tkey_j})$ and transmits (A, Enc_{fog_j}) to OBU_i.
- $Step_8$: While receiving (A, Enc_{fog_j}) from Fog_j, OBU_i first calculates the shared secret key k_{ij} as follows.

$$k_{ij} = H_1(\mu \cdot A) = H_1(\mu \cdot a \cdot P) = H_1(a \cdot AID_i^1) \tag{6}$$

- $Step_9$: Next, OBU_i decrypts $Dec_{OBU_i} = MAC_{K_{ij}}(Enc_{fog_j})$ to obtain the temporary secret key ξ_{Tkey_j}. Note that OBU_i saves the temporary secret key ξ_{Tkey_j} into a tamper-proof device (TPD).

4.3. Vehicle Signature Phase

To generate the signature of message Msg_i, this phase executes the vehicle signature, as outlined in the following four steps where ts_i is the current timestamp.

- $Step_1$: OBU_i sets the randomly chosen number $\varrho_i \in Z_q^*$ and generates its public anonymous-ID ($PAID_i$) as follows.

$$\begin{aligned} PAID_i &= \langle PAID_{i,1}, PAID_{i,2} \rangle \\ PAID_{i,1} &= \varrho_i \cdot P \\ PAID_{i,2} &= ID_i \oplus H_1(\varrho_i \cdot Pub_{fog_j}) \end{aligned} \tag{7}$$

- $Step_2$: OBU_i calculates signature key SK_i as follows.

$$SK_i = \xi_{Tkey_j} \cdot H_2(PAID_{i,1}||PAID_{i,2}||ts_i) \tag{8}$$

- $Step_3$: OBU_i generates signature σ_i as follows.

$$\sigma_i = \varrho_i \cdot H_3(Msg_i||PAID_{i,1}||PAID_{i,2}||ts_i) + SK_i \tag{9}$$

- $Step_4$: OBU_i broadcasts $Msg_{OBU_i} = (Msg_i, PAID_{i,1}, PAID_{i,2}, ts_i, \sigma_i)$ to the recipient for 5G-enabled vehicular fog computing.

4.4. Message Verification Phase

While receiving $Msg_{OBU_i} = (Msg_i, PAID_{i,1}, PAID_{i,2}, ts_i, \sigma_i)$, the recipient checks if both Equations (10) and (11) hold and accepts Msg_i if it does.

$$ts_i > ts_r - ts_\bigtriangledown \tag{10}$$

where ts_r is the received time of Msg_{OBU_i}, and $ts_\bigtriangledown$ is the predefined delay time.

$$\begin{aligned} \sigma_i \cdot P &\stackrel{?}{=} \left(\varrho_i \cdot H_3(Msg_i||PAID_{i,1}||PAID_{i,2}||ts_i) + SK_i\right) \cdot P \\ &\stackrel{?}{=} \left(\varrho_i \cdot H_3(Msg_i||PAID_{i,1}||PAID_{i,2}||ts_i) + \xi_{Tkey_j} \cdot H_2(PAID_{i,1}||PAID_{i,2}||ts_i)\right) \cdot P \\ &\stackrel{?}{=} H_3(Msg_i||PAID_{i,1}||PAID_{i,2}||ts_i) \cdot \varrho_i \cdot P + H_2(PAID_{i,1}||PAID_{i,2}||ts_i) \cdot \xi_{Tkey_j} \cdot P \\ &\stackrel{?}{=} H_3(Msg_i||PAID_{i,1}||PAID_{i,2}||ts_i) \cdot PAID_{i,1} + H_2(PAID_{i,1}||PAID_{i,2}||ts_i) \cdot Pub_{fog_j} \end{aligned} \tag{11}$$

In addition, while receiving n of $Msg_{OBU_i} = ({Msg_i}^1, {PAID_{i,1}}^1, {PAID_{i,2}}^1, {ts_i}^1, {\sigma_i}^1), \ldots, ({Msg_i}^n, {PAID_{i,1}}^n, {PAID_{i,2}}^n, {ts_i}^n, {\sigma_i}^n)$ from n $OBUs$, the recipient should check the freshness of n timestamps ${ts_i}^n$ and the validity of n signatures ${\sigma_i}^n$ simultaneously. The recipient uses $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ as the small exponent test technology [26,27] to satisfy non-reputation in the batch signature verification. Thereby, the recipient should check the freshness of n timestamp ${ts_i}^n$ and the validity of n signature ${\sigma_i}^n$ by verifying whether both Equations (10) and (12) hold or not.

$$\begin{aligned}
&\left(\sum_{i=1}^{n}\lambda_i\cdot\sigma_i\right)\cdot P \stackrel{?}{=} \sum_{i=1}^{n}\lambda_i\cdot\left(\varrho_i\cdot H_3(Msg_i||PAID_{i,1}||PAID_{i,2}||ts_i)+SK_i\right)\cdot P\\
&\stackrel{?}{=} \sum_{i=1}^{n}\lambda_i\cdot\left(\varrho_i\cdot H_3(Msg_i||PAID_{i,1}||PAID_{i,2}||ts_i)+\xi_{Tkey_j}\cdot H_2(PAID_{i,1}||PAID_{i,2}||ts_i)\right)\cdot P\\
&\stackrel{?}{=} \sum_{i=1}^{n}\lambda_i\cdot\left(H_3(Msg_i||PAID_{i,1}||PAID_{i,2}||ts_i)\cdot\varrho_i\cdot P+H_2(PAID_{i,1}||PAID_{i,2}||ts_i)\cdot\xi_{Tkey_j}\cdot P\right)\\
&\stackrel{?}{=} \sum_{i=1}^{n}\lambda_i\cdot H_3(Msg_i||PAID_{i,1}||PAID_{i,2}||ts_i)\cdot PAID_{i,1}+\left(\sum_{i=1}^{n}\lambda_i\cdot H_2(PAID_{i,1}||PAID_{i,2}||ts_i)\right)\cdot Pub_{fog_j}
\end{aligned} \tag{12}$$

5. Numerical Example

The signing and verifying processes that make up the entirety of the proposed scheme are all laid out here with specific examples to help clarify each step. Parameters used in the examples along with their corresponding values are listed in Table 2.

Table 2. Parameters and their corresponding example values.

Parameters	Assigned Value
q	627710173538668076383578942320766641608390870039032496127 9
b	24551555460089438177402939151974517847691080581611912380 65
a	−3
P	(60606057595869817452252983063315061066059064341580778811 80, 73105973664259701842662865334749264593111963840112646527)
p	627710173538668076383578942320766641608390870039032496127 9
ID_i	MahmoodArif
ts_i	0:00:59
Msg_i	Accident Zone

5.1. Signing Process

At the signing phase, the following procedures are carried out in order to authenticate VANET messages sent by a vehicle:

- The vehicle selects integer r = 112 and then computes $PAID_{i,1}$= (53726855097945814309235191579839265678416106216898003763 46, 184358346550176987 847666348603008754532800063935891689112 3) $PAID_{i,2}$= 17252a1e7c5d2705773689bd03c4653bab4076c4c605e505a;
- Lastly, the recipient receives the message –signature $(Msg_i, PAID_{i,1}, PAID_{i,2}, ts_i, \sigma_i)$, where ts_i is the date and time of the transmission, for example 2023-04-08 03:00:00 pm.

5.2. Verifying Process

The following procedures are carried out by the vehicle throughout the process of verifying messages:

- The authenticity of the timestamp T_i initial.
- Then, the verifying receiver utilises σ_i of the message–signature tuple $Msg_{OBU_i} = (Msg_i, PAID_{i,1}, PAID_{i,2}, ts_i, \sigma_i)$ to verify safety-related message Msg_i.
- When the conditions below are met, the message is validated. If it does not, the reader will probably ignore this message; $\sigma_i \cdot Pub_{fog_j}$ = (247267479250158315543381241 6893176943027481117926105568348, 20662073387568968298012156318948872628596100 71567012052768) + (69596480264700355969739510381540885514621499602386548851 7, 32643854550959692405542821934424560799560732100000182261 87).
- To ensure the authenticity of a large number of messages in a single batch, the recipient can utilize σ_i of the message–signature tuple $Msg_{OBU_i} = (Msg_i, PAID_{i,1}, PAID_{i,2}, ts_i, \sigma_i)$ to verify safety-related message Msg_i.

- To check many messages about risk concurrently, hone in on the following techniques. $\sum_{i=1}^{n} \lambda_i \cdot \sigma_i Pub_{fog_j} = \sum_{i=1}^{n} (\lambda_i \cdot$ (24726747925015831554338124168931769430274811179261055683 48, 20662073387568968298012156318948872628596100715670120527 68)) + $\sum_{i=1}^{n} (\lambda_i \cdot$ (69596480264700355969739510381540885514621499602386548851 7, 32643854550959692405542821934424560799560732100000182261 87)).

6. Security Analysis

This section analyses our work concerning a ProVerif protocol verifier as well as security requirements.

6.1. ProVerif Protocol Verifier

ProVerif is an automatic cryptographic protocol to evaluate the property of security methods, including anonymous authentication, security gusset attacks resistance, confidentiality, etc., by using correspondence assertions and observational equivalence concepts. In the ProVerif specification language [28], a, b, c, ... and x, y, z, ... denote terms name and variables name, respectively. $Enc(M_1, M_2, \ldots)$ denote the function application to process terms. The major general process is described as follows.

- O: Process with no effect.
- $P|Q$: Methods that run in concurrently.
- $!P$: The ability to repeatedly do something indefinitely.
- New $a : P$: Creation of a random number generator procedure a in P.
- Let $x = MinP$: Process P will continue after the assignment of $x = M$.
- Event (N): The actual happening (N).
- If C, then P, or else Q: Conditionals.
- In (M, x): P: Process P will continue until M has been received on channel M.
- Out (M, N): P: Process P will continue after receiving message N on channel M.

The Dolev–Yao adversary is carried out in the ProVerif tool to analyze the proposed ANAA-Fog scheme. This adversary not has full control power of the environment, but also can delete, modify reads, and inject exchanged information through the communication channel. Nevertheless, the adversary can run primitives only based on primitive definitions. For instance, unless decryption primitives are explicitly described, he/she will be unable to decrypt a message.

The ProVerif tool supports security primitives, such as hash function, digital signatures, and symmetric and asymmetric encryption/decryption. By utilizing terms, variables, and functions, other primitives can be modelled to rewrite equations and rules [29]. Protocols are transformed to horn clauses [30].

In the proposed ANAA-Fog scheme, the OBU_i should receive the temporary secret key ξ_{Tkey_j} from Fog_j. If ok, the Fog_j is validated for OBU_i. To verify the fact that the same ξ_{Tkey_j} that is transmitted by Fog_j, is the one received by OBU_i, this paper uses the following query by ProVerif:

$$Queryinj - event : end(x1, x2) ==> inj - event : begin(x1, x2).$$

In order to determine if $eventend(s, sessionkey)$ from $OBUi$ is the same as $eventbegin(SFogT, x)$, a query is run to see if the arguments (secret key ξ_{Tkey_j} and encryption key = $H_1(aAID_i^1)$) are the same.

In agreement with the formal analysis of the proposed ANAA-Fog scheme, ProVerif verifies the claims of the authentication of Fog to OBU. The output of the ProVerif is as follows.

$$Resultqueryend(x1, x2) ==> begin(x1, x2)istrue.$$

It should be stressed that consensus on the identities of $OBUi$ and Fog_j is not crucial to the success of the proposed ANAA-Fog strategy. Therefore, note that their identities

are not included in the events. The full ProVerif authentication script is referred to in the Appendix A.

Observational Equivalence

The ProVerif tool has the ability to analyze and prove whether intractability, unlinkability, anonymity, etc., hold. These ideas are captured by observational equivalence [31]. Informally, it relates to whether or not two components of the attacker are indistinguishable. The "choice" concept accomplishes this by comparing two arguments and determining whether they are equivalent to the attacker or not. The signing phase of the proposed ANAA-Fog scheme is carried out by using the choice construct as follows.

- $Choice[(Paid, Paid1), (Paid, Paid2)]$: The first tuple $(Paid, Paid1)$ indicates to one sender with public anonymous-IDs $Paid$, $Paid1$ who signs safety messages $m1$, $m2$ with different signature keys, whereas the second tuple $(Paid, Paid2)$ indicates to two different senders with public anonymous-IDs $Paid$, $Paid21$ who sign safety messages $m1$, $m2$. Due to distinct public anonymous-IDs chosen to sign different safety messages, the output should be true. Therefore, the adversary does not have the ability to distinguish between the two tuples.
- $Choice[(\delta(sk1, h(m1)), \delta(sk11, h(m2))), (\delta(sk1, h(m3)), \delta(sk123, h(m4)))]$: The first tuple $\delta(sk1, h(m1))$ of the first argument $(\delta(sk1, h(m1)), \delta(sk11, h(m2)))$ of *choice* construct is the signature of the initial sender with the signature key $sk1$ who signs the message $m1$, whereas the second tuple $\delta(sk1, h(m1))$ of the first argument $(\delta(sk1, h(m1)), \delta(sk11, h(m2)))$ indicates to the same sender with the signature key $sk11$ to sign message $m2$. The second argument $(\delta(sk1, h(m3)), \delta(sk123, h(m4))$ indicates to two signatures for the two senders with signature keys $sk1$ and $sk123$, respectively. For the attacker, the two arguments should be observationally equivalent.
- $Choice[(m1, m2), (m1, m2)]$: Plainly, the attacker cannot distinguish between the two tuples $(m1, m2), (m1, m2)$ due to the random messages. Therefore, the claim of the proposed ANAA-Fog scheme is true by ProVerif as shown in Figure 3.

```
-- Observational equivalence in biprocess 1
Translating the process into Horn clauses...
Termination warning: v ≠ v_1 && attacker2(v_2,v) && attacker2(v_2,v_1) -> bad
Selecting 0
Termination warning: v ≠ v_1 && attacker2(v,v_2) && attacker2(v_1,v_2) -> bad
Selecting 0
Completing...
Termination warning: v ≠ v_1 && attacker2(v_2,v) && attacker2(v_2,v_1) -> bad
Selecting 0
Termination warning: v ≠ v_1 && attacker2(v,v_2) && attacker2(v_1,v_2) -> bad
Selecting 0
RESULT Observational equivalence is true.

------------------------------------------------------------
Verification summary:

Observational equivalence is true.

------------------------------------------------------------
```

Figure 3. The Output of ProVerif on intractability.

6.2. Security Attacks Resistance

- Security Attacks Resistance: The proposed ANAA-Fog scheme resists the common security attacks as follows. Note that Figure 4 elaborates on how your proposed scheme is secure against these active and passive attacks.

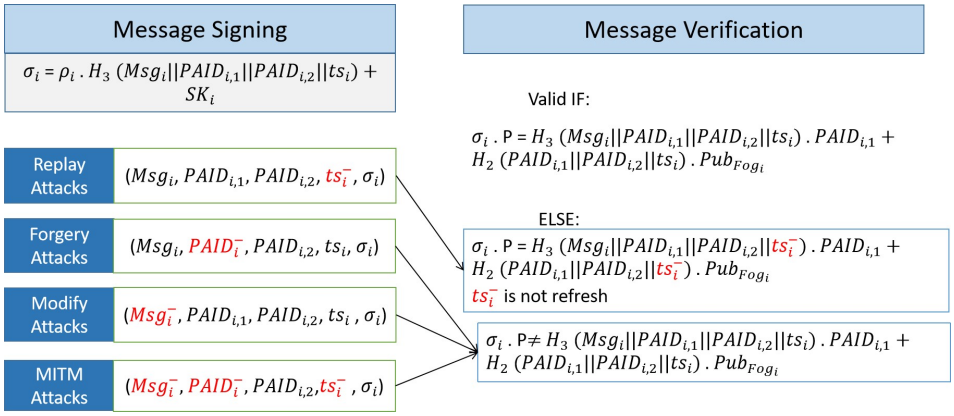

Figure 4. Security attacks resistance.

- Modify Attacks: Since each message's signature includes the master key of the Fog_j and the dynamic random value, the attacker cannot obtain the master key of the Fog_j and the dynamic random value. The attacker cannot modify the message. Otherwise, the receiver's signature authentication is not legal. This means that our work resists modified attacks.
- Forgery Attacks: According to the above proof, no third party can impersonate a valid signature message if he/she does not have the master key of Fog_j. This means that our work resists forgery attacks.
- Replay Attacks: A timestamp ts_i is included in the signature of each message $Msg_{OBU_i} = (Msg_i, PAID_{i,1}, PAID_{i,2}, ts_i, \sigma_i)$, and the signature σ_i cannot be modified. The message receiver can test for replay attacks by checking the signature. This means that our work resists replay attacks.
- Man-In-The-Middle Attacks: According to the above proof, no third party can intercept the communication among nodes (sender and receiver) for 5G-enabled vehicular fog computing. This means that our work resists man-in-the-middle attacks.

6.3. Security Service Comparison

In this subsection, Table 3 shows security service comparison in terms of authentication, the integrity of the message, conditional privacy-preserving, unlikability, traceability, revocability, and low efficiency. These schemes have massive efficiency in terms of computational and communication costs.

Table 3. Comparison of security service.

Security Service	[14]	[15]	[17]	[18]	ANAA-Fog
Authentication	Yes	Yes	Yes	Yes	Yes
Integrity of Message	Yes	Yes	Yes	Yes	Yes
Conditional Privacy-Preserving	Yes	Yes	Yes	Yes	Yes
Unlikability	No	Yes	No	No	Yes
Traceability	No	Yes	Yes	Yes	Yes
Revocability	Yes	No	No	No	Yes
Low Efficiency	No	No	No	No	Yes

Meanwhile, every functionality (security service) explains how exactly our approach will provide for vehicle fog computing enabled by fifth-generation (5G) wireless networks as follows.

- Authentication of Signer and Integrity of Message: Based on the proof analysis in Section 6.1, no third party can forge a valid signature. Thus, the recipient can test the message integrity received from other vehicles by calculating Equations (11) or (12)

for verifying single message or batch messages, respectively. Thus, this work achieves the requirements of authentication of the signer and integrity of the letter.

- Conditional Privacy-Preserving: The proposed ANAA-Fog scheme satisfies the requirement of conditional privacy-preserving in two steps.
  - To prevent an adversary from tracking a OBU_i as it moves between distinct 5G-BSs, each of which has its own unique Fog_j, the $OBUi$ must issue a new public anonymous-ID $(PAID_i)$ by its true identity and the system's public parameters for the period ts_{Tkey}.
  - Once a OBU_i joins the area covered by 5G-BS, it acquires the temporary secret key of Fog_j during period ts_{Tkey}. To protect this key, the $OBUi$ and the Fog_j both use a symmetric secret key, denoted by k_{ij}. Next, it issues a new public anonymous-ID $(PAID_i)$ and its matching signature key as in Equation (8) by the temporary private key of $Fogj$ valid in ts_i, a random value, and its real identity. Since the message is signed with a separate signature key, no third party except *the TA* and $Fogj$ has the capacity to construct a link among the signatures and public anonymous-ID $(PAID_i)$ of OBU_i. When *the TA* and Fog_j know the system's private key, they can construct a link among the signatures and public anonymous-ID $(PAID_i)$ of OBU_i.
- Unlinkability: Each time an OBU_i signs a message, it issues a new public anonymous-ID $(PAID_i)$ to broadcast information. Anonymous-ID is updated regularly. Moreover, dynamic random value is inserted to the signature as Equation (9). Thus, it is so difficult for an attacker to link two messages from the same source.
- Traceability and Revocability: Consider the following scenario to better grasp the need for our work to be traceable and reversible. In the event of an accident, the TA can use Equation (5) to determine the genuine identification of the victim vehicle. After the victim vehicle's genuine identification has been added to the CRL, the TA updates the CRL and sends it to all fog servers. Hence, the impassable vehicle cannot enter the 5G-BS area, where the temporary private key of Fog_j is kept to sign any messages. Therefore, the goals of traceability and revocability are met with this work.

7. Evaluation Metrics

This section analyses the performance of the proposed ANAA-Fog scheme with respect to two evaluation metrics (i.e., computational overhead and communication overhead) for 5G-enabled vehicular fog computing.

7.1. Analysis of Computational Overhead

Concerning the time and energy needed to verify signed messages individually and in bulk, we compare the proposed ANAA-Fog method to some of the most recent alternatives [14,15,17,18]. We build the 80-bit security level for the bilinear pairings-based techniques in [14,15] by using the bilinear pairing e: $G_1 * G_1 \rightarrow G_2$, where G_1 is an additive group with a huge prime $q-$ generated by a point $p-$ on super-singular ECC. The duration of individual cryptographic procedures is listed in Table 4.

Table 4. The time required for various cryptographic operations

Abbr.	Execution Time (ms)	Definition
P_{bp}	5.811	How long a G_1 bilinear pairing takes in time.
M_{bp}	1.5654	The amount of time needed to do a scalar multiplication in the G_1
A_{bp}	0.0106	Time taken to perform a point-sum calculation in G_1
H_{mtp}	4.1724	The amount of time needed by a map-to-point hash function in G_1
M_{ecc}	0.6718	The amount of time needed to do a scalar multiplication in G
A_{ecc}	0.0031	Time taken to perform a point-sum calculation in G

The overhead of computation for the schemes of Zhong et al. [14] and Bayat et al. [15] are based on a bilinear pair as follows. A vehicle Veh_i in the scheme of Zhong et al. [14] signs a message Msg_i with 4 scalar multiplication ($4M_{bp}$) operations and 2 A point addition ($2A_{bp}$) operations. Consequently, the vehicle Veh_i in the scheme of Zhong et al. [14] needs a cost of $4M_{bp} + 2A_{bp} \approx 6.2828$ ms in the vehicle signature process. While a recipient Veh_j requires 2 bilinear pair ($2P_{bp}$), 5 scalar multiplication ($5M_{bp}$) operations, and 2 A point addition ($2A_{ecc}$) operations to verify the concerned signature σ_i. Consequently, the vehicle Veh_j in the scheme of Zhong et al. [14] needs a cost of $2P_{bp} + 5M_{bp} + 2A_{bp} \approx 19.4702$ ms in a single message verification process. To verify the concerned batch signatures ${\sigma_i}^n$ from batch messages sent, a recipient Veh_j needs (n + 1) bilinear pair ($(n+1)P_{bp}$), (5n) scalar multiplication ($(5n)M_{bp}$) operations, and (2n) A point addition ($(2n)A_{bp}$) operations. Consequently, the vehicle Veh_j in the scheme of Zhong et al. [14] needs a cost of $(n+1)P_{bp} + (5n)M_{bp} + (2n)A_{bp} \approx 5.811 + 13.6592n$ ms in a batch message verification process.

A vehicle Veh_i in the scheme of Bayat et al. [15] signs a message Msg_i with 1 scalar multiplication ($1M_{bp}$) operation and 1 point addition ($1A_{bp}$) operation. Consequently, the vehicle Veh_i in the scheme of Bayat et al. [15] needs a cost of $1M_{bp} + 1A_{bp} \approx 1.576$ ms in the vehicle signature process. While a recipient Veh_j requires 3 bilinear pair ($3P_{bp}$), 1 scalar multiplication ($1M_{bp}$) operation, 1 point addition ($1A_{ecc}$) operation, and 1 map-to-point function ($1H_{mtp}$) to verify the concerned signature σ_i. Consequently, the vehicle Veh_j in the scheme of Bayat et al. [15] needs a cost of $3P_{bp} + 1M_{bp} + 1A_{bp} + 1H_{mtp} \approx 23.1814$ ms in a single message verification process. To verify the concerned batch signatures ${\sigma_i}^n$ from batch messages sent, a recipient Veh_j needs (3) bilinear pair ($(3)P_{bp}$), (n) scalar multiplication ($(n)M_{bp}$) operations, (n) A point addition ($(n)A_{bp}$) operations and n map-to-point function ($(n)H_{mtp}$). Consequently, the vehicle Veh_j in the scheme of Bayat et al. [15] needs a cost of $(3)P_{bp} + (n)M_{bp} + (n)A_{bp} + (n)H_{mtp} \approx 17.433 + 5.7484n$ ms in a batch message verification process.

The overhead of computation for the schemes of Asaar et al. [17], Li et al. [18], and our work are based on elliptic curve cryptography as follows. A vehicle Veh_i in the scheme of Asaar et al. [17] signs a message Msg_i with 7 scalar multiplication ($7M_{ecc}$) operations. Consequently, the vehicle Veh_i in the scheme of Asaar et al. [17] needs a cost of $7M_{ecc} \approx 4.7026$ ms in a vehicle signature process. While a recipient Veh_j requires 12 scalar multiplication ($12M_{ecc}$) operations and 8 A point addition ($8A_{ecc}$) operations to verify the concerned signature σ_i, the vehicle Veh_j in the scheme of Asaar et al. [17] needs a cost of $12M_{ecc} + 8A_{ecc} \approx 8.0864$ ms in a single message verification process. To verify the concerned batch signatures ${\sigma_i}^n$ from batch messages sent, a recipient Veh_j needs (4n + 10) scalar multiplication ($(4n+10)M_{ecc}$) operations and (6n + 2) A point addition ($(6n+2)A_{ecc}$) operations. Consequently, the vehicle Veh_j in the scheme of Asaar et al. [17] needs a cost of $(4n+10)M_{ecc} + (6n+2)A_{ecc} \approx 6.7242 + 2.6934n$ ms in a batch message verification process.

A vehicle Veh_i in the scheme of Li et al. [18] signs a message Msg_i with 1 scalar multiplication ($1M_{ecc}$) operations. Consequently, the vehicle Veh_i in the scheme of Li et al. [18] needs a cost of $1M_{ecc} \approx 0.6718$ ms in the vehicle signature process. While a recipient Veh_j requires 4 scalar multiplication ($4M_{ecc}$) operations and 1 A point addition ($1A_{ecc}$) operation to verify the concerned signature σ_i, the vehicle Veh_j in the scheme of Li et al. [18] needs a cost of $4M_{ecc} + 1A_{ecc} \approx 2.6903$ ms in a single message verification process. To verify the concerned batch signatures ${\sigma_i}^n$ from batch messages sent, a recipient Veh_j needs (2n + 2) scalar multiplication ($(2n+2)M_{ecc}$) operations and (n) A point addition ($(n)A_{ecc}$) operation. Consequently, the vehicle Veh_j in the scheme of Li et al. [18] needs a cost of $(2n+2)M_{ecc} + (n)A_{ecc} \approx 1.3436 + 1.3467n$ ms in a batch message verification process.

A vehicle Veh_i in the proposed ANAA-Fog scheme signs a message Msg_i with 2 scalar multiplication ($2M_{ecc}$) operations and 1 point addition ($1A_{ecc}$) operation. Consequently, the vehicle Veh_i in the proposed ANAA-Fog scheme needs a cost of $2M_{ecc} + 1A_{ecc} \approx 1.3467$ ms in the vehicle signature process. While a recipient Veh_j requires 3 scalar multiplication

$(3M_{ecc})$ operations and 1 A point addition $(1A_{ecc})$ operation to verify the concerned signature σ_i, the vehicle Veh_j in the proposed ANAA-Fog scheme needs a cost of $3M_{ecc} + 1A_{ecc} \approx 2.0185$ ms in a single message verification process. To verify the concerned batch signatures σ_i^n from batch messages sent, a recipient Veh_j needs (n + 2) scalar multiplication $((n+2)M_{ecc})$ operations and (n − 1) A point addition $((n-1)A_{ecc})$ operation. Consequently, the vehicle Veh_j in the proposed ANAA-Fog scheme needs a cost of $(n+2)M_{ecc} + (n-1)A_{ecc} \approx 1.3405 + 0.6749n$ ms in a batch message verification process.

The overhead of computational of the proposed ANAA-Fog scheme and the most recent works in [14,15,17,18] with respect to a message signing, single verification, and batch verification are compared graphically in Figures 5–7.

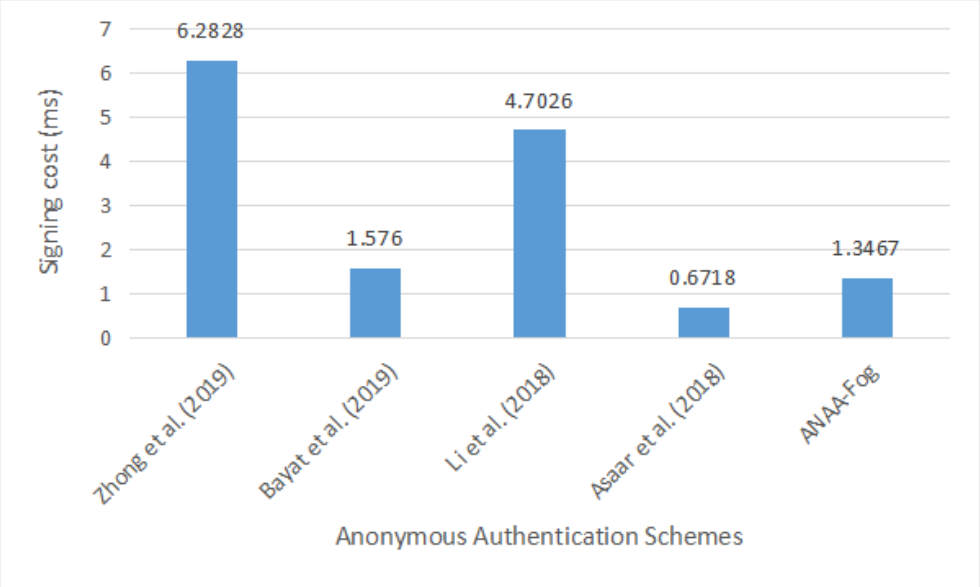

Figure 5. Single-message signing's computational burden.

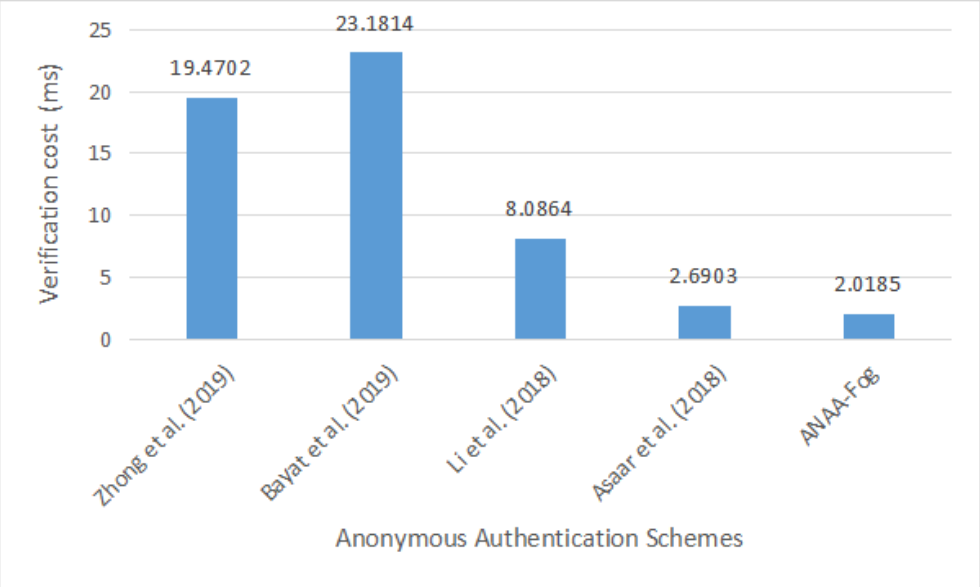

Figure 6. Verifying a single signature involves a large amount of computation.

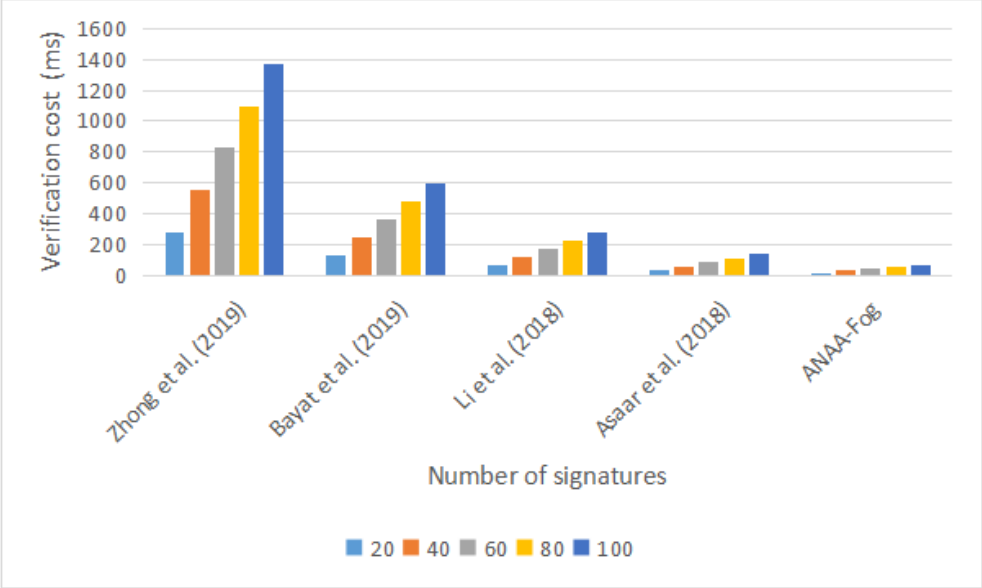

Figure 7. The time cost of verifying a large number of signatures in a batch.

7.2. Analysis of Communication Overhead

For the bilinear pairings-based schemes in [14,15], the sizes of prime numbers p^-, q^- are 128 bytes, and 64 bytes, respectively, since it runs an equation $y^2 = (x^3 + x)$ with embedding degree 2. For the proposed ANAA-Fog scheme and the scheme in [17,18], the size of prime numbers p, q is 64 bytes since it runs an equation $y^2 = x^3 + xmodp$. Moreover, the size of the timestamp's output is 4 bytes, and the general hash function is 20 bytes. Figure 8 shows the communication overhead of authentication schemes.

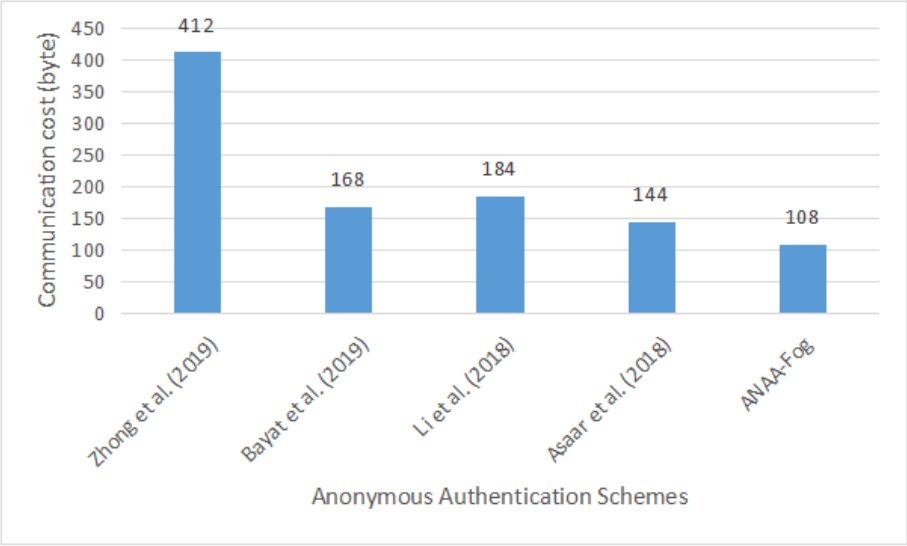

Figure 8. Verification communication overhead for authentication methods.

In Zhong et al. [14], the tuple of a message shared by vehicles is $Msg_{OBU_i} = (Msg_i, PID_i, vpk_i, t_i, \sigma_i)$, where $\sigma_i = R_i, T_i$, $(PID_i = PID_{i,1}, PID_{i,2})$, $(PID_{i,1}, R_i, vpk_i \in G_1)$, $(T_i \in Z_q^*)$, and two timestamps (t_i, VP_i); therefore, the total overhead of communication is $3 \cdot 128 + 20 + 8 = 412$ bytes.

In Bayat et al. [15], the tuple of a message shared by vehicles is $Msg_{OBU_i} = (Msg_i, pid_i, \sigma_i)$, where $(pid_i = PID_{i,l}^1, PID_{i,l}^2)$, $(PID_{i,l}^1 \in G_1)$, and $(PID_{i,l}^2, \sigma_i \in Z_q^*)$; therefore, the total overhead of communication is $128 + 2 \cdot 20 = 168$ bytes.

In Asaar et al. [17], the tuple of a message shared by vehicles is $Msg_{OBU_i} = (Cert_k, sig, Y_k)$; therefore, the total overhead of communication is $3 \cdot 40 + 3 \cdot 20 + 4 = 184$ bytes.

In Li et al. [18], the tuple of a message shared by vehicles is $Msg_{OBU_i} = (Msg_i, PID_{i,l}, PK_{i,l}, R_i, T_i, sig_i)$, where $(PK_{i,l}, R_i, sig_i \in G_1)$, $(PID^1_{i,l}\sigma_i \in Z^*_q)$, and T_i is a timestamp; therefore, the total overhead of communication is $3 \cdot 40 + 20 + 4 = 144$ bytes.

In our work, the tuple of the message shared by vehicles is $Msg_{OBU_i} = (Msg_i, PAID_{i,1}, PAID_{i,2}, ts_i, \sigma_i)$, where $(PAID_{i,1} \in G)$, $(PAID_{i,2}, \sigma_i \in Z^*_q)$, and one timestamp (ts_i); therefore, the total overhead of communication is $64 + 2 \cdot 20 + 4 = 108$ bytes.

In summary, our work needs smaller communication overheads than other schemes when the message is shared by vehicle broadcasts to others in 5G-enabled vehicular fog computing.

8. Conclusions

In this research, we suggested a new anonymous authentication strategy for 5G-enabled vehicle fog computing: the ANAA-Fog technique. This scheme is based on a fog server to generate the temporary secret key to each participating vehicle for a signature verification process. The security analysis section shows that the signing phase of the proposed ANAA-Fog scheme is carried out by using the ProVerif simulator to choose the message construct. Additionally, this work satisfies authentication of the signer, the integrity of the message, conditional privacy-preserving, unlinkability, traceability, revocability, and security attacks resistance in terms of modification, forgery, replay, and man-in-the-middle attacks. The evaluation metrics section shows that our work has low performance overhead compared to related works.

In future work, we will investigate the related results to use a lightweight algorithm instead of ECC for 5G-enabled vehicular fog computing. Meanwhile, we extend this work by adding a complete numerical example with a handshake model explanation and using a network simulator (OMNeT++) and road traffic (SUMO) for the experiment environment.

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

Funding: This research has been funded by Deputy for Research & Innovation, Ministry of Education through Initiative of Institutional Funding at University of Ha'il, Saudi Arabia, through project number IFP-22 169.

Data Availability Statement: Not Applicable.

Acknowledgments: We would like to acknowledge the Deputy for Research & Innovation, Ministry of Education through Initiative of Institutional Funding at University of Ha'il, Saudi Arabia, for funding this research.

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

Abbreviations

The following abbreviations are used in this manuscript:

ANAA-Fog	A Novel Anonymous Authentication Scheme for 5G-enabled Vehicular Fog Computing
TA	Trusted Authority
CDH	Computational Diffie–Hellman
ECDL	Elliptic Curve Discrete Logarithm
G	An additive group
p, q	Large prime numbers of generator P
$MAC(\cdot)$	Message authentication code (MAC) function
$H_i(\cdot)$	Three secure hash functions (i = 1, 2, 3)
ξ_{ta}	Secret key of TA system
Pub_{ta}	Public key of TA system
ξ_{fog_j}	The secret key for each fog server Fog_j
μ	Randomly chosen number
(AID_i)	Anonymous-ID
k_{ij}	Shared secret key between OBU_i and Fog_j
ξ_{Tkey_j}	Temporary secret key into TPD
Msg_i	Signature of message
$\|\|$	Operations of Concatenation
$\oplus$	Operation of X-OR
T_i	Current Timestamp

Appendix A. ProVerif Authentication Script

```
free c,P: channel.
type host.
query inj-event(end(x1,x2)) ==> inj-event(begin(x1,x2)).
fun sign(bitstring, sskey): bitstring.
reduc checksign(sign(x,y), pk(y))=x.
fun enc(bitstring, key): bitstring.
reduc dec(enc(x,y),y)=x.
reduc Xor1(Xor(x,y),x)=y.
reduc Xor2(Xor(x,y),y)=x.
reduc multinv(mult(x,mult(y,z)),z)=mult(x,y).
not sTA.
let OBU =
new r: nonce;
let PAID1= mult (r, P) in
let PAID2= Xor(RID, h(mult(r, PubTA))) in
out(c,(PAID1,PAID2));
in(c,PFog);
in(c,(Rx,encrypted));
let a = multinv(Rx, r) in
let sessionkey = h2(a) in
let (s, sig) = dec(encrypted,sessionkey) in
if s = checksign(sig, pkTA) then
event end(s, sessionkey).
let Fog =
in (c, sig);
new a: nonce;
in(c,(paid1,paid2));
let R=mult(a, paid1) in
let SFogt = h(SFog) in
event begin(SFogt, x);
let PFog = mult(h(SFog),p) in
out(c, PFog);
if SFog= checksign(sig, pkTA) then
out(c,(R,enc((SFogt,sig),x))).
```

```
let TA=
new SFog: nonce;
let sig= sign (SFog, sTA) in
out(c, sig);
out(FogTA,SFog).
process
new sTA: bitstring;
let pkTA= pk(sTA) in
out(c, pkTA);
((!OBU)|(!Fog)|(!TA))

ProfVerif output:
Result query end(x1, x2) ==> begin(x1, x2) is true.
```

References

1. Li, C.; Zhang, X.; Wang, H.; Li, D. An enhanced secure identity-based certificateless public key authentication scheme for vehicular sensor networks. *Sensors* **2018**, *18*, 194. [CrossRef]
2. Al-Shareeda, M.A.; Anbar, M.; Hasbullah, I.H.; Manickam, S. Survey of authentication and privacy schemes in vehicular ad hoc networks. *IEEE Sens. J.* **2020**, *21*, 2422–2433. [CrossRef]
3. Zhou, X.; Luo, M.; Vijayakumar, P.; Peng, C.; He, D. Efficient certificateless conditional privacy-preserving authentication for vanets. *IEEE Trans. Veh. Technol.* **2022**, *71*, 7863–7875. [CrossRef]
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. Provably secure with efficient data sharing scheme for fifth-generation (5G)-enabled vehicular networks without road-side unit (RSU). *Sustainability* **2022**, *14*, 9961. [CrossRef]
5. Yang, Y.; He, D.; Wang, H.; Zhou, L. An efficient blockchain-based batch verification scheme for vehicular ad hoc networks. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3857. [CrossRef]
6. 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]
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-cppa: Chaotic map-based conditional privacy-preserving authentication scheme in 5G-enabled vehicular networks. *Sensors* **2022**, *22*, 5026. [CrossRef] [PubMed]
8. Li, Q.; He, D.; Yang, Z.; Xie, Q.; Choo, K.K.R. Lattice-Based Conditional Privacy-Preserving Authentication Protocol for the Vehicular Ad Hoc Network. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4336–4347. [CrossRef]
9. Hou, P.S.; Fadzil, L.M.; Manickam, S.; Al-Shareeda, M.A. Vector Autoregression Model-Based Forecasting of Reference Evapotranspiration in Malaysia. *Sustainability* **2023**, *15*, 3675. [CrossRef]
10. Zhang, J.; Cui, J.; Zhong, H.; Bolodurina, I.; Liu, L. Intelligent drone-assisted anonymous authentication and key agreement for 5G/B5G vehicular ad-hoc networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 2982–2994. [CrossRef]
11. Cui, J.; Wang, Y.; Zhang, J.; Xu, Y.; Zhong, H. Full Session Key Agreement Scheme Based on Chaotic Map in Vehicular Ad hoc Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8914–8924. [CrossRef]
12. Mohammed, B.A.; Al-Shareeda, M.A.; Manickam, S.; Al-Mekhlafi, Z.G.; Alreshidi, A.; Alazmi, M.; Alshudukhi, J.S.; Alsaffar, M. FC-PA: Fog Computing-based Pseudonym Authentication Scheme in 5G-enabled Vehicular Networks. *IEEE Access* **2023**, *11*, 18571–18581. [CrossRef]
13. Al-Mekhlafi, Z.G.; Al-Shareeda, M.A.; Manickam, S.; Mohammed, B.A.; Alreshidi, A.; Alazmi, M.; Alshudukhi, J.S.; Alsaffar, M.; Alsewari, A. Chebyshev Polynomial-Based Fog Computing Scheme Supporting Pseudonym Revocation for 5G-Enabled Vehicular Networks. *Electronics* **2023**, *12*, 872. [CrossRef]
14. Zhong, H.; Han, S.; Cui, J.; Zhang, J.; Xu, Y. Privacy-preserving authentication scheme with full aggregation in VANET. *Inf. Sci.* **2019**, *476*, 211–221. [CrossRef]
15. Bayat, M.; Pournaghi, M.; Rahimi, M.; Barmshoory, M. NERA: A New and Efficient RSU based Authentication Scheme for VANETs. *Wirel. Netw.* **2019**, *26*, 3083–3098. [CrossRef]
16. Liu, Y.; Wang, L.; Chen, H.H. Message authentication using proxy vehicles in vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **2014**, *64*, 3697–3710. [CrossRef]
17. Asaar, M.R.; Salmasizadeh, M.; Susilo, W.; Majidi, A. A secure and efficient authentication technique for vehicular ad-hoc networks. *IEEE Trans. Veh. Technol.* **2018**, *67*, 5409–5423. [CrossRef]
18. Li, J.; Choo, K.K.R.; Zhang, W.; Kumari, S.; Rodrigues, J.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]

19. Zhang, J.; Fang, H.; Zhong, H.; Cui, J.; He, D. Blockchain-Assisted Privacy-Preserving Traffic Route Management Scheme for Fog-Based Vehicular Ad-Hoc Networks. *IEEE Trans. Netw. Serv. Manag.* **2023**. [CrossRef]
20. Cui, J.; Yu, J.; Zhong, H.; Wei, L.; Liu, L. Chaotic Map-Based Authentication Scheme Using Physical Unclonable Function for Internet of Autonomous Vehicle. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 3167–3181. [CrossRef]
21. Chen, B.; Xiang, T.; Li, X.; Zhang, M.; He, D. Efficient Attribute-Based Signature With Collusion Resistance for Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2023**. [CrossRef]
22. Xiao, Y.; Zhu, C. Vehicular fog computing: Vision and challenges. In Proceedings of the 2017 IEEE 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, Big Island, HI, USA, 13–17 March 2017; pp. 6–9.
23. Miao, D.; Liu, L.; Xu, R.; Panneerselvam, J.; Wu, Y.; Xu, W. An efficient indexing model for the fog layer of industrial internet of things. *IEEE Trans. Ind. Inf.* **2018**, *14*, 4487–4496. [CrossRef]
24. Zhang, K.; Mao, Y.; Leng, S.; Maharjan, S.; Zhang, Y. Optimal delay constrained offloading for vehicular edge computing networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.
25. Tang, C.; Xia, S.; Li, Q.; Chen, W.; Fang, W. Resource pooling in vehicular fog computing. *J. Cloud Comput.* **2021**, *10*, 1–14. [CrossRef]
26. Horng, S.J.; Tzeng, S.F.; Pan, Y.; Fan, P.; Wang, X.; Li, T.; Khan, M.K. b-SPECS+: Batch verification for secure pseudonymous authentication in VANET. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1860–1875. [CrossRef]
27. Jianhong, Z.; Min, X.; Liying, L. On the security of a secure batch verification with group testing for VANET. *Int. J. Netw. Secur.* **2014**, *16*, 351–358.
28. Abadi, M.; Fournet, C. Mobile values, new names, and secure communication. *ACM Sigplan Not.* **2001**, *36*, 104–115. [CrossRef]
29. Blanchet, B.; Chaudhuri, A. Automated formal analysis of a protocol for secure file sharing on untrusted storage. In Proceedings of the 2008 IEEE Symposium on Security and Privacy (sp 2008), Washington, DC, USA, 2008; pp. 417–431.
30. Küsters, R.; Truderung, T. Using ProVerif to analyze protocols with Diffie-Hellman exponentiation. In Proceedings of the 2009 22nd IEEE Computer Security Foundations Symposium, Port Jefferson, NY, USA, 8–10 July 2009; pp. 157–171.
31. Cheval, V.; Blanchet, B. Proving more observational equivalences with ProVerif. In Proceedings of the International Conference on Principles of Security and Trust, Prague, Czech Republic, 6–11 April 2013; pp. 226–246.