


Article

An Anonymous Certificateless Signcryption Scheme for Secure and Efficient Deployment of Internet of Vehicles

Insaf Ullah ¹, Muhammad Asghar Khan ¹ , Mohammed H. Alsharif ^{2,*}  and Rosdiadee Nordin ³ 

¹ Hamdard Institute of Engineering & Technology, Islamabad 44000, Pakistan; insafktk@gmail.com (I.U.); khayyam2302@gmail.com (M.A.K.)

² Department of Electrical Engineering, College of Electronics and Information Engineering, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Korea

³ Department of Electrical, Electronic & Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia; adee@ukm.edu.my

* Correspondence: malsharif@sejong.ac.kr

Abstract: Internet of Vehicles (IoV) is a specialized breed of Vehicular Ad-hoc Networks (VANETs) in which each entity of the system can be connected to the internet. In the provision of potentially vital services, IoV transmits a large amount of confidential data through networks, posing various security and privacy concerns. Moreover, the possibility of cyber-attacks is comparatively higher when data transmission takes place more frequently through various nodes of IoV systems. It is a serious concern for vehicle users, which can sometimes lead to life-threatening situations. The primary security issue in the provision of secure communication services for vehicles is to ensure the credibility of the transmitted message on an open wireless channel. Then, receiver anonymity is another important issue, i.e., only the sender knows the identities of the receivers. To guarantee these security requirements, in this research work, we propose an anonymous certificateless signcryption scheme for IoV on the basis of the Hyperelliptic Curve (HEC). The proposed scheme guarantees formal security analysis under the Random Oracle Model (ROM) for confidentiality, unforgeability, and receiver anonymity. The findings show that the proposed scheme promises better security and reduces the costs of computation and communication.

Keywords: Internet of Things; Internet of Vehicles; security; hyperelliptic curve; random oracle model



Citation: Ullah, I.; Khan, M.A.; Alsharif, M.H.; Nordin, R. An Anonymous Certificateless Signcryption Scheme for Secure and Efficient Deployment of Internet of Vehicles. *Sustainability* **2021**, *13*, 10891. <https://doi.org/10.3390/su131910891>

Academic Editor:
Manuel Fernandez-Veiga

Received: 8 August 2021
Accepted: 28 September 2021
Published: 30 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the past few decades, academic and industry researchers have worked harder to push the technology of Mobile Ad hoc Networks (MANETs) to a new extreme [1]. Mobile devices may now establish a network with flying, self-organizing, and dynamic connections to one another without the need for any fixed communication infrastructure [2]. MANETs evolved over time too, and one of the most advanced forms, Vehicular Ad hoc Networks (VANETs), was introduced, in which peer vehicles exchange information [3]. Vehicles, in collaboration with transportation infrastructure, engage in vehicle-to-everything (V2X) communication, which includes vehicle-to-vehicle (V2V), vehicle-to-sensor (V2S), vehicle-to-pedestrian (V2P), and vehicle-to-infrastructure (V2I) interactions [4]. The Internet of Vehicles (IoV), a hybrid of VANETs and the Internet of Things, sometimes known as IoT on wheels, relies heavily on V2X communication. In both non-safety and safety-critical automotive applications, IoV is truly a breakthrough technology.

IoV is a dynamic mobile connectivity infrastructure that provides a low-cost networking solution for connecting vehicles to the public network in order to improve transportation system safety and performance [5]. Therefore, a strong, but versatile, communication, networking, and computing technology foundation is needed for this complex ecosystem. Fifth-generation (5G) technology would be a safer option in such a setting to offer ultra-low latency, ultra-fast reliability, high data rate, and everywhere access. Because

5G networks enable vehicles to accommodate different types of IoV message deliveries to support intelligent transportation systems, where all vehicles and infrastructure systems are interconnected. A 5G-enabled IoV system is very important for the automotive industry because of its infrastructure and large capacity to support communication services. As a result, connected vehicles represent the next IoV frontier, while ongoing 5G innovation is necessary to enable high-reliability and low-latency radio access for essential communications, even in high-density IoV systems [6].

Since IoV has widespread interconnected networks with numerous users, there is an increased risk of security and privacy concerns [7]. The possibility of cyber-attacks is comparatively higher when data transmission takes place more frequently through various nodes of IoV systems. For instance, if a self-governing vehicle needs to execute a certain task, it gets, among other things, the simple safety warning containing appropriate task-specific information such as time, speed, and destination, etc. However, the environments of IoV could be dangerous in the absence of security protections. It gives an enormous opportunity to malicious attackers to modify, intercept, delete, or even insert false information during the on-going transmission.

Most of the existing cybersecurity mechanisms deal with critical system components and provide a solution to the well-known security threats. Some of the well-known common security and privacy issues across the IoV environment include tracking vehicle locations, hardware tampering, unauthorized data access, message modification, and fabrication [8]. The intruders can even introduce an ambiguity across the network and steal the confidential data with the inevitable loss of data integrity and privacy features. Once the identity of the user is compromised, it will put his or her property and safety at risk, and a malicious attacker, such as a stalker, could use the targeted identity to track down a specific driver and/or initiate a malicious attack. Thus, advanced security measures for IoV systems have become the most essential requirement [9].

The primary security concerns in providing secure connectivity in an IoV network is to ensure the authenticity of the transmitted messages on an open wireless channel. Then, receiver anonymity is another important issue, i.e., only the sender knows the identities of the receivers. Providentially, such obstruction can be fulfilled by utilizing a compound scheme, named anonymous signcryption [10]. The scheme is anonymous and can avoid malicious user attack while performing encryption and authentication in one go. To avoid the key escrow problem in the proposed compound scheme, a certificateless cryptosystem is usually preferred [11].

The Key Generation Center (KGC) has no previous knowledge of the secret value of the participant in a certificateless cryptosystem; the key escrow dilemma can, therefore, be avoided. Rivest-Shamir-Adleman (RSA), bilinear pairing, and elliptic curve cryptography, which are typically based on computationally challenging problems, are typically used to achieve security and efficiency in the security scheme [12]. For example, the RSA cryptography uses a large factorization of having key-size stretches as much as 1024-bits. Bilinear pairing is weaker compared to RSA, due to immense pairing and map-to-point function computation. Similarly, the elliptic curve was implemented to resolve the inconsistencies associated with RSA and bilinear pairing, which is a modern cryptography technique. The elliptic curve cryptography is used to provide the security and efficiency with a key-size up to 160 bits. Nevertheless, to provide the same level of security as elliptic curve, an advanced version, called hyperelliptic curve (HEC), was introduced [13]. The HEC uses 80-bit key size, and, at the same time, promises the security features characteristic of elliptic curve, bilinear pairing, and RSA. Therefore, the hyper elliptic curve is alleged as a much better choice for IoV. In short, to adapt an anonymous certificateless signcryption scheme in the IoV environment, the proposed scheme must satisfy the following attributes:

- Confidentiality, unforgeability, and anonymity.
- Immune to key escrow problem.
- Secure in open wireless channels.
- Efficient in terms of computational and communication costs.

- Provably secure using ROM model.

1.1. Motivation and Contributions

This paper is motivated from the aforementioned discussion and solves the problem, to ensure the credibility of the transmitted message and receiver anonymity, by proposing a new scheme, which is certificateless and based on the concept of a hyper-elliptic curve. The main contributions of the undertaken research work are distinguished by the following outstanding attributes:

- An efficient and secure scheme, namely an anonymous certificateless signcryption scheme, has been proposed for an IoV environment.
- The proposed scheme avoids the key escrow problem by employing the certificateless cryptography mechanism.
- Moreover, the proposed scheme makes use of hyperelliptic curve cryptography for encryption and signature verification.
- The proposed scheme guarantees confidentiality, unforgeability, and receiver anonymity on open wireless links under the Random Oracle Model (ROM) analysis.
- Finally, it is revealed that the proposed scheme is superior, particularly in terms of computational and communication costs, while doing a comparative study with relevant state-of-art schemes.

1.2. Organization of the Paper

The article is structured as follows. Related work is discussed in Section 2. Preliminaries are explained in Section 3. System models are given in Section 4. The proposed scheme can be seen in Section 5. Formal security analysis, using ROM, is carried out in Section 6. Section 7 presents performance comparison with existing schemes. Finally, Section 8 contains the concluding thoughts.

2. Related Work

In recent years, IoV has been recognized in a range of applications, ranging from smart transportation to health care and itinerary planning. Vehicles aggregate mission-critical data from the deployed area within IoV systems and disseminate it using their OBUs with other vehicles, RSUs, and cloud servers. IoV data can be analyzed locally or on a cloud server, and actions are taken according to the type of request.

Since IoV has widespread interconnected networks with numerous users, it is obvious that there is an increased risk of security and privacy measures. The use of encryption and digital signature cryptographic tools will overcome these concerns. In addition, if both the tools, i.e., encryption and digital signature, are needed at the same time, an amalgamated form, called signcryption, is used. To remove the key escrow problem associated with signcryption, a certificateless approach is generally taken into account [14].

In 2008, certificateless signcryption (CLSC) scheme was first introduced by Barbosa and Farshim [15]. One year later, in 2009, Xie et al. [16] proposed a certificateless signcryption scheme based on the standard model. Liu et al. [17], in 2010, presented a standard model-based certificateless signcryption. These schemes, on the other hand, have a high computational cost, take a long time to execute, and are not very secure.

In the same year, in 2010, Selvi et al. [18], overcome the security weaknesses of Xie et al. [16] and Liu et al. [17] signcryption schemes.

In 2014, Shi et al. [19] suggested an improvement in the CLSC scheme in terms of security under random oracle model and without bilinear pairings. In 2016, Abdul Wahid and Masahiro Mamb [20] suggested a certificateless signature scheme based on the elliptic curve theorem. In JavaScript, the implementation of the proposed scheme was accomplished. The authors believed that, after cost analysis, their scheme was better than the relevant existing schemes. A standard model-based certificateless signcryption scheme was proposed by Caixue et al. [21] and Parvin Rastegari and Mehdi Berenjkoub [22]. Their analysis shows

that, compared to all random oracle model-based certificateless sign encryption systems, the presented schemes were much more reliable and efficient.

In 2017, certificateless signcryption schemes, without bilinear pairing, were proposed [23,24]. The security specifications of the schemes presented in [23,24] were shown to be secured via the ROM. Later, in 2018, Zhou [25] suggested a new bilinear pairing certificateless signcryption approach, and security verification on the standard model was carried out. In the same year, pairing-free certificateless signcryption based on elliptic curve cryptography was proposed by Cao and Ge [26]. One year later, in 2019, Luo and Ma [27] proposed an efficient and secure certificateless hybrid signcryption for cloud storage.

In order to overcome the significant error in the construction of Luo and Ma [27] schemes, Rastegari et al. [28] revisited the proposed scheme in 2019. However, the schemes presented in [26–28] are based on the concept of elliptic curve cryptography, which incur high computational costs. Additionally, the schemes do not meet security requirements such as anonymity. Finally, Karati et al. [29] implemented a successful pairing-free certificateless signcryption scheme without a secure channel. The findings show that, in terms of communication overhead, the scheme is better than the relevant existing schemes and could be a better option for the Internet of Things (IoT) following the implementation of a proper revocation mechanism.

The IoV system is vulnerable to a variety of security and privacy risks. As a result, a lightweight security system is necessary to protect against a variety of known and unknown threats. Since all of the above schemes rely on complex cryptographic methods such as elliptic curves and bilinear pairing, they all have high computing and communication costs and are not compatible with the IoV system.

3. Preliminaries

This section introduces some of the fundamental concepts and materials that are used in our proposed model.

Hyperelliptic Curve

The $\mathbb{h}\mathbb{E}\mathbb{C}$ is the compressed form of $\mathbb{E}\mathbb{C}$, which contains fewer key and parameters size [30,31]. Equation (1) represents the $\mathbb{h}\mathbb{E}\mathbb{C}$ of genus $\mathcal{G} \geq 2$ over a finite field \mathcal{U}_p , where \mathcal{G} is the non-intersecting curves that is not touching each other when it is drawn on surface.

$$\mathbb{h}\mathbb{E}\mathbb{C}: Q^2 + H(V)Q = F(V) \pmod{p} \quad (1)$$

where $H(V)$ and $F(V)$ are polynomials with coefficients in \mathcal{U}_p . So, the degree of $H(V)$ at most \mathcal{G} and the degree of $F(V)$ is equal to $2\mathcal{G} + 1$. In a sense of non-singularity, there must not exist a point on $\mathbb{h}\mathbb{E}\mathbb{C}$ that satisfy the equation: $2Q + H(V) = 0$ and $H'(V) - F'(V) = 0$.

HEDHP Problem: Suppose $\kappa, \alpha, \mathcal{D}$ is the assumed occurrence of $\mathbb{h}\mathbb{E}\mathbb{C}$ computational defihelman problem (HEDHP). Finding the two unknown variables that are κ and α which belongs to $\{1, 2, 3, p - 1\}$ is called HEDHP. The symbols used in the scheme are illustrated in Table 1.

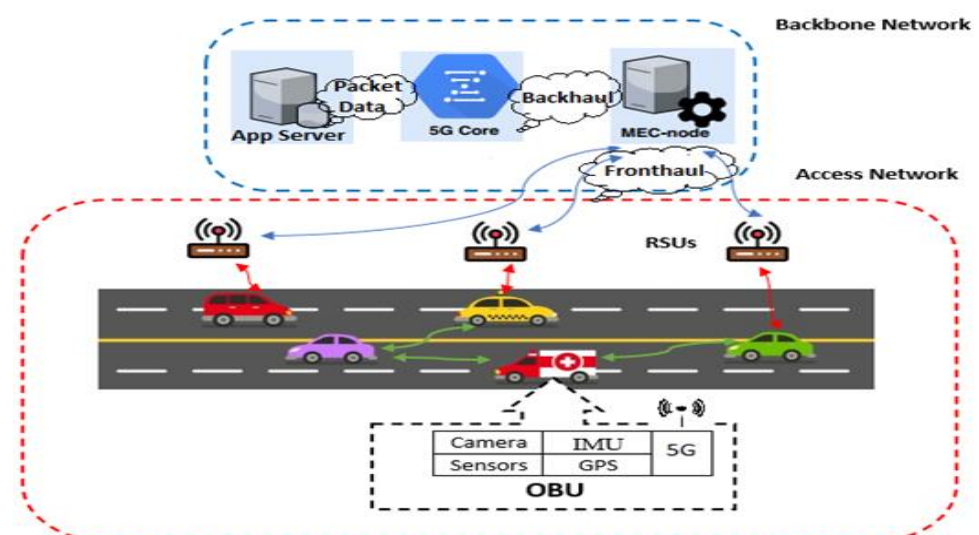
Table 1. Notations used in proposed scheme.

S. No	Symbol	Descriptions
1	σ	The predefined security parameter
2	$\mathcal{G} \geq 2$	Genus of hyper elliptic curve with not less than 2
3	\mathcal{U}_p	finite field of order p and $p \geq 2^{80}$
4	h_v, h_w, h_x, h_y	Irreversible hash functions
5	η and γ	The private key and public key of KGC respectively
6	$E_{\mathcal{F}}$ and $D_{\mathcal{F}}$	An encryption and decryption algorithm
7	\mathcal{D}	Divisor on hyper elliptic curve
8	ξ	The global parameter set
9	ID_s and ID_r	Identity of sender and receiver
10	\mathcal{O}_s and \mathcal{O}_r	Secret value of sender and receiver
11	PB_s and PB_r	Public key of sender and receiver
12	P_s and P_r	Private key of sender and receiver
13	\mathcal{X}, m	Ciphertext and plaintext
14		The equality is hold or not
15	β_s and β_r	The partial private key of sender and receiver
16	Ψ	The signcrypted text generated by sender
17	\perp	Used for null

4. System Models

4.1. Network Model

In this subsection, we propose a network model consisting of cars, Onboard Units (OBUs), Regional Transportation Authority (RTA), and the Roadside Units (RSUs) that make an edge cluster and the vehicles that provide event-driven messages collected through their sensors, as shown in Figure 1, to create the operation and applicability of the proposed scheme in the IoV setting. As a primary component of the proposed network architecture, vehicles are considered. Each vehicle is fitted with OBU, which consists of a camera, IMU, sensors, and a GPS device that can handle various application scenarios. It offers connectivity from vehicle-to-vehicle and vehicle-to-infrastructures. In the proposed framework, RTA acts as a trusted agency that offers registrations facility for the vehicles and edge nodes. RSUs with fixed communication infrastructure are located on the roadside. The key function of RSUs is to collect and validate the event-driven messages provided by the vehicles. RSUs often function as a gateway node in order to access the backbone network. It accomplishes connectivity between these entities using 5G mobile networks. The proposed network model also guarantees flow isolation by individually tunneling data traffic to the app-server from the MEC-node.

**Figure 1.** Proposed network model.

4.2. Threat Model

In this section, we briefly discuss three kinds of threats that will happen with the proposed scheme [32]. The first one will be in the form of an indistinguishable scramble text attack (IN-ACLS-CA) against the opponent of \mathcal{O}_1 and \mathcal{O}_2 , where \mathcal{O}_1 is the Type 1 opponent, which has the capability to replace the public key of a user and struggle with getting access to the plaintext of a transmitted scramble text. Further, \mathcal{O}_2 is the Type 2 opponent, which has the capability to access the private key of KGC and struggle with getting access to the plaintext from the transmitted encrypted-text. The second will be in the form of an existentially unforgeable against adaptive chosen plaintext attack (EF-ACLS-PA) against the opponent of \mathcal{O}_1 and \mathcal{O}_2 . The third will be an indistinguishability of scramble text/identity attack (ANO-ACLS-CA) against the opponent of \mathcal{O}_1 and \mathcal{O}_2 . The basic introduction about IN-ACLS-CA, against the opponent of \mathcal{O}_1 and \mathcal{O}_2 , is presented in the following *Game 1* and *Game 2*. Further, EF-ACLS-PA and ANO-ACLS-CA, are explained in *Game 3*, *Game 4*, *Game 5*, and *Game 6*.

Game 1: Let \mathcal{O}_1 be the Type 1 opponent in the IN-ACLS-CA, and φ can act as challenger and its task to interact with \mathcal{O}_1 during setup and queries of this Game. The task of φ is to solve HEDHP for \mathcal{O}_1 .

Setup: φ compute γ, ξ , and give γ and ξ to \mathcal{O}_1 .

h_i Query (q_i): \mathcal{O}_1 ask for these queries, φ searches whether the requested value subsists in list L_i . If it is subsisting, φ can send this exist value to \mathcal{O}_1 . Otherwise, φ pick a random value and send it to \mathcal{O}_1 , and update L_i accordingly.

CSV Query (q_{csv}): \mathcal{O}_1 needs φ to accomplish CSV Query. After reception, a φ search whether the requested value subsists in list L_k . If it is subsisting, φ send the secret value to \mathcal{O}_1 . Otherwise, φ calls *construct secret value algorithm* and generates the secret value, send it to \mathcal{O}_1 and update L_k accordingly.

CPPK Query (q_{cppk}): \mathcal{O}_1 needs φ to accomplish CPPK Query. After reception, a φ search whether the requested value subsists in list L_k . If it is subsisting, φ send the partial private key to \mathcal{O}_1 . Otherwise, φ calls *construct partial private key algorithm* and generates the partial private key, send it to \mathcal{O}_1 , and update L_k accordingly.

CPBPK Query (q_{cpbpk}): \mathcal{O}_1 needs φ to accomplish CPBPK Query. After reception, a φ search whether the requested value subsists in list L_k . If it is subsisting, φ send the public and private key to \mathcal{O}_1 . Otherwise, φ calls *construct public and private key algorithm* and generates the public and private key, send it to \mathcal{O}_1 , and update L_k accordingly.

PBKR Query (q_{pbkr}): Upon the request of \mathcal{O}_1 , φ convert the user public into his own selected public key.

Signcryption Query (q_s): \mathcal{O}_1 needs φ to make Signcryption Query, φ check if $ID_s \neq ID_d$ then it calls CPBPK Query, produce Ψ and send it to \mathcal{O}_1 .

Un-Signcryption Query (q_{us}): \mathcal{O}_1 needs φ to make Un-Signcryption Query, φ produce m and send it to \mathcal{O}_1 .

Challenge: Here, m_1 and m_2 are the two identical sizes but dissimilar type of messages that are selected by \mathcal{O}_1 for φ . Further, φ chooses a bit $\vartheta \in \{0, 1\}$ at unsystematic way and uses m_ϑ to develop Ψ^* . Then, it returns Ψ^* to \mathcal{O}_1 .

Note that \mathcal{O}_1 can carry with all the above queries except *Un-Signcryption Query (q_{us})* against Ψ^* , further the private key part of *CPBPK Query (q_{cpbpk})* and *CPPK Query (q_{cppk})* of a device, whose public key is replaced.

Guess: \mathcal{O}_1 provides ϑ^* , if $\vartheta^* = \vartheta$, then \mathcal{O}_1 succeeded and φ gives the solution of HEDHP. Otherwise, φ returns \perp .

Game 2: Let \mathcal{O}_2 be the Type 2 opponent in the IN-ACLS-CA and φ can act as challenger and its task to interact with \mathcal{O}_2 during setup and queries of this Game. The task of φ is to solve HEDHP for \mathcal{O}_2 .

Setup: φ give γ, η , and ζ to \mathcal{O}_2 .

Queries: The queries execution is same as *Game 1* except *PBKR Query* (q_{pbkr}).

Challenge: Here, m_1 and m_2 are the two identical sizes but dissimilar type of messages that are selected by \mathcal{O}_2 for φ . Further, φ chooses a bit $\vartheta \in \{0, 1\}$ at unsystematic way and uses m_ϑ to develop Ψ^* . Then, it returns Ψ^* to \mathcal{O}_2 .

Note that \mathcal{O}_2 can carry with all the above queries except *Un-Signcryption Query* (q_{us}) against Ψ^* , further *CSV Query* (q_{csv}) for target identity.

Guess: \mathcal{O}_2 provides ϑ^* , if $\vartheta^* = \vartheta$, then \mathcal{O}_2 succeeded and φ gives the solution of *HEDHP*. Otherwise, φ returns \perp .

Game 3: Let \mathcal{O}_1 be the Type 1 opponent in the EF-ACLS-PA and φ can act as challenger and its task to interact with \mathcal{O}_1 during setup and queries of this Game. The task of φ is to solve *HEDHP* for \mathcal{O}_1 .

Setup: φ give γ and ζ to \mathcal{O}_1 .

The execution of *\mathcal{H}_i Query* (q_i), *Device Key Query* (q_{dk}), *CSV Query* (q_{csv}), *CPPK Query* (q_{cppk}), *CPBPK Query* (q_{cpbpk}), *PBKR Query* (q_{pbkr}), *Signcryption Query* (q_s), and *Un-Signcryption Query* (q_{us}) is same as *Theorem 1*.

Forgery: \mathcal{O}_1 uses m and identity to forge Ψ^* , if Ψ^* is falsified efficaciously, then it gets the solution of *HEDHP*. Otherwise, it returns \perp .

Note that \mathcal{O}_1 can carry with all the above queries except *Un-Signcryption Query* (q_{us}) against Ψ^* .

Game 4: Let \mathcal{O}_2 be the Type 2 opponent in the EF-ACLS-PA and φ can act as challenger and its task to interact with \mathcal{O}_2 during setup and queries of this Game. The task of φ is to solve *HEDHP* for \mathcal{O}_2 .

Setup: φ give γ, η , and ζ to \mathcal{O}_2 .

The execution of *\mathcal{H}_i Query* (q_i), *Device Key Query* (q_{dk}), *CSV Query* (q_{csv}), *CPPK Query* (q_{cppk}), *CPBPK Query* (q_{cpbpk}), *Signcryption Query* (q_s), and *Un-Signcryption Query* (q_{us}) is same as *Theorem 1*.

Forgery: \mathcal{O}_2 uses m and identity to forge Ψ^* , if Ψ^* is falsified efficaciously, then it gets the solution of *HEDHP*. Otherwise, it returns \perp . In this execution, the *Signcryption Query* cannot acquire Ψ^* .

Game 5: Let \mathcal{O}_1 be the Type 1 opponent in the ANO-ACLS-CA and φ can act as challenger and its task to interact with \mathcal{O}_1 during setup and queries of this Game. The task of φ is to solve *HEDHP* for \mathcal{O}_1 .

Setup: φ give γ and ζ to \mathcal{O}_1 .

The execution of *\mathcal{H}_i Query* (q_i), *Device Key Query* (q_{dk}), *CSV Query* (q_{csv}), *CPPK Query* (q_{cppk}), *CPBPK Query* (q_{cpbpk}), *PBKR Query* (q_{pbkr}), *Signcryption Query* (q_s), and *Un-Signcryption Query* (q_{us}) is same as *Theorem 1*.

Challenge: Here, ID_1 and ID_2 are the two identities that are selected by \mathcal{O}_1 for φ . Further, φ chooses a bit $e \in \{0, 1\}$ at unsystematic way to develop Ψ^* . Then, it returns Ψ^* to \mathcal{O}_1 .

Guess: \mathcal{O}_1 provides e^* , if $e^* = e$, then \mathcal{O}_1 succeeded and φ gives the solution of *HEDHP*. Otherwise, φ returns \perp .

Game 6: Let \mathcal{O}_2 be the Type 2 opponent in the ANO-ACLS-CA and φ can act as challenger and its task to interact with \mathcal{O}_2 during setup and queries of this Game. The task of φ is to solve *HEDHP* for \mathcal{O}_2 .

Setup: φ give γ, η , and ζ to \mathcal{O}_2 .

The execution of *\mathcal{H}_i Query* (q_i), *Device Key Query* (q_{dk}), *CSV Query* (q_{csv}), *CPPK Query* (q_{cppk}), *CPBPK Query* (q_{cpbpk}), *PBKR Query* (q_{pbkr}), *Signcryption Query* (q_s), and *Un-Signcryption Query* (q_{us}) is same as *Theorem 1*.

Challenge: Here, ID_1 and ID_2 are the two identities that are selected by \mathbb{O}_2 for φ . Further, φ chooses a bit $e \in \{0, 1\}$ at unsystematic way to develop Ψ^* . Then, it returns Ψ^* to \mathbb{O}_2 .

Guess: \mathbb{O}_2 provides e^* , if $e^* = e$, then \mathbb{O}_2 succeeded and φ gives the solution of HEDHP. Otherwise, φ returns \perp .

5. Proposed Scheme

5.1. Syntax of the Proposed Scheme

- i. **Setup:** KGC makes η as his private key and γ as his public key and also generates ξ as a global parameter set.
- ii. **Keys Generation:** It contains Construct Secrete Value, Construct Partial Private Key, and Construct Public and Private Key, which are as follow:
 - Construct Secrete Value (CSV): The device selects \mathbb{Q}_d and computes \mathbb{O}_d , then sends its identity (ID_d) and \mathbb{O}_d to KGC using a secure channel.
 - Construct Partial Private Key (CPPK): KGC selects δ_d , computes ζ_d , calculates μ_d , makes Y_d , and calculates β_d . Finally, KGC sends ζ_d and β_d to the device with ID_d through a secure link.
 - Construct Public and Private Key (CPBPK): The device with identity (ID_d), computes Y_d and Z_d . Then, set PB_d as a public key and P_d as a private key.
- iii. **Signcryption:** Considering the input parameters such as ξ as his private key and identities (P_s, ID_s), message m , and identity of receiver ID_r , the sending device generates and send $\Psi = (\mathcal{X}, \mathcal{K}, \Omega)$ to receiver.
- iv. **Un-Signcryption:** On the other hand, the receiving device executes the algorithm by considering the received parameter Ψ , and verifies its authenticity.

5.2. Proposed Algorithm

In this phase, we explain the proposed scheme construction steps [27], which are as follows:

- i. **Setup:** Considering a security input σ , the KGC performs the following operations:
 - Define \mathbb{hEc} of genus $\mathcal{G} \geq 2$ over a finite field \mathcal{U}_p , where \mathcal{G} represents the non-intersecting curves.
 - KGC selects $h_v, h_w, h_x,$ and h_y , as irreversible hash functions.
 - KGC also selects η where $0 \leq \eta \leq p$ and computes $\gamma = \eta \cdot \mathcal{D}$.
 - KGC set η as his private key and γ as his public key.
 - KGC selects $E_{\mathcal{G}}$ and $D_{\mathcal{G}}$ as encryption and decryption algorithms.
 - KGC sets $\xi = \{\mathcal{G} \geq 2, \mathcal{U}_p, \mathbb{hEc}, \mathcal{D}, p, h_v, h_w, h_x, h_y, \gamma, E_{\mathcal{G}}, D_{\mathcal{G}}\}$ as a global parameter set.
- ii. **Keys Generation:** It contains Construct Secrete Value, Construct Partial Private Key, and Construct Public and Private Key, which are calculated as follows:
 - Construct Secrete Value (CSV): The device sends its identity (ID_d) and \mathbb{O}_d to KGC using a secure channel, where $\mathbb{O}_d = \mathbb{Q}_d \cdot \mathcal{D}$ and $0 \leq \mathbb{Q}_d \leq p$.
 - Construct Partial Private Key (CPPK): KGC selects δ_d where $0 \leq \delta_d \leq p$ and then, by considering the receptions values that are ID_d and \mathbb{O}_d , it computes $\zeta_d = \delta_d \cdot \mathcal{D}$, calculates $\mu_d = h_v(ID_d, \mathbb{O}_d, \zeta_d)$, makes $Y_d = \delta_d + \mu_d \cdot \eta$, and calculates $\beta_d = Y_d + h_w(ID_d, \eta, \mathbb{O}_d)$. Finally, KGC sends ζ_d and β_d to the device with ID_d through secure link.
 - Construct Public and Private Key (CPBPK): The device with identity (ID_d) considers the reception values that are ζ_d and β_d , computes $Y_d = \beta_d - h_w(ID_d, \gamma, \mathbb{Q}_d)$ and $Z_d = Y_d \cdot \mathcal{D}$. Then, it checks $Y_d \cdot \mathcal{D} \zeta_d + \mu_d \cdot \gamma$. After successful execution, the device then with identity (ID_d) accepts the values of ζ_d and β_d , and sets $PB_d = (\mathbb{O}_d, Z_d)$ as a public key and $P_d = (\mathbb{Q}_d, Y_d)$ as a private key respectively.
- iii. **Signcryption:** Considering the input parameters such as ξ as his private key and identities (P_s, ID_s), message m , and identity of receiver ID_r , the sending device selects

ℓ where $0 \leq \ell \leq p$ and computes $k = \ell \cdot \mathcal{D}$, $\mathcal{F} = h_x(\ell \cdot \mathcal{O}_j)$ and $\mathcal{X} = E_{\mathcal{F}}(m)$, $\mathcal{V} = h_y(m, ID_r, PB_s)$, $\Omega = \mathbb{Q}_s + Y_s + \mu_s \cdot \ell$, respectively, and then sends $\Psi = (\mathcal{X}, k, \Omega)$ to receiver.

iv. **Un-Signcryption:** Finally, the receiving device executes the algorithm by considering the received parameter Ψ , and verifies its authenticity as follows:

- Compute $\mathcal{F} = h_x(\mathbb{Q}_r \cdot k)$ and $m = D_{\mathcal{F}}(\mathcal{X})$
- Compute $\mathcal{V}' = h_y(m, ID_r, PB_s)$ and check $\Omega \cdot \mathcal{D} \mathbb{O}_s + Z_s + \mu_s \cdot k$, if it is successfully processed then receiver accept Ψ .

5.3. Correctness

The device with identity (ID_d), checks the validity of ζ_d and β_d as follows:

$$\begin{aligned} & Y_d \cdot \mathcal{D} \zeta_d + \mu_d \cdot \gamma \\ &= Y_d \cdot \mathcal{D} = (\delta_d + \mu_d \cdot \eta) \cdot \mathcal{D} = (\delta_d \cdot \mathcal{D} + \mu_d \cdot \eta \cdot \mathcal{D}) \\ &= (\zeta_d + \mu_d \cdot \gamma) \text{ where } \zeta_d = \delta_d \cdot \mathcal{D} \text{ and } \gamma = \eta \cdot \mathcal{D} \\ & Y_d \cdot \mathcal{D} = (\zeta_d + \mu_d \cdot \gamma), \text{ hence proved.} \end{aligned}$$

The receiver makes the decryption key as follows:

$$\begin{aligned} \mathcal{F} &= h_x(\mathbb{Q}_r \cdot k) \\ &= h_x(\mathbb{Q}_r \cdot \ell \cdot \mathcal{D}) = h_x(\ell \cdot \mathbb{O}_r) \text{ where } \mathbb{O}_r = \mathbb{Q}_r \cdot \mathcal{D} \\ &= h_x(\ell \cdot \mathbb{O}_r) = \mathcal{F} \text{ hence proved.} \end{aligned}$$

The receiver checks the validity of $\Psi = (\mathcal{X}, k, \Omega)$ as followed

$$\begin{aligned} & \Omega \cdot \mathcal{D} \mathbb{O}_s + Z_s + \mu_s \cdot k \\ &= \Omega \cdot \mathcal{D} = (\mathbb{Q}_s + Y_s + \mu_s \cdot \ell) \cdot \mathcal{D} \text{ where } \Omega = \mathbb{Q}_s + Y_s + \mu_s \cdot \ell \\ &= (\mathbb{Q}_s \cdot \mathcal{D} + Y_s \cdot \mathcal{D} + \mu_s \cdot \ell \cdot \mathcal{D}) \\ &= \mathbb{O}_s + Z_s + \mu_s \cdot k \text{ where } \mathbb{O}_s = \mathbb{Q}_s \cdot \mathcal{D}, Z_s = Y_s \cdot \mathcal{D}, \text{ and } k = \ell \cdot \mathcal{D} \\ & \Omega \cdot \mathcal{D} = \mathbb{O}_s + Z_s + \mu_s \cdot k \text{ hence proved.} \end{aligned}$$

6. Security Analysis

In this section, we provide the security proofs for our scheme on the basis of random oracle model.

It includes the six games, which are explained in the following theorems.

Theorem 1. Let \mathbb{O}_1 be the Type 1 opponent in the IN-ACLS-CA and its winning advantage is ω which cannot be ignored during a time t . The φ can act as challenger and its task to give an access when \mathbb{O}_1 ask for the queries such as Device Key Query (q_{dk}), CSV Query (q_{csv}), CPPK Query (q_{cppk}), CPBPK Query (q_{cpbpbk}), Public Key Replacement (PBKR) Query (q_{pbkr}), Signcryption Query (q_s), Un-Signcryption Query (q_{us}), and h_i Query (q_i) where ($i = v, w, x, y$). Further, within the time t it can help to recuperate the solution of HEDHP for \mathbb{O}_1 . Here, the advantage of \mathbb{O}_1 will be as $\omega \succcurlyeq 2(\omega - q_{us} q_y / 2^\sigma) / n q_x$.

Proof of Theorem 1: Suppose $\kappa, \alpha \cdot \mathcal{D}$ is the assumed occurrence of HEDHP and the task of φ with \mathbb{O}_1 is to find the two unknown variables that are κ and α . For this task, \mathbb{O}_1 with by using the following sub-steps. \square

Setup. φ select a random number η , compute $\gamma = \eta \cdot \mathcal{D}$, make ξ , and give γ and ξ to \mathbb{O}_1 .

h_v Query (q_v): The triple $(ID_j, \mathbb{O}_j, \zeta_j)$ is reserved as input, and \mathbb{O}_1 needs φ to accomplish h_v Query. After reception, φ searches whether triple $(ID_j, \mathbb{O}_j, \zeta_j)$ subsists in list L_v . If it is subsisting, μ_j can send by φ to \mathbb{O}_1 . Otherwise, φ pick μ_j in a random manner, send μ_j to \mathbb{O}_1 , and update L_v using $(ID_j, \mathbb{O}_j, \zeta_j, \mu_j)$.

h_w Query (q_w): The pair $(ID_j, \eta \cdot \mathbb{O}_j)$ and $(ID_j, \gamma \cdot \mathbb{O}_j)$ is reserved as input, and \mathbb{O}_1 needs φ to accomplish h_w Query. After reception, φ searches whether pair $(ID_j, \eta \cdot \mathbb{O}_j)$ and $(ID_j, \gamma \cdot \mathbb{O}_j)$ is subsists in list L_w . If it is subsisting, ε_j and ∂_j can send by φ to \mathbb{O}_1 . Otherwise, φ pick ε_j

and ∂_j in a random manner, send ε_j and ∂_j to \mathcal{O}_1 , and update L_w using $(ID_j, \eta, \mathcal{O}_j, \varepsilon_j)$ and $(ID_j, \gamma, \mathcal{Q}_j, \partial_j)$.

\mathcal{H}_x Query (q_x): The pair (ℓ, \mathcal{O}_j) is reserved as input, and \mathcal{O}_1 needs φ to accomplish \mathcal{H}_x Query. After reception, a φ search whether pair (ℓ, \mathcal{O}_j) is subsists in list L_x . If it is subsisting, \mathcal{F}_j can send by φ to \mathcal{O}_1 . Otherwise, φ pick \mathcal{F}_j in a random manner, send \mathcal{F}_j to \mathcal{O}_1 , and update L_x using $(\ell, \mathcal{O}_j, \mathcal{F}_j)$.

\mathcal{H}_y Query (q_y): The triple (m, ID_j, PB_j) is reserved as input, and \mathcal{O}_1 needs φ to accomplish \mathcal{H}_y Query. After reception, a φ search whether pair (m, ID_j, PB_j) is subsists in list L_y . If it is subsisting, \mathcal{V}_j can send by φ to \mathcal{O}_1 . Otherwise, φ pick \mathcal{V}_j in a random manner, send \mathcal{V}_j to \mathcal{O}_1 , and update L_y using $(m, ID_j, PB_j, \mathcal{V}_j)$.

Device Key Query (q_{dk}): The tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ is reserved as input, and \mathcal{O}_1 needs φ to accomplish Device Key Query. After reception, a φ search whether pair $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ is subsists in list L_k . If it is subsisting, φ reserves the tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$. Otherwise, φ do the following steps.

- If $ID_j \neq ID_d$, φ pick \mathcal{Q}_j, β_j in a random manner, set $\mathcal{O}_j = \mathcal{Q}_j \cdot \mathcal{D}$, $Y_j = \beta_j - \mathcal{H}_w(ID_j, \gamma, \mathcal{Q}_j)$, $Z_j = Y_j \cdot \mathcal{D}$, $PB_j = (\mathcal{O}_j, Z_j)$, $P_j = (\mathcal{Q}_j, Y_j)$, and then update L_k using $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ and L_v using $(ID_j, \mathcal{O}_j, \zeta_j, \mu_j)$.
- If $ID_j = ID_d$, φ pick \mathcal{Q}_j, δ_j in a random manner, set $\mathcal{O}_j = \mathcal{Q}_j \cdot \mathcal{D}$, $P_j \perp$, $\zeta_j = \delta_j \cdot \mathcal{D}$, $Y_j = \delta_j + \mathcal{H}_v(ID_j, \mathcal{O}_j, \zeta_j) \cdot \eta$, $Z_j = Y_j \cdot \mathcal{D}$, $PB_j = (\mathcal{O}_j, Z_j)$, and then update L_k using $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ and L_v using $(ID_j, \mathcal{O}_j, \zeta_j, \mu_j)$.

CSV Query (q_{csv}): The tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ is reserved as input, and \mathcal{O}_1 needs φ to accomplish CSV Query. After reception, a φ search whether tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ is subsists in list L_k . If it is subsisting, φ sends \mathcal{Q}_j to \mathcal{O}_1 . Otherwise, φ calls Device Key Query and generates the tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ and sends \mathcal{Q}_j to \mathcal{O}_1 . Then, it updates L_k using $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$.

CPPK Query (q_{cppk}): The tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ is reserved as input, and \mathcal{O}_1 needs φ to accomplish CPPK Query. After reception, φ does the following steps.

- If $ID_j = ID_d$, φ returns \perp .
- If $ID_j \neq ID_d$, φ calls Device Key Query, generates the tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ and send β_j to \mathcal{O}_1 . Then, update L_k using $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$.

CPBPK Query (q_{cpbpk}): Upon the request of \mathcal{O}_1 , φ first of all give the response for public key that are, a φ search whether tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ subsists in list L_k . If it is subsisting, φ send PB_j to \mathcal{O}_1 . Otherwise, φ calls Device Key Query and generates the tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ and send PB_j to \mathcal{O}_1 .

Secondly, φ first of all give the response for private key that are followed.

- If $ID_j = ID_d$, φ returns \perp .
- If $ID_j \neq ID_d$, φ calls Device Key Query, generates the tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ and send P_j to \mathcal{O}_1 . Then, update L_k using $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$.

PBKR Query (q_{pbkr}): Upon the request of \mathcal{O}_1 , φ convert PB_j into PB_j' and update L_k using $(ID_j, PB_j', P_j, \mathcal{Q}_j, \beta_j)$.

Signcryption Query (q_s): \mathcal{O}_1 needs φ to make Signcryption Query, φ check if $ID_s \neq ID_d$ then it calls CPBPK Query and performs the following computations.

- Select ℓ where $0 \leq \ell \leq p$ and compute $\mathcal{K} = \ell \cdot \mathcal{D}$
- Compute $\mathcal{F} = \mathcal{H}_x(\ell, \mathcal{O}_j)$ and $\mathcal{X} = E_{\mathcal{F}}(m)$
- Compute $\mathcal{V} = \mathcal{H}_y(m, ID_j, PB_j)$
- Compute $\Omega = \mathcal{Q}_j + Y_j + \mu_j \cdot \ell$ and send $\Psi = (\mathcal{X}, \mathcal{K}, \Omega)$ to \mathcal{O}_1

Un-Signcryption Query (q_{us}): \mathcal{O}_1 needs φ to make Un-Signcryption Query, φ check if $ID_j = ID_d$, φ returns \perp . Otherwise, it performs the following computations.

- Search for a tuple $(ID_j, P_j, PB_j, Q_j, \beta_j)$ in list L_k and compute $\mathcal{F} = \mathcal{H}_x(Q_j, \mathcal{K})$ and $m = D_{\mathcal{F}}(\mathcal{X})$
- Check $\Omega, \mathcal{D}_{\mathcal{C}_s} + Z_s + \mu_s \cdot \mathcal{K}$, if it is successfully processed then φ send m to \mathcal{C}_1 . Otherwise, φ returns \perp .

Challenge: m_1 and m_2 are the two identical sizes but dissimilar type of messages that are selected by \mathcal{C}_1 for φ . Further, φ chooses a bit $\vartheta \in \{0, 1\}$ at an unsystematic way and uses m_{ϑ} to develop Ψ^* . The detail steps are followed.

- Set $\mathcal{K} = \alpha \cdot PB_d, \ell \cdot \mathcal{C}_d = \alpha(\gamma + PB_d)$, and $\mathcal{F} = \mathcal{H}_x(\ell \cdot \mathcal{C}_d)$
- Set $\mathcal{X}^* = E_{\mathcal{F}}(m)$ and select Ω randomly
- Return $\Psi^* = (\mathcal{X}^*, \Omega, \mathcal{K})$ to \mathcal{C}_1

Note that \mathcal{C}_1 can carry with all the above queries, except *Un-Signcryption Query* (q_{us}), against Ψ^* .

Guess: \mathcal{C}_1 provides ϑ^* , if $\vartheta^* = \vartheta$, then \mathcal{C}_1 succeeded and φ gives the solution of $\kappa \cdot \alpha \cdot \mathcal{D} = \ell \cdot \mathcal{C}_d - \mathcal{K}$. Otherwise, φ returns \perp . We can observe the following probability events from the aforementioned explanations.

- \mathcal{H}_y hash offers a valid scramble text during q_{us} and its probability as $q_y/2^\sigma$
- \mathcal{C}_1 needs φ to perform *Un-Signcryption Query* (q_{us}) during the attack process, the decryption success probability of φ as $\omega_{us} = \omega - q_{us}q_y/2^\sigma$
- During the guess phase the probability for $\kappa \cdot \alpha \cdot \mathcal{D}$ as $2/nq_x$

So, \mathcal{C}_1 the advantage of \mathcal{C}_1 will be as $\omega \succcurlyeq 2(\omega - q_{us}q_y/2^\sigma)/nq_x$, for the solution of HEDHP.

Theorem 2. Suppose \mathcal{C}_2 is the Type 2 opponent in the IN-ACLS-CA and its winning advantage is ω which cannot be ignored during a time t . The φ can act as challenger and its task to give an access when \mathcal{C}_2 ask for the queries as performed in Theorem 1 except PBKR Query (q_{pbkr}). Further, within the time t it can help to recuperate the solution of HEDHP for \mathcal{C}_2 . Here, the advantage of \mathcal{C}_2 will be as $\omega \succcurlyeq 2(\omega - q_{us}q_y/2^\sigma)/nq_x$.

Proof of Theorem 2: Assume $\kappa \cdot \alpha \cdot \mathcal{D}$ is the expected manifestation of HEDHP and the job of φ with \mathcal{C}_2 is to discover the two unknown variables that are κ and α . For this mission, \mathcal{C}_2 with by using the following sub-steps. \square

Setup: φ choose a random number η , calculate $\gamma = \eta \cdot \mathcal{D}$, make ζ , and give γ, η , and ζ to \mathcal{C}_2 . Then, set $K = \mathcal{D}$.

\mathcal{H}_i Query (q_i): The process for this query is same as Theorem 1.

Device Key Query (q_{dk}): The tuple $(ID_j, P_j, PB_j, Q_j, \beta_j)$ is reserved as input, and \mathcal{C}_2 needs φ to accomplish Device Key Query. After reception, a φ search whether pair $(ID_j, P_j, PB_j, Q_j, \beta_j)$ is subsists in list L_k . If it is subsisting, φ reserves the tuple $(ID_j, P_j, PB_j, Q_j, \beta_j)$. Otherwise, φ do the following steps.

- If $ID_j = ID_d$, φ pick Q_j, δ_j in a random manner, compute $\zeta_j = \delta_j \cdot \mathcal{D}, Y_j = \delta_j + \mathcal{H}_v(ID_j, Q_j, \zeta_j) \cdot \eta, Z_j = Y_j \cdot \mathcal{D}, PB_j = (Q_j, Z_j)$, and then update L_k using $(ID_j, P_j, PB_j, Q_j, \beta_j)$ and L_v using $(ID_j, Q_j, \zeta_j, \mu_j)$, where $Q_j = Q_j \cdot \mathcal{D}, P_j \perp$.
- If $ID_j \neq ID_d$, φ pick Q_j, β_j in a random manner, set $Q_j = Q_j \cdot \mathcal{D}, Y_j = \beta_j - \mathcal{H}_w(ID_j, \gamma, Q_j), Z_j = Y_j \cdot \mathcal{D}, PB_j = (Q_j, Z_j), P_j = (Q_j, Y_j)$, and then update L_k using $(ID_j, P_j, PB_j, Q_j, \beta_j)$ and L_v using $(ID_j, Q_j, \zeta_j, \mu_j)$.

CSV Query (q_{csv}): \mathcal{C}_2 needs φ to accomplish CSV Query. After reception, a φ does the following executions.

- If $ID_j = ID_d$, φ returns \perp .
- If $ID_j \neq ID_d$, φ calls Device Key Query, generates the tuple $(ID_j, P_j, PB_j, Q_j, \beta_j)$ and send Q_j to \mathcal{C}_2 . Then, update L_k using $(ID_j, P_j, PB_j, Q_j, \beta_j)$.

CPPK Query (q_{cppk}): The tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ is reserved as input, and \mathcal{O}_2 needs φ to accomplish CPPK Query. After reception, a φ searches whether tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ subsists in list L_k . If it is subsisting, φ send β_j to \mathcal{O}_2 . Otherwise, φ calls Device Key Query and generates the tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ and send β_j to \mathcal{O}_2 . Then, update L_k using $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$.

CPBPK Query (q_{cpbpk}): Upon the request of \mathcal{O}_2 , φ first of all gives the response for public key that are, a φ searches whether tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ subsists in list L_k . If it is subsisting, φ sends PB_j to \mathcal{O}_2 . Otherwise, φ calls Device Key Query and generates the tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ and send PB_j to \mathcal{O}_2 .

Secondly, φ first of all gives the response for private key that are followed.

- If $ID_j = ID_d$, φ returns \perp .
- If $ID_j \neq ID_d$, φ calls Device Key Query, generates the tuple $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$ and send P_j to \mathcal{O}_2 . Then, update L_k using $(ID_j, P_j, PB_j, \mathcal{Q}_j, \beta_j)$.

Signcryption Query (q_s): The process for this query is same as Theorem 1.

Un-Signcryption Query (q_{us}): The process for this query is same as Theorem 1.

Challenge: m_1 and m_2 are the two identical sizes but dissimilar type of messages that are selected by \mathcal{O}_2 for φ . Further, φ chooses a bit $\vartheta \in \{0, 1\}$ at unsystematic way and uses m_ϑ to develop Ψ^* . The detail steps are followed.

- Set $\mathcal{K} = \alpha.(PB_d + T)$, where $T = \gamma + K$, $\ell.\mathcal{O}_d = \alpha(\gamma + PB_d)$, and $\mathcal{F} = \mathcal{H}_x(\ell.\mathcal{O}_d)$
- Set $\mathcal{X}^* = E_{\mathcal{F}}(m)$ and select Ω randomly
- Return $\Psi^* = (\mathcal{X}^*, \Omega, \mathcal{K})$ to \mathcal{O}_2

Note that \mathcal{O}_2 can carry with all the above queries, except Un-Signcryption Query (q_{us}), against Ψ^* .

Guess: \mathcal{O}_2 provides ϑ^* , if $\vartheta^* = \vartheta$, then \mathcal{O}_2 succeeded and φ gives the solution of $\kappa.\alpha.\mathcal{D} = \ell.\mathcal{O}_d - \mathcal{K}$. Otherwise, φ returns \perp .

So, we mentioned explanations.

- \mathcal{H}_y hash offers a valid scramble text during q_{us} and its probability as $q_y/2^\sigma$
- \mathcal{O}_2 needs φ to perform Un-Signcryption Query (q_{us}) during the attack process, the decryption success probability of φ as $\omega_{us} = \omega - q_{us}q_y/2^\sigma$
- During the guess phase the probability for $\kappa.\alpha.\mathcal{D}$ as $2/nq_x$

For \mathcal{O}_2 the advantage of \mathcal{O}_2 will be as $\omega \succ 2(\omega - q_{us}q_y/2^\sigma)/nq_x$, for the solution of HEDHP.

Theorem 3. Suppose \mathcal{O}_1 is the Type 1 opponent in the EF-ACLS-PA and its winning advantage is ω which cannot be ignored during a time t . The φ can act as challenger and its task to give an access when \mathcal{O}_1 ask for the queries as performed in Theorem 1. Further, within the time t it can help to recuperate the solution of HEDHP for \mathcal{O}_1 . Here, the advantage of \mathcal{O}_1 will be as $\omega \succ (\omega - q_s/2^\sigma)/2$.

Proof of Theorem 3: Assume $\kappa.\alpha.\mathcal{D}$ is the expected manifestation of HEDHP and the job of φ with \mathcal{O}_1 is to discover the two unknown variables that are κ and α . For this mission, \mathcal{O}_1 with by using the following sub-steps. \square

Setup. φ chooses a random number η , calculates $\gamma = \eta.\mathcal{D}$, make ξ , and gives γ and ξ to \mathcal{O}_1 .

The execution of \mathcal{H}_i Query (q_i), Device Key Query (q_{dk}), CSV Query (q_{csv}), CPPK Query (q_{cppk}), CPBPK Query (q_{cpbpk}), PBKR Query (q_{pbkr}), Signcryption Query (q_s), and Un-Signcryption Query (q_{us}) are same as Theorem 1.

Forgery: \mathcal{O}_1 forges Ψ^* and m , if $\Omega.\mathcal{D}\mathcal{O}_s + Z_s + \mu_s.\mathcal{k}$, is successfully processed, Ψ^* falsified efficaciously, describing $\mathcal{O}_d = \frac{\mathcal{O}_d}{\alpha}$ and $\ell.\mathcal{O}_d = \alpha(\gamma + PB_d)$, φ computes $\ell.\mathcal{O}_d = \mathcal{O}_d + \kappa.\alpha.\mathcal{D}$, returns $\ell.\mathcal{O}_d - \mathcal{O}_d = \kappa.\alpha.\mathcal{D}$, and $\kappa.\alpha.\mathcal{D}$ is the solution of HEDHP. Otherwise, it returns \perp .

Hence, we can observe the following probability events from the aforementioned explanations.

- The success probability of Signcryption Query (q_s) φ as $\omega - q_s/2^\sigma$
- During the forgery phase, the success probability of solving $\kappa.\mathcal{D}$ as $1/2$

So, \mathcal{O}_1 the advantage of \mathcal{O}_1 will be as $\omega \succ (\omega - q_s/2^\sigma)/2$, for the solution of HEDHP.

Theorem 4. Suppose \mathcal{O}_2 is the Type 2 opponent in the EF-ACLS-PA and its winning advantage is ω which cannot be ignored during a time t . The φ can act as challenger and its task to give an access when \mathcal{O}_2 ask for the queries as performed in Theorem 1 except PBKR Query (q_{pbkr}). Further, within the time t it can help to recuperate the solution of HEDHP for \mathcal{O}_2 . Here, the advantage of \mathcal{O}_2 will be as $\omega \succ (\omega - q_s/2^\sigma)/2$.

Proof of Theorem 4: Assume $\kappa.\alpha.\mathcal{D}$ is the expected manifestation of HEDHP and the job of φ with \mathcal{O}_2 is to discover the two unknown variables that are κ and α . For this mission, \mathcal{O}_2 will by using the following sub-steps. \square

Setup. The execution of this phase is same as Theorem 2.

The execution of \mathcal{h}_i Query (q_i), Device Key Query (q_{dk}), CSV Query (q_{csv}), CPPK Query (q_{cppk}), CPBPK Query (q_{cpbpk}), Signcryption Query (q_s), and Un-Signcryption Query (q_{us}) are same as Theorem 1.

Forgery: \mathcal{O}_2 forges Ψ^* and m , if $\Omega.\mathcal{D}\mathcal{O}_s + Z_s + \mu_s.\mathcal{k}$, is successfully processed, Ψ^* falsified efficaciously, describing $\mathcal{O}_d = \mathcal{O}_d/\alpha$ and $\ell.\mathcal{O}_d = \alpha(\gamma + PB_d)$, φ compute $\ell.\mathcal{O}_d = \mathcal{O}_d + \kappa.\alpha.\mathcal{D}$, returns $\ell.\mathcal{O}_d - \mathcal{O}_d = \kappa.\alpha.\mathcal{D}$, and $\kappa.\alpha.\mathcal{D}$ is the solution of HEDHP. Otherwise, it returns \perp .

Therefore, we can observe the following probability events from the aforementioned explanations.

- The success probability of Signcryption Query (q_s) φ as $\omega - q_s/2^\sigma$
- During the forgery phase the success probability of solving $\kappa.\mathcal{D}$ as $1/2$

For \mathcal{O}_2 the advantage of \mathcal{O}_2 will be as $\omega \succ (\omega - q_s/2^\sigma)/2$, for the solution of HEDHP.

Theorem 5. Let \mathcal{O}_1 be the Type 1 opponent in the ANO-ACLS-CA and its winning advantage is ω which cannot be ignored during a time t . The φ can act as challenger and its task to give access when \mathcal{O}_1 ask for the queries same as Theorem 1. Further, within the time t it can help to recuperate the solution of HEDHP for \mathcal{O}_1 . Here, the advantage of \mathcal{O}_1 will be as $\omega \succ 2(\omega - q_{us}q_y/2^\sigma)/nq_x$.

Proof of Theorem 5: Suppose $\kappa.\alpha.\mathcal{D}$ is the assumed occurrence of HEDHP and the task of φ with \mathcal{O}_1 is to find the two unknown variables that are κ and α . For this task, \mathcal{O}_1 will by using the following sub-steps. \square

The execution of \mathcal{h}_i Query (q_i), Device Key Query (q_{dk}), CSV Query (q_{csv}), CPPK Query (q_{cppk}), CPBPK Query (q_{cpbpk}), PBKR Query (q_{pbkr}), Signcryption Query (q_s), and Un-Signcryption Query (q_{us}) are same as Theorem 1.

Challenge: Here, ID_1 and ID_2 are the two identities that are selected by \mathcal{O}_1 for φ . Further, φ chooses a bit $e \in \{0, 1\}$ at unsystematic way to develop Ψ^* . The detail steps are followed.

- Set $\mathcal{k} = \alpha.PB_d$, $\ell.\mathcal{O}_d = \alpha(\gamma + PB_d)$, and $\mathcal{F} = \mathcal{h}_x(\ell.\mathcal{O}_d)$
- Set $\mathcal{X}^* = E_{\mathcal{F}}(m)$ and select Ω randomly
- Return $\Psi^* = (\mathcal{X}^*, \Omega, \mathcal{k})$ to \mathcal{O}_1

Note that \mathcal{O}_1 can carry with all the above queries except *Un-Signcryption Query* (q_{us}) against Ψ^* .

Guess: \mathcal{O}_1 provides e^* , if $e^* = e$, then \mathcal{O}_1 succeeded and φ gives the solution of $\kappa.\alpha.\mathcal{D} = \ell.\mathcal{O}_d - \mathbb{k}$. Otherwise, φ returns \perp .

Hence, we can observe the following probability events from the aforementioned explanations.

- \mathbb{h}_y hash offers a valid scramble text during q_{us} and its probability as $q_y/2^\sigma$
- \mathcal{O}_1 needs φ to perform *Un-Signcryption Query* (q_{us}) during the attack process, the decryption success probability of φ as $\omega_{us} = \omega - q_{us}q_y/2^\sigma$
- During the guess phase, the probability for $\kappa.\alpha.\mathcal{D}$ as $2/nq_x$

So, \mathcal{O}_1 the advantage of \mathcal{O}_1 will be as $\omega \succcurlyeq 2(\omega - q_{us}q_y/2^\sigma)/nq_x$, for the solution of HEDHP.

Theorem 6. Let \mathcal{O}_2 be the Type 2 opponent in the ANO-ACLS-CA and its winning advantage is ω which cannot be ignored during a time t . The φ can act as challenger and its task to give an access when \mathcal{O}_1 ask for the queries same as Theorem 1 except PBKR Query (q_{pbkr}). Further, within the time t it can help to recuperate the solution of HEDHP for \mathcal{O}_2 . Here, the advantage of \mathcal{O}_2 will be as $\omega \succcurlyeq 2(\omega - q_{us}q_y/2^\sigma)/nq_x$.

Proof of Theorem 6: Suppose $\kappa.\alpha.\mathcal{D}$ is the assumed occurrence of HEDHP and the task of φ with \mathcal{O}_2 is to find the two unknown variables that are κ and α . For this task, \mathcal{O}_2 will by using the following sub-steps. \square

Setup: The execution of this phase as Theorem 2.

The execution of \mathbb{h}_i Query (q_i), Device Key Query (q_{dk}), CSV Query (q_{csv}), CPPK Query (q_{cppk}), CPBPK Query (q_{cpbpk}), Signcryption Query (q_s), and Un-Signcryption Query (q_{us}) are same as Theorem 1.

Challenge: Here, ID_1 and ID_2 are the two identities sizes that are selected by \mathcal{O}_2 for φ . Further, φ chooses a bit $e \in \{0, 1\}$ at unsystematic way to develop Ψ^* . The detail steps are followed.

- Set $q = \alpha.(PB_d + T)$, where $T = \gamma + K$, $\ell.\mathcal{O}_d = \alpha(\gamma + PB_d)$, and $\mathcal{F} = \mathbb{h}_x(\ell.\mathcal{O}_d)$
- Set $\mathcal{X}^* = E_{\mathcal{F}}(m)$ and select Ω randomly
- Return $\Psi^* = (\mathcal{X}^*, \Omega, \mathbb{k})$ to \mathcal{O}_2

Note that \mathcal{O}_2 can carry with all the above queries, except *Un-Signcryption Query* (q_{us}), against Ψ^* .

Guess: \mathcal{O}_2 provides e^* , if $e^* = e$, then \mathcal{O}_2 succeeded and φ gives the solution of $\kappa.\alpha.\mathcal{D} = \ell.\mathcal{O}_d - \mathbb{k}$. Otherwise, φ returns \perp .

Therefore, we can observe the following probability events from the aforementioned explanations.

- \mathbb{h}_y hash offers a valid scramble text during q_{us} and its probability as $q_y/2^\sigma$
- \mathcal{O}_2 needs φ to perform *Un-Signcryption Query* (q_{us}) during the attack process, the decryption success probability of φ as $\omega_{us} = \omega - q_{us}q_y/2^\sigma$
- During the guess phase the probability for $\kappa.\alpha.\mathcal{D}$ as $\frac{2}{n}q_x$

So, \mathcal{O}_2 the advantage of \mathcal{O}_2 will be as $\omega \succcurlyeq 2(\omega - q_{us}q_y/2^\sigma)/nq_x$, for the solution of HEDHP.

7. Cost Analysis

7.1. Computational Cost

The proposed scheme is compared, in terms of computational cost, with the relevant existing schemes proposed by Zhou [25], Cao and Ge [26], Luo and Ma [27],

Rastegari et al. [28], and Karati et al. [29], as shown in Table 2. The existing schemes utilize exponential operations, pairing, and elliptic curve point multiplication, which are costlier options. Comparatively, our scheme is based on the hyperelliptic divisor multiplication. The time required for processing a single Elliptic Curve Point Multiplication (ECPM) is 0.97 ms; bilinear pairing is 14.90 ms; pairing-based point multiplications is 4.31 ms; modular exponentiation is 1.25 ms [33]. The Hyperelliptic Curve Divisor Multiplication (HCDM) is assumed to be 0.48 milliseconds [34–38]. Multi-precision Integer and Rational Arithmetic C Library (MIRACL) [39] is used to measure the computational performance. The simulation results are obtained with a machine equipped with the specifications as follows: Intel Core i7-4510U CPU @ 2.0 GHz, 8 GB RAM, and Windows 7 Home Basic 64-bit Operating System [33]. It is evident that our scheme is efficient, in terms of computational cost, from the findings illustrated in Table 2 and Figure 2.

Table 2. Computational cost regarding major operations and milliseconds (MS).

Schemes	Signcryption	Unsigncryption	Total	Total (ms)
Caixue Zhou [25]	$\mathcal{P} + 7\mathcal{E}$	$4\mathcal{P} + 5\mathcal{E}$	$5\mathcal{P} + 12\mathcal{E}$	$11.1 + 22.09 = 33.19$
Cao and Ge [26]	$7\mathcal{E}_\varphi$	$5\mathcal{E}_\varphi$	$12\mathcal{E}_\varphi$	11.64
Luo and Ma [27]	$6\mathcal{E}_\varphi$	$5\mathcal{E}_\varphi$	$11\mathcal{E}_\varphi$	10.67
Rastegari et al. [28]	$2\mathcal{P} + 4\mathcal{E}$	$8\mathcal{P} + 2\mathcal{E}$	$10\mathcal{P} + 6\mathcal{E}$	50.60
Karati et al. [29]	$3\mathcal{E}_\varphi$	$4\mathcal{E}_\varphi$	$7\mathcal{E}_\varphi$	6.79
Proposed scheme	$3\mathcal{H}_\varphi$	$3\mathcal{H}_\varphi$	$6\mathcal{H}_\varphi$	2.88

Note: \mathcal{E} = single exponential operation, \mathcal{P} = pairing based point multiplication, \mathcal{H}_φ = hyperelliptic curve divisor multiplication, and \mathcal{E}_φ = elliptic curve point multiplication.

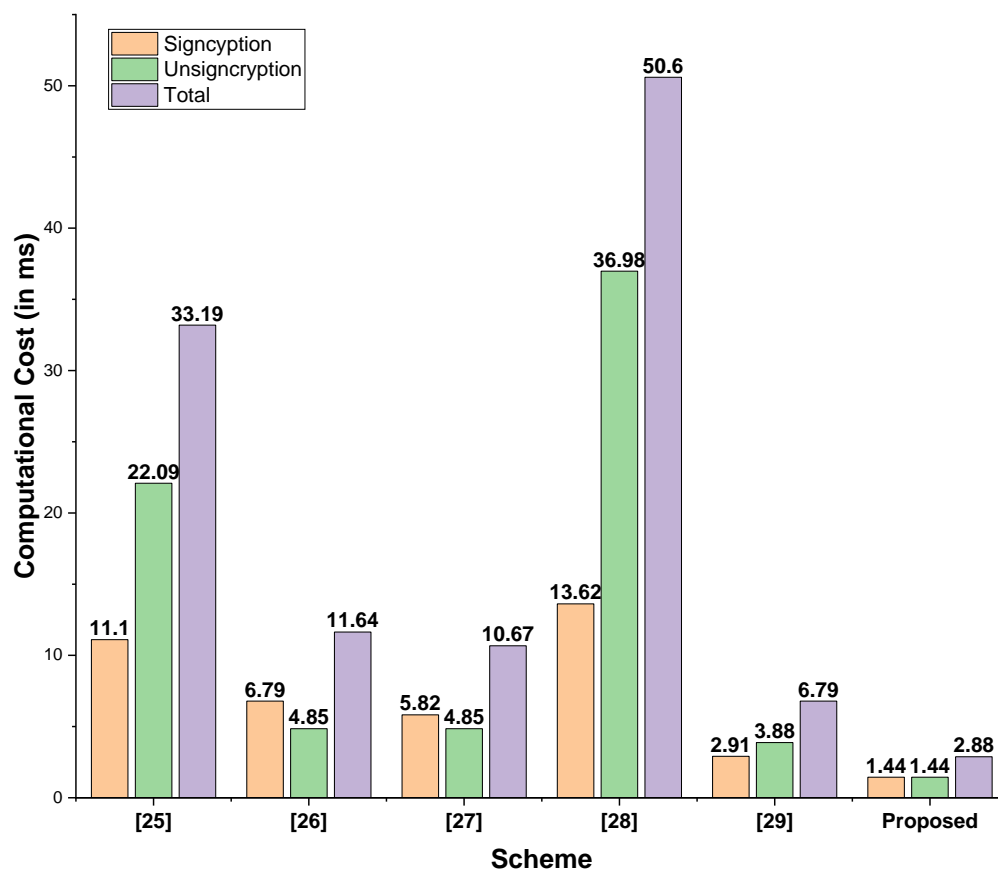


Figure 2. Total computational cost (in milliseconds).

7.2. Communication Cost

In this subsection, the proposed approach is compared, in terms of communication cost, with the schemes presented by Zhou [25], Cao and Ge [26], Luo and Ma [27], Rastegari et al. [28], and Karati et al. [29]. In Table 3, the comparative analysis is provided for communication cost, which is also illustrated in Figure 3. The variables where, m = plaintext, \mathcal{G} = bilinear pairing bits, q = elliptic curve bits, and n = hyperelliptic curve bits used, along with the respective values shown in Table 4, are given as follows.

Table 3. Communication cost comparisons.

Schemes	Communication Cost	Total (in Bits)
Caixue Zhou [25]	$ m + 5 \mathcal{G} $	6144
Cao and Ge [26]	$ m + 2 q $	1344
Luo and Ma [27]	$ m + 2 q $	1344
Rastegari et al. [28]	$ m + 4 \mathcal{G} $	5120
Karati et al. [29]	$ m + 2 q $	1344
Proposed scheme	$ m + 2 n $	1184

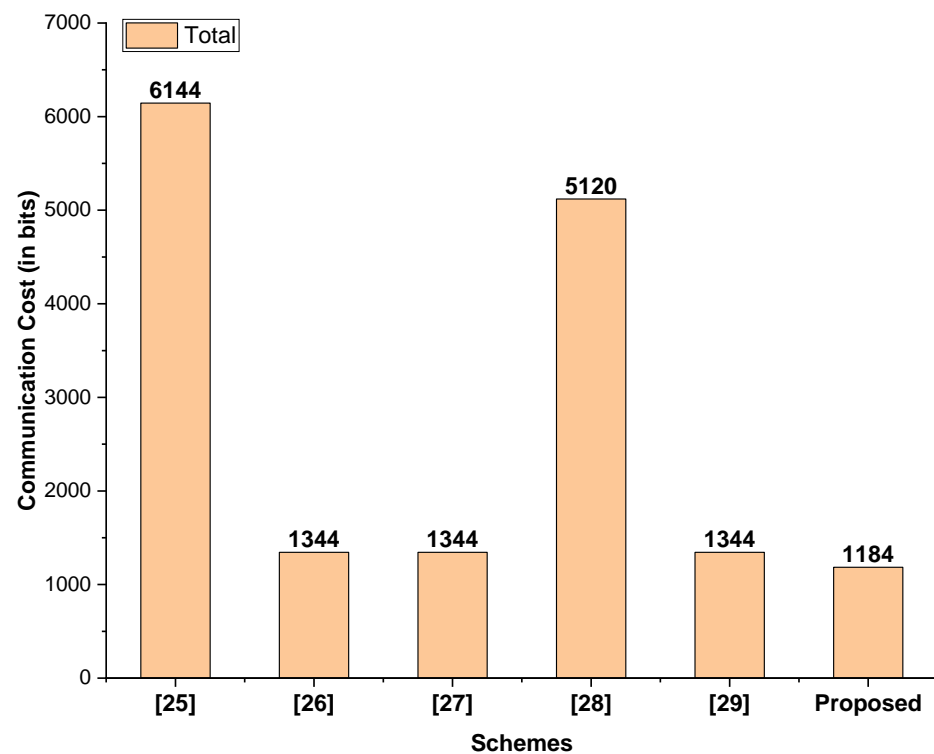


Figure 3. Total communication cost (in bits).

Table 4. Variables used for communication cost comparison.

Variable	Value
$ m $	1024 bits
$ q $	160 bits
$ n $	80 bits
$ \mathcal{G} $	1024 s

7.3. Security Functionalities

The comparisons, with respect to security functionalities, with the existing schemes are listed in Table 5. The outcomes of these comparisons are based on the security parameters as follows: unforgeability, confidentiality, and anonymity. From the Table 5,

it can be witnessed that none of the schemes proposed by Zhou [25], Cao and Ge [26], Luo and Ma [27], Rastegari et al. [28], and Karati et al. [29] offer anonymity.

Table 5. Comparison with relevant existing schemes. Symbol: \checkmark satisfy the security functionality, $\not\checkmark$: does not satisfy the security functionality.

Schemes	Unforgeability	Confidentiality	Anonymity
Caixue Zhou [25]	\checkmark	\checkmark	$\not\checkmark$
Cao and Ge [26]	\checkmark	\checkmark	$\not\checkmark$
Luo and Ma [27]	\checkmark	\checkmark	$\not\checkmark$
Rastegari et al. [28]	\checkmark	\checkmark	$\not\checkmark$
Karati et al. [29]	\checkmark	\checkmark	$\not\checkmark$
Proposed scheme	\checkmark	\checkmark	\checkmark

8. Conclusions

Internet of Vehicles (IoV) is the set of Internet of Things (IoT) with Intelligent Transport Systems (ITS) to provide information for common services, which builds the foundation of a next generation of traffic management systems. However, the environments of IoV could be dangerous in the absence of security protections. It gives an enormous opportunity to malicious attackers to modify, intercept, delete, or even insert false information during the on-going transmission. In this paper, using the HEC concept, we introduced an anonymous certificateless signcryption scheme for the IoV environment to resolve such deficiencies. The HEC approach is efficient at producing small keys and is therefore appropriate for a highly dynamic IoV environment. Moreover, because of the certificateless cryptography mechanism, the proposed scheme avoids the key escrow problem. The scheme also ensures receiver anonymity in open wireless channels. The formal security analysis demonstrates the ability of the proposed scheme to thwart different cyber-attacks, and it is competitive with its current counterparts in terms of computational and communication costs. In the future, we intend to implement the same scheme by including the ability to distribute partial private keys over an open channel; this ensures that the KGC would no longer need a secure channel to share partial private keys with vehicles in the IoV system.

Author Contributions: Conceptualization, I.U. and M.A.K.; Formal analysis, I.U. and M.A.K.; Methodology I.U., M.H.A. and M.A.K.; Resources I.U., M.H.A. and M.A.K.; Software, I.U., M.H.A. and M.A.K.; Supervision, M.A.K.; Writing—original draft, I.U., M.H.A., R.N. and M.A.K.; Writing—review and editing, I.U., M.H.A., R.N. and M.A.K. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the financial support from CRIM, Universiti Kebangsaan Malaysia, under the Dana Padanan Kolaborasi (DPK), under the grant ref number: DPK-2020-014.

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. Siddiqui, S.A.; Mahmood, A.; Sheng, Q.Z.; Suzuki, H.; Ni, W. A Survey of Trust Management in the Internet of Vehicles. *Electronics* **2021**, *10*, 2223. [\[CrossRef\]](#)
2. Cho, J.-H.; Swami, A.; Chen, I.-R. A Survey on Trust Management for Mobile Ad Hoc Networks. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 562–583. [\[CrossRef\]](#)
3. Alfadhli, S.A.; Lu, S.; Fatani, A.; Al-Fedhly, H.; Ince, M. SD2PA: A fully safe driving and privacy-preserving authentication scheme for VANETs. *Hum. Cent. Comput. Inf. Sci.* **2020**, *10*, 38. [\[CrossRef\]](#)
4. Mahmood, A.; Zhang, W.E.; Sheng, Q.Z. Software-Defined Heterogeneous Vehicular Networking: The Architectural Design and Open Challenges. *Futur. Internet* **2019**, *11*, 70. [\[CrossRef\]](#)

5. Ullah, I.; Khan, M.A.; Khan, F.; Jan, M.A.; Srinivasan, R.; Mastorakis, S.; Hussain, S.; Khattak, H. An Efficient and Secure Multi-message and Multi-receiver Signcryption Scheme for Edge Enabled Internet of Vehicles. *IEEE Internet Things J.* **2021**, *1*. Available online: https://ieeexplore.ieee.org/abstract/document/9466941/?casa_token=8H8AaNzIKYAAAAA:GHQCSORNkCi9k6NDdka5rqZmc7zZARKW5qiMM5o1Ypg7NDygvVW7yux7ZXoJrZIAD3cyQWOgX91pNfg (accessed on 1 August 2021). [[CrossRef](#)]
6. Storck, C.R.; Duarte-Figueiredo, F. A Survey of 5G Technology Evolution, Standards, and Infrastructure Associated With Vehicle-to-Everything Communications by Internet of Vehicles. *IEEE Access* **2020**, *8*, 117593–117614. [[CrossRef](#)]
7. Sharma, S.; Kaushik, B. A survey on internet of vehicles: Applications, security issues & solutions. *Veh. Commun.* **2019**, *20*, 100182. [[CrossRef](#)]
8. Zou, Y.; Zhu, J.; Wang, X.; Hanzo, L. A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends. *Proc. IEEE* **2016**, *104*, 1727–1765. [[CrossRef](#)]
9. Nkenyereye, L.; Tama, B.A.; Shahzad, M.K.; Choi, Y.-H. Secure and Blockchain-Based Emergency Driven Message Protocol for 5G Enabled Vehicular Edge Computing. *Sensors* **2019**, *20*, 154. [[CrossRef](#)] [[PubMed](#)]
10. Bagga, P.; Das, A.K.; Wazid, M.; Rodrigues, J.J.P.C.; Park, Y. Authentication Protocols in Internet of Vehicles: Taxonomy, Analysis, and Challenges. *IEEE Access* **2020**, *8*, 54314–54344. [[CrossRef](#)]
11. Zhang, L.; Guo, C.; Xu, Z.; Zhang, L. An Anonymous Signcryption Scheme Based on One-Off Public Key. In Proceedings of the International Conference on Cyberspace Technology (CCT 2013), Beijing, China, 23 November 2013; pp. 81–86.
12. Khan, M.A.; Ullah, I.; Nisar, S.; Noor, F.; Qureshi, I.M.; Khanzada, F.U.; Amin, N.U. An Efficient and Provably Secure Certificateless Key-Encapsulated Signcryption Scheme for Flying Ad-hoc Network. *IEEE Access* **2020**, *8*, 36807–36828. [[CrossRef](#)]
13. Suárez-Albela, M.; Fraga-Lamas, P.; Fernández-Caramés, T.M. A Practical Evaluation on RSA and ECC-Based Cipher Suites for IoT High-Security Energy-Efficient Fog and Mist Computing Devices. *Sensors* **2018**, *18*, 3868. [[CrossRef](#)]
14. Ullah, I.; Amin, N.U.; Khan, M.A.; Khattak, H.; Kumari, S. An Efficient and Provable Secure Certificate-Based Combined Signature, Encryption and Signcryption Scheme for Internet of Things (IoT) in Mobile Health (M-Health) System. *J. Med. Syst.* **2021**, *45*, 4. [[CrossRef](#)] [[PubMed](#)]
15. Barbosaand, M.; Farshim, P. Certificateless Signcryption. In Proceedings of the 2008 ACM symposium on Information, computer and communications security, Tokyo, Japan, 18–20 March 2008; pp. 18–20.
16. Xie, W.; Zhang, Z. Efficient and Provably Secure Certificateless Signcryption from Bilinear Maps. In Proceedings of the 2010 IEEE International Conference on Wireless Communications, Networking and Information Security, Beijing, China, 25–27 June 2010; pp. 558–562.
17. Liu, Z.; Hu, Y.; Zhang, X.; Ma, H. Certificateless signcryption scheme in the standard model. *Inf. Sci.* **2010**, *180*, 452–464. [[CrossRef](#)]
18. Selvi, S.S.D.; Vivek, S.S.; Rangan, C.P. Security Weaknesses in Two Certificateless Signcryption Schemes. *IACR Cryptol. Eprint Arch.* **2010**, *2010*, 92.
19. Shi, W.; Kumar, N.; Gong, P.; Zhang, Z. Cryptanalysis and improvement of a certificateless signcryption scheme without bilinear pairing. *Front. Comput. Sci.* **2014**, *8*, 656–666. [[CrossRef](#)]
20. Wahid, A.; Mambo, M. Implementation of certificateless signcryption based on elliptic curve using Javascript. *Int. J. Comput. Inform. (IJCANDI)* **2016**, *1*, 90–100.
21. Zhou, C.; Gao, G.; Cui, Z. Certificateless Signcryption in the Standard Model. *Wirel. Pers. Commun.* **2016**, *92*, 495–513. [[CrossRef](#)]
22. Rastegari, P.; Berenjkoub, M. An efficient certificateless signcryption scheme in the standard model. *ISeCure* **2017**, *9*, 3–16.
23. Yu, H.; Yang, B. Pairing-Free and Secure Certificateless Signcryption Scheme. *Comput. J.* **2017**, *60*, 1187–1196. [[CrossRef](#)]
24. Lin, X.-J.; Sun, L.; Qu, H.; Liu, D. Cryptanalysis of A Pairing-Free Certificateless Signcryption Scheme. *Comput. J.* **2017**, *61*, 539–544. [[CrossRef](#)]
25. Zhou, C. Certificateless Signcryption Scheme Without Random Oracles. *Chin. J. Electron.* **2018**, *27*, 1002–1008. [[CrossRef](#)]
26. Cao, L.; Ge, W. Analysis of Certificateless Signcryption Schemes and Construction of a Secure and Efficient Pairing-free one based on ECC. *KSII Trans. Internet Inf. Syst.* **2018**, *12*, 4527–4547. [[CrossRef](#)]
27. Luo, W.; Ma, W. Secure and Efficient Data Sharing Scheme Based on Certificateless Hybrid Signcryption for Cloud Storage. *Electronics* **2019**, *8*, 590. [[CrossRef](#)]
28. Rastegari, P.; Susilo, W.; Dakhalian, M. Efficient Certificateless Signcryption in the Standard Model: Revisiting Luo and Wan's Scheme from Wireless Personal Communications (2018). *Comput. J.* **2019**, *62*, 1178–1193. [[CrossRef](#)]
29. Karati, A.; Fan, C.-I.; Huang, J.-J. An Efficient Pairing-Free Certificateless Signcryption Without Secure Channel Communication During Secret Key Issuance. *Procedia Comput. Sci.* **2020**, *171*, 110–119. [[CrossRef](#)]
30. Naresh, V.S.; Sivaranjani, R.; Murthy, N.V. Provable secure lightweight hyper elliptic curve-based communication system for wireless sensor networks. *Int. J. Commun. Syst.* **2018**, *31*, e3763. [[CrossRef](#)]
31. Ullah, S.; Li, X.-Y.; Zhang, L. A Review of Signcryption Schemes Based on Hyper Elliptic Curve. In Proceedings of the 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM), Chengdu, China, 10–11 August 2017; pp. 51–58.
32. He, D.; Ma, M.; Zeadally, S.; Kumar, N.; Liang, K. Certificateless Public Key Authenticated Encryption With Keyword Search for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3618–3627. [[CrossRef](#)]

33. Zhou, C.; Zhao, Z.; Zhou, W.; Mei, Y. Certificateless Key-Insulated Generalized Signcryption Scheme without Bilinear Pairings. *Secur. Commun. Netw.* **2017**, *2017*, 8405879. [[CrossRef](#)]
34. Khan, M.A.; Qureshi, I.M.; Ullah, I.; Khan, S.; Khanzada, F.; Noor, F. An Efficient and Provably Secure Certificateless Blind Signature Scheme for Flying Ad-Hoc Network Based on Multi-Access Edge Computing. *Electronics* **2019**, *9*, 30. [[CrossRef](#)]
35. Khan, M.A.; Ullah, I.; Kumar, N.; Oubbati, O.S.; Qureshi, I.M.; Noor, F.; Khanzada, F.U. An Efficient and Secure Certificate-Based Access Control and Key Agreement Scheme for Flying Ad-Hoc Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 4839–4851. [[CrossRef](#)]
36. Khan, M.A.; Ullah, I.; Alkhalifah, A.; Rehman, S.U.; Shah, J.A.; Uddin, I.I.; Alsharif, M.H.; Algarni, F. A Provable and Privacy-Preserving Authentication Scheme for UAV-Enabled Intelligent Transportation Systems. *IEEE Trans. Ind. Inform.* **2021**, *1*. Available online: https://ieeexplore.ieee.org/abstract/document/9506932/?casa_token=KM4jty33DqIAAAAA:ovZBkgiHCawZEePPUFvMga8slG8CwddPd-xcxzteSDE1dRg88q8EqImgEAahNHIG1pCA0wzTPzS5HA (accessed on 2 August 2021). [[CrossRef](#)]
37. Khan, M.A.; Ullah, I.; Nisar, S.; Noor, F.; Qureshi, I.M.; Khanzada, F.; Khattak, H.; Aziz, M.A. Multiaccess Edge Computing Empowered Flying Ad Hoc Networks with Secure Deployment Using Identity-Based Generalized Signcryption. *Mob. Inf. Syst.* **2020**, *2020*, 8861947. [[CrossRef](#)]
38. Khan, M.A.; Shah, H.; Rehman, S.U.; Kumar, N.; Ghazali, R.; Shehzad, D.; Ullah, I. Securing Internet of Drones With Identity-Based Proxy Signcryption. *IEEE Access* **2021**, *9*, 89133–89142. [[CrossRef](#)]
39. Shamus Sofware Ltd. Miracl Library. Available online: <http://github.com/miracl/MIRACL> (accessed on 2 August 2021).