applied sciences

MDPI

Article

Secure Key Agreement and Authentication Protocol for Message Confirmation in Vehicular Cloud Computing

JoonYoung Lee [1], SungJin Yu [1], MyeongHyun Kim [1], YoungHo Park [1,*], SangWoo Lee [2] and BoHeung Chung [2]

[1] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea; harry250@knu.ac.kr (J.L.); darkskiln@knu.ac.kr (S.Y.); kimmyeong123@knu.ac.kr (M.K.)
[2] Electronics and Telecommunications Research Institute, Daejeon 34129, Korea; ttomlee@etri.re.kr (S.L.); bhjung@etri.re.kr (B.C.)
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

Received: 29 July 2020; Accepted: 4 September 2020; Published: 9 September 2020

Abstract: With the development of vehicular ad-hoc networks (VANETs) and Internet of vehicles (IoVs), a large amount of useful information is generated for vehicle drivers and traffic management systems. The amount of vehicle and traffic information is as large as the number of vehicles and it is enormous when compared to vehicle calculation and storage performance. To resolve this problem, VANET uses a combined cloud computing technology, called vehicular cloud computing (VCC), which controls vehicle-related data, and helps vehicle drivers directly or indirectly. However, VANETs remain vulnerable to attacks such as tracking, masquerade and man-in-the-middle attacks because VANETs communicate via open networks. To overcome these issues, many researchers have proposed secure authentication protocols for message confirmation with vehicular cloud computing. However, many researchers have pointed out that some proposed protocols use ideal tamper-proof devices (TPDs). They demonstrated that realistic TPDs cannot prevent adversaries attack. Limbasiya et al. presented a message confirmation scheme for vehicular cloud computing using a realistic TPD in order to prevent these problems. However, their proposed scheme still has security weaknesses over a TPD and does not guarantee mutual authentication. This paper proposes a secure key agreement and authentication protocol to address the security weaknesses inherent in the protocol of Limbasiya et al. The suggested protocol withstands malicious attacks and ensures secure mutual authentication for privacy-preserving. We prove that the proposed protocol can provide session key security using Real-Or-Random (ROR) model. We also employed Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation tool to show that the proposed protocol is able to defeat replay and man-in-the-middle attacks. Furthermore, we established that the proposed protocol can resist other malicious attacks by conducting the informal security analysis. We proved that our proposed protocol is lightweight and suitable for VCC environments.

Keywords: VANET; vehicular cloud computing; message confirmation; cryptanalysis; mutual authentication; AVISPA; ROR model

1. Introduction

Embedded devices, such as sensors and on-board units (OBUs) of Internet of vehicles (IoVs), collect a variety of information including traffic conditions and road conditions. The driver and traffic management system can share and use various services by sharing this information with other IoVs. Therefore, the role of embedded devices in IoV has been increasing with the increase in the size of a vehicle system, and traffic information has been increasing in complexity. However, enhancing the

computing power and extending the storage space of the embedded devices is not technically possible or financially viable. Vehicular cloud computing (VCC) has been suggested to address these limitations of embedded devices. VCC is a system that controls vehicle-related data. IoVs send traffic information to the vehicular cloud. Subsequently, other vehicles can obtain information from the vehicular cloud when required.

However, this information is transmitted through open and unsecured channels; therefore, malicious attackers can threaten the VCC environment. When malicious attackers steal and manipulate traffic information, the lives of pedestrians, and drivers are endangered. Therefore, VCC should provide a key agreement with secure authentication that protects the information by providing message confirmations. Therefore, many studies have been suggested for VCC to provide a secure authentication. Recently, proposed schemes are used for ideal tamper-proof devices (TPDs). The TPD is safe from malicious attacks and it is impossible to tamper with it, according to the proposed schemes. However, the ideal TPD has a strong assumption that an attacker cannot obtain or tamper with values in the TPD according to [1–4]. They pointed out that attackers can obtain stored values in the realistic TPD through power analysis attacks and side-channel attacks.

In 2019, Limbasiya et al. [4] presented a message confirmation scheme based batch verification, and a VCC environment to address OBU computation limitations. They proposed secure authentication to address an issue, where realistic TPD cannot prevent side channel and power analysis attacks. And also, they proposed a session key agreement for secure transmitting information. However, we figure out that the proposed protocol of Limbasyia et al. is vulnerable to side channel attacks of TPD and cannot defeat various attacks, including session key disclosure and impersonation attacks. Their protocol is also unable to provide secure mutual authentication and privacy-preserving.

This paper suggests a secure key agreement and authentication protocol for message confirmation in the VCC environment in order to overcome their security flaws. We design the protocol to use only the hash function and XOR operation, and assume that realistic OBUs can be deployed realistic environments. We also assume that an attacker can perform side channel attacks on TPDs to obtain secret values stored in TPDs. Consequently, our proposed protocol does not strongly rely on TPDs; instead, it uses only OBUs. Furthermore, we propose a key agreement protocol for secure data transmission. We analyze the security aspects of our proposed protocol using Real-Or-Random (ROR) model and the Automated Validation of Internet Security Protocols and Applications (AVISPA) software for the formal analysis. This paper also compares the computation cost and security features with [4] and previous similar protocols. Finally, this paper demonstrates that the proposed protocol is able to be deployed in a real VANET.

The rest of this paper is as follows: Section 2 reviews related works and presents the network model, threat model and notations used in this paper. At Section 3, we review the Limbasiya et al.'s protocol. We cryptanalyze its security flaws in Section 4. At Sections 5 and 6, we propose a secure key agreement and authentication protocol for VCC environment in VANET and perform informal and formal security analysis. We use ROR model, AVISPA simulation, and informal analysis for verification. Subsequently, we compare the security properties and computational cost with related previous researches in Section 7. Finally, in Section 8, we present our conclusions with the results of the proposed protocol.

2. Related Works

This section reviews the literature regarding the authentication protocol for vehicle communication and examines the limitations of ideal TPDs. We also introduce the network model, threat model, and notations used in this paper.

2.1. Literature Reviews

This section briefly reviews secure authentication protocols and key agreement protocols that are involved in two aspects, i.e., general authentication protocols for vehicular communication or VANETs, and authentication protocol using a practical TPD that points out the limitations of the ideal TPD.

2.1.1. Authentication Protocol for Vehicle Communication

Authentication is considered a basic security service that allows subjects to mutually authenticate with other subjects [5–9]. In 2007, Lin et al. [10] suggested an authentication protocol while using a group signature based on bilinear pairing. In their protocol, the verifier can verify multiple signatures simultaneously, which improves authentication efficiency. However, Zhang et al. [11] pointed out a significant flaw in Lin et al.'s protocol, that validation required at least two pairing operations that could not be extended. In addition, their protocol uses many exponential operations that require complex computing. Therefore, they suggested an authentication protocol based on bilinear pairing and used addition operation, which is simpler than exponential operation. In 2013, Lee and Lai [12] found that Zhang et al.'s proposed scheme also has security weaknesses. They demonstrated that Zhang et al.'s protocol cannot achieve the signature non-repudiation and is insecure against replay attack. Moreover, Zhang et al.'s scheme cannot provide security to masquerade and tracking attacks. However, Jianhong et al. [13] proved that Lee and Lai's protocol is insecure to the impersonation and tracing attacks and violates the non-repudiation. Further, Bayat et al. [14] also found an impersonation attack in Lee and Lai's protocol. After that, Bayat et al. [14] proposed a secure authentication scheme for VANETS with batch verification to overcome [12]'s security weaknesses. Unfortunately, He et al. [15] pointed out that [14]'s protocol cannot defeat against modification, replay, and impersonation attacks. Then, He et al. [15] designed a novel secure protocol using Elliptic Curve Cryptographic (ECC) for vehicle communication. Zhong et al. [16] analyzed the protocol in [15] and concluded that using complex cryptographic functions can result in enormous operational costs and, consequently, the system faces network disruption problems. Therefore, they proposed a system to distribute pseudonymized signatures to verify user identities. In 2014, Chuang et al. [17] proposed a trust-extended authentication scheme in VANETs. Under their protocol, vehicles are divided into three types and they only used hash and exclusive-or functions to create lightweight communication. However, Zhou et al. [18] found out that Chuang et al.'s protocol cannot guarantee privacy-preserving and is vulnerable to impersonation and insider attacks. They also argued that the assumption of TPD is strong. Therefore, Zhou et al. proposed a more secure authentication protocol to improve Chuang et al.'s protocol. They use an ECC to protect entities' real identities and protect against internal attacks. In 2019, Wu et al. [19] pointed out that Zhou et al.'s proposed protocol cannot prevent identity guessing and impersonation attacks and also cannot guarantee user's anonymity. In 2017, Zhang et al. [1] proposed a personal information protection system based on distributed aggregation to conditionally block user's anonymity. However, this method takes more time to verify the signature, so the recipient must spend more time immediately verifying the correctness of the message. In 2019, Limbasiya et al. [4] proposed a secure message confirmation in vehicular cloud environment. They are pointed out that Zhong et al's protocol [16] has a security flaws using side channel attack over the OBU and TPD. Therefore, they suggested a more secure protocol for overcoming computational limitations of OBU and TPD through cloud computing. However, we revealed that their proposed protocol does not defeat several malicious attacks, such as session key disclosure attack and masquerade attack and so on. Additionally, their protocol does not provide privacy preserving and mutual authentication and has a correctness problem.

2.1.2. Ideal Tpd Limitation

In 2017, Zhang et al. [1] proposed a privacy-preserving authentication protocol for VANET communication with a realistic TPD in OBUs. They showed that the general TPD used in many

previous studies was not realistic. The ideal TPD has a strong assumption that an attacker cannot obtain or tamper with values stored in the OBU. However, Zhang et al. [1] demonstrated that attackers can perform side channel attack on TPDs in realistic situations to eventually control the entire VANET. In 2017, Zhang et al. [2] proposed a Chinese remainder theorem based authentication protocol for VANETs. They pointed out the heavy reliance on the ideal TPD. If a single TPD is obtained by a malicious user, reliance on the ideal TPD created a single point of failure and fail to preserve privacy of entire network. Therefore, they use biometrics of the drivers to help prevent attack over TPDs.

In 2018, Liu et al. [3] proposed an authentication scheme for VANETs to balance the reliance on the TPD. They demonstrated that strong reliance of the TPD provokes that attacker can compromise the whole system, because of key leakage, and they designed a protocol, such that, even if the TPD is compromised, the whole system will not be in danger.

2.2. Network Model

In the general architecture of vehicular networks, the communication of vehicles among the other vehicles or with the road side units (RSUs) is based on dedicated short-range communication [20], where the vehicle-to-Infrastructure (V2I) communication is the external network among the vehicles and RSUs.

Our proposed network model is based on Limbasiya et al.'s network model, but it addresses the problems regarding flaws in communication and authentication. Under their protocol, the process of transmitting the session key between RSUs and vehicles is unclear. Therefore, we propose a network model, in which vehicles and RSUs register at a trusted authority. The key agreement consists of all entities, including the vehicle, RSU, and trusted authority. Figure 1 illustrates our proposed network model and gives a detailed description of the entities.

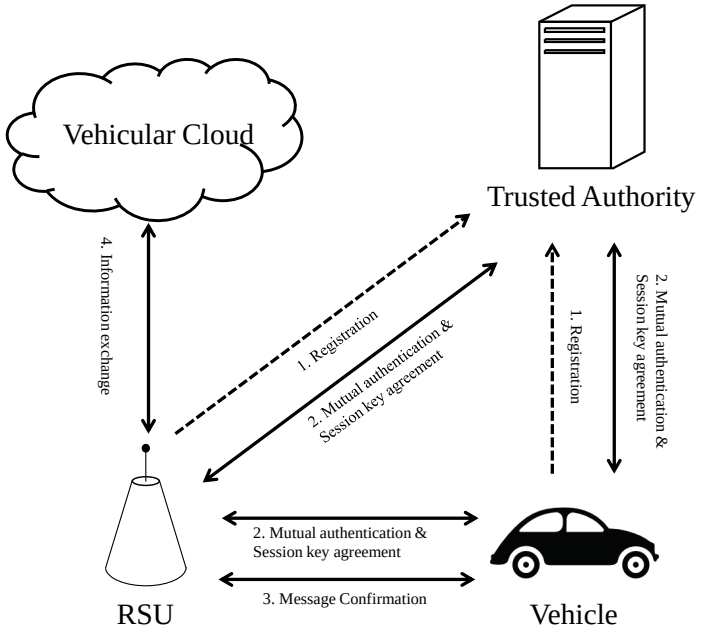

Figure 1. Proposed network model.

- Vehicle: vehicles have embedded devices, sensors and wireless communication device, such as velocity or location measurement equipment, Bluetooth, Wi-Fi, and OBU. In particular, the OBU collects information generated by sensors or devices. However, the OBU has relatively restricted memory. Therefore, the OBU sends the collected information to RSUs; subsequently, RSUs transmit the data to the vehicular cloud.
- RSU: RSUs are intermediary devices to transmit data between vehicles and the vehicular cloud. RSUs register with the trusted authority to generate a session key with vehicles. RSUs have more

memory and computing performance than OBUs. Therefore, RSUs can obtain data from many vehicles. However, RSUs cannot store data from multiple vehicles. Therefore, RSUs send specific data to the vehicular cloud.

- Trusted authority: a trusted authority is the top-level entity that an attacker can never attack. RSUs and vehicles should register with the trusted authority to generate the session key, and then, the trusted authority, RSUs, and vehicles perform mutual authentication.
- Vehicular cloud: a vehicular cloud is a storage server used to save a huge amount of data of different kinds within a VANET system. Each vehicle needs to collect and share the data with other vehicles. Therefore, the OBU collects data and communicates with other OBUs. However, OBUs have low computational performance and small storage space. Thus, vehicles send the data securely to RSUs and RSUs forward it to the vehicular cloud.

2.3. Threat Model

We cryptanalyze protocol security using the popular Dolev-Yao(DY) model [21]. By using this threat model, malicious attackers can capture, modify, add, or delete messages sent over insecure channels. And we also consider the following assumptions:

- A malicious adversary can steal or obtain a legitimate user's device, and perform side-channel attacks [22] to obtain key information stored in the device.
- A malicious adversary is able to masquerade as a legitimate user and trick authority entities for accessing resources.
- An adversary may obtain an authority entity's secret key. Subsequently, the adversary can compute a previous session key to trick user or authority entities.

We also follow the claims of [1–3]. Therefore, we assume that attackers can perform side channel attack or power analysis attack over TPDs or OBUs. Subsequently, attackers can obtain values stored in TPDs. Adversaries can perform a variety of attacks including impersonation, spoofing, identity guessing attacks using values obtained from compromised TPDs.

2.4. Notations

The used notations in this paper are given in Table 1.

Table 1. Notations.

Notations	Meanings
OBU	On board unit
TPD	Tamper-proof device
P	Elliptic curve generator
P_{pri_i}	A server private key
$s_i, a_i, b_i, r_i, r_j, a_j$	Selected random numbers
RID_i, ID_i	Registered vehicle identity
ID_{RSU_j}	Road-side unit identity
PW_{TPD_i}, PW_i	Registered vehicle password
v_i	Vehicle i in the network
RSU	Road-side unit
TA	Trusted authority
$h(\cdot)$	Hash function
$\|\|$	Connection symbol
$\oplus$	XOR operator

3. Review of Limbasiya et al.'s Protocol

We review Limbasiya et al.'s message confirmation scheme for VCC environment, which includes formation, key generation and message signature, and message confirmation phases.

3.1. Formation Phase

If a new vehicle requests registration with trusted authority TA, TA computes and sends OBU_i and TPD_i, which store the necessary values to the vehicle. Before registration, each vehicle computes parameters using unique identity RID_i, password PWD_{TPD_i}, and random number s_i. The detailed equations are shown in Figure 2 and steps are as follows.

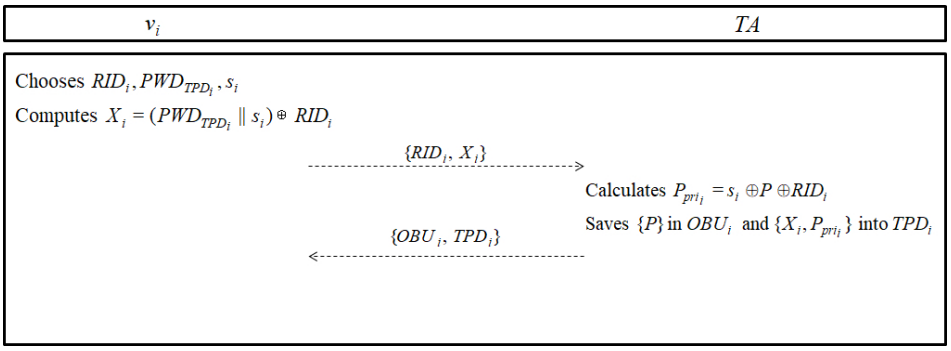

Figure 2. Formation phase of Limbasiya et al.'s protocol.

Step 1: Vehicle v_i chooses unique identity RID_i, password PWD_{TPD_i} and generates a random number s_i. v_i computes $X_i = (PWD_{TPD_i}||s_i) \oplus RID_i$, and then sends RID_i, X_i to TA through a secure channel.

Step 2: After receiving RID_i and X_i, TA calculates $P_{pri_i} = s_i \oplus P \oplus RID_i$ and saves $\{P\}$ in OBU_i and $\{X_i, P_{pri_i}\}$ in TPD_i. Subsequently, TA sends OBU_i and TPD_i to v_i via a secure channel.

3.2. Key Generation Phase

The vehicle v_i begins a key agreement process in TPD_i for message signature. v_i generates a session key SK_{RID_i} and transmits it to a concerned RSU. The detailed equations are illustrated in Figure 3 and the steps are as following.

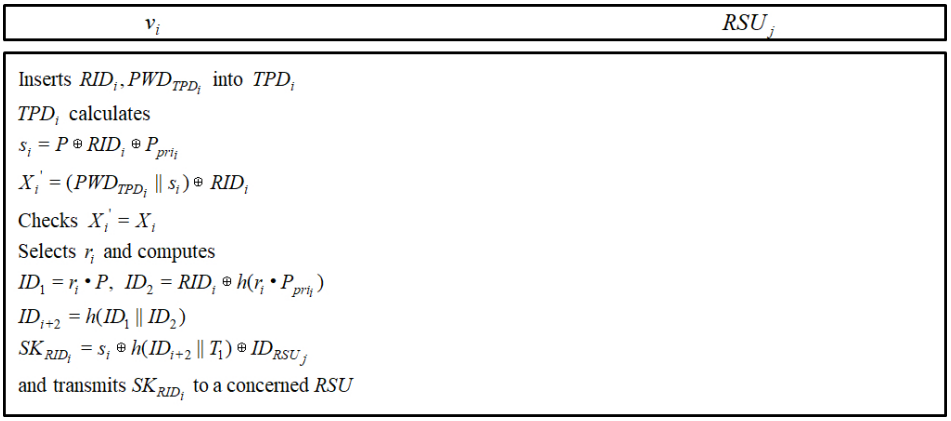

Figure 3. Key generation phase of Limbasiya et al.'s protocol.

Step 1: v_i inserts RID_i and PWD_{TPD_i} into TPD_i.

Step 2: then TPD_i computes $s_i = P \oplus RID_i \oplus P_{pri_i}$ and $X_i^{'} = (PWD_{TPD_i}||s_i) \oplus RID_i$. Then TPD_i compares $X_i^{'}$ with X_i stored in itself.

Step 3: if they are same, TPD_i selects random number r_i and computes $ID_1 = r_i \cdot P$, $ID_2 = RID_i \oplus h(r_i \cdot P_{pri_i})$ and $ID_{i+2} = h(ID_1||ID_2)$. Then TPD_i generates the session key $SK_{RID_i} = s_i \oplus h(ID_{i+2}||T_1) \oplus ID_{RSU_j}$ and transmits the session key to a concerned RSU.

3.3. Message Signature and Confirmation Phase of Limbasiya et al.'s Protocol

TPD_i signs the information with the session key and forwards to the connected RSU_j. Figure 4 shows the detailed equations with process steps, as follows.

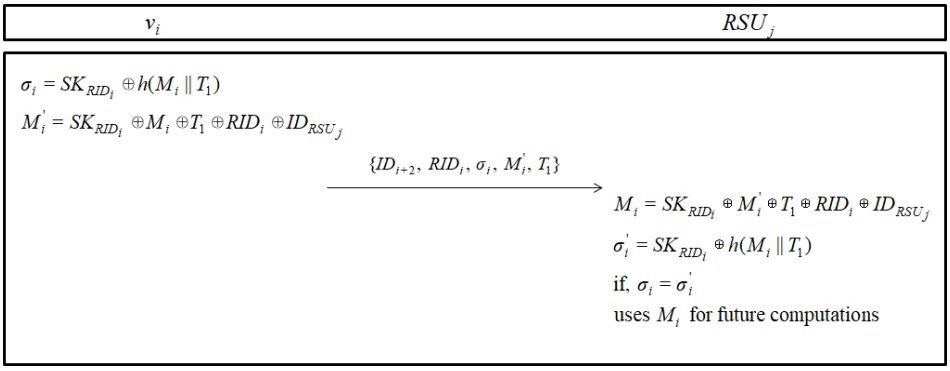

Figure 4. Message signature phase of Limbasiya et al.'s protocol.

Step 1: for signing the message, TPD_i computes $\sigma_i = SK_{RID_i} \oplus h(M_i||T_1)$ and $M_i^{'} = SK_{RID_i} \oplus M_i \oplus T_1 \oplus RID_i \oplus ID_{RSU_j}$. Subsequently, TPD_i sends message $\{ID_{i+2}, RID_i, \sigma_i, M_i^{'}, T_1\}$ to the concerned RSU_j.

Step 2: after receiving the message, RSU_j computes $M_i = SK_{RID_i} \oplus M_i^{'} \oplus T_1 \oplus RID_i \oplus ID_{RSU_j}$ and $\sigma_i^{'} = SK_{RID_i} \oplus h(M_i||T_1)$.

Step 3: then, RSU_j compares the σ_i with $\sigma_i^{'}$. If they are equal, RSU_j uses M_i for future computations. Additionally, Generally for batch verification, RSU_j inspects the exaction by a following equation:

$$(\sum_{i=1}^{n} v_i \cdot \sigma_i) = \sum_{i=1}^{n} v_i \cdot SK_{RID_i} \oplus \sum_{i=1}^{n} v_i \cdot h(M_i||T_1)$$

4. Cryptanalysis of Limbasiya et al.'s Protocol

Limasyia et al. demonstrated that their protocol provides privacy-preserving and mutual authentication and so on. However, in this section, we cryptanalyze Limbasiya et al.'s scheme for the VCC environment. Additionally, we figure out their protocol has several security flaws.

4.1. Correctness Problem

In the formation phase, a vehicle v_i sends only $\{RID_i, X_i\}$. Thus, TA cannot know information s_i. However, in Limbasiya et al.'s protocol, TA computes P_{pri_i} using s_i. Therefore, Limbasiya et al.'s protocol has a correctness problem and it may derive the incorrect formation of v_i.

4.2. Session Key Disclosure Attack

A malicious attacker $\mathcal{A}$ can perform the side channel attack on TPD [1–3] and OBU. Accordingly, $\mathcal{A}$ can obtain values stored in OBU and TPD, and also obtain transmitted messages through insecure channels. Thus, $\mathcal{A}$ can compute the session key using the obtained values.

Step 1: $\mathcal{A}$ can obtain P in OBU_i and X_i, P_{pri_i} in TPD_i using side channel attack. And $\mathcal{A}$ also can obtain the value RID_i through transmitted message. Subsequently, $\mathcal{A}$ can compute $s_i = P \oplus RID_i \oplus P_{pri_i}$.

Step 2: $\mathcal{A}$ can obtain ID_{i+2} and T_1 from transmitted messages and $\mathcal{A}$ obtains the value ID_{RSU_j}, which is public value. Therefore, $\mathcal{A}$ can compute $SK_{RID_i} = s_i \oplus h(ID_{i+2}||T_1) \oplus ID_{RSU_j}$.

Step 3: finally, $\mathcal{A}$ obtains the previous session key SK_{RID_i} and can trick other OBUs or RSUs.

4.3. Impersonation Attack

$\mathcal{A}$ can impersonate vehicles to compute message confirmation request messages. Section 4.2 shows that $\mathcal{A}$ can compute the session key. Therefore, $\mathcal{A}$ can compute confirmation request messages while using the computed session key and transmitted messages. The detailed steps are as follows.

Step 1: $\mathcal{A}$ can obtain $M_i^{'}$ through the transmitted message and compute previous session key as above session key disclosure attack Section. Subsequently, $\mathcal{A}$ can compute $M_i = SK_{RID_i} \oplus M_i^{'} \oplus T_1 \oplus RID_i \oplus ID_{RSU_j}$.
Step 2: $\mathcal{A}$ can also compute $\sigma_i = SK_{RID_i} \oplus h(M_i||T_1)$.
Step 3: finally, $\mathcal{A}$ can generate the confirmation request message $\{ID_{i+2}, RID_i, \sigma_i, M_i^{'}, T_1\}$ to impersonate the vehicle.

4.4. Privacy Preserving Problem

In Limbasiya et al.'s scheme, the legitimate identity of the vehicle RID_i is transmitted through public channels. This may cause the tracing attack and cannot preserve the user's privacy. As above sections, the attacker can masquerade legitimate vehicles and make a session key to access sensitive information. Therefore, the protocol of Limbasiya et al. is not able to provide privacy-preserving.

4.5. Mutual Authentication

In above section, we prove that $\mathcal{A}$ can generate the session key SK successfully, and impersonate the legitimate vehicle. Therefore, the protocol of Limbasiya et al. cannot achieve key agreement and mutual authentication.

5. Secure Key Agreement and Authentication Protocol for VCC

This section provides the proposed protocol to resolve the security flaws in Limbasiya et al.'s protocol. We use only an OBU instead of a TPD. Limbasiya et al.'s protocol cannot provide secure key agreement, because the TPD sends the session key without encryption. Therefore, we register vehicles and RSUs at the TA to generate secure key agreement. Thereafter, the vehicle transmits the information encrypted with the session key to the RSU. RSUs validate the message and send it to the vehicular cloud. We also consider performance and storage of OBU because of its relatively low computational power and small storage. Thus, we design the protocol using only exclusive-or and one-way hash function, which have low computational cost.

5.1. Registration Phase

For message confirmation with VCC and communicating with other vehicles or RSUs, the vehicle must register with the TA. Additionally, RSUs also register through TA to make secure session key with the vehicle. The detailed steps are as following and shown in Figure 5.

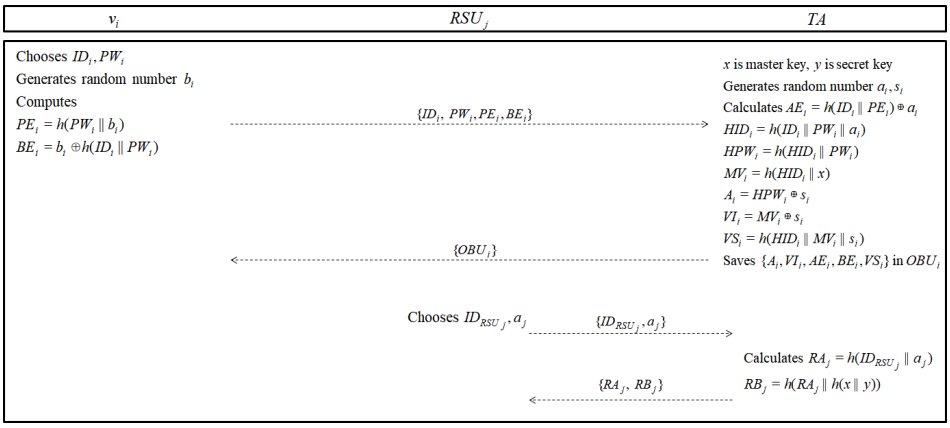

Figure 5. Registration phase of our proposed protocol.

Step 1: vehicle v_i chooses identity ID_i, password PW_i and random number b_i. And vehicle computes $PE_i = h(PW_i||b_i)$ and $BE_i = b_i \oplus h(ID_i||PW_i)$. v_i sends the message $\{ID_i, PW_i, PE_i, BE_i\}$ to TA.

Step 2: TA has master key x and secret key y. After receiving the registration request message from v_i, TA generates random numbers a_i and s_i for the vehicle. Subsequently, TA calculates $AE_i = h(ID_i||PE_i) \oplus a_i$, $HID_i = h(ID_i||PW_i||a_i)$, $HPW_i = h(HID_i||PW_i)$, $MV_i = h(HID_i||h(x||y))$, $A_i = HPW_i \oplus s_i$, $V_i = MV_i \oplus s_i$ and $VS_i = h(HID_i||MV_i||s_i)$. Afterwards, TA saves A_i, VI_i, AE_i, BE_i and VS_i in the OBU_i, and then sends OBU_i to the vehicle through a closed channel.

Step 3: road side unit RSU_j chooses ID_{RSU_j} and random nonce a_j and sends these values to TA via a closed channel.

Step 4: when TA receives values from RSU_j, TA calculates $RA_j = h(ID_{RSU_j}||a_j)$ and $RB_j = h(RA_j||h(x||y))$. Subsequently, TA sends the message $\{RA_j, RB_j\}$ to RSU_j via a secure channel.

5.2. Key Agreement and Authentication Phase

The vehicle and RSU must have key agreement through generating the session key for secure communication among the RSU and other OBUs. Vehicle and RSU are authenticated by TA. If the TA checks that vehicle and RSUs are legitimate entities, vehicle and RSU generate a session key. The detailed steps are given below. See Figure 6.

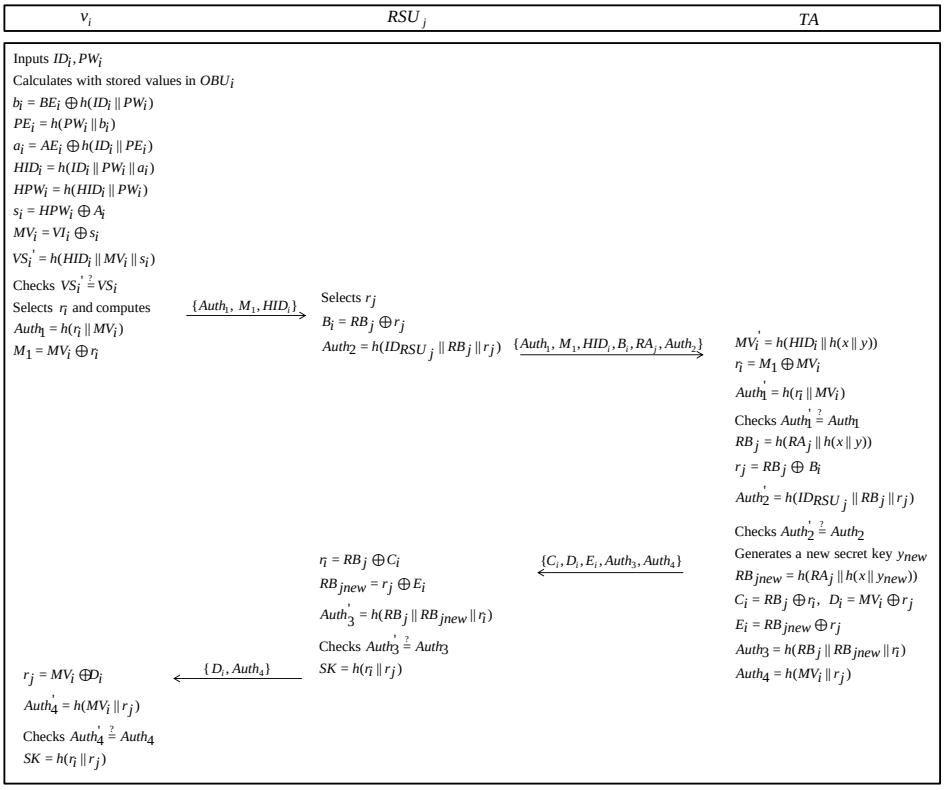

Figure 6. Key agreement and authentication phase of our proposed protocol.

Step 1: vehicle v_i inputs ID_i and PW_i. Subsequently, v_i extracts $b_i = BE_i \oplus h(ID_i||PW_i)$ with stored values BE_i in the OBU_i. v_i calculates $PE = h(PW_i||b_i)$, $a_i = AE_i \oplus h(ID_i||PE_i)$, $HID_i = h(ID_i||PW_i||a_i)$, $HPW_i = h(HID_i||PW_i)$, $s_i = HPW_i \oplus A_i$, and $MV_i = VI_i \oplus s_i$ and $VS_i^{'} = h(HID_i||MV_i||s_i)$. Then, v_i checks whether $VS_i^{'} \stackrel{?}{=} VS_i$. If valid, v_i selects a random number r_i and computes $Auth_1 = h(r_i||MV_i)$ and $M_1 = MV_i \oplus r_i$. Finally, v_i sends the message $\{Auth_1, M_1, HID_i\}$ to the concerned RSU_j via an insecure channel.

Step 2: RSU_j selects r_j, and computes $B_i = RB_j \oplus r_j$ and $Auth_2 = h(ID_{RSU_j}||RB_j||r_j)$. Then, RSU_j sends the values $\{Auth_1, M_1, HID_i, B_i, RA_j, Auht_2\}$ to the TA via an insecure channel.

Step 3: when TA receives the message from RSU_j, TA computes $MV_i^{'} = h(HID_i||h(x||y))$, $r_i = M_1 \oplus MV_i$ and $Auth_i^{'} = h(r_i||MV_i)$. Then, TA compares $Auth_1^{'}$ and $Auth_1$. If they are equal, TA extracts the values $RB_j = h(RA_j||h(x||y))$ and $r_j = RB_j \oplus B_i$. TA computes $Auth_2^{'} = h(ID_{RSU_j}||RB_j||r_j)$ and compares it with $Auth_2$. If they are same, TA generates a new secret key y_{new}. TA computes $RB_{jnew} = h(RA_j||h(x||y_{new})$, $C_i = RB_j \oplus r_i$, $D_i = MV_i \oplus r_j$, $E_i = RB_{jnew} \oplus r_j$, $Auth_3 = h(RB_j||r_i)$ and $Auth_4 = h(MV_i||r_j)$. Finally, TA sends the message $\{C_i, D_i, E_i, Auth_3, Auth_4\}$ to RSU_j through an open channel.

Step 4: after receiving the values from TA, RSU_j extracts $r_i = RB_j \oplus C_i$, $RB_{jnew} = r_j \oplus E_i$ and computes $Auth_3^{'} = h(RB_j||RB_{jnew}||r_i)$. Then RSU_j checks whether $Auth_3^{'}$ and $Auth_3$ are equal or not. If they are equal, RSU_j updates RB_j to RB_{jnew} and generates the session key $SK = h(r_i||r_j)$. RSU_j sends the message $\{D_i, Auth_4\}$ to v_i via a public channel.

Step 5: v_i extracts the value $r_j = MV_i \oplus D_i$, computes $Auth_4^{'} = h(MV_i||r_j)$ and checks whether $Auth_4^{'}$ and $Auth_4$ are same or not. If they are equal, v_i computes the session key $SK = h(r_i||r_j)$. Finally, v_i and concerned RSU_j have the same session key.

5.3. Message Signature and Message Confirmation Phase

If the v_i wants to send information to the concerned RSU, v_i must sign the message using the session key and sends it to the RSU_j. Additionally, RSU_j checks whether the message is legitimate or not. If the message is legitimate, RSU_j validates the message and sends it to a cloud server. The detailed steps are as following and are shown in Figure 7.

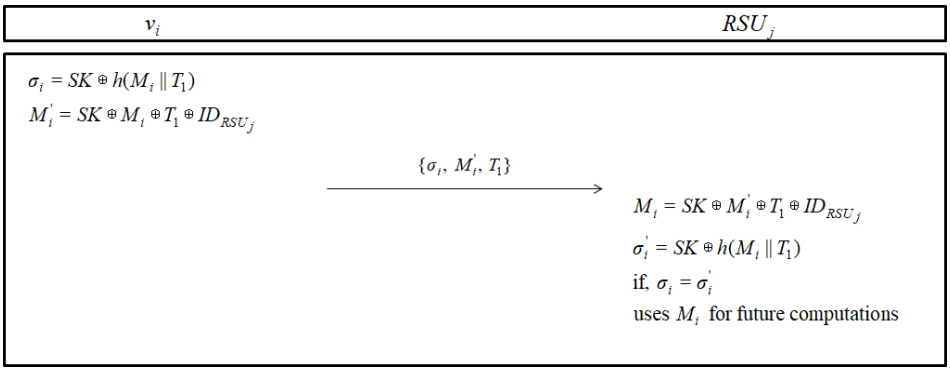

Figure 7. Message signature and confirmation phase of our proposed protocol.

Step 1: for signing the information M_i, v_i computes $\sigma_i = SK \oplus h(M_i||T_1)$ and $M_i^{'} = SK \oplus M_i \oplus T_1 \oplus ID_{RSU_j}$ and sends the message $\{\sigma_i, M_i^{'}, T_1\}$ to the concerned RSU_j.

Step 2: after receiving the message, RSU_j extracts information $M_i = SK \oplus M_i^{'} \oplus T_1 \oplus ID_{RSU_j}$, computes $\sigma_i^{'} = SK \oplus h(M_i||T_1)$ and checks whether σ_i and $\sigma_i^{'}$ are equal or not. If they are the same, RSU_j uses the information M_i for the future computations. Additionally, generally for batch verification, RSU_j inspects the exaction by a following equation:

$$(\sum_{i=1}^{n} v_i \cdot \sigma_i) = \sum_{i=1}^{n} v_i \cdot SK \oplus \sum_{i=1}^{n} v_i \cdot h(M_i||T_1)$$

6. Security Analysis

We simulate with the AVISPA simulation tool [23,24] in order to demonstrate that the proposed protocol is able to prevent against replay and man-in-the-middle attacks. We also prove the session key security using the ROR model [25] and conduct the informal security analysis. Therefore, our proposed

protocol can provide security against various attacks including impersonation, side channel attack over TPD, trace attack, and so on.

6.1. ROR Model

In this section, we use the universally-accepted real-or-random (ROR) model [25] in order to prove the security of the session key in our proposed protocol.We provide the similar proof as adopted in [26,27].

Short Discussion about ROR Model

In the ROR model [25], the malicious attacker $\mathcal{A}$ is modeled using the DY model, which interacts with the instance of the participants in the protocol. In our proposed protocol, v_i, RSU_j and TA are considered as participants. Additionally, $P_{v_i}^{t^1}$, $P_{RSU_j}^{t^2}$, and $P_{TA}^{t^3}$, which are called *oracles* denoting the instances t^1, t^2, and t^3 of v_i, RSU_j, and TA, respectively. Table 2 shows various queries that simulate attacks, such as eavesdropping, modifying, and deleting or inserting the transmitted messages among the entities. $h(\cdot)$ and Collision-resistant one-way hash function $Hash$ are modeled as a random oracle and they can be used by all participants including $\mathcal{A}$.

Wang et al. [28] showed that the password chosen by the user follows the Zipf's law, which is quite different from the uniform distribution. They also found that the size of password dictionary is quite limited in the sense that users do not generally use the entire space of the passwords; instead, they use a small space of the allowed characters space. We apply the Zipf's law in order to prove the session key security of our proposed protocol.

Theorem 1. *If Adv_P is the advantage function of an attacker $\mathcal{A}$ in breaking the session key SK security of the proposed protocol P, respectively, q_h, q_{send}, and $|Hash|$ are the number of Hash queries, Send queries, and the range space of the hash function, respectively. Subsequently,*

$$Adv_P \leq \frac{q_h^2}{|Hash|} + 2max\{C^{'} \cdot q_{send}^{s^{'}}\}$$

where $C^{'}$ and $s^{'}$ are the Zipf's parameters [28].

Table 2. Various queries and their meanings.

Query	Meaning
$Execute(P_{v_i}^{t^1}, P_{RSU_j}^{t^2}, P_{TA}^{t^3})$	This query means that the model of the eavesdropping attack between the entities v_i, RSU_j and TA via an insecure channels.
$CorruptOBU(P_{v_i}^{t^1})$	Under this corrupt on-board-unit (OBU) query, $\mathcal{A}$ can fetch all sensitive credentials stored in the OBU of v_i. This is modeled as an active attack.
$Send(P^t)$	Under this query, $\mathcal{A}$ can transmits a message to P^t, and in response, it also receives a message from P^t. This is also modeled as an active attack.
$Reveal(P^t)$	The query means that $\mathcal{A}$ reveals session key SK created by P^t and its partner to $\mathcal{A}$ in the current session.
$Test(P^t)$	Before the game begins, under this query, an unbiased coin c is flipped. Depending on the output, the following decisions are made. $\mathcal{A}$ executes this query and if the session key SK among v_i and RSU_j is fresh, P^t returns SK if $c = 1$ or a random nonce if $c = 0$; otherwise, it returns a null value($\perp$).

Proof. We define four games, called game GM_i, $i \in [0, 1, 2, 3]$. The probability associated with GM_i in which $\mathcal{A}$ can guess the random bit c and wins the game and denoted by $Succ_i$. Moreover, $Pr[.]$ denotes the probability. We discuss the details for these four defined games below.

- **Game** GM_0: in this game, $\mathcal{A}$ chooses a random bit c. Additionally, this game involves a practical attack executed by $\mathcal{A}$ against the protocol in the ROR model. Because GM_0 and protocol are identical, we get,

$$Adv_P = |2 \cdot Pr[Succ_0] - 1|. \tag{1}$$

- **Game** GM_1: under this game, $\mathcal{A}$ performs the eavesdropping attack to all transmitted messages during key generation and message confirmation process of the proposed protocol using the *Execute* query. At the end of the this game, $\mathcal{A}$ makes *Reveal* and *Test* queries. The output of the *Reveal* and *Test* queries decide if $\mathcal{A}$ obtains the derived session key SK between v_i and RSU_j or a random number. In our proposed protocol, v_i and RSU_j computes the session key as $SK = h(r_i||r_j)$. To derive SK, $\mathcal{A}$ needs the short-term (temporal) secrets (r_i and r_j), which are unknown to $\mathcal{A}$. However, the transmitted messages are not helpful to increase winning probability. As both the game GM_0 and GM_1 are indistinguishable, we can get

$$Pr[Succ_1] = Pr[Succ_0]. \tag{2}$$

- **Game** GM_2: this game is modeled as an active attack which includes the simulation of *Hash* and *Send* queries. In proposed protocol, all of the messages are protected by the collision-resistant one-way hash function except M_1, B_j, C_i and D_i. However, random numbers are used in values M_1, B_j, C_i and D_i. Furthermore, deriving r_i from the intercepted $Auth_1$, C_i, and M_1, and also r_j from intercepted B_i, $Auth_2$, D_i, and $Auth_4$ are computationally infeasible task because of collision-resistant property of the hash function. Therefore, no collision occurs when $\mathcal{A}$ executes *Hash* query. Using the birthday paradox results, we can have,

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2|Hash|}. \tag{3}$$

- **Game** GM_3: this is the final game that executes the *CorruptOBU* query by $\mathcal{A}$. $\mathcal{A}$ can extract all the information $\{A_i, V_i, AE_i, BE_i, VS_i\}$ from the OBU of v_i. Note that $HPW_i = h(HID_i||PW_i)$, $AE_i = h(ID_i||PE_i) \oplus a_i$, $PE_i = h(PW_i||b_i)$, $BE_i = b_i \oplus h(ID_i||PW_i)$, and $VS_i = h(HID_i||MV_i||s_i)$. To derive the secrets s_i, a_i, and b_i from A_i, VI_i, BE_i, and AE_i, $\mathcal{A}$ needs unknown ID_i and PW_i. Without having secret credentials b_i, ID_i, and PW_i of v_i, it is a computationally difficult problem for $\mathcal{A}$ to guess password PW_i of v_i correctly using the *Send* queries. Because GM_2 and GM_3 are identical when password guessing attack is absent. Therefore, using the Zipf's law on passwords, we obtain

$$|Pr[Succ_3] - Pr[Succ_2]| \leq C^{'} \cdot q_{send}^{s^{'}}. \tag{4}$$

All of the games are executed; therefore, $\mathcal{A}$ needs to guess the correct bit c. Therefore, we have

$$Pr[Succ_3] = \frac{1}{2}. \tag{5}$$

Equations (1) and (2) give the following result:

$$\begin{aligned} \frac{1}{2}Adv_P &= |Pr[Succ_0] - \frac{1}{2}| \\ &= |Pr[Succ_1] - \frac{1}{2}|. \end{aligned} \tag{6}$$

Again, Equations (5) and (6) give the following result:

$$\frac{1}{2}Adv_P = |Pr[Succ_1] - Pr[Succ_3]|. \tag{7}$$

We obtain the following equation using the triangular inequality and Equations (3) and (4):

$$\begin{aligned}\frac{1}{2}Adv_P &= |Pr[Succ_1] - Pr[Succ_3]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| + |Pr[Succ_2] - Pr[Succ_3]| \\ &\leq \frac{q_h^2}{2|Hash|} + max\{C^{'} \cdot q_{send}^{s^{'}}\}. \end{aligned} \tag{8}$$

At last, we obtain the required result by multiplying both sides of Equation (8) by a factor of 2:

$$Adv_P \leq \frac{q_h^2}{|Hash|} + 2max\{C^{'} \cdot q_{send}^{s^{'}}\}.$$

Therefore, the Theorem 1 is proved. □

6.2. Formal Security Analysis through AVISPA

We perform a formal security analysis of the proposed protocol using the AVISPA validation tool in order to demonstrate that the protocol can resist replay and man-in-the-middle attacks. The AVISPA adopts the High-Level Protocol Specification Language (HLPSL) code. We briefly discuss AVISPA and present HLPSL codes of our protocol. After that, we present the simulation results of the AVISPA to show that our protocol can protect against man-in-the-middle and replay attacks. Numerous studies verified with the AVISPA tool have been presented [29–31].

6.2.1. Proposed Protocol's HLPSL Code

The AVISPA uses the four back-ends, such as On-the-fly Model-Checker (OFMC) [32], CL-based Attack Searcher (CL-AtSe) [33], SAT-based Model-Checker (SATMC), and Tree Automate-based Protocol Analyser (TA4SP) in order to verify security of a protocol. The code is translated into intermediate format (IF), and IF uses four back-ends to convert to output format (OF). Especially, OFMC and CL-AtSe are commonly used for verification.

The proposed protocol has three basic *roles* which denote entities: VI denotes a vehicle, RSU denotes a roadside unit and TA denotes a trusted authority. Roles of *session* and *environments* are illustrated in Figure 8. In *session* and *environments*, we set up the intruder knowledge, five authentication goals and four secrecy goals. We briefly discuss HLPSL code for role VI shown in Figure 9.

At transition 1, VI begins registration phase at 0 state value with start message, and VI updates the state to 1. VI sends message $\{ID_i, PW_i, PE_i, BE_i\}$ to TA through closed channels and declares the function $secret(\{IDi, PWi, Bi'\}, sp1, \{VI\})$, which means that $sp1$ denotes values $\{IDi, PWi, Bi'\}$ which are only known to VI. At transition 2, VI receives the OBU_i from TA and updates the state to 2. At the state 2, VA generates a random number r_i, sends the message $\{Auth_1, M_1, HID_i\}$ to the RSU_j through an open channel, and declares function $witness(VI, TA, vi\_ta\_ri, Ri')$, which means that vi_ta_ri denotes a weakness authentication factor is used by VI to authenticate TA. At transition 3, VI receives the message from RSU. After that VI generates the session key SK, performs message confirmation and declares $witness(VI, RSU, vi\_rsu\_sig, SK')$ and $request(VI, TA, ta\_vi\_auth4, Rj')$. The function $request(VI, TA, ta\_vi\_auth4, Rj')$ means that ta_vi_auth4 represents a strong authentication factor. The codes of RSU and TA are similar to the code of VI.

```
role session(EV, OP, EAG: agent, SKevop:
symmetric_key, H: hash_func)

def=
local SN1, SN2, SN3, RV1, RV2, RV3: channel(dy)
composition
vehicle(EV, EAG, OP, SKevop,H, SN1, RV1)
/\ operator(EV, EAG, OP, SKevop,H,  SN2, RV2)
/\ agg(EV, EAG, OP, H, SN3, RV3)
end role

role environment()
def=
const ev, eag, op : agent,
skevop: symmetric_key,
h, mul, add: hash_func,
idev, ideag: text,
sp1, sp2, sp3, sp4: protocol_id,
ev_eag_m2, eag_ev_m4: protocol_id

intruder_knowledge = {ev,eag,op,h,mul,add,idev,ideag}
composition
session(ev,eag,op,skevop,h)/\session(i,eag,op,skevop, h)
/\session(ev,i,op,skevop,h)
/\session(ev,eag,i,skevop,h)

end role

goal
secrecy_of sp1, sp2, sp3, sp4
authentication_on ev_eag_m2, eag_ev_m4
end goal

environment()
```

Figure 8. Code of session and environments.

```
role vehicle(EV, OP, EAG : agent, SKevop : symmetric_key, H:
hash_func, SND, RCV : channel(dy))

played_by EV
def=
local State: nat,
   MUL, ADD : hash_func,
   HIDi,IDi, PWi, A1, Rev, PKi, G, HID, Kop, PKop, Ai, Bi, Ci, Di,
Ei : text,
   IDeag, PKeag, M2, M3, M4, T1, T2, B1, SK : text
const sp1, sp2, sp3, sp4, ev_eag_m2, eag_ev_m4 : protocol_id
init State := 0
transition
1. State = 0 /\ RCV(start) =|>
State' := 2 /\ A1' := new() /\ Rev' := new()
      /\ PKi' := MUL(Rev'.G)
      /\ HIDi' := H(IDi.A1')
      /\ SND({H(IDi.A1').A1'}_SKevop)
         /\ secret({IDi, PWi, Rev'}, sp1, {EV})
/\ secret({A1'},sp2,{EV, OP})

2. State = 2 /\ RCV
({xor(H(H(IDi.A1').A1'),Kop').H(H(IDi.A1').A1'.Kop')}_SKevop)=
|>
 State' := 4 /\ Rev' := new() /\ Di' := xor(H(IDi.PWi),A1')
       /\ Ei' := xor(H(A1'.IDi.PWi),Rev')
       /\ T1' := new()
       /\ M2' := H(A1'.H(IDi.A1').T1')
       /\ SND(MUL(Rev'.G).ADD((A1'.H(IDi.A1').T1').MUL(Rev'.P
Keag)))
          /\ witness(EV,EAG, ev_eag_m2, A1')

3. State = 4 /\ RCV(xor(B1',A1').H(IDeag.A1'.B1'.T2')) =|>
State' := 6 /\ SK' := H(H(IDi.A1').IDeag.A1'.B1')
      /\ request(EAG, EV, eag_ev_m4, B1')
end role
```

Figure 9. Code of vehicle.

6.2.2. Results of Verification

The verification results using models OFMC and CL-AtSe are shown in Figure 10. Two simulations are able to check whether the protocol withstands man-in-the-middle and replay attacks. The CL-AtSe verification shows that three states are analyzed and translated to 0.11 s. The results of OFMC shows that it visits 1040 nodes with a search time of 9.57 s and 9 plies depth. The summary part of CL-AtSe and OFMC indicates SAFE, so we can say that the proposed protocol resists replay and man-in-the-middle attacks.

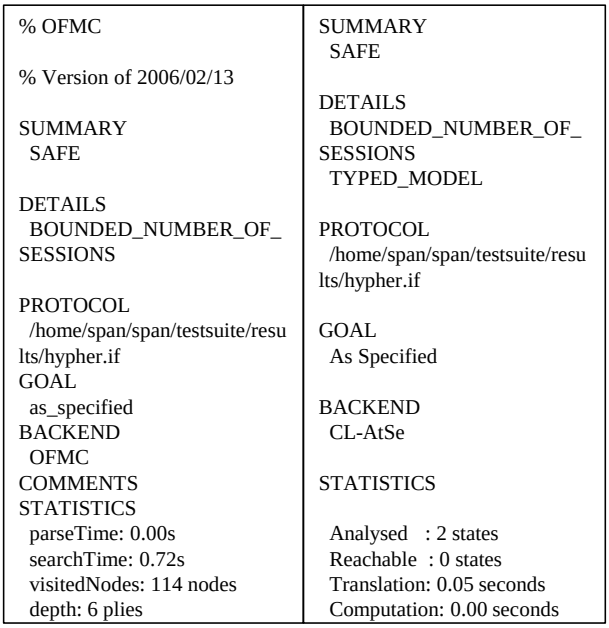

Figure 10. Result of Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation using On-the-fly Model-Checker (OFMC) and CL-based Attack Searcher (CL-AtSe) models.

6.3. Informal Analysis

In this section, we analyze informal security verification in order to prove that the proposed protocol can resist numerous attacks, such as OBU stolen, impersonation, session key disclosure, off-line guessing attacks, and so on. Moreover, we show that the proposed protocol can achieve privacy-preserving and mutual authentication.

6.3.1. Vehicle Impersonation Attack

If an adversary $\mathcal{A}$ attempts to impersonate a vehicle v_i, $\mathcal{A}$ should generate message $\{Auth_1, M_1, HID_i\}$ and $\{\sigma_i, M_i^{'}, T_1\}$. However, $\mathcal{A}$ cannot extract a_i, r_i and MV_i even if $\mathcal{A}$ extracts the value stored in the OBU. Because a_i, r_i and MV_i are masked with random numbers b_i, s_i, and session key SK. Therefore, the proposed protocol resists impersonation attacks, because $\mathcal{A}$ cannot generate the correct messages.

6.3.2. Side Channel Attack over OBU

We assume that $\mathcal{A}$ can extract values from the OBU based on our assumed threat model. Therefore, $\mathcal{A}$ can perform side channel attack over OBU and extract $\{A_i, VI_i, AE_i, BE_i, VS_i\}$. However, $\mathcal{A}$ cannot obtain any useful information without identity, password, and secret random numbers, because all of the values stored in OBU are masked with one-way hash function or XOR operation on a_i, b_i, and s_i. Thus, $\mathcal{A}$ does not have any advantage of side channel attack over OBU.

6.3.3. Off-Line Guessing Attack

$\mathcal{A}$ cannot guess the identity or password, because $b_i = BE_i \oplus h(ID_i||PW_i)$, $a_i = AE_i \oplus h(ID_i||PE_i)$ and $PE_i = h(PW_i||b_i)$ are masked with random numbers and the secret values ID_i and PW_i. $\mathcal{A}$ must also check whether VS_i and calculate $VS_i^{'}$ by $\mathcal{A}$ to see whether the identity and password are guessed correctly. For this, $\mathcal{A}$ can perform a side channel attack over OBU to obtain the stored values $\{A_i, VI_i, AE_i, BE_i, VS_i\}$. However, to calculate $VS_i^{'}$, $\mathcal{A}$ needs to know the secret random number s_i and secret parameter MV_i. This allows for being computationally expensive to guess identity or password. Therefore, we show that the proposed protocol can prevent off-line guessing attacks.

6.3.4. Man-in-the Middle Attack and Replay Attack

The adversary $\mathcal{A}$ can obtain the transmitted messages over an open channel and stored parameters in the OBU according to the threat model. However, we show that $\mathcal{A}$ cannot generate valid vehicle's messages as mentioned above. Furthermore, $\mathcal{A}$ also cannot generate the RSU_j's message, because $\mathcal{A}$ does not know secret random numbers r_i, r_j and secret parameter RB_j. Thus, $\mathcal{A}$ cannot impersonate v_i or RSU_j by replaying intercepted messages as all messages are dynamic with random numbers r_i and r_j. Therefore, the proposed protocol prevents man-in-the-middle and replay attacks.

6.3.5. Session Key Disclosure Attack

Even if $\mathcal{A}$ has obtained values, as mentioned above, $\mathcal{A}$ cannot generate the session key SK. The SK comprises the hash function with secret random numbers r_i and r_j. However, $\mathcal{A}$ cannot extract random numbers, because they are masked with secret parameters MV_i and RB_j. Moreover, MV_i and RB_j are also masked with random numbers. Therefore, $\mathcal{A}$ does not know about the session key SK.

6.3.6. Trace Attack and Privacy-Preserving

The vehicle v_i does not send its real identity ID_i over an open channel. The vehicle generates the pseudonym identity $HID_i = h(ID_i||PW_i||a_i)$. And also RSU_j uses the RA_j instead of real identity ID_{RSU_j}. Moreover, as above mentioned Sections, $\mathcal{A}$ cannot impersonate legitimate vehicles and also cannot generate a validated session key. Therefore, the proposed protocol provides the privacy-preserving. v_i and RSU_j communicate the information using the session key without pseudonym identities. Thus, we can say that the proposed protocol can prevent trace attack.

6.3.7. Mutual Authentication

After receiving message the $\{Auth_1, M_1, HID_i, B_i, RA_j, Auth_2\}$ from v_i and RSU_j, TA checks whether $Auth_1^{'} \stackrel{?}{=} Auth_1$. If it is equal, TA also checks $Auth_2^{'} \stackrel{?}{=} Auth_2$. Subsequently, TA sends $\{C_i, D_i, Auth_3, Auth_4\}$ to the v_i and RSU_j for authenticating. RSU_j checks $Auth_3^{'} \stackrel{?}{=} Auth_3$ and v_i checks also $Auth_4^{'} \stackrel{?}{=} Auth_4$. If they are valid, v_i, RSU_j, and TA successfully authenticate each other. Previous sections have shown that $\mathcal{A}$ cannot generate valid messages. Furthermore, all of the transmitted messages are refreshed for every session with secret random numbers. Therefore, our proposed protocol successfully ensures secure mutual authentication and achieves session key agreement.

7. Performance Analysis

In this section, we compare our proposed protocol with other related protocols for VANETs. We consider computation, communication costs, and security features.

7.1. Computation Cost

We show the comparison outcomes in Table 3. Our proposed protocol is lightweight as compared to other related protocols. Therefore, we can demonstrate that the proposed protocol is suitable for vehicular cloud environment in VANETs.

Table 3. Computation cost of key generation and message confirmation phase.

Protocols	Computational Complexity	Total Cost
Jianhong et al. [13]	$T_{bpsm} + 3T_{bp} + T_{MPH}$	18.748 ms
Zhong et al. [16]	$5T_{sem}$+$3T_h$+Tea	0.0711 ms
Limbasiya et al. [4]	$4T_h$+$2T_{sem}$	0.0280 ms
Ours	$22T_h$	0.0022 ms

XOR operation is negligible as compared to other operations.

For comparing the computational cost, we define following notations. T_{bp}, T_{bpsm}, T_{MPH}, T_h, T_{sem} and T_{ea}, which denotes the execution time of bilinear mapping, multiplication related to bilinear pairing, map-To-point hash, one-way hash, small scale multiplication related to elliptic curve cryptography (ECC), and addition related to ECC. We focus on time overhead in the process of authentication message generation and message verification. For rough estimation, we consider the existing results reported by [34]. The execution time of each operation is as following.

- T_{bp}: Time for bilinear pairing operation (≈4.2110 ms)
- T_{bpsm}: Time for small scale multiplication related to bilinear pairing (≈1.7090 ms)
- T_{MPH}: Time for map-To-point hash operation (≈4.406 ms)
- T_h: Time for one-way hash operation (≈0.0001 ms)
- T_{sem}: Time for small scale multiplication related to ECC (≈0.0138 ms)
- T_{ea}: Time for point addition related to ECC (≈0.0018 ms)

7.2. Communication Cost and Storage Cost

We compare communication cost overheads among related protocols and proposed protocol during the message confirmation phase in Table 4. We assume that the identity, password, and normal variable needs eight bytes, the time-stamp needs four bytes, an ECC encryption/decryption needs 32 bytes, a bilinear pairing needs 128 bytes, and one-way hash function needs 32 bytes [4]. As the results of the comparison, the proposed protocol is the most efficient when compared with other related protocols. The storage overhead is calculated based on the total number of bytes required to store required parameters in OBU or TPD and RSU. The proposed protocol has 224 bytes storage cost, where OBU has 160 bytes and RSu has 64 bytes. Although the total memory of our protocol is slightly higher than that of other protocols, our protocol ensures security.

Table 4. Communication cost and storage cost.

Protocols	Communication Cost	Storage Cost	Total Memory
Jianhong et al. [13]	132 bytes	528 bytes	660 bytes
Zhong et al. [16]	100 bytes	136 bytes	236 bytes
Limbasiya et al. [4]	124 bytes	32 bytes	156 bytes
Ours	100 bytes	224 bytes	324 bytes

7.3. Energy Consumption

Researchers need to consider the size and speed of the message being sent to the recipient. This is because data transmission occurs under Dedicated Short-Range Communication (DSRC) and, in the case of vehicle networks defined in IEEE 802.11p, it belongs to the physical protocol layer. This IEEE standard operates at 10 MHz channel bandwidth, 5.8 GHz frequency, 25 dBm transmit power, and 6 Mbps data rate [35]. The energy consumption for the verification scheme can be calculated as E_{et} (for the execution time of key generation and message confirmation) E_{co} (for the communication cost for message confirmation) and it is measured in millijoule (mJ). For the execution time, $E_{et} = T_c * C$, where T_c = Total computation cost, C = cpu maximum power, which is 10.88 W

for wireless communication networks [36]. $E_{et} = (D_m * C)/(D_r)$, where D_m = the size of message, D_r = the data rate for vehicular communications (6000 Kbps). By referring to Table 5, we can say that the proposed protocol consumes the least energy.

Table 5. Energy consumption.

Protocols	Execution Energy Consumption	Communication Energy Consumption
Jianhong et al. [13]	203.978 mJ	0.239 mJ
Zhong et al. [16]	0.774 mJ	0.181 mJ
Limbasiya et al. [4]	0.305 mJ	0.225 mJ
Ours	0.024 mJ	0.181 mJ

7.4. Propagation Delay

The propagation delay ($d_p = T_2 - T_1$) is determined by computing the difference between the timestamps of a message received (T_2) and transmitted (T_1). But d_p expects some time interval, which can be stated as in $d_{p(V2V)} = \frac{L*h}{f}$ and $d_{p(V2I)} = \frac{L}{f_{RSU}}$ for L length messages (i.e., communication cost) at f transmitted data rate along with h hops through which a message is traveled [37]. Thus, the propagation delay of our protocol is the lowest, because the communication cost of the proposed protocol is the lowest.

7.5. Security Properties

In Table 6, we present the results of protocols related to security comparisons and our proposed protocol based on batch verification. The suggested protocol prevents more attacks than other related previous studies, and also provide privacy-preserving and mutual authentication. Therefore, our proposed protocol is significantly safer than the considered related protocols. The system consumes some energy during implementation, depending on the real time and communication overhead of the system.

Table 6. Security Properties.

Security Properties	Jianhong et al. [13]	Zhong et al. [16]	Limbasiya et al. [4]	Ours
Impersonation attack	x	x	x	o
Side channel attack over OBU or TPD	-	x	x	o
Trace attack	o	o	o	o
Replay attack	x	o	o	o
Man-in-the-middle attack	x	x	o	o
Privacy-preserving	o	o	o	o
Mutual authentication	x	x	x	o

x: Insecure. o: Secure. -: Does not concern.

8. Conclusions

Vehicle systems have developed significantly and they have recently helped people to drive more comfortably and safely. However, unsolved security problems and large quantities of traffic information have limited the use of vehicle systems. The VCC with message confirmation is the one of solutions to decline burdens of OBU's storage. And VCC helps to use the vast amount of vehicle information easily. In addition, to protect the vehicle information, key agreement and authentication process is also necessary to address malicious attacks, including communication security problems. Additionally, previous studies and the protocol of Limbasiya et al. are not safe for stored values in ideal or realistic TPDs. In this paper, we first showed that protocol of Limbasiya et al. is not secure against session key disclosure and impersonation attacks because of information leaked from a TPD. Their protocol also does not provide privacy of the users and mutual authentication property. Subsequently, we proposed a secure key agreement and authentication protocol for message confirmation in VCC. The proposed protocol withstands various attacks and provides privacy of

users and mutual authentication. We conducted formal security analysis and simulation to prove the security of the proposed protocol. Moreover, we compared computation, communication costs and the security properties with other related protocols. Thus, our proposed protocol is lightweight and suitable for VCC environments. As part of the future, we will put effort into developing a better protocol by applying the developed protocol to the real environment.

Author Contributions: Conceptualization, J.L.; Formal analysis, J.L., S.Y. and M.K.; Software, J.L. and M.K.; Supervision, Y.P.; Validation, S.Y., Y.P., S.L. and B.C.; Writing—original draft, J.L.; Writing—review & edigint, S.Y., M.K., Y.P., S.L. and B.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government(MIST) (No.2018-0-00312, Developing technologies to predict, detect, respond, and automatically diagnose security threats to automotive Ethernet-based vehicle).

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

References

1. Zhang, L.; Wu, Q.; Domingo-Ferrer, J.; Qin, B.; Hu, C. Distributed aggregate privacy-preserving authentication in VANETs. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 516–526. [CrossRef]
2. Zhang, J.; Cui, J.; Zhong, H.; Chen, Z.; Liu, L. PA-CRT: Chinese Remainder Theorem Based Conditional Privacy-preserving Authentication Scheme in Vehicular Ad-hoc Networks. *IEEE Trans. Dependable Secur. Comput.* **2019**.[CrossRef]
3. Liu, Z.; Xiong, L.; Peng, T.; Peng, D.; Liang, H. A realistic distributed conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Access* **2018**, *6*, 26307–26317. [CrossRef]
4. Limbasiya, T.; Das, D. Secure message confirmation scheme based on batch verification in vehicular cloud computing. *Pysical Commun.* **2019**, *34*, 310–320. [CrossRef]
5. Wazid, M.; Das, A.K.; Kumar, N.; Odelu, V.; Reddy, A.G.; Park, K.; Park, Y. Design of lightweight authentication and key agreement protocol for vehicular ad hoc networks. *IEEE Access* **2017**, *5*, 14966–14980. [CrossRef]
6. Kim, M.; Park, K.; Yu, S.; Lee, J.; Park, Y.; Lee, S.-W.; Chung, B. A Secure Charging System for Electric Vehicles Based on Blockchain. *Sensors* **2019**, *19*, 3028. [CrossRef] [PubMed]
7. Lee, J.; Yu, S.; Kim, M.; Park, Y.; Das, A.K. On the Design of Secure and Efficient Three-Factor Authentication Protocol Using Honey List for Wireless Sensor Networks. *IEEE Access* **2020**, *8*, 107046–107062. [CrossRef]
8. Yu, S.; Lee, J.; Park, Y.; Park, Y.; Lee, S.; Chung, B. A Secure and Efficient Three-Factor Authentication Protocol in Global Mobility Networks. *Appl. Sci.* **2020**, *10*, 3565. [CrossRef]
9. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.; Park, Y.H. AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment. *IEEE Internet Things J.* **2019**, *6*, 8804–8817. [CrossRef]
10. Lin, X.; Sun, X.; Ho, P.; Shen, X. GSIS: A secure and privacy-preserving protocol for vehicular communications. *IEEE Trans. Veh. Technol.* **2007**, *56*, 3442–3456.
11. Zhang, C.; Ho, P.; Tapolcai, J. On batch verification with group testing for vehicular communications. *Wirel. Netw.* **2011**, *17*, 1851–1865. [CrossRef]
12. Lee, C.C.; Lai, Y.M. Toward a secure batch verification with group testing for VANET. *Wirel. Netw.* **2013**, *19*, 1441–1449. [CrossRef]
13. 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.
14. Bayat, M.; Barmshoory, M.; Rahimi, M.; Aref, M.R. A secure authentication scheme for VANETs with batch verification. *Wirel. Netw.* **2015**, *21*, 1733–1743. [CrossRef]
15. He, D.; Zeadally, S.; Xu, B.; Huang, X. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2681–2691. [CrossRef]
16. Zhong, H.; Wen, J.; Cui, J.; Zhang, S. Efficient conditional privacy-preserving and authentication scheme for secure service provision in VANET. *Tsinghua Sci. Technol.* **2016**, *21*, 620–629. [CrossRef]

17. Chuang, C.M.; Lee, F.J. TEAM: Trust-extended authentication mechanism for vehicular ad hoc networks. *IEEE Syst. J.* **2014**, *8*, 749–758. [CrossRef]
18. Zhou, Y.; Zhao, X.; Jiang, Y.; Shang, F.; Deng, S.; Wang, X. An enhanced privacy-preserving authentication scheme for vehicle sensor network. *Sensors* **2017**, *17*, 2854. [CrossRef]
19. Wu, L.; Sun, Q.; Wang, X.; Wang, J.; Yu, S.; Zou, Y.; Liu, B.; Zhu, Z. An Efficient Privacy-Preserving Mutual Authentication Scheme for Secure V2V Communication in Vehicular Ad Hoc Network. *IEEE Access* **2019**, *7*, 55050–55063. [CrossRef]
20. Kenney, J. Dedicated short-range communications (DSRC) standards in the United States. *Proc. IEEE* **2011**, *99*, 1162–1182. [CrossRef]
21. Dolev, D.; Yao, A.C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
22. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology*; Springer Science + Business Media: Berlin, Germany; New York, NY, USA, 1999; pp. 388–397.
23. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 17 July 2019).
24. SPAN: A Security Protocol Animator for AVISPA. Available online: http://www.avispa-project.org/ (accessed on 17 July 2019).
25. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password based authenticated key exchange in the three-party setting. In *Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography*; Springer: Les Diablerets, Switzerland, 2005; pp. 65–84.
26. Park, K.; Park, Y.; Das, A.K.; Yu, S.; Lee, J.; Park, Y. A dynamic privacy-preserving key management protocol for V2G in social Internet of Things. *IEEE Access* **2019**, *7*, 76812–76832. [CrossRef]
27. Park, K.; Noh, S.; Lee, H.; Das, A.K.; Kim, M.; Park, Y.; Wazid, M. LAKS-NVT: Provably Secure and Lightweight Authentication and Key Agreement Scheme without Verification Table in Medical Internet of Things. *IEEE Access* **2020**, *8*, 119387–119404. [CrossRef]
28. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
29. Yu, S.; Park, K.; Park, Y. A secure lightweight three-Factor authentication scheme for IoT in cloud computing environment. *Sensors* **2019**, *19*, 3598. [CrossRef]
30. Park, Y.; Park, K.; Lee, K.; Song, H.; Park, Y. Security analysis and enhancements of an improved multi-factor biometric authentication scheme. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 1–12. [CrossRef]
31. Lee, J.; Yu, S.; Park, K.; Park, Y.; Park, Y. Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors* **2019**, *19*, 2358. [CrossRef]
32. Basin, D.; Modersheim, S.; Vigano, L. OFMC: A symbolic model checker for security protocols. *Int. J. Inf. Secur.* **2005**, *4*, 181–208. [CrossRef]
33. Turuani, M. The CL-Atse protocol analyser. In Proceedings of the International Conference on Rewriting Techniques and Applications (RTA), Seattle, WA, USA, 12–14 August 2006; pp. 227–286.
34. Cui, J.; Zhang, J.; Zhong, H.; Xu, Y. SPACF: A secure privacy-preserving authentication scheme for VANET with cuckoo filter. *IEEE Trans. Veh. Tech.* **2017**, *66*, 10283–10295. [CrossRef]
35. Mir, Z.H.; Fethi, F. LTE and IEEE 802.11 p for vehicular networking: A performance evaluation. *EURASIP J. Wirel. Commun. Netw.* **2014**, *1*, 89.
36. He, D.; Chen, C.; Chan, S.; Bu, J. Secure and efficient handover authentication based on bilinear pairing functions. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 48–53. [CrossRef]
37. Mostafa, A.; Vegni, A.M.; Singoria, R.; Oliveira, T.; Little, T.D.; Agrawal, D.P. A V2X-based approach for reduction of delay propagation in Vehicular Ad-Hoc Networks. In Proceedings of the 2011 11th International Conference on ITS Telecommunications (ITST), St. Petersburg, Russia, 23–25 August 2011; pp. 756–761.

© 2020 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 (http://creativecommons.org/licenses/by/4.0/).