A Secure and Efficient Group Key Agreement Scheme for VANET

Eko Fajar Cahyadi[1] and Min-Shiang Hwang[2,3]
(Corresponding author: Min-Shiang Hwang)

Faculty of Telecommunication and Electrical Engineering, Institut Teknologi Telkom Purwokerto[1]
Purwokerto, Indonesia (ekofajarcahyadi@ittelkom-pwt.ac.id)
Department of Computer Science & Information Engineering, Asia University[2]
Fintech and Blockchain Research Center, Asia University[3]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taichung, Taiwan (R.O.C.)
Email: mshwang@asia.edu.tw

(Received Jan. 4, 2023; Revised and Accepted Oct. 9, 2024; First Online Oct. 31, 2024)

Abstract

In order to protect the security and privacy of group communications in VANET, a secure and efficient group key negotiation scheme needs to be designed. In order to achieve efficient anonymous privacy and authentication mechanisms, batch authentication schemes and shared key mechanisms are generally used. In this article, we propose improving a secure and efficient group key agreement scheme for VANET in several areas to explicitly clarify the process and ensure the safety of the disseminated information.

Keywords: Authentication; Security and Privacy; Symmetric Cryptography; VANET

1 Introduction

Vehicular Ad hoc NETworks (VANET) has intelligent transportation system features that allow all vehicles on the road to communicate with each other through vehicle-to-vehicle (V2V) or vehicle-to-road infrastructure (V2I) [2, 8]. It consists of three main entities: Trust Authorization (TA), Road-Side Unit (RSU), and On-Board Unit (OBU). TA is the trust and security management center of the entire VANET. Its main job is to accept vehicle RSU and OBU registrations and make the vehicle a legal VANET vehicle. RSUs are semi-trusted fixed infrastructure placed along roads. The RSU is connected to nearby RSUs or the Internet and serves as the transfer point between the vehicle OBU and the Internet. Each vehicle is equipped with an OBU as the core processor of the vehicle, which performs identity authentication and receives and sends external messages [1, 5]. It can broadcast traffic-related messages such as location, speed, and direction to hundreds of other vehicles or RSUs every 100-300 milliseconds.

In [9], the authors proposed a secure and efficient group key agreement (SEGKA) scheme for vehicular ad hoc networks (VANET). The aim is to build a secure and effective group membership authentication key agreement mechanism for group communication. In VANET, vehicles with identical attributes, such as their location in the same roadside unit (RSU)'s coverage area, are organized as the same group communication [6]. Therefore, vehicle group communication refers to interactions among vehicles within the same attribute, which in [9], among vehicles in the same RSU area. In 2022, Want *et al.* proposed a new identity-based anonymous authentication scheme that aims to reduce the cost of pseudonym generation and key leakage faced by conditional privacy protection schemes in VANETs [12]. In 2023, Liu *et al.* proposed a blockchain-based decentralized identification code model to protect privacy and enhance security in vehicle cloud computing solutions [10]. Yang *et al.* Proposed a new attribute-based anonymous broadcast protocol to support the secure and anonymous transmission of vehicle safety broadcast messages and protect the privacy of the receiver [13].

VANET has the nature of high mobility and rapid topology changes, so there are always vehicles joins or leaving the communication group [3, 4, 7]. Therefore, a secure secret group key agreement mechanism must ensure the legality of that vehicles. In [9], as the manager, RSU must be able to verify a new vehicle before it joining the group and update the group key when another vehicle joins or leaves the group. Liu *et al.*'s [9] SEGKA uses symmetrical key encryption to reduce the cost of computing and improve efficiency. Their SEGKA scheme is practical and easy to implement. However, our investigation exhibited that the scheme suffered from some issues related to some sensitive information leakages due to the unencrypted messages. To introduce the scheme, we briefly review Liu *et al.*'s scheme in Section 2 and point out the problems in Sections 3 and 4. Meanwhile, to remedy the weaknesses, we give our modification and improvement in Section 5. Finally, the conclusion is given in Section 6.

2 Review of Liu *et al.*'s SEGKA

Liu *et al.*'s [9] SEGKA scheme consists of seven phases: *parameter initialization*, *vehicle and RSU registration*, *vehicle signing*, *RSU verification*, *group key generation*, *group member joining*, and *group member leaving*.

2.1 Parameter Initialization

In this early phase, TA generates some initial system parameters *params* for vehicles and RSU. First, it selects a cyclic additive group G_1 generated by P, and a cyclic multiplicative group G_2 with the same prime order q, to construct a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$. Then, TA selects a secret parameter $s \in Z_q^*$ as its master key and computes $P_{pub} = sP$ as its public key. TA selects a map-to-point hash function $H(\cdot) : \{0,1\}^* \rightarrow G_1$ and a one-way hash function $h(\cdot) : \{0,1\}^* \rightarrow Z_q^*$. Finally, TA broadcasts $params = \{G_1, G_2, \hat{e}, q, P, P_{pub}, H(\cdot), h(\cdot)\}$ to vehicles and RSU in the network.

2.2 Vehicle and RSU Registration

TA registers both vehicle V_i and RSU for being able to communicate in VANET. The a_i and b_i denote a shared secret key of TA - V_i and a shared secret key of V_i - RSU, respectively. TA computes $c_i = sH(a_i \oplus TID_i)$ and sends $REG_V = TID_i \parallel a_i \parallel b_i \parallel c_i$ to V_i. Finally, TA computes V_i's verification $VID_i = a_i \oplus TID_i$ and sends $REG_{RSU} = VID_i \parallel b_i$ to RSU.

2.3 Vehicle Signing

In this phase, V_i selects a random nonce $r_i \in Z_q^*$ to generates its pseudo-identity $PID_i = (PID_{i,1}, PID_{i,2})$, where $PID_{i,1} = r_iP$ and $PID_{i,2} = VID_i \oplus H(b_iPID_{i,1})$. Then, V_i computes its signature $\sigma_i = c_i + b_ic_ih(M_i)$, where $M_i = PID_i \parallel T_i$, and T_i is the signing time. Finally, V_i sends $D_i = r_i \parallel PID_i \parallel \sigma_i \parallel T_i$ to RSU.

2.4 RSU Verification

Upon receiving $D_i = r_i \parallel PID_i \parallel \sigma_i \parallel T_i$ from V_i, RSU will decrypts D_i using its secret key $DEC_{SK_{RSU}}(r_i \parallel PID_i \parallel \sigma_i \parallel T_i)$ and checks the freshness of T_i. In the single verification mode, RSU verifies σ_i, by checking whether $\hat{e}(\sigma_i, P) = \hat{e}(H(VID_i)(1 + b_ih(M_i)), P_{pub})$ is holds or not. Meanwhile, in the batch verification mode, RSU verifies σ_i, by checking whether $\hat{e}\left(\sum_{i=1}^{n} \sigma_i, P\right) = \hat{e}\left(\sum_{i=1}^{n} H(VID_i)(1 + b_ih(M_i)), P_{pub}\right)$ is holds or not.

2.5 Group Key Generation

After σ_i is authenticated, the RSU will generate the group key for vehicles in its area. RSU selects a random nonce $d_{RSU} \in Z_q^*$, and computes $D_i = d_{RSU}PID_{i,1}$ and $K_{RSU} = \hat{e}\left(\sum_{i=1}^{n} D_i, d_{RSU}P\right)$. Then, it computes its signature $\sigma_{RSU} = SK_{RSU}H(D)$, where $D = D_1 \parallel D_2 \parallel \cdots \parallel D_n$, and broadcasts $Z = \sigma_{RSU} \parallel D$ to vehicles in its area. After receiving Z, V_i verifies σ_{RSU} by checking whether $\hat{e}(\sigma_{RSU}, P) = \hat{e}(H(D), PK_{RSU})$ is holds or not. If yes, V_i computes the group key $K_i = \hat{e}\left(\sum_{i=1}^{n} D_i, r_i^{-1}D_i\right)$.

2.6 Group Member Joining

When a new vehicle V_a joins the network, it will selects a random nonce $r_a \in Z_q^*$ to generates its pseudo-identity $PID_a = (PID_{a,1}, PID_{a,2})$, where $PID_{a,1} = r_aP$ and $PID_{a,2} = VID_a \oplus H(b_aPID_{a,1})$. Then, V_a calculates its signature $\sigma_a = r_aH(PID_a) + b_ac_aH(T_a)$ and sends the encrypted $D_a = ENC_{PK_{RSU}}(r_a \parallel PID_a \parallel \sigma_a \parallel T_a)$ to RSU, with PK_{RSU} is the public key of RSU. After receiving D_a, RSU decrypts it using its secret key $DEC_{SK_{RSU}}(ENC_{PK_{RSU}}(r_a \parallel PID_a \parallel \sigma_a \parallel T_a))$ and check the freshness of T_a. The RSU verifies whether $PID_{a,2} = VID_a \oplus H(b_aPID_{a,1})$. If holds, RSU verifies σ_a by checking whether $\hat{e}(\sigma_a, P) = \hat{e}(H(VID_a)(1 + b_ah(M_a)), P_{pub})$ is holds or not. If holds, RSU allows V_a for joining the network. When V_a joins the network, RSU will update the group key by selects a random nonce $d'_{RSU} \in Z_q^*$, recomputes $D'_i = d'_{RSU}PID_{i,1}; (1 \leq i \leq n)$ and $D_a = d'_{RSU}PID_{a,1}$. Then, RSU computes $K'_{RSU} = \hat{e}\left(\sum_{i=1}^{n} D'_i + D_a, d'_{RSU}P\right)$ and its new signature $\sigma'_{RSU} = SK_{RSU}H(X')$, where $X' = D'_1 \parallel D'_2 \parallel \cdots \parallel D'_n \parallel D_a$. RSU broadcasts $Z' = \sigma'_{RSU} \parallel X'$ to the new group of vehicles. Upon receiving Z', vehicles will check whether $\hat{e}(\sigma'_{RSU}, P) = \hat{e}(H(X'), PK_{RSU})$ is holds or not. If holds, compute the new group key $K'_i = \hat{e}\left(\sum_{i=1}^{n} D'_i + D_a, r_i^{-1}D'_i\right)$.

2.7 Group Member Leaving

When V_i leaves the network, RSU will update the group key for the remaining $n-1$ vehicles. RSU selects $d'_{RSU} \in Z_q^*$ and computes $D'_i = d'_{RSU}PID_{i,1}; (1 \leq i \leq n-1)$. Then, RSU computes $K'_{RSU} = \hat{e}\left(\sum_{i=1}^{n-1} D'_i, d'_{RSU}P\right)$ and its new signature $\sigma'_{RSU} = SK_{RSU}H(X')$, where $D' = D'_1 \parallel D'_2 \parallel \cdots \parallel D'_{n-1}$. RSU broadcasts $Z' = \sigma'_{RSU} \parallel X'$ to the remaining vehicles. Upon receiving Z', vehicles will check whether $\hat{e}(\sigma_{RSU}, P) = \hat{e}(H(D), PK_{RSU})$ is holds or not. If holds, compute the new group key $K'_i = \hat{e}\left(\sum_{i=1}^{n-1} D'_i, r_i^{-1}D'_i\right)$.

3 Analysis of the Problems in the Original Paper

The original paper [9] is exposed to two main drawbacks, such as replaying and DoS attacks in the *vehicle signing* phase. However, first, we need to clarify the sequential group agreement process of Liu *et al.*'s scheme. For a better understanding, we explicitly describe them as follows:

3.1 Inconsistency on Protocol Sequence

In Section 4 and Figure 2 of the original paper, the authors explain the sequential group agreement process as follows:

1) TA generates the public parameters for vehicles and RSUs;
2) The RSU requests for the registration process to TA;
3) TA registered the RSU;
4) The vehicle sends its registration information to TA;
5) TA registered the vehicle;
6) TA sends vehicle's verification information and shared key to RSU;
7) The vehicle sends its signature to the RSU;
8) The RSU verifies the vehicle's signature and generates a group key for vehicles in its area.

In Step (2), it is more likely that TA directly registers the RSU without any request, so, this step is unnecessary. Meanwhile, in Step (4), we expect the vehicle V_i will approach TA to register and provide its personal information, such as name, address, phone number, email, *etc* [11]. However, from the *vehicle and RSU registration* phase of the original paper, it is shown that TA directly registers the vehicle as $REG_V = TID_i \parallel a_i \parallel b_i \parallel c_i$, without a prerequisite step done by the vehicle. Therefore, an inconsistency happens in [9], where Steps (2) and (4) are not being executed. As the authors declare that TA sends REG_V to V_i through a secure channel, so REG_V is presumably transacted in the offline mode, where Step (4) should be executed. To emphasize this section, Steps (1) to (8) described by the authors above are correct, except they don't follow those steps correctly. Furthermore, all of these offline processes should be mentioned for clarity since the authors ensured the channel is secured.

3.2 Problem in Replaying and DoS Attacks

In the *vehicle signing* phase, V_i sends information $D_i = r_i \parallel PID_i \parallel \sigma_i \parallel T_i$ to RSU, containing V_i's random nonce r_i, which is used to prevent $\mathcal{A}$ from tracking the vehicle, V_i's pseudo-identity PID_i, and shared secret key between V_i and TA a_i, that not being encrypted. Since the channel between V_i and RSU is not secure (contrary to the statement mentioned by the authors), $\mathcal{A}$ can get into the message and launch several kinds of attacks.

Before discussing the attacks, we need to address the utilization of notation D_i in this comment first. In the original paper, the authors using notation D_i for two different definitions. First, it is used in the *vehicle signing* phase to describe the signature message $D_i = r_i \parallel PID_i \parallel \sigma_i \parallel T_i$ sends by V_i to RSU. Second, it is used in the *group key generation* phase by RSU to computes its group key $K_{RSU} = \hat{e}\left(\sum_{i=1}^{n} D_i, d_{RSU}P\right)$, where $D_i = d_{RSU}PID_{i,1}$. For the sake of consistency, since one notation only can represent one definition, we assume that the first D_i used by V_i to describe its signature message should be written as X_i (refers to Figure 5 of the original paper). This problem also implies the subsequent operation of $\sigma'_{RSU} = SK_{RSU}H(X')$ in *group member joining* and *group member leaving* phases, where the operation should be written as $\sigma'_{RSU} = SK_{RSU}H(D')$. So, from the now on, we redefine $D_i = r_i \parallel PID_i \parallel \sigma_i \parallel T_i$ as $X_i = r_i \parallel PID_i \parallel \sigma_i \parallel T_i$.

3.2.1 Problems

In replaying attack problem, upon achieving $X_i = r_i \parallel PID_i \parallel \sigma_i \parallel T_i$, the $\mathcal{A}$ replaces the previous timestamp T_i with T'_i generated by itself. Then, in a future time point, $\mathcal{A}$ sends $X'_i = r_i \parallel PID_i \parallel \sigma_i \parallel T'_i$ to challenge the RSU. On the RSU side, the forged message, which contains both the previous information and future timestamp, could pass the verification process. In this way, $\mathcal{A}$ could impersonate the legitimate vehicle by eavesdropping on the transmitted message. Hence, the Liu *et al.*'s [9] SEGKA scheme is vulnerable to replay attack.

Similar to the replaying attack method, by changing T_i to T'_i then sends X'_i to RSU, $\mathcal{A}$ can make a denial of service (DoS) attack towards challenged RSU. Upon receiving X'_i, RSU would verify it in the single or batch verification mode in the *RSU verification* phase. In this case, since $\mathcal{A}$ able to sends multiple X'_is with the future timestamp T'_i, then $\mathcal{A}$ can flood the RSU with many unofficial requests and take down the communication networks in that RSU's area by doing it numerous times.

3.2.2 Solution

To overcome the replaying and DoS attacks discussed above, in the *vehicle signing* phase, the authors can simply encrypts X_i using RSU's public key $ENC_{PK_{RSU}}(r_i \parallel PID_i \parallel \sigma_i \parallel T_i)$. Then upon receiving the information from V_i, RSU decrypts the information using its secret key $DEC_{SK_{RSU}}(ENC_{PK_{RSU}}(r_i \parallel PID_i \parallel \sigma_i \parallel T_i))$.

4 Correction to Writing Errors

In our opinion, there are some mistakes in [9] due to writing errors and ambiguous explanations that need to be addressed. Here, we described them in the following items:

- First, in the *parameter initiation* phase, the authors state if TA broadcasts $params = \{G_1, G_2, \hat{e}, q, P, P_{pub}, H(\cdot), h(\cdot)\}$ to vehicles in the network. However, since the *parameter initiation* phase is sequenced before the *vehicle and RSU registrations* phase, it means TA does not broadcast it to vehicles, instead of giving it in an offline manner or coupling the process together with the *vehicle and RSU registrations* phase.

- If we refer to Steps (2) to (4) in Section 3.1, after the *parameter initiation* phase, RSU registers itself to TA, then subsequently verified by TA with $REG_{RSU} = VID_i \parallel b_i$. Next, vehicle V_i registers itself to TA, then verified with $REG_V = TID_i \parallel a_i \parallel b_i \parallel c_i$. Unfortunately, this procedure sequence is not correct since VID_i in $REG_{RSU} = VID_i \parallel b_i$ must be computed after TA verifies the vehicle registration.

- In the *group member joining* phase (Section 4.6, Point (4)) of the original paper, the authors cite Equation (4), expressing $\hat{e}(\sigma_{RSU}, P) = \hat{e}(H(D), PK_{RSU})$, where vehicles, including V_a, verifying the new RSU's signature σ'_{RSU} before they compute their new group key $K'_i = \hat{e}\left(\sum_{i=1}^{n} D'_i + D_a, r_i^{-1} D'_i\right)$. At this point, the verification process done by vehicles should be written as $\hat{e}(\sigma'_{RSU}, P) = \hat{e}(H(D'), PK_{RSU})$ since notation D, and D' have a different interpretation. Notation D expresses $(D_1 \parallel D_2 \parallel \cdots \parallel D_n)$, meanwhile, notation D' expresses $(D'_1 \parallel D'_2 \parallel \cdots \parallel D'_n \parallel D_a)$.

- Still in the *group member joining* phase, the calculation of new joining vehicle V_a's signature $\sigma_a = r_a H(PID_a) + b_a c_a H(T_a)$ is different from the existing vehicle V_i's signature calculation $\sigma_i = c_i + b_i c_i h(M_i)$. We can see if the calculation of σ_a is more expensive than σ_i since it has two map-to-point hash functions $H(\cdot)$. There is no further explanation about this particularly same process.

- The scheme analysis in Section 5.2. of the original paper has two sub-sections that discuss replaying attack resistance, which the first one (Section 5.2.3.) is wrongly written. Meanwhile, in Section 5.2.5., the authors write a new notation VPK_i to prove the traceability feature of the scheme, which is not written or discussed in the previous sections. It seems meant to be VID_i, not VPK_i.

5 Modification of the SEGKA

Based on our corrections in Sections 3 and 4, we propose a modification and light improvement towards the SEGKA [9] scheme. We modify the *vehicle signing*, *RSU verification*, *group key generation*, *group member joining* and *group member leaving* phases.

5.1 Vehicle Signing

V_i selects $r_i \in Z_q^*$ to generates $PID_i = (PID_{i,1}, PID_{i,2})$, where $PID_{i,1} = r_i P$ and $PID_{i,2} = a_i \oplus TID_i \oplus H(b_i PID_{i,1})$. Then, V_i computes $\sigma_i = c_i + b_i c_i h(M_i)$, where $M_i = PID_i \parallel T_i$. As discussed in Section 3.2., in this comment, we redefine D_i used by V_i to describe its signature as X_i. Information X_i then encrypted using RSU's public key $X_i = ENC_{PK_{RSU}}(r_i \parallel PID_i \parallel \sigma_i \parallel T_i)$, and sends X_i to RSU.

5.2 RSU Verification

RSU decrypts X_i by $DEC_{SK_{RSU}}(ENC_{PK_{RSU}}(r_i \parallel PID_i \parallel \sigma_i \parallel T_i))$ and checks the freshness of T_i. RSU verifies σ_i, by checking whether $\hat{e}(\sigma_i, P) = \hat{e}(H(VID_i)(1 + b_i h(M_i)), P_{pub})$ and $\hat{e}\left(\sum_{i=1}^{n} \sigma_i, P\right) = \hat{e}\left(\sum_{i=1}^{n} H(VID_i)(1 + b_i h(M_i)), P_{pub}\right)$ is holds or not, in the single and batch verification modes, respectively.

5.3 Group Key Generation

After σ_i is authenticated, RSU selects $d_{RSU} \in Z_q^*$ and computes $D_i = d_{RSU} PID_{i,1}$. In this phase, we make a modification where the RSU will do the summation of D_i as $D_G = \sum_{i=1}^{n} D_i$. Therefore, $K_{RSU} = \hat{e}\left(D_G, d_{RSU} P\right)$ and $\sigma_{RSU} = SK_{RSU} H(D)$, where $D = D_G \parallel D_1 \parallel D_2 \parallel \cdots \parallel D_n$. By this modification, all vehicles do not need to compute $\sum_{i=1}^{n} D_i$ and can save $(n-1)$ summation operations. RSU then broadcasts $Z = \sigma_{RSU} \parallel D$ to vehicles in its area. After receiving Z, V_i verifies σ_{RSU} by checking whether $\hat{e}(\sigma_{RSU}, P) = \hat{e}(H(D), PK_{RSU})$ is holds or not. If yes, V_i computes $K_i = \hat{e}\left(D_G, r_i^{-1} D_i\right)$. The process of this phase is shown in Figure 1.

5.4 Group Member Joining

When V_a joins the network, it selects $r_a \in Z_q^*$ to generates $PID_a = (PID_{a,1}, PID_{a,2})$, where $PID_{a,1} = r_a P$ and $PID_{a,2} = a_a \oplus TID_a \oplus H(b_a PID_{a,1})$. Next, V_a calculates its signature σ_a. As we mentioned in Section 4, in [9], the calculation of σ_i is different from σ_a. Therefore, we synchronize $\sigma_a = c_a + b_a c_a h(M_a)$, where $M_a = PID_a \parallel T_a$. Then, V_a sends $X_a = ENC_{PK_{RSU}}(r_a \parallel PID_a \parallel \sigma_a \parallel T_a)$ to RSU. After receiving X_a, RSU decrypts it $DEC_{SK_{RSU}}(ENC_{PK_{RSU}}(r_a \parallel PID_a \parallel \sigma_a \parallel T_a))$ and check the freshness of T_a. The RSU verifies whether $PID_{a,2} = VID_a \oplus H(b_a PID_{a,1})$ is holds or not. If holds, RSU verifies whether $\hat{e}(\sigma_a, P) = \hat{e}(H(VID_a)(1 + b_a h(M_a)), P_{pub})$ is holds or not. If holds, RSU allows V_a for joining the network. When V_a joins the network, RSU reselects $d'_{RSU} \in Z_q^*$, then computes $D'_i = d'_{RSU} PID_{i,1}$ with $(1 \le i \le n)$, and $D_a = d'_{RSU} PID_{a,1}$. RSU do the summation of $D'_G = \sum_{i=1}^{n} D'_i + D_a$, then computes

Group key generation

RSU	**Vehicle**
Selects a random nonce $d_{RSU} \in Z_q^*$	
Computes $D_i = d_{RSU} PID_{i,1}$.	
Computes $D_G = \sum_{i=1}^{n} D_i$.	
Computes $K_{RSU} = \hat{e}\left(D_G, d_{RSU}P\right)$	
Computes $\sigma_{RSU} = SK_{RSU}H(D)$, where $D = D_G \parallel D_1 \parallel D_2 \parallel \cdots \parallel D_n$.	
Broadcasts $Z = \sigma_{RSU} \parallel D$ to vehicles.	
$\xrightarrow{\text{Sends } Z}$	
	Obtains Z.
	Verifies whether $\hat{e}(\sigma_{RSU}, P) = \hat{e}(H(D), PK_{RSU})$.
	If valid, computes $K_i = \hat{e}\left(D_G, r_i^{-1}D_i\right)$.

Figure 1: Modification of *group key generation* phase in [9]

$K'_{RSU} = \hat{e}\left(D'_G, d'_{RSU}P\right)$ and $\sigma'_{RSU} = SK_{RSU}H(D')$, where $D' = D'_G \parallel D'_1 \parallel D'_2 \parallel \cdots \parallel D'_n \parallel D_a$. RSU broadcasts $Z' = \sigma'_{RSU} \parallel D'$ to the new group of vehicles. Upon receiving Z', vehicles check whether $\hat{e}(\sigma'_{RSU}, P) = \hat{e}(H(D'), PK_{RSU})$ is holds or not. If holds, computes $K'_i = \hat{e}\left(D'_G, r_i^{-1}D'_i\right)$. The main advantage of this improvement is the same as the previous *group key generation* phase. The vehicles in the group do not need to perform $(n-1)$ summation operations of $(D_i + D_a)$ to compute K'_i. The process of this improvement is shown in Figure 2.

5.5 Group Member Leaving

When V_i leaves the network, RSU updates K_i for $n-1$ vehicles. RSU selects $d'_{RSU} \in Z_q^*$ and computes $D'_i = d'_{RSU}PID_{i,1}$ with $(1 \leq i \leq n-1)$. Then, RSU computes $D'_G = \sum_{i=1}^{n-1} D'_i$, $K'_{RSU} = \hat{e}\left(D'_G, d'_{RSU}P\right)$, and $\sigma'_{RSU} = SK_{RSU}H(D')$, where $D' = D'_G \parallel D'_1 \parallel D'_2 \parallel \cdots \parallel D'_{n-1}$. RSU broadcasts $Z' = \sigma'_{RSU} \parallel D'$ to the remaining vehicles. Upon receiving Z', vehicles will check whether $\hat{e}(\sigma_{RSU}, P) = \hat{e}(H(D), PK_{RSU})$ is holds or not. If holds, computes $K'_i = \hat{e}\left(D'_G, r_i^{-1}D'_i\right)$. Similar to the previous *group key generation* and *group member joining* phases, the advantage of this improvement in this phase is the existing vehicles do not need to perform $(n-2)$ summation operations of D_i to compute K'_i.

6 Conclusion

In this comment, we show that the SEGKA scheme proposed by Liu *et al.* is suffered from several attacks such as identity-privacy preserving violation, replaying, and DoS attacks. To preserve the disseminated information from those three attacks, we encrypt both REG_V and X_i in the *vehicle and RSU registration* and *vehicle signing* phases, respectively. We also addressed some mistakes due to writing errors and ambiguous explanations that appeared in the original paper. Finally, to minimize the computation cost on the vehicle's side, we shift the summation operation of D_G in the *group key generation*, *group member joining*, and *group member leaving* phases, are only done by RSU.

Acknowledgments

The National Science and Technology Council partially supported this research, Taiwan (ROC), under contract no.: NSTC 112-2221-E-507 -005 -MY3, MOST 109-2221-E-468-011-MY3, and NSTC 112-2221-E-468-007.

References

[1] E. F. Cahyadi, C. Damarjati, M. S. Hwang, "Review on identity-based batch verification schemes for security and privacy in VANETs," *Journal of Electronic Science and Technology*, vol. 20, no. 1, pp. 92–110, 2022.

[2] E. F. Cahyadi, M. S. Hwang, "An improved efficient anonymous authentication with conditional privacy-preserving scheme for VANETs," *PLOS ONE*, vol. 16, no. 9, pp. 1–13, 2021.

[3] E. F. Cahyadi, M. S. Hwang, "A comprehensive survey on certificateless aggregate signature in vehicular ad hoc networks," *IETE Technical Review*, vol. 39, no. 6, pp. 1265–1276, 2022.

[4] E. F. Cahyadi, M. S. Hwang, "A lightweight BT-based authentication scheme for illegal signatures identification in VANETs," *IEEE ACCESS*, vol. 10, pp. 133869–133882, 2022.

Improvement in *group member joining*

New & existing vehicles	**RSU**
	V_a is verified.
	Selects $d'_{RSU} \in Z_q^*$.
	Computes $D'_i = d'_{RSU} PID_{i,1}; (1 \leq i \leq n)$, and $D_a = d'_{RSU} PID_{a,1}$.
	Computes $D'_G = \sum_{i=1}^{n} D_i + D_a$.
	Computes $K'_{RSU} = \hat{e}\left(D'_G, d'_{RSU} P\right)$.
	Computes $\sigma'_{RSU} = SK_{RSU} H(D')$, where $D' = D'_G \parallel D'_1 \parallel D'_2 \parallel \cdots \parallel D'_n \parallel D_a$.
	Computes $Z' = \sigma'_{RSU} \parallel D'$.
	Sends Z' ←
Obtains Z'.	
Verify $\hat{e}(\sigma'_{RSU}, P) = \hat{e}(H(D'), PK_{RSU})$.	
If σ'_{RSU} valid, compute $K'_i = \hat{e}\left(D'_G, r_i^{-1} D'_i\right)$.	

Figure 2: Improvement of *group member joining* phase in [9]

[5] E. F. Cahyadi, M. S. Hwang, "An improved efficient authentication scheme for vehicular ad-hoc networks with batch verification using bilinear pairings," *International Journal of Embedded Systems*, vol. 15, no. 2, pp. 139–148, 2022.

[6] E. F. Cahyadi, F. Khair, T. Ariyadi, M. S. Hwang, "An improved dual authentication techniques for secure data transmission in VANETs," in *International Conference on Information Technology and Computing (ICITCOM'23)*, Universitas Muhammadiyah Yogyakarta, Indonesia, pp. 1–6, 2023.

[7] E. F. Cahyadi, T. W. Su, C. C. Yang, M. S. Hwang, "A certificateless aggregate signature scheme for security and privacy protection in VANET," *International Journal of Distributed Sensor Networks*, vol. 18, no. 5, pp. 1–21, 2022.

[8] C. T. Li, M. S. Hwang, Y. P. Chu, "A secure and efficient communication scheme with authenticated key etablishment and privacy preserving for vehicular ad hoc networks," *Computer Communications*, vol. 31, no. 12, pp. 2803–2814, 2008.

[9] L. Liu, Y. Wang, J. Zhang, Q. Yang, "A secure and efficient group key agreement scheme for VANET," *Sensors*, vol. 19, no. 3, pp. 482, 2019.

[10] Z. Liu, X. Ma, J. Bai, M. Xiao, F. Tang, "Privacy-preserving vehicular cloud computing based on blockchain and decentralized identifier," *International Journal of Network Security*, vol. 25, no. 5, pp. 849–858, 2023.

[11] P. Vijayakumar, M. Azees, A. Kannan, L. J. Deborah, "Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1015–1028, 2016.

[12] X. Wang, Q. Chen, Z. Peng, Y. Wang, "An efficient and secure identity-based conditional privacy-preserving authentication scheme in VANETs," *International Journal of Network Security*, vol. 24, no. 4, pp. 661–670, 2022.

[13] G. Yang, L. Zhang, R. Ma, "Privacy-preserving broadcast protocol in vehicular ad hoc networks," *International Journal of Network Security*, vol. 25, no. 3, pp. 468–476, 2023.

Biography

Eko Fajar Cahyadi is a lecturer in the Faculty of Telecommunication and Electrical Engineering, Institut Teknologi Telkom Purwokerto, Indonesia. He received Ph.D. degree in the Department of Computer Science and Information Engineering, Asia University, Taiwan in 2021 and the B. Eng. and M. Sc. degree in electrical engineering from Institut Sains dan Teknologi Akprind Yogyakarta in 2009, and Institut Teknologi Bandung in 2013, respectively. His research interest includes information security, VANETs, and WLANs.

Min-Shiang Hwang received M.S. in industrial engineering from National Tsing Hua University, Taiwan in 1988, and a Ph.D. degree in computer and information science from National Chiao Tung University, Taiwan, in 1995. He was a professor and Chairman of the Department of Management Information Systems, NCHU, during 2003-2009. He was also a visiting professor at the University of California (UC), Riverside, and UC. Davis (USA) during 2009-2010. He was a distinguished professor of the Department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Excellent Research Award of National Science Council (Taiwan). Dr. Hwang was a dean of the College of Computer Science, Asia University (AU), Taichung, Taiwan. He is currently a chair professor with the Department of Computer Science and Information Engineering, AU. His current research interests include cryptography, Steganography, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.