IET Intelligent Transport Systems

Research Article

IET Journals
The Institution of Engineering and Technology

SAPSC: SignRecrypting authentication protocol using shareable clouds in VANET groups

ISSN 1751-956X
Received on 6th October 2018
Revised 15th May 2019
Accepted on 20th May 2019
E-First on 19th June 2019
doi: 10.1049/iet-its.2018.5474
www.ietdl.org

Sneha Kanchan[1] ✉, Garima Singh[1], Narendra S. Chaudhari[2]

[1]Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology (VNIT), Nagpur, Maharashtra, India
[2]Department of Computer Science and Engineering, Indian Institute of Technology (IIT), Indore, Madhya Pradesh, India
✉ *E-mail: sneha.kanchan159@gmail.com*

Abstract: Security and mutual reliability are the crucial requirements of an *ad hoc* network as nodes are dependent upon each other for routing and forwarding their messages. Vehicular *ad hoc* networks (VANETs) are no exception and are always at the risk of impersonation attack. An authentication protocol protects the identity of network entities from being impersonated. Occasionally, they require re-encryption technique which enables any other node to communicate on behalf of an unavailable node. The technology is used to provide backup in emergencies. We propose a secure authentication algorithm to efficiently re-encrypt the messages using signcryption. For faster computation and routing, the authors have used shareable clouds in VANET groups. Security of the protocol is proved using Burrows–Abadi–Needham logic and validated by simulation tool automated validation of internet security protocols and applications.

1 Introduction

Vehicular *ad hoc* network (VANET) is the technology call of the intelligent transport system, which enables vehicles to talk with each other in order to avoid road accidents. It also facilitates supplementary services in cars such as self-activated brakes, self-executing toll collection, platooning and traffic updates, which are helpful in fast and safe driving. Many lives are at risk on road, and a single mistake or delay can cause serious accidents. Hence, security and robustness of such networks are of pivotal importance.

Being a dynamic real-time *ad hoc* network, VANET is always vulnerable to security threats, and nodes need to protect themselves against these threats. Existing protocols for VANET do not provide a complete authentication between all entities of the network causing several attacks. A vehicle is authenticated based on its identity. If attackers steal this identity, they can harm the system drastically by various types of impersonation attacks such as Sybil attack and man-in-the-middle attack [1]. Vehicles neither have a fixed mobility pattern nor any fixed infrastructure. If there is no proper authentication, it is easier for intruders to invade in the network. While designing the authentication protocol, the sender's privacy should never be compromised as it would increase intruder's knowledge base which can be used to attack the network [2].

Cloud computing is the prime choice of VANET researchers to cope up with the need for real-time calculation. It has a remarkable impact on traffic management availing faster computation and ample storage [3]. A shareable cloud can be used among different entities. However, requests sent by vehicles to the cloud should be kept anonymous. Even if the cloud is compromised, it should not be able to distinguish individual vehicles. Anonymity helps in preserving the privacy of the nodes. Group signature (GS) is a popular approach to hide the identity of the vehicle, where each member of the group shares the same signature. Hence, a receiver can easily identify that the sender is from a legitimate group but cannot determine the exact sender.

Table 1 Members of communication

Symbols	Definition	Symbols	Definition
MM_1	main MM	C	cloud
MM_2	deputy MM	P	proxy
TM	tracing manager	V_1, V_2	vehicles

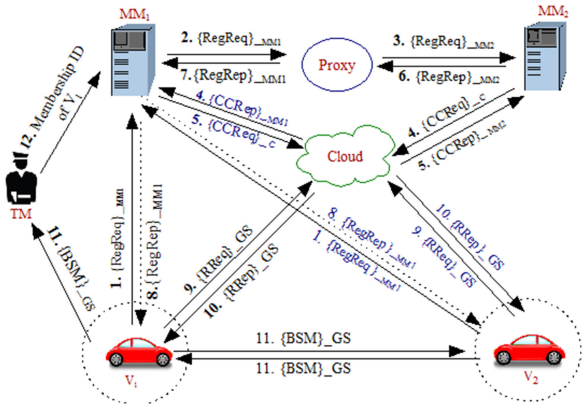


Fig. 1 *Communication network in VANET*

1.1 Our contribution

We propose a secure communication network in VANET, where vehicles are divided into groups, and GSs are used to authenticate them. At first, when a vehicle enters the network, it needs to register itself with the membership manager (MM). In reply, it receives the latest GS, which is used to prove its group authenticity. MM_1 and MM_2 both works as MM; MM_1 is the main MM, while MM_2 works as an alternative of MM_1. Table 1 represents the members of communication which are MM_1, MM_2 tracing manager (TM), cloud, proxy, and vehicles. Fig. 1 shows the fully connected network, in which black, blue and dotted lines represent the flow of normal messages, duplicate steps, and the ideal communication, respectively. Since VANET is an *ad hoc* network and identities of nodes are secret, it is always difficult to have a pre-determined authentication key shared between them. We have to develop an authentication technique among all six entities mentioned in Table 1. Vehicles need to communicate with MM_1 and cloud; MM_1 needs to communicate with TM, cloud, proxy, MM_2 and vehicles; TM needs to communicate with MM_1; proxy needs to communicate with MM_1 and MM_2; and cloud needs to communicate with vehicles. Without a strong authentication algorithm, an intruder can invade in the network by impersonating any poorly authenticated

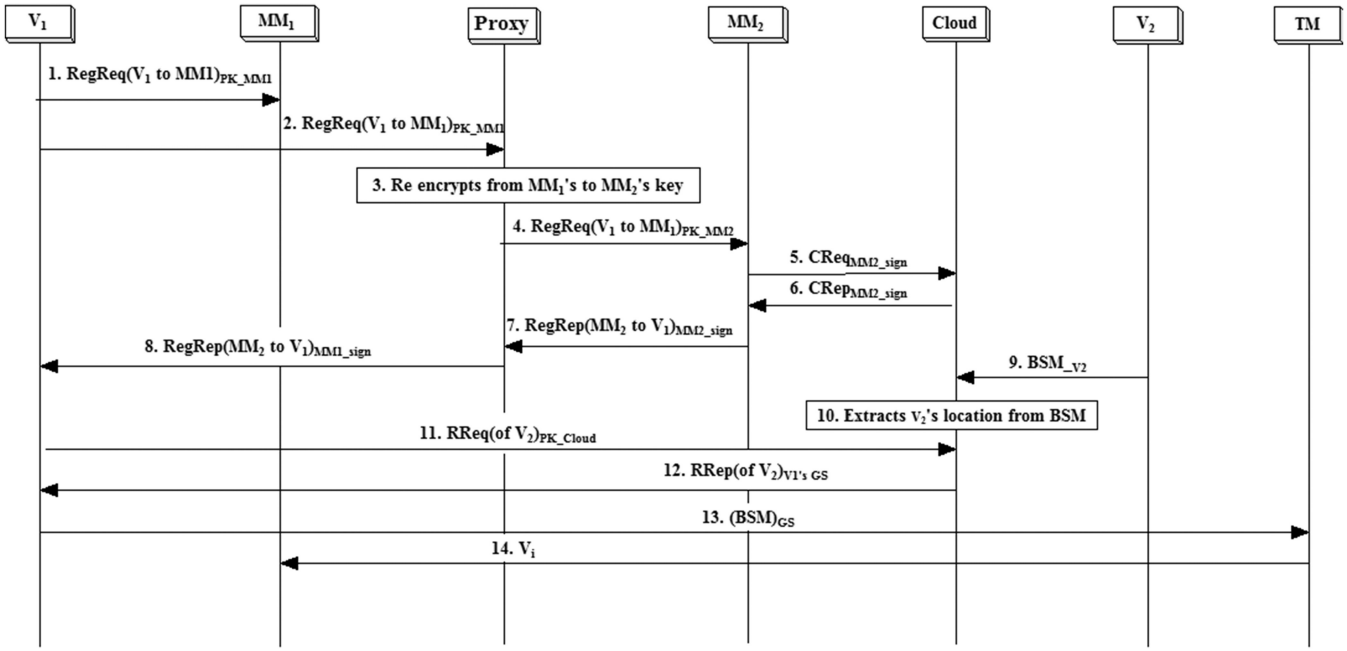


Fig. 2 *Proposed authentication protocol for VANET*

entity. The common communication messages, which are needed to be authenticated can be given as follows:

i. RegReq($A \Rightarrow MM_1$): A sends a request to MM_1 for its registration in the group. It is picked up by proxy if MM_1 is unavailable.
ii. RegReq($A \Rightarrow MM_1$): Proxy picks the message if MM_1 is not available.
iii. RegReq(Proxy $\Rightarrow MM_2$): Proxy forwards registration request (RegReq) after re-encrypting it, so that MM_2 may open message on behalf of MM_1.
iv. CCReq($MM_1/MM_2 \Rightarrow$ Cloud): MM sends computing request to the cloud whenever necessary.
v. CCRep($MM_1/MM_2 \Leftarrow$ Cloud): Cloud replies to MM.
vi. RegRep($A \Leftarrow MM_2$): MM_2 replies to A with group keys.
vii. RegRep($A \Leftarrow$ Proxy): Proxy re-signs the registration reply (RegRep) with MM_1's signature and again forwards it to A.
viii. RegRep($A \Leftarrow MM_1$): Since the message is signed with MM_1's signature, A assumes that the RegRep is sent by MM_1.
ix. RReq($A \Rightarrow$ Cloud): A sends routing request to the cloud whenever it needs to find out the route to a particular vehicle.
x. RRep($A \Leftarrow$ Cloud): Cloud replies back with the route of the requested vehicle.
xi. BSM($A \Rightarrow B$) or BSM($A \Rightarrow$ TM): A transmits basic safety messages (BSM) in the network which is eventually received by other vehicles and TM.
xii. V_{ID}(TM $\Rightarrow MM_1$): In case of accidents, TM calculates vehicle's ID and forwards it to MM_1.

The expansion of the abbreviated form of packet names has been given in the text. MM and TM are managers of a group, and they are very much insiders to the network. Vehicles enter the network, and request for registration. After registration, those are also legitimate group members. Cloud is an external entity, whose services are used to fasten the computation process. Fig. 2 represents the time sequence diagram to show the order of the actions performed in the system. Entities encrypt their message before sending it into the network. All these messages should be from authorised entity only and, receivers must be able to authenticate at their end as well. In this research, we are integrating signcryption with re-cryptography to make our protocol faster and robust even if there is network failure. Cloud provides faster computing making it more efficient. Our protocol aims to provide an end to end authentication mechanism for the above technologies. Hence, the main contribution of the proposed protocol is summarised as an integration of the following techniques:

i. *SignCryption*: Combining Signature and encryption steps into a single step to improve efficiency.
ii. *Re-encryption*: Re-encrypting the message from one manager to another manager.
iii. *Re-signature*: Re-signing messages from one manager to another.
iv. *Secure sharable clouds*: Using a cloud which is shared among different groups and entities in a secure way.
v. *End-to-end authentication protocol*: Providing an end-to-end authentication protocol for the described network.

1.2 Required criteria for the authentication protocol

i. The new protocol must follow the existing communication rules in VANET network.
ii. All entities registered in the network must have been mutually authenticated before starting the communication.
iii. The keys used in the process must be kept secret even from those entities which are part of the network but not part of communication.
iv. The number of computations should be minimal in order to lower the communication computation and signalling overhead.
v. The identity of the vehicles must not be revealed, and the authentication should be done on the basis of GS.
vi. Security and privacy should never be compromised while authenticating the entities.
vii. Overall performance should increase by a perceptible amount.

1.3 Organisation of the paper

Starting with the introduction in Section 1, Section 2 proceeds with the related work. Section 3 proposes our SignRecrypting authentication protocol using shareable clouds (SAPSC) protocol and Section 4 deals with the formal verification of the security of the protocol. In Section 5, efficiency analysis of our protocol with existing protocol has been given which compares performance and bandwidth consumed. Finally, the paper ends with the conclusion and future scope in Section 6.

2 Related work

Ever since its initiation in 1997 by Zheng, signcryption has been a subject which is undergoing intense studies [4]. The paper showed that because of the reduced size of the signature, packet size could be reduced to almost 90% which also conserves 50% of the computation and transmission time spent on a specific message. Later, the outlook was diverted from a mere constitution of encryption with signature to collaborate the same with already successful means in various areas.

Ateniese *et al.* advanced with the idea of SignREcryption for the first time in 2006 [5]. Though it was initiated as a security supplementation to the actual re-encryption objective by Blaze, Bleumer, and Strauss (BBS) [6], signcryption appeared captivating to many intellectuals, and hence the authors published their patent in the year 2012 [7]. To master the bidirectional perspective of BBS, a unidirectional character-based re-encryption scheme was proposed in [8], where a verified proxy converts the individuality of one node to another. Canetti and Hohenberger, in cooperation, computed a formula for chosen-ciphertext secure proxy re-encryption contrary to the prevailing defined safety procedures at that time [9]. Their work laid the foundation for the research works of Ateniese, Benson, and Hohenberger who published their work on key-private re-encryption for chosen plaintext attack [10]. This scheme preserved the identity of representatives of the re-encryption key (ReK) even from the proxy.

GSs were accepted widely after its introduction in 1991 because of its privacy preservation property [11]. In 2007, Lin *et al.* proposed an excellent paper on the certainty and privacy conservation of VANET network, which was based on identity-based cryptography in GSs [12]. It was more advantageous than the programmes having a large number of nameless certificate and long certificate revocation list (CRL). However, the execution of the algorithm was troubled due to the considerable signature size (192 bytes). Discriminative association with direct detection was put forward in 2014 by Mamun, Saiful, and Miyaji as a polished approach to friendly GS which leads to revocation by re-keying the signature and verifier-local revocation (VRL) [13]. These programmes depend on the count of individuals either existing or vacating the group. In contrast to the VRL approach, a dynamic accumulator was proposed as a revocation scheme by Camenisch and Lysyanskaya, which updated a group public key without relying on the number of entities in the system [14]. Therefore, after any addition or revocation, the number of updates needed is not determined by the count of already existing nodes. In 2017, Kuo and team proposed an algorithm which showed the probability of impersonation attack on simple dynamic accumulator [15]. To remove the possibility of identity theft, they used separate spectator sets in which nodes cannot imitate others. Certificate list and group keys are needed to be updated only once for many members' revocation, instead of updating the group key after revoking each member.

Proxy re-signature is not in much use in VANET, in spite of being a precise way to relieve the burden from the authorities. However, now scholars are embracing it in wireless system promptly. In 2017, Wei, Yang, and Mu presented the idea of a proxy re-signature plan for wireless networks where the proxy can be reformed into the appointed affirmer whenever required [16]. This idea was beneficial for anonymous authentication. Progressively, Yang and team proposed their work on re-signature, which was proved to be advantageous even in clouds [17]. If any resource was missing, and if the cloud is authenticated, it was able to re-sign the data blocks. The statistical representation was on the sharable basis, and re-signature was planned to acquire such data which notably decreased the cost of validation by common affirmer. They also claimed that their scheme was helpful in privacy preservation.

In 2013, Jaime and team proposed a vehicular network, in which nodes were acting as mobile internet media for other nodes. It increases the access area for other vehicles as well as for the passengers. Each vehicle was able to join and leave the group at their will, supporting the *ad hoc* nature of the group [18]. A centralised group model for VANET often suffers from communication trust and user privacy issues, resulting in a heavy workload. Shao *et al.* used a decentralised group model by making use of GSs which made the authentication protocol secure, fast, efficient and independent [19]. Their extension plan also included making an algorithm having fixed batch verification computational overhead. In 2015, Bayat *et al.* proposed an authentication protocol for VANETs using batch verification in which they used the pseudo identity of the vehicles for communication so that identity of the vehicles may not be revealed [20]. The main aim of this algorithm is to avoid impersonation attack. However, other security parameters were not discussed in the paper. Following this, in 2016, a short GS scheme was proposed by Liqun Chen *et al.*, in which they devised a privacy-aware announcement scheme for VANETs [21]. Vehicles receiving a message decided to trust it based on the reputation score of sender vehicle. However, they did not focus on establishing the secured network channel and claimed that their network is reliable and robust even without it.

In 2016, Ubaidullah Rajput, Fizza Abbas, and Heekuck Oh presented an algorithm where the duration of usage of the two-degree ranking determined the order of pseudonyms [22]. Their scheme was technology-progressive towards making vehicles smarter by enabling them to connect with each other and form webs. It facilitated those with safety and traffic efficiency in general conditions. However, in hostile conditions, the individuality of the user might be revealed to certain authorities. They abolished the use of prolonged CRL because the hierarchy assured genuine and protected message to the receiver, thereby making it more vigorous and well organized. This was beneficial over the existing systems as there were not many dependencies upon certification authority (CA), revocation authority (RA), and road side unit (RSU). Moreover, it also possessed low computational and communication cost.

While taking existing progress into account, our project integrates all the favourable features into a single programme where each node communicates in a more reliable and vigorous network even in malicious conditions.

3 Proposed SAPSC algorithm

The proposed protocol covers four important approaches; signcryption, proxy re-encryption, proxy re-signature, and cloud computing. Members participating in this communication are listed in Table 1. There are two kinds of manager, MM and TM. MM is responsible for registering and revoking any member, and TM is responsible for finding the whereabouts and identity of a guilty vehicle [23]. Cloud is used to fasten the communication, but revealing identity to cloud is also prohibited because of security reasons [24]. Hence, one time credentials have been used to communicate instead of vehicle's personal IDs. Therefore, the protocol flows in the following manner: first a vehicle comes in the network and sends a RegReq to MM. MM updates group keys with the help of the cloud and registers the vehicle into the network. Group keys are transmitted to the vehicle in the RegRep. To start communication, vehicle signcrypts its messages with group keys and start broadcasting in the network. If it needs to find the route of a particular vehicle or entity, it asks cloud for the route using one time credential. In case of any accident or any suspicious activity, TM identifies the vehicle and reports it to MM. MM gets the real ID of the vehicle from the registration table and revokes it. Again, with the help of the cloud, MM updates group keys and distributes them in the network.

All these communication messages are duly authenticated at both ends. Both parties should trust the authentication model via a cross-validation scheme to guarantee credible interaction dynamically [25]. Here, MAC is created at each authentication step for all communicating entities to check the integrity and ensure the authenticated access. If calculated MAC is different from received MAC, the message is assumed to be altered in between and is discarded. The symbols used in the algorithm have been given in Table 2. Our protocol consists of the following steps.

Table 2 Parameters and descriptions

Notations	Description	Nature
ϕ	MM_2's public key	integer
π	MM_1's private key	integer
b, c	SSK shared between MM_1&MM_2	prime
VID_i	actual identity of vehicle i	alphanumeric
v_i	vehicle i's VID saved as v_i at MM's	integer
f_1	cryptographic function	function
$h, H, k_e h_f$	hash functions and keyed hash function	function
α, β	random numbers to calculate TP	prime
$s, r, \hat{c}$	signcrypted message	ciphertext
D_1, D_2, D_3	tracing parameters (TP)	integer
G_1, G_2, G_T	multiplicative cyclic groups	bilinear

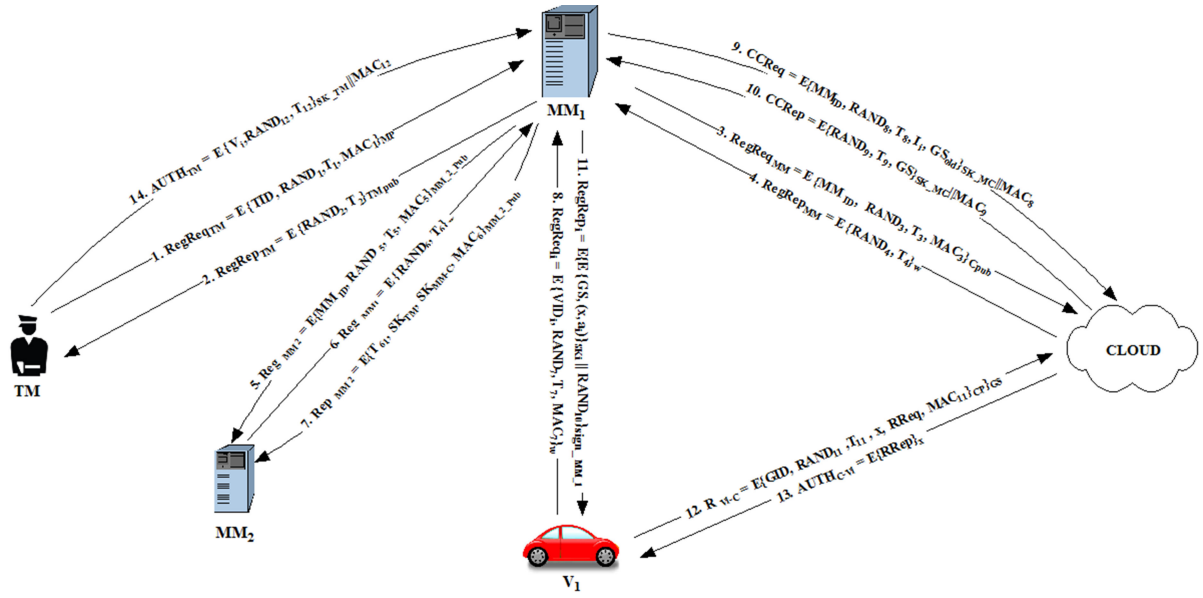


Fig. 3 *Authentication message exchange between entities*

3.1 Key generation

- Two multiplicative cyclic groups $\left(\hat{G}_1, \hat{G}_2\right)$ of the same prime order p with their generators g_1 and g_2 are taken. The bilinear map $\hat{e}$ can be computed as: $\hat{e}: \hat{G}_1 X \hat{G}_2 \rightarrow \hat{G}_T$, and $\hat{e}(g_1, g_2) = g \neq 1_{G_T}$.
- Keys related to communication and re-cryptography are generated by MM. The public keys which should be available to all group members. w is MM_1's public key. All these system parameters are enclosed in a parameter list called as Param and are transmitted

$$\text{Param} = (\hat{G}_1, \hat{G}_2, \hat{G}_T, g_1, g_2, \text{g}, \text{p}, \hat{e}, \text{H}, K_e H_f, \text{w}, \text{h}) \quad (1)$$

- If there is a demand for ReK and re-signature key (RsK), these are created by the MM

$$rk_{\text{MM}_1 - > \text{MM}_2} = (\phi - \pi)^{c^{-1}b}, \quad (2)$$

$$rs_{\text{MM}_2 - > \text{MM}_1} = (\pi - \phi)^{b^{-1}c}. \quad (3)$$

Below sections show the registration steps of various entities and generation of the shared secret key (SSK) between them Fig. 3 represents messages exchanged for registration and further authentication.

3.1.1 TM registration with MM: TM is responsible for declaring a vehicle guilty. If someone masquerades TM, non-guilty vehicles may be targeted and eventually revoked from the group. Hence, secure communication with authenticated TM is the first priority for MM. When a group is generated, TM registers itself with MM. A secret key is created and shared as follows:

Step 1: TM chooses a random variable RAND_1, hash it with its identity TID and timestamp T_1, generates a message digest MAC_1, and sends it with RegReq after encrypting it with the public key of the MM

$$\begin{aligned} \text{MAC}_1 &= \text{HASH}(\text{TID} \parallel \text{RAND}_1 \parallel T_1), \\ \text{RegReq}_{\text{TM}} &= E\{\text{TID} \parallel \text{RAND}_1 \parallel T_1 \parallel \text{MAC}_1\}_w. \end{aligned} \quad (4)$$

Step 2: MM extracts RAND_1 from $\text{RegReq}_{\text{TM}}$, chooses its own random number RAND_2 and generates a secret key, SK_{TM} by applying key derivation function (KDF) on RAND_T, TID, and RAND_2. This key is kept safe at MM's side. TM_{Pub} is the public key of TM which is used to encrypt RAND_2 as RegRep

$$\begin{aligned} \text{SK}_{\text{TM}} &= \text{KDF}(\text{RAND}_1 \parallel \text{RAND}_2 \parallel \text{TID}), \\ \text{RegRep}_{\text{TM}} &= E\{\text{RAND}_2 \parallel T_2\}_{\text{TM}_{\text{Pub}}}. \end{aligned} \quad (5)$$

After receiving RegRep, TM decrypts it using its private key to extract RAND_2. Then, TM computes SK_{TM} at its own side which is used for further communication between them.

3.1.2 MM registration with cloud: Clouds are needed by MM to generate GSs which ease the computation overhead at MM side. Since private keys belonging to the GS is highly confidential, it cannot be leaked at any cost. Hence, the cloud and MM must communicate properly through a secured and pre-registered

communication channel. The registration process between MM and cloud can be given as follows:

Step 1: MM chooses a random variable $RAND_3$, takes its identity MM_{ID} and hash it with current timestamp T_3 to generate a message digest MAC_3 which is sent in RegReq after encrypting it with the public key of the cloud

$$\begin{aligned} MAC_3 &= HASH(MM_{ID} \parallel RAND_3 \parallel T_3), \\ RegReq_{MM} &= E\{MM_{ID} \parallel RAND_3 \parallel T_3 \parallel MAC_3\}_{C_{Pub}}. \end{aligned} \tag{6}$$

Step 2: Based on random number $RAND_3$ received in $RegReq_{MM}$, cloud chooses another random variable $RAND_4$, generates a secret key, SK_{MM-C}, by applying KDF on $RAND_3$, $RAND_4$, and MM_{ID}

$$\begin{aligned} SK_{MM-C} &= KDF(RAND_3 \parallel RAND_4 \parallel MM_{ID}), \\ RegRep_{MM} &= E\{RAND_4 \parallel T_4\}_w. \end{aligned} \tag{7}$$

After receiving this reply from the cloud, MM calculates SK_{MM-C} by using $RAND_4$ and save it for further use.

3.1.3 MM_2 *registration with* MM_1*:* MM_2 needs to take over registration tasks if MM_1 is not present or overloaded. All messages incoming for MM_1 are either encrypted with w or with pre-shared secret keys. To share the pre-sharednec keys, MM_1 registers MM_2 at its side as follows:

Step 1: MM_1 chooses a random variable $RAND_5$, takes its identity MM_{ID} and hash it with a current timestamp T_5 to generate a message digest MAC_5 which is sent in Reg_{MM_2} after encrypting it with the public key of the MM_2

$$\begin{aligned} MAC_5 &= HASH(MM_{ID} \parallel RAND_5 \parallel T_5), \\ Reg_{MM_2} &= E\{MM_{ID} \parallel RAND_5 \parallel T_5 \parallel MAC_5\}_{MM_{2_{Pub}}}. \end{aligned} \tag{8}$$

Step 2: Based on random number $RAND_5$ received in Reg_{MM_2}, MM_2 chooses another random variable $RAND_6$, generates a secret key, SK_{MM}, by applying KDF on $RAND_5$, $RAND_6$, and MM_{ID}

$$\begin{aligned} SK_{MM} &= KDF(RAND_5 \parallel RAND_6 \parallel MM_{ID}), \\ Reg_{MM_1} &= E\{RAND_6 \parallel T_6\}_w. \end{aligned} \tag{9}$$

After receiving this reply from MM_2, MM_1 calculates SK_{MM} by using $RAND_6$ and sends all shared secret keys after encrypting them with SK_{MM}

$$\begin{aligned} MAC_6 &= HASH(T_{61} \parallel SK_{TM} \parallel SK_{MM-C}), \\ Rep_{MM_2} &= E\{T_{61} \parallel SK_{TM} \parallel SK_{MM-C} \parallel MAC_6\}_{MM_{2_{Pub}}}. \end{aligned} \tag{10}$$

3.2 Vehicle registration in the group

3.2.1 RegReq by vehicle: Vehicles and MM are the two most important entities of a VANET, working in a group. Vehicles request MM for registering themselves in a group. MM replies with the GS which will be the new identity of the vehicle in the network, and a secret key SK_i which is used for further authentication between vehicle and MM. The entire registration process takes place as follows:

Step 1: Vehicle chooses a random variable $RAND_7$, does a hash operation on it with the current timestamp, T_7 and its identity, VID_i to generate a message digest MAC_7 which is sent in RegReq after encrypting it with the public key of the MM

$$MAC_7 = HASH(VID_i \parallel RAND_7 \parallel T_7), \tag{11}$$

$$RegReq_i = E\{VID_i \parallel RAND_7 \parallel T_7 \parallel MAC_7\}_w. \tag{12}$$

3.2.2 Proxy re-encryption: If MM_1 is available, all RegReqs are entertained by itself. Otherwise proxy re-encrypts the incoming message to MM_2's keys using $rk_{MM_1 \to MM_2}$ calculated in (2) as follows:

$$RegReq'_i = \hat{e}\left(RegReq, g^{rk_{MM_1}}\right), \tag{13}$$

RegReq′ is a new ciphertext which is encrypted with MM_2's public keys as below

$$RegReq'_i = E\{VID_i \parallel RAND_7 \parallel T_7 \parallel MAC_7\}_{MM_{2_{Pub}}}. \tag{14}$$

Now, this RegReq can be decrypted using MM_2's private key. However, for our convenience, we consider MM as symbolising any MM, which is available.

- After decrypting RegReq, MM saves VID_i of the i_{th} vehicle as v_i and calculates intermediate ID I_i for it as given in (15)

$$I_i = H(\pi, v_i). \tag{15}$$

The complex calculations are done by cloud instead of MM. MM sends computing request to the cloud using the authenticated secret keys created earlier.

3.2.3 Authentication by cloud:

Step 1: MM sends the computing request to the cloud. It creates MAC_8 by applying the standard cryptographic function f_1 on the message encrypted with SK_{MC}. This request consists of an old GS, GS_{old} and I_i. Please note that actual identity of the vehicle if never disclosed even to the cloud

$$MAC_8 = f_1\{MM_{ID} \parallel RAND_8 \parallel T_8 \parallel I_i \parallel GS_{old}\}_{SK_{MC}}, \tag{16}$$

$$CCReq = E\{MM_{ID} \parallel RAND_8 \parallel T_8 \parallel I_i \parallel GS_{old}\}_{SK_{MC}} \parallel MAC_8. \tag{17}$$

Step 2: Cloud creates $MAC_{8'}$ using the same function as used by MM, i.e. f_1. If $MAC_8 = MAC_{8'}$, it means MM has used the same secret key as shared with cloud, and hence MM is authenticated.

3.2.4 Computing new group keys: Cloud extracts I_i and GS_{old} from cloud computing request (CCReq) and computes V_i as given in (18)

$$V_i = f(ACC, I_i^{-1}) = g_1^{pI_i^{-1} \bmod \phi(N)} \bmod N. \tag{18}$$

This new V_i is merged with the old signature to generate a new GS.

3.2.5 Authentication by MM:

Step 1: Cloud generates $RAND_9$ and takes it with a new GS to create another MAC_9 using f_1 again, and send to MM

$$MAC_9 = f_1\{RAND_9 \parallel T_9 \parallel GS\}_{SK_{MC}}, \tag{19}$$

$$CCRep = E\{RAND_9 \parallel T_9 \parallel GS\}_{SK_{MC}} \parallel MAC_9. \tag{20}$$

Step 2: MM calculates $MAC_{9'}$ using f_1 and matches with MAC_9. If positive, the sender is authenticated as cloud using the comprehensive logic mentioned in previous sections.

3.2.6 RegRep by MM:

- MM chooses a random x and calculates: $a_i = v_i^x$.
- It replies with new GS and one-time usable credential pair as (GS, x, a_i).

To prepare the RegRep, MM chooses its own random number RAND_{10} and generates a secret key, SK_i by applying KDF on RAND_{10}, RAND_7, VID_i, (x, a_i) and T_7. This is used to encrypt the RegRep and vehicle can decrypt the message by calculating SK_i again at its own side using the parameters available with it. MM also sends a one-time credential pair to the vehicle which will be used for communicating with the cloud

$$\text{SK}_i = \text{KDF}(\text{RAND}_{10} \parallel \text{RAND}_7 \parallel \text{VID}_i), \tag{21}$$

$$\text{RegRep}_i = E\{E\{\text{GS} \parallel (x, a_i)\}_{\text{SK}_i} \parallel \text{RAND}_{10}\}_{\sigma_{\text{MM}_2}}. \tag{22}$$

3.2.7 Proxy re-signature: While replying to the same message, MM_2 signs with its own key which is later changed into MM_1 signature by proxy using RsK $rs_{\text{MM}_2 \to \text{MM}_1}$, calculated in (3). Hence, the vehicle does not notice any change in manager, which avoids unnecessary confusion

$$\sigma_{\text{MM}1} = \hat{e}\left(\sigma_{\text{MM}2}, g^{rs_{C \to B}}\right), \tag{23}$$

$$\text{RegRep}_i = E\{E\{\text{GS} \parallel (x, a_i)\}_{\text{SK}_i} \parallel \text{RAND}_{10}\}_{\sigma_{\text{MM}_1}}. \tag{24}$$

3.3 Secure navigation with cloud

Finding routes in VANET is a necessary overhead for vehicles, which also act as routers in the network. When a vehicle tries to find out a particular route or traffic scenario of a particular area, it can either find it out by vehicle-to-vehicle/vehicle-to-infrastructure communication or can send a request to cloud. Predicting the distance or number of hopes required, vehicle requests to the cloud if it presumes the cloud as a better option. To navigate securely with the cloud MM has already sent a secret key pair (x, a_i) in the registration phase. It also notifies cloud about this credential, and hence cloud knows that this x can be sent by an authenticated member only. Hence, the vehicle does not need to show its identity.

The basic procedure for this navigation includes the following steps:

i. *Routing request*: Vehicle sends $(\text{RReq}, x)_{\text{Gsign}}$ to cloud.
ii. Cloud confirms the validity of member from GS, and hence the vehicle's actual identity is hidden.
iii. x sent along with Gsign can be used only once, which means the same credentials cannot be used twice.
iv. Cloud computes the optimised route and replies with RRep_x, where routine reply (RRep) is encrypted with x.

3.3.1 Authentication between cloud and vehicle:

Step 1: To communicate with the cloud, the vehicle needs to authenticate itself. Since, we are preserving the privacy of members in a group, the identity of the vehicle cannot be revealed even to cloud. Hence, GS is used by vehicle instead of a personal signature. Vehicle generates RAND_{11}, and the message digest is calculated as

$$\text{MAC}_{11} = f_1\{\text{RAND}_{11} \parallel T_{11} \parallel \text{GID}\}_{\text{CP}}, \tag{25}$$

$$R_{V_iC} = E\{\{\text{GID} \parallel \text{RAND}_{11} \parallel T_{11} \parallel x \parallel \text{RReq} \parallel \text{MAC}_{11}\}_{\text{CP}}\}_{\text{GS}}. \tag{26}$$

Step 2: Cloud authenticates vehicle by its GS and matching group identification number (GID) sent along in the message. Then at the time of replying, it encrypts the message with x, sent along in R_{V_iC}. Since only the cloud can decrypt this authentication code (encrypted with its public key) and find x, the cloud gets authenticated by vehicle automatically

$$R_{\text{CV}_i} = E\{\text{RRep}\}_x. \tag{27}$$

Vehicle decrypts R_{CV_i} using x's paired key, a_i, to extract RRep.

3.4 SignCryption

After receiving the GS from MM and routes from the cloud, the vehicle starts transmitting in the network. It may broadcast BSM in the network or send personal message depending upon the need. In this step, instead of encrypting the message and then signing it, it uses signcryption. To signcrypt the message, at first a K is calculated by hashing GS (not a personal signature) with a public key of the recipient. Suppose any node is trying to send a message to MM again, then K will be calculated using

$$K = H(\text{GS}, w) \text{mod} N. \tag{28}$$

K gets divided into k_1 and k_2 and tracing parameters are calculated

$$D_1 = k_1^{\alpha};\ D_2 = k_2^{\beta};\ D_3 = (\hat{e}(V_i, h)^{\alpha+\beta}) \bmod N, \tag{29}$$

$$\delta_1 = v_i\alpha\ ;\ \delta_2 = v_i\beta. \tag{30}$$

Challenger $\complement$ is used to validating m at the receiver side. k_1 is encrypted with m to produce $\hat{c}$ and k_2 is hashed with $\hat{c}$ using K_eH_f

$$\complement = H(m, D_1, D_2, D_3, \delta_1, \delta_2), \tag{31}$$

$$\hat{c} = E(k_1(m)), \tag{32}$$

$$r = K_{\text{e}}H_{\text{f}}(k_2(\hat{c})), \tag{33}$$

$$s = w + (r * V_i) \bmod N, \tag{34}$$

$$d_\alpha = (D_1 + \hat{c}\alpha) \bmod \delta_1, \tag{35}$$

$$d_\beta = (D_2 + \hat{c}\beta) \bmod \delta_2, \tag{36}$$

$$\sigma = (\hat{c}, D_1, D_2, D_3, d_\alpha, d_\beta, \complement) \tag{37}$$

Then the signed message, σ, $(\hat{c}, s, r, \sigma)$ is sent in the network.

3.5 Verification and unsigncryption

The receiver collects the message and verifies whether

- It is from a valid group member.
- The message is received in the valid time domain.
- Then it calculates $\complement'$ and matches it with the received $\complement$.

If all three conditions are verified, it means the message is valid. Then the receiver retrieves $(\hat{c}, s, r)$ from it to calculate K

$$K = H((g^s * (\text{gpk})^r), \pi') \bmod p, \tag{38}$$

where gpk is the group public key available with all group member and π' is the private key of the receiver. Now, K gets divided into k_1 and k_2 to calculate r' and eventually, message m is extracted

$$r' = K_{\text{e}}H_{\text{f}}(k_2(\hat{c})), \tag{39}$$

$$m = D(k_1(\hat{c})). \tag{40}$$

3.6 Membership tracing and revocation

TM always monitors the activities of a vehicle belonging to its group. Nodes are traced, and their identities are extracted in case of an accident or any other faulty behaviour. If found guilty, this identity is sent to MM. MM revokes the guilty nodes from the group and updates the GS.

To trace the identity of the member, TM calculates V_i from the GS

$$V_i = \hat{e}\left(D_3, \left(D_2^{\theta_1} \cdot D_3^{\theta_2}\right)\right) \bmod N. \tag{41}$$

This V_i is sent to MM as VID_i using the shared secured key SK_{TM} generated above.

3.6.1 Authentication by MM: Step 1: TM calculates another MAC, MAC_{12} by applying the standard cryptographic function f_1 and encrypting it with SKTM

$$\text{MAC}_{12} = f_1\{V_i \parallel \text{RAND}_{12} \parallel T_{12}\}_{\text{SK}_{\text{TM}}}, \tag{42}$$

$$\text{AUTH}_{\text{TM}} = E\{V_i \parallel \text{RAND}_{12} \parallel T_{12}\}_{\text{SK}_{\text{TM}}} \parallel \text{MAC}_{12}\,. \tag{43}$$

Step 2: MM decrypts the received message and creates MAC'_{12} using the parameters extracted using f_1. If $\text{MAC}_{12} = \text{MAC}'_{12}$, it means TM has an encrypted message with the SSK, proving its authenticity.

After extracting V_i, MM revokes member i from the group using dynamic reverse accumulator by the following algorithm (Fig. 4):

After the member has been revoked from the group or any member leaves the group voluntarily, MM updates the registration table and revocation table to keep track of all members and ex-members of the group.

4 Formal verification and security analysis of the proposed protocol

4.1 Formal verification using automated validation of internet security protocols and applications (AVISPA) tool

The main objective of the proposed protocol is to add authentication feature in already existing SignRecryption and re-signature techniques. This section gives the formal verification for the security of our protocol using on-the-fly-model-checker (OFMC) and constraint-logic-based attack searcher (CL-AtSe) model checkers of AVISPA tool [26].

4.1.1 Simulating SAPSC: For the simulation of our protocol, we have taken a scenario where communication between managers, vehicle, proxy, and cloud is taking place. Fig. 5*a* gives the possibility of intruder intervention during message transmission. Corresponding to those events, actual messages communicating in the network are shown in Fig. 4b, the result of which is SAFE as shown in Fig. 6. We can also observe from intruder knowledge as given in Fig. 7 that intruder is getting messages only in the encrypted forms, and it is not able to decrypt any useful information from it.

Our goals have been described in Fig. 8 which has been written in high level protocol specification language:

The goal of this simulation is defined in two phases, secrecy of keys, and authentication of communicated messages. The reason to select these parameters is to ensure that the proposed protocol fulfils authentication, confidentiality, and integrity features. The security goal 'sec_P_M1' is defined on ReK by a predicate secret(Re,sec_P_M1,M1,P) which means ReK is to be kept a secret between MM_1 and proxy. Similarly, RsKs must be kept a secret between MM_2 and proxy, which is defined by secret(Rs,sec_P_M2,M2,P). Regreq and Regrep are the secret messages transferred between MM_1 and vehicle, and respective goals are defined by secret(Regreq1,sec_A_M1,M1,A) and secret(Regrep1,sec_A_M1,M1,A). Secrecy goals are tested under intruder simulation to verify the confidentiality and integrity of the transmitted messages.

Authentication on messages is defined by two facts which are known as witness and request (wrequest in case of weak authentication which does not concern about the replay messages). These facts verify the session, state and freshness parameter of the communicating partner. In RegReq, when M2 receives MAC_1', it wants to be sure that this was indeed created by vehicle A. For this, we write a request predicate as 'request(M2,A mac1_auth,MAC_1')', which means that M2 accepts the value MAC_1′ and now it relies that A exists and agrees on MAC_1′ with M2. The matching witness predicate, witness(A, M2, mac1_auth,MAC_1′), must be added at the sender side, which means that A wants be a partner with M2 and agrees on the value

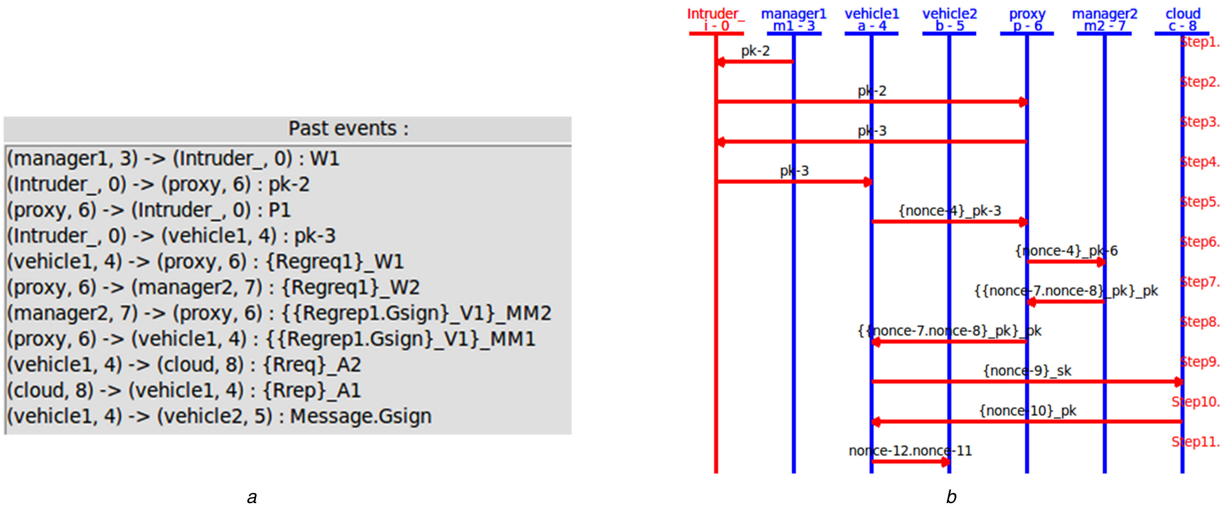
Data: Input$(ACC, V_i, Gsign)$
Result: Output(ACC')
for $i = 1$ **to** m **do**

$$ACC' = ACC^{V_i{}^{-1} mod \phi n} mod n$$

delete V_i **from** Y
insert V_i **into** Y'

end
return(ACC',V_i)

Fig. 4 *Algorithm 1: Dynamic reverse accumulator revocation*



a *b*

Fig. 5 *Simulating our protocol with intruder intervention*
(a) Events took place in our simulation, ***(b)*** Intruder simulation window

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/SignRecryptionInCloud.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 2.10s
  visitedNodes: 1658 nodes
  depth: 10 plies
```

a

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/SignRecryptionInCloud.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 695 states
  Reachable  : 160 states
  Translation: 0.04 seconds
  Computation: 0.00 seconds
```

b

Fig. 6 *Simulation result by AVISPA tool*
(a) OFMC simulation result, ***(b)*** ATSE simulation result

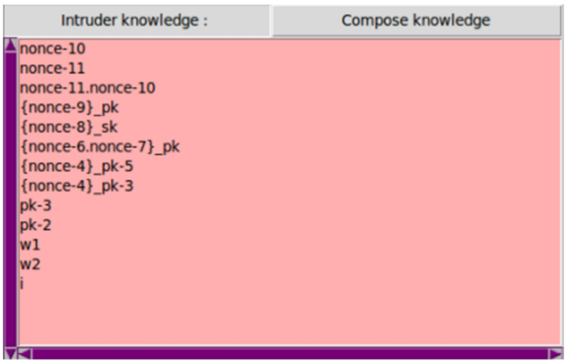


Fig. 7 *Intruder knowledge*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 Goal
   secrecy_of sec_P_M1, sec_P_M2,
              sec_A_M1, sec_A_M2

   authentication_on mac1_auth, mac2_auth,
                     mac3_auth, mac4_auth
 end goal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 8 *Goal*

mac1_auth. Similarly, witness and request parameters have been added for the protocol IDs mac2_auth, mac3_auth, and mac4_auth, where 'mac2_auth' is defined between A and M2 for Regrep, 'mac3_auth' is defined between A and C for CReq, and 'mac4_auth' is defined between A and C for CRep.

The goal of the AVISPA code gives the ultimate goal for MM_1, MM_2, and vehicles to secure their keys from disclosing and securing their communication. The intermediate procedures are given in past events in Fig. 5*a*. The protocol has been tested by two AVISPA tools; OFMC and ATSE, the result of which has been shown in Figs. 6*a* and *b*. Both diagrams show that our protocol is SAFE from various attacks.

4.2 Formal proof using Burrows–Abadi–Needham (BAN) logic

This section proves the security of our protocol using BAN logic. At first, communication messages given in Section 1.1 are converted into a standardised form, and after that, it has been verified by BAN logic to check whether our protocol is free from attacks.

4.2.1 BAN logic rules:

- $P| \equiv X$: P believes in trueness of X in current time and can repeat it in its messages.
- $P| \Rightarrow X$: P has jurisdiction over X, and P can be trusted for trueness of X.
- $P| \sim X$: P once said and believed that X was true. However, it may not believe the same in current time.
- $P \triangleleft X$: P sees X and can repeat it in its own messages.
- $\#X$: X is fresh.
- $key(K, P \leftrightarrow Q)$: P shares a secret key with Q which is known only to them or to any other third party trusted by both.
- $\overset{K}{\rightarrow} P$ K and K^{-1} are the public key-private key pair of P.
- $P \overset{X}{\Leftrightarrow} Q$: X is a secret known only to P, Q or a trusted third party.
- $(X)_K$: X has been encrypted with the key, K.

The below rules can be implied from the above rules:

- $P\Big| \equiv (P \overset{K}{\leftrightarrow} Q) \wedge P \triangleleft (X)_K$: implies: P believes that Q once said X.
- $P| \equiv (Q| \Rightarrow X) \wedge P| \equiv (Q| \equiv X)$: implies: P believes X is true.
- $P| \equiv (Q| \sim X) \wedge P| \equiv Q| \equiv \#X$: implies: P believes that Q believes in truthiness of X.

The communication messages with their preliminary conditions can be given as follows:

i. RegReq($A \Rightarrow MM_1$): M_1: It should be fresh and must not be revealed to a third party other than A and MM_1.
ii. RegReq(Proxy $\Rightarrow MM_2$): M_2: It should be fresh and must not be revealed to a third party other than A and MM_2.
iii. CCReq($MM_1 \Rightarrow$ Cloud) or CC($MM_2 \Rightarrow$ Cloud): M_3: It should be fresh and must not be revealed to a third party other than MM_1 and Cloud.
iv. CCRep($MM_1 \Leftarrow$ Cloud) or CC($MM_2 \Leftarrow$ Cloud): M_4: It should be fresh and must not be revealed to a third party other than MM_1 and Cloud.
v. RegRep($A \Leftarrow MM_2$): M_5: It should be fresh and must not be revealed to a third party other than MM_2 and A.
vi. RegRep($A \Leftarrow$ Proxy): M_6: It should be fresh and must not be revealed to a third party other than MM_2 and A.
vii. RReq($A \Rightarrow$ Cloud): M_7: It should be fresh and must not be revealed to a third party other than A and Cloud.
viii. RRep($A \Leftarrow$ Cloud): M_8: It should be fresh and must not be revealed to a third party other than A and Cloud.

ix. BSM($A \Rightarrow B$) or BSM($A \Rightarrow$ TM): M_9: This message must be signed with GS and traceable by TM.
x. V_{ID}(TM $\Rightarrow$ MM_1): $M_1$0: It should be fresh and must not be revealed to a third party other than MM_1 and TM.

The standardised form of the above messages can be given as

i. M_1: $A \Rightarrow MM_1$: Na, A_5(Na, VID_i, $RAND_7$, MAC_7)$_{PK_{MM_1}}$
ii. M_2: $P \Rightarrow MM_2$: Na, A_5(Na, VID_i, $RAND_7$, MAC_7)$_{PK_{MM_2}}$.
iii. M_3: $MM_2 \Rightarrow C$: Nb, (A_5(Nb, MM_{ID}, GS_{old}, $RAND_8$)$_{SK_{MC}}$) $\cdot$ MAC_8
iv. M_4: $MM_2 \Leftarrow C$: Nc, (A_5(Nc, GS, $RAND_9$)$_{SK_{MC}}$) $\cdot$ MAC_9
v. M_5: $A \Leftarrow MM_2$: Nd, A_3(A_5(Nd, GS, (x, a_i))$_{RAND_{10}}$)$_{\sigma_{MM_2}}$
vi. M_6: $A \Leftarrow P$: Nd, A_3(A_5(Nd, GS, (x, a_i))$_{RAND_{10}}$)$_{\sigma_{MM_1}}$
vii M_7: $A \Rightarrow C$: Ne, A_3(A_5(Ne, GID, x, RReq, $RAND_{11}$)$_{CP}$)$_{GS}$ $\cdot$ MAC_{11}
vii M_8: $A \Leftarrow$ Cloud: Nf, A_3(A_5(Nf, RRep)$_x$
i.
ix. M_9: $A \Rightarrow B$: Ng, A_3(Ng, BSM)$_{GS}$
x. M_{10}: TM $\Rightarrow$ MM_1: Nh, A_3(Nh, V_i)$_{\sigma_{TM}}$

Before progressing further, we have assumed that the following properties are true:

i. M_1 and proxy have agreed upon to share ReK.
   - MM_1 securely sends ReK to the proxy
   $$MM_1 \mid \sim rk. \tag{44}$$
   - Proxy believes that ReK sent by MM_1 is fresh
   $$\frac{\text{Proxy} \lhd rk \wedge \text{Proxy} \mid \equiv \#rk}{\text{Proxy} \mid \equiv MM_1 \mid \equiv rk}. \tag{45}$$
   - MM_1 believes that this key has been known only to proxy and itself.
   $$MM_1 \mid \equiv \left(MM_1 \overset{rk}{\leftrightarrow} \text{Proxy}\right). \tag{46}$$
   - Proxy believes that this key has been known only to MM_1 and itself
   $$\text{Proxy} \mid \equiv (\text{Proxy} \overset{rk}{\leftrightarrow} MM_1). \tag{47}$$
ii. MM_2 and proxy shares the RsK
   - MM_2 securely sends RsK to the proxy
   $$MM_2 \mid \sim rs. \tag{48}$$
   - Proxy believes that RsK sent by MM_2 is fresh
   $$\frac{\text{Proxy} \lhd rs \wedge \text{Proxy} \mid \equiv \#rs}{\text{Proxy} \mid \equiv MM_2 \mid \equiv rs}. \tag{49}$$
   - MM_2 believes that this key has been known only to proxy and itself
   $$MM_2 \mid \equiv \left(MM_2 \overset{rs}{\leftrightarrow} \text{Proxy}\right). \tag{50}$$
   - Proxy believes that this key has been known only to MM_2 and itself
   $$\text{Proxy} \mid \equiv \left(\text{Proxy} \overset{rs}{\leftrightarrow} MM_2\right). \tag{51}$$
iii. All authorities assume that the pre-shared information and the communication channels are safe
   $$MM_1 \mid \equiv \left(MM_1 \overset{X}{\leftrightarrow} P\right) \wedge P \mid \equiv \left(MM_1 \overset{X}{\leftrightarrow} P\right), \tag{52}$$
   $$MM_1 \mid \equiv \left(MM_1 \overset{Y}{\leftrightarrow} MM_2\right) \wedge MM_2 \mid \equiv \left(MM_1 \overset{Y}{\leftrightarrow} MM_2\right), \tag{53}$$
   $$MM_1 \mid \equiv \left(MM_1 \overset{Z}{\leftrightarrow} \text{Cloud}\right) \wedge \text{Cloud} \mid \equiv \left(MM_1 \overset{Z}{\leftrightarrow} \text{Cloud}\right). \tag{54}$$

4.2.2 Applying BAN logic: BAN logic can be applied to find out the loophole present in the communication and to check whether the protocol is safe from attacks. For this, below formulae are used which are inferred from the above rules:

i. MM_2 believes that VID_i is secret ID of A, hence it also believes that Na, A_5(Na, VID_i, $RAND_7$, MAC_7)$_{PK_{MM_2}}$ is sent by A, hence verifying the origin of the message as below
   $$\begin{aligned} &MM_2 \mid \triangleright \text{Na}, A_5(\text{Na}, VID_i, RAND_7, MAC_7)_{PK_{MM_2}}, \\ &MM_2 \mid \triangleright VID_i \wedge MM_2 \mid \equiv VID_i \text{ is secret ID of } A, \\ &MM_2 \mid \equiv A \mid \sim \text{Na}, A_5(\text{Na}, VID_i, RAND_7, MAC_7)_{PK_{MM}} \end{aligned} \tag{55}$$
   - *Conclusion*: Origin of RegReq is authenticated over VID_i which is known to A only. Since this message is encrypted with PK_{MM_2}, MM_2 believes that this message is sent by A, and has not been modified in between. Hence, integrity and authenticity are checked.
ii. A believes that $sign_{MM_1}$ is signature of MM_1, hence it also believes that Nd, A_3(A_5(Nd, GS, (x, a_i))$_{RAND_{10}}$)$_{\sigma_{MM_2}}$ is sent by MM_1, hence verifying the origin of the message as below
   $$A \mid \equiv MM_1 \mid \sim \overset{\text{sign}}{\rightarrow} MM_1 \wedge \tag{56}$$
   (see (57))
   - *Conclusion*: Authentication and integrity are verified as above.
iii. For CCReq and cloud computing reply (CCRep), cloud sees MAC_8 and MM_1 sees MAC_9, which is created using SK_{MC} shared between them.

   (a) For CCReq
   $$C \mid \equiv k(SK_{MC}, C \leftrightarrow MM_1) \wedge \tag{58}$$
   $$\begin{aligned} &C \lhd \text{Nb}, \left(A_5(\text{Nb}, MM_{ID}, GS_{old}, RAND_8)_{SK_{MC}}\right), \\ &C \mid \equiv MM_1 \mid \sim \left(A_5(\text{Nb}, MM_{ID}, GS_{old}, RAND_8)_{SK_{MC}}\right) \wedge C \mid \equiv \#\text{Nb}, \\ &C \mid \equiv MM_1 \mid \equiv \left(A_5(\text{Nb}, MM_{ID}, GS_{old}, RAND_8)_{SK_{MC}}\right). \end{aligned} \tag{59}$$

   (b) For CCRep

$$\frac{A \lhd \text{Nd}, A_3\left(A_5(\text{Nd}, \text{GS}, (x, a_i))_{RAND_{10}}\right)_{\sigma_{MM_2}}}{A \mid \equiv MM_1 \mid \sim [\text{Nd}, \text{Nd}, A_3(A_5(\text{Nd}, \text{GS}, (x, a_i))_{RAND_{10}})_{\sigma_{MM_2}}}. \tag{57}$$

$$\frac{\begin{array}{c}\mathrm{MM}_2 \mid \equiv k(\mathrm{SK}_{\mathrm{MC}}, C \leftrightarrow \mathrm{MM}_1 \parallel \mathrm{MM}_2) \wedge \\ \mathrm{MM}_2 \lhd (A_5(\mathrm{Nc}, \mathrm{GS}, \mathrm{RAND}_9)_{\mathrm{SK}_{\mathrm{MC}}}) \cdot MAC_9\end{array}}{\begin{array}{c}\mathrm{MM}_2 \mid \equiv C \mid \sim (A_5(\mathrm{Nc}, \mathrm{GS}, \mathrm{RAND}_9)_{\mathrm{SK}_{\mathrm{MC}}}) \cdot \mathrm{MAC}_9 \\ \wedge\ \mathrm{MM}_2 \mid \equiv \#Nc\end{array}}, \quad \frac{}{\mathrm{MM}_2 \mid \equiv C \mid \equiv (A_5(\mathrm{Nc}, \mathrm{GS}, \mathrm{RAND}_9)_{\mathrm{SK}_{\mathrm{MC}}}) \cdot \mathrm{MAC}_9}. \tag{60}$$

• *Conclusion*: Incoming message encrypted with $\mathrm{SK}_{\mathrm{MC}}$ shows that it could be sent only by the member which has the secret key $\mathrm{SK}_{\mathrm{MC}}$. Since this key is available only with cloud and MM, the message could not be modified in the middle, hence proving the origin, confidentiality, and integrity of the received message.

iv. Proxy believes that Na is fresh, and it also believes that A once said $\mathrm{Na}, \mathrm{A}_5(\mathrm{Na}, \mathrm{VID}_i, \mathrm{RAND}_7, \mathrm{MAC}_7)_{\mathrm{PK}_{\mathrm{MM}_1}}$, hence Proxy believes A believes $\mathrm{A}_5(\mathrm{Na}, \mathrm{VID}_i, \mathrm{RAND}_7, \mathrm{MAC}_7)_{\mathrm{PK}_{\mathrm{MM}_1}}$, hence proving the freshness of the first message M_1

$$P \mid \equiv \#\mathrm{Na} \wedge \tag{61}$$

(see (62))

- *Conclusion*: Proxy avoids replay attack by confirming the freshness of M_1.

v. MM_2 believes that Na is fresh, and it also believes that A once said $\mathrm{Na}, \mathrm{A}_5(\mathrm{Na}, \mathrm{VID}_i, \mathrm{RAND}_7, \mathrm{MAC}_7)_{\mathrm{PK}_{\mathrm{MM}_2}}$. Therefore, it can be concluded that MM_2 believes that A believes $\mathrm{Na}, A_5(\mathrm{Na}, \mathrm{VID}_i, \mathrm{RAND}_7, \mathrm{MAC}_7)_{\mathrm{PK}_{\mathrm{MM}_2}}$, hence proving the freshness of second message M_2

$$\mathrm{MM}_2 \mid \equiv \#\mathrm{Na} \wedge \tag{63}$$

(see (64))

- *Conclusion*: MM_2 avoids replay attack by confirming the freshness of M_2.

vi. Cloud believes that Nb is fresh, and it also believes that MM once said $\left(A_5(\mathrm{Nb}, \mathrm{MM}_{\mathrm{ID}}, \mathrm{GS}_{\mathrm{old}}, \mathrm{RAND}_8)_{\mathrm{SK}_{\mathrm{MC}}}\right)$, hence cloud believes that MM believes $\mathrm{Nb}, \left(A_5(\mathrm{Nb}, \mathrm{MM}_{\mathrm{ID}}, \mathrm{GS}_{\mathrm{old}}, \mathrm{RAND}_8)_{\mathrm{SK}_{\mathrm{MC}}}\right)$, hence proving the freshness of third message M_3

$$C \mid \equiv \#\mathrm{Nb} \wedge \tag{65}$$

(see (66))

- *Conclusion*: Cloud avoids replay attack by confirming the freshness of M_3. Similarly, it can be proved for the freshness of the rest of the messages as well. Hence, we can say by above formulae and inferences that

$$\begin{array}{l} A \mid \equiv \#\mathrm{Na} \wedge \mathrm{MM}_2 \mid \equiv \#\mathrm{Na}, \\ \mathrm{MM}_2 \mid \equiv \#\mathrm{Nb} \wedge \mathrm{Cloud} \mid \equiv \#\mathrm{Nb}, \\ \mathrm{Cloud} \mid \equiv \#\mathrm{Nc} \wedge \mathrm{MM}_2 \mid \equiv \#\mathrm{Nc}, \\ \mathrm{MM}_2 \mid \equiv \#\mathrm{Nd} \wedge A \mid \equiv \#\mathrm{Nd}, \\ A \mid \equiv \#\mathrm{Ne} \wedge \mathrm{Cloud} \mid \equiv \#\mathrm{Ne}, \\ \mathrm{Cloud} \mid \equiv \#\mathrm{Nf} \wedge A \mid \equiv \#\mathrm{Nf}, \end{array} \tag{67}$$

and finally,

$$\mathrm{TM} \mid \equiv \#\mathrm{Nh} \wedge \mathrm{MM}_1 \mid \equiv \#\mathrm{Nh}. \tag{68}$$

All the valid and fresh messages are having fresh nonces, which is checked and believed by the senders and receivers both. Therefore, a replay attack is not possible in any case.

From all the equations, logical derivations and conclusion given above, it is verified by BAN logic that our protocol is secured in terms of integrity, freshness, correctness, confidentiality, authenticity and other security tests. It proves that our protocol is secured against security attacks.

4.3 Security analysis

This section represents the types of attacks possible on our network and how our protocol prevents them

- *Impersonation attack*: Being an infrastructure-less network, there is a high possibility of impersonation of MM, vehicle or cloud, but we have created MAC before starting any communication on the basis of pre-shared keys, which is re-created at receiver's side to verify the authenticity of any member.
- *Attack on identity*: The identity of the vehicle can be tracked by other vehicles, cloud or any third party entity if the communicated message gives any information about that particular sender. However, the message is signed by the signature of the group instead of that particular sender. Even while requesting for the route, the routing request is encrypted with a one-time credential, and it is not possible for an intruder to track the sender's identity based on routing requests as well.
- *Transitive key derivation attack*: While deriving ReK and RsK, if the keys are multiplicative derivative, keys can be transitively derived from one another. However, in our algorithm, keys are not directly multiplied, but secret keys are exponentiated after being differenced. Hence, it is not possible to derive a third key, if any two keys are given.
- *Sybil attack*: If any intruder enters the network and creates pseudonymous identities, it can be hazardous for secure communication. The intruder can also create many other types of attack within the system. Hence, in our protocol, to prevent this attack, it is made compulsory for the vehicles to register themselves in the network with their vehicle ID, which is unique by nature. These IDs are stored in the membership table, and only unique IDs are allowed there. Hence, a vehicle cannot register twice for itself and Sybil attack is prevented.

$$\frac{P \mid \equiv A \mid \sim \mathrm{Na}, A_5(\mathrm{Na}, \mathrm{VID}_i, \mathrm{RAND}_7, \mathrm{MAC}_7)_{\mathrm{PK}_{\mathrm{MM}_1}}}{P \mid \equiv A \mid \equiv \mathrm{Na}, A_5(\mathrm{Na}, \mathrm{VID}_i, \mathrm{RAND}_7, \mathrm{MAC}_7)_{\mathrm{PK}_{\mathrm{MM}_1}}}. \tag{62}$$

$$\frac{\mathrm{MM}_2 \mid \equiv A \mid \sim \mathrm{Na}, A_5(\mathrm{Na}, \mathrm{VID}_i, \mathrm{RAND}_7, \mathrm{MAC}_7)_{PK_{\mathrm{MM}_2}}}{\mathrm{MM}_2 \mid \equiv A \mid \equiv \mathrm{Na}, A_5(\mathrm{Na}, \mathrm{VID}_i, \mathrm{RAND}_7, \mathrm{MAC}_7)_{\mathrm{PK}_{\mathrm{MM}_2}}}. \tag{64}$$

$$\frac{C \mid \equiv \mathrm{MM} \mid \sim \mathrm{Nb}, \left(A_5(\mathrm{Nb}, \mathrm{MM}_{\mathrm{ID}}, \mathrm{GS}_{\mathrm{old}}, \mathrm{RAND}_8)_{\mathrm{SK}_{\mathrm{MC}}}\right)}{C \mid \equiv \mathrm{MM} \mid \equiv \mathrm{Nb}, \left(A_5(\mathrm{Nb}, \mathrm{MM}_{\mathrm{ID}}, \mathrm{GS}_{\mathrm{old}}, \mathrm{RAND}_8)_{\mathrm{SK}_{\mathrm{MC}}}\right)}. \tag{66}$$

- *Unauthorised backward traceability*: Messages are never signed with personal IDs, and GS can only be traced by an authorised entity TM, therefore restricting any unauthorised entity to track down a vehicle.
- *Channel establishment by an intruder*: After vehicles register themselves in the group, those are assigned with a GS. If the message is not signed with these GS or if it signed with any old GS, it can be assumed that it is not coming from a valid group member, hence can be easily discarded. Therefore intruder will not be able to start communication with a legitimate group member. It is also shown in Section 3.6.1 that an intruder is not able to establish a secure channel in our protocol.
- *Password guessing attack*: In our protocol, GSs are updated using reverse dynamic accumulators, which do not contain any information related to the past signature. Moreover, vehicles are not traceable in the network, so it is difficult to even for a legitimate group member to find out the actual sender of the message, leaving no clues for guessing. As we can see in Fig. 7, an intruder is able to retrieve only public keys and encrypted messages, hence composing a secret key, without any additional information is impossible.
- *Forgeability attack*: Challenger $\mathbb{C}$ is computed to signcrypt the message with signature σ which is re-calculated at the receiver side to validate if the received message has been modified in between. If so, the message is immediately discarded. Hence, any kind of forgeability attack is easily detectable in our protocol.
- *Man in-the middle attack*: Entities verify other entities before starting any communication by encrypting the message with secret keys and message digests. These MACs are irreversible and one cannot create it again unless they do not have all the parameters available with them. Since parameters have been encrypted with the receiver's secret key, a man in the middle cannot decrypt it, hence cannot create the MAC again. So, if the data has been modified by an intruder in between, the receiver will be able to identify it.
- *Denial of service attack*: Denial of service is mitigated to the most possibility by having one extra MM to perform when main MM is not present. The replay attacks are restricted in the network; identities cannot be stolen in our protocol. Hence there is very less chance for intruders to jam the network.
- *Intervention of ex-member*: Any guilty ex-members' entry in the group can be restricted by MM, as it maintains the revocation table. Any message signed with old GS can be discarded at the moment since valid group members always have the updated group keys. Hence, there is no possibility of intervention by an ex-group member.
- *Replay attack*: In each message, unique random number and TS has been used which is digested into MAC, reverting which is impossible. The only possible way is to have secret keys of the receiver to fetch RAND and TS, but the secret key is never disclosed. Hence, a replay attack is not possible.
- *Redirection attack*: Redirection is only possible by the intervention of proxy between two authorised members using re-encryption and re-signature techniques such as MM1 and MM2 here, which cannot be considered as an attack. Other than that everyone uses MAC to identify the source of the message, hence re-direction attack is not possible.
- *Bogus information attack*: Vehicles believe the incoming message only if it is signed with a valid GS. TM always monitors BSM sent by group members. If any member spreads false information, TM reports it, and MM removes that guilty member from the group. Moreover, legitimate users avoid doing any fraud. This helps in restricting the bogus information attack.
- *Traffic analysis attack*: Attackers cannot extract the identity of a node from its BSM. Hence, it is not aware of the path followed by that particular node. Thus, attackers cannot analyse traffic.
- *Unauthorised communication between network entities*: Each message is transmitted after computing its MAC, which is sent along with the message and re-calculated at the receiver side. The above-mentioned authentication protocol in section gives the complete structure for an end-to-end authentication protocol, asking for authorisation at each step. Hence, any kind of unauthorised communication is prevented by our protocol.

The above analysis gives a clear explanation of how our network is secure from various attacks. These security features are given in Table 3, where our protocol has been compared with other existing protocols for VANET and other related wireless protocols. In this table, we can see that in comparison with other protocols, our algorithm is much more secure and robust, satisfying almost all the security requirements of a vehicular network.

5 Performance analysis of the proposed protocol

Performance of any protocol is determined by how much computation power it needs and in how much time, the message can be delivered to its recipient successfully. In this section, we have analysed our protocol in terms of bandwidth and CPU cycles consumption. Bandwidth consumption will decide the overall speed of the protocol, and CPU cycle consumption will decide the computation need. Hence, these two parameters have been given below in two subsections: computation cost and communication cost. Signalling overhead section represents the overhead caused by the number of transmissions in the network.

5.1 Computation cost

Computation cost is the cost for generating the final packet after including signature and other parameters in it. To compute this, we

Table 3 Comparing resistance to security threats

Attacks ↓ references →	[6]	[9]	[20]	[21]	[27]	[28]	[29]	[12]	[23]	[24]	[SAPSC]
resists impersonation attack	no	no	yes	no	no	no	yes	yes	yes	yes	yes
prevents an attack on identity	no	no	yes	yes	no	no	no	yes	yes	yes	yes
resists transitive key derivation attack	no	no	—	—	no	yes	yes	—	yes	yes	yes
resists sybil attack	no	no	no	yes	no	no	no	yes	yes	yes	yes
resists unauthorized backward traceability	no	no	yes	yes	no	yes	yes	yes	yes	yes	yes
resists secure channel establishment by an intruder	no	no	yes	no	no	no	no	no	no	no	yes
resists password guessing attack	no	no	no	yes	no	no	yes	no	no	yes	yes
detects forgeability attack	no	no	yes	yes	no	no	no	yes	—	yes	yes
resists man in middle (MIM) attack	no	no	yes	no	no	no	yes	no	no	no	yes
prevents denial of service attack	no	no	yes	yes	no	no	yes	yes	yes	yes	yes
prevents the intervention of ex-member of the group	—	—	no	—	yes	—	no	yes	yes	yes	yes
identifies replay attack	no	no	yes	yes	no	no	no	yes	yes	yes	yes
restricts redirection attack	no	no	no	no	no	no	no	yes	yes	yes	yes
prevents bogus information attack	no	no	yes	no	no	no	—	yes	yes	yes	yes
prevents traffic analysis attack	—	no	yes	yes	no	no	—	yes	yes	yes	yes
prevents unauthorized communication between network entities	no	no	yes	yes	no	no	no	no	no	no	yes

Table 4 Cost of arithmetic and cryptographic operations

Parameters	Meaning	Cost
T_A	time required for modular addition	0.0001
T_S	time required for modular subtraction	0.0001
T_M	time required for modular multiplication	0.004
T_D	time required for modular division	0.0625
T_{Exp}	time required for modular exponentiation	1.0
T_{PMUL}	time required for point multiplication	0.125
T_{BP}	time required for bilinear pairing	4.211
T_{MOD}	time required for modulus	0.124
T_{RND}	time required for random no. generation	0.045
T_{XOR}	time required for XORing variables	0.002
T_{HASH}	time required for hashing	0.0001
T_S	time required for calculating Schnorr sign	2.004

Table 5 Computation cost spent to calculate final packet

Algorithms	Operations used	Cost
[5]	$2T_{BP} + T_M$	8.43
[12]	$T_{BP} + 8T_M + T_{HASH} + 5T_A + 10T_{Exp}$	14.23
[19]	$10T_{Exp} + 3T_{HASH} + T_{MOD} + 4T_M$	10.14
[20]	$7T_M + 4T_{HASH} + 2BP$	8.45
[23]	$5T_{Exp} + 4T_A + 3T_M + 2T_{BP}$	13.54
[24]	$6T_{Exp} + 5T_A + 4T_M + 2T_{BP}$	14.55
[30]	$11T_{Exp} + T_{SCH}$	13.00
[31]	$T_{Exp} + (n+1)T_{PMUL} + T_D + nT_F$	13.63
[32]	$6T_{BP} + 4T_{HASH} + 2T_{AS} + T_{MTP}$	14.66
[SAPSC]	$5T_{MOD} + 4T_A + 4T_{HASH} + 4T_M + 3T_{Exp} + T_{BP}$	7.325

have taken help of the cost of different parameters given in Table 4. The cost required to compute a final message is the addition of all the intermediate steps in which encrypting and signing the message takes major participation. For computing this, our protocol is mainly taking five modulus operations, four modular addition operations, four hashing, four modular multiplications, three exponentiations and one bilinear pairing operation which can be added as below

$$(5T_{MOD} = 5 \times 0.124) + 4T_A = 4 \times 0.0001) + (4T_{HASH} = 4 \times 0.0001) + (4T_M = 4 \times 0.004) + (3T_{Exp} = 3 \times 1.0) + (T_{BP} = 4.211) = 7.325\,\text{ms}. \quad (69)$$

Some protocols first encrypt the message, then signs it which takes two steps and much more computation cycles than the signcryption scheme, used in our protocol. Hence, the comparison has been made based on both criteria.

Table 5 gives the computation cost of various needed to acquire the final signed and encrypted packet with or without using signcryption. As we can see in Table 5 and the graph given in Fig. 9, our protocol has the minimum computation cost among discussed algorithms, making our algorithm very efficient in terms of CPU cycle consumption.

5.2 Communication cost

The communication cost is basically decided by what is the size of the message and its associated parameters which will be sent along with it, that means what is the size of σ in our protocol. To compute this, the sizes of the parameters have been given in Table 6. Keys are of 128 bits, and hashing those also produces the 128 bits output. MAC used is a product obtained from hashing operation, making its size as 128 bit as well. RAND variables used is also of 128 bits. One time usable random/pseudo-random nonces are taken of 64 bits. Timestamps used are of 32 bits. With every packet one

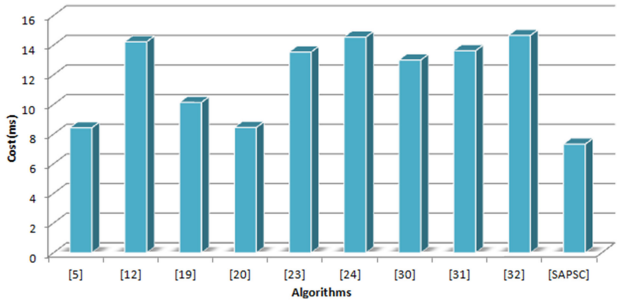


Fig. 9 *Computation cost in packet formation of various protocols*

Table 6 Key-size of the Parameters (in bits)

Parameters	GID	MID	PK	SK	RAND	TTL	TS
size	16	16	128	128	128	8	32

Table 7 Final Packet Size and signalling overhead of the various protocol

Algorithms	Message size, bytes	Signalling overhead
[5]	—	$9n$
[12]	192	$10n$
[19]	826	$7n$
[20]	388	$6n$
[23]	170	$4 + 5n$
[24]	186	$4 + 7n$
[30]	84	$4 + 6n$
[31]	144	$4 + 5n$
[32]	388	$2 + 4n$
[33]	123	$5n$
[SAPSC]	77	$9 + 3n$

time to live (TTL) is sent, denoting how much time that packet will remain live/valid. Size of this TTL is 8 bits. Hence, the total size of our packet can be given as $128 \times 4 + 64 + 32 + 8 = 616$ bits $= 77$ bytes, which comprises our final packet after being signed as well.

Table 7 gives the list of various protocols with their signature size which clearly shows that the size of the final packet in our protocol is much lesser than others. This has been depicted by the graph shown in Fig. 10. Hence, bandwidth consumption by our protocol is lesser as well which guarantees a better network performance in terms of speedy communication.

5.3 Signalling overhead

Signalling overhead gives the overhead caused in the network by the total number of transmissions. In this section, we are calculating the number of signalling messages transmitted by a vehicle for normal communication. Suppose n is the number of messages sent by vehicle in the network. Since registration and revocation are one-time processes, we are considering it constant. For each BSM transmission, it will transmit only one message. If it requires a particular route to send messages, it will take the help of the cloud. In that case, the total number of transmission messages is calculated as no. of routine request (RReq) + no. of RRep + no. of messages sent by vehicle in the network. In the worst case scenario, where the vehicle needs a route for each of its communication, we consider the number of RReq (or RRep) is equal to the number of messages sent by vehicle in the network. In that case, the total message sent by vehicle will be $3n$. Since there are nine more steps in communication which requires 1 message transmission each, total signalling overhead can be given as:

$9 + n$(no. of RReq) $+ n$(no. of RRep) $+ n$(no. of messages sent by vehicle in the network) $= 9 + 3n$.

In a similar way, signalling overhead for various references has been calculated and given in Table 7. Fig. 11 gives the chart for varying number of n from 5 to 50. It can be significantly observed

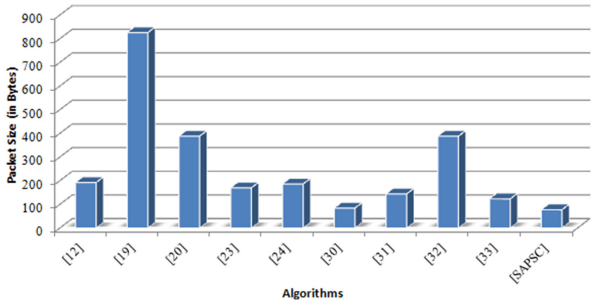


Fig. 10 *Final packet-size comparison of various protocols*

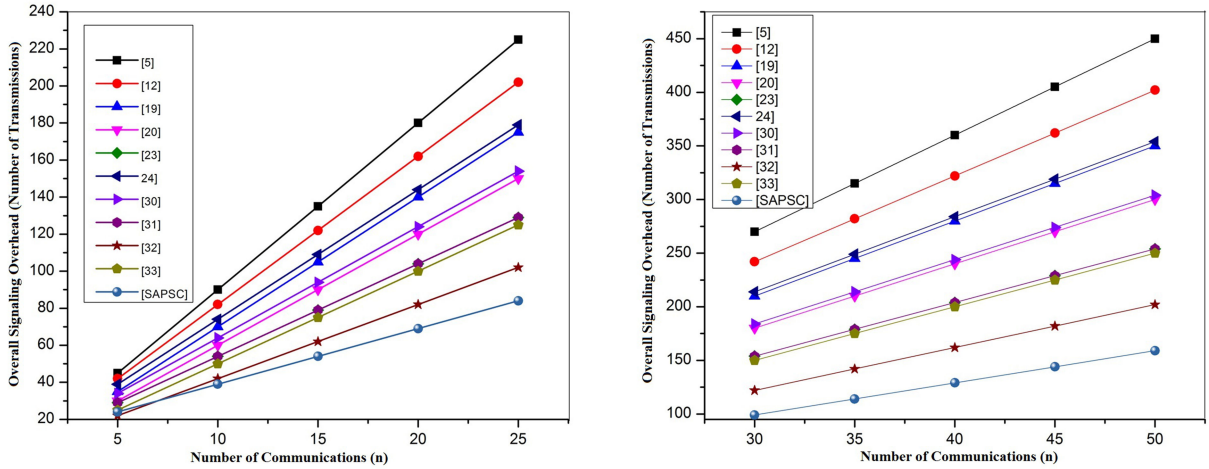


Fig. 11 *Signalling overhead of various protocols*

from the chart that our protocol has the least overhead when a number of communication increases.

6 Conclusion

To enhance the capabilities of the existing authentication protocol for VANET network, many authentication protocols are proposed by various researchers, but none of them authenticated their signcryption scheme. In this study, we proposed a robust SAPSC protocol, which is secure as well. The main objective of privacy preservation has been maintained. This algorithm also ensures mutual authentication among communicating entities before starting any transmission. Since the protocol combines signature with an encryption step, the computational cost has been significantly reduced hence reducing the overall computation overhead. The performance analysis clearly shows that our protocol has lesser communication and computation costs along with minimum signalling overhead, hence promising a supreme network performance. Moreover, our algorithm prevents the network from various attacks from outsiders and insiders both. Therefore, our protocol helps in enhancing the security of the vehicular network in an efficient way.

7 References

[1] Chhatwal, S.S., Sharma, M: 'Detection of impersonation attack in VANETs using buck filter and VANET content fragile watermarking (VCFW)'. Int. Conf. on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2015, pp. 1–5

[2] Rehman, S.U., Arif Khan, M., Zia, T.A., *et al.*: 'Vehicular ad-hoc networks (VANETs) – an overview and challenges', *J. Wirel. Netw. Commun.*, 2013, **3**, pp. 29–38

[3] Whaiduzzaman, M.D., Sookhak, M., Gani, A., *et al.*: 'A survey on vehicular cloud computing', *J. Netw. Comput. Appl.*, 2014, **40**, pp. 1226–1229

[4] Zheng, Y.: 'Digital signcryption or how to achieve cost (signature & encryption) cost (signature) cost (encryption)'. Annual Int. Cryptology Conf., Berlin, Germany, 1997, pp. 165–179

[5] Ateniese, G., Fu, K., Green, M., *et al.*: 'Improved proxy re-encryption schemes with applications to secure distributed storage', *ACM Trans. Inf. Syst. Secur. (TISSEC)*, 2006, **9**, pp. 1–30

[6] Blaze, M., Bleumer, G., Strauss, M.: 'Divertible protocols and atomic proxy cryptography'. Int. Conf. on the Theory and Applications of Cryptographic Techniques, Berlin, Germany, 1998, pp. 127–144

[7] Hohenberger, S.R., Fu, K., Ateniese, G., *et al.*: 'Unidirectional proxy re-encryption', US Patent 8,094,810

[8] Green, M., Ateniese, G.: 'Identity-based proxy re-encryption'. Applied Cryptography and Network Security, Berlin, Germany, 2007, pp. 288–306

[9] Canetti, R., Hohenberger, S.: 'Chosen-ciphertext secure proxy re-encryption'. Proc. 14th ACM Conf. on Computer and Communications Security, Alexandria, VA, USA, 2007, pp. 185–194

[10] Ateniese, G., Benson, K., Hohenberger, S.: 'Key-private proxy re-encryption'. In Cryptographers'™ Track at the RSA Conf., Berlin, Germany, 2009, pp. 279–294

[11] Chaum, D., Van Heyst, E.: 'Group signatures'. In Workshop on the Theory and Application of Cryptographic Techniques, Berlin, Germany, 1991, pp. 257–265

[12] Lin, X., Sun, X., Ho, P.-H., *et al.*: 'GSIS: a secure and privacy-preserving protocol for vehicular communications', *IEEE Trans. Veh. Technol.*, 2007, **56**, pp. 3442–3456

[13] Mamun, M.S.I, Miyaji, A.: 'Secure VANET applications with a refined group signature'. 2014 Twelfth Annual Int. Conf. on Privacy, Security and Trust (PST), IEEE, Toronto, Canada, 2014

[14] Camenisch, J., Lysyanskaya, A.: 'Dynamic accumulators and application to efficient revocation of anonymous credentials'. Annual Int. Cryptology Conf., Berlin, Germany, 2002, pp. 61–76

[15] Kuo, T.-M., Yen, S.-M., Han, M.-C.: 'Dynamic reversed accumulator', *Int. J. Inf. Secur.*, 2018, **17**, (2), pp. 183–191

[16] Wei, J., Yang, G., Mu, Y.: 'Designated verifier proxy re-signature for deniable and anonymous wireless communications', *Wirel. Pers. Commun.*, 2017, **97**, (2), pp. 3017–3030

[17] Yang, T., Yu, B., Wang, H., *et al.*: 'Cryptanalysis and improvement of panda-public auditing for shared data in cloud and internet of things', *Multimedia Tools Appl.*, 2017, **76**, (19), pp. 19411–19428

[18] Lloret, J., Canovas, A., Catalá, A., *et al.*: 'Group-based protocol and mobility model for VANETs to offer internet access', *J. Netw. Comput. Appl.*, 2013, **36**, (3), pp. 1027–1038

[19] Shao, J., Lin, X., Lu, R., *et al.*: 'A threshold anonymous authentication protocol for VANETs', *IEEE Trans. Veh. Technol.*, 2016, **65**, (3), pp. 1711–1720

[20] Bayat, M.B., Rahimi, M., Aref, M.R.: 'A secure authentication scheme for VANETs with batch verification', *Wirel. Netw.*, 2015, **21**, (5), pp. 1733–1743

[21] Chen, L., Li, Q., Martin, K.M., *et al.*: 'Private reputation retrieval in public – a privacy-aware announcement scheme for VANETs', *IET Inf. Sec.*, 2016, **11**, (4), pp. 204–210

[22] Rajput, U., Abbas, F., Oh, H.: 'A hierarchical privacy preserving pseudonymous authentication protocol for VANET', *IEEE Access*, 2016, **4**, pp. 7770–7784

[23] Kanchan, S., Chaudhari, N.S: 'Integrating group signature scheme with non-transitive proxy re-encryption in VANET'. Int. Conf. on Computing, Analytics and Security Trends (CAST), Pune, India, 2016, pp. 227–231

[24] Kanchan, S., Singh, G., Chaudhari, N.S.: 'Re-encrypting secure and efficient routing in VANET groups using sharable clouds'. 4th Int. Conf. on Recent Advances in Information Technology (RAIT), Dhanbad, India, 2018, pp. 1–6
[25] Sun, D., Zhao, H., Cheng, S.: 'A novel membership cloud model-based trust evaluation model for vehicular *ad hoc* network of T-CPS', *Secur. Commun. Netw.*, 2016, **9**, (18), pp. 5710–5723
[26] Armando, A., Basin, D., Boichut, Y.*, et al.*: 'The AVISPA tool for the automated validation of internet security protocols and applications'. Int. Conf. on computer aided verification, Berlin, Germany, 2005, pp. 281–285
[27] Ateniese, G., Hohenberger, S.: 'Proxy re-signatures: new definitions, algorithms, and applications'. ACM Proc. 12th ACM Conf. on Computer and Communications Security, Alexandria, VA, USA, 2005, pp. 310–319
[28] Zhang, J., Wang, X.: 'Non-transitive bidirectional proxy re-encryption scheme'. IEEE Int. Conf. on networking and Digital Society, Guiyang, China, 2009, pp. 213–216
[29] Ma, C., Ao, J.: 'Group-based proxy re-encryption scheme secure against chosen ciphertext attack', *Int. J. Netw. Secur.*, 2009, **8**, (3), pp. 266–270
[30] Sur, C., Park, Y., Rhee, K.H.: 'An efficient and secure navigation protocol based on vehicular cloud', *Int. J. Comput. Math.*, 2016, **93**, (2), pp. 325–344
[31] Kumari, S., Khan, M.K.: 'More secure smart card-based remote user password authentication scheme with user anonymity', *Secur. Commun. Netw.*, 2014, **7**, (11), pp. 2039–2053
[32] 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**, (5), pp. 351–358
[33] Jinila, N, Komathy, K.: 'A privacy preserving authentication framework for safety messages in VANET'. IET Chennai Fourth Int. Conf. on Sustainable Energy and Intelligent Systems, Chennai, India, 2013, pp. 456–461