Internet of Things 20 (2022) 100605

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

CREASE: Certificateless and REused-pseudonym based Authentication Scheme for Enabling security and privacy in VANETs

Shafika Showkat Moni [a], D. Manivannan [b,*]

[a] Department of Computer Science, Montclair State University, Montclair, NJ 07043, USA
[b] Department of Computer Science, University of Kentucky, Lexington, KY 40508, USA

ARTICLE INFO

Keywords:
VANETs
Intelligent transportation systems
Authentication
Security
Privacy-preserving authentication

ABSTRACT

Due to the customers' growing interest in using various intelligent and connected devices, we are surrounded by the Internet of Things (IoT). It is estimated that the number of IoT devices will exceed 60 billion by 2025. One of the primary reasons for such rapid growth is the Internet of Vehicles (IoV). Internet of Vehicles (IoV) has evolved into an emerging concept in intelligent transportation systems (ITS) that integrates VANETs and the IoT to enhance their capabilities. With the emergence of IoV and the interest shown by customers, Vehicular Ad hoc NETworks (VANETs) are likely to be widely deployed in the near future. However, for this to happen, wide participation of vehicle owners in VANET is needed. The primary concerns of vehicle owners to participate in VANET are privacy and security. In this paper, we present a Certificateless and REused-pseudonym based Authentication Scheme for Enabling security and privacy (CREASE) in VANETs. One of the ways to preserve the privacy of vehicles/drivers is to allow vehicles/drivers to use pseudo identities (pseudonyms) instead of their real identities (such as VIN number or driving license number) in all communications. The pseudonym used by a vehicle needs to be changed frequently to prevent the vehicle from being tracked. Our scheme uses Merkle Hash Tree and Modified Merkle Patricia Trie to efficiently store and manage the pseudonyms assigned to a vehicle. This enables a vehicle to pick and use a random pseudonym from a given set of pseudonyms assigned to it as well as change its pseudonym frequently and securely to ensure privacy. Unlike many of the existing schemes, our scheme does not use certificates and certificate revocation lists for authentication. Moreover, it allows vehicles to get a set of pseudonyms only once from the trusted authority. We present a formal proof of correctness of our scheme and also compare our scheme with some of the other contemporary schemes to show the effectiveness of our scheme.

1. Introduction

Due to the customers' growing interest in using various intelligent and connected devices, we are surrounded by the Internet of Things (IoT). It is estimated that the number of IoT devices will exceed 60 billion by 2025. One of the primary reasons for such rapid growth is the Internet of Vehicles (IoV). Internet of Vehicles (IoV) is an emerging concept in intelligent transportation systems (ITS) that integrates Vehicular Ad hoc NETworks (VANETs) and the IoT to enhance their capabilities. With the advent of the Internet of Things (IoT), traditional vehicular ad-hoc networks (VANETs) are evolving into the IoV. VANETs are expected to assist drivers in driving safely and also provide pleasant driving experience to both drivers and passengers. Moreover, VANETs

* Corresponding author.
E-mail addresses: monis@montclair.edu (S.S. Moni), mani@cs.uky.edu (D. Manivannan).
URL: http://www.cs.uky.edu/~manivann (D. Manivannan).

https://doi.org/10.1016/j.iot.2022.100605
Received 13 July 2022; Received in revised form 11 August 2022; Accepted 15 August 2022
Available online 18 August 2022

are likely to play an important role in intelligent transportation systems (ITSs) to improve transportation efficiency and security. Generally, a VANET consists of On-Board Units (OBUs), Roadside Units (RSUs), and Trusted Authorities (TAs). The entities in VANETs communicate with each other through Dedicated Short Range Communication (DSRC) [1] or Transport Layer Security (TLS) protocols in Vehicle-to-Vehicle (V2V) communication, Vehicle-to-Infrastructure (V2I) communication, and Vehicle-to-Everything (V2X) communication.

In spite of the potential benefits of VANETs, widespread deployment of VANETs face some serious challenges. Due to the wireless nature of VANET communication, it is vulnerable to a large number of attack vectors. Authentication plays an important role in secure message dissemination. Without an effective authentication framework, attackers could compromise other drivers on VANETs easily. For example, malicious vehicles spreading false information about an accident might block the road, leading to a traffic jam. It may also spoof an RSU or electronic toll booth to steal other drivers' sensitive data. Moreover, vehicle users may refuse to take part in VANET due to lack of privacy and security. Therefore, privacy-preserving authentication and message dissemination schemes need to be designed and implemented. Pseudonym based authentication and message dissemination is one of the most popular solutions proposed in the literature to protect vehicles'/drivers' privacy; in such authentication, pseudo IDs (also known as pseudonyms) are used by vehicles instead of their real ID in all communication. Each vehicle is equipped with an OBU to communicate with other vehicles as well as with RSUs. In pseudonym based approach, a vehicle's OBU is loaded with a set of pseudonyms by a Trusted Authority. Vehicles are required to change their pseudonym frequently to avoid traceability. However, periodically changing pseudonym of a vehicle is not effective to prevent pseudonym linking attacks. For example, suppose out of 100 vehicles, only one vehicle changes pseudonym. In that case, an intruder can easily link the old and the new pseudonyms used by the vehicle by linking two messages to the same vehicle and track the path traversed by the vehicle. In addition to that, more research needs to be done in devising efficient method for managing pseudonyms of vehicles.

1.1. Contributions of this paper

In this paper, we address the above issues and propose a **C**ertificateless and **RE**used-pseudonym based **A**uthentication **S**cheme for **E**nabling Security and Privacy in VANETs (**CREASE**) that leverages Merkle Hash Tree (MHT) [2] and Modified Merkle Patricia Trie (MMPT). The main contributions of our paper are:

(i) We propose a distributed and decentralized certificateless authentication scheme for efficient authentication of vehicles. We use Modified Merkle Patricia Trie (MMPT) combined with Merkle Hash Tree (MHT) for storing vehicles' pseudonyms and their corresponding 'current status' values efficiently.
(ii) By using MHT, CREASE allows a vehicle to authenticate an RSU whereas many of the existing schemes assume that RSUs are fully trusted. Moreover, our approach does not use certificates for authentication.
(iii) To prevent vehicles' routes being tracked, each RSU assists vehicles in its region to change their pseudonym simultaneously; this is accomplished by assigning the same expiration time for all the pseudonyms of all vehicles in its region. After the expiration time elapses, each vehicle will again communicate with an RSU to activate a new pseudonym from a pool of pseudonyms received from its home RTA during initial registration. We assume that a vehicle will always have sufficient number of pseudonyms, so that it will not need to reuse a pseudonym within a year. Expiration time associated with a pseudonym helps vehicles within the same RSU's region to simultaneously and frequently change their pseudonym to reduce the chance of linking messages sent by the same vehicle with two different pseudonyms.

The rest of the paper is organized as follows: We introduce the system model in Section 2. In Section 3, we present our privacy-preserving authentication and pseudonym changing scheme CREASE. We analyze the security of CREASE, compare it with other related schemes, and also formally verify the correctness of CREASE in Section 4. In Section 5, we discuss some of the related works. Lastly, Section 6 concludes the paper.

2. System model

Fig. 1 shows an overview of our system model. It consists of two tiers: the top tier is made up of the Trusted Authority (TA) and Regional Trusted Authorities (RTAs), and the bottom tier includes RSUs and OBUs. Every RTA operates as a lower-level local TA for its region, while the TA is the root of the entire system. In Table 1 we present the notations used in this paper.

In CREASE, the TA generates its own public and private key pairs (PU_{TA}, PR_{TA}). Each RTA registered with the TA generates its public and private key pairs (PU_{RTA}, PR_{RTA}) and lets its public key known to the TA. The RTA acts as a local TA. Each RSU under a TA generates its (public, private) key pair (PU_{RSU}, PR_{RSU}) and informs its public key to its TA. Each RTA maintains a Merkle Hash Tree (MHT) [2] of public keys of all RSUs registered with it. Each vehicle is registered with its home RTA to participate in VANET. For this, each vehicle V generates its (public, private) key pair (PU_V, PR_V) and registers its public key along with its ID with its home RTA. When a vehicle registers with its home RTA, its OBU is loaded with a set of pseudonyms $\{PID_1, \ldots, PID_n\}$, and an initial pseudonym $PID_{V_{initial}}$ signed by its home RTA as $(E(H(PID_{V_{initial}})||t_{exp}), PR_{RTA})$, where $PID_{V_{initial}} \in \{PID_1, \ldots, PID_n\}$ and t_{exp} is the expiration time of $PID_{V_{initial}}$. The OBU also stores MHT root generation timestamp $T_{mhtRoot}$ generated and assigned by the RTA as well as the public key of the TA PU_{TA}. To preserve its privacy, a vehicle never uses its real identity in its communications. There are various approaches for generating and assigning pseudonyms for vehicles [3–6]. However, we do not address this issue in this paper.

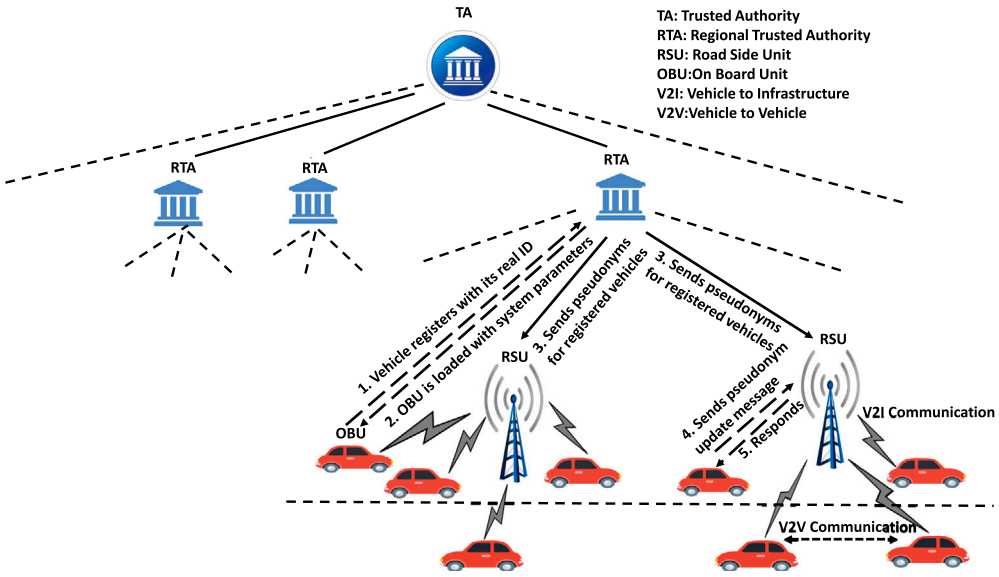


Fig. 1. Proposed VANET architecture for CREASE.

Table 1
Notation and description.

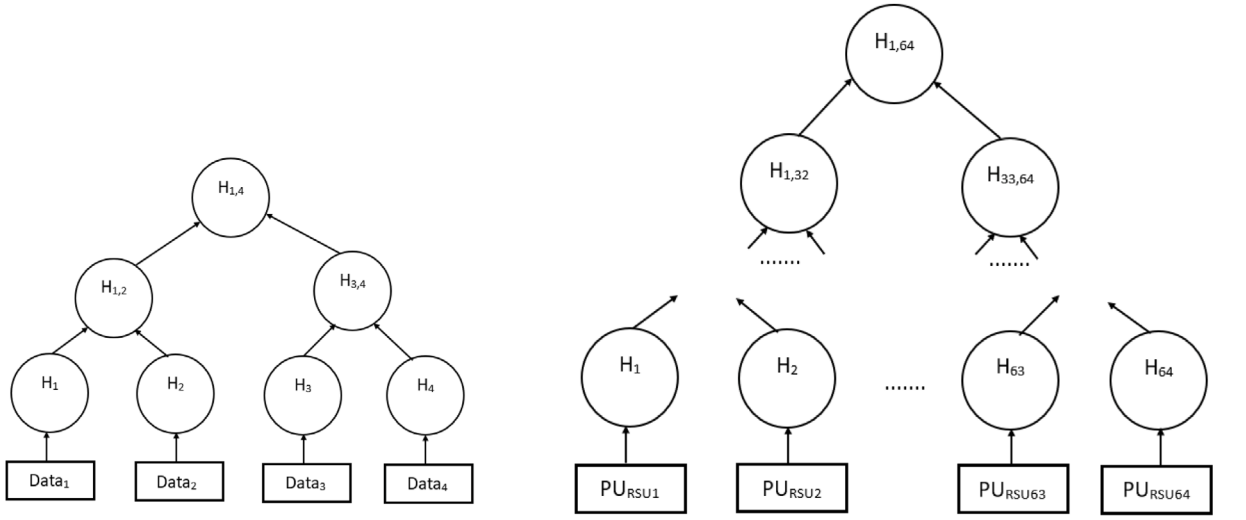
Notation	Description
TA	Trusted Authority
RTA	Regional TA
RSU	Road Side Unit
OBU	On Board Unit
PU_{TA}, PR_{TA}	Public and Private Keys of the TA
PU_{RTA}, PR_{RTA}	Public and Private Keys of RTA
PU_{RSU}, PR_{RSU}	Public and Private Keys of RSU
PU_V, PR_V	Public and Private Keys of vehicle V
ID_{RSU}	ID of RSU
$PID_{V_{initial}}$	Initial pseudonym assigned to Vehicle V
$PID_{V_{curr}}$	Pseudonym currently used by Vehicle V
$PID_{V_{new}}$	New Pseudonym activated for Vehicle V
t_{exp}	Initial Pseudonym Expiration Time of V
t_{new}	Current Pseudonym Expiration Time of V
t'_{new}	Newly Activated Pseudonym Expiration Time of V
t_s	Message generation timestamp
$signbyTA$	Signature of TA
$signbyRTA$	Signature of RTA
E	Encryption algorithm
H	SHA-256 hash function
MHT	Merkle Hash Tree
$MHVs$	Missing Hash Values of MHT for corresponding RSU

3. Proposed CREASE scheme

In this section, we describe CREASE that leverages MHT and MMPT for privacy-preserving authentication and efficient management of pseudonyms of vehicles. Firstly, we present the basic idea behind CREASE. Then, we present a summary of assumptions used. Next, we describe MHT and MMPT, which are used in our scheme. Then, we present a description of pseudonym distribution and privacy-preserving authentication in detail. We assume that the reader is familiar with the cryptographic terminologies/concept, such as, RSA encryption, RSA decryption, RSA signature, RSA signature verification [7], and secure hashing algorithm (SHA) [8].

3.1. Basic idea

Each vehicle registers with its home RTA with its real ID to take part in VANET. The home RTA generates and assigns a set of pseudonyms for each vehicle during its registration. The vehicle's OBU is loaded with its public–private key pair, the set of pseudonyms, and an initial pseudonym signed by its home RTA during registration. RTAs also send the registered vehicle's pseudonyms to all RSUs in its region using a secure protocol such as Transport Layer Security (TLS). RSUs maintain an MHT



(a) Merkle Hash Tree (b) Merkle Hash Tree of public keys of RSUs registered under an RTA

Fig. 2. Merkle Hash Tree examples.

combined with MMPT to facilitate the efficient management and authentication of pseudonyms of vehicles (how this is done is explained later). When a vehicle enters an area covered by an RSU for the first time after registration, it uses the credentials in the RSU's beacon message to authenticate the public key of the RSU. Upon successful authentication of the public key of the RSU, the vehicle sends its public key and initial pseudonym signed by its home RTA to authenticate itself. After mutual authentication, RSU sets a new pseudonym expiration time for the initial pseudonym and generates a symmetric key for encrypting and decrypting messages between RSU and the vehicle. The RSU also generates a group key to be used by all vehicles within its region for vehicle to vehicle (V2V) communication. Next, the RSU sends the new pseudonym expiration time, the symmetric key, and the group key to the vehicle securely using a reliable transport layer protocol. RSUs assist vehicles in their region to change their pseudonym by providing a pseudonym expiration time associated with the pseudonym. When the validity time of a vehicle's current pseudonym is about to expire, the vehicle will again communicate with its RSU to activate a new pseudonym from the pool of pseudonyms received from its home RTA during initial registration. This pseudonym changing strategy speeds up the pseudonym changing frequency to avoid traceability of the vehicle. These steps involved in the authenticated communication process is depicted in Fig. 1.

3.2. Assumptions

The following assumptions are made in this work:

(i) Each vehicle is aware of the public key of the TA, namely, PU_{TA} and public key of its home RTA, namely, PU_{RTA} under which it is registered. These are loaded into the vehicle's OBU during its initial registration with its home RTA.
(ii) The home RTA of a vehicle generates and assigns a pool of pseudonyms for the vehicle during its registration. These pseudonyms are loaded into the vehicle's OBU.
(iii) An MHT of the public keys associated with the RSUs registered with an RTA is created by the RTA (as shown below). Each RSU in the region is then given the relevant Missing Hash Values (MHVs). MHVs are described below.
(iv) Each RSU maintains the MHT accompanied by an MMPT to manage the pseudonyms of vehicles efficiently. Each RSU also distributes a symmetric key S_k for each vehicle in its region, used for secure communication between itself and the vehicle; it also distributes a group key G_k securely to all vehicles in its region for group communication.
(v) RSUs registered under the same RTA know the public keys of each other.
(vi) Vehicle's OBU is tamper resistant and has enough storage to store a large set of pseudonyms. This is not a serious restriction considering the current hardware capabilities.
(vii) Clocks of TA, RTAs, RSUs, and OBUs are loosely synchronized. Using GPS, such synchronization can be achieved.

3.3. Preliminaries

This section introduces how Merkle Hash Tree (MHT) and Modified Merkle Patricia Trie (MMPT) facilitate the mutual authentication process.

Table 2
Missing Hash Values (MHVs) for corresponding RSUs.

RSU_i	$MHVs$
RSU_1	$H_2,\ H_{3,4},\ H_{5,8},\ H_{9,16},\ H_{17,32}\ H_{33,64}$
RSU_2	$H_1,\ H_{3,4},\ H_{5,8},\ H_{9,16},\ H_{17,32}\ H_{33,64}$
RSU_3	$H_4,\ H_{1,2},\ H_{5,8},\ H_{9,16},\ H_{17,32},\ H_{33,64}$
...	...
RSU_{33}	$H_{34},\ H_{35,36},\ H_{37,40},\ H_{41,48},\ H_{49,64},\ H_{1,32}$
...	...
RSU_{64}	$H_{63},\ H_{61,62},\ H_{57,60},\ H_{49,56},\ H_{33,48},\ H_{1,32}$

3.3.1. Merkle Hash Tree

A Merkle Hash Tree (MHT) [2] provides a secure and efficient way to verify data in a large data structure by using a hash [9] based tree structure. An MHT stores data in each leaf node, while non-leaf nodes store the hashes of their children. A sample MHT with four leaf nodes is presented in Fig. 2(a). Data are stored in the leaf nodes while values in non-leaf nodes are derived from the hash of its children. To prove the integrity of $Data_1$ in Fig. 2(a), we only need the relative Missing Hash Values (MHVs) of MHT (H_2, $H_{3,4}$) and the root value $H_{1,4}$. The MHVs are used to recalculate the root hash value, by first computing $H_{1,2} = H(H(Data_1), H_2)$ and then $H'_{1,4} = H(H_{1,2}, H_{3,4})$. If the recalculated value $H'_{1,4}$ and original root value $H_{1,4}$ are same, then the integrity of $Data_1$ is verified.

Each RTA constructs an MHT containing the public keys of all RSUs registered under it. Fig. 2(b) illustrates an example of an MHT consisting of public keys of sixty-four RSUs registered under an RTA. Each leaf node in the MHT stores the public key of an RSU registered with the RTA, and each non-leaf node contains the hash of its children. Each RTA sends the following information to all RSUs in its region [10]:

- its own public key signed by the TA.
- root value of the MHT signed by the RTA.
- corresponding MHVs (described above) that fall along the authentication path of that RSU's public key.

Table 2 presents the MHVs corresponding to public keys of different RSUs registered under an RTA. Each RTA sends the MHT root generation time $T_{mhtRoot}$ to the TA whenever it constructs or reconstructs (i.e., when a new RSU is added or an existing RSU is found to be compromised) the MHT. An RTA can use any of the existing algorithms [11–13] or new algorithms to detect the compromised or malicious RSUs or vehicles. The TA broadcasts the latest value of $T_{mhtRoot}$ to all RTAs; the RTAs broadcast this value to all RSUs in its region. RSUs also broadcast it to vehicles within its region. Suppose that RSU_p is under RTA_x, and RSU_q is connected with RTA_y. If a new RSU_r is put in service under RTA_x, it generates a new MHT root based on the newly added RSU_r and sends the new MHT root generation timestamp $T_{mhtRoot_x}$ to the TA. At the same time, if RSU_q is found to be malicious, RTA_y discards its public key from its MHT and reconstructs its MHT. The MHT root generation timestamp $T_{mhtRoot_y}$ of this new MHT is also sent to the TA by RTA_y. The TA compares $T_{mhtRoot_x}$ and $T_{mhtRoot_y}$, and broadcasts only the latest timestamp to all other RTAs. If $T_{mhtRoot_x}$ and $T_{mhtRoot_y}$ are coincidently the same, it randomly selects one for the broadcasting. Each RTA then broadcast this latest timestamp to their corresponding RSUs through which vehicles are also aware of this updated timestamp. *Note that the MHT of the public keys of the RSUs under an RTA, its root value and the root generation timestamp change at the RTA which sees a new or malicious RSU. As a result of this, at all other RTAs, only the root generation timestamp of their MHT changes, not the values stored in their MHTs*. When a vehicle moves under a new RSU, it compares this latest timestamp $T_{mhtRoot_y}$ with the timestamp that the RSU broadcasts, let us say $T_{mhtRoot_n}$. If $T_{mhtRoot_n} < T_{mhtRoot_y}$, the vehicle rejects the request for connection from that RSU. RSUs are generally static by nature and less likely to be compromised. Thus, the frequency of such broadcasts is very low. Therefore, CREASE provides a viable solution for revoking an RSU without maintaining a CRL database.

3.3.2. Modified Merkle Patricia Trie

Modified Merkle Patricia Trie (MMPT) is a combination of Merkle Tree and Patricia Trie with additional optimizations to meet the requirements of Ethereum [14]. It takes $O(\log(n))$ time for insert, lookup, and delete operation (where n represents the number of leaf nodes in the MMPT). We use an MMPT to store the pseudonyms assigned to vehicles. Every node in MMPT is expressed as a key–value pair [15]. Following are the three types of nodes in an MMPT:

- **Leaf Node:** A leaf node does not have a child node. The prefix of a node indicates the type of the node; prefix 2 indicates it is a leaf node. Each leaf node contains the (key, value) pair (pseudonym, status) for each pseudonym assigned to a vehicle; the status (1 or 0) of that pseudonym indicates whether that pseudonym is currently being used by the vehicle or not.
- **Branch Node:** Branch nodes are indicated with prefix 1. Branch nodes can have at most 16 children nodes, one for each hexadecimal number from 0 to f.
- **Extension Node:** Extension nodes have prefix 0. It is an optimized version of a branch node and its key field contains a partial path (shared nibble) that allows us to skip ahead and a pointer to the next node.

Next we explain how an RSU uses an MMPT to store the pseudonyms (in hexadecimal representation) of a vehicle along with their status. Table 3 contains a sample list of four pseudonyms of a vehicle and their current status. Fig. 3 shows the MMPT storing

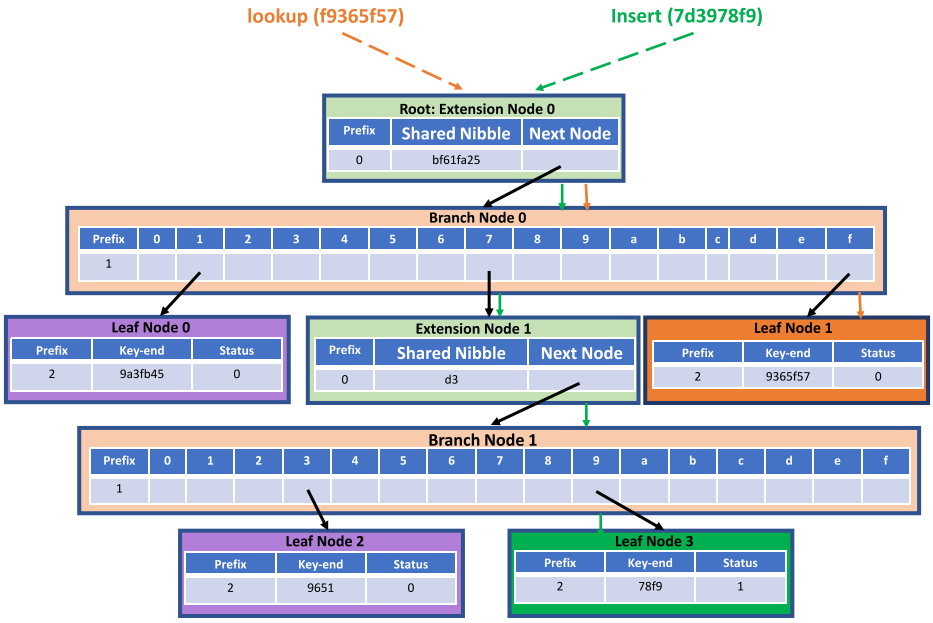


Fig. 3. Modified Merkle Patricia Trie for storing the contents of Table 3.

Table 3
Pseudonyms and Current Status.

Pseudonym	Status
19a3fb45	0
f9365f57	0
7d339651	0
7d3978f9	1

these four pseudonyms along with their status. In the MMPT, the root node is an extension node that contains the shared nibble $bf61fa25$, the public key of the vehicle which is concatenated with its pseudonyms. The root node's "next node" field points to the node right after it, which is a branch node (Branch Node 0) in our case. If we look at the second pseudonym in Table 3 after concatenation, we can find f after $bf61fa25$. With this f, we can proceed into the next level, the leaf node (Leaf Node 1) in Fig. 3, which stores both the remaining part of the key and its current status. Therefore, We must start the search at the root node to lookup a key in MMPT and then proceed to the subsequent nodes based on the shared nibbles and remaining nibbles in the key. Finally, we can find the pseudonym and its status when we reach a leaf node. Note that the pseudonym is obtained by concatenating all the keys along the path that leads to the leaf node.

The insertion operation creates an entry for a pseudonym of a vehicle and the current status of the pseudonym in MMPT. We should first start from the root node to insert a key–value pair. Next, determine the current node's prefix value and its nibbles. If the current node has a prefix of 1, then check whether the slot following the next nibble points to NULL. If this is the case, generate a new leaf node or a new extension node based on the residual nibbles left in the key to be inserted. Otherwise, navigate to the next node. If the current node's prefix is 0, find the shared nibbles and remaining nibbles left in the key. Afterwards, generate a new leaf node, or a new branch node, or a new extension based on the leftover nibbles in the key after sharing. For example, in Fig. 3, we first start from the root node to insert key "$7d3978f9$" after concatenating it with the public key $bf61fa25$ of the vehicle. Next, we check the prefix of the current node. The root node's prefix is 0, and it is an Extension Node 0 in Fig. 3. After that, we traverse to the Branch Node 0, pointed by the root node's next node field. Since the slot corresponding to the next nibble in the Branch Node 0 is not NULL and the remaining nibbles left in the key are greater than 1. Therefore, we travel down to the Extension Node 1, where the partial path diverges at Branch Node 1. We find that the slot corresponding to the Branch Node 1 is NULL. Next, we generate a new leaf node (Leaf Node 3) into this branch and set the status to 1 to indicate that this is the current pseudonym used by the vehicle with public key $bf61fa25$. The use of branch nodes, extension nodes, and leaf nodes in MMPT reduces the length of a unique path to leaf nodes, and makes it more efficient for inserting, retrieving, and removing pseudonyms. MMPT has worst case complexity of $O(n)$ for lookup, insert, and delete, where n is the length of the pseudonym (in hexadecimal representation).

MMPT is combined with the conventional block-chain to store the certificates of vehicles for authentication in BPPA [16]. In CREASE, each RSU maintains an MHT combined with MMPT as shown in Fig. 4 to store and manage vehicles' pseudonyms for efficient privacy-preserving authentication. Each RSU also maintains a database containing the public key of vehicle PU_V, set of pseudonyms assigned to the vehicle $\{PID_1, \ldots, PID_n\}$, and corresponding MHVs. RSUs use the MHVs of MHT to verify the

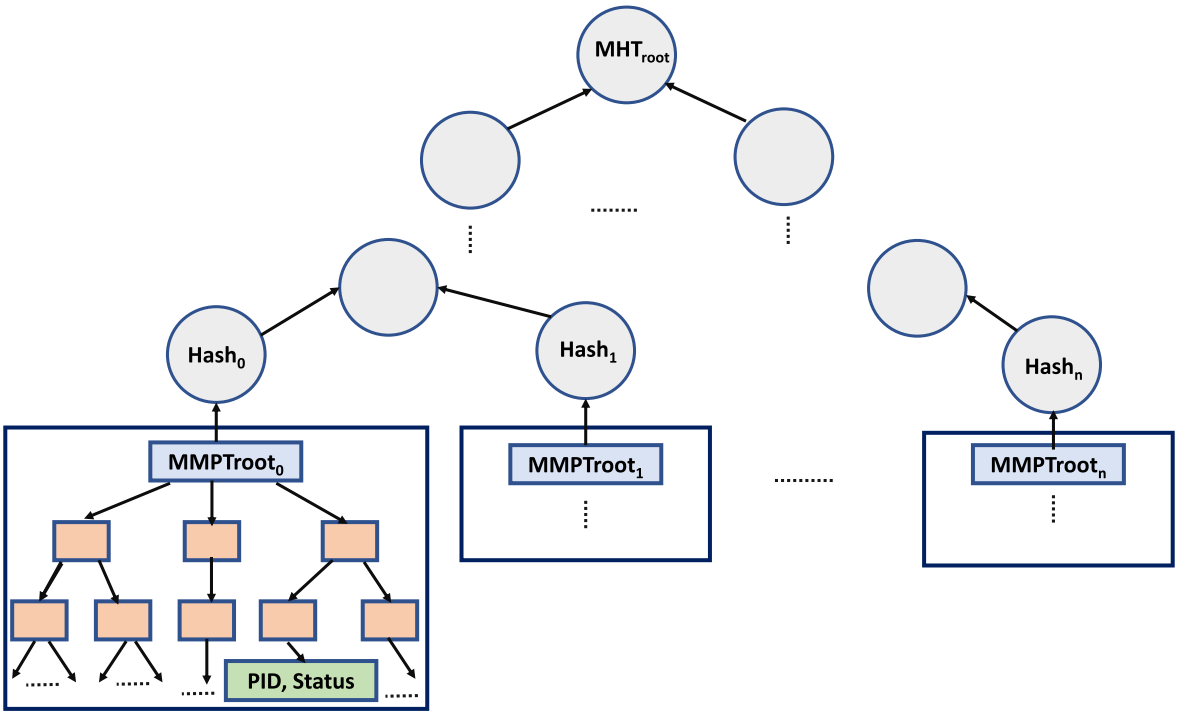


Fig. 4. MMPT combined with Merkle Hash Tree for storing pseudonyms of vehicles.

presence/absence of the pseudonym in the MMPT. Each MMPT stores the pseudonyms of a vehicle along with their latest status (active/inactive). The lookup operation in MMPT makes it efficient for an RSU to check and update the status of a specific pseudonym of a vehicle.

3.4. Detailed description of CREASE

3.4.1. Distribution of pseudonyms to vehicles by their home RTA

In CREASE, RTAs generate pseudonyms to be used by all vehicles registered with it. Initially, each vehicle V registers with its real ID with its home RTA. Each vehicle's OBU is loaded with its public–private key pair (PU_V, PR_V), a set of pseudonyms $\{PID_1, \ldots, PID_n\}$, and one of the pseudonym is designated as an initial pseudonym $PID_{V_{initial}}$ after registration. $PID_{V_{initial}}$ along with its expiration time t_{exp} is signed by its home RTA ($E(H(PID_{initial})||t_{exp})$, PR_{RTA}), where $PID_{V_{initial}} \in \{PID_1, \ldots, PID_n\}$ and t_{exp} is the expiration time of $PID_{V_{initial}}$. V uses this $PID_{V_{initial}}$ to communicate with an RSU that it encounters for the first time after the registration. RTA also sends $\{PID_1, \ldots, PID_n\}$, $PID_{V_{initial}}$, and PU_V of V to all RSUs within its region using a secure protocol such as TLS. Upon receiving this, RSUs concatenate the PU_V with the pseudonyms and inserts them along with their status into their MMPT. Initially, the status of all pseudonyms is set to 0 (inactive) except for $PID_{V_{initial}}$. Thereby, all the RSUs under an RTA get the pseudonyms of all vehicles registered under the RTA. The summary of the above pseudonym distribution process is presented in **Algorithm 1**.

Algorithm 1: Distribution of Pseudonyms by RTA

When a vehicle V registers with its home RTA

1 V's OBU is loaded with its (PU_V, PR_V), a set of pseudonyms $\{PID_1, \ldots, PID_n\}$, and a $PID_{V_{initial}}$ signed by the RTA ($E(H(PID_{V_{initial}})||t_{exp})$, PR_{RTA}), where $PID_{V_{initial}} \in \{PID_1, \ldots, PID_n\}$ and t_{exp} is the expiration time of $PID_{V_{initial}}$;

2 RTA sends $\{PID_1, \ldots, PID_n\}$, $PID_{V_{initial}}$, and PU_V to all RSUs in its region using a secure protocol such as TLS;

3.4.2. When a vehicle V enters an area covered by an RSU after registration

V listens to the beacon message of the RSU which includes its ID ID_{RSU}, public key PU_{RSU}, PU_{RTA} signed by TA $PU_{RTA(signbyTA)}$, root value of the MHT signed by RTA $root_{signbyRTA}$ (where $root_{signbyRTA}$ contains (root of MHT || RSA signature || $T_{mhtRoot}$)), MHVs corresponding to the public key of the RSU PU_{RSU}, and timestamp t_s. V first checks the freshness of the beacon message from RSU using t_s. Next, V verifies the signature of the TA and RTA. A vehicle can move across different regions covered by several

RTAs. We do not require vehicles to store all of the public keys for all the RTAs. CREASE requires V only to store the PU_{TA} to get the PU_{RTA}, which then can be used to verify the public key of the RSU. After verifying PU_{TA} and PU_{RTA}, V compares $T_{mhtRoot}$ in beacon message with the stored value of $T_{mhtRoot}$. If the received value of $T_{mhtRoot}$ is greater than or equal to the stored value of $T_{mhtRoot}$, then the vehicle recalculates the root of MHT using the MHVs and hash value of the public key PU_{RSU} of the sender RSU received in the beacon message. Next, V compares the received MHT root value with the calculated root value of MHT. If the two values are equal, PU_{RSU} is considered authentic. The detailed description of the authentication of RSU using MHT is presented in first part of **Algorithm 2**. After authenticating the RSU, V sends the following message to the RSU:

$V \rightarrow RSU$: $(E(PID_{V_{initial}}, (E(H(PID_{V_{initial}})\|t_{exp}), PR_{RTA}), PU_V, t_s), PU_{RSU})$

where, $(E(H(PID_{V_{initial}})\| t_{exp}), PR_{RTA})$ is the initial pseudonym of vehicle signed by its home RTA.

Algorithm 2: When a Vehicle V enters an area covered by an RSU after registration

Upon receiving the beacon message (ID_{RSU}, $PU_{RTA\,signbyTA}$, $MHVs$, $root_{signbyRTA}$, t_s) from the RSU, V authenticates the RSU in the following way

Firstly, V checks t_s;
if t_s *is valid* **then**
  Verifies the signatures of TA and RTA;
  Retrieves root value and root generation timestamp of MHT from $root_{signbyRTA}$;
  if *root generation timestamp is valid* **then**
    Calculates root value using $MHVs$ and PU_{RSU};
    if *(calculated root value == received root value)* **then**
      RSU's public key is authenticated;
    else
      RSU's public key is not valid;
    end if
  end if
end if

After authenticating the RSU, V sends ($E(PID_{V_{initial}}, (E(H(PID_{V_{initial}})\|t_{exp}), PR_{RTA}), PU_V, t_s), PU_{RSU}$) to the RSU;
When the RSU receives the above message from V

Decrypts the message using its private key PR_{RSU};
Checks the freshness of the received message using t_s;
if t_s *is valid* **then**
  Verifies the signature of RTA;
  Retrieves t_{exp};
  if t_{exp} *is valid* **then**
    Retrieves $H(PID_{V_{initial}})$ and checks it against the calculated hash of $PID_{V_{initial}}$;
    if *the hash values match* **then**
      if $PID_{V_{initial}} \in MMPT$ **then**
        Sets the status of the $PID_{V_{initial}}$ to 1;
        Sets new expiration time t_{new} for $PID_{V_{initial}}$;
        Sends $(E(ID_{RSU}, (E(H(PID_{V_{initial}})\| t_{new}), PR_{RSU}), S_k, G_k, t_s), PU_V)$ to V, where G_k is the group key and S_k is the symmetric key between V and RSU;
      else
        $PID_{V_{initial}}$ is not valid;
      end if
    else
      Authentication fails;
    end if
  end if
end if

RSU, upon receiving the above message, uses received t_s to verify the freshness of the above message. Then, RSU verifies the signature of the RTA. Next, it retrieves the pseudonym expiration time t_{exp}. If t_{exp} is valid, then it calculates the hash of the received initial pseudonym $PID_{V_{initial}}$ and compares this calculated hash value with received hash value $H(PID_{V_{initial}})$. If these two hash values are equal, then RSU concatenates public key of the vehicle PU_V with the $PID_{V_{initial}}$ and sets the status of the $PID_{V_{initial}}$ to 1 in the MMPT. RSU also sets new expiration time t_{new} for $PID_{V_{initial}}$ and signs it. RSU generates a symmetric key S_k for encrypting and exchanging messages between the V and RSU. After that, it sends the following message to the V:

$RSU \rightarrow V$: $(E(ID_{RSU}, (E(H(PID_{V_{initial}})\| t_{new}), PR_{RSU}), S_k, G_k, t_s), PU_V)$

where, $(E(H(PID_{V_{initial}})\| t_{new}), PR_{RSU})$ is the initial pseudonym with new expiration time signed by the RSU and G_k is the group key to be used by all vehicles authenticated by the same RSU. Encrypting large messages using public key cryptography is not

efficient. A detailed description of the above discussion is presented in the second part of the **Algorithm 2**. After obtaining the group key G_k, vehicles under an RSU use G_k to securely broadcast messages to all other vehicles under that RSU. A vehicle attaches its current pseudonym $PID_{V_{curr}}$ signed by the RSU $(E(H(PID_{V_{curr}})\| t_{exp}), PR_{RSU})$ to the message m for broadcasting m to other vehicles under the RSU. It also appends message generation timestamp t_s to prevent replay attacks. Whenever a vehicle needs to broadcast a message m to other vehicles in the current RSU's region, it encrypts m as follows:

$V(Sender) \rightarrow V(Receiver)$: $(E(PID_{V_{curr}}, (E(H(PID_{V_{curr}}) \| t_{new}), PR_{RSU}), m, t_s), G_k)$.

The receivers can verify the authenticity of the received message by checking the current pseudonym expiration time t_{new} and checking the received $H(PID_{V_{curr}})$ against its calculated hash of received $PID_{V_{curr}}$ in the above message. If both verifications are successful, the received message is considered authentic. Otherwise, the receiving vehicles ignore the message.

Algorithm 3: Updating the Status of Pseudonyms by RSU

When the validity time of V's current pseudonym $PID_{V_{curr}}$ expires
V selects a new pseudonym $PID_{V_{new}}$ from the set of pseudonyms allocated to it;
Marks $PID_{V_{curr}}$ as inactive;
Sends $(E(PID_{V_{curr}}, PID_{V_{new}}, PU_V, t_s), S_k)$ to the RSU;
When the RSU receives the above update message
Decrypts the message using secret key S_k;
Checks the freshness of the received message using t_s;
if t_s *is valid* **then**
    Looks up into MMPT for $PID_{V_{curr}}$ and $PID_{V_{new}}$;
    if $PID_{V_{curr}} \in MMPT$ *and* $PID_{V_{new}} \in MMPT$ **then**
        Sets the status of these pseudonyms to 0 and 1 respectively;
        Sets expiration time t'_{new} for $PID_{V_{new}}$;
        Sends$(E(E(H(PID_{V_{new}})\| t'_{new}), PU_{RSU}), t_s), S_k)$ to V;
    else
        Does not update the pseudonyms and ignores the message;
    end if
else
    Drops the received message;
end if

3.5. Updating the status of pseudonym of a vehicle by RSU

Each vehicle is supposed to change its pseudonym frequently to ensure privacy. The US-based SAE J2735 standard [17] recommends changing pseudonym every 120 s or after 1 km distance traveled (whichever comes last), while the European standard ETSI TS 102 867 [18] recommends changing pseudonym every five minutes. While a vehicle is parked, it is probably not necessary to change pseudonym that frequently. So, a vehicle needs 720 pseudonyms in 24 h and 262,800 pseudonyms in 1 year according to the US-based SAE J2735 standard. In CREASE, we assume the vehicle's OBU is loaded with sufficient number of pseudonyms so that it will not need to reuse a pseudonym within a year. The size of each pseudonym is 16 bytes. Therefore, a vehicle requires approximately 4 MB of storage for storing its pseudonyms. We assume that the vehicles have enough storage capability to store its pseudonyms considering the current hardware capabilities. RSUs in our scheme assist the vehicles in their region to change pseudonyms by attaching an expiration time for each pseudonym. The expiration time indicates when a vehicle needs to change its current pseudonym. Once this expiration time elapses, the vehicle will again communicate with its RSU to activate a new pseudonym from the pool of pseudonyms received from its home RTA during initial registration. Considering that an RSU possesses more powerful computation and storage capabilities than a vehicle, we assume RSUs compute each request message efficiently. RSU determines the time when all the vehicles in its region need to perform the pseudonym change. **Thus, vehicles within the same RSU's region change their pseudonym simultaneously, resulting in reducing the chance of linkability between the new pseudonym and the old pseudonym**. We assume that the clocks of the TA, RTAs, RSUs, and Vehicles (OBUs) are loosely synchronized. When the validity time of a vehicle's current pseudonym $PID_{V_{curr}}$ is about to expire, it randomly selects a new pseudonym $PID_{V_{new}}$ from the set of pseudonyms allocated to it and informs the RSU securely about the new pseudonym it wants to use. After receiving this new pseudonym, the RSU concatenates the public key PU_V of the vehicle with $PID_{V_{curr}}$ and $PID_{V_{new}}$ and looks up into its MMPT. RSU sets the status of the pseudonyms $PID_{V_{curr}}$ and $PID_{V_{new}}$ to 0 and 1 in the MMPT respectively after authenticating the message. The RSU also sets expiration time t'_{new} for $PID_{V_{new}}$ and sends it to V. The detailed description of how pseudonyms are changed is presented in **Algorithm 3**.

Algorithm 4: When a Vehicle V moves from RSU_i to RSU_j

After verifying the authenticity of the RSU_j as described in Algorithm 2, V sends ($E(PID_{V_{curr}}$, ($E(H(PID_{V_{curr}}) \| t_{new})$, PR_{RSU_i}), PU_V, t_s), PU_{RSU_j}) to RSU_j
When RSU_j receives the above message from V

```
Decrypts the message using its private key PR_RSU_j;
Checks the freshness of the received message using t_s;
Case 1: RSU_j is registered with V's home RTA
if t_s is valid then
    Verifies the signature of RSU_i;
    Retrieves t_new;
    if t_new is valid then
        Retrieves H(PID_V_curr) and checks it against the hash of received PID_V_curr;
        if the hash values match then
            if PID_V_curr ∈ MMPT then
                Sets the status of the PID_V_curr to 1;
                Sets new expiration time t'_new for PID_V_curr;
            else
                Ignores the update message;
            end if
        else
            PID_V_curr is not valid;
        end if
    end if
end if
Case 2: RSU_j is not registered with V's home RTA
if t_s is valid then
    Gets the set of pseudonyms allocated to V, PU_V, and PU_RSU_i from its home RTA through a secure protocol such as TLS;
    RSU_j verifies the signature of RSU_i;
    Next, retrieves t_new;
    if t_new is valid then
        Retrieves H(PID_V_curr) and checks it against the hash of received PID_V_curr;
        if the hash values match then
            Inserts the pseudonyms of the V into its MMPT;
            Sets the status of all pseudonyms to 0 except PID_V_curr which is set to 1;
            Sets expiration time t'_new for PID_V_curr;
        else
            PID_V_curr is not valid;
        end if
    end if
end if
```

3.5.1. When a vehicle V moves from one RSU's region to another RSU's region

When a vehicle V moves from the region covered by one RSU RSU_i to the region covered by another RSU RSU_j, V first verifies the authenticity of the RSU_j as described in **Algorithm 2**. Next, V sends following message to RSU_j to authenticate itself for communication:

$V \rightarrow RSU_j$: ($E(PID_{V_{curr}}$, ($E(H(PID_{V_{curr}}) \| t_{new})$, PR_{RSU_i}), PU_V, t_s), PU_{RSU_j})

The following two cases arise.

Case 1: RSU_j is registered with the V's home RTA: RSU_j, upon receiving the above message, uses received t_s to check the freshness of the message and verifies the signature of the RSU_i. If the verification is successful, then RSU_j verifies the authenticity of the current pseudonym $PID_{V_{curr}}$ of the V as described in **Algorithm 2**.

Case 2: RSU_j is not in the region covered by V's home RTA: The RSU_j first checks the freshness of the received message using t_s. Considering the assumption that RSUs registered under the same RTA know the public keys of each other, RSU_j forwards the received message to its (RSU_j's) home RTA. Next, RSU_j's home RTA communicates with the V's home RTA and gets the set of pseudonyms allocated to V, public key of V PU_V, and public key of RSU_i PU_{RSU_i}. RSU_j's home RTA sends the required credentials to the all RSUs in its region using a secure protocol such as TLS.

After obtaining the public key of RSU_i, RSU_j verifies the received hash value from V as described in **Algorithm 2**. Next, RSU_j inserts the set of pseudonyms of V concatenating with PU_V along with their status in its MMPT.

After authenticating V, RSU_j sets the new expiration time t'_{new} for $PID_{V_{curr}}$ and sends the following message to the V:
$RSU_j \rightarrow V$: $(E(ID_{RSU_j}, (E(H(PID_{V_{curr}}) \| t'_{new}), PR_{RSU_j}), S'_k, G'_k, t_s), PU_V)$
where, G'_k is the group key shared by all authenticated vehicles in the region of RSU_j and S'_k is the shared symmetric key between V and RSU_j. Detailed description of the above discussion is presented in **Algorithm 4**.

4. Performance evaluation

In this section, we first analyze the security of CREASE. Then, we present a formal proof of correctness of CREASE using BAN logic [19]. Next, we verify the security of CREASE using SPAN [20] and AVISPA [21] tools. Finally, we compare CREASE with LIAP [22], ASPA [23], and NERA [24] protocols with respect to security features and protocol overhead.

4.1. Security analysis

In this subsection, we discuss the security features of CREASE.

4.1.1. Mutual authentication

Under CREASE, a vehicle V and RSU authenticate each other before communicating with each other as follows. After obtaining the Missing Hash Values (MHVs), MHT root signed by RTA $root_{signbyRTA}$, public key of RTA signed by TA $PU_{RTA signbyTA}$, and public key of the RSU PU_{RSU}, the vehicle recalculates the root value of the MHT and compares it with the received $root_{signbyRTA}$. If the two values are equal PU_{RSU} is considered authentic. Next, the vehicle sends a message containing its initial pseudonym $PID_{V_{initial}}$, signed by the RTA along with pseudonym expiration time $E((H(PID_{V_{initial}}) \| t_{exp}), PR_{RTA})$. Upon receiving this message, the RSU calculates the hash of received $PID_{V_{initial}}$ and compares this calculated hash value with received hash value $H(PID_{V_{initial}})$. If these two hash values are equal, then RSU looks up into the MMPT and sets the status of $PID_{V_{initial}}$ to 1. After mutual authentication, RSU sends a symmetric key S_k and a group key G_k to the vehicle. The group key G_k is used by all vehicles authenticated by the same RSU for secure group communication. All vehicles authenticated by the RSU in its region get the group key G_k; the group key is being used by all vehicles under an RSU and hence we do not need to worry about forward and backward secrecy.

4.1.2. Vehicle anonymity and conditional privacy

A sender uses only its pseudonym in all communication, which ensures that the receivers cannot obtain the sender's real identity. Receivers authenticate the sender based on temporary credentials. In CREASE, vehicles use a pseudonym and pseudonym expiration time to send messages. Receivers authenticate the sender vehicle based on the pseudonym and its expiration time. The real identity of a vehicle is never used in communication. The RTA only knows the real identity of the vehicle. The pseudonym of a vehicle is resolvable to its real identity only by its home RTA and hence conditional privacy is preserved.

4.1.3. Unlinkability

Unlinkability requires that an adversary cannot link messages sent with two different pseudonyms by the same vehicle. In CREASE, a vehicle V uses its initial pseudonym $PID_{V_{initial}}$ for mutual authentication with the first RSU it encounters after registration. At that time, the RSU sets a new expiration time t_{new} for $PID_{V_{initial}}$. When this expiration time is about to expire, V communicates with the RSU using the symmetric key S_k established between the vehicle and RSU during the mutual authentication process to activate a new pseudonym from the pool of pseudonyms allocated to it. **RSU assists vehicles in its region to change their pseudonym simultaneously and frequently by assigning the same expiration time for the currently used pseudonyms of all vehicles in its region. Therefore, CREASE reduces the chance for linking two messages sent by the same vehicle with two different pseudonyms because vehicles change pseudonyms simultaneously.**

4.1.4. Non-repudiation

All messages sent by a vehicle contain its current pseudonym and the hash of its current pseudonym $PID_{V_{curr}}$ along with the pseudonym expiration time signed by its RSU $E((H(PID_{V_{curr}}) \| t_{new}), PR_{RSU})$ in CREASE. The receiver firstly verifies the signature of the RSU. Next, it checks the pseudonym expiration time. If it is valid then it computes the hash of the received pseudonym and checks it against the received $H(PID_{V_{curr}})$. If the two hash values are same, then the sender is considered authentic. Since a vehicle uses pseudonym from its stored set of pseudonym and registers it with an RSU for communication, the vehicle cannot deny the messages sent by it. Besides, it is not possible for an attacker to forge the signature of the RSU.

4.1.5. Resistance to replay attacks

The message generation timestamp t_s is encrypted along with all messages in CREASE to resist the replay attack. The TA, RTAs, RSUs, and Vehicles' clocks are assumed to be loosely synchronized (this can be achieved using GPS) in our scheme. Due to this addition of t_s, each entity can detect whether the message is fresh enough to prevent a replay attack.

Table 4
BAN logic notation.

Notation	Description
$P\| \equiv X$	P believes X
$P\| \sim X$	P once said X
$P \lhd X$	P sees X
$P \Rightarrow X$	P controls X
$P \overset{S_k}{\longleftrightarrow} Q$	Only P and Q know the shared secret key S_k
$\#(X)$	X is fresh
$\{X\}_k$	X is encrypted with the key k
$\wp\kappa(P, K_P)$	P has public key K_P
$\Pi(K_P^{-1})$	P has private key K_P^{-1}
$\sigma(X, K_P^{-1})$	X signed with private key K_P^{-1}

4.1.6. Resistance to Sybil attacks

A Sybil attack occurs when a malicious vehicle uses multiple pseudonyms in parallel to impersonate a number of vehicles. Therefore, the number of pseudonyms and their validity period that a vehicle can use should be limited. In CREASE, each vehicle's OBU is loaded with a set of pseudonyms, and the RTA signs one initial pseudonym along with the pseudonym expiration time $E((H(PID_{V_{initial}}) \parallel t_{exp}), PR_{RTA})$. When a vehicle enters an RSU's region after registration with an RTA, it uses this initial pseudonym for authentication. The RSU sets a new expiration time t_{new} for $PID_{V_{initial}}$ and sends it to the vehicle. When the t_{new} is about to expire, the vehicle communicates with the RSU to activate a new pseudonym from its pool of pseudonyms. Then the RSU activates and signs a new pseudonym for the vehicle along with the expiration time for that pseudonym $E((H(PID_{V_{new}}) \parallel t'_{new}), PR_{RSU})$. Therefore, only one pseudonym of a vehicle is valid at a time in CREASE and hence it resists the Sybil attacks.

4.1.7. Resistance to message injection attack

Under CREASE, when a vehicle V enters an RSU's region after registration with an RTA, it first verifies the signatures of TA $PU_{RTA\,signbyTA}$ and RTA $root_{signbyRTA}$. Next, it compares the MHT root value signed by the RTA with the MHT root value calculated from the MHVs information received in the beacon message. If these two values are equal, then the RSU is considered legitimate; otherwise, a message injection attack is detected. After authenticating the RSU, V sends its initial pseudonym $PID_{V_{initial}}$ signed by its home RTA along with its expiration time t_{exp}. The receiver RSU verifies the signature of the RTA and then it checks the t_{exp}. If t_{exp} is valid then it computes the hash of the received pseudonym and checks it against the received $H(PID_{V_{initial}})$. If the two values are not equal, then a message injection attack is detected. It is not possible for an attacker to forge the signature of TA or RTA without knowing their private key.

4.2. Formal proof of correctness of CREASE based on BAN logic

Borrows, Abadi, and Needham (BAN) logic [19] is a popular authentication protocols analysis model to formally verify the correctness of authentication protocols [15,25–29]. In this subsection, we analyze CREASE using BAN logic and the PKI-based extended BAN logic [30] and demonstrate its correctness. First, we present a brief overview of the BAN logic and the inference rules for BAN logic in this subsection. Next, we discuss a formal idealization of the proposed CREASE scheme's messages, the list of initial assumptions, goals of our scheme, and logical derivation to achieve the goals.

4.2.1. Notations used in BAN logic

The list of BAN logic notations used in this paper are presented in Table 4.

4.2.2. Inference rules for BAN logic

The inference rules for BAN logic that we use to derive our goals are stated below:

- *R1: Message meaning rules*

$$\frac{P| \equiv \wp\kappa(Q, K_Q), P| \equiv \Pi(K_Q^{-1}), P \lhd \sigma(X, K_Q^{-1})}{P| \equiv Q| \sim X} \tag{1}$$

Let us consider that P and Q are two communication entities. The above rule states that, P believes Q generated X, if it believes that Q has public key K_Q and private key K_Q^{-1}, and P sees X encrypted with K_Q^{-1}.

$$\frac{P| \equiv \wp\kappa(P, K_P), P| \equiv \Pi(K_P^{-1}), P \lhd S\{(X, Q)\}_{K_P}}{P| \equiv Q| \sim X} \tag{2}$$

Here, $S(X, Q)$ specifically says "message X together with Q as the stated sender of the message". The above rule deals with situation where the stated sender is concealed within the encrypted message. The above rule states that P believes that Q once

said X, if P believes that it has public key K_P and private key K_P^{-1}, and P sees $S(X,Q)$ encrypted with K_P for which P is the intended recipient.

$$\frac{P| \equiv P \xleftrightarrow{S_k} Q, P \lhd \{X\}S_k}{P| \equiv Q| \sim X} \tag{3}$$

The above rule states that if P believes that Q shared a key S_k with it and P sees X encrypted with S_k, then P believes Q once said X.

- *R2: Nonce Verification rule*

$$\frac{P| \equiv \#(X), P| \equiv Q| \sim X}{P| \equiv Q| \equiv X} \tag{4}$$

The above rule states the freshness of the message with respect to the time. P believes Q believes in the freshness of X, if P believes that X could have been uttered only recently and Q once said X.

- *R3: Jurisdiction rule*

$$\frac{P| \equiv Q \Rightarrow X, P| \equiv Q| \equiv X}{P| \equiv X} \tag{5}$$

The above rule states that if P believes that Q controls X and P believes Q believes X, then P trusts Q on the truth of X.

- *R4: Freshness rule*

$$\frac{P| \equiv \#(X)}{P| \equiv \#(X,Y)} \tag{6}$$

This rule is concerned with the freshness of message. It states that if one of the components of a message is fresh, then a combination of components of a message must also be fresh.

- *R5: Sees rule*

$$\frac{P \lhd (X,Y)}{P \lhd (X)} \tag{7}$$

$$\frac{P| \equiv \wp\kappa(Q,K_Q), P| \equiv \Pi(K_Q^{-1}), P \lhd \{X\}_{K_Q^{-1}}}{P \lhd X} \tag{8}$$

$$\frac{P| \equiv \wp\kappa(Q,K_Q), P| \equiv \Pi(K_Q^{-1}), P \lhd \sigma(X,K_Q^{-1})}{P \lhd X} \tag{9}$$

$$\frac{P| \equiv \wp\kappa(P,K_P), P| \equiv \Pi(K_P^{-1}), P \lhd \{X\}_{K_P^{-1}}}{P \lhd X} \tag{10}$$

$$\frac{P| \equiv P \xleftrightarrow{S_k} Q, P \lhd \{X\}_{S_k}}{P \lhd X} \tag{11}$$

This rule states that if P sees a message, then it can see the components of the message as P knows the necessary keys.

4.2.3. Protocol idealization

For the formal analysis, the messages exchanged between RSU and Vehicle to achieve mutual authentication is simplified and formally idealized as follows:

When a vehicle V enters an area covered by RSU_i after registration following messages are exchanged for mutual authentication:

- M1: RSU_i broadcasts the message: $\langle ID_{RSU_i}, \sigma(K_{RTA}, K_{TA}^{-1}), MHVs, \sigma(MHTroot, K_{RTA}^{-1}), K_{RSU_i}, t_s \rangle$
- M2: $V \rightarrow RSU_i$: $\langle \{PID_{V_{initial}}, \{H(PID_{V_{initial}}) \parallel t_{exp}\}_{K_{RTA}^{-1}}, K_V, t_s\}_{K_{RSU_i}} \rangle$
- M3: $RSU_i \rightarrow V$: $\langle \{ID_{RSU_i}, \{H(PID_{V_{initial}}) \parallel t_{new}\}_{K_{RSU_i}^{-1}}, RSU_i \xleftrightarrow{S_k} V, G_k, t_s\}_{K_v} \rangle$

When a vehicle moves from RSU_i to RSU_j following messages are exchanged for mutual authentication:

- M4: RSU_j broadcasts the message: $\langle ID_{RSU_j}, \sigma(K_{RTA}, K_{TA}^{-1}), MHVs, \sigma(MHTroot, K_{RTA}^{-1}), K_{RSU_j} t_{s_{RSU_j}} \rangle$
- M5: $V \rightarrow RSU_j$: $\langle \{PID_{V_{curr}}, \{H(PID_{V_{curr}}) \parallel t_{new}\}_{K_{RSU_i}^{-1}}, K_V, t_s\}_{K_{RSU_j}} \rangle$
- M6: $RSU_j \rightarrow$ V : $\langle \{ID_{RSU_j}, \{H(PID_{V_{curr}}) \parallel t'_{new}\}_{K_{RSU_j}^{-1}}, RSU_j \xleftrightarrow{S'_k} V, G'_k, t_s\}_{K_v} \rangle$

4.2.4. Initial assumptions

In CREASE, the TA and RTA are trusted by both RSUs and vehicles. RTAs, RSUs, and vehicles know the public key of TA (K_{TA}). RTA is registered with the TA and gets its public and private key pairs (K_{RTA}, K_{RTA}^{-1}). Every RTA generates the public and private key pairs for each RSU (K_{RSU}, K_{RSU}^{-1}) and vehicle (K_V, K_V^{-1}) registered under it. Each vehicle's OBU is loaded with a set of pseudonyms and an initial pseudonym ($PID_{V_{initial}}$) signed by its home RTA. The initial assumptions of the protocol is summarized as follows:

- A1: $V| \equiv \wp\kappa(TA, K_{TA})$; A2: $V| \equiv \Pi(K_{TA}^{-1})$; A3: $RSU_i| \equiv \wp\kappa(RTA, K_{RTA})$;
- A4: $RSU_i| \equiv \Pi(K_{RTA}^{-1})$; A5: $RSU_j| \equiv \wp\kappa(RTA, K_{RTA})$; A6: $RSU_j| \equiv \Pi(K_{RTA}^{-1})$;
- A7: $V| \equiv \wp\kappa(RTA, K_{RTA})$; A8: $V| \equiv \Pi(K_{RTA}^{-1})$; A9: $RSU_j| \equiv \wp\kappa(RSU_i, K_{RSU_i})$;
- A10: $RSU_j| \equiv \Pi(K_{RSU_i}^{-1})$; A11: $V| \equiv \wp\kappa(V, K_V)$; A12: $V| \equiv \Pi(K_V^{-1})$;
- A13: $RSU_i| \equiv RTA \Rightarrow MHTroot$; A14: $RSU_j| \equiv RTA \Rightarrow MHTroot$; A15: $V| \equiv RTA \Rightarrow MHTroot$
- A16: $RSU_i| \equiv RSU_i \Rightarrow (RSU_i \overset{S_k}{\longleftrightarrow} V, G_k)$; A17: $V| \equiv RSU_i \Rightarrow (RSU_i \overset{S_k}{\longleftrightarrow} V, G_k)$
- A18: $RSU_j| \equiv RSU_j \Rightarrow (RSU_j \overset{S_k'}{\longleftrightarrow} V, G_k')$; A19: $V| \equiv RSU_j \Rightarrow (RSU_j \overset{S_k'}{\longleftrightarrow} V, G_k')$
- A20: $RSU_i| \equiv RTA \Rightarrow (H(PID_{V_{initial}}) \parallel t_{exp})$; A21: $RSU_j| \equiv RTA \Rightarrow (H(PID_{V_{initial}}) \parallel t_{exp})$
- A22: $RSU_j| \equiv RSU_i \Rightarrow (H(PID_{V_{curr}}) \parallel t_{new})$; A23: $RSU_i| \equiv \#(t_s)$
- A24: $RSU_j| \equiv \#(t_s)$; A25: $V| \equiv \#(t_s)$

4.2.5. Goal of CREASE

The goals of CREASE are:

- G1: $V| \equiv RTA| \equiv MHTroot$; G2: $V| \equiv MHTroot$; G3: $RSU_i| \equiv RTA| \equiv (H(PID_{V_{initial}}) \parallel t_{exp})$;
- G4: $RSU_i| \equiv (H(PID_{V_{initial}}) \parallel t_{exp})$; G5: $V| \equiv RSU_i| \equiv (RSU_i \overset{S_k}{\longleftrightarrow} V, G_k)$; G6: $V| \equiv (RSU_i \overset{S_k}{\longleftrightarrow} V, G_k)$;
- G7: $RSU_j| \equiv RSU_i| \equiv (H(PID_{V_{curr}}) \parallel t_{new})$; G8: $RSU_j| \equiv (H(PID_{V_{curr}}) \parallel t_{new})$;
- G9: $V| \equiv RSU_i| \equiv (RSU_i \overset{S_k'}{\longleftrightarrow} V, G_k')$; G10: $V| \equiv (RSU_i \overset{S_k'}{\longleftrightarrow} V, G_k')$;

4.2.6. Derivation of the above goals

On the basis of logical postulates and initial assumptions, we derive the above goals as follows.
From message M1, we deduce G1 and G2 as follows:

- D1. $V \lhd \langle ID_{RSU_i}, \sigma(K_{RTA}, K_{TA}^{-1}), MHVs, \sigma(MHTroot, K_{RTA}^{-1}), K_{RSU_i}, t_s \rangle$
- D2. V $\lhd \sigma(K_{RTA}, K_{TA}^{-1})$ (From D1 and $R5$ (7)); D3. V $| \equiv \#(\sigma(K_{RTA}, K_{TA}^{-1}))$ (From A25 and $R4$)
- D4. V $| \equiv TA| \sim K_{RTA}$ (From A1, A2, and $R1$ (1)); D5. V $| \equiv \#\, \sigma(MHTroot, K_{RTA}^{-1})$ (From $R4$ and A21)
- D6. V $| \equiv RTA| \sim MHTroot$ (From D11, A9, A10, and $R1$ (1))
- D7. V $| \equiv RTA | \equiv MHTroot$ (From D5, D6, and $R2$) ..... **(G1)**
- D8.V $| \equiv MHTroot$ (From A15, D7, and $R3$) ..... **(G2)**

From message M2, we deduce G3 and G4 as follows:

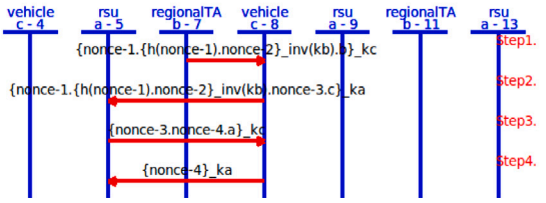
- D9. $RSU_i \lhd \langle \{PID_{V_{initial}}, \{H(PID_{V_{initial}}) \parallel t_{exp}\}_{K_{RTA}^{-1}}, K_V, t_s\}_{K_{RSU}} \rangle$
- D10. $RSU_i \lhd \langle \{H(PID_{V_{initial}}) \parallel t_{exp}\}_{K_{RTA}^{-1}} \rangle$ (From D9 and $R5$ (7))
- D11. RSU_i $| \equiv \#\, (H(PID_{V_{initial}}) \parallel t_{exp})$ (From A23 and $R4$)
- D12. RSU_i $| \equiv RTA| \sim (H(PID_{V_{initial}}) \parallel t_{exp})$ (From A3, A4, and $R1$ (1))
- D13. $RSU_i| \equiv RTA$ $| \equiv (H(PID_{V_{initial}}) \parallel t_{exp})$ (From D11, D12, and $R2$) ..... **(G3)**
- D14. $RSU_i| \equiv (H(PID_{V_{initial}}) \parallel t_{exp})$ (From A18, D13, and R3) ..... **(G4)**

From message M3, we deduce G5 and G6 as follows:

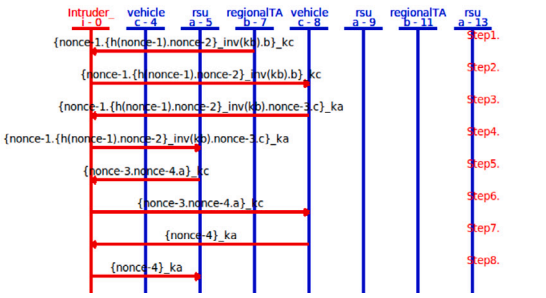
- D15. $V \lhd \langle \{ID_{RSU_i}, \{H(PID_{V_{initial}}) \parallel t_{new}\}_{K_{RSU_i}^{-1}}, RSU_i \overset{S_k}{\longleftrightarrow} V, G_k, t_s\}_{K_v} \rangle$
- D16. $V| \equiv \#\, (RSU_i \overset{S_k}{\longleftrightarrow} V, G_k)$ (From A25 and $R4$)
- D17. $V| \equiv RSU_i$ $| \sim (RSU_i \overset{S_k}{\longleftrightarrow} V, G_k)$ (From A11, A12, D15, and $R1$ (2))
- D18. $V| \equiv RSU_i$ $| \equiv (RSU_i \overset{S_k}{\longleftrightarrow} V, G_k)$ (From D16, D17, and $R2$) ..... **(G5)**
- D19.$V| \equiv (RSU_i \overset{S_k}{\longleftrightarrow} V, G_k)$ (From A17, D18, and $R3$) ..... **(G6)**

From message M4, we deduce G1 and G2 as follows:

- D20. $V \lhd \langle ID_{RSU_j}, \sigma(K_{RTA}, K_{TA}^{-1}), MHVs, \sigma(MHTroot, K_{RTA}^{-1}), K_{RSU_j}, t_s \rangle$
- D21. V $\lhd \sigma(K_{RTA}, K_{TA}^{-1})$ (From D20 and $R5$ (7))



(a) Message sequence chart of our scheme generated by SPAN and AVISPA tool



(b) Message sequence chart of our scheme in the presence of intruder generated by SPAN and AVISPA tool

Fig. 5. Message sequence chart schemes generated by SPAN and AVISPA.

- D22. V $\mid\equiv \#\,(\sigma(K_{RTA}, K_{TA}^{-1}))$ (From A25 and $R4$)
- D23. V $\mid\equiv TA\mid\sim K_{RTA}$ (From A1, A2, and $R1$ (1))
- D24. V $\mid\equiv \#\,\sigma(MHTroot, K_{RTA}^{-1})$ (From $R4$ and A25)
- D25. V $\mid\equiv RTA\mid\sim MHTroot$ (From D20, A9, A10, and $R1$ (1))
- D26. V $\mid\equiv RTA \mid\equiv MHTroot$ (From D24, D25, and $R2$) ..... **(G1)**
- D27.V $\mid\equiv MHTroot$ (From A15, D26, and $R3$) ..... **(G2)**

From message M5, we deduce G7 and G8 as follows:

- D28. $RSU_j \lhd \langle\{PID_{V_{curr}}, \{H(PID_{V_{curr}}) \parallel t_{new}\}_{K_{RSU_j}^{-1}}, K_V, t_V\}_{K_{RSU_j}^{-1}}\rangle$
- D29. $RSU_j \lhd \langle\{H(PID_{V_{curr}}) \parallel t_{new}\}_{K_{RSU_i}^{-1}}\rangle$ (From D28 and $R5$ (7))
- D30. $RSU_i \mid\equiv \#\,(H(PID_{V_{curr}}) \parallel t_{new})$(From A24 and $R4$)
- D31. $RSU_j \mid\equiv RSU_i\mid\sim (H(PID_{V_{curr}}) \parallel t_{new})$ (From A5, A6, and $R1$ (1))
- D32. $RSU_j\mid\equiv RSU_i \mid\equiv (H(PID_{V_{curr}}) \parallel t_{new})$ (From D30, D31, and $R2$) ..... **(G7)**
- D33. $RSU_j \mid\equiv (H(PID_{V_{curr}}) \parallel t_{new})$ (From A21, D32, and $R3$) ..... **(G8)**

From message M6, we deduce G9 and G10 as follows:

- D34. $V \lhd \langle\{ID_{RSU_j}, \{H(PID_{V_{curr}}) \parallel t'_{new}\}_{K_{RSU_j}^{-1}}, RSU_j \overset{S'_k}{\longleftrightarrow} V, G'_k, t_{RSU_j}\}_{K_v}\rangle$
- D35. $V\mid\equiv \#\,(RSU_j \overset{S'_k}{\longleftrightarrow} V, G'_k)$ (From A25 and $R4$)
- D36. $V\mid\equiv RSU_j \mid\sim (RSU_j \overset{S'_k}{\longleftrightarrow} V, G'_k)$ (From A11, A12, D34, and $R1$ (2))
- D37. $V\mid\equiv RSU_i \mid\equiv (RSU_j \overset{S'_k}{\longleftrightarrow} V, G'_k)$ (From D35, D36, and $R2$) ..... **(G9)**
- D38.$V\mid\equiv (RSU_j \overset{S'_k}{\longleftrightarrow} V, G'_k)$ (From A19, D37, and $R3$) ..... **(G10)**

The above BAN logic analysis shows that our scheme achieves all the goals (G1–G10) and vehicles can get the correct symmetric key and group key after a mutual authentication process for secure communication.

4.3. Automated verification of CREASE based on SPAN ad AVISPA tools

SPAN (Security Protocol ANimator) [20] and AVISPA (Automated Validation of Internet Security Protocols and Applications) [21] tools are widely used in literature [31–34] for analyzing the security of the protocols. In this research, we also verify the security of CREASE against replay attack, man-in-the-middle attack, and impersonation attack using SPAN and AVISPA tools.

In our model, we consider three basic roles which we call *rsu*, *regionalTA*, and *vehicle*, and are denoted by *a*, *b*, and *c* respectively. Here, *ka*, *kb*, and *kc* represent the public keys of the *rsu*, *regionalTA*, and *vehicle* respectively, and *h* represents the cryptographic hash function. In the proposed protocol, *regionalTA* first activates the start signal and sends a message to *vehicle* containing hash of initial pseudonym *nonce*-1 and the expiration time *nonce*-2 of *nonce*-1 which are encrypted with its private key *inv*(*kb*). The encrypted hash value and the initial pseudonym expiration time are used by *rsu* to authenticate the initial pseudonym *nonce*-1 of the *vehicle*. Fig. 5(a) shows the message sequence chart of the proposed scheme using SPAN and AVISPA tools.

The message sequence chart in the presence of an intruder is presented in Fig. 5(b). This sequence chart demonstrates that the intruder is unable to read and/or modify the messages. The intruder is only able to listen and forward the messages. We describe the sequence of exchanged messages in presence of an intruder *i* as follows:

Table 5
Comparison of Security Features.

	CREASE	NERA [24]	ASPA [23]	LIAP [22]
Mutual Authentication	Yes	Yes	No	Yes
Vehicle Anonymity	Yes	Yes	Yes	Yes
Unlinkability	Yes	No	Yes	Yes
Non-repudiation	Yes	Yes	Yes	Yes
Resistance to Replay Attack	Yes	No	Yes	Yes
Resistance to Sybil Attack	Yes	No	No	No
Resistance to Message Injection Attack	Yes	Yes	Yes	Yes

Step 1: The $regionalTA$ initiates session and sends a message containing the $vehicle$'s initial pseudonym $nonce$-1 and the expiration time $nonce$-2 of the $nonce$-1 (where, h(nonce-1) and $nonce$-1 is encrypted with its private key $inv(kb)$) to the $vehicle$.

Step 2: Since the intruder does not know the private key of the $vehicle$, the intruder only listens to the message.

Step 3: The $vehicle$ sends the received message along with a nonce to the rsu.

Step 4: The intruder is unable to read and/or modify the message as it is encrypted using public key ka of rsu. The intruder only views the message and passes it to the rsu.

Step 5: Upon receiving the message, rsu retrieves the hash value and initial pseudonym expiration time using public key kb of the $regionalTA$. Firstly, rsu verifies the validity of the initial pseudonym expiration time. Next, it recalculates the hash of the initial pseudonym received in the message to verify the initial pseudonym of the $vehicle$. Then it sends a new nonce to the $vehicle$, along with the one it received.

Step 6: The intruder only listens the message and passes it to the $vehicle$.

Step 7: The $vehicle$ decrypts the message using its private key $inv(ka)$ and retrieves the nonces. Next, the $vehicle$ sends back the received nonce to the rsu.

Step 8: The intruder is only able to view the message but unable to read or modify it. He/she passes the message to the rsu.

4.4. Comparison with other related protocols

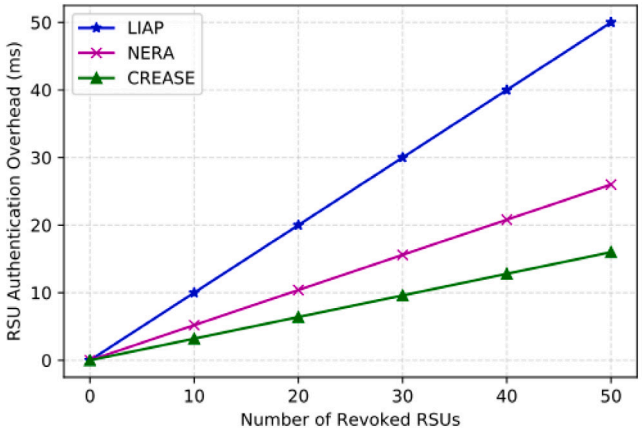
In this section, we compare the security features and the protocol overheads of CREASE with those of the LIAP [22], ASPA [23], and NERA [24] schemes.

4.4.1. Comparison of security features

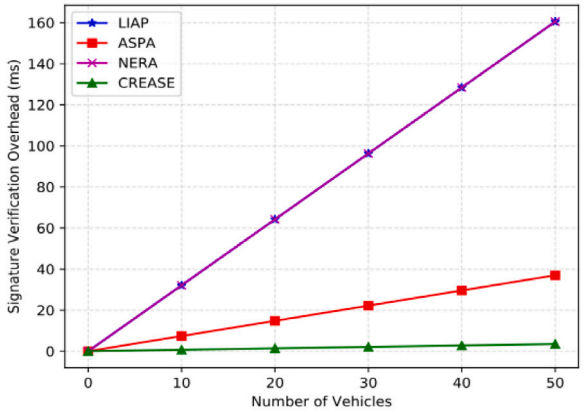
In CREASE, RSUs provide pseudonym expiration time t_{exp} for each authentic vehicle within its transmission range. Each vehicle uses its current pseudonym $PID_{V_{curr}}$ signed by the RSU $E((H(PID_{V_{curr}}) \| t_{exp}), PR_{RSU})$ for communication. It is not possible for an attacker to forge the signature of the RSU. Besides, the message generation timestamp is encrypted along with the message in our scheme to resist the replay attack. We assume that the clocks of the TA, RTAs, RSUs and vehicles are loosely synchronized (this can be done using GPS). In contrast, NERA [24] is susceptible to message replay attacks since there is no timestamp associated with the messages. In this scheme, RSU generates a set of pseudonym for all vehicles in its region. In LIAP [22], an RSU manages and assigns a local master key to every vehicle in its region after the mutual authentication. A vehicle uses this master key to generate pseudonyms for VANET communication. In ASPA [23], vehicles get multiple short-time pseudonym certificates from the Pseudonym Provider (PP) and they are all valid at the same time interval. However, LIAP, ASPA, and NERA are not secure against Sybil attacks as multiple pseudonyms of a vehicle are valid at the same interval. We present the comparison of the security features of the CREASE with other schemes in Table 5. The results show that CREASE is more secure than LIAP, ASPA, and NERA.

4.4.2. Protocol overhead analysis

In CREASE, a vehicle first verifies the signature of the TA and RTA. Then, the vehicle calculates the MHT root value using the Missing Hash Values (MHVs) received in the beacon message of the RSU for authentication. On an Intel Core 2 1.83 GHz processor machine running Windows Vista in 32-bit mode, RSA 2048 signature verification takes 0.16 ms, and SHA-256 hash computation takes 111 MiB/s using Crypto++ 5.6.0 [35]. With much lower computation costs for the hash function calculation, the total cost of RSU authentication in CREASE depends mainly on RSA signature verification. On the other hand, a vehicle uses a linear search to check the RSU certificate revocation list (RCRL) for authentication in the LIAP scheme [22]. NERA scheme [24] uses the bilinear pairing and Map-To-Point operations, which cause overhead in RSU authentication. Since the authentication of RSUs by vehicles is not taken into consideration in ASPA protocol [23]. Therefore, Fig. 6(a) presents the comparison of RSU authentication overhead involved in CREASE with that of LIAP and NERA. There is a significant increase in RSU authentication cost when the number of revoked RSUs increases in LIAP. On the contrary, the RSU authentication cost is low under NERA and very low under CREASE.



(a) RSU Authentication Overhead Comparison



(b) Signature Verification Overhead Comparison

Fig. 6. Authentication and signature verification overhead comparison.

Fig. 6(a) shows that the authentication overhead under LIAP is almost three times as much as CREASE when the number of revoked RSUs reaches 30.

In CREASE, RSU first decrypts the message from a vehicle using its private key PR_{RSU}. Then it verifies the signature of the RTA. Then, it concatenates the public key of the vehicle with the vehicle's initial pseudonym and looks up into the MMPT. In both LIAP [22] and NERA [24], RSU checks the Vehicle Certificate Revocation List (VCRL) after decrypting the message from vehicle. Next, it verifies the signature of the Certificate Authority (CA) to authenticate the vehicle. The signature verification overhead in LIAP and NERA protocol using bilinear pairing is $T_{mul}+T_{mtp}+3T_{par}$. Here, T_{mul} denotes the time of performing one point multiplication (T_{mul} = 0.39 ms), T_{mtp} denotes the time for performing a Map-To-Point hash operation (T_{mtp}=0.09 ms), and T_{par} denotes the time for performing a pairing operation (T_{par} = 3.21 ms) [22]. In ASPA protocol, a vehicle uses its initial pseudonym provided by the Vehicular Manufacturing Company (VMC) to request a longterm certificate (LTC) from the Certificate Authority (CA). CA checks the CRL to issue a LTC for the vehicle. Next, the vehicle gets a Pseudonym Certificate (PC) from the LTC Authority using the LTC. The vehicle sends a message to the Pseudonym Provider (PP) directly or through RSU for pseudonyms using the PC. After verifying the PC of the vehicle, Pseudonym Provider sends multiple pseudonyms to the vehicle. The signature verification cost in this scheme using Digital Signature Algorithm (DSA) is 0.37 ms [23].

A comparison of the signature verification overhead of CREASE with LIAP [22], NERA [24], and ASPA [23] is presented in Fig. 6(b). It is observed that CREASE has significantly lower signature verification overhead compared to LIAP, NERA, and ASPA.

For example, when the number of vehicles reaches 30, the overall signature verification cost is approximately 92 ms for both LIAP [22] and NERA [24] and 22 ms for ASPA [23], whereas it is only 4.8 ms for CREASE.

In CREASE, vehicle and RSU authenticates each other without using certificate and certificate revocation lists (CRLs). Vehicles use Missing Hash Values (MHVs) and Public key of the RSU PU_{RSU} received in the RSU's beacon message to recalculate the MHT root value. Next, the vehicle checks this hash value against the received root value of MHT signed by the RTA $root_{signby_{RTA}}$ for authentication. After authenticating the RSU, the vehicle sends its initial pseudonym signed by the RTA along with the pseudonym expiration time $E((H(PID_{V_{initial}}) \parallel t_{exp}), PR_{RTA})$, initial pseudonym $PID_{V_{initial}}$, and its public key PU_V for authentication. RSU first checks if t_{exp} is valid. After that, it computes the hash of the $PID_{V_{initial}}$ in the received message and compares it with the received $H(PID_{V_i nitial})$. If the hash values matches then $PID_{V_{initial}}$ is considered as valid. On the contrary, vehicles use long-term certificates for authentication in both LIAP [22] and ASPA [23] schemes. Next, the RSUs or Pseudonym Providers (PP) check the latest Certificate Revocation List (CRLs) of vehicles to verify the vehicles' authenticity. In NERA scheme [24] the TA revokes the malicious vehicle and adds its real ID to the CRL. In these schemes, the CA (Certificate Authority) or TA maintains a certificate revocation list (CRL) and distributes the updated CRL of vehicles to all entities in VANET. A CRL typically consists of a header, the current date, the last time it has been updated, the next time it will be updated, and a complete list of revoked certificates signed by the CA [36]. Moreover, the size of CRL increases significantly as the number of entities grows. Therefore, the use of CRLs for authentication incurs significant computation and communication overhead. Besides, the CA needs to send the CRLs very often to keep the communication updated and secure. So, the LIAP, ASPA, and NERA schemes introduce significant additional communication overhead on RSUs or Pseudonym Providers (PPs).

5. Related works

Security and privacy issues in VANET have attracted the attention from both academia and industry. Many privacy preserving authentication schemes that include pseudonym based schemes, group signature-based schemes, ID-based schemes, symmetric cryptography-based schemes, and anonymous certificate based schemes have been proposed in recent years.

Pseudonym-based schemes mostly use Public Key Infrastructure (PKI). Raya and Hubaux [37] proposed a pseudonymous scheme, where the CA generates public–private key pairs and corresponding certificates for vehicles. In this scheme, each vehicle requires to preload a huge quantity of public-key certificates. This scheme provides message authentication and conditional privacy-preservation. However, a huge storage space is needed to store certificates of all vehicles, while the CA also needs to store certificates of all vehicles.

Jiang et al. [38] proposed an anonymous batch authentication scheme (ABAH) based on hashed message authentication code (HMAC). In this scheme, they divide a large area into several domains, and each RSU manages the vehicles in its domain in a localized manner. The TA generates enough pseudonyms for each vehicle to take part in VANET. Vehicles use their pseudonym to send a join request to an RSU. The RSU checks the revocation status of the vehicle's pseudonym in the CRL for authentication. Then, the RSU sends a group key to the authenticated vehicle. In this scheme, vehicles calculate HMAC using the group key and include it in the safety-related messages for secure communication. The revocation status checking process using the CRL has associated overhead.

Wang and Yuo [22] proposed a local identity-based anonymous message authentication protocol (LIAP) using bilinear pairing. In this protocol, both vehicles and RSUs get long-term certificates from the CA during registration. A vehicle uses its long-term certificate when it enters an RSU's region for authentication. RSUs use the stored vehicle certificate revocation list (VCRL) to authenticate vehicles. Similarly, vehicles also authenticate RSUs using the RSUs certificate revocation list (RCRL). Vehicles get keys upon mutual authentication from RSUs in order to generate pseudonyms for V2V communication. However, the CA still needs to distribute RCRL and VCRL in this scheme.

Paruchuri and Durresi [39] proposed a certificate-based scheme that uses smart cards to provide anonymous authentication. The smart card stores vehicle's real identity, certificate, and required cryptographic keys. In this scheme, a vehicle uses its certificate to authenticate itself to an RSU for receiving and sending messages. The RSU generates a session key and sends it to all the vehicles that have been authenticated by it. Authenticated Vehicles under an RSU share the same session key for communication. Vehicles do not need to store the computation-intensive CRLs.

Ali et al. [23] proposed a pseudonym-based authentication scheme that allows vehicles with a valid pseudonym for communication. In this scheme, firstly, each vehicle gets an initial pseudonym from Vehicular Manufacturing Company (VMC) using the VMC's pre-loaded secret key. Next, it gets a long term certificate (LTC) from the CA, which is used by the LTC Authority to issue a Pseudonym Certificate (PC) for the vehicle. Then, a vehicle requests for pseudonyms from the Pseudonym Provider (PP) directly or through RSUs. PP sends a set of pseudonyms to the vehicle and they are all valid within the same time interval. However, this scheme is not secure against Sybil attacks as multiple pseudonyms of a vehicle are valid at the same time interval. Besides, CA uses CRL for authenticating vehicles.

Cui et al. [40] proposed a message authentication scheme based on Edge computing for VANETs. In this scheme, RSU distributes the message authentication tasks of all vehicles in its region to the Edge Computing Vehicles (ECVs). RSU verifies the feedback from the ECVs. After that, the RSU broadcasts the authentication information to all vehicles in its region using a cuckoo filter. As a result, a vehicle only needs to query the cuckoo filter to verify the authenticity of a received message. This scheme reduces redundant authentication of the same message thereby enhancing authentication efficiency. However, use of the cuckoo filter can introduce communication and computation overhead.

Bayat et al. [24] proposed an efficient RSU-based authentication scheme using bilinear pairing and Map-To-Point operation. In this scheme, a vehicle joins an RSU's region to take part in VANET communication. After mutual authentication, RSU generates a set of pseudonyms and the corresponding secret keys for each vehicle in its region. In this scheme, the real IDs of malicious vehicles are added to CRLs to ensure secure communication. Thus, it incurs authentication overhead on all RSUs. Also, message generation timestamps are not attached to every message, which makes it vulnerable to replay attacks.

To prevent linkability of two messages sent by the same vehicle using two different pseudonyms, most of the pseudonym based schemes [41–44] presented in the literature focus on the frequency at which pseudonyms are changed, or the best situation for changing pseudonym. Moreover, most of the existing schemes require a vehicle to get a new set of pseudonyms after all the previously assigned pseudonyms have been used. This is really not necessary. A vehicle can get large number of pseudonyms once and pick a random pseudonym from this set every time it wants to change its pseudonym. That is exactly what our scheme does. However, for accomplishing this the pseudonyms assigned to a vehicle need to be managed efficiently. Our scheme achieves this using the data structures MHT and MMPT for storing and changing pseudonyms efficiently and securely. Moreover, MHT and MMPT data structures facilitate RSUs in authenticating vehicles for changing pseudonyms efficiently. MHT also helps vehicles in authenticating RSUs efficiently without requiring certificates and CRLs.

6. Conclusion

We presented a novel and efficient privacy-preserving authentication scheme that leverages both Merkle Hash Tree and Modified Merkle Patricia Trie; our scheme allows vehicles to get a set of pseudonyms once and pick one random pseudonym from this set at a time and use it to preserve privacy. MHT and MMPT data structures help in managing and storing these pseudonyms efficiently for verifying the authenticity of the vehicles while at the same time preserving their privacy. MHT of public keys of RSUs also help in authenticating RSUs efficiently without certificates. Our scheme does not require the RSUs and vehicles store the certificates and CRL for authentication. Our scheme is robust against replay attacks, man-in-the-middle attacks, and impersonation attacks. We compared the security properties and protocol overhead of our scheme with those of other related protocols and also presented a formal proof of correctness.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J.B. Kenney, Dedicated short-range communications (DSRC) standards in the united states, Proc. IEEE 99 (7) (2011) 1162–1182.
[2] R.C. Merkle, Protocols for public key cryptosystems, in: Proceedings of 1980 IEEE Symposium on Security and Privacy, IEEE, 1980, pp. 122–134.
[3] J. Petit, F. Schaub, M. Feiri, F. Kargl, Pseudonym schemes in vehicular networks: A survey, IEEE Commun. Surv. Tutor. 17 (1) (2015) 228–255.
[4] D. Förster, H. Löhr, A. Grätz, J. Petit, F. Kargl, An evaluation of pseudonym changes for vehicular networks in large-scale, realistic traffic scenarios, IEEE Trans. Intell. Transp. Syst. 19 (10) (2017) 3400–3405.
[5] S. Wang, N. Yao, N. Gong, Z. Gao, A trigger-based pseudonym exchange scheme for location privacy preserving in VANETs, Peer Peer Netw. Appl. 11 (3) (2018) 548–560.
[6] S. Wang, N. Yao, A RSU-aided distributed trust framework for pseudonym-enabled privacy preservation in VANETs, Wirel. Netw. 25 (3) (2019) 1099–1115.
[7] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126.
[8] D. Eastlake, P. Jones, US secure hash algorithm 1 (SHA1), 2001, RFC 3174.
[9] R.L. Rivest, et al., RFC 1321: The MD5 Message-Digest Algorithm, Technical report, Internet Activities Board, 1992.
[10] S.S. Moni, D. Manivannan, An efficient RSU authentication scheme based on Merkle Hash Tree for VANETs, in: Proceedings of 2020 IEEE International Conference on Communications, ICC, IEEE, 2020, pp. 1–7.
[11] S. Gyawali, Y. Qian, R.Q. Hu, A privacy-preserving misbehavior detection system in vehicular communication networks, IEEE Trans. Veh. Technol. 70 (6) (2021) 6147–6158.
[12] V.-L. Nguyen, P.-C. Lin, R.-H. Hwang, Enhancing misbehavior detection in 5G vehicle-to-vehicle communications, IEEE Trans. Veh. Technol. 69 (9) (2020) 9417–9430.
[13] N.V. Abhishek, M.N. Aman, T.J. Lim, B. Sikdar, DRiVe: Detecting malicious roadside units in the internet of vehicles with low latency data integrity, IEEE Internet Things J. 9 (5) (2021) 3270–3281.
[14] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper, 151 (2014), 2014, pp. 1–32.
[15] S.S. Moni, D. Manivannan, A scalable and distributed architecture for secure and privacy-preserving authentication and message dissemination in VANETs, Internet of Things 13 (2020) 100350.
[16] Z. Lu, Q. Wang, G. Qu, H. Zhang, Z. Liu, A blockchain-based privacy-preserving authentication scheme for VANETs, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 27 (12) (2019) 2792–2801.
[17] S.V. Standard, Dedicated Short Range Communications (DSRC) Message Set Dictionary, SAE International, 2009.
[18] T.S. ETSI, 102 941 V1. 1.1—Intelligent transport systems (ITS); security; trust and privacy management, 2012, Standard, TC C-ITS.
[19] M. Burrows, M. Abadi, R.M. Needham, A logic of authentication, ACM Trans. Comput. Syst. 8 (1) (1990) 18–36.
[20] Irisa, SPAN, 2006, http://people.irisa.fr/Thomas.Genet/span/.
[21] Avispa, AVISPA, 2002, http://www.avispa-project.org.
[22] S. Wang, N. Yao, LIAP: A local identity-based anonymous message authentication protocol in VANETs, Comput. Commun. 112 (2017) 154–164.
[23] Q.E. Ali, N. Ahmad, A.H. Malik, W.U. Rehman, A.U. Din, G. Ali, ASPA: Advanced strong pseudonym based authentication in intelligent transport system, PLoS One 14 (8) (2019) e0221213.
[24] M. Bayat, M. Pournaghi, M. Rahimi, M. Barmshoory, NERA: A new and efficient RSU based authentication scheme for VANETs, Wirel. Netw. 26 (5) (2020) 3083–3098.

[25] Y. Liu, W. Guo, Q. Zhong, G. Yao, LVAP: Lightweight V2I authentication protocol using group communication in VANETs, Int. J. Commun. Syst. 30 (16) (2017) e3317.
[26] K. Chain, K.-H. Chang, W.-C. Kuo, J.-F. Yang, Enhancement authentication protocol using zero-knowledge proofs and chaotic maps, Int. J. Commun. Syst. 30 (1) (2017) e2945.
[27] S. Dolev, Ł. Krzywiecki, N. Panwar, M. Segal, Dynamic attribute based vehicle authentication, Wirel. Netw. 23 (4) (2017) 1045–1062.
[28] X. Li, Y. Liu, X. Yin, An anonymous conditional privacy-preserving authentication scheme for VANETs, in: Proceedings of 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems, HPCC/SmartCity/DSS, IEEE, 2019, pp. 1763–1770.
[29] X. Li, T. Liu, M.S. Obaidat, F. Wu, P. Vijayakumar, N. Kumar, A lightweight privacy-preserving authentication protocol for VANETs, IEEE Syst. J. (2020).
[30] Sufatrio, R.H.C. Yap, Extending BAN logic for reasoning with modern PKI-based protocols, in: Proceedings of 2008 IFIP International Conference on Network and Parallel Computing, IEEE, 2008, pp. 190–197.
[31] S. Adhikari, S. Ray, G.P. Biswas, M.S. Obaidat, Efficient and secure business model for content centric network using elliptic curve cryptography, Int. J. Commun. Syst. 32 (1) (2019) e3839.
[32] R. Amin, S.H. Islam, M.S. Obaidat, G. Biswas, K.-F. Hsiao, An anonymous and robust multi-server authentication protocol using multiple registration servers, Int. J. Commun. Syst. 30 (18) (2017) e3457.
[33] J. Moon, Y. Lee, J. Kim, D. Won, Improving an anonymous and provably secure authentication protocol for a mobile user, Secur. Commun. Netw. (2017).
[34] L. Benarous, B. Kadri, S. Bitam, A. Mellouk, Privacy-preserving authentication scheme for on-road on-demand refilling of pseudonym in VANET, Int. J. Commun. Syst. (2019) e4087.
[35] W. Dai, Crypto++ library 5.6.0, 2009, http://www.cryptopp.com/benchmarks.html.
[36] S. Micali, Certificate revocation system, Google Patents, US Patent 5,666,416, 1997.
[37] M. Raya, J.P. Hubaux, Securing vehicular ad hoc networks, J. Comput. Secur. 15 (1) (2007) 39–68.
[38] S. Jiang, X. Zhu, L. Wang, An efficient anonymous batch authentication scheme based on HMAC for VANETs, IEEE Trans. Intell. Transp. Syst. 17 (8) (2016) 2193–2204.
[39] V. Paruchuri, A. Durresi, PAAVE: Protocol for anonymous authentication in vehicular networks using smart cards, in: Proceedings of GLOBECOM, IEEE, 2010.
[40] J. Cui, L. Wei, J. Zhang, Y. Xu, H. Zhong, An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks, IEEE Trans. Intell. Transp. Syst. 20 (5) (2019) 1621–1632.
[41] B. Palanisamy, L. Liu, MobiMix: Protecting location privacy with mix-zones over road networks, in: Proceedings of IEEE 27th International Conference on Data Engineering, April 2011 Hanover, Germany, IEEE, 2011, pp. 494–505.
[42] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, Y. Fang, Traffic-aware multiple mix zone placement for protecting location privacy, in: Proceedings of IEEE INFOCOM, Orlando, FL, USA, 2012, IEEE, 2012, pp. 972–980.
[43] R. Yu, J. Kang, X. Huang, S. Xie, Y. Zhang, S. Gjessing, Mix-Group: Accumulative pseudonym exchanging for location privacy enhancement in vehicular social networks, IEEE Trans. Dependable Secure Comput. 13 (1) (2016) 93–105.
[44] B. Ying, D. Makrakis, Pseudonym changes scheme based on candidate-location-list in vehicular networks, in: 2015 IEEE International Conference on Communications, ICC, IEEE, 2015, pp. 7292–7297.