Using I2P Encrypted Virtual Tunnels for a Secure and Anonymous Communication in VANets: I2P Vehicular Protocol (IVP)

Tayeb Diab
University of Haute Alsace
Marc Gilg
University of Haute Alsace
Frederic Drouhin
University of Haute Alsace
Pascal Lorenz (pascalorenz@gmail.com)
Université de Haute-Alsace https://orcid.org/0000-0003-3346-7216

Research Article

Keywords: VANets, Invisible Internet Project, Anonymity, encryption, tunnel maintenance, THELLO.

Posted Date: December 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-798089/v1

Using I2P (Invisible Internet Protocol) encrypted virtual tunnels for a secure and anonymous communication in VANets: I2P Vehicular Protocol (IVP)

DIAB Tayeb
Institute of Research in Computer Science, Mathematics, Automation and Signal (IRIMAS)
University of Haute Alsace
68000 Colmar, France
tayeb.diab@uha.fr

GILG Marc
Institute of Research in Computer Science, Mathematics, Automation and Signal (IRIMAS)
University of Haute Alsace
68000 Colmar, France
marc.gilg@uha.fr

DROUHIN Frederic
Institute of Research in Computer Science, Mathematics, Automation and Signal (IRIMAS)
University of Haute Alsace
68000 Colmar, France
Frederic.Drouhin@uha.fr

LORENZ Pascal
Institute of Research in Computer Science, Mathematics, Automation and Signal (IRIMAS)
University of Haute Alsace
68000 Colmar, France
lorenz@ieee.org

Abstract—***Providing security and anonymity within VANet requires application of robust and secure models that meet several characteristics of VANet. I2P as a secure protocol designed to anonymize the communication on the internet, can be used as a reference model to develop new mechanisms of security and anonymity in VANet.***

I2P uses robust mechanisms and strong algorithms to reinforce the security and the anonymity of the communication. However, the difference between internet and VANet in terms of mobility and connectivity of nodes presents a big issue that needs to be treated when using I2P mechanisms in VANet.

In the previous work [1], we propose a protocol based on tunnels and encryption algorithms that use digital signatures and authentication mechanisms. Tunnels are created in static scenarios and without maintaining their existence. In this paper, we complete the last version of the proposed protocol (I2P Vehicular Protocol) by integrating a tunnel maintenance algorithm for maintaining the existence of the created tunnels during the communication. This algorithm allows the implementation of the protocol in mobile scenarios of VANet.

The effectiveness and security of IVP protocol are proved by analyzing the added part related to the tunnel maintenance process and showing performance results (end-to-end delay, PDR and overhead). Simulation scenarios were executed using NS3 simulator.

Keywords—VANets, Invisible Internet Project, Anonymity, encryption, tunnel maintenance, THELLO.

I. INTRODUCTION

I2P as a secure protocol used in internet uses several algorithms and mechanisms ensuring security and anonymity of communication in internet. I2P uses symmetric and asymmetric encryption algorithms, signatures and certificates according to four levels: I2CP (I2P Client Protocol), garlic, tunnel and transport encryption.

VANet as a specific kind of wireless network, it has different characteristics such as high velocity and rapid topology change. These characteristics need to be treated carefully when applying algorithms to provide security on VANet.

Using I2P mechanisms in VANets can provide security and anonymity of messages during the communication. However, it requires to be adapted to respond to several characteristics of VANet. The objective is to propose a model of security to ensure anonymity of communication in VANet. This model is based on creating tunnels and maintaining their existence. Thus, we implement the transport encryption layer in our first work [2], then the tunnel and garlic encryption layers in the last contribution [1].

In the previous work, the first version of the protocol has been proposed by adapting some of the I2P mechanisms. This version is based on creating tunnels between nodes (vehicles and RSUs) statically and without maintenance of these tunnels. In addition to implement the tunnel and garlic encryption layers during the communication.

This paper is an extension of our previous contribution [1] presented in IEEE Global Communication Conference (GLOBCOM 2019). In this work, a second version of the protocol (IVP) is proposed to be implemented in real scenarios of VANet and deal with the mobility of vehicles. This version develops a tunnel maintenance algorithm to maintain the existence of tunnels during the communication in real VANet.

The proposed protocol is supposed to be implemented in urban environments, where speeds of vehicles are less than 50 km/h and large number of RSUs and vehicles can exist, which help to facilitate the creation and maintenance of the encrypted tunnels.

The rest of the paper is organized as follow; several related works are discussed in section 2. Section 3 presents the second version of the proposed protocol (IVP) and explains the operation and several steps of the maintenance algorithm. Section 4 proves the security and efficiency of the second version of the protocol. Finally, we end the paper by a conclusion and some perspectives.

II. Related work

The last contribution in [1] is about anonymizing communication in VANets by Applying I2P Mechanisms. The paper [1] describes the proposed protocol according to two phases:

1) Tunnel creation process which is responsible for getting the tunnel node identities then requesting the creation of tunnels.

2) Communication where nodes use their inbound and outbound tunnels to communicate between each other.

The solution proposed in this work [1] suffers from the problem of maintenance. Tunnels are not maintained during the communication, which interrupt sending and reception of messages.

In this paper, we resolve this problem by integrating an algorithm responsible for the tunnel maintenance.

In [3], authors propose a dynamic pseudonym changes system for privacy in VANet (RIN). This solution is based on three entities: vehicles, RSUs and the trust authority (TA). Vehicles communicate with each other using pseudonyms. TA generates for each vehicle a real pseudonym to use during the communication and sends a copy to RSUs. The vehicle changes its pseudonym each time t that's calculated before using the exponential distribution function. The vehicle identifies and searches for trusted neighbors to determine their directions. Based on that, the vehicle calculates the time for the next change of a new pseudonym. This change is performed locally and it's done this way till the detection of a new RSU then starts with the generation of the initial pseudonym by RSU.

Similarly in [4], authors present an anonymous and lightweight authentication for Secure Vehicular Networks which is based on three types of nodes; vehicles, RSUs and TA. In this approach, the TA assigns each user in the network with anonymous certificate and session key. Each vehicle is supposed to have a smart card which is assigned by calculated parameters through the TA. To secure the communication, the TA authenticates each vehicle in the network and attribute it with a session key. Data messages are validated by using two hash chains to reduce computation complexity. The data message is sent with the key message including dynamic login identity of the vehicle, the message authentication code, etc. At reception, the vehicle verifies the validity of the received values to authenticate the data message.

Authors in [5], propose a VANet privacy protection scheme based on fair blind signature and secret sharing algorithm. In this work, three types of entities exist: vehicles, RSUs and a trusted center that contains Authentication Centre (AC), Pseudo Authentication Centre (PAC) and Tracking Centre (TC). Each entity generates the corresponding public and private keys. In order to get the pseudonym certificate, the vehicle needs to be registered firstly at AC by sending a message including its signature to be verified by AC. The vehicle signs a message and sends it to PAC to check the signature and issues its pseudonym certificate.

Vehicles use pseudonyms during the anonymous communications. When receiving a message, the nearby vehicles verify the validity of signature of this message using the public key

information. TC uses the secret sharing algorithm to allocate its private key to a tracking group that can recover the private key information when it's needed. In this case, TC can convene the tracking group members to restore the true identity of the illegal node and punish him.

In [6], a local identity-based anonymous message authentication protocol (LIAP) is proposed to VANet. The infrastructure in this protocol is based on four types of nodes classified into two-layer network model: vehicles and RSUs constitute the application layer, they connect with each other using the DSRC technology; the second layer consists of the management layer in which Certificate Authority (CA) and Traffic Management Center (TMC) communicate with RSUs via the wired channel.

The CA delivers a long-term certificate and generates public and private keys and signatures to the different nodes. RSUs and vehicles check each other if they are not revoked and have valid certificates and signatures. After that, vehicles can ask RSUs for the local master keys. The RSU uses a secure channel to send its local master keys to the vehicle that can generate its anonymous identities and the corresponding private keys to sign the safety-related message. This message includes the public key of the sender RSU, which allows to vehicles to know which RSU sends this message, then check its validity. If any verified information is invalid the message will be dropped.

In [7] authors propose an anonymous authentication scheme based on group signature in VANets (AAAS) that is based on group signature mechanism. AAAS adds region trust authority (RTA) as a group manager to provide anonymous authentication and alleviate computation and communication pressure of Trust Authority (TA) and RSU with low computation and storage capacity. This scheme is proposed to establish the trust relationship between vehicles and RSUs anonymously, where TA is trusted by all entities. RTA is trusted by all RSUs in the assigned areas, but there is no trust relationship between RTAs. RSU trusts TA and RTA in its area but not vehicles and other RSUs. Vehicles do not trust other vehicles and RSUs.

The paper [8] presents an efficient anonymous authentication scheme using registration list in VANets. This work is based on ASC scheme (Authentication based on Smartcard) that supports V2V and V2R communications provides necessary security requirements such as anonymity, traceability and unlinkability in the VANETs. ASC scheme consists of five phases: the user registration phase where users register with the *TA*; the user login phase during which the vehicle owner logs in to the vehicle with a valid smart card; the user authentication phase where vehicle efforts to establish a secure connection with the TA; the data authentication phase where authenticated datagrams are sent; the password change phase where the vehicular owner changes his/her passwords

Authors in [9] proposes a fog computing-based anonymous-authentication scheme for VANets; the scheme reduces the communication burden of the TA by enabling self-authentication between vehicles and RSUs. The proposed model consists of three major layers: Cloud layer which mainly includes the TA, computing and storage resources. TA is responsible for registering and managing the local authorities and vehicles, as well as allocating certification and system parameters to them simultaneously. Secondly, fog layer that has dedicated local fog servers to provide a wireless access to internet and to computing and storage resources. Fog servers use the network-function virtualization technology in order to virtualize the physical resources by building virtual machines and realize flexible resource allocation. The third layer: vehicles which shares some value information (traffic safety warnings), possesses sensitive information (public, private keys) and has the GPS (Global Positioning system).

III. The I2P Vehicular Protocol (IVP)

A. Model overview

In the previous work [1], a secure model is developed to ensure the anonymity of the communication in vehicular ad hoc networks. This model is inspired by the protocol I2P, which is mainly based on using tunnels and implementing encryption and signature mechanisms.

The proposed protocol uses the same principle in VANet by creating tunnels and secure communication using encryption and signature algorithms. This protocol is developed according to two versions; the first version is represented in the previous contribution [1], in which the tunnels are created and used to exchange data but not

maintained. Once a tunnel is created, it is supposed to exist for a long time. This version is implemented using static scenarios where no movement in the network and each vehicle has the same position during the time of the simulation.

In this contribution, we propose the second version of the protocol. We continue the previous work by developing an algorithm of maintenance that keeps the existence of tunnels and provides the tunnel nodes with the current information about the tunnel during the communication. This algorithm is based on exchanging messages between the tunnel nodes to provide them information about the status of the tunnel. At any time, if a breakage occurred or the tunnel becomes long, tunnel nodes launch the maintenance process to repair the tunnel, in other cases when the tunnel cannot be repaired, a tunnel creation process is launched to create a new tunnel.

In this work, the proposed protocol is improved by developing a tunnel maintenance algorithm, which allows using this protocol in real mobile scenarios of VANet. The proposed tunnel maintenance process is detailed as follows.

B. Tunnel maintenance process

Vehicular networks have special characteristics comparing to other kinds of networks. High vehicle velocities and fast change of the topology make maintenance of tunnels and even its creation very difficult in some situations.

For that, the developed mechanisms used for the creation and maintenance of tunnels should consider these characteristics and face these limitations. After the tunnel creation process, tunnel nodes have to maintain consistently their existence in the network. In case of breaking of such links between tunnel nodes, these nodes can detect this breakage and know which process of reparation can be launched. In this contribution, a tunnel is a set of tunnel nodes (3 nodes max) related to each other by normal nodes (2 intermediate nodes max) as it is supposed in the previous work.

a. THELLO exchange

After creating tunnels, tunnel nodes exchange beacons between each other to keep information about the state of the tunnel, which is based on the state of the link between tunnel nodes. This approach considers that the state of the link depends on the number of hops between the two tunnel nodes. Each two successive tunnel nodes exchange periodically encrypted Tunnel HELLO (THELLO) messages between each other through intermediate nodes; TN1 to TN2, TN2 to TN1, TN2 to TN3 and TN3 to TN2 as represents figure 1 and detailed in algorithm 1.

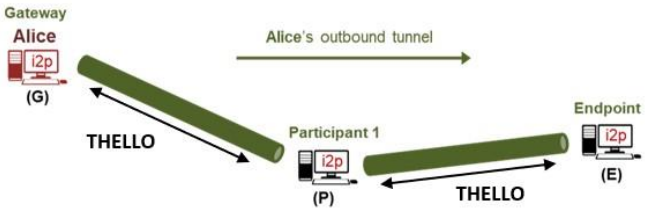


Fig. 1. THELLO exchange in the tunnel

Alg. 1. Algorithm of the THELLO exchange

For each time interval:

$TN1 \rightarrow TN2\ \{e\_Pk_{(TN2)}\ (THELLO\ [TID, HP, Flag])\};$

$TN2 \rightarrow TN1\ \{e\_Pk_{(TN1)}\ (THELLO\ [TID, HP, Flag])\};$

$TN2 \rightarrow TN3\ \{e\_Pk_{(TN3)}\ (THELLO\ [TID, HP, Flag])\};$

$TN3 \rightarrow TN2\ \{e\_Pk_{(TN2)}\ (THELLO\ [TID, HP, Flag])\};$

THELLO message is encrypted using an asymmetric encryption algorithm. The sender encrypts it using the public key of the receiver which uses its private key for the decryption. THELLO includes the tunnel ID, the hop count between tunnel nodes and other fields shown in figure 2.

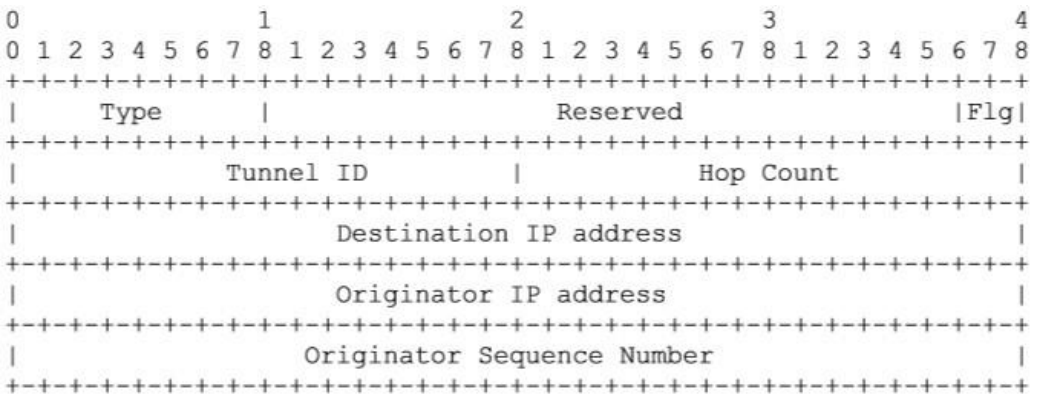
After receiving TACK from TN3 during the tunnel creation, TN2 start exchanging encrypted THELLO messages each interval of time T with TN3. For each sent message, the real number t (time interval) should be defined as follow:

$$0 < t \leq T \quad \ldots(1)$$

The same process is launched between TN1 and TN2 after receiving TACK from TN2.

THELLO message indicates the distance between tunnel nodes. It is used to inform tunnel nodes about the distance between each other. When THELLO is sent to the next tunnel node, each intermediate node increments the number of hops (*Hp*) and then forwards the message to the next node until being received by the next tunnel node, which notifies the tunnel nodes when the number of intermediate nodes is increased.

THELLO message includes Flag field that describes how long is the tunnel. Each THELLO can have one of the group elements in formula (2):



Type: Type of the AODV message
Reserved: Not used
Flg: Flags about distance (NORMAL, LONG, CONFIRMED_LONG)
Tunnel ID: Identifier of the tunnel in the network
Hop Count: Hop count between tunnel nodes
Destination IP Address: IP address of the destination of THELLO
Originator IP Address: IP address of the sender of THELLO
Originator Sequence Number: Sequence Number of the sender of THELLO

Fig. 2. THELLO message format

$$\text{Flag} \in \{\text{NORMAL, LONG, CONFIRMED\_LONG}\} \quad \ldots(2)$$

When receiving a THELLO message, the tunnel node checks the number of hops to decide if the route is long or not. If *Hp* is under than a predefined threshold (L_{max}) according to relation (3), the link is considered as normal (using the NORMAL flag in THELLO message) even if it was declared as long (LONG flag) by the sender tunnel node.

For each tunnel segment, the natural number l (segment length) must be defined as follow:

$$1 \leq l \leq L_{max} \quad \ldots(3)$$

If *Hp* is more than the threshold, the tunnel node checks the state of the link indicated by the sender:

- If it is declared as LONG, the node confirms this declaration and marks the state flag as CONFIRMED_LONG.
- If it is declared as NORMAL, it marks the state as LONG for the first time.

THELLO messages must be exchanged within T_{max} as a maximum interval of time. For each delayed message when $t > T$, the real number t (time interval) must be defined as:

$$T < t \leq T_{max} \quad \ldots(4)$$

From relations (1), (2), (3) and (4), a tunnel can be defined as active when:

$$t \in]0, T_{max}] \ \textit{and} \ l \in [1, L_{max}] \ \textit{and} \ \text{Flag} \in \{\text{NORMAL, LONG, CONFIRMED\_LONG}\} \quad \ldots(5)$$

During the communication, tunnels can be broken in many cases. In this part of the contribution, a maintenance algorithm is presented to maintain the existence of tunnels. This work is presented in the following section.

b. The tunnel breakage and maintenance cases

The tunnel maintenance process is launched differently from a tunnel node to another, in which each one has a special role in this process. According to the link that is broken, the maintenance process can be launched in such a manner. In the proposed algorithm, two tunnel nodes can detect the breakage between each other; TN1 and TN2. A breakage means the long-distance or the link break between tunnel nodes.

Each of the two tunnel nodes acts in the maintenance process according to its position in the tunnel. The creator maintains the existence of TN2 that maintains the existence of TN3. Two cases of breakage can occur during the communication: a breakage between TN1 and TN2, or/and breakage between TN2 and TN3. For that, two algorithms are developed to maintain the existence of the tunnel; one algorithm launched by TN2 and the second launched by TN1.

1. Maintenance of TN2

After sending TACK to TN2, TN3 waits for the THELLO message from TN2. If no THELLO is received after a long interval of time (T_{max}) according to relation (4), TN3 considers that the link TN2-TN3 is broken and it deletes the entry of the tunnel from its TunnelT table as represented below in figure 3.

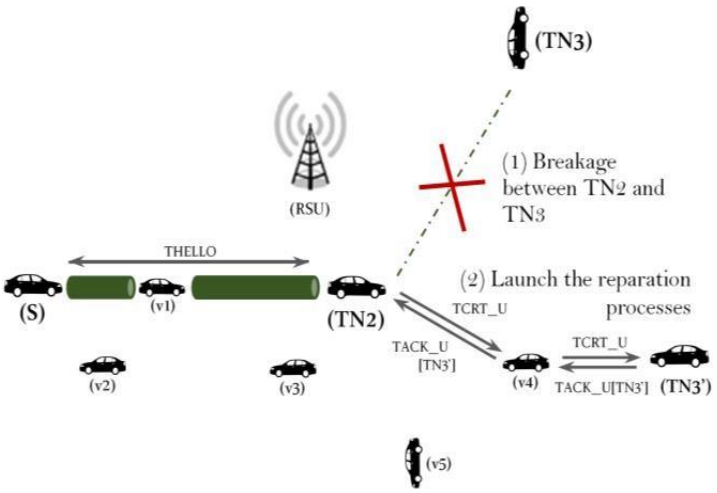


Fig. 3. Tunnel maintenance of TN2

At the reception of THELLO, TN3 checks the state of the route declared by TN2 and the number of hops. According to these two parameters, TN3 decides as detailed in the following algorithm (Algorithm 2).

- If the distance is accepted, TN3 resends an encrypted THELLO message to TN2 indicating the state of the route as NORMAL in the THELLO message;

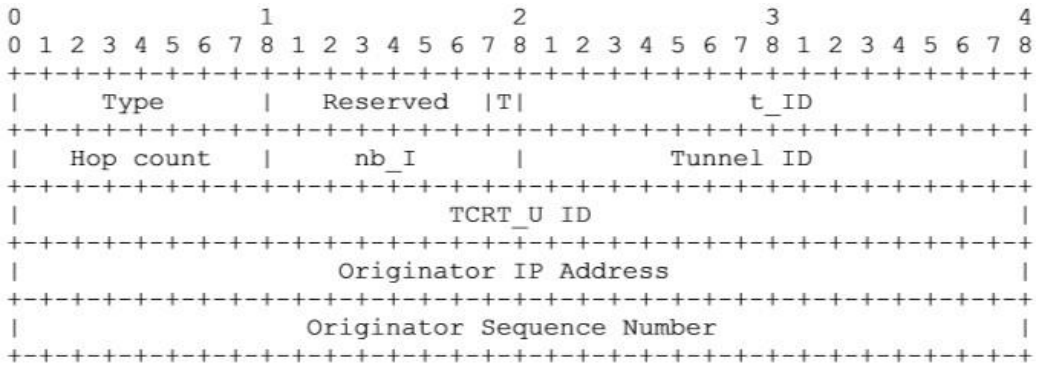
- Otherwise, it checks the state declared by the sender: If it is declared as NORMAL, it indicates that the route is long (LONG flag) for the first time. If it is declared as LONG, the node confirms this state by CONFIRMED_LONG flag, then resends the encrypted THELLO message to TN2. If the sender has already confirmed that the distance is long (by CONFIRMED_LONG), TN3 stops sending encrypted THELLO to TN2, then deletes the tunnel entry from its TunnelT table.

When TN2 receives THELLO from TN3, the same process is repeated, except when the distance is long and THELLO sent with the flag CONFIRMED_LONG (or when no THELLO received from TN3 after an interval of time), TN2 checks in its TunnelT table if it has already the entry about the tunnel; If so, it starts a repair process (figure 3), otherwise, it ignores the received THELLO.

Alg. 2. The maintenance algorithm of TN2

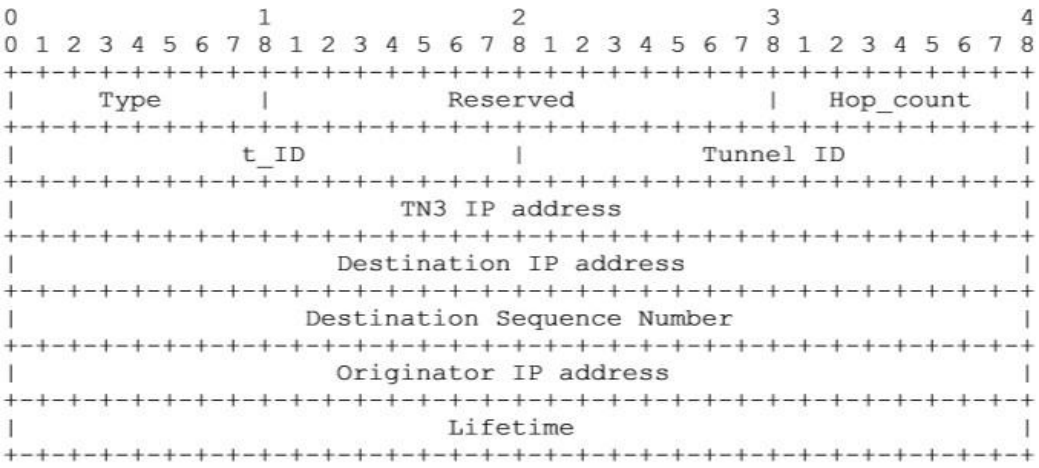
```
if the sender is TN3 then
    if FLAG = C_LONG then
      tunnelTable.FLAG ← IN_REPATION;
      TN2: delete TN3;
      TN2 => TN'3 {TCRT_U [TID', T_type, i]};
      TN'3 => TN2 {e_Pk(TN2) (TACK_U [TID", @TN'3])};
      TN2 => TN1 { e_Pk(TN1) (TACK_U [TID", @TN'3])};
    else if nb_hop <= Top then
          FLAG ← NORMAL;
          Send e_Pk(TN3) (THELLO) to TN3;
    else if FLAG = NORMAL then
          FLAG ← LONG;
          Send e_Pk(TN3) (THELLO) to TN3;
     else
          FLAG ← C_LONG;
          Send e_Pk(TN3) (THELLO) to TN3;     Delete the entry;
     end if
end if
if the sender is S then
    if FLAG = C_LONG then
          Delete the entry;
          Send e_Pk(TN3) (THELLO) with FLAG = C_LONG to TN3;
    else if nb_hop <= Top then
          FLAG ← NORMAL;
          Send e_Pk(S) (THELLO) to S;
    else if FLAG = NORMAL then
          FLAG ← LONG;
          Send e_Pk(S) (THELLO) to S;
    else
          FLAG ← C_LONG;
          Send e_Pk(S) (THELLO) to S;    Delete the entry;
    end if
end if
```

The repair process is based on two messages: Tunnel Creation Update (TCRT_U) and Tunnel Acknowledgement (TACK_U) messages. After receiving THELLO with CONFIRMED_LONG flag and long distance from TN3, TN2 makes the entry flag of the tunnel as IN_REPARATION then broadcasts TCRT_U message in the network to select another tunnel node TN3' nearby. TCRT_U is treated similarly as the TREQ message and it contains almost the same fields as shown in the figure 4.



Type: Type of the AODV message
Reserved: Not used
T: Tunnel type (outbound or inbound)
t_ID: Temporary identifier generated by TN2 to identify its address
Hop Count: Hop count between tunnel nodes
nb_I: Number of intermediate nodes between TN2 and the new TN3
Tunnel ID: Identifier of the tunnel in the network
TCRT_U ID: Identifier of the TCRT_U message
Originator IP Address: IP address of TN2
Originator Sequence Number: Sequence Number of TN2

Fig. 4. Tunnel Creation Update (TCRT_U) message format



Type: Type of the AODV message
Reserved: Not used
Hop Count: Hop count between tunnel nodes
t_ID: The temporary identifier sent by TN2
Tunnel ID: Identifier of the tunnel in the network
TN3 IP address: IP address of TN3
Destination IP Address: IP address of the sender of TACK_U
Destination Sequence Number: Sequence Number of the sender of TACK_U
Originator IP Address: IP address of the receiver tunnel node of TACK_U
Lifetime: Lifetime of TACK_U (in milliseconds)

Fig. 5. Tunnel Acknowledgement Update (TACK_U) format

The selection of TN3' is based on a random number R of hops chosen by TN2. At each reception, the node decreases the number of hops until reaching the distance required between TN2 and the new third tunnel node (TN3'). In this case, for each node ($N_{n\ (dist=R)}$) has the distance R to TN2, it has the probability P to be TN3' according to the relation as follow:

$$P(N_{n(dist=R)}) = 1 / \sum (N_{i(dist=R)}) \quad \ldots(6)$$

TN2 indicates in TCRT_U the type of the created tunnel already existed, in addition to a new TID different from the real TID of the tunnel. After reaching the number of hops required, the concerned node (TN3') creates an entry in its TunnelT table about the existed tunnel with a new TID generated by TN3'. Then, it replies with an encrypted TACK_U to TN2 including its address and the new TID (figure 5).

When TN2 receives TACK_U, it places the address of TN3' and the new TID in the entry of

the tunnel in its tunnel table and makes the entry as CREATED. Then, TN2 forwards TACK_U to TN1, which updates the entry about the tunnel in its TunnelT table. If this tunnel is inbound, TN1 sends the new TN3's identity and the new TID to the NetDB database to update the previous TN3 identity and TID.

2. Maintenance of TN1

After sending a TACK message to TN1 (during the tunnel creation process), TN2 waits for a period of time (T_{max}), if no THELLO had been received during this interval (relation (4)), it considers that the route to TN1 is ruptured (figure 6). In this case, TN2 deletes the entry of the tunnel from its TunnelT table, then sends an encrypted THELLO message with CONFIRMED_LONG flag and with a high number of hops to TN3. When TN3 receives the THELLO message, it deletes the entry about the tunnel from its TunnelT table.

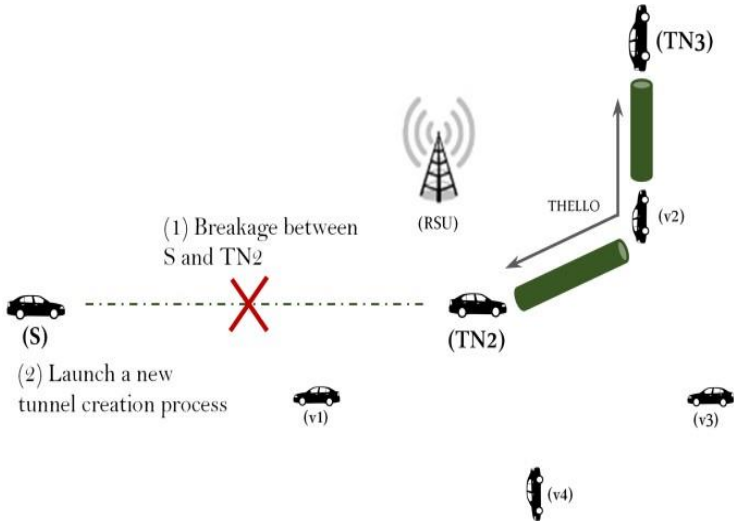


Fig. 6. Tunnel maintenance of TN1

When TN1 receives TACK from TN2, it starts sending THELLO messages (encrypted) to TN2 for each interval of time (T3). Similarly to the previous algorithm of the tunnel maintenance, when TN2 receives THELLO, it checks the state declared by TN1 and the number of hops.

- If the distance is acceptable, TN2 resends an encrypted THELLO message to TN1 indicating that the state of the route is NORMAL.

- If the distance to TN1 is long (more than the threshold) and the state of the route was indicated as NORMAL, TN2 replies TN1 by an encrypted THELLO message with LONG flag, if TN1 has declared the state as CONFIRMED_LONG, TN2 deletes the entry of the tunnel from its TunnelT table, then sends an encrypted THELLO to TN3 with CONFIRMED_LONG flag and a high number of hops as described in the previous algorithm ("Maintenance of TN2").

Similarly, at reception of THELLO from TN2, TN1 does the same verification of the two parameters; the state and the number of hops. It processes like TN2 except when the state was declared by TN2 as CONFIRMED_LONG and the distance is long. In this case or when no THELLO received after an interval of time, TN1 considers that the tunnel is broken. Thus, it deletes the tunnel from its TunnelT table and sends an encrypted THELLO message to TN2 with CONFIRMED_LONG flag and a high number of hops. TN2 deletes the tunnel from its TunnelT table and forwards the message to TN3. Similarly, TN3 deletes the tunnel from its TunnelT table. At the same time, TN1 launches the tunnel creation process to create a new tunnel. All this process is detailed in the algorithm 3.

Alg. 3. The maintenance algorithm of TN1

```
if FLAG = C_LONG then
    Delete the entry;
    // Launch the tunnel creation process;
else if nb_hop <= Top then
    FLAG ← NORMAL;
    Send e_Pk(TN2) (THELLO) to TN2;
else if FLAG = NORMAL then
    FLAG ← LONG;
    Send e_Pk(TN2) (THELLO) to TN2;
else
    FLAG ← C_LONG;
    Send e_Pk(TN2) (THELLO) to TN2;
    // Launch the tunnel creation process
end if
```

The followed section analyzes the security and performance of the second version of the proposed protocol during the communication.

IV. PERFORMANCE AND SECURITY ANALYSIS

In this section, we show the anonymity and security of the enhanced version of the proposed protocol compared to the previous version and

AODV protocol. Similarly to the last contribution, the comparison is based on three QoS parameters: packet delivery ratio, end-to-end delay and overhead.

A. Security analysis

The improved version of the proposed protocol (IVP) uses the same mechanisms and algorithms to provide anonymity of communication and secure messages. It is based on using tunnels and encryption algorithms inspired by the I2P protocol. Besides, it uses a tunnel maintenance algorithm to maintain the existence of tunnels in the network. The tunnel creation part is discussed in the previous contribution [1].

In this section, we analyze the added part related to the tunnel maintenance process. This process is detailed into three phases: the THELLO exchange, maintenance of TN2 and maintenance of TN1 (which represents the tunnel creation process developed in the previous work).

The THELLO exchange is based on using THELLO messages. These messages include the tunnel ID, which must be secret between the tunnel nodes during the communication. For that, THELLO is encrypted between successive nodes in the tunnel (using an asymmetric encryption algorithm).

Thus, only the communicating tunnel nodes can know the content of THELLO messages. During the maintenance process of TN2, two messages are used: Tunnel Creation Update (TCRT_U) and Tunnel acknowledgement update (TACK_U) messages (in addition to the encrypted THELLO message).

TCRT_U is not encrypted; however, its content is not critical to the point where it threatens the anonymity of the communication when an intermediate node read this content. TCRT_U includes the TID used before the breakage. When the new TN3 is selected, it ignores the TID sent in the TCRT_U message and generates a new TID for the tunnel. After that, it sends it in the TACK_U message to TN2. TACK_U is encrypted using an asymmetric algorithm, which prevents intermediate nodes to know this TID sent in the TACK_U message.

For the tunnel node TN1, the maintenance of the tunnel consists on sending an encrypted THELLO message to TN2 to delete the tunnel, then launching the tunnel creation process from the beginning as detailed in the previous work [1].

B. Performance analysis

In this part of the contribution, we present a simulation of three protocols: AODV, the first version [1] and the new version IVP. We launch the simulation in the NS3 platform and we show the results using box plots. Similarly to the previous work, the comparison between these protocols is based on three QoS parameters: the packet delivery ratio (PDR) [10], overhead [11] and end-to-end delay ($D_{e\text{-}e}$) [12] as calculated below:

$$PDR = 100 * \frac{\sum Data\ packets\ received}{\sum Data\ packets\ sent}\ (in\ \%)$$

$$Overhead = \frac{\sum Control\ packets\ sent}{\sum Data\ packets\ sent}$$

$$D_{e\text{-}e} = N * (D_{tran} + D_{prop} + D_{proc} + D_{que})$$

// *N:* number of links
// D_{tran}, D_{prop}, D_{proc}, D_{que} : transmission, propagation, processing, Queuing delays.

The simulation is launched for six scenarios with a different number of nodes and for a duration of 150s. We use scenarios with different number of nodes to see the impact of this number on the IVP protocol. The speeds of vehicles are chosen between 20 and 35 km/h as average speeds in urban environments where the limited speed is up to 50km/h. Table 1 presents the settings used in the simulation.

TABLE I. SIMULATION SETTINGS

Area size	3320*2035m
Number of nodes	10 to 20
Speed of nodes	7m/s, 9m/s
Number of communications	3
Number of messages per 2 seconds	1
Simulation time	150s
Packet size	256
Routing protocol	AODV
Propagation model	Friis
Transmission power	0dbm
Frequency band	5GHz
Transmission bandwidth	5MHz

The use of box plots to show the simulation results refers to the way of generation of random values in NS3. These values are based on the "seed" value, which causes differences in the result values even for the same scenarios. For that, we launch the simulation of each scenario many times to have a

set with different possible values during the simulation. Then, we use box plots for each set (each scenario) to show the distribution and concentration of values.

As represents figure 7, the packet delivery ratio in the previous version of the proposed protocol is low, in which, the values are mainly between 0% and 5%. That makes the first version inappropriate to VANet due to the mobility of vehicles.

In version 1 of the proposed protocol, the mobility of vehicles is not taken into account by that version. Besides, no maintenance process is integrated to maintain the existence of tunnels. This version is dedicated for being implemented statically where no mobility, thus, tunnels can be established for a long time.

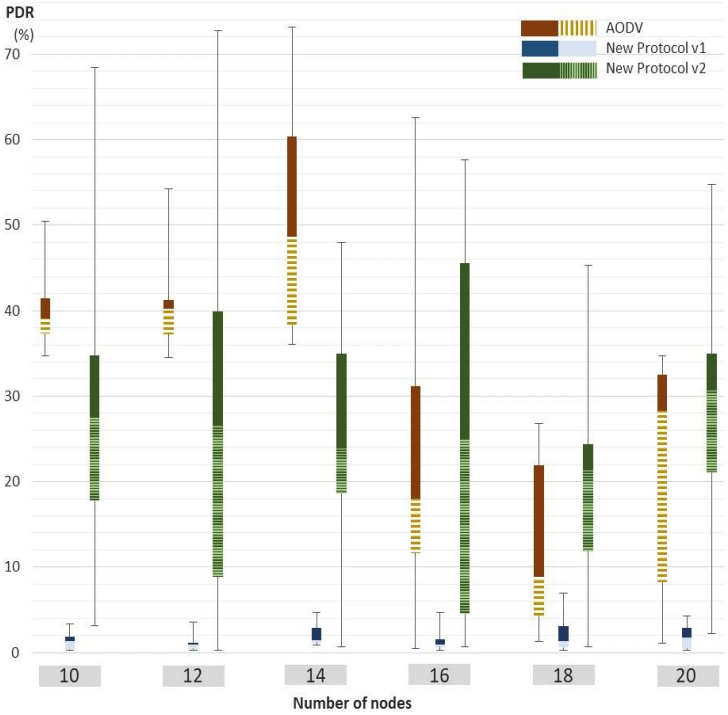


Fig. 7: Packet Delivery Ratio of the proposed protocol (version 1 and IVP) and AODV

In real VANet, tunnels can be broken at any time, which can cut the communication and requires to launch the tunnel creation process many times. In the new version (IVP), the PDR is high compared to the previous version of the protocol thanks to the proposed algorithm of maintenance of tunnels, in which, the majority of PDR values are generally between 10% and 40%, which are significant compared to the previous version (between 0% and 5%).

Compared to AODV, the new version of the protocol has a PDR with low values in small scenarios (10, 12 and 14 nodes), and with relatively high values in other scenarios (with high number of nodes), which shows the ability of the proposed protocol to be implemented in real VANet with high number of vehicles.

Concerning the end-to-end delay, it is noticed in figure 8 that the proposed protocol (version 1 and IVP) is better than the AODV protocol from the routing point of view.

In AODV, the end-to-end delay values are mostly between 0.1s and 0.6s, which make the difference with the proposed protocol that has an end-to-end delay with values between 0.004s and 0.02s.

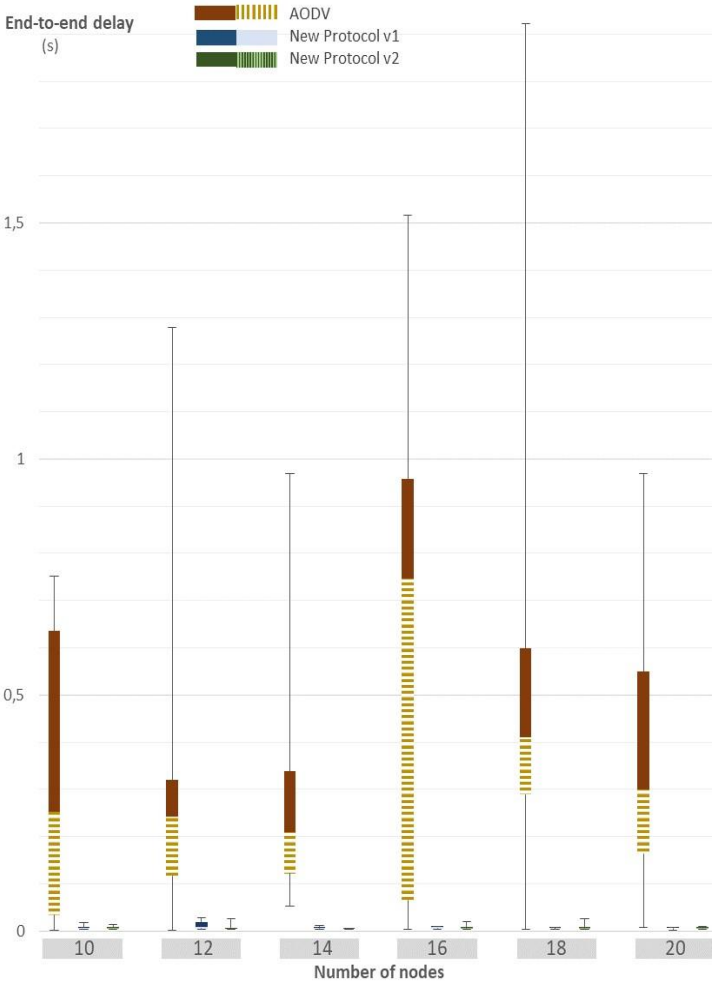


Fig. 8. End-to-end delay of the proposed protocol (version 1 and IVP) and AODV

The AODV protocol has an end-to-end delay with high values compared to the proposed protocol because of the on-demand process to discover routes between the communicant nodes. This process can be launched frequently because of the mobility of vehicles. However, in the proposed protocol, the on-demand process is launched during the tunnel creation and before the

communication, which helps to discover several routes between nodes in the network. The discovered routes can exist thanks to the periodic exchanged messages used in the creation and maintenance of tunnels (IVP). As a result, the communicant nodes can use these routes without launching the on-demand process (the problem with AODV).

Adding security algorithms in VANet can cause a degradation in some QoS parameters. In the proposed protocol, we integrate algorithms for creating and maintaining tunnels and encrypting data, which use messages more than AODV. The first version of the protocol uses TREQ, TREP, TCRT and TACK, and the second version uses in addition three messages: THELLO, TCRT_U and TACK_U. That is why our protocol has high values for the overhead compared to the AODV protocol as represents figure 9.

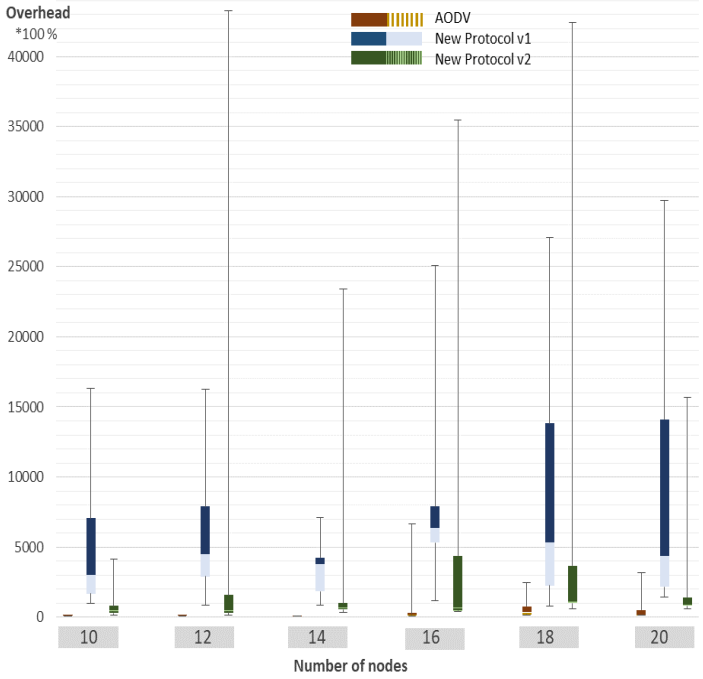


Fig. 9. Overhead of the proposed protocol (version 1 and 2) and AODV

The IVP protocol is better than the first one in term of overhead. This improvement refers to the integrated tunnel maintenance process. When a tunnel is broken (very frequently), the protocol (version 1) requires to relaunch the tunnel creation process from the beginning, which generates a high number of messages. However, in the second version, the tunnels are maintained most of the time using THELLO messages and (TCRT_U and TACK_U messages when breakage occurs). For that, the number of messages used in the second version is less than the first one.

As a conclusion for this discussion, we can say that the second version of the protocol (IVP) is better than the first one thanks to the integrated tunnel maintenance algorithm. Even by providing security and anonymity in VANet, this protocol can allow an acceptable level of performance compared to the AODV protocol from the PDR and end-to end delay point of view, while mentioning the high values of the overhead, which refers to the additional messages used in this protocol to provide security and anonymity of the communication, which makes this version discussable to be improved and has good results.

V. CONCLUSION AND PERSPECTIVES

Critical requirements of VANet make development of security approaches and protocols difficult to implement. I2P as a secure protocol providing secure and anonymous communication uses heavy algorithms and mechanisms dedicated to internet which has completely different characteristics to VANet. This difference needs to be treated carefully by applying sophistic adaptation and amendments that allows the use of I2P in VANet.

In this paper, we create the second version of the proposed protocol in [1] by developing the tunnel maintenance algorithm. In which, tunnels can be created and maintained in real mobile scenarios of VANet. The simulation results show that the new version of the protocol is better than the first one thanks to the integrated tunnel maintenance algorithm.

In the end, we become able to implement important I2P mechanisms and algorithms providing security and anonymity in the network. Therefore, we have achieved our objective to secure and anonymize the communication in the vehicular ad hoc network.

As perspectives, we aim to enhance the proposed algorithms to better respond to VANet characteristics while maintaining a satisfactory level of security and anonymity in VANet.

In this paper, the IVP protocol uses the AODV routing protocol and it is executed with a speed less than 35km/h. As future contributions, we aim to use other routing protocols, implement realistic propagation models and rise the speed of vehicles.

Another work to do, concerns the tunnel creation process. In our contributions, we create tunnels through two steps; “Getting tunnel nodes identities” then “Tunnel creation”. As an improvement to this process, we will merge the two steps into one-step, in which we obtain the identities of tunnel nodes and create tunnels in the same time, which reduces the time needed to complete the tunnel creation process compared to the proposed protocol.

REFERENCES

[1] Tayeb Diab & Marc Gilg & Frederic Drouhin & Pascal Lorenz, “Anonymizing Communication in VANets by Applying I2P Mechanisms”, IEEE Global Communications Conference, 2019.

[2] Tayeb Diab & Marc Gilg & Pascal Lorenz, “A secure communication model using lightweight Diffie-Hellman method in vehicular ad hoc networks”, International Journal of Security and Networks, Vol. 14, No. 2, PP. 61-77, 2019.

[3] Bouksani, Walid & amar bensaber, Boucif. (2018). RIN: A dynamic pseudonym change system for privacy in VANET. Concurrency and Computation: Practice and Experience. 31. 10.1002/cpe.4719.

[4] Ying, Bidi & Nayak, Amiya. (2017). Anonymous and Lightweight Authentication for Secure Vehicular Networks. IEEE Transactions on Vehicular Technology. PP. 1-1. 10.1109/TVT.2017.2744182.

[5] Wang, Xiaoliang & Li, Shuifan & Zhao, Shujing & Xia, Zhihua. (2017). A VANET privacy protection scheme based on fair blind signature and secret sharing algorithm. Automatika. 58. 287-294. 10.1080/00051144.2018.1426294.

[6] Wang, Shibin & Yao, Nianmin. (2017). LIAP: A local identity-based anonymous message authentication protocol in VANETs. Computer Communications. 112. 154-164. 10.1016/j.comcom.2017.09.005.

[7] Jiang, Yanji, Shaocheng Ge, and Xueli Shen. AAAS: An Anonymous Authentication Scheme Based on Group Signature in VANETs. *IEEE Access* (2020).

[8] Aghabagherloo Alireza, et al. An Efficient Anonymous Authentication Scheme Using Registration List in VANETs. *arXiv preprint arXiv:2004.00282* (2020).

[9] HAN, Mu, LIU, Shuai, MA, Shidian, *et al.* Anonymousauthentication scheme based on fog computing for VANET. *PLoS one*, 2020, vol. 15, no 2, p. e0228319.

[10] Saddiki, Kamel, et al. "Black hole attack detection and ignoring in OLSR protocol." *International Journal of Trust Management in Computing and Communications* 4.1 (2017): 75-93.

[11] Aouiz, Amir Abdelkader, et al. "Network Life Time maximization of the AOMDV Protocol Using Nodes Energy Variation." (2018).

[12] https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/End-to-end_delay.html Accessed on 12/2020.

Declarations:

Funding: This study was funded by University of Haute Alsace.

Conflict of Interest//Competing interests: The authors declare that they have no conflict of interest and competing interests.

Availability of data and material: Not applicable.

Code availability: Not applicable.

Ethics approval: All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Consent to participate: Informed consent was obtained from all individual participants included in the study.

Consent for publication: We give our consent for the publication of identifiable details within the text to be published in the Wireless Personal Communications journal.