# Zero-knowledge Authentication for Self-managed Vehicular Networks

C. Caballero-Gil, P. Caballero-Gil and J. Molina-Gil

Department of Statistics, Operations Research and Computing,
University of La Laguna, 38271 Tenerife, Spain

**Abstract.** This paper proposes a new solution for the practical, fast and secure deployment of vehicular networks. Its main contribution is a self-managed authentication method that does not require the participation of any certification authority because the nodes themselves certify the validity of the public-keys of the nodes they trust, and issue the corresponding certificates that are saved in local key stores according to an algorithm here proposed. In addition, the new node authentication method includes a cryptographic protocol that each node can use to convince another node about the possession of certain secret without revealing anything about it. Thanks to all these tools, cooperation among involved vehicles can be used to detect and warn about abnormal traffic conditions. One of the most interesting aspects of the proposal is that the required devices can be simple existing mobile devices equipped with wireless connection. This work includes a performance analysis of a simulation of the proposed algorithms and the obtained results are promising.

## 1 Introduction

A Vehicular Ad hoc NETwork (VANET) is a spontaneous wireless network that allows providing communications among nearby vehicles with the primary goal of improving road traffic conditions. VANETs can be seen as a special type of Mobile Ad hoc NETworks (MANETs) where nodes are vehicles. In many situations, communications among vehicles might be used to prevent road accidents and to avoid traffic collapses. Consequently, a fast VANET deployment could help drivers to save time and money, and to reduce contamination of the environment and consumption of fuel reserves.

There are several general security requirements, such as authentication, scalability, privacy, anonymity, cooperation, stability and low delay of communications, which must be considered in any wireless network. Those requisites are even more challenging in VANETs because of specific characteristics such as no fixed infrastructure and rapidly changing scenarios that range from rural roads with little traffic to cities or highways with a huge number of communications. In particular, wireless authentication in VANETs has deep implications for privacy as it could lead to electronic tracking of vehicles. Consequently, VANET security may be considered one of the most difficult research topics that need to be taken into account before a wide deployment of such networks [11].

Caballero-Gil C., Caballero-Gil P. and Molina-Gil J..
Zero-knowledge Authentication for Self-managed Vehicular Networks.
DOI: 10.5220/0004098400390048
In *Proceedings of the 9th International Workshop on Security in Information Systems* (WOSIS-2012), pages 39-48
ISBN: 978-989-8565-15-0

The proposal here presented has as starting point the consideration that the introduction of a complete model of VANET including Road Side Units (RSUs) and On Board Units (OBUs) would be extremely expensive both for users, who would have to buy new cars or install specific devices in their vehicles, and for the State, which would have to deploy a huge infrastructure to support VANET services. Therefore, this work proposes a self-managed VANET that does not require any infrastructure, and might be used as a fast and secure introduction to more complex and complete VANETs.

This paper is organized as follows. Related work is reviewed in Section 2. In Section 3, proposals for the generation of public/private key pairs, for node characterization and for beacon management are included. A zero-knowledge authentication protocol is proposed and its simulation is shown in Section 4. In Section 5, a new method to save certificates in key stores is briefly described and its implementation is studied. Finally, the paper is concluded with Section 6.

## 2 Related Work



The main goal of this paper is the definition of a simple, scalable and cross-layer design for the immediate deployment of VANETs in order to exploit the potential of existing mobile systems. The proposed scheme involves that users can cooperate through their mobile devices and can start to obtain updated information of their interest about road traffic in order to choose the best updated route to their destinations. Our proposal takes into account that the practical implementation of VANETs will be gradual, without any RSUs or OBUs, and with only a few mobile devices at the beginning. The growth of VANETs will be faster or slower depending on their popularity, acceptance and ease of use. In this paper we focus on the first phase, when the number of cooperating devices on the road will be low. Once VANETs have grown, the model should be checked to avoid unnecessary communications that might degrade the network. This particular issue has been studied and possible solutions based on clusters have been proposed in [1], where specific characteristics of inter-vehicle and vehicle-to-roadside communications were taken into account to define different authentication services.

With respect to requirement minimization, several papers focus on different aspects and applications of VANETs. [9] proposes a parking notification scheme that does not need any extensive infrastructure, but in that paper RSUs are required in the supported parking spaces. [12] proposes a key management scheme for VANETs, which is used to authenticate messages, identify legitimate vehicles and remove malicious vehicles. However, such a proposal is based on the use of a public-key infrastructure, which involves several difficulties in VANETs.

There are other bibliographic references that propose different types of authentication schemes for self-managed VANETs, following approaches that are entirely different to the one here presented. [3] proposes an authentication scheme that is based on pseudonyms, while [8] describes a scheme that combines authentication, key establishment and blind signature techniques. With respect to public-key certification, [7] presents a method for certificate revocation based on car-to-car epidemic distribution, and [6] proposes another mechanism for revoking security certificates, which needs a certification authority and certificate revocation lists.

Another paper with the same general objective as this work is [5], but it does not address the communication security issue. Unlike the previous work, two papers that analyze privacy issues in VANETs are [10], which is based on asymmetric and symmetric cryptography, and [13], which uses session keys.

Neither of the aforementioned works proposes a self-managed and secure approach to VANETs, which is the main objective of this work. In particular, our proposal focuses on allowing the immediate and secure deployment of VANETs through existing devices.

## 3 Components of the Proposal

The authentication proposal is based on a Zero-Knowledge Proof (ZKP), which is a cryptographic protocol that a prover can use to prove possession of a certain piece of information to a verifier without revealing anything about it. During the authentication procedure, the prover, denoted $A$, must answer to a number of challenges issued by the verifier, denoted $B$. The admission control included in the authentication proposal described below uses the general scheme of ZKP described in [2] based on the graph isomorphism problem, for the particular case of the Hamiltonian Cycle Problem (HCP), which involves the determination of whether a cycle that visits each node exactly once exists in a graph.

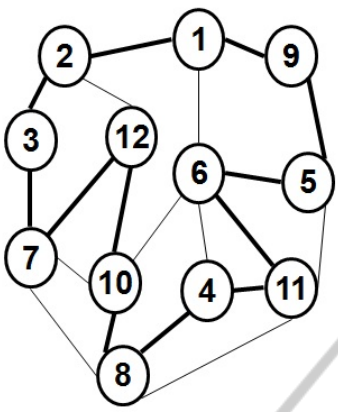
In the scheme here used based on certificate graphs [4], each node $A$ has a private/public key pair and a key store ($KeyStore_A$) including a list of all node certificates that $A$ trusts. The set of stored public-keys and certificates might be represented as an undirected graph $G = (V, E)$, known as certificate graph, in which each vertex represents both a public-key and its owner, and each edge $(A, B)$ symbolizes two public-key certificates: of node $A$ signed with the private key of node $B$ and vice versa. A certificate chain is an undirected path in a certificate graph. The subgraph $G_A$ of the certificate graph $G$ contains exactly the current certificates stored by node $A$ in $KeyStore_A$.

The generation of the public/private key pairs, and the characterizations of nodes and management of beacons are explained in the following subsections.

### 3.1 Public/Private Key Pair Generation

In order to make it easier the implementation of the ZKP for the Hamiltonian cycle in the node authentication process described below, the public-key is computed from the decimal value of the binary representation for the upper triangular submatrix of the symmetric adjacency matrix containing the elements corresponding to a Hamiltonian cycle in a graph (see Figure 1). Such a decimal number corresponding to the binary representation is used in the proposal as the exponent of the public-key in RSA encryption used by the mobile device to encrypt and decrypt messages and to sign public-key certificates.

Figure 2 shows a trace of an election of a public/private key pair by using the HCP for the generation of the public-key. After choosing the prime numbers $p$ and $q$, the public exponent $e$ is generated from a random Hamiltonian cycle, so that it is lower than and coprime with $(p-1)(q-1)$. Afterwards, the private exponent $d$ is generated.




| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | **1** | 0 | 0 | 0 | 1 | 0 | 0 | **1** | 0 | 0 | 0 |
| 1 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | **1** | 0 | 0 | **1** | 0 |
| 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | **1** | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | **1** | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | **1** |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | **1** | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | **1** |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

**$KU_{ID}$: 10000001000100000000000010000000010010100100000001000001010000010**

**Fig. 1.** Example of Hamiltonian Cycle Based Public-Key.



```
PRIVATE INFORMATION:
--------------------
 p = 24247
  Is prime.

 q = 25357
  Is prime.
choosing e lower than fi=614781576 which is coprime
Random list: 4, 6, 5, 2, 3, 1,
 Matrix:
001100
001010
110000
100001
010001
000110
8192 , 4096 , 512 , 128 , 2 , 1 ,
 PUBLIC EXPONENT e=12931 ,
  Prime with Fi(n) = (p - 1) (q - 1)614781576.

PRIVATE INFORMATION:
--------------------

 MODULO n = 614831179

 PRIVATE EXPONENT  d = 439014235
```

**Fig. 2.** Implementation of Hamiltonian Cycle Based RSA.

### 3.2 Node Characterization and Beacons

The present proposal assumes that each node in the network is characterized by the following parameters:

$ID, (KU_{ID}, KR_{ID}), (ID_i, KU_{ID_i}, Cert(KU_{ID_i}))_{ID_i \in KeyStore}$

which include:

– A unique IDentifier (denoted ID), obtained as the output of a one-way function on a single value. For example, if the used device is a mobile phone the value can be its

number, while in other cases an email address might be used. The one-way function could be any hash function.
- A fixed public/private key pair (denoted (KU,KR)) and called identity keys, which are used in an asymmetric cryptosystem such as RSA.
- A key store containing various IDs and corresponding public-keys and certificates, which the node keeps always updated.

Sending multicast beacons containing variable sender IDs are required both for the active node discovery process and also to avoid vehicle tracking. In the same step where beacons are sent, each node commits to its secret by sending to its neighbours also a witness of its secret. The variable ID of each node that is sent as part of its beacon is the hash of the IDs that are present in its key store at that moment.

In particular, the beacons sent by a node are formed by the following elements:

- Frame-Control (FC), which indicates the type of data being sent.
- Pseudonym (Pseu), which is a temporal ID of the node.
- Timestamp (Time), which allows knowing the specific time when the information was generated.
- Pair formed by public-key and timestamp (KU, Time) encrypted with the private-key (KR) of the node, which is used by nodes who have already authenticated it when its Pseu changes.



## 4 Authentication

Within this proposal, the device associated to each network node should be able to generate its public/private key pair and also to sign the public-keys of other nodes that are trustable and want to become part of the network. In order to be able to authenticate its public-key, every node must exchange signatures with a number of legitimate network nodes that depends on the width of the VANET. In the birth of the VANET, two signatures are considered enough to prove that the user is reliable and cannot self-sign certificates to compromise the network security, but the number of required signatures must grow with the expansion of the VANET.

When a node $A$ wants to check the validity of the public-key of another node $B$, $A$ must find a certificate chain from it to $B$ in the certificate graph that results from merging the subgraphs $G_A$ and $G_B$ corresponding to $KeyStore_A$ and $KeyStore_B$ respectively. In particular, the authentication process of the public-key of a node $A$ by another node $B$ and vice versa, is based on a chain of correct and not expired certificates between $A$ and $B$ in the graph resulting from the union of the two key stores because:

1. The first certificate in the chain can be verified directly by $A$ (respectively $B$) because it was signed by itself.
2. Each of the other certificates in the chain can be verified by using the public-key of the previous certificate in the chain.
3. The last certificate is $B$'s public-key (respectively $A$).

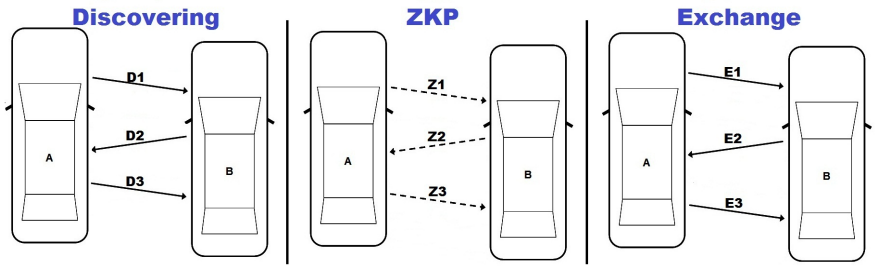



**Fig. 3.** Self-Managed Authentication Protocol.

Special packets are sent between users to authenticate each other. Among other information, they contain the data FC, source Pseu and destination Pseu. Figure 3 shows schematically the three phases of interaction included in the proposed self-managed protocol for authentication between two nodes $A$ and $B$.

The phases are fully described below. The first phase is the discovering process, which includes part of the beacons sent by nodes $A$ and $B$, containing the hash of the IDs stored respectively in $KS_A$ and $KS_B$. Within this phase, both nodes find out whether a common public-key $x$ exists in both keystores $KS_A \cap KS_B$. In such a case, a graph is generated by each node from such an element. Those graphs $G_A(x)$ and $G_B(x)$ are used in the second phase, based on a ZKP, to mutually prove the knowledge of the common key $x$. In this way, during the last phase, both nodes are sure that they can use the shared key $x$ to exchange their temporal secret keys and key stores.



**Algorithm 1:** Authentication Scheme.

**function** $Authentication\_Scheme()()$ (...)

D1. $A \rightarrow B$: beacon with $\{h(ID_i) : ID_i \in KS_A\}$
D2. $B \rightarrow A$: $\{h(ID_i) : ID_i \in KS_B\}$ and a graph $G_B(x)$, if $\exists x \in KS_A \cap KS_B$
D3. $A \rightarrow B$: a graph $G_A(x)$ if $\exists x \in KS_A \cap KS_B$
Z1. $A \rightarrow B$ $(B \rightarrow A)$: a graph $GI_A(x)(GI_B(x))$ isomorphic with $G_A(x)(G_B(x))$
Z2. $B \rightarrow A$ $(A \rightarrow B)$: a binary random challenge $b(a)$
Z3. $A \rightarrow B$ $(B \rightarrow A)$: if $b = 0(a = 0)$ $GI_A(x) \approx G_A(x)(GI_B(x) \approx G_B(x))$
Z3. Otherwise a Hamiltonian circuit in $GI_A(x)(GI_B(x))$
E1. $A \rightarrow B$ $(B \rightarrow A)$: $E_x(KU_A)(E_x(KU_B))$
E2. $B \rightarrow A$ $(A \rightarrow B)$: $KU_A(K_B)(KU_B(K_A))$
E3. $A \rightarrow B$ $(B \rightarrow A)$: $E_{KB}(KS_A)E_{KA}(KS_B)$

**end function**

The above algorithm allows any node to authenticate another node as well as to exchange both secret keys to update both key stores.

Figure 4 shows an implementation of the proposed authentication scheme performed using Microsoft Visual Studio in $C\#$.

A client-server capable of multiple connections at the same time is implemented in each device. All signals about authentication and beacons are performed with UDP

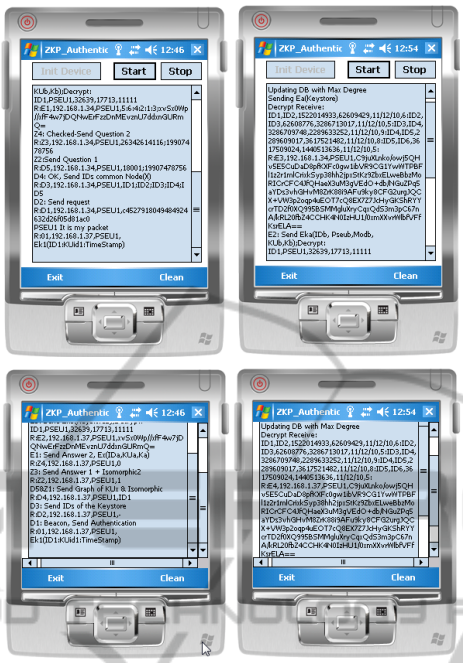



**Fig. 4.** Implementation of the Authentication Scheme.



packets. Each client broadcasts beacons periodically to all connected devices in the network. Each beacon is formed by the following data:

*"01," + thisIpAddr + "," + PSEU + "," + Ek1(ID1,KUid1,TimeStamp)*

Before starting to use the device, the node needs information to communicate with other devices, and in particular a database with three tables is loaded. These tables keep data for a low number of users whose data (certificates and public key) are generated with the generator:

*certificateStore (idcolumn INT PRIMARY KEY, idA NTEXT, idB NTEXT, certAB BIGINT, certBA BIGINT, date DATETIME);*

*keyStore (idcolumn INT PRIMARY KEY, idA NTEXT, PseuA NTEXT, module BIGINT, publicKey BIGINT, secretKey BIGINT, degree INT );*

*myStore (idcolumn INT PRIMARY KEY, idA NTEXT, PseuA NTEXT, modulo BIGINT, publicKey BIGINT, privateKey BIGINT, secretKey BIGINT, degree INT );*

Incoming connections are managed on the server so that when one is received, the server checks the identity of the node who sent the packet. After that, it checks whether the node is already authenticated in the network, and if not, the authentication protocol begins. The procedure the nodes use to send and receive information as indicated in Algorithm corresponding to the Authentication Scheme.

## 5 Key Store Update

For the self-managed VANET here proposed, it is required that each node has its own key store to authenticate other nodes. In these networks the number of users might be huge, so we propose a scheme for storage of public-key certificates that exploits the aforementioned principle of six degrees of separation. Thanks to such a property, it is not necessary that each user stores the certificates of all nodes to be able to authenticate them. Instead, only a minimum number of certificates have to be stored by each node so that by merging the key stores of two users who want to authenticate each other, the probability to find at least one certificate chain in the merged graph will be high.

Consequently, the optimal update of the key stores is an important part of the proposal as it allows limiting the number of stored keys to a value here denoted $lim$. Such a value is generally less than the number of users forming the network, and equal to the minimum number that allows any node to connect to any other node in the network.

In order to maximize the probability that any node is able to authenticate to any other node while limiting the size of the key stores, different algorithms to update the key stores can be used. A possible algorithm is proposed below. To update its key store, each node chooses those public-key certificates corresponding to nodes that have issued or received more valid certificates, what is represented by the degrees of the vertices is the corresponding certificate graph. This choice maximizes the probability of intersection between key stores, what is necessary for the authentication process.



**Algorithm 2:** Key Store Update.

01:**function** Update_KeyStore() (...)
02: Initialize data structures;
03: Union:= $KS_A \cup KS_B$;
04: $KS_B = \{B\}$
05:**for** each $i \in KS_B$
06: **for** each $j \notin KS_B : (i,j) \in Union$
07: **if** ((degree $(j) = max(degree(neighborofiinUnion)))$
$\&\&(cardinal(KS_B) < lim))$
08: $(i,j) \in (KS_B)$
09: **end if**
09: **end for**
09: **end for**
09: **end for**
10: **end function**

An implementation of the proposed key store update scheme has been performed with the Network Simulator tool NS-2. In the performed simulation, an initial wireless network where nodes are randomly located produces the first certificate graph. Each node saves in its local key store the certificates of the nodes at distance 1. Then, the nodes begin to move randomly, and when two nodes are at distance 1, they verify whether they can trust each other and initiate a key store exchange for key store update. New nodes can form part of the network by inserting new certificates in the certificate graph. Also, any node can get out of the certificate graph if its certificate is not renewed.

After performing 25 simulations for 15, 20, 30 and 60 nodes, the average results of executions with different types of networks show that performance may be considered in general acceptable. According to the simulations we can also conclude that the scheme is affected by the mobility of the nodes because a higher mobility leads to a faster increase and balance of the key stores. Thus, this is a convincing argument for considering vehicular networks, since these are high mobility networks.

| | 15 nodes | 20 nodes | 30 nodes | 60 nodes |
|---|---|---|---|---|
| Total Connections | 966,9 | 1011,52 | 2764,7 | 5309,0 |
| Successful Connections | 909,7 | 985,4 | 2749,8 | 5216,5 |
| Failed Connections | 57,15 | 26,12 | 14,92 | 92,5 |
| Added Information | 102,28 | 56,4 | 80,51 | 191,9 |
| Key Store Updates | 628,7 | 420,2 | 690,24 | 1475,9 |

## 6 Conclusions

This paper proposes a self-managed authentication method for VANETs that does not require deploying any infrastructure on the road or any special equipment on vehicles, what allows a gradual introduction of VANETs without any economic investment. To make it possible, existing devices like smartphones can be used to run the algorithms here proposed. The main contributions of this paper are a self-managed node authentication protocol based on public/private key pairs and certificate graphs, and a vehicle discovering scheme with variable pseudonyms to protect node privacy and prevent vehicle tracking. Another contribution of this paper is an algorithm to update key stores that maximizes the probability of communication between any pair of nodes. Our proposal allows the use of a hybrid scheme that combines secret and public-key cryptography, what can be used both to optimize resources and to enforce node cooperation by avoiding passive behavior of nodes. All the proposed algorithms have been simulated with the Network Simulator NS-2 and implemented with Microsoft Visual Studio in $C\#$ in mobile phones. The obtained results in both cases show a high level of performance.



## Acknowledgements

Research supported by the Ministerio de Ciencia e Innovación and the European FEDER Fund under Project TIN2011-25452 and FPI scholarship BES-2009-016774, and by the ACIISI under FPI scholarship BOC Number 60.

## References

1. Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Hernández-Goya, C.: Flexible Authentication in Vehicular Ad Hoc Networks. 15th Asia-Pacific Conf. Communications, pp. 576–879 (2009)
2. Caballero-Gil, P., Hernández-Goya, C.: Zero-Knowledge Hierarchical Authentication in MANETs. IEICE - Trans. Inf. Syst. E89-D(3), 1288–1289 (2006)

3. Calandriello, G., Papadimitratos, P., Hubaux, J. P., Lioy, A.: Efficient and Robust Pseudonymous Authentication in VANET. VANET '07: 4th ACM international workshop on Vehicular Ad Hoc networks, pp. 19–28. ACM, New York, NY, USA (2007)
4. Capkun, S., Buttyan, L., Hubaux, J. P.: Self-organized Public-Key Management for Mobile Ad Hoc Networks. IEEE Transactions on Mobile Computing. 2(1), 52–64 (2003)
5. Dornbush, S., Joshi, A.: StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's. VTC2007-Spring Vehicular Technology Conf. pp. 11–15 (2007)
6. Haas, J. J., Hu, Y. C., Laberteaux, K. P.: Design and Analysis of a Lightweight Certificate Revocation Mechanism for VANET. VANET '09: Sixth ACM international workshop on VehiculAr InterNETworking, pp. 89–98. ACM, New York, NY, USA (2009)
7. Laberteaux, K. P., Haas, J. J., Hu, Y. C.: Security Certificate Revocation List Distribution for VANET. VANET '08: Fifth ACM international workshop on VehiculAr Inter-NETworking, pp. 88–89. ACM, New York, NY, USA (2008)
8. Li, C. T., Hwang, M. S., Chu, Y. P.: A Secure and Efficient Communication Scheme with Authenticated Key Establishment and Privacy Preserving for Vehicular Ad Hoc Networks. Comput. Commun. 31(12), 2803–2814 (2008)
9. Panayappan, R., Trivedi, J. M., Studer, A., Perrig, A.: VANET-Based Approach for Parking Space Availability. VANET '07: Fourth ACM international workshop on Vehicular Ad Hoc Networks, pp. 75–76. ACM, New York, NY, USA (2007)
10. Plobl, K., Federrath, H.: A Privacy Aware and Efficient Security Infrastructure for Vehicular Ad Hoc Networks. Comput. Stand. Interfaces 30(6), 390–397 (2008)
11. Raya, M., Hubaux, J. P.: The Security of Vehicular Ad Hoc Networks. 3rd ACM workshop on Security of Ad Hoc and Sensor Networks, pp. 11–21 (2005)
12. Studer, A., Shi, E., Bai, F., Perrig, A.: TACKing together Efficient Authentication, Revocation, and Privacy in VANETs. 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 484–492 (2009)
13. Wang, N. W., Huang, Y. M., Chen, W. M.: A Novel Secure Communication Scheme in Vehicular Ad Hoc Networks. Comput. Commun. 31(12), 2827–2837 (2008)