Scientific Research Publishing

Journal of Computer and Communications, 2020, 8, 22-35
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

VANET Security through Group Broadcast Encryption

Eric Eduardo Bunese, Eduardo Todt, Luiz Carlos Pessoa Albini

High Performance and Efficient Systems Group, Department of Computer Science, Federal University of Paraná, Curitiba, Brazil
Email: ericbunese@gmail.com, todt@inf.ufpr.br, albini@inf.ufpr.br

How to cite this paper: Bunese, E.E., Todt, E. and Albini, L.C.P. (2020) VANET Security through Group Broadcast Encryption. *Journal of Computer and Communications*, **8**, 22-35.
https://doi.org/10.4236/jcc.2020.88003

Received: July 8, 2020
Accepted: August 8, 2020
Published: August 11, 2020

Abstract

VANET security is an evolving topic in mobile networks, as providing a secure layer of communications in such a dynamic and fast network is a challenge. The work presented in this article was conducted in order to verify and evaluate the feasibility of applying group broadcast cryptography to the VANET environment, as an attempt to gain performance by decreasing the number of messages in the wireless network. Group broadcast is a symmetric/asymmetric hybrid cryptography method, aiming to merge the best of the two approaches without their major drawbacks. Simulations were set-up and run using the ONE simulator, comparing the usage of the three different cryptography approaches for VANETs. Results consider the number of connections, the number messages and the number of revocation messages per day. The resulting data promises that group broadcast encryption can be used to simplify the encrypting phase, reduce required storage and significantly decrease the number of messages in the network.

Keywords

VANET Security, Group Broadcast Encryption, Group Based VANET

1. Introduction

Cellular technology and mobile devices already connect vehicles to the Internet and other networks, as they are able to plot online courses on the GPS/Cellular, stream music in the entertainment media center via Bluetooth and provide a hands-free solution for using the phone. Extending the connectivity creates a new range of applications, as vehicles become capable of following other's positions and predicting their physical movements. Accidents can be prevented, road hazards can be notified, and the creation of dangerous events in traffic decrease along with the development of these solutions [1].

In VANETs, any vehicle with network capabilities can be regarded as a node. This node can exchange information with other nodes in the network using wireless technology such as Wi-Fi, Bluetooth, 4G or 5G. Along with moving nodes, static nodes can take part in the network in the form of Road-side Units (*RSUs*) or cloud computation. Static nodes provide a trusted environment for moving nodes, and usually help in the connection or routing phases of the VANET [1]-[8].

Messages exchanged in the VANET can be separated in two groups. *Vehicle-to-vehicle communication*, usually called as *V2V*, defines messages shared between vehicles. While *Vehicle-to-infrastructure*, *V2I*, defines the information exchange between mobile and static nodes, such as RSUs, traffic lights, radars or even buildings.

In most related work, it is also common for nodes to have a tamper-proof device on-board, known as the On-board unit (*OBU*), that is responsible for computing cryptography and holding security information such as trusted certificates and public keys [1] [2] [9]. The OBU is generally implemented as a separate secure computer on board of the vehicle, with its own processing, storage and network capabilities.

Table 1 presents the different types of attackers in VANETs, as presented by [3] [5]. Active attackers are those that fully participate in the VANET, sending and receiving messages normally, as expected from any node. However, these attackers may attempt to extract information from the network and inject false or misleading information for their benefit. Passive attackers, on the other hand, do not fully participate in the network. They lurk the VANET waiting for interesting information or the perfect opportunity to attack. Usually, this type of attacker only extracts information. Internal attackers are active nodes in the VANET. It can be performed by either adding a node to the network or by corrupting an existing node. This type of attacker is very difficult to detect, as they might even have proper certificates required to participate in the VANET. Implementing hardware security such as temper-proof devices or adding trust management to the network helps prevent and detect such attackers. External attackers do not take part in the VANET. Such attackers generally intercept or steal data outside the network, such as capturing wireless signals or intercepting authority requests.

In order to bring traffic safety and practical applications to life via a VANET, it is imperative that proper security layers are implemented. However, the dynamic network topology and latency sensitive information transmitted contribute to

Table 1. Types of attackers.

	Active	**Passive**
Internal	Participate in the network by sending and receiving data from other nodes.	Participate in the network, but only read and extract data from within.
External	Create input for nodes from outside the network and can intercept messages to obtain data.	Intercept data in the network, usually known as a man-in-the-middle.

a more vulnerable environment, as it is not viable to apply market standard security implementations [4].

Recent reviews in VANET security [4] [5] present requirements and existing vulnerabilities in the technology. In order to create a fully secure network, five properties need to be ensured:

Privacy: the use pseudonyms is almost mandatory to provide privacy. Connected nodes only need to share their pseudonym with each other, while the identity is only exchanged between the soliciting node and the trusted certificate authority. No other personal or private information can be derived from the pseudonym. Asymmetric cryptography can ensure user privacy.

Non-Repudiation: it must prevent nodes from repudiate or deny any previous behavior on the network. It can be achieved through messages signature, as only the owner of the private key can use it to sign and authenticate messages.

Availability: the network must be available, and all the dependencies must be accessible at any time. Network infrastructure and trusted authorities help protect availability.

Integrity: exchanged messages must be received in pristine condition, as they cannot be modified. One way to guarantee Integrity is through messages signature.

Authenticity: exchanged messages can never be modified. Every message must be verified along with its origin. Messages encryption and signature can guarantee Authenticity.

It is possible to notice that cryptographic algorithms play a central role to guarantee such properties. Symmetric cryptography solutions require that two nodes share a common key to cipher and decipher messages. Once a pair of nodes has agreed upon a key, they apply the algorithm and key to the message generating a new string to be sent in the network. In VANETs, it is not practical to store and apply a different key for every neighboring node in the network, as it might be necessary to cipher a single message dozens of times. When using symmetric cryptography for many different nodes, storing and identifying which key should be used can also become a challenge.

Current market algorithms for asymmetric cryptography such as RSA, ECDMA can be applied to the VANET context. Vehicles can share their public key, which will be used to cipher information to be sent in the network. Whenever a destination node receives a message, it can attempt to decipher the information using its private key. Managing the VANET on an asymmetric environment might be a challenge, though. Having a key that must be used in order to cipher information to every destination node means using a lot of storage and processing power spent on securing the same information. Doing so dozens of times for every outgoing message will become a burden and generate delays in the network. However, when deciphering messages, nodes only need to store and utilize their own private keys, meaning that the cardinality of received messages is always one. Several solutions created for securing VANETs rely on using traditional asymmetric cryptography, such as [7] [8] [10].

This work proposes the use of *Group Broadcast Encryption* [11] as a security

framework for VANETs. Group Broadcast Encryption merges the benefits of symmetric and asymmetric algorithms without any additional drawbacks, presenting a flawless solution. Simulations were run using The ONE simulator, comparing three different cryptography algorithms on top of a VANET. Simulation results demonstrate that group broadcast encryption presents asymmetric-like security, with a symmetric-like complexity, decreasing encryption and decryption times and the number of messages in the network.

The remaining of this article include the general concept of Group Broadcast Encryption and how it can be applied in a VANET environment (Section 2), the cost analysis of this implementation (Section 3), simulation setup and algorithm comparison (Section 4), discussion and simulation results (Section 5), followed by the conclusions (Section 6).

2. Group Broadcast Encryption

A Group Broadcast Encryption system would simplify the ciphering and signing phases of a secure communication system in the VANET. Using such a solution could provide an asymmetric like security, while consuming fewer resources, like using a symmetric algorithm, while decreasing the number of messages sent in the network. Using group broadcast encryption implies the creation of a hybrid private/public key-pair. In this solution, several public keys or pseudonyms are used and processed into a new private key [11]. Messages encrypted with a private key that was created using N public keys can only be accessible by the corresponding N private keys. This property ensures the authenticity, privacy, non-repudiation and integrity of the system. Every vehicle in the network is responsible for keeping its own 'Group View', managing the authorized nodes that will be able to read messages sent by it. Whenever a new vehicle is added to a source node's group view, the incoming vehicle's public key is added into the group view private key, enabling this vehicle to use its own private key to read received messages. The source node can also revoke destination nodes, simply by subtracting their public key from the group view private key. Using this solution, it is very likely that two different source nodes could have a different group view, and messages sent from source *A* might not be readable by authorized destinations for *B*. However, handling this situation is not necessary, as it's up to each source node to protect its information. Data can be replicated by source node *B*, repeating the information received from source *A* to its authorized destinations, however, *A*'s privacy is ensured due to the use of pseudonyms, allowing only the replication of VANET sensitive information, and not vehicle sensitive information.

Given the current situation on VANET security, we define another possible configuration for speeding-up message verification that maintains a more secure environment, relying on Group Broadcast Communication, providing symmetric-like efficiency and asymmetric-like security.

2.1. Identity

In order to protect a given node's identity in the network, pseudonyms should

be created and used in the VANET. At any given point, the node can set-up the pseudonym certificate and keys using the Internet (HTTPS), communicating directly to a competent organ responsible for long term verification of vehicles. This can be done either by using Wi-Fi access points, 3G, 4G, LTE and 5G, or by using a Road-Side Unit (RSU) as a trusted access point. Once the vehicle has a verifiable pseudonym, it is eligible to connect to a VANET network. A pseudonym is used in order to protect the node's privacy, keeping its true identity safe, but still traceable by an authority if needed. Algorithm 1 describes the set-up step.

2.2. Setting up Groups

Accessing the VANET is a simple process; groupless nodes broadcast their pseudonyms to the local network and wait for connection responses. If no reply is received, it's possible to create a new group and start waiting for other vehicle's broadcasts. Whenever a vehicle is requesting to join a group, each group member can verify the joining pseudonym certificate using known certificate authorities (CA) public keys, or, by polling the CA directly using a wireless internet link (HTTPS), in order to verify the incoming member online. Whenever a group member fails to authenticate the joining node, it will let the others know by sending a revocation message in the group, and then will not include the newcomer's public key into its private group view key.

Vehicles in favor of adding the newcomer to the group will add its public key to their own private group view key, enabling it to decipher messages sent by these vehicles in favor. Afterwards, they use this new key to send a connection message to the newcomer, giving them their public key, so that they can create their own private group key. Algorithm 2 and Algorithm 3 present the two sides of the group creation process.

Algorithm 1. Getting a verified certificate.

1: Given a node A and the Certificate Authority CA
2: A sends its identity and a new certificate to CA, using HTTPS on top of a wireless access-point.
3: CA will sign A's certificate with its private key.

Algorithm 2. Group joining process.

1: Given a node A, a time threshold T.
2: A broadcasts its signed certificate and waits to be invited to a VANET group.
3: If the time T passes and no invitations were received, A will create a new group, alone.
4: If an invitation was received within the time threshold T, A will add the incoming public keys to its private group key, thus, joining the group.

Algorithm 3. Replying to a group join request.

1: Given a node A and an incoming signed certificate C from node B.
2: A attempts to verify the certificate using known Certificate Authorities' public keys.
3: If the certificate is valid, A adds C to its private group key, and sends its public key to Node B.
4: If the certificate cannot be verified, A ignores the request, and warns its group that it could not verify C.

2.3. Exchanging Information

Once every node's private group key has been updated to include the newcomer, messages can be sent within the group using the wireless link. It is possible to send information in three different levels of security:

Plain messages: Simply uses the wireless link to send open information. This is the fastest way to transfer data between nodes, but provides no security, privacy or authentication. Algorithm 4 shows this exchange.

Algorithm 4. Sending a plain message in the network.

1: Given two nodes A and B;
2: A broadcasts the message M;
3: B receives and interprets the message m from A.

Group messages: A source node S sends the message M to the group, using its group view private key, protecting the message from any receiving node whose public key was not used to compose S's private group key. Algorithm 5 shows this exchange.

Algorithm 5. Sending a group message in the network.

1: G_1 ciphers the message m using its private group key and broadcasts it to the network.
2: $G_1 \Rightarrow PKG_1[m]$
3: $G_2 \cdots G_N$ receive and decipher the message using their group keys.
4: $G_2 \cdots G_N \Rightarrow M = PKG_i[m]$

Secure group message: A source node S sends the message M to the group. The message is signed using S's private key, then ciphered using the private group key. This extra layer of authentication is recommended for control messages within the group, such as when S attempts to deny access to a newcomer or kick and revoke an untrustworthy member. Algorithm 6 shows this exchange.

Algorithm 6. Sending a verified group message in the network.

1: Given N nodes in the group $G_1 \cdots G_N$.
2: G_1 ciphers the message m using its private group key, signs it with its private key, and broadcasts it to the network.
3: $G_1 \Rightarrow PKG_1[M + PKG_1(m)]$
4: $G_2 \cdots G_N$ receive, decipher and verify the message using their group keys.
5: $G_2 \cdots G_N \Rightarrow M = PKG_i[m]$

Member revocation: Revoking a group member is a very simple process. The source node S removes the malicious node's M public key from its private group key. After this, whenever S sends a message, M cannot read it. S can also notify its other neighbors of this process in order to build a trust management system between the nodes. Algorithm 7 presents this situation.

Algorithm 7. Member revocation.

1: Given a source node S, a malicious node E and n other nodes in S's group view.
2: S determines that E is malicious and warns the n vehicles that E is being revoked.
3: Message $GK[PK_n(REVOKE(E)]$ is sent to the network.

2.4. Bridging Groups

As vehicles possess their own group view, it is possible that two neighbors have quite different groups. It is possible to allow the communication between these different groups. Bridging is the process when a message is sent between different groups. **Algorithm 8** presents this concept.

Algorithm 8. Group bridge cost analysis.

1: Given two vehicles V_1 and V_2, and their respective group views G_1 and G_2.
2: V_1 sends a message m all members of G_1, which includes V_2.
3: V_2 receives this message and repeats it to all members of G_2.

2.5. Trust Management

While Trust Management is not in the scope of this paper, it is important to note its effect when applied to VANETs and Group Broadcast communications [12]. When vehicles are responsible for tracking other vehicles actions in the network, a score can be used to represent how trustworthy neighboring nodes are to a source node. Through testing or behavior analysis, a source node can detect and distrust a malign neighbor in the network and act in order to protect itself and other trustworthy neighbors. If enough secure nodes distrust a potentially malign neighbor, it can be effectively removed from participating in the network, by having the source nodes revoke its part in their private keys. Further actions can be taken in order to investigate the malign intent in the network, by reporting the suspect to a competent organ, which will be able to trace the origin of the pseudonym certificate.

In this particular set-up, every node within a local group can have their own "group view". A group view is a subset of the actual local network group, and the source vehicle manages its trustworthy connections within it. It can be viewed as a direct graph.

Nodes are responsible for their own group views, deciding which vehicles should they keep as destinations and which should be revoked. This decision process happens based on the input of a trust management system [12] defines a solution for trust management in VANETs, where every node will evaluate how much it can trust its neighbors. If a trust weight is past a threshold value for a single node, that node can be revoked from the group view. Keeping the trust value for neighboring nodes is also important for additional judgment on top of received messages. Whenever a node receives a message from another node, the content can be ignored or processed, depending on how much the source node is trusted.

3. Implementation Cost Analysis

In this section, a mathematical cost analysis is presented concerning the number of messages required to maintain the system working. The work presented here is derived from [11]. Table 2 presents a set of symbols used; it is based on [11].

Table 2. Notation for key management.

Item	Description
G_1	a cyclic additive group with order p
G_2	a cyclic multiplicative group with order p
e	a bilinear pairing in which $e: G_1 \times G_1 \to G_2$
$H_1(x)$	hash function in which $H_1(x) = \{0,1\}^* \to G_1^*$
$H_2(x)$	hash function in which $H_2(x) = G_2 \to Z_p^*$
$N_{\mathcal{F}}$	Founder nodes
N_i	Identification of node i
SK_i	Private key of node i
PK_i	Public key of node i
MSK	Master private key of system
MPK	Master public key of system
MSK_i	Share of master private key hold by node i

Decreasing the average amount of messages is an interesting approach towards VANET security efficiency. Fewer messages on the network directly contribute to reducing communications delay and interference, while also keeping processing time and power usage low on the CPU, as it is not required to compute many different encryption keys to the same message. Finding the ideal balance between security and efficiency should enable VANETs to operate in a low latency, fast response time environment as it's expected.

3.1. Group setup

Setting up the group requires joining vehicles to broadcast their verified public keys to nearby vehicles. Whenever an existing group captures this message, some sort of a vote, as described in Algorithm 9, takes place to determine if the joining vehicle should be a part of the group. As explained in the algorithm, the number of messages required should be linear to the number of nodes in the existing VANET group. Considering a set of founding nodes with m members, the cost to initialize the key management, denoted by IC, is:

$$IC = m \cdot (m-1) \cdot \text{size}\left(f_m^i(x)\right)$$

in which $\text{size}\left(f_m^i(x)\right)$ is the size of each sub-share of the MSK generated by nodes. As nodes must be close during the initialization phase, hop count is not considered [12].

Algorithm 9. Group setup cost analysis.

1: Given a candidate node C, and N vehicles in a group $G(V_1 \cdots V_N)$.
2: C begins to broadcast its signed certificate.
3: K vehicles ($1 \le K \le N$) receive C's certificate and copy the message to the group.
GK[PRVi(m)]xK messages are sent to the group.
4: J vehicles ($0 \le J \le N$), who could not verify C's certificate, should notify others.
GK[PRvi(NOT(C))]xJ messages are sent to the group.
5: Vehicles who could verify C or accept the neighbor's view on the matter, should transmit their public keys to C, and add C's public key to their private group key $PUc[m]$.
6: In total, up to $2N + 1$ messages are sent in the network to setup a new vehicle to the VANET group.

When a new node joins an existing group, the communication overhead is defined as follows: Considering that a new node contact $\Omega \ge t$ members of the set of founding nodes (Ω) in order to request authorization to act as a group member, the cost for a new member to join the group, denoted by NM, is:

$$NM = \left(\Omega \cdot \text{size of } (ReqMsg) + \Omega \cdot \text{size}\left(f_{new}^{i}(x)\right)\right) \cdot \Delta h$$

in which $ReqMsg$ is the message sent by n_{new} to the nodes of Ψ, $f_{new}^{i}(x)$ is each sub-share of MSK sent to n_{new} and Δh is the average of hops between nodes [11].

3.2. Revocation

Whenever a node is detected as malicious, or cannot be trusted anymore by a source node, the source node warns the rest of the group that it is revoking the malicious node from its private group view. While this should pose no effect to other nodes, it can be taken as input for trust management solutions. **Algorithm 7** presents this step.

The cost to revoke the private key of a given node N_b depends on the number of nodes which have considered N_b compromised. Each node which detects the misbehavior of N_b sends a accusation message to all nodes of the group. Thus, considering γ accusers, the key revocation cost is:

$$(\gamma \times t) \cdot \text{size of } (AcMsg) + (t)^2 \cdot \text{size of } (revMsg) + BcastMsg$$

in which $AcMsg$ is the accusation message sent by accusers to the whole group, $revMsg$ is the revocation message and $BcastMsg$ is the broadcast encryption message sent to all nodes [11].

3.3. Sending a Message

Whenever a vehicle needs to send a message to the VANET, all it has to do is encrypt the message using its private group key, composed with all of its neighbors' public keys. Using this sort of cardinality, it is possible to considerably decrease the number of messages in the network, when compared to maintaining several different encryption keys for different destination nodes.

4. Security and Performance Benchmark

In this section, the environment preparation, simulations and results are presented.

4.1. Simulation Set-Up

In order to evaluate the solution, a VANET simulation environment was used to generate the movement model that makes sense for VANETs. The Opportunistic Network Environment [13] was the chosen application to execute these simulations. As it is a widely regarded simulation tool for VANETs, easily extendable and simple to set-up. Our simulations have been run on the Helsinki city map, the standard map used in the simulator, also using the Working Day Movement Model [14], which will route vehicles in the map from their homes, to their work location and back. Some vehicles also run errands during the day and after the work hours, going by the city with less traffic.

Simulations were run thirty times for twelve hour days with an eight hour work shift.

4.2. Simulating Security Implementations

Three network security protocols were executed on top of every simulated day in Helsinki: One utilizing Symmetric cryptography, one running Asymmetric cryptography, and a final one using Group Broadcast encryption, as defined by [11]. The implementations are defined as follows:

Symmetric Cryptography: Every reachable node is added to a growing group that shares a common key. Whenever a vehicle is added or removed from the group, a new key is generated and shared between all members.

Asymmetric Cryptography: Every reachable node is managed by a source node, keeping all the neighbor's public key. Every message sent is ciphered once for every connected neighbor.

Group Broadcast: Every reachable node is added to the source node's group view. Messages sent from this particular node are readable by every node whose public key was used in the creation of the group view key.

During this work's practical part, simulations are run considering the use of a single symmetric key for a forever merging and growing group, in order to decrease the key cardinality to a single key, used to both cipher and decipher. While this will create many different security vulnerabilities and further management complications, this experiment is interesting further on, in order to prove the Group Broadcast efficiency. The vulnerabilities created by sharing a single key for several vehicles are ignored and not a part of this project's scope.

The implementation used in the symmetric cryptography is easily identifiable as the least secure since every node gets the common key. This was used in order to properly demonstrate the performance gain of using Group Broadcast Encryption, as it's only necessary to encrypt a message once, and it will be readable by the whole group, the same amount of work required for using a single symmetric key. Better security could be achieved using symmetric cryptography, if every pair of nodes had a single key. This latter solution would be greatly outperformed by using group broadcast encryption, as fewer messages would be necessary to share the same information in the network.

The main variable observed in these simulations is the number of messages. Counters are used in order to identify how many messages are required to add a vehicle to a group, how many to remove a vehicle from a group, and, mainly, how many messages are required in order to send a secure message within this group. Table 3 describes how the three types of algorithms chosen interact with the number of messages.

5. Results and Discussion

In this section, simulation data is presented and discussed. As seen in Table 3, the group broadcast solution should be bound to have the same cardinality and number of messages as one of the other two algorithms, in each of the three types of messages that are required. For connection messages, the amount is equal to the number of messages required to set-up the asymmetric encryption. This happens because, by definition, group broadcast encryption is an asymmetric algorithm, and sharing the public keys is required. The symmetric solution is by far the simplest solution to set-up because every vehicle in the simulation is sharing a common key. While this is not exactly useful for comparing connection messages, it will prove very important for the comparison of secure group messages. For sending secure messages in the group, the group broadcast encryption solution works just like the symmetric solution, by sending a single message to the whole group view, while the asymmetric solution is burdened to send one message for every neighbor. Finally, when revoking a vehicle from the group, the source node only notifies its neighbors that it is doing so. In conclusion, the symmetric and group broadcast solution only send a single message, while the asymmetric solution is burdened again to send one message for every other node.

Figure 1 presents the number of messages used by each algorithm in order to set-up the encryption. In this image, the number of messages used to set-up the asymmetric encryption and the group broadcast encryption were exactly the same, as discussed on Table 3.

Figure 2 displays the amount of exchanged secure messages for each algorithm in every simulation. In this image, symmetric encryption and group broadcast encryption share the exact same number of messages (one for each neighbor every update) and are about twenty times lower than the number of messages required by the asymmetric encryption.

Figure 3 shows the number of revocation messages sent. For the simulations,

Table 3. Number of messages per algorithm.

	Symmetric	Asymmetric	Group Broadcast
Connection	Three-way handshake between two nodes, plus message for sharing pseudonyms	Diffie-Helman between each pair of nodes.	Three-way handshake between two nodes, plus message for sharing pseudonyms
Messages	Single encrypted message for all neighbors	One encrypted message for each neighbor	Single encrypted message for all neighbors.
Revocation	Single encrypted message for all neighbors	One encrypted message for each neighbor	Single encrypted message for all neighbors.

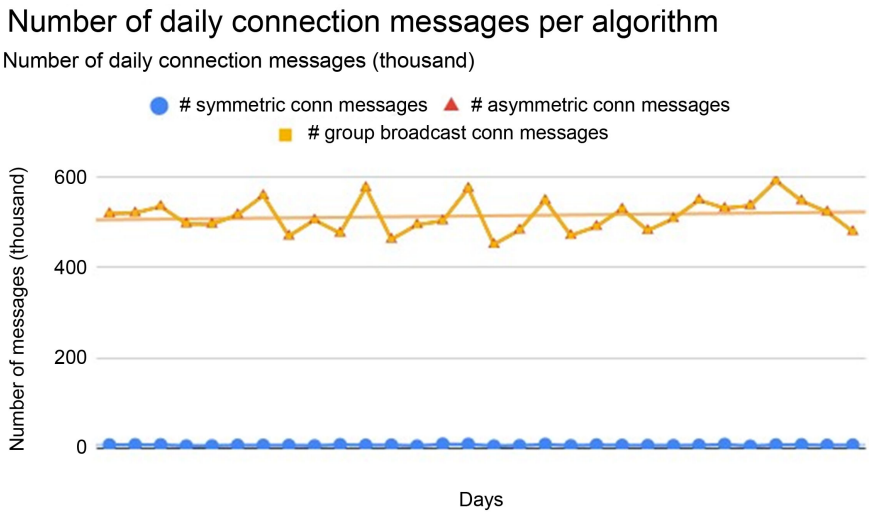

Figure 1. Number of connection messages per algorithm per day.

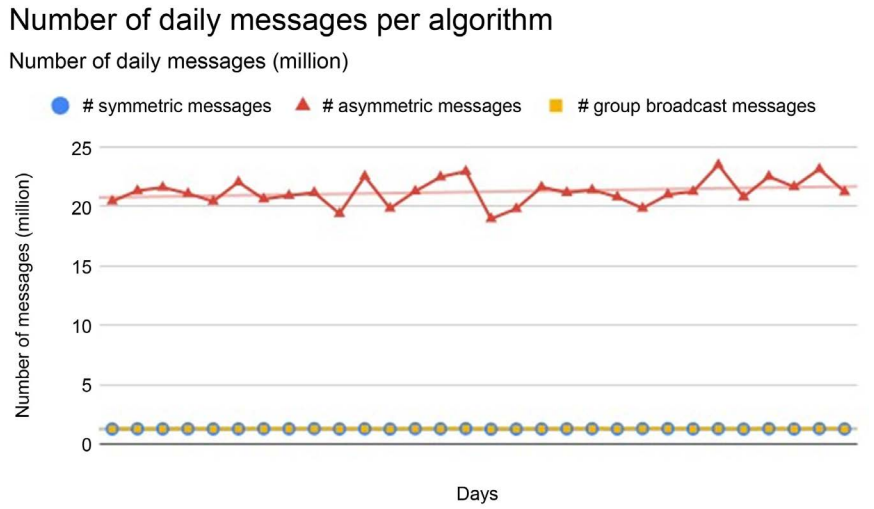

Figure 2. Number of secure messages per algorithm per day.

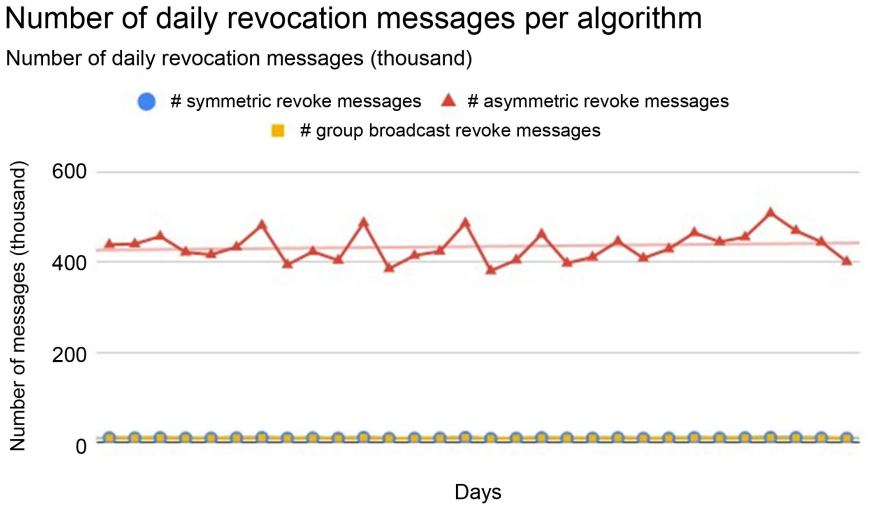

Figure 3. Number of revocation messages per algorithm per day.

the only type of disconnection message sent was for out-of-range disconnections. Once again, the number of messages sent in the network was equal between symmetric and group broadcast encryption, with the number of messages required for the asymmetric encryption being about four hundred times bigger.

While group broadcast encryption does not necessarily decrease CPU load for encrypting messages, using it in a VANET environment can significantly decrease the amount of work required to send secure messages in the network, because the source node works just as if it had a single key for every other neighboring node. This ensures that the scaling of the number of messages is linear to the number of neighboring nodes instead of exponential, guaranteeing that the network is not flooded with repeated messages that were encrypted with a different key.

6. Conclusion

In this paper, a solution for enhancing VANET security performance was presented, along with a performance benchmark which indicates that Group Broadcast algorithms are effective in decreasing the ammount of exchanged messages between vehicles. This is important to create a simpler and faster network, which requires fewer mechanisms for controlling the general state of the network, while also decreasing the resources consumed by each node, such as memory and CPU-time. While the solution is not as lean as simply using symmetric cryptography, it is far more economical than using a fully asymmetric system, keeping the security principles of asymmetric cryptography. As the main focus of the conducted research was to evaluate Group broadcast encryption as a VANET security application in a simulated environment, future developments are able to build and test similar solutions in a physical environment.

Acknowledgements

This research was partially funded by CAPES—Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Karagiannis, G., Altintas, O. and Ekici, E. (2011) Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions. *IEEE Communications Surveys & Tutorials*, **13**, 584-616. https://doi.org/10.1109/SURV.2011.061411.00019

[2] Hartenstein, H. and Laberteaux, K.P. (2008) A Tutorial Survey on Vehicular Ad Hoc Networks. *IEEE Communications Magazine*, **46**, 164-171. https://doi.org/10.1109/MCOM.2008.4539481

[3] Mishra, R., Singh, A. and Kumar, R. (2016) VANET Security: Issues, Challenges and

Solutions. *International Conference on Electrical, Electronics, and Optimization Techniques* (*ICEEOT*), Chennai, 3-5 March 2016. https://doi.org/10.1109/ICEEOT.2016.7754846

[4] Bariah, L., Shehada, D. and Yeun, C.Y. (2015) Recent Advances in VANET Security: A Survey. *IEEE* 82*nd Vehicular Technology Conference* (*VTC*2015-*Fall*), Boston, 6-9 September 2015. https://doi.org/10.1109/VTCFall.2015.7391111

[5] Deeksha, A.K., Bansal, M., *et al.* (2017) A Review on VANET Security Attacks and Their Countermeasure. 4*th IEEE International Conference on Signal Processing, Computing and Control* (*ISPCC* 2*k*17), Solan, 21-23 September 2017. https://doi.org/10.1109/ISPCC.2017.8269745

[6] Hao, Y., Cheng, Y. and Zhou, C. (2011) A Distributed Key Management Framework with Cooperative Message Authentication in VANETs. *IEEE Journal on Selected Areas in Communications*, **3**, 616-629. https://doi.org/10.1109/JSAC.2011.110311

[7] Sun, J.Y., Zhang, Y.C. and Fang, Y.G. (2010) An Identity Based Security System for User Privacy in Vehicular Ad Hoc Networks. *IEEE Transactions on Parallel and Distributed Systems*, **21**, 1227-1239. https://doi.org/10.1109/TPDS.2010.14

[8] Tzeng, S.-F., Horng, S.-J., Li, T.R., Wang, X., Huang, P.-H. and Khan, M.K. (2017) Enhancing Security and Privacy for Identity-Based Batch Verification Scheme in VANETs. *IEEE Transactions on Vehicular Technology*, **66**, 3235-3248. https://doi.org/10.1109/TVT.2015.2406877

[9] Hasrouny, H., Bassil, C., Samhat, A.E. and Laouiti, A. (2015) Group-Based Authentication in V2V Communications. *Fifth International Conference on Digital Information and Communication Technology and Its Applications* (*DICTAP*), Beirut, 29 April-1 May 2015. https://doi.org/10.1109/DICTAP.2015.7113193

[10] Jin, H.Y. and Papadimitratos, P. (2015) Scaling VANET Security through Cooperative Message Verification. *IEEE Vehicular Networking Conference* (*VNC*), Kyoto, 16-18 December 2015. https://doi.org/10.1109/VNC.2015.7385588

[11] da Silva, E. and Albini, L.C.P. (2013) Towards a Fully Self-Organized Identity-Based Key Management System for MANETs. *International Conference on Wireless and Mobile Computing, Networking and Communications* (*WiMob*), Lyon, 7-9 October 2013. https://doi.org/10.1109/WiMOB.2013.6673435

[12] Greca, R.D.M. (2018) TruMan: Trust Management for Vehicular Networks. Pós-Graduação em Informática-Universidade Federal do Paraná.

[13] Keranen, A. (2008) Opportunistic Network Environment Simulator. Helsinki University of Technology, Helsinki.

[14] Ekman, F., Keranen, A., Karvo, J. and Ott, J. (2008) Working Day Movement Model, Mobility Models '08. *Proceedings of the* 1*st ACM SIGMOBILE Workshop on Mobility Models*, **1**, 33-40. https://doi.org/10.1145/1374688.1374695