Implementation of SHA-1 and ECDSA for Vehicular Ad-Hoc Network using NS-3

Sinan Nacy
Rochester Institute of Technology
Sector 906, Street 48, House 16
Baghdad, Iraq
+964 790-138-7055
ssn2252@rit.edu

Tae Oh
Rochester Institute of Technology
152 Lomb Memorial Drive
Rochester, NY 14623
585-475-7642
thoics@rit.edu

Jim Leone
Rochester Institute of Technology
152 Lomb Memorial Drive
Rochester, NY 14623
585-475-6451
Jim.Leone@rit.edu

ABSTRACT
VANET, the Vehicular Ad-Hoc Network, treats cars as nodes in a mobile network. Not surprisingly, VANET must be very secured since one of the network characteristics allows the network to be open to public. The digital signature used in VANET is the standard, ECDSA, or Elliptic Curve Digital Signature Algorithms. ECDSA provides network security by employing a digital signature for messages being transmitted over the network. An ECDSA developed in C++ is described here. VANET messages were sent using the NS-3 network simulator. Two scenarios were created to test the code and the differences before and after implementing the digital signature.

Categories and Subject Descriptors
C.2.0 [Computer-Communication Networks]: General—Security and protection; C.2.1 [Computer-Communication Networks]: Network Architecture and Design—Network communications, Wireless communication.

Keywords
VANET, Digital signature, Security, ECDSA

1. INTRODUCTION
Vehicular Ad Hoc Network (VANET) is a emerging technology that creates a mobile network by using moving vehicles as nodes. In VANET, messages are transferred between vehicles as well as roadside units. As with any other emerging technology, standards and protocols for VANET have to be created and tested before being adopted. The implementation of such a technology in a real world scenario will increase the safety of drivers and help police, fire, EMT and other emergency vehicles in life-critical situations. As with any network, VANET security is considered the number one issue when discussing the challenges associated with such a new technology. Because the VANET network is open to public, eavesdropping on data will be a major concern as data can be altered while being transferred between the nodes. Any digital security implementation must be sufficiently efficient to allow anticipated (message) traffic to occur without degradation to the network's real-time capabilities.

RIIT'13, October 10–12, 2013, Orlando, Florida, USA.

http://dx.doi.org/10.1145/2512209.2512221

The NS-3 network simulator provides a convenient test-bed with which to test an algorithm's efficiency. The process involved the use of a Secure Hash Algorithm (SHA-1) to create a unique key for each message. The ECDSA then generates a two key pair signature using the SHA-1 key that incorporates ECDSA domain parameters. The signature is sent to the destination along with the message. The receiver verifies the signature using SHA-1 key for the message, ECDSA domain parameters and the two key pair. If the output of the verification process matches one of the key pairs, then the signature is verified; otherwise, the message was corrupted on transmission.

As for the organization of the paper, related works is discussed in Section 2 and the problem statement was covered in Section 3. The implementation is explained in Section 4 and the results are discussed in Section 5. Finally, the paper is concluded with conclusion and summary in Section 6.

2. RELATED WORK
In this section, several previous and enhancement of SHA-1 and ECDSA are described. The sub-sections are divided into different categories of security development and the sub-section starts with utilizing NS-3 to develop a layer 2 technology that is related to VANET implementation.

2.1 Applications of NS-2/NS-3 Network Simulator
Arbabi and Weigle developed a simulator in NS-3 to model customized onboard and roadside units.[1] Chen, and et al. used NS-2 network simulator tool in other layer 2 areas to model Dedicated Short Range Communication (DSRC) technology using IEEE 802.11 protocols.[2] However, the more accurate DSRC technology is an improvement to the IEEE 802.11 PHY and MAC modules.

2.2 Signature Security in VANET
Biswas et al. proposed a scheme for securing broadcast messages from RSU to OBU in VANET that uses a mechanism called legacy warrant, which is a modification to the original proxy signature.[3] Daihoon, Jaeduck and Souhwan proposed an identification and key exchange scheme that is based on group signature.[4] Aslam and Zou proposed a security architecture that is based on revised Blind signature scheme, which provides one-way link-ability.[5]

2.3 Authentication Security in VANET
Giorgio, Panos, Jean-Pierre and Antonio proposed a new scheme mechanism to manage credentials in VANET that depends on the

pseudonymous authentication concept, which the authors of the article term Baseline Pseudonym (BP).[6] Fan, Hsu, and Tseng proposed an efficient pseudonym PKI mechanism that is based on bilinear mapping which improves the certificate revocation and certificate tracing, implementation cost, management cost, and performance of the message authentication.[7] Studer, Shi, Fan and Perrig proposed a new VANET key management scheme that is based on Temporary Anonymous Certified Keys (TACKs). The scheme provides strong security, and privacy for key management in VANET.[8]

Manvi, Kakkasageri, and Adiga proposed an Elliptic Curve Digital Signature Algorithm (ECDSA) based message authentication for VANET.[9] The operation process in the scheme takes the following steps:

1. The source vehicle will generate a pair of keys: a private key, and a public key.
2. The public key is available to all the vehicles in the network.
3. The source vehicle creates a hash to the message using a hash function.
4. The hash value is encrypted using the vehicle's private key, and then send to its destination.
5. The destination vehicle receives the message, decrypts it using the sender's vehicle public key, and gets the hash value.
6. The destination vehicle generates a hash value to the message using the same procedure that the sender's vehicle used to generate the hash, and then it compares the two hashes. If the two hashes are identical, then that means the message was not modified while transmission. Otherwise, the message was altered. Using this scheme, strong authentication and message integrity are maintained.

Hsin-Te et al. developed a comprehensive message authentication scheme that makes use of the Diffie-Hellman key establishment protocol, and the Hash Message Authentication Code (HMAC).[10] Their protocol enables the authentication of the message in intra- and inter-roadside units range, and the handoff inside different roadside units. Vighnesh and et al. has proposed a novel sender authentication scheme to enhance the security of VANET. The scheme uses hash chaining to authenticate vehicles by hashing an input a number of times.[11] Behera, Mishra, Navak and Jena proposed a security protocol that uses a cryptographic technique called Elliptic Curve Digital Signature Algorithm (ECDSA) to provide message authentication and privacy for VANET users.[12]

Biswas, Misic, and Misic proposed a safety message authentication scheme for VANET that uses ID-based proxy signature and verification mechanism. The scheme provides strong security and high performance.[13] Tat, Lucas, Siu, and Victor proposed a Multiple Level Authentication Scheme (MLAS) that has six basic modules. The scheme makes use of tamper-proof devices.[14]

2.4 Other Areas of VANET Security

Yan, Choudhary, Weigle and Olariu proposed a novel position security system to enhance the security in VANET that makes use of on-board radar to detect neighbor vehicles and confirms with them their coordinates. It provides both global and local position security.[15] Fonseca et al. proposed a driver protection framework consisting of four different features, 1) extended location service, 2) cross-layer addressing, 3) link layer callbacks and 4) pseudonymity-enhanced packet forwarding schemes.[16]

Yi and Moayeri developed a framework for security and application oriented network design that has two basic schemes: application aware control scheme, and unified routing scheme.[17] Ching-Hung, Yueh-Min, Tzone-I and Hsiao-Hwa created a decentralized IVC (Inter Vehicle Communication) scheme without the need of infrastructure called a Dynamic Establishment of Secure Communication in VANET (DESCV) that provides random wireless connections between vehicles.[18]

Gosman, Dobre and Cristea have proposed a security protocol designed for VANET environment that guarantees the integrity of the data in the messages against different attackers.[19]

Samara, Al-Salihy and Sures proposed a simple, scalable, and flexible design for VANET certifications that includes new methods for efficient certificate management.[20]

Wasef, Rongxing, Xiaodong and Xuemin developed new complementary security mechanism to prevent Denial of Service attack (DoS).[21]

The IEEE standard 1609.2 is used to define secure message format and identifies security mechanism and algorithms for use in a Wireless Access in Vehicular Environment (WAVE). Rabadi proposed the use of implicit certificates in the WAVE standard because the size of implicit certificate is shorter than explicit certificate.[22]

Trusted Platform Module (TPM) is a security hardware component. It can handle software attacks on VANET. Moreover, it maintains the data integrity in the network. Based on this TPM, Sumra, Hasbullah, Ahmed and Ab Manan proposed a new 'chain of trust' model that is built within vehicles to manage a variety of attacks and preserves the message integrity in the vehicular network.[23]

Mershad and Artail developed a new system that makes use of the Roadside Units (RSUs) that have Internet connectivity and provides information to VANET users. The system is called secuRe and Efficient dAta aCquisiTion in VANETs (REACT). The system provides a novel privacy and security mechanism.[24]

Rawat, Bista, Yan and Weigle proposed a new algorithm to secure the communication between vehicles with the help of trust measured for given period using probabilistic approach.[25]

3. PROBLEM STATEMENT

VANET is an emerging technology and is still under development. The primary security issue in VANET is checking the integrity of messages exchanged among vehicles. In the paper, implementation of security algorithms in modeling and simulation for VANET are discussed. Our modeling and simulation could be used in many VANET research activities and streamlines the security protocol development for VANET. Our implementation provides a new security approach to secure the messages using ECDSA. The following steps described the functionality of our implementation:

1) The sender node will create a hash value for its message using SHA-1 algorithm.

2) The node will then create a digital signature for the message using ECDSA.

3) The receiver node will calculate the hash value for the message and verify the digital signature.

4) If the digital signature is verified, the receiver node will assume that the message came from its original sender.

4. OUR IMPLEMENTATION

The main tool used to implement the project is NS-3 in a Linux environment. We have developed and implemented a C++ code that provides ECDSA digital signature inside NS-3. We also added a SHA-1 code to our work since ECDSA requires a hashing function. We have used a simulation in NS-3 that provides VANET functionality. A Controller class controls the scenarios in the VANET simulation.

The ECDSA signature scheme is used by two entities: a signer X, and a verifier Y. The signer X will sign the message M and send it to the verifier Y. The verifier Y will receive the message M and verify it. In fact, any entity can verify the signature if it has X's public key. Sometimes third party can be involved to verify the signature of the message.

Entity X should use the key deployment procedure to establish a key pair. Entity Y should be able to obtain X's public key. And X will use the key pair in order to control the signing operation while Y uses the public key required to control the verification step. When X wants to send a message M, it should sign the message using its key pairs and generate a signature S. Entity X will create a message using M and S, and send it to Y. When Y receives the message, it applies the verifying operation using X's public key in order to verify the message authenticity. If the output of the verifying operation is valid, then Y will know that the message M is authentic. In other words, it came from the sender X.

4.1 ECDSA Domain Parameters

ECDSA algorithm requires that the private, and public keys used for digital signature generation and verification be generated with respect to a set of domain parameters. The domain parameters can be the same to a group of users and may be public. Domain parameters can remain fixed for an extended time period. [26] [27] [28] The ECDSA domain parameters are:

- q or p, the size of the underlying field,
- a, elliptic curve parameter that is used to define the equation of the curve,
- b, elliptic curve parameter that is used to define the equation of the curve,
- $G = (Gx, Gy)$, a point on the elliptic curve, and is called a base point,
- n, the order of the base point G,
- h, the order of the elliptic curve divided by the order n, and is called the cofactor.

4.2 ECDSA Private Key/Public Key

ECDSA key pair consists of private key d, and public key Q. Each key pair is associated with a specific set of domain parameters. The private key d, the public key Q, and the domain parameters are mathematically related to one another via the relation $Q = dG$, where dG is the sum of d copies of the base point G. It is also know as elliptic curve scalar multiply of G by d. The sum operation is done using elliptic curve arithmetic. The private key d can be used for a limited period of time (i.e. the cryptoperiod). On the other hand, the public key Q can be used as long as the digital signature that is generated using the associated private key is still in use because the digital signature needs to be verified. [26] [27] [28]

ECDSA private key and public key are only used in the generation and the verification of the ECDSA digital signature. They should not be used for other purposes (e.g. key establishment). [26] [27] [28]

4.3 ECDSA Key Generation

In order for an entity to generate the key pair, it must make sure that the domain parameters are valid. Each key pair is associated with a specific set of domain parameters [26] [27] [28]. Generating the key pair is done as follows:

1. Select a random integer d in the interval $[1, n-1]$.
2. Compute $Q = dG$.

The results are d and Q, where d is the private key, and Q (Qx, Qy) is the public key.

4.4 ECDSA Signature Generation

An entity can sign a message m using the key pair and the domain parameters. The output from the signing operation is a signature, and is represented by (r, s) [26] [27] [28]. An entity does the following to sign a message:

1. Select an integer k, where $1 \le k \le n-1$.
2. Compute $kQ = (x1, y1)$.
3. Compute $r = x1 \pmod n$. If $r = 0$ then go to step 1.
4. Compute $k^{-1} \pmod n$. Note: $k^{-1} \pmod n$ is computed using the inverse theory in Appendix A.
5. Compute SHA-1(m), and convert this string to an integer H(m).
6. Compute $s = k^{-1} (H(m) + dr) \pmod n$. If $s = 0$, then go to step 1.

The signature for the message is (r, s).

4.5 ECDSA Signature Verification

To verify a signature (r, s) on a message m, the receiver obtains a copy of the sender's domain parameters, and its public key Q [26] [27] [28]. The receiver does the following:

1. Verify that r and s are integers, and in the interval $[1, n-1]$.
2. Compute SHA-1(m), and convert this string to an integer H(m).
3. Compute $w = s^{-1} \pmod n$. Note: $s^{-1} \pmod n$ is computed using the inverse theory in Appendix A.
4. Compute $u1 = H(m)w \pmod n$, and $u2 = rw \pmod n$.
5. Compute $X = (x1, y1) = u1G + u2Q$.
6. If $X = 0$, reject the signature. Otherwise, compute $v = x1 \pmod n$.
7. Accept the signature if $v = r$.

The signature to the message m is verified if $v = r$.

5. SIMULATION RESULTS

The code was implemented inside the Controller class. Two scenarios were created and ran the simulation twice for each scenario. The first run is before implementing our code, and the second run is after implementing our code. Then we compared between the two runs for each scenario.

5.1 Scenario 1

In this scenario, we have a bi-directional highway with 1000 meters of roadway. There are two lanes on each direction. The width of the lane is 5 meters. The median width is 5 meters. The highway has 20% of trucks, and 80% of sedan vehicles. A broken car has stopped in the middle of the highway, at the location (x = 500 meters, direction = 1, lane = 0).

The broken vehicle broadcasts warning/safety messages asking for help, and revealing its location every 5 seconds. A police car has been created in the scenario, and will start moving from the beginning of the highway. The police car is faster than any regular car, and has a higher wireless transmission range. It listens to broadcast messages and sends a unicast for each received message. The police car will start to decelerate once it gets closer to the broken car, and will eventually stop nearby.

Figure 1 shows the output result from the gnuplot after 2 minutes and 20 seconds from running the simulation. The police car has reached the broken car after 50 seconds, and stopped in the second lane of the highway. As you can see in the Figure, the police car has made congestion behind it.

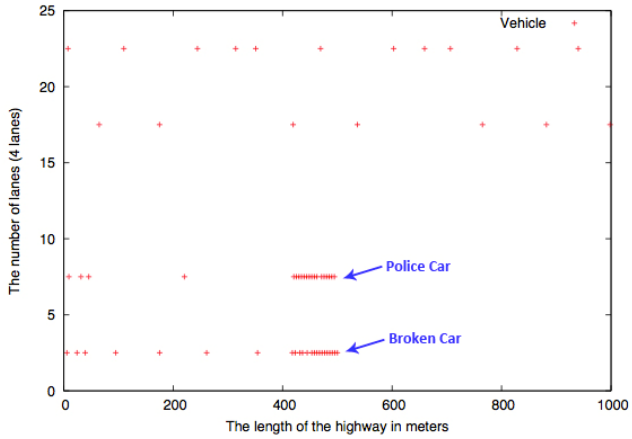

Figure 1. Scenario 1, before implementing ECDSA on broadcast messages.

In Figure 2, we have used the same scenario, with the exception of implementing the ECDSA code on the broadcast messages.

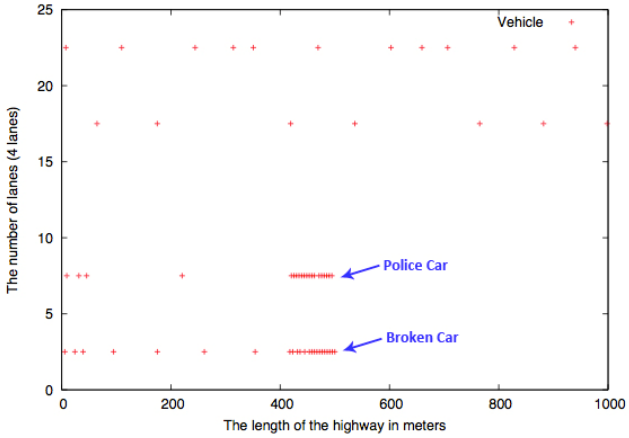

Figure 2. Scenario 1, after implementing ECDSA on broadcast messages.

Figure 3 represents the number of packets received, dropped, and the police acknowledgment for receiving a packet. The number of packets is the same before and after implementing the ECDSA code,.

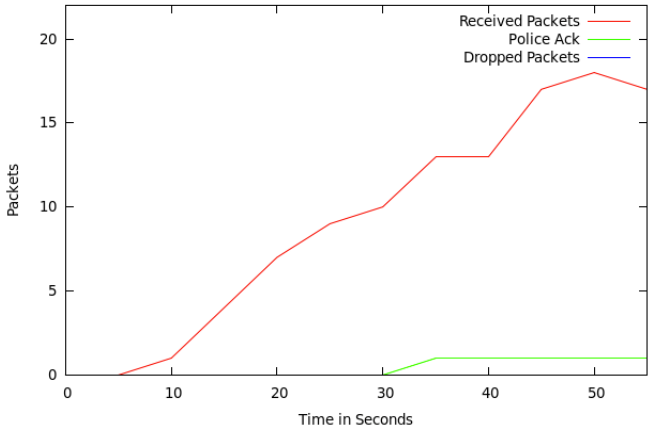

Figure 3. Scenario 1, the number of packets on the network.

Comparing between the results from Figure 1, and Figure 2, we found that there are no major differences between the two output results. There could be a minor delay time in mille seconds, but there is no problem. As for Figure 3, the number of packets is the same before and after implementing the ECDSA code. For this scenario, we conclude that, implementing the ECDSA does not affect the work of our simulation environment; instead, it makes our implementation more secure, robust, and reliable.

5.2 Scenario 2

In this scenario, we have a bi-directional highway with 1400 meters of roadway. There are three lanes on each direction. The width of the lane is 5 meters. The median width is 5 meters. The highway has 20% of trucks, and 80% of sedan vehicles. A broken car has stopped at the location (x = 850 meters, direction = 1, lane = 0). The broken vehicle broadcasts warning/safety messages asking for help, and revealing its location every 5 seconds. A police car has been created in the scenario, and will start moving from the beginning of the highway. The police car is faster than any regular car, and has a higher wireless transmission range. It listens to broadcast messages and sends a unicast for each received message. The police car will start to decelerate once it gets closer to the broken car, and will eventually stop nearby.

In Figure 4, we can see the output result from the gnuplot after 2 minutes and 35 seconds from the time of the simulation. The police car has reached the broken car at location x=850 after 1 minute, and stopped in the second lane of the highway. As you can see in the Figure 4, the police car is starting to make congestion on the highway. The third lane on the highway is open, but because two lanes are closed the traffic is being directed to the third lane. As a result, congestion will occur on the highway.

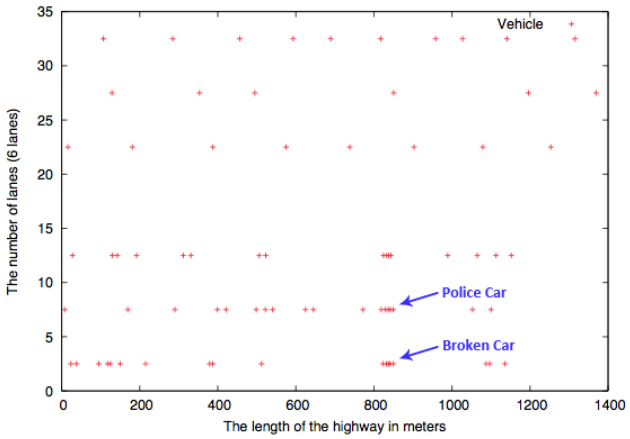

Figure 4. Scenario 2, before implementing ECDSA on broadcast messages.

In Figure 5, we have used the same scenario, with the exception of implementing the ECDSA code on the broadcast messages.

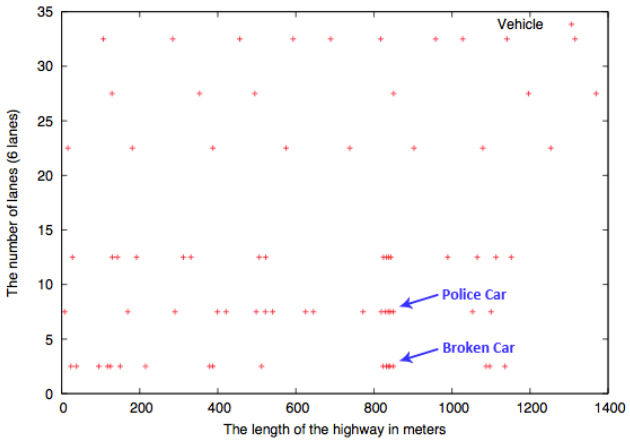

Figure 5. Scenario 2, after implementing ECDSA on broadcast messages.

Figure 6 represents the number of packets received, dropped, and the police acknowledgment for receiving a packet. The number of packets is the same before and after implementing the ECDSA code, that's why we have only added one figure.

Comparing between the two results from Figure 4, and Figure 5, there are no major differences between the two output results. Just like the first scenario, there could be a minor delay time in mille seconds. As for Figure 6, the number of packets is the same before and after implementing the ECDSA code. As a result, we conclude that, implementing the ECDSA does not affect the work of our simulation environment, and does not cause any delay; instead, it makes our implementation more secure, robust, and reliable.

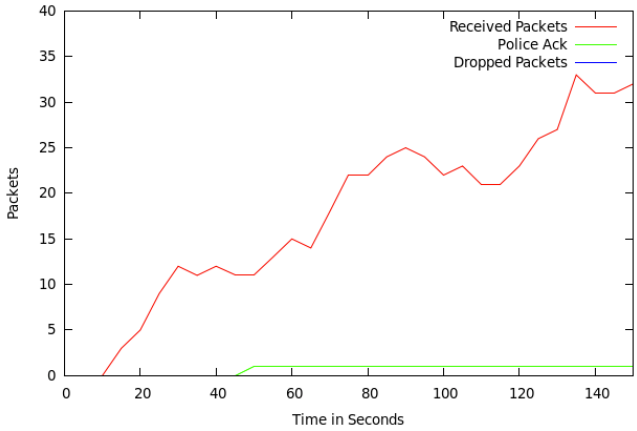

Figure 6. Scenario 2, the number of packets on the network.

6. CONCLUSION

As a conclusion, the ECDSA code was implemented successfully inside NS-3. Comparing between the two results (before, and after implementing the ECDSA code) for each scenario, we can see that there is no difference, which is somehow suspicious. In addition to that, the final results don't show much information, and this is due to a lack in the VANET simulation environment. At the same time, implementing the ECDSA code on messages didn't show major delay in time inside the environment. Instead, we can tell that it provides more security to the messages by creating a digital signature for each message on the network.

7. REFERENCES

[1] Arbabi, H.; Weigle, M.C. 2010. Highway mobility and vehicular ad-hoc networks in ns-3. *Simulation Conference (WSC), Proceedings of the 2010 Winter*, vol., no., pp.2991-3003, 5-8 Dec. 2010.

[2] Q. Chen, D. Jiang, V. Taliwal, and L. Delgrossi. 2006. IEEE 802.11 Based Vehicular Communication Simulation Design for NS-2. *In Proceedings of the International Workshop on Vehicular Ad Hoc Networks (VANET)*, Los Angeles, CA, USA, Sept. 2006.

[3] Biswas, S.; Mišić, J. 2010. Proxy signature-based RSU message broadcasting in VANETs. *Communications (QBSC), 2010 25th Biennial Symposium on*, vol., no., pp.5-9, 12-14 May 2010.

[4] Daihoon Kim; Jaeduck Choi; Souhwan Jung. 2010. Mutual Identification and Key Exchange Scheme in Secure VANETs Based on Group Signature. *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, vol., no., pp.1-2, 9-12 Jan. 2010.

[5] Aslam, B.; Zou, C.C. 2011. One-way-linkable blind signature security architecture for VANET. *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, vol., no., pp.745-750, 9-12 Jan. 2011.

[6] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. 2007. Efficient and robust pseudonymous authentication in VANET. *In VANET '07*, pages 19–28, New York, NY, USA, September 2007. ACM.

[7] C. I. Fan, R. H. Hsu, and C. H. Tseng. 2008. Pairing-based message authentication scheme with privacy protection in

vehicular ad hoc network. *In Proceedings of the International Conference on Mobile Technology, Applications and Systems*, September 2008.

[8] Studer, A.; Shi, E.; Fan Bai; Perrig, A. 2009. TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs. *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, vol., no., pp.1-9, 22-26 June 2009.

[9] Manvi, S.S.; Kakkasageri, M.S.; Adiga, D.G. 2009. Message Authentication in Vehicular Ad Hoc Networks: ECDSA Based Approach. *Future Computer and Communication, 2009. ICFCC 2009. International Conference on*, vol., no., pp.16-20, 3-5 April 2009.

[10] Hsin-Te Wu; Wei-Shuo Li; Tung-Shih Su; Wen-Shyong Hsieh. 2010. A Novel RSU-Based Message Authentication Scheme for VANET. *Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on*, vol., no., pp.111-116, 22-27 Aug. 2010.

[11] Vighnesh, N.V.; Kavita, N.; Urs, S.R.; Sampalli, S. 2011. A novel sender authentication scheme based on hash chain for Vehicular Ad-Hoc Networks. *Wireless Technology and Applications (ISWTA), 2011 IEEE Symposium on*, vol., no., pp.96-101, 25-28 Sept. 2011.

[12] Behera, S.; Mishra, B.; Nayak, P.; Jena, D. 2011. A secure and efficient message authentication protocol for vehicular Ad hoc Networks with privacy preservation (MAPWPP). *Internet Multimedia Systems Architecture and Application (IMSAA), 2011 IEEE 5th International Conference on* , vol., no., pp.1-6, 12-13 Dec. 2011.

[13] Biswas, S.; Misic, J.; Misic, V. 2011. ID-based Safety Message Authentication for Security and Trust in Vehicular Networks. *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on* , vol., no., pp.323-331, 20-24 June 2011.

[14] Tat Wing Chim, Lucas Chi Kwong Hui, Siu-Ming Yiu, Victor O. K. Li. 2011. *MLAS*: multiple level authentication scheme for VANETs. *Published in: Proceeding ASIACCS '11 Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ASIACCS: pp.471-475, 2011.

[15] G. Yan, G. Choudhary, M. Weigle, and S. Olariu. 2007. Providing vanet security through active position detection (poster). *In Proceedings of ACM VANET 07*, Sept. 2007.

[16] Fonseca, E.; Festag, A.; Baldessari, R.; Aguiar, R.L. 2007. Support of Anonymity in VANETs - Putting Pseudonymity into Practice. *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, vol., no., pp.3400-3405, 11-15 March 2007.

[17] Yi Qian; Moayeri, N. 2008. Design of Secure and Application-Oriented VANETs. *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, vol., no., pp.2794-2799, 11-14 May 2008.

[18] Ching-Hung Yeh, Yueh-Min Huang, Tzone-I Wang, Hsiao-Hwa Chen. 2009. DESCV - A Secure Wireless Communication Scheme for Vehicle ad hoc Networking. MONET. *Published in Journal Mobile Networks and Applications*, Volume 14, Issue 5, pp.611-624, October 2009.

[19] Gosman, C.; Dobre, C.; Cristea, V. 2010. A Security Protocol for Vehicular Distributed Systems. *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2010 12th International Symposium on*, vol., no., pp.321-327, 23-26 Sept. 2010.

[20] Samara, G.; Al-Salihy, W.A.H.; Sures, R. 2010. Efficient certificate management in VANET. *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, vol.3, no., pp.V3-750-V3-754, 21-24 May 2010.

[21] Wasef, A.; Rongxing Lu; Xiaodong Lin; Xuemin Shen. 2010. Complementing public key infrastructure to secure vehicular ad hoc networks [Security and Privacy in Emerging Wireless Networks]. *Wireless Communications, IEEE* , vol.17, no.5, pp.22-28, October 2010.

[22] Rabadi, N.M. 2010. Implicit certificates support in IEEE 1609 security services for Wireless Access in Vehicular Environment (WAVE). *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, vol., no., pp.531-537, 8-12 Nov. 2010.

[23] Sumra, I.A.; Hasbullah, H.; Ahmad, I.; bin Ab Manan, J.-L. 2011. Forming vehicular web of trust in VANET. *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International*, vol., no., pp.1-6, 24-26 April 2011.

[24] Mershad, K.; Artail, H. 2011. REACT: Secure and efficient data acquisition in VANETs. *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, vol., no., pp.149-156, 10-12 Oct. 2011.

[25] Rawat, D.B.; Bista, B.B.; Yan, G.; Weigle, M.C. 2011. Securing Vehicular Ad-hoc Networks Against Malicious Drivers: A Probabilistic Approach. *Complex, Intelligent and Software Intensive Systems (CISIS), 2011 International Conference on*, vol., no., pp.146-151, June 30 2011-July 2 2011.

[26] ANSI X9.62-2005, 2005. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). American National Standards Institute, November 2005.

[27] FIPS 186-3, 2009. Digital Signature Standard (DSS). Federal Information Standards Processing Publication 186-3, National Institute of Standards and Technology, June 2009.

[28] SEC1 Standards for Efficient Cryptography Group, SEC 1: Elliptic Curve Cryptography, Version 2.0, 2009.