Impact of Message Authentication on Braking Distance in Vehicular Networks

Jonathan Petit, Zoubir Mammeri

IRIT - Paul Sabatier University
Toulouse, France
{Jonathan.Petit@irit.fr, Zoubir.Mammeri@irit.fr}

Abstract: Transport safety applications aim at avoiding vehicular accidents by using secure broadcast vehicle-to-vehicle (V2V) communications. However, any security mechanism used for authenticating broadcast V2V messages comes with overhead in terms of computation and communications. The IEEE1609.2 standard for vehicular ad hoc networks is based on the ECDSA algorithm for supporting the authentication mechanism. This paper provides an assessment of the processing and communication overhead of ECDSA. We analyze the impact of this mechanism on VANET performance. Then we focus on the impact of authentication communication on the braking distance.

Keywords: VANET, ECDSA, authentication overhead, braking distance, message authentication

1. Introduction

Due to the huge life losses and the economic impacts resulting from vehicular collisions, many governments, automotive companies, and industry consortia have made the reduction of vehicular fatalities a top priority [1]. On average, vehicular collisions cause 102 deaths and 7900 injuries daily in the United States, leaving an economic impact of $230 billion [2]. The damage is similarly devastating in the European Union, where there are more than 110 deaths and 4600 injuries daily, costing €160 billion annually [3].

A major evolution for the automotive industry is the context awareness, meaning that a vehicle is aware of its neighborhood. Modern cars now include a set of processors connected to a central computing platform that provides many wired and wireless interfaces. Smart vehicles are those vehicles that are equipped with On-Board Unit (OBU), which has recording, processing, positioning, and location capabilities and that supports wireless security protocols. Roads can be made smart, too. Road-Side Units (RSU) installed along a road can inform passing vehicles about the road traffic conditions. With more smart cars and roads [4], we can expect many changes. Particularly, it is expected that the number and severity of accidents should decrease.

Automotive safety applications aim to assist drivers in avoiding vehicular accidents, by providing advisories and early warnings to drivers, using broadcast vehicle-to-vehicle (V2V) communications. Vehicles typically communicate as per the Dedicated Short Range Communication standard (DSRC) [5], and broadcast messages in response to certain notified events (emergency message) or periodically (beacon message) [6]. V2V communications enable an entire space of applications, in addition to automotive safety, as infotainment and commercial.

Since drivers of vehicles participating in V2V communications are expected to act on messages received from other participants, it is clearly necessary that these messages be transmitted in a secure fashion. In order to secure vehicular communications, Wireless Access in Vehicular Environments (WAVE) architecture mandates the use of PKI mechanisms, where service application messages are encrypted and vehicle safety messages are digitally signed. All implementations of IEEE1609.2 standard [7] shall support the Elliptic Curve Digital Signature Algorithm (ECDSA) [8] over the two NIST curves P-224 and P-256. Unfortunately, security mechanisms come with overhead that affects the performance of the V2V communications, and hence that of the safety applications.

Many of the envisioned safety and driver-assistance applications require tight deadlines for message delivery. Consequently, security mechanisms must take these constraints into consideration and impose low processing and communication overhead.

In this paper, we assess the processing and communication overhead of the authentication mechanism provided by ECDSA. As the braking distance is an important metric in emergency braking application, we investigate the impact of the authentication key size on the braking distance. We analyze the effects of signature overhead and network density.

The paper is organized as follows. First, we survey previous work. The overhead of authentication mechanism is discussed in section 3. In section 4, we present simulations results of safety message and discuss the impact of ECDSA on the braking distance. Section 5 concludes the paper.

2. Related work

In [6], Iyer et al. provided an evaluation of the computational overhead in V2V communications. They observed that the performance bottlenecks could shift from security layer to MAC layer, depending on the system parameters. They provided interesting values like buffer size at MAC layer. But their work is independent of security protocols and computational capabilities. It does not give results about ECDSA overhead. Moreover, they didn't analyze the communication overhead.

Haas et al. [9] performed simulation using real vehicle mobility from I-80 in Emeryville, California, United States. They compared ECDSA (with P-224) and TESLA, analyzing the communication range and the MAC layer delay. Moreover, they provided an assessment of verification latency for various hardware configurations. Their simulation results show that TESLA performs poorer than ECDSA but has a lower latency because of the smaller packet size.

Our work differs from the above-mentioned studies. First, we investigate the effect of authentication key size. Then we translate the overhead into an issue — namely braking distance — of safety application like cooperative collision avoidance (CCA).

3. Overhead of ECDSA

The total time overhead is the sum of the processing delay and the communication delay. In [10], we investigated the time complexity of ECDSA, and provided an assessment of the processing overhead of ECDSA. In this paper, we extend our work by focusing on the communication overhead.

3.1. Packet size

The Wave Short Message (WSM) is used for safety message like cooperative collision warning or periodic information message. Figure 1 describes the WSM format.

A WSM is signed with ECDSA, and sent using the WAVE Short Message Protocol (WSMP). When a WSM is signed, authentication protocol adds a signed certificate (*signer*) and a signature to the payload.

As we see in figure 1, the unsigned WSM payload is 53 bytes long. The WSM header is 19 bytes long. A certificate of size S_{cert} (plus 1 byte for the certificate type) and a signature of size S_{sign} are appended. In annex C.3 of [5], an example of OBU signing certificate is given. We observe that the length of a certificate depends on two parameters:

- the point size of the elliptic curve *G* depending on the public key algorithm associated with the key: S_{pu}(in bits).
- the size of the signature used to sign the certificate: $S_{sigcert}$(in bits).

Length	Field			
1	WSM version			
1	Security Type = signed(1)			
1	Channel Number			
1	Data Rate			
1	TxPwr_Level			
1	Application Class Identification			
1	ACM Field Length			
10	ACM			
2	WSM Length			
1	WSM Data	signer	type = certificate	
125			certificate (see C.3 for details of fields)	
2		payload	mf (encoded as 01 0a)	
32			application_ data	
8			transmission_ time	
4			transmission_ location	latitude
4				longitude
3				elevation_ and_confidence
28		signature	ecdsa_ signature	r
28				s

Figure 1. WAVE Safety Message format

The length of a certificate S_{cert} (in bytes) is defined in (1):

$$S_{cert} = \left(\frac{S_{pu}}{8} + 1\right) + \left(\frac{S_{sigcert}}{8} \times 2\right) = \left(\frac{S_{pu}}{8} + 1\right) + \left(\frac{S_{sigcert}}{4}\right) \quad (1)$$

The length of a signature S_{sign} (in bytes), attached to a message, depends on the elliptic curve $S_{sigmess}$ (in bits) used in ECDSA.

$$S_{sign} = \frac{S_{sigmess}}{8} \times 2 = \frac{S_{sigmess}}{4} \quad (2)$$

The size overhead of authentication is defined (in bytes) by:

$$S_{ov} = S_{cert} + S_{sign} = \frac{S_{pu}}{8} + 1 + \frac{S_{sigcert}}{4} + \frac{S_{sigmess}}{4} \quad (3)$$

The total length of a WSM (in bytes) is defined in (4):

$$S_{WSM} = \frac{S_{sigmess}}{8} \times 2 + \frac{S_{sigcert}}{8} \times 2 + \frac{S_{pu}}{8} + 1 + 32 + 20 + 53$$
$$S_{WSM} = \frac{S_{sigmess}}{4} + \frac{S_{sigcert}}{4} + \frac{S_{pu}}{8} + 106 = S_{ov} + 105 \quad (4)$$

where 32 bytes is the header length in the certificate.

We assume a certificate signed by ECDSA (P-224) of 125 bytes, which simplify (4):

$$S_{WSM} = S_{sign} + 198 \quad (5)$$

3.2. Communication delay

The *communication delay* is defined as the time elapsed between the generation of a packet and its successful reception at the application layer. It includes the queuing delay and the medium service time (due to backoff, transmission delay, and propagation delay, etc.).

Many delay analysis models for IEEE 802.11 MAC protocol have been proposed. To our knowledge, model from [11] is the best suited for VANET environment where there is no acknowledgement, and MAC layer retransmissions.

In [11], the mean beacon transmission delay T_{tx} is defined as:

$$T_{tx} = \frac{W-1}{2}[\sigma P_e + T_s P_s + T_c P_c] + \\ +(1-\pi)^{n-1}(1-e)T_s + (1-(1-\pi)^{n-1}(1-e))T_c \quad (6)$$

where W is the contention window, σ is the slot time, P_e, P_s, P_c are the probabilities of empty channel, successful transmission and collision respectively. T_s and T_c are the duration of successful transmission and collision respectively. These values depend on the packet size. π is the transmission probability in a slot by an active station, n is the total number of vehicles, e is the probability of a beacon packet corruption by noise.

From (6), we conclude that the communication delay depends on the packet size and the network density.

3.3. Total delay

The authentication time overhead of a message *M* is given as follows:

$$T_{ov}(M) = T_{sign}(M) + T_{tx}(Sign_{\Pr K_V}[M]) + T_{verify}(M) \quad (7)$$

where:

- $T_{sign}(M)$: time to sign M.
- $T_{verify}(M)$: time to verify M.
- $Sign_{\Pr K_V}[M]$: signature of M by the sender V and includes the Certificate Authority's certificate of the signing key.
- $T_{tx}(Sign_{\Pr K_V}[M])$: time to transmit the signature.

Since we have $Sign_{\Pr K_V}[M] = S_{ov}$, the transmission delay is given by:

$$T_{tx}(S_{ov}) = \frac{W-1}{2}\left[\sigma P_e + (T_h + \frac{S_{ov}\times 8}{D_R} + DIFS + \delta)P_s + (T_h + \frac{S_{ov}\times 8}{D_R} + EIFS + \delta)P_c\right] \\ +(1-\pi)^{n-1}(1-e)(T_h + \frac{S_{ov}\times 8}{D_R} + DIFS + \delta) \quad (8) \\ +(1-(1-\pi)^{n-1}(1-e))(T_h + \frac{S_{ov}\times 8}{D_R} + EIFS + \delta)$$

Using (8), (7) becomes:

$$T_{ov}(M) = T_{sign}(M) + T_{tx}(S_{ov}) + T_{verify}(M) \quad (9)$$

3.4. Consensus

Many applications depend on WAVE Short Message reception. For example, CCA application warns the driver in function of information included into the WSM. To avoid false information, application waits for x WSMs before warning the driver. This mechanism is called *consensus* [17]. The selection of x is an open issue and is out of the scope of this paper. We assume that each vehicle has at least x one-hop neighbors. As we already mentioned, each vehicle sends a WSM every λ second. A vehicle V_i sends a k-th WSM at $t_i^k = t_0^i + k\lambda$ second where t_0^i is the first sending time. If a vehicle has to wait for two messages, the worst case is when $\Delta t_0 = t_0^j - t_0^i = \lambda$. To compute the total time overhead needed in our case, we extend formula (9) taking into account x and add the maximum delay between two packet generations Δt_0:

$$T'_{ov}(M) = x \times (T_{sign}(M) + T_{tx}(S_{ov}) + T_{verify}(M)) + \lambda \quad (10)$$

4. Safety mechanism performance analysis

4.1. Simulation setup

All DSRC parameters used in this paper are listed in table 1.

We conduct simulations using ns-2.34 while we make use of a Nakagami's probabilistic radio propagation model, because recent research has shown that a fading radio propagation model, such as the Nakagami's model is best for simulation of a WAVE environment [12][13]. We use the ns-2 extensions provided by Chen et al. [14] as physical and MAC layer. We consider the following scenario.

Table 1. Simulation parameters

Parameters	**Value**
Propagation delay δ (μs)	1
Time slot σ (μs)	13
Packet size S_A (bytes)	73, 198, 254, 262
Vehicle density β (veh/km/lane)	[1;45]
DIFS (μs)	64
EIFS (μs)	248
Packet interarrival time λ (sec)	0.1
CWMin	15
Data rate D_R (Mbps)	6
Link Layer queue size (packets)	50
Vehicle speed (m/s)	v_1=27.7 v_2=30.5 v_3=36.1
Radio range R (meters)	300

In a highway of 5 km long, with 3 lanes in one direction, vehicles have a max velocity v_i where *i* is the lane number. Vehicle speeds are chosen

according to speed limitation on French highway and average speed on a three lanes highway. We assume a uniform density β in veh/km/lane. Each node sends WSM of size S_A where A is the authentication chosen (WSM payload, WSM+certificate, WSM+certificate+P-224, WSM+certificate+P-256). According to safety-applications requirements [15], each node generates one packet every 100 ms, and uses a transmission range of 300 m for message exchange. We increase the density from 1 to 45 veh/km/lane, which means from free-flow to jam scenario. In our simulations, vehicles should enter the system in such a way that the network density remains stable. But in ns-2, all nodes are generated at the beginning of the simulation. So, if the simulation has 600 nodes, then at $t = 0$ there are 600 nodes at the same place. As in standard ns-2, nodes could not be in sleep mode, they will participate to the network traffic even if they do not exist in reality. To avoid this undesirable effect, we monitor the traffic in an area $y = [2000; 3000]$, denoted by the square in figure 2.

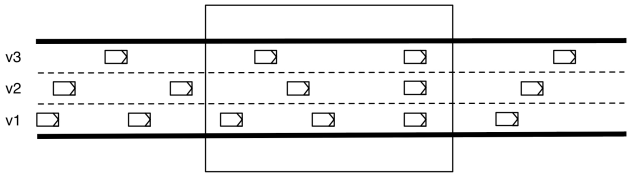

Figure 2. Highway scenario

4.2. Simulation results

4.2.1. Processing delay

Vehicles have to generate a signature for each sent message, and verify signature for each received message. The time required for these operations is called *processing delay*. ECDSA with a P-224 curve (respectively P-256) fits with an authentication key size of 224 bits (respectively 256). Table 2 taken from [10], which gives T_{sign} and T_{verify} for a Pentium D 3.4GHz workstation, shows that using P-256 instead of P-224 in the signature generation adds a time overhead of 33.2%. Using P-256 instead of P-224 in the signature verification adds a time overhead of 33.4%. Theoretical analysis of ECDSA shows a linear-time complexity depending on the key size [10]. In table 2, the processing delay increases when key size increases. As discussed in next sections, this increase may have a significant impact in high-density scenarios.

Table 2. Signature generation and verification times on a Pentium D 3.4Ghz workstation [10]

Key size (bits)	Signature generation (ms)	Signature verification (ms)
224	2.50	4.97
256	3.33	6.63

4.2.2. Communication delay

There are six impacting parameters in a DSRC scenario: packet size, data rate, vehicle density, transmission power, message frequency and the number of lanes. The communication delay depends on the message size, the arrival rate at the MAC layer queuing system, the number of vehicles within radio range, the probability of collision and the probability that the channel is busy. It is well established that larger the message size is, larger the probability of collision is. We focus on the effect of vehicle density because it affects the number of vehicles within radio range and the arrival rate at the MAC layer, and it increases the probability of collision or channel availability.

Figure 3 shows the impact of density on communication delay for one packet transmission. When the density increases, more vehicles are within the radio range. They compete for channel access, thus increasing the probability of collision. We simulated the same scenario with different packet size. WSM payload represents the WSM without security, i.e. 73 bytes. WSM+certificate represents the WSM appended with a signed certificate, i.e. 198 bytes. P-224 (respectively P-256) represents WSM+certificate signed with ECDSA and P-224 curve (resp. P-256 curve), i.e. 254 bytes (resp. 262 bytes). Figure 3 shows that without security, the density has a lower impact on the communication delay than in the other cases. According to the small size of WSM in this case, this result is obvious. Adding a security mechanism doubles the communication delay for density lower than 30 veh/km/lane. In high-density conditions, communication delay is multiplied by three. If we focus on the authentication key size, the comparison between P-224 and P-256 shows an overhead from 3% to 8%.

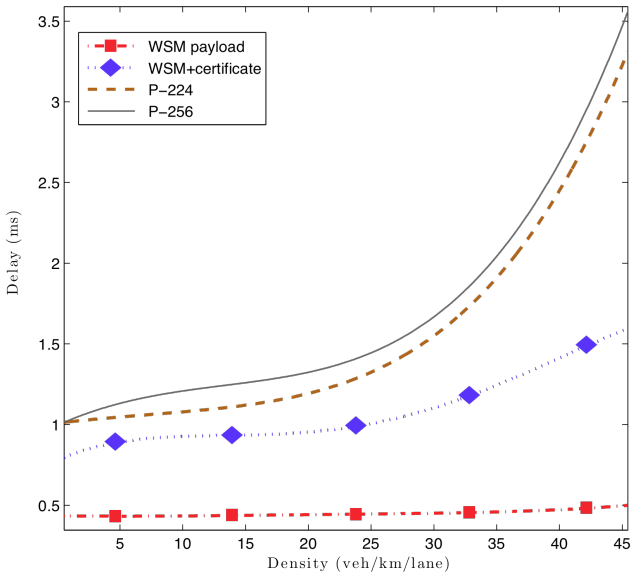

Figure 3. Communication overhead: effect of density on delay for different packet size

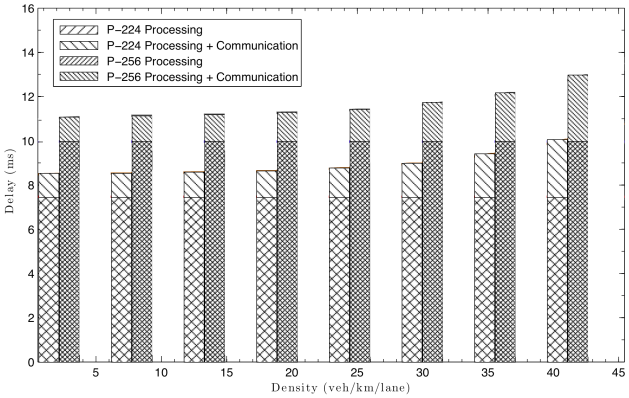

Figure 4. P-224/P-256 overhead: Processing vs Total overhead

Figure 4 highlights the ratio of processing overhead into the total overhead. More than 80% of ECDSA overhead is due to the processing mechanism.
Safety applications need to check every received message to make decisions (lane change, brake, etc.). These applications have to wait for the message (generation and transmission) and verify it. The total delay overhead (processing and communication) is denoted by $T_{ov}^{N_{TX}}$ and defined as

$$T_{ov}^{N_{TX}}(M) = N_{TX} \times T_{sign}(M) + N_{TX} \times T_{tx}(S_{ov}) + N_{TX} \times T_{verify}(M)$$

N_{TX} is the number of neighboring vehicles equipped with the DSRC system, which are in transmission range and defined in [16]. For example, in a high-density scenario, one vehicle will have to wait for 80 signature generations, 80 WSMs transmissions, and 80 signature verifications.

As shown in figure 5, the processing delay for N_{TX} messages is higher than the communication delay. For a density of 35 veh/km/lane, the communication delay is about 100 ms, while processing delay is 400 ms. Moreover, figure 5 details the difference between P-224 and P-256. Using P-256 instead of P-224 has a greater impact on processing delay than on communication delay. Indeed, the communication curves are slightly different, while there is a gap between the two processing curves.

Processing and communication delay are merged into the figure 6, which shows that the total overhead introduced by ECDSA is greater than 1 second for P-256 and 800 milliseconds for P-224 in high-density situations. The comparison between P-256 and P-224 shows an overhead of 30%. Then, the authentication key size could have a high impact on the delay overhead and on the behavior of the application.

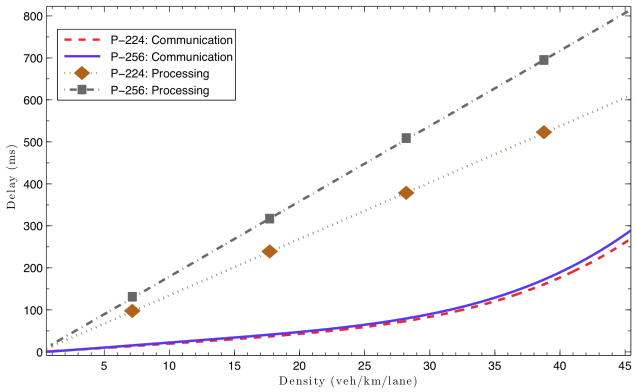

Figure 5. Delay overhead: Processing vs Communication

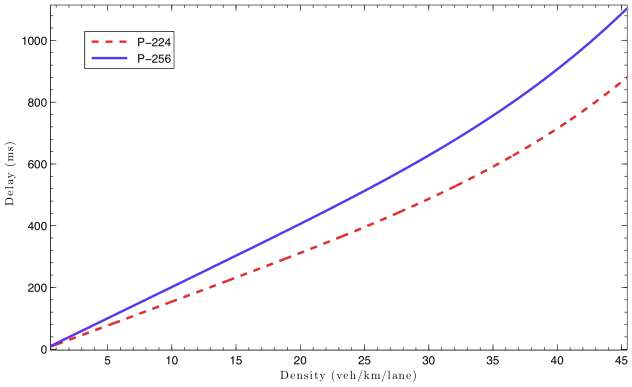

Figure 6. Communication overhead for one packet per vehicle

4.2.3.Braking distance

In the context of a CCA system, the distance is a critical metric. As described in [10], the braking distance (in meters) for a vehicle *V* is defined as $D_B = \frac{{v_V}^2}{2a}$.
In normal environment, on a dry road with a deceleration rate a=6.8 m/s^2 and a velocity v_v of 36.1 m/s, *V* stops in D_B=95.82 m. But, if *V* has to verify N_{TX} messages, it stops in $D_B + N_{TX} \times (v_v \times T_{verify})$ m, resulting in an increase of 16.6% of the braking distance for 80 signature verifications (with P-256). Moreover, if the vehicle does not make decisions (brake, lane change) without the driver agreement, we have to add the driver's reaction time of 1.5 seconds. In spite of technologic advances in pneumatic and automotive increase driver's safety by offsetting the non-respect of the safety distance, the deployment of ECDSA may jeopardize these ameliorations.

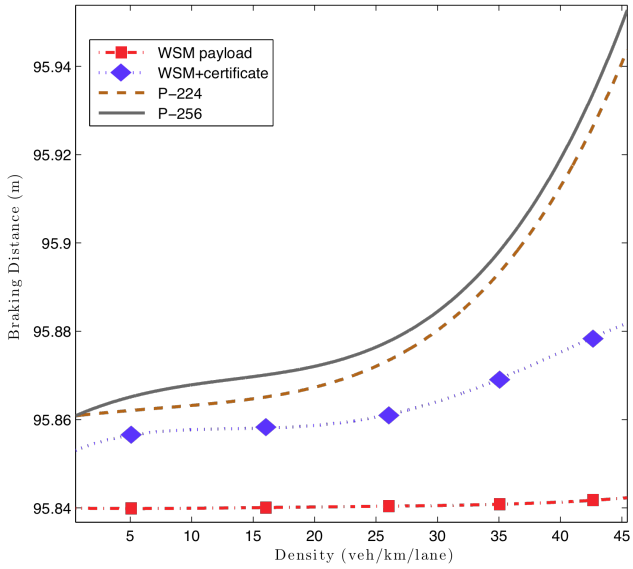

Figure 7. Communication overhead: impact of density on braking distance for different packet size

As figure 3, figure 7 shows the impact of density for different packet size, but focused on braking distance. We transformed the communication delay into distance and added it to D_B. For one packet transmission, we observe that adding a security mechanism increases the braking distance. In low-density scenario, ECDSA adds less than 0.04 m. In high-density, the communication overhead adds from 0.04 to 0.1 m.

Figure 8 shows the processing and communication overhead of ECDSA. We observe that the processing mechanism adds 0.3 m to the braking distance for P-224 and 0.4 m for P-256. For one packet transmission, the processing overhead is greater than the communication overhead.

Figure 9 shows the difference between processing and communication delay in function of density for N_{TX} messages to verify. As in figure 5, the processing overhead is greater than the communication overhead. For a density of 35 veh/km/lane, communication adds 5 m to the braking distance, while the processing adds more than 17 m. One more time, using P-256 instead of P-224 has a greater impact on the processing. Indeed, there is a gap of more than 5 m, which is greater than the average length of a car.

As figure 6, figure 10 shows the total ECDSA overhead for one vehicle, which has to wait and check for N_{TX} messages, depending on the density. In high-density scenarios, the braking distance is increased by more than 20 m. If P-256 is used, it adds an overhead from 1% to 8% higher than P-224. We conclude that the authentication has a great impact on the braking distance. Consequently, the key size should be chosen carefully.

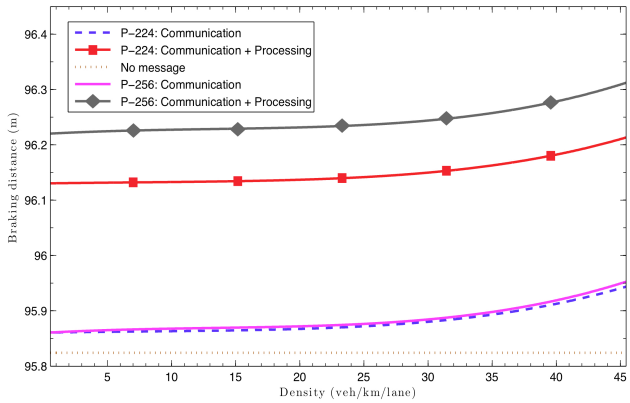

Figure 8. Processing and communication overhead of ECDSA: impact of density on braking distance for one packet transmission

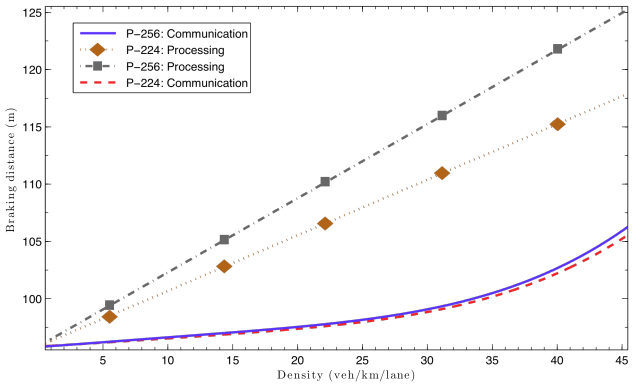

Figure 9. Braking distance: Processing vs Communication

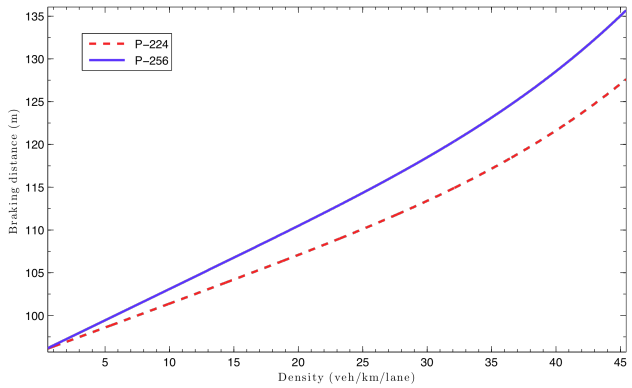

Figure 10. ECDSA overhead for one packet per vehicle

4.2.4. Effect of consensus

For fault tolerance, critical safety applications need $x(t)$ messages at time *t* before warning the driver or making a decision. This parameter is defined by formula (11):

$$x(t) = \min\left(x_{MAX}, \lceil y(t) \times N_{TX}(t) \rceil\right) \quad (11)$$

where $N_{TX}(t)$ is the number of neighbors at time *t*, and x_{MAX} is the maximum number of messages

needed to insure real-time constraints of the application. As $T_{ov}(M)$ and T_{MAX} (the maximum time allowed by the application before having critical impact) are known, x_{MAX} could be computed. $y(t)$ is a weight function between 0 and 1, defined by formula (12):

$$y(t) = \begin{cases} 1 \ if \quad N_{TX}(t) < 3 \\ \dfrac{1}{\ln(N_{TX}(t))} \quad otherwise \end{cases} \qquad (12)$$

With this assumption and formula (10), we obtain the figures 11 and 12. We observe in figure 11 that the authentication key size has an impact on delay when a consensus mechanism is used. When using P-256 instead of P-224, the driver will be warned, on average, 16 ms later in low-density and 54 ms in high-density scenarios. Figure 12 shows that the braking distance is increased, on average, by 12 m when we add a security mechanism in high-density scenarios. For x WSM generations, transmissions and verifications, the authentication key size P-256 adds from 0.6 m to 2 m compared to P-224.

Safety applications will need to compute the critical distance in order to predict or make preventive decisions. We could reduce the time before making decision by combining radar or lidar to V2V communications. Moreover this merge could help to detect false data dissemination.

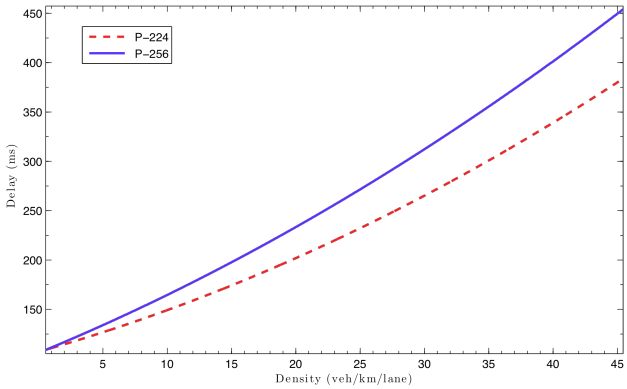

Figure 11. Impact of consensus on communication delay

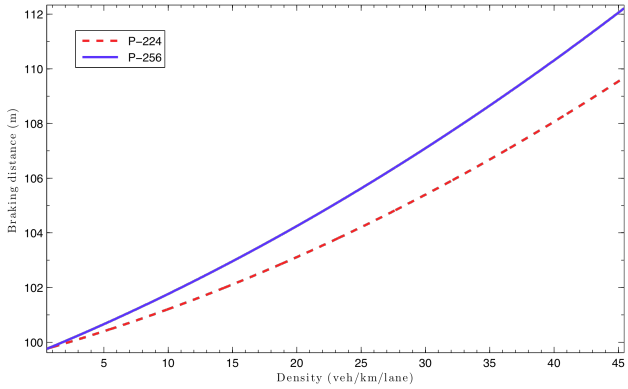

Figure 12. Impact of consensus on braking distance

As highlighted in previous sections, the processing overhead is higher than the communication overhead. To reduce the total authentication overhead, we have to first decrease the processing time. One way is to improve the signature algorithm or provide into OBU a specific crypto-processor.

Another way is to only verify signature every *i* messages. *i* could be defined in function of the network density. Then, for reducing the communication delay, reduce the packet transmission rate is not a solution, because it is an application requirement. We could reduce the packet size by appending the certificate only every *i* message.

To improve the total authentication overhead, we could combine these optimizations.

5. Conclusion

VANETs deployment has the potential to greatly increase vehicular safety and improve driving experience. Vehicular communications need to be secured. The DSRC standard for vehicular ad hoc networks is based on the ECDSA algorithm for supporting authentication mechanism. Security mechanisms come with overheads that affect the performance of the V2V communications, and hence that of the safety applications. In this paper, we investigate the total overhead of ECDSA, combining the packet size, processing and communication overheads. We focus on safety applications, and analyze the impact of the authentication on the braking distance. We conduct simulation study in order to evaluate the performance of secured beacon safety message dissemination in vehicular ad hoc networks. We pay special attention to safety requirements while studying networking performance issues.

Our results show that the processing overhead is higher than the communication overhead. Depending on the application requirements, the braking distance is increased by more than an average length of a car in high-density scenario.

We highlight the impact of the authentication key size in order to adapt security parameters to the application requirements. Some optimizations were proposed.

To avoid false data dissemination, we introduce the problem of consensus. A formula is proposed to dynamically change the number of messages needed to check the data consistency.

As of future work, we intend to enhance the authentication overhead assessment by adding the certificate distribution, verification and revocation mechanisms. Indeed, when a node receives a WAVE short message, it has to check the certificate appended to the message.

6. References

[1] Blincoe L., “The Economic Impact of Motor Vehicle Crashes”, U.S. Department of Transportation National Highway Traffic Safety Administration, Technical Report, 2002.

[2] NHTSA, “Traffic Safety Fact Sheet”, 2008.

[3] CARE, “Community Road Accident Database”, 2007.

[4] Hubaux J.P, Capkun S., Luo J., “The security and privacy of smart vehicles”, *IEEE Journal of Security & Privacy*, vol. 2 (3), pp. 49-55, 2004.

[5] IEEE Vehicular Technology Society, “5.9 GHz Dedicated Short Range Communications (DSRC) - Overview”, http://grouper.ieee.org/groups/scc32/dsrc/

[6] Iyer A., Kherani A., Rao A., Karnik A., “Secure V2V communications: Performance impact of computational overheads”, *IEEE Conference on Computer Communications Workshops (INFOCOM)*, pp. 1-6, Phoenix, USA, April, 2008.

[7] IEEE, “IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages”, IEEE Standard 1609.2-2006, 2006.

[8] ANSI, “Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm”, ANSI X9.62-1998.

[9] Haas J.J., Hu Y., Laberteaux K.P., “Real-World VANET Security Protocol Performance”, *IEEE Globecom Symposium on Selected Areas in Communications*, Honolulu, November 2009.

[10] Petit J., “Analysis of ECDSA Authentication Processing in VANETs”, *3rd IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Cairo, December 2009.

[11] Vinel A., Andreev S., Koucheryavy Y., Staehle D., “Estimation of a Successful Beacon Reception Probability in Vehicular Ad-hoc Networks”, *ACM International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly (IWCMC),* pp. 416-420, Leipzig, Germany, June 2009.

[12] Schmidt-Eisenlohr F., Torrent-Moreno M., Mittag J., Hartenstein H., “Simulation Platform for Inter-vehicle Communications and Analysis of Periodic Information Exchange”, *4th Conference on Wireless On demand Network Systems and Services (WONS)*, pp. 50-58, Obergurgl, Austria, January 2007.

[13] Taliwal V., Jiang D., Mangold H., Chen C., Sengupta R., “Empirical Determination of Channel Characteristics for DSRC Vehicle-to-vehicle Communications”, *1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, pp. 88, Philadelphia, USA, October 2004.

[14] Chen Q., Schmidt-Eisenlohr F., Jiang D., Torrent-Moreno M., Delgrossi L., Hartenstein H., “Overhaul of IEEE 802.11 Modeling and Simulation in NS-2”, *10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems (MSWiM)*, pp.159-168, Chania, Crete Island, Greece, October 2007.

[15] Vehicle Safety Consortium, “Task 3 Final Report - Identify Intelligent Vehicle Safety Applications Enabled by DSRC”, Vehicle Safety Communications Project, 2005.

[16] Wischhof L., “Self-Organizing Communication in Vehicular Ad Hoc Networks”, Ph.D thesis, Hambourg-Harburg University, 2007.

[17] Cao Z., Kong J., Lee U., Gerla M., Chen Z., “Proof-of-Relevance: Filtering False Data via Authentic Consensus in Vehicle Ad-hoc Networks”, *IEEE Conference on Computer Communications Workshop (INFOCOM)*, pp. 1-6, Phoenix, USA, April 2008.