

# Verification of a VANET protocol

## IC-29 miniproject

Marcin Poturalski  
marcin.poturalski@epfl.ch

March 10, 2006

## 1 Introduction

This report summarizes a miniproject carried out in the scope of the IC-29 course (*Self-Organized Wireless and Sensor Networks*). The goal of the project was to verify a VANET protocol, namely the *Distributed Revocation Protocol* (DRP, [1]). This protocol was chosen because of its ad-hoc nature, which makes the verification both interesting and challenging, while being relatively simple, this protocol works.

The initial idea was to apply (semi)-automatic verification techniques and tools, like the security model-checker OFMC ([3]). However, in the end manual verification was performed, because OFMC and similar security verification techniques proved not to be suitable for the purpose of DRP verification; additionally, it was possible to detect attacks against the protocol remaining on the non-automatic level.

The report develops as follows. First, the VANET environment and the general revocation scheme is described, to give the reader an understanding where DRP fits in. Next, the definition of DRP that will be used for verification is introduced; this section is based on [1] and on interaction with the authors of this paper. The latter was necessary, because [1] does not describe in all detail how the protocol works. The following sections presents the adversary model and specify the properties that the protocol should fulfil (these were partly inspired by [4]). Finally, I give a list of discovered attacks, and conclude.

This order of presentation, quite common in verification practice, is usual in cases where verification comes into the picture after the protocol is defined. This differs from the ideal verification approach, where the formal properties are specified before the protocol is defined.

## 2 Context

DRP is build on top of the security scheme of VANETs proposed in the paper [2]. The features important for this paper are the following:

- Every vehicle has a clock and a GPS device, thus being aware of time and location.

- Vehicles exchange (via local broadcasting) so called safety messages. This messages contain information about traffic conditions, dangers on the roads, etc. Vehicles (or drivers) are intended to make their decisions based on this messages (among others factors). Thus, an potential adversary will be able to attacks the system by sending *bogus* safety messages, and making other drivers react in his favor.
- Each vehicle has a set of so called safety keys. This are in fact public key pairs, that are issued by a Certification Authority, along with certificates. They are used to sign messages.
- Each key has a validity period, contained in the certificate; the validity periods of safety keys overlap: at any given moment a vehicle has a substantial number valid keys.
- The keys are anonymous: only the CA can deduce if two keys belong to the same vehicle (besides the owner of the keys).
- Keys are stored in a *Tamper-Proof Device* (TPD) that prohibits the owner of a vehicle from retrieving the keys. The TPD will however sign any message on request with a key of its choice.
- A safety message contains (among other data) a timestamp and sender location, is signed with a safety key and includes a certificate. Thus every vehicle that receives a safety messages can tell if it was send by a legitimate traffic participant.
- The safety key used to sign message is changed frequently (every 2 minutes in normal conditions) to ensure anonymity of the vehicle.
- Safety messages are broadcasted by a vehicle with a certain minimal frequency.

It is easy to see that the CA needs a way to revoke the keys of a misbehaving vehicle. A solution to this is proposed in [1]. The revocation process described in this paper contains the following steps:

1. Misbehaving node is detected by peers.
2. Vehicles exchange messages to decide if to report the misbehavior to the CA.
3. Misbehavior is reported to the CA.
4. CA evaluates the report against its reputation system.
5. CA performs revocation.

DRP is responsible for steps 2 and 3, but for the scope of the project I am only interested in step 2. Note, that although the decision is made based on exchanging messages, it is autonomous for each vehicle (i.e. there is no consensus algorithm involved). Step 4 is left undefined, which affects the verification: the properties that can be specified can only refer to reporting, not the actual revocation. Step 1 is also mostly undefined (some examples are given), which will also have an impact on the DRP model defined in the next section. Step 5 is covered by other protocols defined in [1].

### 3 Defining DRP

The definition (or model) given in this section does not attempt to capture all the aspects of the protocol. On the contrary – whenever necessary or suitable, abstractions from details not important for the verification process are being made.

$R$	Broadcast range of an agent
$N_s$	Number of safety keys belonging to a single agent that are valid at any given time (large)
$B_{max}$	Maximal message bogusness/accusation severity
$T_s$	Maximum time interval between two consecutive safety messages
$T_t$	Time tolerance for received messages
$T_l$	Location tolerance for received messages
$T_b$	Maximum delay between a receiving a message and broadcasting the triggered accusation
$T_p$	Time before purging a message from <b>received</b> or accusation databases
$F$	Severity of a <i>false accusation</i> accusation reason
$\tau$	Revocation threshold

Table 2: The summary of constants used throughout the paper.

#### 3.1 Prerequisites

**Environment model** A vehicle is represented by an *agent*. An agent is a mobile entity, meaning that it has a time-dependant *location*. The space in which agent move is the  $\mathbb{R}^2$  plane with the Euclidean distance. An agent has accurate knowledge of time and its location; this is based on an assumption that a vehicles clock and GPS device never fail.

An agent is able to receive, process and send messages. The sending mechanism is broadcast. The message broadcasted by an agent at some time  $t$  is received at the same time  $t$  by all the other agents in its range, that is in the ball of radius  $R$  (figure 1). Note that this definition implies that sending is reliable.

I will distinguish two types of agents: honest agents and dishonest agents; the latter are controlled by the adversary.

**Keys** Each vehicle is equipped with two unique, distinct sets of cryptographic keys: safety keys and DRP keys; they are used to sign messages. Every key has a validity period, and it is consider to be invalid before and after this period. Safety keys of an agent have overlapping validity periods, i.e. at any time an agent has exactly  $N_s$  valid safety keys; DRP keys have non-overlapping validity periods, i.e. at any time an agent has exactly one valid DRP key.

A related notion is an agent’s active safety/DRP key. At a given time  $t$ , the active DRP key is the single DRP key valid at  $t$ . The active safety key is one

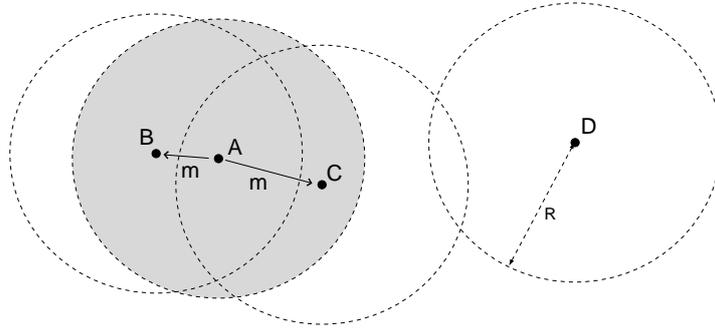


Figure 1:  $A$  broadcasts message  $m$ ; it is received by  $B$  and  $C$ , but no  $D$ .

of the valid keys, with a restriction: once a key stops being active, it can never become active again.

I assume that the cryptosystem used is secure. Therefore in this DRP definition the only role that keys play is to represent the (anonymous) identities of the agent who signed a message with them.

### 3.2 Concept

The idea behind DRP is majority voting. At each agent (*evaluating*) DRP collects data about accusation messages sent against an agent suspected of misbehavior (*accused*). The evaluating needs to be aware of both the accusations against the accused and the agents that do not accuse.

For the voting to be meaningful, only the agents in the accused *neighborhood* should be taken into account. A neighborhood is the set of agents in range of both the evaluating and the accused, as figure 2 illustrates. The logic behind this definition should be clear: first, an agent can only receive messages from agents in its range; second, the votes of agents which are out of range of the accused, and do not receive messages from it, should not be taken into account. In the end, if a majority of the agents in the accused neighborhood do accuse it, it is reported by the evaluating to the CA.

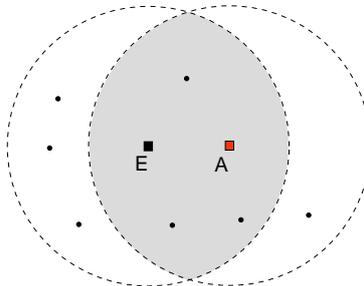


Figure 2: Only agents in range of both the evaluating ( $E$ ) the accused ( $A$ ) are considered the neighborhood of  $E$  from  $A$ 's point of view.

This seems pretty straightforward. However, when one goes into the details, the neighborhood concept becomes more troublesome. First, an agent only bases its knowledge about the neighborhood on the messages received (some of which might be intentionally misleading); second an agent receives messages signed with anonymous keys and is not able to deduce that two keys belong to the same agent; on top of that, the agents are mobile.

### 3.3 Messages

Because DRP collects data from safety messages, it is necessary to include them in the DRP model. However, while DRP messages in the model reflect the messages exchanged by the real protocol faithfully, for safety messages stronger abstraction is needed. The reason is that there is no clear definition of what data a safety message can contain, and in what ways it can be bogus. Thus, a safety message is defined with a *bogus* field, which explicitly expresses the intention of the message sender to make the message bogus or not. When the bogus field is equal to 0, then the message is non-bogus. Otherwise it is bogus.

timestamp	$\mathbb{R}$
location	$\mathbb{R}^2$
bogus	0 or $[1, B_{max}]$
sender	safety key

DRP messages defined in [1] are *accusation messages* and *disregard messages*. For simplicity, in this model only the first are defined – it is enough to perform various attacks.

accused key	safety or DRP key
timestamp	$\mathbb{R}$
accused location	$\mathbb{R}^2$
reason	$[1, B_{max}]$
sender	DRP key

If the **sender** field of a message  $m$  is  $k$ , I will on occasion write that the message  $m$  is signed with  $k$ .

**Secure cryptosystem assumption** I assume that the cryptosystem and the TPD are secure. In the DRP model this is reflected by the following restriction:

- ( $M_0$ ) An agent can only send messages with the sender field being its active key (safety or DRP, with respect to the type of message), unless it is re-sending a previously received message.

### 3.4 Details

The details of DRP definition given in this section only apply to honest agents. Dishonest agent will be covered in the section about the adversary.

**Stored data** The DRP stores the following data at every honest agent:

**received database** – received safety messages;

**accusation database** – accusation messages; at most one message per (accusing key, accused key) pair;

**neighborhood database** – for every accused key, a set of keys that are considered to be its neighborhood;

**suspect database** – for every accused key, the status: **accused** or **disregarded**.

The last database is crucial for reporting: a key is reported if and only if it is given the **disregard** status.

**Safety messages** An honest agent broadcasts non-bogus safety messages with time and location equal to the broadcasting time and its location at that time. Time interval between two such messages is not greater than  $T_s$ .

**Triggers** The DRP is triggered if and only if a message is received.

**Message preprocessing** Before any other operations are performed, a received message  $m$  is checked for integrity, taking into account the receiving time  $t$  and current agent location  $l$ . In detail, if any of the following conditions does not hold, the message is not further processed:

1. for some time  $t' \in [t - T_t, t]$  the key  $\mathbf{sender}(m)$
2.  $\mathbf{timestamp}(m)$  is in the validity period of  $k$
3.  $|\mathbf{timestamp}(m) - t| \leq T_t$
4.  $|\mathbf{location}(m) - l| \leq T_l$

Note that while the two first conditions are straightforward, the latter two, which I will denote **anti-replay** are more controversial. In fact, they are not coherent with [1], where a detection of one of these conditions causes an agent to accuse the sender. However, it was later discovered that such behavior leads to some replay attacks. I will describe this in more detail in the section devoted to attacks.

**Databases updates** Databases are updated as follows:

**received database** – every received safety messages is stored (thus only messages send by other agents are stored);  
a message is purged after  $T_p$ ;

**accusation database** – an accusation message  $m$  is inserted into the database only when:

- an accusation message  $m$  is send,
- an accusation message  $m$  is received that is not against one of agent's keys,

but only if there is no message for a given (accusing, accused) key pair in the database or the severity of the new messages is larger then the severity of the old message (the old message is replaced);

a message is purged after  $T_p$ ;

**neighborhood database** – a key is added in two cases:

- an accusation message  $m$  is received that is not against one of agent’s keys; the key  $\mathbf{sender}(m)$  is added to the neighborhood of key  $\mathbf{accused}(m)$ ;
- before the revocation quotient (see below) against key  $k$  at location  $l$  is checked, the **received database** is scanned for all messages with locations in the range of  $l$ ; for every such message  $m'$  the key  $\mathbf{sender}(m')$  is added to the neighborhood of  $k$ ;

**suspect database** – a key is added to this database with an **accused** status when a accusation message against it is added to the **accusation database** for the first time;

it is given a **disregard** status when the revocation threshold (see below) is reached.

Note that updating the **neighborhood database** described above is the way that the fundamental concept of neighborhood is defined for the purpose of this model.

**Accusation** An accusation message  $m$  is send by an agent only when:

1. A bogus safety message  $m'$  is received and *detected*; this means that if an agent receives an accusation message it can non-deterministically decide to detect or not.

The accusation message  $m$  is then the following:

accused key	<b>sender</b> ( $m'$ )
timestamp	current time
accused location	<b>location</b> ( $m'$ )
reason	<b>bogus</b> ( $m'$ )
sender	active DRP key

2. An accusation message  $m'$  is received with **accusedkey**( $m'$ ) belonging to the agent;

The accusation message  $m$  is then the following:

accused key	<b>sender</b> ( $m'$ )
timestamp	current time
accused location	current location
reason	$F$
sender	active DRP key

The actual broadcast of the message takes place some time  $0 < \Delta < T_b$  after the triggering message was received.

Non-determinism used in point 1 is the standard way to model complex protocol behavior, which in this case is the mechanism detecting bogus messages (recall that it is not defined). It completes the abstract way of defining bogusness of messages.

Note that in the case 1, the precaution made in [1] against DoS attacks is skipped: an accusation is always send. It is reasonable as the proposed model is not able to cover these kinds of attacks.

**Revocation quotient** The revocation quotient against key  $k$  is checked every time a bogus safety message signed by  $k$  is detected or an accusation message against  $k$  is received. Note that in both cases the location  $l$  of the accused key is known.

Key  $k$  is given a *disregard* status by an agent, if the revocation threshold is reached:

$$R_k = \frac{1}{N_k + 1} \sum_i \omega_i \mu_{ik} > \tau$$

- $N_k$  is the size of neighborhood of key  $k$  (**neighborhood database**),
- $\mu_{ik}$  is the severity of accusation against  $k$  signed with key  $i$  (0 if no accusation was received),
- $\omega_i$  indicates how trustworthy key  $i$  is: this is calculated by dividing the number of accusations against  $i$  by  $(N_i + 1)$ ,
- $\tau$ , the *Revocation Threshold*, is by default equal to 0.5.

The accusations are taken from the **accusation database**.

## 4 Adversary model

As mentioned before the adversary is considered to have control over all the dishonest agents.

A dishonest agent is allowed to send any messages, but it must respect the restriction ( $M_0$ ). In practice, this means that a dishonest agent is allowed to send both bogus and non-bogus safety messages and arbitrary accusation messages, as long as they are signed with its active key. It can also restrain itself from sending the obligatory safety messages, thus hiding its presence from the DRP point of view.

**Stronger adversary** It is possible to make the adversary stronger by relaxing the condition ( $M_0$ ). First, the assumption about a secure TPD can be dropped. This will result in the following relaxed restriction:

- ( $M_1$ ) An agent can only send messages with the sender field being one of its keys (safety or DRP, with respect to the type of message), unless it is re-sending a previously received message.

The second, orthogonal way to make the adversary stronger is to allow dishonest agents to exchange messages by a medium different than the one used to send other messages. Thus an agent can broadcast messages created by other agents. The relaxed condition is:

- ( $M_2$ ) An agent can only send messages with the sender field being an active key (safety or DRP, with respect to the type of message) of its or some dishonest agent, unless it is re-sending a previously received message.

Both of these relaxed conditions seem to be possible to implement in practice by an adversary sophisticated enough.

## 5 Properties

Intuitively, on the top level an ad-hoc revocation protocol, like DRP, should allow to revoke misbehaving agents, while not allowing the revocation of agents that do not misbehave. As mentioned before, it is not possible to reason about revocation in this framework, as it is done by the CA, behavior of which is left undefined. Thus, it is necessary to express the desired properties in terms of reporting, i.e. giving a key the **disregards** status.

The "misbehaving are revoked" property can be expressed as follows, where  $T$  is threshold value:

- ( $P_T$ ) If at least  $T$  honest agents detect bogus messages signed with some key  $k$ , then some honest agent will eventually give key  $k$  a **disregard** status.

The "honest are not revoked" property can be expressed as follows:

- ( $S$ ) A key belonging to an honest agent is never given a **disregard** status by another honest agent.

Note that only honest agents are taken into account here. This is because a dishonest agent does not need to give a disregard status to a key to report it (in fact, it does not need to even have a **suspect database**).

Each property will be evaluated against all possible executions, and should always hold – otherwise it will be considered violated. However, since DRP is in principle a majority voting protocol, if the dishonest agents are in majority, they will be able to prevent the protocol from working correctly. Taking that into account I need to introduce an additional ratio condition ( $r \in [0, 1)$ ):

- ( $R_r$ ) At any time for any honest agent the ratio of dishonest to all agents in its range is not greater than  $r$ .

Thus, the properties that should hold are: ( $R_r$ )  $\Rightarrow$  ( $P_T$ ) and ( $R_r$ )  $\Rightarrow$  ( $S$ ) for some  $0 < r \leq 0.5$  and  $T \geq 1$ .

## 6 Discovered attacks

In this section I describe the executions that violate the properties defined in the previous section. The first one is described in a detailed manner, to allow the reader seeing how the protocol model works in practice. The subsequent examples are more concise.

**Bogus message** Assume the following set of agents: honest agents  $A_1, A_2, \dots, A_n$  and a dishonest agent  $E$ . The locations are such that throughout the entire execution every two agents are in range. Note, that condition  $(R_{\frac{1}{n}})$  fulfilled. I will evaluate the execution at agent  $A_1$ , but because of symmetry the behavior will be identical for every honest agent. The execution proceeds as follows:

1. Agents  $A_1, A_2, \dots, A_n$  send non-bogus safety messages  $m_1, m_2, \dots, m_n$  (respectfully), where  $\mathbf{sender}(m_i) = k_i$ .

After receiving all these messages (except, obviously,  $m_1$ ) the **received database** at  $A_1$  is updated with messages  $m_2, m_3, \dots, m_n$ .

2. Agent  $E$  sends a bogus safety message  $m_E$ , such that  $\mathbf{sender}(m_E) = k_E$  and  $\mathbf{bogus}(m_E) = 1$ . It is added to the **received database**.

Assume that every agent  $A_i$  detects this message bogusness. Thus each agent updates the **accusation database** with  $m_i^a$ , which is its accusation message against  $k_E$ ;  $\mathbf{sender}(m_i^a) = k_E$ .

The revocation quotient against  $k_E$  is checked. The neighborhood of  $k_E$  is first updated with the  $k_2, \dots, k_n$  and  $k_E$  keys from the safety messages in the **received database**. Thus, the revocation quotient is  $R_{k_E} = \frac{1}{n+1}(1) = \frac{1}{n+1} \leq \tau = \frac{1}{2}$  and the threshold is not reached.

3. The accusation messages  $m_2^a, \dots, m_n^a$  are received by  $A_1$ . When  $m_i^a$  is received the **accusation database** is updated, and  $K_i$  is added to  $k_E$  neighborhood. The revocation quotient is checked every time, but the threshold is not reached.

After receiving the last accusation message, the revocation quotient is checked for the last time:  $R_{k_E} = \frac{1}{n+n-1+1}(\underbrace{1 + \dots + 1}_n) = \frac{n}{2n} = \frac{1}{2} \leq \tau$ .

The threshold is not reached.

4. In this execution  $E$  does not send any more bogus messages signed with  $k_E$ . Thus, the quotient against the key  $k_E$  will never be checked again and it will never be given the **disregard** status.

If  $T \leq n$  then the property  $(R_{\frac{1}{n}}) \Rightarrow (P_T)$  is violated by this execution. This is a strong negative result, as  $n$  can be made large, making the threshold  $T$  large and the ratio constant in the  $(R)$  condition close to 0.

**Discussion** This problem is caused by the fact that each accusing agent in the neighborhood is counted twice: once as a safety key, and a second time as a DRP key. The problem can be solved by redefining the neighborhood: e.g. only putting safety keys into the neighborhood. Simple and effective as it might seem, it can help the adversary to violate the dual  $(S)$  property: by not sending safety messages a dishonest agent can make the neighborhood smaller than it actually is, thus making its accusation more likely to reach the revocation threshold.

A more fundamental, but in my opinion better solution is to use a single set of keys, instead of different keys for safety and DRP messages, possibly making the keys non-overlapping (which also prevents the sybil attack described later).

**Newcomer attack** Assume the following set of agents: honest agents  $A_1, A_2, \dots, A_n$  and a dishonest agent  $E$ . The location are such that throughout the entire execution agents  $A_2, \dots, A_n$  and  $E$  are in range. Initially  $A_1$  is out of range of all the other agents. Note, that condition  $(R_{\frac{1}{n-1}})$  fulfilled. I will evaluate the execution at agent  $A_1$ ; note that its databases are initially empty. The execution proceeds as follows:

1. Agent  $A_1$  comes into range of all the other agents.
2. Agent  $E$  sends an accusation message  $m$ :

accused key	$k_2$ (some key of $A_2$ )
timestamp	sending time
accused location	the location of $A_2$
reason	1.1
sender	$K_E$

3. After receiving this message  $A_1$  updates its databases (the neighborhood of  $k_2$  is  $\{K_E\}$ , and check the revocation quotient against  $k_2$ :  $R_{k_2} = \frac{1}{1+1}(1.1) = 0.55 > \frac{1}{2} = \tau$ . Key  $k_2$  is given the disregard status.

This execution violates the  $(R_{\frac{1}{n-1}}) \Rightarrow (S)$  property.

**Discussion** The problem that is exposed by this attack is checking the revocation quotient immediately after the first accusation message. A simple solution is to wait  $T_s$ , so that every agent in range has the chance to send its safety message, updating the newcomer **received database** and giving it the proper picture of the neighborhood of the accused. This will prevent this particular attack, however similar attack based on agents mobility might be possible.

## 6.1 Stronger adversary attacks

**Sybil attack** This attack is performed with the relaxed  $(M_1)$  restriction. It is an attack against  $(P_T)$ . Only the main concept is described.

An misbehaving agent, before sending a bogus safety message can send many non-bogus safety messages, each signed with a different (but valid) safety key (remember that the number of valid safety keys is very large). This pollutes the **received databases** of honest agent, letting them believe that there are many more agent in the range of the accused, then there actually is. It will prevent the revocation threshold from being reached.

**Cooperation attacks** This group of attacks can be performed in the relaxed  $(M_2)$  model.

First, an attack similar to the sybil attack above can be performed – instead of sending messages signed with its keys an agent send messages signed with the keys of other dishonest agents (again, a large number, as the agent can be far away).

Second, an dishonest agent can send many accusation against a certain key  $k$ , each signed with a different DRP key, thus making the honest agents in range set the status of  $k$  to **disregard**. This violates the ( $S$ ) property.

## 6.2 Replay attacks

This section describes an attack that is not possible in the given model of DRP, and justifies a decision made in the definition phase that prevents it.

Recall the **anti-replay** assumption, that prevented an agent from accusing the sender of messages that are out of time or out of place. In [1] the authors propose to accuse instead of dropping, since the most probable cause of such behavior is a malfunctioning clock or GPS device. The authors conclude that this should be reported to the CA, to revoke the vehicle keys, forcing the vehicle owner to fix the problem.

However, a malicious adversary can use this to revoke a vehicle that has correctly working clock and GPS. The only thing that he needs to do is replay a non-bogus safety message send by some vehicle  $A$  at a time or location beyond tolerance range, to make the honest vehicles that receive this message report the key of  $A$  to the CA. The time instance is simple replay, while the location instance requires a more sophisticated adversary, that can record messages at one place, transmit them real-time to a far away location, and broadcast them there. However, both are possible to perform.

## 7 Conclusion

To sum up, let me describe the contribution of this project. First, building a model helped to identify some ambiguities in the description of DRP given in [1]. Specifying the properties allowed to semi-formally reason about the protocol correctness, or in other words to verify it. The verification led to identifying some attacks. These attacks expose some of the difficulties that the protocol should overcome, the foremost being a more robust definition of a neighborhood.

The verification of DRP is not over, and there are many things that can still be done. After the discovered flaws are fixed, a proof of correctness would in desired. Also different approaches to modelling the DRP could be implemented, to give the verification a broader perspective: for example damage-control in the case where dishonest agents are in majority, or dealing with lower level problems, like DoS attacks or jamming. Another interesting extension would be to model the CA reputation system and specify properties that better suit the top-level goals of the protocol – i.e. refer to revocation, not reporting.

## References

- [1] Daniel Jungels, Maxim Raya, Imad Aad and JeanPierre Hubaux. Certificate Revocation in Vehicular Ad Hoc Networks.
- [2] M. Raya and J. P. Hubaux. The security of vehicular ad hoc networks. In *Proceedings of SASN'05*, Alexandria, VA, USA, November 2005.
- [3] D. Basin, S. Mödersheim, and L. Viganò. An On-The-Fly Model-Checker for Security Protocol Analysis. In E. Sneekenes and D. Gollmann, editors,

*Proceedings of the 8th European Symposium on Research in Computer Security (ESORICS 2003)*, volume 2808 of *Lecture Notes in Computer Science*, pages 253-270. Springer, 2003.

- [4] Haowen Chan, Virgil D. Gligor, Adrian Perrig, Gautam Muralidharan. On the Distribution and Revocation of Cryptographic Keys in Sensor Networks. *IEEE Transactions on Dependable and Secure Computing*, vol. 02, no. 3, pages 233-247, Jul-Sept, 2005.