Formal Analysis of Privacy for Vehicular Mix-Zones

Morten Dahl[1,2], Stéphanie Delaune[2], and Graham Steel[2]

[1] Department of Computer Science, Aalborg University
[2] LSV, ENS Cachan & CNRS & INRIA Saclay Île-de-France

Abstract. Safety critical applications for recently proposed vehicle to vehicle ad-hoc networks (VANETs) rely on a beacon signal, which poses a threat to privacy since it could allow a vehicle to be tracked. Mix-zones, where vehicles encrypt their transmissions and then change their identifiers, have been proposed as a solution to this problem.
In this work, we describe a formal analysis of mix-zones. We model a mix-zone and propose a formal definition of privacy for such a zone. We give a set of necessary conditions for any mix-zone protocol to preserve privacy. We analyse, using the tool ProVerif, a particular proposal for key distribution in mix-zones, the CMIX protocol. We show that in many scenarios it does not preserve privacy, and we propose a fix.

Keywords: Privacy, VANETs, Mix-Zones, Security Protocols

1 Introduction

Road traffic accidents are the most common cause of death in young adults in industrialized countries [13]. To improve road safety, a vehicle-to-vehicle communication platform is currently being developed by consortia of car manufacturers and legislators [15, 17]. Safety-related applications such as collision warning systems and high speed toll payment are envisaged. Dubbed vehicular ad-hoc networks (VANETs), the platform is based on decentralised mobile ad-hoc networks in order to retain scalability despite the high average speed of vehicles, and the large size of the network. As a consequence, the protocols used within the network are designed to use few steps, short messages, and not rely heavily on infrastructure for e.g. obtaining trust. To facilitate safety-critical applications there is a consensus that all vehicles must periodically broadcast a beacon message consisting of the vehicle's location (in the form of a GNSS coordinate), velocity, and identifier. Broadcasting this data several times per second raises privacy issues.

Fortunately, many of the envisioned applications, including collision avoidance, do not need a real-world identifier such as the vehicle's license plate, but can instead make do with a random identifier known as a pseudonym. However, long term tracking may still reveal the real-world identity of the driver. One can change pseudonym from time to time, but for this to have any effect the vehicles must change pseudonyms under the right circumstances. It seems preferable to change pseudonyms e.g. at intersections where several vehicles are close together and their paths unpredictable. This mimics the ubiquitous computing idea of a *mix-zone*, where beacon signals are turned off in a mixing area [3]. Vehicles cannot turn off beacon messages since many accidents happen at intersections, hence the idea is to have all vehicles encrypt their beacon signals when inside the zone [11].

Related Work. Several papers discuss the background to the VANET privacy problem and the merits of the pseudonymous authentication solution [10, 12, 14]. Previous analysis work aims to evaluate the effectiveness of (a larger network of) general mix-zones in terms of the probability of the attacker correctly linking two pseudonyms based on assumed prior known statistics about vehicles movement [6], when the effectiveness of each single mix-zone is already assumed. Privacy for mobile devices with RFID tags has recently been treated formally [2, 5, 18] . It is not clear how the definitions of privacy in these papers relate to each other, and even less so to our own definition. We, for instance, have to exclude scenarios where privacy is broken independently of the key establishment protocol and must moreover require synchronised behaviour of vehicles. These requirements for obtaining privacy are closer to the requirements made for electronic voting protocols [8].

Our contributions. In this paper, we investigate formally the effectiveness of vehicular mix-zone proposals. We model the network traffic inside a mix-zone, and examine under which conditions it is reasonable to expect any gain in privacy. We use the formal notion of indistinguishability to formalise the privacy property for a mix-zone. We analyse the CMIX protocol [11] that has been proposed to distribute keys to vehicles entering the mix-zone. We report some scenarios in which the use of the CMIX protocol can prevent privacy from being achieved. These scenarios have been discovered with the aid of the protocol analysis tool ProVerif [4]. We propose a fix to the protocol. We believe this is the first work to investigate the privacy property of an encrypted mix-zone, in particular when the key distribution protocol is also taken into account.

Paper outline. In the next section, we present the concept of a mix-zone and we give a description of the CMIX protocol. Then, we give our formal model (see Section 3) and we explain our formal definition of mix-zone privacy, which corresponds to an indistinguishability property (Section 4). In Section 5, we give our results obtained on mix-zones, first assuming an ideal key distribution protocol, and then using the CMIX protocol. Finally, we evaluate the protocol and our modelling approach, we propose a fix, and we give conclusions.

2 Mix-Zones and CMIX Protocol

This section describes mix-zones, and in particular the CMIX protocol used to distribute keys to vehicles entering a zone.

2.1 Mix-Zones

As discussed in the previous section, mix-zones are needed for the change of pseudonyms to have any effect in preserving privacy. However, changing pseudonyms while close to other vehicles is not sufficient to guarantee 'unlinkability', which we define informally as the property that an attacker cannot know that the old and new pseudonym belong to the same vehicle. To obtain this, pseudonyms must also be changed synchronously from the point of view of the attacker. More precisely, by synchronously, we mean that once one vehicle has started broadcasting using a new pseudonym then all future broadcasts heard by the attacker from at least one other vehicle must be using a new pseudonym as well.

If two vehicles in a mix-zone can agree on a precise point in time to change their pseudonyms (for instance by one of the vehicles broadcasting the time of when it is going to change its pseudonym) then synchronised change of pseudonym is sufficient for unlinkability. In practice however, for several reasons, one might want to allow a larger time interval for pseudonym change, e.g. to have a better chance that another vehicle is nearby to synchronise with, to ensure a certain level of unpredictability of trajectory, to account for clock differences, etc. Using a longer time interval has the undesirable effect of causing a radio-silence period during which none of the safety beacon messages can be broadcast. Encrypted mix-zones are suggested to remedy these short-comings: beacon messages can still be broadcast during the synchronisation time interval as long as the attacker cannot read them. A recent proposal argues that turing off radios at intersctions might be a worthwhile trade-off for privacy [7], but in this paper we concentrate on investigating what privacy can be achieved when beacon signals are required to be left on.

2.2 The CMIX Protocol

The CMIX protocol [11] distributes keys for encrypting beacon messages while in the mix-zone with the goal of preventing an attacker from linking the pseudonym of an in-coming vehicle with the pseudonym it uses when leaving. Every vehicle is equipped with a tamper-resistant device (TRD) allowing access to its contents only through its API. An offline Certification Authority (CA) run by a trusted third party is responsible for issuing certificates cryptographically binding a pseudonym P together with the public part ($\mathsf{pub}(K)$) of an asymmetric key K. Every vehicle has a fresh non-empty set of these key-pseudonym pairs stored in its TRD. One pair is marked as current, to be used when sending messages.

Vehicles entering the mix-zone (part A of Figure 1) are alerted of the presence of a road-side unit (RSU) by a radio broadcast. This triggers the vehicles to initiate a key establishment session.

1. $V \rightarrow RSU : \mathsf{sign}_{K_V}(\mathsf{request}, T_s),\ \mathsf{sign}_{K_{CA}}(P_V, \mathsf{pub}(K_V))$
2. $RSU \rightarrow V : \mathsf{aenc}_{\mathsf{pub}(K_V)}(\mathsf{sign}_{K_{RSU}}(P_V, zk, T_s)),\ \mathsf{sign}_{K_{CA}}(P_{RSU}, \mathsf{pub}(K_{RSU}))$
3. $V \rightarrow RSU : \mathsf{sign}_{K_V}(\mathsf{ack}, T_s),\ \mathsf{sign}_{K_{CA}}(P_V, \mathsf{pub}(K_V))$

The first message is a signed timestamp T_s together with the constant $\mathsf{request}$ used as a tag. The reply made by the RSU contains the zone encryption key zk

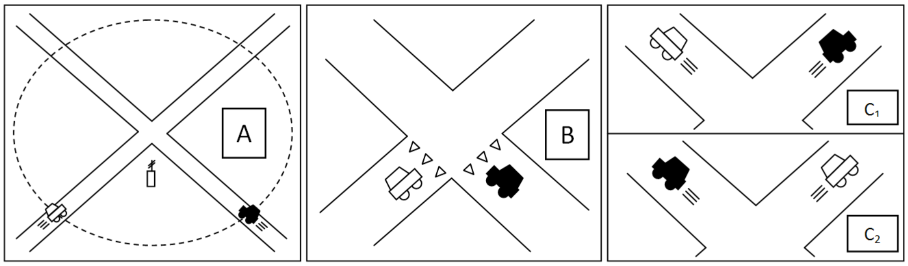

Fig. 1. Intended usage of encrypted mix-zones

encrypted under the public key $\mathsf{pub}(K_V)$ associated with the vehicle's current pseudonym P_V. The corresponding private key is assumed to be only known by the vehicle's on board tamper-resistant cryptographic device, which can decrypt the packet, store the zone key, and make available an encryption and decryption service using this key. In this way, the zone key remains unknown to everyone, including an attacker with a vehicle and a tamper-resistant device of his own. The last message is an acknowledgement sent by the vehicle. Every message is appended with the principal's current certificate.

The zone key is then used to encrypt and decrypt beacon messages while inside the geographical area dictated by the RSU. During their journey through the mix-zone, the vehicles will come in close enough proximity that the attacker is assumed unable to distinguish their locations (part B of Figure 1). Before leaving the mix-zone the vehicles change their pseudonyms leaving the attacker unable to determine if they leave according to part C_1 or part C_2 of Figure 1.

In the CMIX proposal [11], it is not specified whether a deterministic or probabilistic encryption scheme is used to encrypt beacon messages. Probabilistic encryption might seem the best solution, but due to the tight size constraints of messages in VANETs, it may be preferable to use a deterministic scheme. Deterministic schemes might still prevent the easy comparison of ciphertexts due to the rapidly changing content of beacon messages (such as the coordinate). Since this would depend on the exact cipher mode, beacon message format, etc, and this is not yet fixed [17], we consider both types of encryption scheme in our analysis.

A short informal analysis of the CMIX protocol is provided by Freudiger et al. [11]. The threat scenario they consider is unclear: they first state that their adversary is a passive outsider [11, §2.2] but then describe the resistance of the protocol to attacks where the adversary sends messages to try to impersonate an RSU [11, §3.2]. In general, VANET protocols are assumed to be required to withstand attack by active adversaries, as described e.g. by Raya and Hubaux (a subset of the authors of the CMIX paper) [14]. In this paper, therefore, we consider both the passive attacker and an active attacker that can forge and broadcast messages, but not prevent messages from being received. We will explain during our analysis under what assumptions particular attacks would be effective. Note that our adversary is assumed to have no visual contact as he would otherwise be able to track a vehicle using e.g. the license plate.

3 Formal Modelling

The process calculus of Blanchet *et al.* [4] used by the ProVerif tool is a variant of the applied pi calculus [1], a process calculus for formally modelling concurrent systems and their interactions. We recall the basic ideas and concepts of this calculus that are needed for our analysis.

3.1 Messages

To describe messages, we start with a set of *names* (which are used to name communication channels and other atomic data), a set of *variables*, $x, y, \ldots$ and a signature Σ formed by a finite set of *function symbols* each with an associated arity. Function symbols are distinguished by two categories: *constructors* and *destructors*. We use standard notation for function application, i.e. $f(M_1, \ldots, M_n)$. Constructors are used for building messages. Destructors represent primitives for taking messages apart and can visibly succeed or fail (while constructors always succeed). Messages $M, N, \ldots$ are obtained by repeated application of constructors on names and variables whereas a term evaluation D can also use destructors. The semantics of a destructor g of arity n is given by a set of rewrite rules of the form $g(M_1, \ldots, M_n) \rightarrow M_0$ where $M_0, \ldots, M_n$ are messages that only contains constructors and variables. Given a term evaluation D, we write $D \Downarrow M$ when D can be reduced to M by applying some destructor rules.

In the following, we consider constructors to model signatures and different kinds of encryptions (symmetric/asymmetric and deterministic/probabilistic). The symbol pub is a constructor representing the public key associated to the private key given in argument. The semantics of our destructors are given below:

$$\begin{array}{rr} \mathsf{checksign}(\mathsf{sign}(x,y),\mathsf{pub}(y)) \rightarrow x & \mathsf{sdec}(\mathsf{senc}(x,y),y) \rightarrow x \\ \mathsf{getmsg}(\mathsf{sign}(x,y)) \rightarrow x & \mathsf{rsdec}(\mathsf{rsenc}(x,y,z),y) \rightarrow x \\ & \mathsf{adec}(\mathsf{aenc}(x,\mathsf{pub}(y)),y) \rightarrow x \end{array}$$

We model probabilistic encryption by $\mathsf{rsenc}(m,k,r)$ where the r component is fresh for every encryption, thus preventing comparison. We model a signature scheme by two rewrite rules: the first one is used to verify a signature and the second one models the fact that the signature scheme is not message concealing.

3.2 Processes

Processes are built from the grammar described below, where N is a message, D is a term evaluation, a is a name, c is a channel name, and x a variable.

$P, Q, R ::=$	processes
0	null process
$P \mid Q$	parallel composition
$!P$	replication
$\text{new } a; P$	name restriction
$\text{let } N = D \text{ in } P \text{ else } Q$	term evaluation
$\text{in}(c, N); P$	message input
$\text{out}(c, N); P$	message output

The process "let $N = D$ in P else Q" tries to evaluate D; if this succeeds and if the resulting message matches the term N then the variables in N are bound and P is executed; if not then Q is executed. The rest of the syntax is quite standard. To ease the presentation, we will use tuples of messages, denoted by parentheses, while keeping the reduction rules for these tuples implicit. We will omit "else Q" when the process Q is 0.

An *evaluation context* is a context, that is a process with a hole, built from $[\_]$, $C \mid P$, $P \mid C$ and new $a; C$. We obtain $C[P]$ as the result of filling $C[\_]$'s hole with P. A process P is *closed* if all its variables are bound through an input or a let construction.

The RSU process. To illustrate the calculus used throughout this paper, we give below a description of the RSU part of the CMIX protocol. We follow the description given in the previous section. The RSU sends and receives all messages using some public channel c and holds a freshly generated zone key zk. We also model its pseudonym p_{rsu} and its private key k_{rsu} by fresh names. We assume that the RSU already knows its certificate $\mathsf{sign}((p_{rsu}, \mathsf{pub}(k_{rsu})), k_{ca})$. Below, we only model the reception of the first message with its decomposition. After some checks, the reply to the vehicle containing zk is constructed and sent. We do not model the reception of the acknowledgement.

$$\begin{aligned}\mathsf{RSU}_{\mathsf{CMIX}} \overset{\mathsf{def}}{=}\ & \text{in}(c, (x^s, x^c)); \\ & \text{let } (x_{pv}, x_{pkv}) = \mathsf{checksign}(x^c, \mathsf{pub}(k_{ca})) \text{ in} \\ & \text{let } (\mathsf{request}, x_T) = \mathsf{checksign}(x^s, x_{pkv}) \text{ in} \\ & \text{let } y^s = \mathsf{sign}((x_{pv}, zk, x_T), k_{rsu}) \text{ in} \\ & \text{let } y^c = \mathsf{sign}((p_{rsu}, \mathsf{pub}(k_{rsu})), k_{ca}) \text{ in} \\ & \text{out } (c, (\mathsf{aenc}(y^s, x_{pkv}), y^c)); \ldots\end{aligned}$$

The operational semantics of processes in the calculus of ProVerif , are essentially defined by two relations, namely *structural equivalence* $\equiv$ and *reduction* $\rightarrow$. We write $\rightarrow^*$ for the reflexive and transitive closure of $\rightarrow$. *Structural equivalence* is the smallest equivalence relation on processes that is closed under application of evaluation contexts and some other standard rules such as associativity and commutativity of the parallel operator. *Reduction* is the smallest relation closed under structural equivalence and application of evaluation contexts such that:

RED I/O	$\mathrm{out}(c, M).Q \mid \mathrm{in}(c, N).P \rightarrow Q \mid P\sigma$	
RED FUN 1	let $N = D$ in P else $Q \rightarrow P\sigma$	if $D \Downarrow M$
RED FUN 2	let $N = D$ in P else $Q \rightarrow Q$	if there is no M such that $D \Downarrow M$
REPL	$!P \rightarrow P \mid !P$	

where σ is the substitution defined on the variables that occur in N and such that $M = N\sigma$. In case such a substitution does not exist, the resulting process will be $Q \mid \mathrm{in}(c, N).P$ for RED I/O rule and Q for the RED FUN 1 rule.

3.3 Observational Equivalence

The notion of observational equivalence was introduced in [1]. We write $P{\downarrow_c}$ when P emits a message on the channel c, that is, when $P \equiv C[\mathrm{out}(c, M); Q]$ for some evaluation context C that does not bind c and some process Q.

Definition 1. Observational equivalence $\sim$ *is the largest symmetric relation* $\mathcal{R}$ *on closed processes such that* $P \; \mathcal{R} \; Q$ *implies:*

1. *if* $P{\downarrow_c}$ *then* $Q{\downarrow_c}$*;*
2. *if* $P \rightarrow P'$ *then there exists* Q' *such that* $Q \rightarrow^* Q'$ *and* $P' \; \mathcal{R} \; Q'$*;*
3. $C[P] \; \mathcal{R} \; C[Q]$ *for all evaluation contexts* C*.*

Intuitively, a context may represent an attacker, and two processes are observationally equivalent if they cannot be distinguished by any attacker. Note that such an attacker is too powerful for our purpose since the nature of broadcast communication does not allow him to block all messages. When performing the analysis we will exclude attacks that are not possible for our attacker; as we will see, the attacks we find do not rely on the attacker blocking messages.

ProVerif is not able to check observational equivalence directly but actually checks a stronger notion that implies observational equivalence [4]. However, this notion is too strong in many situations. This problem has recently been studied in and a method has been proposed to extend the class of equivalences which ProVerif is able to verify [9]. We will use this method to overcome the limitations of ProVerif and to automatically verify the equivalences allowing us to model our privacy property.

4 Privacy for Vehicular Mix-Zones

In this section we show how the privacy property informally described in Section 2 can be formalised in our setting. We build on the classical approach of formalising privacy properties as some kind of observational equivalence in a process algebra or calculus [8, 16], and extend this to take into consideration mix-zones and vehicle mobility.

4.1 Mix-Zones

In the previous sections we have informally used the term mix-zone to describe a place suitable for vehicles to change their pseudonym by being able to mix or hide among each other. We formally define a mix-zone as consisting of five locations entry_L, entry_R, *proximity*, exit_L, and exit_R. We use public channels to model these locations. If two messages are emitted on different channels, then our attacker will be able to see a difference. This corresponds to the fact that he is able to tell that they were transmitted from geographically different locations. Note that messages sent on a public channel can be received on another public channel with the help of our active attacker. Vehicles enter the mix-zone by one of the entry locations and exit by one of the exit locations. The *proximity* location models a stretch within the mix-zone where vehicles are so close to each other that our attacker cannot tell them apart geographically.

Beacon messages are defined as consisting only of a pseudonym p_v modelled by a fresh name. This pseudonym is signed using the vehicle's current key k_v and appended with the CA signed certificate binding the pseudonym together with the public part of k_v. Formally a beacon message is defined as $\Big(\mathsf{sign}(p_v, k_v),\ \mathsf{sign}((p_v, \mathsf{pub}(k_v)), k_{ca})\Big)$ where k_{ca} is the private key of the CA. Note that all the location data in beacon messages are modelled by the channel on which they are sent.

4.2 Privacy

The formal privacy property aims to capture the fact that an attacker cannot track a vehicle. We assume that the attacker can listen on the entire network and hence on all public channels. Thus, in order to achieve privacy, we need to suppose the presence of at least two vehicles.

We consider a single mix-zone with two vehicles V_A and V_B, as in Figure 1. V_A will always start in entry_L and V_B always in entry_R. Going through the mix-zone, each vehicle emits a series of beacon messages. They can do this in two different ways:

1. The vehicle V_A moves from entry_L to *proximity* to exit_L while V_B moves from entry_R to *proximity* to exit_R (as in part C_1 of Figure 1).
2. The vehicle V_A moves from entry_L to *proximity* to exit_R while V_B moves from entry_R to *proximity* to exit_L (as in part C_2 of Figure 1).

Intuitively, we achieve privacy if an attacker cannot tell the two cases apart. Formally, let $\mathsf{V}(\mathit{entry}, \mathit{exit})$ stand for the vehicle that moves from *entry* to *proximity* to *exit*. Privacy holds if the following equivalence holds:

$$C\Big[\mathsf{V}(\mathit{entry}_L, \mathit{exit}_L) \mid \mathsf{V}(\mathit{entry}_R, \mathit{exit}_R)\Big] \sim C\Big[\mathsf{V}(\mathit{entry}_L, \mathit{exit}_R) \mid \mathsf{V}(\mathit{entry}_R, \mathit{exit}_L)\Big].$$

The next section presents the analysis we have performed, including the definition of the vehicles processes, and also the C contexts with which the analysis has been performed.

5 Privacy Analysis

The analysis is performed in two models: an ideal model where the vehicles are assumed to know the mix-zone encryption key and a CMIX model where this key is distributed using the CMIX protocol. From our ideal model analysis, we extract a set of scenarios where it is possible for a 'perfect' key distribution protocol to guarantee privacy. We then evaluate the CMIX protocol with respect to these scenarios.

5.1 Privacy in the Ideal Model

In the ideal model the vehicles magically know the mix-zone encryption key, the attacker does not know it, and the only communications are the beacon messages. As discussed in previous sections, we consider both deterministic and probabilistic encryption of beacon messages.

Experimental Analysis. We model each vehicle using a fixed sequence of beacon message emissions $\overline{p_v^1}$; $\{\overline{p_v^1}\}_{zk}$; $\{\overline{p_v^2}\}_{zk}$; $\overline{p_v^2}$ where:

- $\overline{p_v^i} \overset{\mathsf{def}}{=} \mathsf{sign}(p_v^i, k_v^i),\ \mathsf{sign}((p_v^i, \mathsf{pub}(k_v^i)), k_{ca})$, and
- $\{\overline{p_v^i}\}_{zk} \overset{\mathsf{def}}{=} \mathsf{senc}(\overline{p_v^i}, zk)$ or $\mathsf{rsenc}(\overline{p_v^i}, zk, r)$ depending on whether we are considering respectively deterministic or probabilistic encryption. In this last case, each occurrence of r represents a fresh nonce.

From this fixed sequence we generate a set of relevant scenarios by adding two changes of location, from *entry* to *proximity* and from *proximity* to exit, and we perform a geographical synchronisation either coming into or going out of the *proximity* location. We allow each vehicle to emit each beacon three times, so it is possible to change locations at any position in the sequence. The first $\overline{p_v^1}$ is always emitted at an *entry* location and the last $\overline{p_v^2}$ is always emitted at an *exit* location. We then investigate whether we can prove privacy if two vehicles in the mix-zone conform to this pattern.

We write each scenario as a process. For instance, the scenario where all $\overline{p_v^1}$ beacon messages are emitted at the *entry* location, the $\{\overline{p_v^1}\}_{zk}$ spread out over *entry* and *proximity*, the $\{\overline{p_v^2}\}_{zk}$ over *proximity* and *exit*, and the $\overline{p_v^2}$ at *exit* with deterministic encryption and synchronisation before leaving the *proximity* location, is represented by:

$$\begin{array}{rl}\mathsf{Vehicle}(\mathit{entry}, \mathit{exit}) \overset{\mathsf{def}}{=} & \mathrm{new}\ p_v^1;\ \mathrm{new}\ k_v^1;\ \mathrm{new}\ p_v^2;\ \mathrm{new}\ k_v^2;\ \mathrm{out}(\mathit{entry}, \overline{p_v^1}); \\ & \textit{(* key establishment *)} \\ & \mathrm{out}(\mathit{entry}, \{\overline{p_v^1}\}_{zk}); \\ & \mathrm{out}(\mathit{proximity}, \{\overline{p_v^1}\}_{zk});\ \mathrm{out}(\mathit{proximity}, \{\overline{p_v^2}\}_{zk}); \\ & \textit{(* geographical synchronisation *)} \\ & \mathrm{out}(\mathit{exit}, \{\overline{p_v^2}\}_{zk});\ \mathrm{out}(\mathit{exit}, \overline{p_v^2})\end{array}$$

where for sake of clarity we have removed duplicate instructions. The *(* key establishment *)* marker is left empty since we consider an ideal model where the vehicles magically know the mix-zone encryption key. The *(* geographical synchronisation *)* marker indicates that the two vehicles will have to synchronise at this point. In other words, a vehicle can execute the instructions after this point only once all the instructions before this point have been executed by both vehicles.

Having turned the scenario into a process, we instantiate this process twice using different values for *entry* and *exit* to obtain the two Vehicle processes needed for the equivalence checking. We consider the context

$$C_{\mathsf{ideal}} = \mathsf{new}\ k_{ca}; \mathrm{out}(c, \mathsf{pub}(k_{ca})); \mathsf{new}\ zk; \_.$$

and ask ProVerif to try to prove observational equivalence. To overcome the limitations due to the ProVerif tool, we perform *data swapping* as described in [9].

From previous discussions it is clear that geographical synchronisation is a necessary condition for privacy, i.e. that two vehicles either enter or exit the mix-zone at the same time. More precisely, the necessary condition is that no message is sent from an *entry* location after a message has been sent from an *exit* location. If this is not satisfied then the attacker can trivially link $\overline{p_v^1}$ with $\overline{p_v^2}$, so we did not include any such scenarios in our experiments.

Results. All the scenarios we consider are listed in Figure 2 along with the obtained results. Each row is a scenario with the first columns showing where the beacon messages in the sequence are emitted. The columns to the right of the sequence show the results in the different encryption models: the first two give the results for when deterministic encryption is used and the last two for when probabilistic encryption is used. In each encryption model, the left column shows the result if the vehicles synchronise before going into the *proximity* location and the right column if they synchronise before leaving. A minus symbol ($-$) indicates that ProVerif could not prove equivalence (and found an attack trace) and a plus symbol ($+$) means that it could.

Analysis. Our results show a second necessary condition for privacy: that vehicles do not change pseudonym too early or too late. This is shown by Scenario 1 and 2 where the vehicles are still sending unencrypted beacon messages using the first pseudonym at the exit location. Similarly, Scenario 31 and 32 show that privacy is lost if they move too late; in this case the second pseudonym is used in an unencrypted beacon message at the entry location.

In the deterministic encryption model, we only have privacy in scenarios where geographical synchronisation coincides with a change of message. This condition is illustrated by Scenarios 10-14. In this group, ProVerif can prove privacy if the synchronisation is before the *proximity* location since the link between $\overline{p_v^1}$ and $\{\overline{p_v^1}\}_{zk}$ is broken. However, in Scenarios 15-20 we see from ProVerif's

Scenario	Message Sequence												Ideal (determ)		Ideal (prob)	
	pv1	pv1	pv1	{pv1}zk	{pv1}zk	{pv1}zk	{pv2}zk	{pv2}zk	{pv2}zk	pv2	pv2	pv2	before	after	before	after
1	▒	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	-	-	-	-
2	▒	░	▓	▓	▓	▓	▓	▓	▓	▓	▓	▓	-	-	-	-
3	▒	░	░	▓	▓	▓	▓	▓	▓	▓	▓	▓	-	+	-	+
4	▒	░	░	░	▓	▓	▓	▓	▓	▓	▓	▓	-	-	-	+
5	▒	░	░	░	░	░	▓	▓	▓	▓	▓	▓	-	+	-	+
6	▒	░	░	░	░	░	░	▓	▓	▓	▓	▓	-	-	-	+
7	▒	░	░	░	░	░	░	░	░	▓	▓	▓	-	+	-	+
8	▒	░	░	░	░	░	░	░	░	░	▓	▓	-	-	-	-
9	▒	▒	▒	▓	▓	▓	▓	▓	▓	▓	▓	▓	+	+	+	+
10	▒	▒	▒	░	▓	▓	▓	▓	▓	▓	▓	▓	+	-	+	+
11	▒	▒	▒	░	░	░	▓	▓	▓	▓	▓	▓	+	+	+	+
12	▒	▒	▒	░	░	░	░	▓	▓	▓	▓	▓	+	-	+	+
13	▒	▒	▒	░	░	░	░	░	░	▓	▓	▓	+	+	+	+
14	▒	▒	▒	░	░	░	░	░	░	░	▓	▓	+	-	+	-
15	▒	▒	▒	▒	▓	▓	▓	▓	▓	▓	▓	▓	-	-	+	+
16	▒	▒	▒	▒	░	▓	▓	▓	▓	▓	▓	▓	-	-	+	+
17	▒	▒	▒	▒	░	░	▓	▓	▓	▓	▓	▓	-	+	+	+
18	▒	▒	▒	▒	░	░	░	▓	▓	▓	▓	▓	-	-	+	+
19	▒	▒	▒	▒	░	░	░	░	░	▓	▓	▓	-	+	+	+
20	▒	▒	▒	▒	░	░	░	░	░	░	▓	▓	-	-	+	-
21	▒	▒	▒	▒	▒	▒	▓	▓	▓	▓	▓	▓	+	+	+	+
22	▒	▒	▒	▒	▒	▒	░	▓	▓	▓	▓	▓	+	-	+	+
23	▒	▒	▒	▒	▒	▒	░	░	░	▓	▓	▓	+	+	+	+
24	▒	▒	▒	▒	▒	▒	░	░	░	░	▓	▓	+	-	+	-
25	▒	▒	▒	▒	▒	▒	▒	▓	▓	▓	▓	▓	-	-	+	+
26	▒	▒	▒	▒	▒	▒	▒	░	▓	▓	▓	▓	-	-	+	+
27	▒	▒	▒	▒	▒	▒	▒	░	░	▓	▓	▓	-	+	+	+
28	▒	▒	▒	▒	▒	▒	▒	░	░	░	▓	▓	-	-	+	-
29	▒	▒	▒	▒	▒	▒	▒	▒	▒	▓	▓	▓	+	+	+	+
30	▒	▒	▒	▒	▒	▒	▒	▒	▒	░	▓	▓	+	-	+	-
31	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▓	▓	-	-	-	-
32	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	░	▓	-	-	-	-

Key: ▒ entry ░ proxim ▓ exit

Fig. 2. Result of the analysis in the ideal model

counterexamples that when synchronisation is before the *proximity* location, the attacker can link $\overline{p_v^1}$ and $\{\overline{p_v^1}\}_{zk}$ since they are both emitted at the same *entry* location. After the synchronisation one vehicle can move to an *exit* location and emit $\overline{p_v^2}$ while the other is still at *proximity* and emitting $\{\overline{p_v^1}\}_{zk}$. By comparing ciphertexts the attacker will know which vehicle has "fallen behind" and which vehicle is at the *exit* location, in turn allowing him to link $\overline{p_v^1}$ and $\overline{p_v^2}$.

The situation changes when probabilistic encryption is used. In this case we have that ProVerif can prove equivalence for all the cases where deterministic encryption allows privacy, and in addition, scenarios where the geographical synchronisation is between two encrypted messages, e.g. Scenarios 4, 6, 15, 16. This is an important result, since it means that two vehicle only need to get into a mix zone and encrypt at the same time as another vehicle, then change the pseudonym before leaving. It seems clear that an encryption scheme that renders encrypted beacons incomparable must be used.

As a final remark we note that the results show that in our model, use of encryption is not necessary to obtain privacy: if the vehicles agree on when to change their pseudonym then no encryption is needed. This is best illustrated in Scenario 21. Although encryption is used, it has no effect since beacon messages can be trivially linked with their encryption by the location where they are emitted. Furthermore, no messages are emitted at the *proximity* location.

5.2 Privacy in the CMIX Model

Based on the conclusions of the previous section, we consider only probabilistic encryption when analysing the CMIX key distribution protocol. We consider all scenarios where privacy is provable in the ideal model. First, we add one session of the CMIX protocol to both vehicle processes, to be executed before entering the *proximity* zone. We found that in all cases where privacy was possible in the ideal model, it was also possible here[3].

We recall that according to the CMIX paper [11], a key request message is triggered in the vehicle when it either receives a message that it cannot decrypt, or when it receives an alert message from the RSU. The former situation could be used by an active attacker to trigger a second CMIX session (after the first was finished). The nearby presence of other mix-zones or simply a corrupted broadcast might also trigger a second session in the presence of a passive attacker. Hence we consider all variations of the scenarios obtained by interleaving two sequential sessions of the key establishment protocol. One session is always at the *entry* location using the first pseudonym and before emitting any encrypted beacon messages, but the location of the second session is varied between *proximity* and *exit*, and further by which of the pseudonyms it uses.

To illustrate the modelling of subscenarios we consider the variation of the scenario from the previous subsection obtained by placing the second key establishment session at the *exit* location after changing pseudonym. The process

[3] Full results can be found online at `http://www.cs.aau.dk/~dahl/mixzoneprivacy/`

```
$\mathsf{Vehicle}_{\mathsf{CMIX}}(c, p_v, k_v) \overset{\mathsf{def}}{=}$  new $t_s$;
    let $x^s = \mathsf{sign}((\mathsf{request}, t_s), k_v)$ in
    let $x^c = \mathsf{sign}((p_v, \mathsf{pub}(k_v)), k_{ca})$ in
    $\mathsf{out}(c, (x^s, x^c))$;
        $\mathsf{in}(c, (y^e, y^c))$;
        let $(x_{prsu}, x_{pkrsu}) = \mathsf{checksign}(y^c, \mathsf{pub}(k_{ca}))$ in
        let $y^s = \mathsf{adec}(y^e, k_v)$ in
        let $(p_v, x_{zk}, t_s) = \mathsf{checksign}(y^s, x_{pkrsu})$ in
        let $z^s = \mathsf{sign}((\mathsf{ack}, t_s), k_v)$ in
        let $z^c = \mathsf{sign}((p_v, \mathsf{pub}(k_v)), k_{ca})$ in
        $\mathsf{out}\ (c, (z^s, z^c))$
```

Fig. 3. Vehicle's part of CMIX key establishment protocol

for this subscenario is similar to the vehicle process given in Section 5.1 expect that $\mathsf{Vehicle}_{\mathsf{CMIX}}(entry, p_v^1, k_v^1)$ defined in Figure 3 replaces the marker *(* key establishment *)* and $\mathsf{Vehicle}_{\mathsf{CMIX}}(exit, p_v^2, k_v^2)$ is inserted just after the *(* geographical synchronisation *)* marker. Note that to make the analysis practical the operations of the TRD are inlined.

For the analysis, we place the two instantiated vehicle processes in the context given by:

$$C_{\mathsf{CMIX}} \overset{\mathsf{def}}{=} \text{new } k_{ca}; \text{out}(c, \mathsf{pub}(k_{ca})); \\ \text{new } k_{rsu}; \text{new } p_{rsu}; \text{out}(c, (p_{rsu}, \mathsf{pub}(k_{rsu}))); \text{new } zk; \big(!\mathsf{RSU}_{\mathsf{CMIX}} \mid \_\big).$$

which, contrary to the context used in the ideal model, includes the RSU.

Results. The experiments show that the CMIX key establishment protocol as described in the paper can break privacy in scenarios where it is assured in the ideal model. The reason is that the pseudonym is sent in clear in the request message. More precisely, the experiments show that if a key establishment session is triggered at the *exit* location then there is an attack when the vehicle has not yet changed its pseudonym: the key establishment session reveals the first pseudonym which can be link to the second pseudonym by the location. Perhaps less obviously, if a key establishment session is triggered at the *proximity* location then there is also an attack when the geographical synchronisation does not separate it from the unencrypted beacon messages sent using the other pseudonym. This attack is an instance of the general "fallen behind" attack that arises when both pseudonyms are revealed in locations not separated by a geographical synchronisation.

Contrary to the analysis in the ideal model, where the running time of ProVerif on a 2.5 GHz Intel Xero processor was less than a few minutes for each variation, the running time in the CMIX model ranged between a few seconds and 3 hours for each scenario.

5.3 Fixing the Key Establishment Protocol

A simple fix to the CMIX key establishment protocol that does not increase the number of rounds is to encrypt the request and the acknowledgement message under the RSU's public key. This assumes vehicles know the certificate of the RSU before performing a key request, which could be ensured by, for instance, including the certificate in the messages broadcast from the RSU to inform vehicles about the mix-zone.

We modelled this revised protocol in ProVerif and retried all the scenarios. For most of them ProVerif was able to prove privacy in the CMIX model when there was privacy in the ideal model, but in a fraction of the scenarios (1/13) a false attack was reported. The false attack seems to be due to the stronger equivalence that ProVerif tried to prove, and arises when two key establishment sessions using the same pseudonyms are separated by a geographical synchronisation. By recording the RSU's response in the first session with the vehicle using key kv and replaying this message to a vehicle during the second session, the vehicle not using kv will fail at decryption whereas the vehicle using kv will correctly decrypt but fail at a different step in the process, namely when comparing time stamps. The observations are the same, but the processes execute differently, so ProVerif is unable to prove equivalence.

6 Conclusion

In this paper, we have proposed a formal notion of privacy for mix-zones based on classical ideas of equivalence: if the equivalence is satisfied then no attacker can link the pseudonyms used by two vehicles entering a mix-zone with the pseudonyms they use when exiting. We have seen that for an idealised vehicular mix-zone to achieve privacy requires geographical and pseudonym change synchronisation. Our experiments on a variety of scenarios suggest that probabilistic encryption gives a significantly better chance of achieving privacy than deterministic encryption. We have analysed the CMIX proposal for key distribution in mix-zones, and shown that the use of the protocol can inadvertently prevent privacy from being achieved in many scenarios. We have shown that the CMIX protocol can be modified to preserve privacy.

As future work it seems natural to examine to what extent our experiments on a fixed series of beacon signals identical for both vehicles captures the space of possible scenarios satisfactorily. Although some cases of vehicles performing different scenarios are captured by our experiments, the case where one vehicle changes pseudonym at the entry location while the other changes at the exit location is for instance not captured. Another limitation of our modelling is that the messages of a key establishment session cannot be emitted across several locations. If the attacker can identify to which session messages belong then a session spanning across a geograpical synchronisation might break privacy, even against a passive attacker. Capturing this type of attack is also left for future work.

We plan to examine the API of the on board tamper-resistant cryptographic device to see how it might prevent insider attacks, i.e. attacks by an adversary who owns a legitimate vehicle. We also plan to investigate more fully the properties of our modelling approach, by e.g. comparing our notion of privacy to existing notions of anonymity, untraceability and unlinkability in the literature.

References

1. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *Proc. 28th ACM Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115, New York, USA, 2001. ACM Press.
2. M. Arapinis, T. Chothia, E. Ritter, and M. Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *Proc. 23rd IEEE Computer Security Foundations Symposium (CSF'10)*, 2010. To appear.
3. A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.
4. B. Blanchet, M. Abadi, and C. Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, 75(1):3–51, 2008.
5. M. Brusó, K. Chatzikokolakis, and J. den Hartog. Formal verification of privacy for RFID systems. In *Proc. 23rd IEEE Computer Security Foundations Symposium (CSF'10)*, 2010. To appear.
6. L. Buttyán, T. Holczer, and I. Vajda. On the effectiveness of changing pseudonyms to provide location privacy in VANETs. In *Pro. 4th European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS 2007)*, pages 129–141, Cambridge, 2007.
7. L. Buttyn, T. Holczer, A. Weimerskirch, and W. Whyte. Slow: A practical pseudonym changing scheme for location privacy in vanets. In *IEEE Vehicular Networking Conference (VNC)*, pages 1–8, Tokyo, Japan, October 2009.
8. S. Delaune, S. Kremer, and M. D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, July 2009.
9. S. Delaune, M. D. Ryan, and B. Smyth. Automatic verification of privacy properties in the applied pi-calculus. In Y. Karabulut, J. Mitchell, P. Herrmann, and C. D. Jensen, editors, *Proc. 2nd Joint iTrust and PST Conferences on Privacy, Trust Management and Security (IFIPTM'08)*, volume 263 of *IFIP Conference Proceedings*, pages 263–278, Trondheim, Norway, June 2008. Springer.
10. F. Doetzer. Privacy issues in vehicular ad hoc networks. In *Workshop on Privacy Enhancing Technologies*, number 3856 in LNCS, pages 197–209, Cavtat, Croatia, May 2005.
11. J. Freudiger, M. Raya, M. Flegyhzi, P. Papadimitratos, and J.-P. Hubaux. Mix-zones for location privacy in vehicular networks. In *Proc. of ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS'07)*, 2007.
12. B. Parno and A. Perrig. Challenges in securing vehicular networks. In *Proc. 4th Workshop on Hot Topics in Networks*, November 2005.
13. S. D. Peden M, Scurfield R. World report on traffic injury prevention. World Health Organization Report, 2004.
14. M. Raya and J.-P. Hubaux. The Security of Vehicular Ad Hoc Networks. In *Proc. 3rd ACM workshop on Security of ad hoc and sensor networks (SASN'05)*, pages 11–21, 2005.

15. Safespot project. http://www.safespot-eu.org/, 2006-2010.
16. S. Schneider and A. Sidiropoulos. CSP and anonymity. In *Proc. 4th European Symposium On Research In Computer Security (ESORICS'96)*, volume 1146 of *LNCS*, pages 198–218. Springer, 1996.
17. I. Standard. IEEE standard. IEEE Trial-Use Standard for Wireless Access in Vehicular Environments – Security Services for Applications and Management Messages, Approved 8 June 2006.
18. T. van Deursen, S. Mauw, and S. Radomirovic. Untraceability of RFID protocols. In *Proc. Workshop on Information Security Theory and Practices (WISTP'08)*, volume 5019 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2008.