# Privacy Enhanced Fraud Resistant Road Pricing

Jaap-Henk Hoepman, George Huitema

# Privacy Enhanced Fraud Resistant Road Pricing

Jaap-Henk Hoepman[1,2] and George Huitema[1,3]

[1] TNO Information and Communication Technology, Eemsgolaan 3,
9701 BK, Groningen, The Netherlands
[2] Institute for Computing and Information Sciences,
Radboud University Nijmegen, Nijmegen, the Netherlands
[3] Faculty of Economics and Business,
University of Groningen, the Netherlands
jaap-henk.hoepman@tno.nl, jhh@cs.ru.nl; george.huitema@tno.nl, g.b.huitema@rug.nl

**Abstract.** A naive implementation of a road pricing system will collect an enormous amount of personal location data. In this paper we present a sophisticated system that is privacy friendly, i.e. where the invoices contain aggregated pricing information and where only the driver has insight into all the underlying details. Enforcement based on spot checking is used to keep drivers honest, and to make the system fraud resistant. These spot checks are integrated in the system in a novel way that does not impair the privacy of the overall system, as is the case in systems previously proposed. Our solution can easily be applied to other privacy sensitive contexts such as smart metering and e-ticketing in public transport.

**Keywords:** Privacy, Road Pricing, Accountability, Security.

## 1 Introduction

For many years, many European governments have been discussing the introduction of some type of road pricing in order to reduce both the number of kilometres driven (thereby avoiding traffic jams and congestion), as well as carbon emissions. Road pricing can be achieved in many different ways depending on the size of the area covered and the level of road usage details that need to be collected. Here we focus on the full mode case where a nationwide scheme is applied and detailed road usage details, such as precise locations and time of day, are registered by means of a navigation satellite system together with a tracking device in the car, also called an On Board Unit (OBU). As an example, in the last category the Dutch government recently proposed an overall kilometre charging system for drivers as a replacement for the current road tax. This announcement by the Dutch authorities stirred a large public debate that still continues and mainly focuses on the costs for driving and privacy. This case inspired us to come up with an improved privacy friendly road pricing system.

In the full mode case where detailed usage information has to be gathered in order to compose a correct invoice for vehicle owners, there seems to be a conflict between privacy friendliness on the one hand (not everybody should have access to all the details) and on the other, the necessity of enforcement where details are necessary to

combat fraud. In this paper we present a balanced solution that is both privacy friendly and fraud resistant. The first is assured by distributing the necessary information over all the parties involved in the process of road pricing, so that no party can combine personal data with location data, while fraud is avoided by integrating random spot checks in a novel way.

The rest of this paper is organised as follows. In Section 2, we describe the assumptions and the requirements that will form the basics of our proposed road pricing system. In Section 3 related work is explained. The design principles we use are listed in Section 4 followed by Section 5 describing the architecture of the proposed system. In Section 6 the level of security and privacy of the proposed solution is analysed, followed by Section 7 concluding the paper together with some issues for further research.

## 2 Assumptions and Requirements

We make the following assumptions for our proposed road pricing system. *Business parties*: besides vehicle owners and vehicle drivers, business parties related to the back office processes - aggregation and pricing of usage details – will also be present. *Personal responsibility*: drivers will be held personally responsible for the generation of correct road usage details. *Enforcement*: an enforcement party tries to detect fraud and malfunctioning by spot checking. The spot checks will be secret, but not for long (cf. [9]). *Pricing function*: the so-called pricing function that assigns the charge for the distance driven is linear in its argument, (cf. [11]). *OBU setup*: the OBU will be 'thin', a relatively simple device that does not need to store complex maps. It will submit its registered location data in near real time to related road pricing parties. The OBU is not required to be a trusted element.

Furthermore we set the following requirements. *Correctness*: the road pricing system generates invoices for vehicles based on actual road usage. *Privacy friendliness*: only the driver has access to the detailed road usage information. Other business parties only see partial or aggregated information that does not reveal the full picture. *Fraud resistance*: enforcement will apply checks at random spots and time frames to ensure that correct usage details have been generated at the OBU.

## 3 Related Work

Road pricing systems are an example of the use of smart vehicle technology [8] with potential privacy ramifications [7]. Several proposals for cryptographic protocols based on generic secure multi-party computations have been proposed [3, 13]. According to Popa [12] these systems run several orders of magnitude slower than necessary, and scale poorly with increasing number of users. Therefore they propose a simpler scheme, where *thin* OBUs submit their location, labelled with a pseudonymous tag, to a server at regular time intervals. Using a zero-knowledge protocol, the vehicle owner collects the charges corresponding to their vehicle, and proves to the server they collected the right ones. They also propose an enforcement

protocol by random and secret spot checks. All vehicles spotted are reported to the server, and the enforcement essentially requires all spotted vehicles to prove that they submitted a record for the spot check location. This presents a privacy leak though: the protocol allows a server to query a vehicle whether it was present at a certain place at a certain time while the vehicle has no way to determine the validity of this query. The same privacy leak is present in the work of de Jonge and Jacobs [9] (see also [6]) and the PrETP system [2]. In these solutions a *thick* OBU is capable of computing the road charges based on repeated sampling of its current location. Vehicles have to commit to the trips they made during a day. Each recorded vehicle is asked by a server to reveal the pre-image of the committed hash value corresponding to the spot check location. If a vehicle cannot comply, this indicates a possible case of fraud. Again, a vehicle has no way to verify the validity of the server request.

Related to this are systems for pay-as-you-drive insurance, for example, the PriPAYD system [15]. In this system privacy is protected because the OBU in the vehicle locally computes the aggregated insurance premium based on sampled GPS locations. The OBU needs to be trusted and tamper proof though to ensure correct computation of the premium and to prevent fraud by users.

In summary, current proposals for a full-mode road pricing system do not fully guarantee privacy and accountability while relying on a thin, untrusted, OBU only.

## 4 Design Principles for Achieving Privacy in Road Pricing

In our proposal, privacy is achieved using the following two design principles. First of all, we use a novel way of splitting up *trips* (the trajectory followed by a car from its place of departure to its destination) into short segments, so-called *legs*, that are not linkable to each other. Secondly, the different process steps needed to determine the overall road charge are distributed over several system components. This way, no single component has enough information to reconstruct a particular route travelled.

### 4.1 Trips and Legs

Since a pricing function is assumed to be linear, trips can be split into shorter legs, and the charge for the trip equals the sum of the charges for the legs (cf. [11]). Therefore if we ensure that legs are not linkable (it cannot be determined whether two legs are part of the same trip), this approach is a first step to achieving privacy.

Traditionally, road pricing systems charge for the distance travelled on a particular road. The problem with this approach is that when splitting a trip into legs, one needs to make the last location in the previous leg equal to the location in the next leg, in order not to lose information needed to compute the charge. In order to keep legs

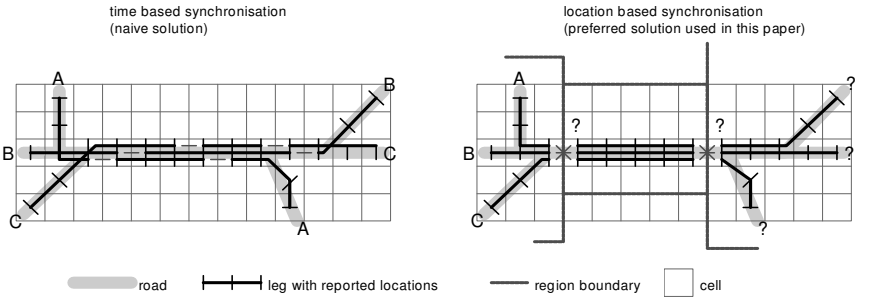


**Fig. 1**. Dividing a trip of a vehicle into legs can be done time-based (left) or location-based (right).

non-linkable then, locations (and their associated visiting times) need to be rounded rather coarsely, and even then the same combination of location and time will not occur that often.

To overcome this problem we propose to charge for road use slightly differently.; namely, to charge for travelling in a certain area instead. To this end, the ground surface is divided into cells (for instance using a so called Voronoi diagram [1], but other methods can also be used). The location of a vehicle is recorded as the 'centre' point of such a cell. In urban areas these cells may be small, to differentiate ring roads, congested roads, etc. A trip then, is defined as a sequence of *different* but consecutive cells (together with the time the cell is entered). A leg consists of a subsequence of this sequence, and the pricing function assigns a charge to each cell (depending on the time the cell was visited). With this setup, the first cell in a leg does not have to equal the last cell in the previous leg.

By splitting trips in short non-linkable legs, the point of entry of a particular car that enters a busy highway cannot be linked to its point of exit[1]. However, the effectiveness of this 'hiding in the crowd' depends on the way trips are subdivided in legs (see Fig. 1). Synchronisation on time, i.e., starting a leg at globally predefined times, is not a good idea. In such a setup the last location of the previous leg is correlated with the first location of the next leg. Therefore it is better to synchronise on location: whenever you cross the boundary region (a collection of cells), you start a new leg.

Privacy is improved by increasing the size of a cell, and by reporting visiting times in multiples of a sufficiently high step value (e.g., several seconds to a few minutes). Ideally, cell sizes should be chosen such that a relatively large amount of cars will be reported within such a cell within every fixed time interval (at least during normal traffic conditions). On the other hand, if cells are big, short trips within one cell essentially consist of one leg, reducing the privacy. For the purpose of this paper we simply note that the division of trips into legs increases the privacy of the average user considerably.

[1] Clearly a single car on a quiet country road can be followed. But even in this case the system guarantees that the identity of the car is not revealed.

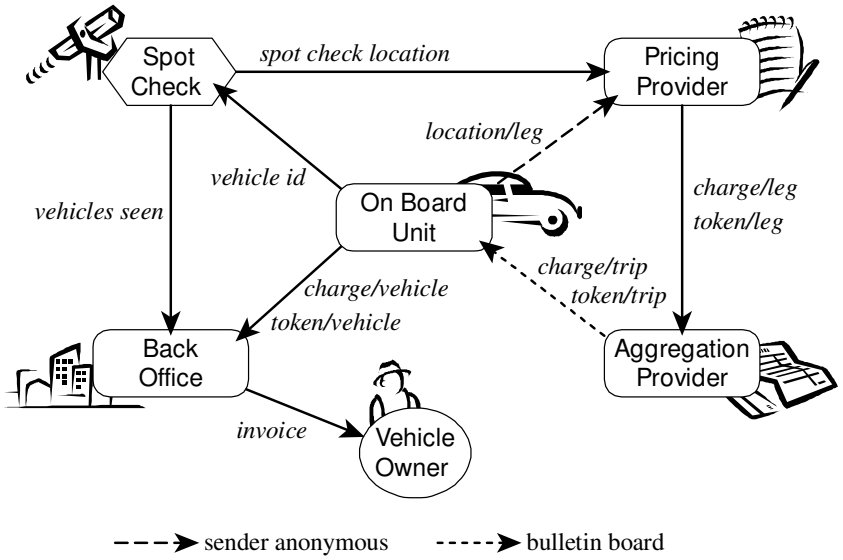


**Fig. 2.** Road pricing system components with related information flows.

### 4.2 System Components

We distinguish the following components in the system (see also Figure 2):

- **On Board Unit (OBU)** Equipment installed in each vehicle. Contains all the hardware and software needed to interface the vehicle with the road pricing system. Contains a GNSN (Global Navigation Satellite System) receiver, or an external connection to it.
- **Pricing Provider (PP)** A service provider that computes the charge for a particular leg (of a trip). Privacy is assured because the PP cannot determine whether two legs belong to the same trip or not.
- **Aggregation Provider (AP)** A service provider that aggregates the leg charges to compute the total trip charge. Privacy is assured because the AP does not see the trajectory (cells visited) that corresponds to a particular leg.
- **Back Office (BO)** The road pricing back office receives the submitted trip reports from the OBUs, verifies this with spot check data, and sends out the invoices to the vehicle owners.
- **Spot Check (SC)** Actual road activity is measured using Spot Checks that record vehicles seen at certain road locations (cells).
- **Vehicle Owner (VO)** The owner of the vehicle that will receive the invoice for the road usage charges to be paid.

**Table 1.** Overview of the information known to each system component

| Component | Knows | Does not know |
|---|---|---|
| On Board Unit | locations visited by vehicle | spot check location<br>vehicles seen by spot check |
| Pricing Provider | locations visited during leg | link of leg to trip or vehicle<br>locations visited by vehicle |
| Aggregation Provider | charge for a leg<br>charge for a trip | link between trip and vehicle<br>locations visited during leg |
| Back Office | charge for a vehicle | locations visited by vehicle<br>link of vehicle to trips or legs[2] |
| Spot Check | vehicles passing | locations visited by vehicle<br>legs, trips |

The description and analysis in this paper assumes that the pricing and aggregation service is offered by separate entities. We assume that in our business context there may be more pricing providers and aggregation providers, which vehicle owners are free to choose.

Note that we need to take some precautions regarding the communication from and to the OBU in order to ensure privacy. In particular, location messages sent from the OBU to the PP must be sender anonymous. Also, the total charge records for a trip calculated by the AP cannot be sent to the OBU directly. Instead, they must be published on a bulletin board from which the OBU (knowing which trip it engaged in) can collect it in an anonymous fashion.

## 5 Architecture

Having discussed the assumptions, requirements and design principles we are ready to discuss our architecture for a privacy friendly yet fraud-resistant road pricing system.

### 5.1 Definitions and Notations

We use the RSA cryptosystem [14], and write $i = (e, n)$ for the private key of entity $i$, and $i = (d, n)$ for the corresponding public key. The encryption of a message $m$ with public key is denoted by $\{m\}_K$. We assume that the encryption is non-malleable [5]: any modification to the encrypted message will be detected. Because we want to use the blinding techniques from Chaum [4], we define the following variants of RSA signatures on a message $m$ with private key $k = (e, n)$ (where we write $a^k$ to denote $a^e$, i.e., we identify keys with the exponents they represent). The *normal* signature is defined similar to RSASSA-PKCS1-V1_5 from PKCS #1 [10].

[2] Although if an OBU submits a charge that is unique, this may be linked to the particular trip that happens to have the same charge as computed by the AP.

$$[m_1, \ldots, m_z]_k \stackrel{\text{def}}{=} h(m_1, \ldots, m_z)^k \bmod n$$

$$[m_1, \ldots, m_z]_k \stackrel{\text{def}}{=} m_1, \ldots, m_z, [m_1, \ldots, m_z]_k .$$

where $h$ is some secure cryptographic hash function. The second form is a shorthand for the message followed by its signature. A *blindable* signature is defined as follows.

$$(m_1, \ldots, m_z, b)_k \stackrel{\text{def}}{=} (h(m_1, \ldots, m_z) \cdot b)^k \bmod n$$

$$((m_1, \ldots, m_z, b))_k \stackrel{\text{def}}{=} m_1, \ldots, m_z, b, (m_1, \ldots, m_z, b)_k .$$

For this signature the following lemma holds.

**Lemma 5.1** For any RSA key pair $K, k, n$ and any $m_1, \ldots, m_z, x$ and $r$ we have

$$(m_1, \ldots, m_z, x \cdot r^K)_k = r \cdot (m_1, \ldots, m_z, x)_k \ (\text{mod} n).$$

Each vehicle has a unique public identifier $v$. Trips (with identifier $t$) are subdivided into legs (with identifier $\lambda$) as discussed in Sect 4. The position (i.e., the cell it is in) of vehicle $v$ at time $t$ is denoted $p_v(t)$. The vehicle's OBU samples a trip at regular time intervals, at least frequent enough to ensure that no 'gaps' occur in the traversed cells it records. All vehicles report the cells visited at time slots 0,1,2,3,.... If more than one cell is visited, all extra cells are reported for the same time slot.

### 5.2 Initial Setup

The system initialises the following private and public key pairs: $k_v, K_v$ for the OBU, $k_{DO}, K_{DO}$ for the Back Office, $k_{PP}, K_{PP}$ for the Pricing Provider, $k_{AP}, K_{AP}$ for the Aggregation Provider, and $k_{SC}, K_{SC}$ for the Spot Check authority. A private key is only known to its owner. Public keys are known to all. Naturally, these keys change from time to time. However, key management issues are outside the scope of this paper.

Vehicle-, trip- and leg identifiers are cryptographically related, but can only be linked to each other by certain parties. This is the essential idea behind our protocol to achieve privacy while maintaining accountability and fraud resistance. They are computed by the OBU in the following way. The trip identifier $\tau$ is derived from the vehicle identifier $v$ using randomisation similar to a blinded signature technique setting $\tau = v \cdot r_\tau^{K_{AP}}$ for some random $r_\tau$. Because of this, using Lemma 5.1, for arbitrary message $m$ the OBU can compute a signature $(m, v)_{k_{AP}}$ from $(m, \tau)_{k_{AP}}$

by dividing out $r_\tau$. This way trip charges computed by the AP cannot be linked to reports submitted by the OBU to the back office.

The leg identifier $\lambda$ is, in turn, derived from the trip identifier $\tau$ by encrypting it to the key of the aggregation provider, together with a random $r_\lambda$, i.e., $\lambda = \{\tau, r_\lambda\}_{K_{AP}}$. The AP can recover the trip identifier corresponding to a leg identifier by decrypting with $k_{AP}$ and dropping the random (which is only included to make different legs non-linkable).

### 5.3 Pricing Protocol

We first describe the protocol to compute the charge for a trip. In the next section we will describe the additional messages that are exchanged to detect fraud.

We note that the messages between PP and PP and OBU and BO are encrypted to the public key of the intended receiver. This is not explicitly written in the description of the protocol messages to avoid superfluous notation.

*At the start of a trip.* OBU $v$ generates a random $r_\tau$ and stores it. It computes $\tau = v \cdot r_\tau^{K_{AP}}$ and stores $\tau$ and $r_\tau$.

*At the start of a leg.* The OBU generates a new random $r_\lambda$ and sets $\lambda = \{\tau, r_\lambda\}_{K_{AP}}$. It initialises a message sequence counter $c_\lambda = 0$. The OBU sends a special marker message to signal the start of a new leg.

*During a leg.* The OBU anonymously[3] sends[4], at each time slot $t$, all visited cells during that time slot as $\{p_v(t), t, \lambda, c_\lambda\}_{K_{PP}}$ to the PP. The data is encrypted to prevent an eavesdropper collecting information about legs and locations. Because the encryption is non-malleable, secrecy of the leg identifier provides authenticity of the message. The message sequence counter $c_\lambda$ is incremented by $1$ for each such message sent. This prevents replay of location records. The PP receives this location record, decrypts it, verifies the expected message sequence counter for this leg, and if valid stores this record with all other location records for this particular leg $\lambda$. The expected message sequence counter is incremented by $1$.

*At the end of a leg.* The OBU sends a special marker message to signal the end of a leg. The PP computes the total charge for the cells visited according to the location

[3] For privacy reasons, the communication medium must hide the sender.

[4] We assume the communication link is reliable. This is a hard requirement in practice, because for privacy reasons the link is essentially uni-directional, which makes it impossible to send acknowledgements. The OBU may decide to send the same message several times to increase the likelihood of proper reception. The PP simply accepts the first message with the correct sequence number.

records stored for leg $\lambda$ . It uses the pricing function to do so. It then sends the charge record $[\![€, \lambda]\!]_{K_{PP}}$ to the AP. The AP receives this message, verifies the signature, and, if valid, decrypts $\lambda$ (which equals $\{\tau, r\}_{K_{AP}}$) to obtain $\tau$ . It stores the record with all other charge records for this particular trip $\tau$ . The AP only accepts a charge record for a particular leg once.

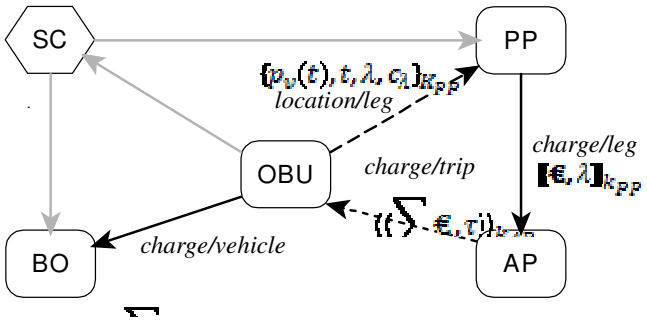


**Fig. 3.** Exchange of pricing messages between the system components.

*At the end of a trip.* The OBU sends a special marker message to signal the end of a trip $\tau$ , which the PP forwards to the AP. After that, the AP no longer accepts charge records for trip $\tau$ . The AP computes the total trip charge $\sum €$ by summing all the charge records for this particular trip $\tau$ . It then computes $\langle\langle \sum €, \tau \rangle\rangle_{K_{AP}}$ , and stores this on a bulletin board. At any time, the OBU can visit the bulletin board[5] (e.g. a web server), and, by knowledge of the trip identifiers it used, it can retrieve the payment records it needs from that bulletin board. The OBU then computes $\frac{\{\sum €, \tau\}_{K_{AP}}}{r_\tau}$ for the value $r_\tau$ stored for $\tau$ . By Lemma 5.1 this equals $\{\sum €, v\}_{K_{AP}}$ (because $\tau = v \cdot r_\tau^{K_{AP}}$. The OBU then sends the record $\langle\{\sum €, v\}\rangle_{K_{AP}}$ to the BO for further processing.

[5] Because the whole purpose of the exercise is to hide the vehicle identity from the AP, the AP cannot send the data to the right OBU directly.

### 5.4 Enforcement Protocol

In this section the enforcement protocol is presented together with the messages that are exchanged to detect fraud.

*Setting up a spot check.* Each spot check has a unique identifier $\gamma$. The spot check SC submits its location $p$ and the time it starts checking vehicles and the time it stops checking vehicles to the pricing provider in the message $[p_\gamma, t_\gamma^s, t_\gamma^s, \gamma]_{k_{SC}}$. A new $\gamma$ is generated for every new location or time period of a spot check.

*Checking vehicles.* The spot check records all passing vehicles[6] and stores all vehicle identifiers it sees in the set $V_\gamma$. A vehicle passing multiple times is only recorded once. This set is sent to the BO when the spot check is closed.

*Generating tokens.* When a PP receives a location record $\{p_v(t), t, \lambda, c_\lambda\}_{k_{PP}}$, it checks for every spot check $\gamma$ whether $p_v(t)$ is close to $p_\gamma$ received for the spot check $\gamma$, and whether $t_\gamma^s \leq t \leq t_\gamma^s$. If this is the case, the PP generates a token $[\gamma, \lambda]_{k_{PP}}$, testifying that this vehicle faithfully submitted its location record for this particular spot check location and time. At the end of a leg, all tokens for that leg are forwarded to the AP.

*Attaching tokens to pricing messages.* Tokens for a leg (and a trip, see below) are piggy bagged onto the pricing message for that leg (or trip). The pricing message contains a fixed number $b$ of bins for this. If the number of tokens is less than $b$, the remaining bins are filled with random data (that look like genuine tokens except they correspond to non-existent spot checks). If the number of tokens exceeds $b$, then remaining tokens are sent in a next message that contains the same pricing message.

*Forwarding tokens.* When the AP receives a token for a leg $\lambda$, it will determine the corresponding trip $\tau$ as before, and store the spot check identifier $\gamma$ with that trip. When the trip is closed, each spot check location recorded is appended to the pricing message using the fixed number of bins approach outlined above, and the AP generates the blind signature over the *whole* message: pricing data and the token bins. The OBU retrieves the charge record for the trip from the bulletin board as explained before, converting the blind signature by dividing our the value $r_\tau$ stored for $\tau$, and forwards the result to the BO.

---

[6] There are several techniques to do so, without the vehicle noticing. We do not discuss this issue in this extended abstract.

*Matching tokens and checked vehicles.* For each trip charge record, the BO verifies the signature. It discards any tokens that contain invalid spot check identifiers $\gamma$, and stores the remaining tokens as tuples $(\gamma, v)$. The BO also receives, for each spot check $\gamma$ a set of vehicle identifiers $V_\gamma$ observed by that spot check. Vehicles must submit their records (charges and tokens) within a certain time frame. The BO processes the data of a spot check when the closing time of the spot check plus extended by the time frame lies in the past. For such a spot check $\gamma$, all valid tuples $(\gamma, v)$ are processed, and any observed vehicle $v$ is removed from the set $V_\gamma$. Any remaining vehicle identifiers in this set indicates a vehicle that did not submit a location record for that spot check location. This either indicates a system failure, or a possible case of fraud. Adequate action towards the Vehicle Owner can then be taken.

## 6 Security and Privacy Analysis

Due to space constraints we only mention the main lemmas that prove the correctness, security and privacy properties of our system.

**Proposition 6.1.** Let trip $\tau = v \cdot r_\tau^{K_{AP}}$ be generated by OBU $v$ (using some random $r_\tau$), and let $\tau$ not be closed yet. If the PP accepts a message for leg $\lambda = \{\tau, r_\lambda\}_{K_{AP}}$ (for some random $r_\lambda$), then that message was sent by OBU $v$.

*Proof.* By non-malleability of the encryption used between OBU and PP, and the fact that $\lambda$ is only known to the OBU and the PP at least until the leg is closed, and $\tau$ is only known to the OBU and the AP at least until the trip is closed.

**Proposition 6.2.** Let $\tau$ be generated by OBU $v$. Let $p$ the set of all location records $\{p_v(t), t, \lambda, c_\lambda\}_{K_{PP}}$ sent by the OBU $v$ during this trip. If the PP and AP are honest, and no messages are lost, then the invoice $\{(\sum \epsilon, \tau)\}_{K_{AP}}$, computed by AP corresponds to the total charge due for $p$.

*Proof.* By Prop. 6.1, as long as $\tau$ is not closed, the charge for a one of its legs $\lambda$ is only based on data sent by OBU $v$. No charges for $\tau$ are accepted by the AP after $\tau$ is closed. By honesty of PP and AP, if no messages are lost, the total charge must correspond to $p$.

**Lemma 6.3 (Pricing Correctness).** If the PP and AP are honest, an honest OBU only submits charges for trips it makes.

*Proof.* Immediately follows from Prop. 6.2.

**Lemma 6.4 (Enforcement Correctness).** If the SC, PP and AP are honest, an honest OBU cannot be accused of fraud.

*Proof.* Honest OBU's receive tokens for each spot check they pass.

**Lemma 6.5 (Fraud resistance).** If the SC and PP are honest, an OBU that submits incorrect, lower, invoices for trips it makes (or submits no such charges at all) will eventually be detected.

*Proof.* By Prop. 6 to submit an incorrect, lower, invoice the OBU must not submit location records for all locations actually visited. Eventually, one such location happens to be spot checked. Alternatively, if the OBU fails to submit a charge record to the BO, it cannot submit the corresponding tokens either.

**Lemma 6.6 (Vehicle Privacy).** The location of a vehicle at a certain time is only known to the OBU, unless that location was being spot checked at the time.

*Proof.* For non-spot checked locations, locations in location records are only bound to leg identifiers. These can be related to trip identifiers by the AP, but no party except the OBU can relate them to vehicle identifiers.

**Lemma 6.7 (Trip Privacy)**. Except for the OBU, no other entity can reconstruct the full trip of a vehicle (unless that trip consisted of a single leg).

*Proof.* As discussed in section 4, if the number of vehicles in a boundary cell of a region is $z$, then the possible combinations of legs that make up a trip that corresponds to some vehicle is multiplied by $z$ with every region change.

## 7 Conclusions and Further Research

In this paper we propose an architecture for a road pricing system that is privacy friendly as well as fraud resistant. In order to obtain fraud resistance we do not rely on trusted elements (for example a smart card or on-board-unit in the vehicle) but use a novel way of enforcement based on spot checking together with proof-of-honesty tokens given to vehicles that faithfully submit location data for spot-checked locations.

Privacy is guaranteed by splitting up trips in short legs that cannot be linked to each other, except by the aggregation provider. By the time the aggregation provider processes a leg, the privacy sensitive location data associated with a leg is already stripped. Hence, the aggregation provider does not get to see the actual locations corresponding to a particular leg. Legs and trips cannot be linked to a particular vehicle, except by the vehicle itself.

We note that our proposed architecture is open to extension of services. That is, the generated road usage data may not only be input for the application of road pricing itself, but can also be input for other road pricing services satisfying other apparent information needs that drivers, car owners, etc, may have. Moreover our solution can easily be applied to other contexts such as smart metering and e-ticketing in public transport where privacy and fraud-resistance seem to be conflicting issues.

Further work could address improvements in the privacy level of the enforcement protocol. In the current system, spot checks record the vehicle identifier of all vehicles passing. It may be possible to encrypt this identifier so that the spot checks themselves do not have access to the identifier, while the back-office can only decrypt the identifier if the corresponding proof-of-honesty token has *not* been received.

We thank the members, in particular Han Vogel, of the TNO project LERP for fruitful discussions on this topic.

## References

1. Aurenhammer, F., Klein, R.: Voronoi Diagrams. In: Sack, J.-R., Urrutia, J. (eds.), Handbook of Computational Geometry, pp. 201-290. North-Holland, Amsterdam, Netherlands (2000).
2. Balasch, J., Rial, A., Troncoso, C., Geuens, C., Preneel, B., Verbauwhede, I.: PrETP: Privacy-Preserving Electronic Toll Pricing. Technical report, COSIC, KU Leuven (2010).
3. Blumberg, A., Keeler, L., Shelar, A.: Automated traffic enforcement which respects driver privacy. 7th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC) (2004).
4. Chaum, D.: Blind Signatures for Untraceable Payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology: Proceedings of CRYPTO '82, pp. 199-203. Plemum, New York (1983).
5. Dolev, D., Dwork, C., Naor, M.: Nonmalleable Cryptography. SIAM J. Comput., 30(2), 391-437 (2000).
6. Eisses, S., de Jonge, W., Habers, V.: Privacy and distance-based charging for all vehicles on all roads. Proceedings of the 13th ITS World Congress (2006).
7. Gruteser, M., Liu, X.: Protecting Privacy in Continuous Location-Tracking Applications. IEEE Security & Privacy, 2(2), 28-34 (2004).
8. Hubaux, J.P., Capkun, S., Luo, J.: The Security and Privacy of Smart Vehicles. IEEE Security & Privacy, 2(3), 49-55 (2004).
9. de Jonge, W., Jacobs, B.: Privacy-Friendly Electronic Traffic Pricing via Commits. In: Degano, P., Guttman, J.D., Martinelli, F. (eds.) Formal Aspects in Security and Trust, 5th International Workshop, FAST 2008, Malaga, Spain, October 9-10 2008, , pp. 143-161. Revised Selected Papers. Lecture Notes in Computer Science 5491. Springer (2009).
10. Jonsson, J., Kaliski, B.: Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, RFC 3447 (2003).
11. Pieper, R.: MobiMiles: Bewust op weg. (report produced for the Ministry of Transport, Public Works and Water Management) (2001).
12. Popa, R.A., Balakrishnan, H., Blumberg, A.J.: VPriv: Protecting Privacy in Location-Based Vehicular Services. Proc. 18th USENIX Security Symposium, pp. 335-150, San Jose, CA, USA (2009).
13. Rass, S., Fuchs, S., Schaffer, M., Kyamakya, K.: How to protect privacy in floating car data systems. In: Sadekar, V.K., Santi, P., Hu, Y.C., Mauve, M. (eds.) Proceedings of the Fifth

International Workshop on Vehicular Ad Hoc Networks, pp. 17-22. VANET 2008, San Francisco, California, USA, September 15, 2008. ACM (2008).
14. Rivest, R.L., Shamir, A., Adleman, L.M.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM, 21(2), 120-126 (1978).
15. Troncoso, C., Danezis, G., Kosta, E., Preneel, B.: PriPAYD: privacy friendly pay-as-you-drive insurance. WPES, pp. 99-107 (2007).