

Cell-based Roadpricing

Flavio D. Garcia, Eric R. Verheul, and Bart Jacobs

Institute for Computing and Information Sciences,
Radboud University Nijmegen.
P.O. Box 9010, NL-6500 GL Nijmegen, The Netherlands.
{e.verheul,flaviog,bart}@cs.ru.nl

June 24, 2011

Abstract This paper proposes a new approach to electronic roadpricing, based on a division of the roadpricing area into cells, each with their own fee. Some of the cells are secretly marked as check cells. On board equipment contains a secure element that is made aware of these check cells and helps the pricing authorities to monitor the user. This approach is not only original but it also improves upon earlier approaches since it solves issues regarding positioning accuracy, collusion between different users, and the required level of interaction.

1 Introduction

Roadpricing, also known as Electronic Traffic Pricing (ETP), refers to a location-based charge for road use. It exists in several forms, for instance via charges for entering the city center (London) or for particular motorways (France or Italy), or for lorries (Germany). Here we consider satellite-based systems, using GPS or Galileo, for personal vehicles. Such approaches have been elaborated to some extent in the Netherlands, as part of earlier, now abandoned, government plans. However, at the European level this form of roadpricing is still on the political agenda (based on the framework [1,2]). The main reasons for replacing a flat road tax by a location-based approach are: (1) fairness of charges, since one only pays for actual road use, and (2) the possibility to steer the traffic supply via a flexible pricing policy (*e.g.* making busy roads expensive during rush hours).

Location-based roadpricing for personal vehicles is a highly privacy sensitive matter since it requires detailed location information of individual vehicles. Public support depends on proper privacy protection via an architecture guaranteeing data minimalisation focused on the goal of roadpricing itself, and not on secondary goals (like speed limit enforcement). The topic has been picked up in the computer security research community, see [3,7,6], and several different protocols have been proposed. This paper contributes with a novel protocol, that uses cells (covering the roadpricing area) both for fee calculation and for fraud detection (via a secure element, like a smart card, embedded in the car's toll device). Our approach addresses some problems with fraud detection in earlier protocols, like time-dependence, collusion, GPS-precision and required level of

interaction. The paper does not introduce any new cryptographic primitives and makes use of existing techniques. Its value lies in adapting these techniques in an original manner to a context of considerable societal relevance.

2 Preliminaries

We adopt the main building blocks of the Electronic Toll Pricing (ETP) architecture proposed by the European Union [1]. It distinguishes the following five components/parties of the system:

- an ETP subscriber;
- a toll charger TC;
- a toll service provider TSP;
- an on board unit OBU, in every (personal) vehicle;
- a secure element SE incorporated in the OBU.

An ETP subscriber is the party that uses the road of the toll charger and for this subscribes to a toll service provider. The main information security objectives of the subscriber are twofold. The first objective is that he gets to pay the correct amount for road usage and that he can validate that. The second objective is that his privacy is sufficiently protected in the scheme. This obviously includes more than sufficient protection of his road movements but we concentrate on this aspect in this paper. With *sufficient protection* we mean that the subscriber accepts the privacy protection design of the toll service provider system and has assurance that the actual implementation adheres to the design ('public trust'). Preferably, the subscriber should be able to validate that himself without having to rely on a (trusted) third party.

The toll charger TC is the party that collects a toll fee for the usage of the roads and defines the prices and conditions of use. Typically national or regional authorities are TCs, but also commercial parties 'owning' the roads, can be a TC.

The toll service provider TSP is the party that provides the ETP service. It periodically determines the fee to be paid by each subscriber (vehicle) for his road usage. In this paper we are often not very explicit about the difference between the TC and TSP, since the distinction is not so relevant for the protocol.

An important information security objective shared by the TC and TSP is that subscribers are correctly charged for the road usage and can detect fraud. A specific security objective of the TC is that he should be able to validate that usage of its roads is correctly reflected in the payments of the TSP.

The on board unit OBU is a satellite (GPS or Galileo) enabled device that will be attached to every vehicle subscribed to the ETP service. We assume that this device stores a pricing function \mathcal{P} that takes as input a location l and a time t and it outputs a price p which corresponds the toll price of the corresponding road at time t . Furthermore, the OBU must have a timing device which is reasonably in sync with the local time. It should also be able to occasionally communicate with the TSP in order to report usage and to update the pricing function (and

a list of checkpoints that corresponds with the location of surveillance cameras, see below). The OBU is untrusted, from a systems perspective: subscribers may try to manipulate it, for instance via the power supply or via the satellite signals (shielding the device or feeding it false signals), in order to reduce the fee they need to pay.

The secure element SE is a tamper resistant device, like a smartcard, which has a modest amount of non-volatile memory and processing power. This processing power must be enough to perform basic (public key) cryptographic operations. Typically, the secure element does not have its own power or clock. It is comparable to a SIM card in a mobile phone.

2.1 Thin and fat OBU's

Before going into the technicalities, we briefly describe the setting and the main idea underlying the protocol that is proposed in this paper.

In all satellite-based roadpricing systems vehicles have an on board unit OBU that can at least determine the location of the vehicle and communicate with the back-office. In [7] a distinction is made between *thin* and *fat* OBU's. A thin OBU just collects location information and passes it on the back-office where the appropriate fee is calculated. This is a simple but extremely privacy-unfriendly approach, since the sensitive location information is stored outside the direct control of the individual involved, in a large database that is vulnerable in various ways. A fat OBU on the other hand is capable of calculating the price itself, via a pricing function that is executed in a secure environment. At the end of a reporting period the OBU sends the cumulative fee to the TSP.

The fat approach is privacy friendly but has two big disadvantages:

1. a fat OBU is more complex than a thin one, and thus more expensive and more vulnerable;
2. fraud detection is more complex with fat OBU's.

One way to handle the first point is to reduce the trusted computing base (TCB) of an OBU to a minimum and to place this TCB on a tamper-resistant secure element SE such as a smart card. This separation makes it easier to add additional (commercial) services to the OBU, or alternatively, to add roadpricing functionality to existing in-car equipment (such as satellite navigation) by adding the secure element¹.

The second point is a serious issue. First, fraud detection with thin OBU's is easy: a TC/TSP just places roadside camera's at random places and check if the vehicles that pass by report the location where they are spotted. If not, the car owners will be fined. One sensitive point in this approach is that camera locations must remain secret, throughout the reporting period. If they become known, drivers may simply switch off their OBU and take a different route, avoiding cameras. With a fat OBU one way of fraud detection is to have the

¹ The secure element is bound to a particular individual, and can via this separation be transferred from one vehicle to another.

TC/TSP communicate briefly from a roadside detection point with the (secure element in the) OBU of each passing car in order to check if the last few locations used by the pricing function are consistent with the check locations. We remark that the ETP architecture [1] actually suggests the usage of 5.8 GHz Dedicated Short-Range Communications (DSRC) for this. However, since such checks are active, involving two-way communication, they will be noticed very quickly by passing cars. The check locations will thus become publicly known, negatively affecting the efficacy of the checks.

Another fraud detection approach is to let the OBU (or its secure element) commit itself to each step of the fee calculation, for instance via a simple hash function [7] or via a more complex non-interactive commitment scheme [3] (using homomorphic encryption and zero-knowledge proofs to exclude negative sub-fees). After receiving a (cumulative) fee report, the TC/TSP may ask the OBU to “open” certain commitments, corresponding to roadside camera locations, in order to check details of the cumulative report. Auditing thus involves interaction with the user, even in the case where everything turns out to be fine. In contrast, with the approach proposed here there will only be interaction with the user in case fraud is detected.

The main problem with the commitment approach is its vulnerability to collusion, exploiting time dependence. During such commitment checks the locations of the road-side camera’s become known. If checks are performed immediately, like in [3], people may collude, where one of them (who drives a lot) reports first and learns about the check locations and the other ones subsequently delete commitments for non-check locations in their fee reports (thus reducing their fee).

Hence such checks can only be performed well after the reporting period, when all OBU’s have sent in their fee report. But such timing dependency may introduce other vulnerabilities. What should happen if you are on holidays at the end of the reporting period, outside the roadpricing area, and outside reach of the TSP? If you are allowed to send in your report with a delay, when you return, you could adapt your fee report based on knowledge of the checkpoints. Such manipulations can exploit the fact that secure elements do not have their own (secure) clock and so you can feed them false timing information. One way of addressing this “absence” problem is forcing OBU’s to report their fee as soon as they leave the roadpricing area. This naturally leads to the idea of carving up the area into cells (where border cells play a special role).

2.2 Current protocol idea

We now assume that the roadpricing area (country, region, or continent) is covered by square, non-overlapping cells of a relatively small size (*e.g.* a square kilometer). When you enter a cell you will have to pay a certain amount, depending on the cell (and possibly the time of day). These payment details are incorporated in a payment function \mathcal{P} that will be left unspecified. As long as you stay within one cell, there is no further charge. But there will be a new

charge as soon as you enter the next cell. In Subsection 2.3 we elaborate a bit more on the organization of these cells.

A certain subset of cells are marked as “check cells” or “checkpoints” and contain (hidden) roadside cameras. These markings are chosen by the TSP and change every billing period (of, say, one or three months). The secrecy of these markings is crucial for the protocol. Another crucial aspect is that the secure element in your OBU gets to know these checkpoint markings, via a secure connection with the TSP, but you or your (untrusted) OBU should not learn about the current checkpoints. This confidential transfer of checkpoints happens before the beginning of each new billing period.

Upon entering a new cell, the OBU reports— or ‘declares’, as we shall say below—this transition to its SE. The SE replies with a “ticket”. The ticket is a (randomized) encryption of ‘OK’ with some padding if the new cell is *not* a checkpoint. But if the cell is a checkpoint—which is known to the SE but not to the OBU—the ticket will be meaningful for fraud detection later on. The encryptions should be such that the OBU cannot see which tickets are for checkpoint cells and which are for non-checkpoint cells. In this way the trusted element works against its owner, in the interest of the TC and TSP: it secretly monitors the user. Thus, the SE is the TC and TSP’s ambassador (or spy, if you like), see also [4].

Driving around *without* a proper, up-to-date list of check cells is dangerous, because without it your SE cannot produce appropriate tickets when you happen to enter a check cell (leading to subsequent fines, when audited). Hence it is in the own interest of users to have their system up-to-date. Whenever the SE does not have—for whatever reasons—an up-to-date list of check cells, it should report so via an alarm signal to the user. There should be procedures in place for malfunction, for instance by pushing a ‘reset’ button (including an implicit notification to the TSP) or by going quickly to some service station.

From a system perspective the secure element is trusted and inaccessible to malicious users. In contrast, the OBU is untrusted. When the trusted element ever gets compromised, the system breaks down. This is usually the case with a TCB: if a SIM gets compromised identities can be stolen and phone bills will probably end up with the wrong person. But having this tamper resistant SE as TCB also simplifies matters a lot: the SE is trusted so it can simply accumulate the fees per cell, store the result in its non-volatile memory, and report the cumulative fee, appropriately authenticated, to the TSP at the end of the reporting period. Because we have such a clearly separated TCB we do not need to use the homomorphic encryptions and proofs as in [3].

Both the TC and TSP trust the accumulation of fees reported by the SE, but they do not trust that the SE has been notified appropriately by the OBU at every cell transition. The TC and TSP thus check these notifications via the tickets. As part of the fee reporting—or possibly only on request of the TSP—the OBU sends all tickets of the reporting period to the TSP. The TSP can then open the tickets for the checkpoint cells where the vehicle has been spotted and

see if the content is appropriate. The TSP can also communicate this evidence to the TC. Details are given below, in Section 3.

2.3 Cells and roads

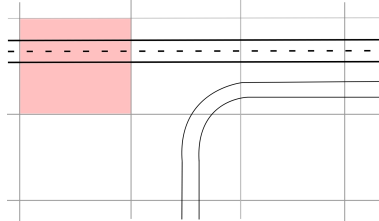


Figure 1. An expensive cell exclusively covering a motorway segment, in between two exits, surrounded by inexpensive cells.

One thing that is new in our approach is the reliance on cells, and not on road segments, as units for payment. It deserves some more explanation. One basic idea in roadpricing is that certain roads (*e.g.* busy motorways) or bridges/tunnels are more expensive than for instance quiet country roads. Simply making all cells in which an expensive motorway occurs expensive is too crude, since there may be multiple roads in a particular cell. What we propose is that the price of a cell with multiple roads corresponds to the price of the cheapest road (in that cell). In order to charge the appropriate fee for an expensive road one cell is made very expensive that only contains this road and cannot be avoided. Such a cell can for instance be found on a stretch of motorway in between two exits, as illustrated in Figure 1. Of course, such a cell is a prime candidate to be (regularly) a check cell.

One (more) advantage of using cells over road segments is that substantial margins can be allowed in the accuracy of the location provided by the positioning system, see Figure 2. This is a great benefit, since in practice the positioning accuracy is influenced for instance by atmospheric disturbances or by reflections of satellite signals in urban areas. Such lack of accuracy may be a problem for a road-segment based approach, when there are adjacent, parallel roads with different prices. In contrast, our cell-based approach is more robust. Also, the system is independent of the (often proprietary) mechanisms for transforming satellite signals into road segments, on which other roadpricing approaches rely. Hence it can be realised more easily via open (international) standards.

An interesting question is what the ideal size of cells is. From a privacy perspective it may seem at first that larger cells are better since they give poorer approximations of actual locations. But the actual location of a vehicle that is photographed becomes known via the location of the camera. With larger cells one has for each check cell:

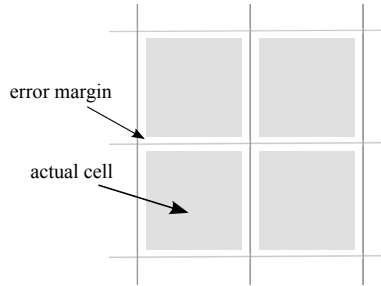


Figure 2. Positioning system error margin around cells.

- a lower chance that a vehicle actually passes a camera within that cell,
- the obligation to—nevertheless—send in a non-void SE-tickets which reveals to the TC that the vehicle at hand was in the cell.

Hence smaller cells mean less unnecessary reporting of presence and thus better privacy protection. In Section 6 we use 100×100 meter as cell size in a sample calculation.

2.4 Timing

As we argued above, the timing of fee reporting, auditing, and feedback is a sensitive matter in road pricing. In our approach, fee reporting takes place:

1. at the end of the reporting period;
2. when a vehicle leaves the road pricing area (*e.g.* the country);
3. when a vehicle is sold or taken out of service.

The last point is rather special. It can be handled by giving an OBU a special ‘report now’ button; after pushing it the embedded SE should be removed.

The passage of a border will be detected by the OBU and should lead to an (automatic) fee report. It may create some administrative overhead for people who frequently cross borders, but since fee reporting is automatic this should not be such a problem.

It is a more serious problem if the fee report is suppressed by a malicious user upon leaving the road pricing area. In this way a vehicle may disappear from the grid. However, upon re-entering there are two options.

- This re-entering happens ‘soon’, still within the same reporting period as the departure; then the vehicle can in principle continue to drive around in the roadpricing area, using the (still valid) list of check cells in its SE. The OBU/SE will have to report normally, at the end of this period.
- Re-entering happens ‘late’, in a subsequent reporting period. There is now a window for fraud, because the check locations from the reporting period of departure must be assumed to be publicly known. However, the vehicle

will need an (updated) list of check cells before it can drive around after re-entering without any risk. It is at this point that it has to communicate with the TSP, to request the new list, and the TSP notices the missing report (from the period of departure); the TSP will impose a (hefty) penalty.

The first, ordinary way of reporting happens at the end of the reporting period, say once a month. Vehicles will be forced to report, because only after reporting they will receive the new list of check cells. Since there is some time in between entering one cell and entering the next one, the fee reporting need not interfere with the ordinary declaring.

After (some time after) receiving the fee reports and distributing the new lists of check cells, the TSP goes into the auditing phase: tickets for check cells will be decrypted and inspected, possibly leading to penalties. Notice that this auditing does not require any interaction with the user—assuming appropriate tickets for check cells are submitted.

(There is one practical issue related to performance. If all vehicles submit their fee report at 24:00 hours of the last day of the month, communication channels will be overloaded. Hence one may wish to allocate different reporting times to different vehicles, in order to balance the load. This involves some additional administration, including management of check cells lists and delay of auditing, but it does not change the system fundamentally.)

3 The protocol

This section describes our new, privacy-friendly ETP protocol. This protocol is a triple $\langle \mathcal{S}, \mathcal{U}, \mathcal{D} \rangle$ of polynomial-time interactive protocols. The Setup Protocol \mathcal{S} is a two party protocol involving the TSP and the SE. This protocol is run when a new SE is added to the system. The Update protocol \mathcal{U} is a three party protocol involving the SE, the TSP and the OBU. This protocol runs periodically, at the end/beginning of a billing period, *e.g.* monthly. The protocol Declare Segment \mathcal{D} is a two party protocol involving the SE and the OBU. This protocol runs every time that the vehicle enters a new cell.

Notation: We write $\{m\}_{pk}$ to denote the encryption of message m with public key pk , while $[m]_{sk}$ denotes message m followed by a signature on m with private key sk .

System Setup \mathcal{S} When the system is initialized,

- the TSP creates a public key pair (pk_{TSP}, sk_{TSP}) and partitions the map of the country (or region under consideration) in small cells, forming a grid that overlaps the relevant area. Let \mathcal{C} be a mapping from GPS coordinates $(lat, long)$ to a cell number c . These cells must be small enough to provide the desired granularity level for road pricing. The function \mathcal{P} assigns to each cell c and time t the corresponding road price p .

- the OBU stores a ticket list T which is initialized to the empty list.
- the SE of a vehicle generates its own public key pair (pk, sk) . The public key is registered by the TSP and the public key pk_{TSP} of the TSP is stored in the SE and in the OBU. The SE also has a counter $balance$ which is initialized to zero.

Update \mathcal{U} The purpose of the update protocol is threefold:

- for the TSP to get the toll charge for the last billing period;
- for the SE to get the updated checkpoint list for the next billing period;
- to perform fraud detection.

At the beginning of each billing period, the TC creates a list CHECKPOINTS of checkpoint cells. The checkpoint cells are those cells where the TC will have a surveillance camera for the next billing period. These checkpoint cells will be used later on for fraud detection. To prevent abuse from the TC/TSP, the maximum size of the CHECKPOINTS list N_{chk} must be enforced by the SE to be a small fraction of the total number of cells in the map N_{cells} . This prevent TC or TSP to set too many cells in the system as checkpoints thus hampering privacy.

At the end of each billing period, the TSP, OBU and SE execute the following protocol.

1. OBU \rightarrow TSP : update vehicle v
2. SE \rightarrow OBU : $[balance, ts]_{sk_{SE}}$
3. OBU \rightarrow TSP : $\{ [balance, ts]_{sk_{SE}}, T \}_{pk_{TSP}}$
4. TSP \rightarrow OBU : $\{ [CHECKPOINTS_{new}, ts]_{sk_{TSP}} \}_{pk_{SE}}$
5. OBU \rightarrow SE : $\{ [CHECKPOINTS_{new}, ts]_{sk_{TSP}} \}_{pk_{SE}}$

The OBU initiates the update protocol identifying the vehicle. Then, the SE sends the current balance $balance$ and the time stamp ts signed with its private key sk_{SE} . The OBU sends this last message, together with the ticket list of tickets T , encrypted to the TSP in step (3). In step (4) the TSP sends the checkpoint list for the next billing period $CHECKPOINTS_{new}$ signed together with a timestamp ts . This list is encrypted with the public key pk_{SE} of the SE. In step (5) of the protocol, the OBU simply forwards this message to the SE. The SE validates that the time stamp ts is in the future with respect of its notion of time i.e., $ts > time$ and sets $time = ts$ and updates its checkpoint list.

When the TSP performs fraud detection, it is assumed that the TC provided TSP with evidence that the vehicle v has been at cell c at time t . Then the TSP will decrypt each entry in the transcript list T provided by v (see Section 6 for optimizations), ignoring those entries that decrypt to zero. One of these entries must be of the form $\langle c_{pre}, c_{cur}, ts, p \rangle$ with $c = c_{cur}$ and $ts \approx t$. If that is not the case it means that this vehicle has committed fraud.

Note that the signature over the new checkpoint list on steps (4) and (5) is there to prevent a trivial denial of service attack on the system but it is not required for the security of the protocol as defined in Section 4.

Declare Segment \mathcal{D} Every time that the vehicle enters a new cell c_{cur} , the OBU and the SE run the following protocol

1. OBU \rightarrow SE : $c_{\text{pre}}, c_{\text{cur}}, ts, p$
2. SE \rightarrow OBU : *ticket*

where c_{pre} is the previous cell, c_{cur} is the current cell $\mathcal{C}(lat, long)$, ts is the local time and p is the toll price $\mathcal{P}(c_{\text{cur}}, ts)$. Upon receipt of message (1), the secure element verifies that the cell c_{pre} correspond to the c_{cur} of the previous Declare Segment protocol run. The SE validates that the time stamp ts is in the future with respect of its notion of time *i.e.*, $ts > time$ and checks that $p > 0$. Then the SE adapts its local time, by setting $time = ts$, and increases the counter *balance* by p , and subsequently it returns a ticket. If the cell $c_{\text{cur}} \in \text{CHECKPOINTS}$ then the SE sets $ticket := \{ [c_{\text{pre}}, c_{\text{cur}}, ts, p]_{\text{SE}} \}_{\text{pk}_{\text{TSP}}}$. Otherwise it sets $ticket := \{ 0 \}_{\text{pk}_{\text{TSP}}}$. The OBU stores the ticket from message (2) in the ticket list T , until the next update protocol execution.

We remark that the subscriber's OBU can independently of the SE store all declarations in plaintext and keep its own balance register allowing the subscriber to validate the charges sent to the TSP during the update protocol.

4 Security notions

This section introduces different security notions. Most of it is standard and it is taken from the literature. It first recalls the notion of indistinguishability under chosen ciphertext attacks for an encryption scheme and the notion of strong existential unforgeability under chosen message attacks for a signature scheme. Then we propose a new security definition for ETP protocols that capture realistic threats that have not been considered in previous security definitions from the literature.

Definition 4.1. *An encryption scheme is a triple $\Gamma = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ of probabilistic polynomial-time algorithms. \mathcal{K} takes as input the security parameter 1^η and produces a key pair (pk, sk) where pk is the public encryption key and sk is the private decryption key. \mathcal{E} takes as input a public key pk and a plaintext m and outputs a ciphertext. \mathcal{D} takes as input a private key sk and a ciphertext and outputs a plaintext or \perp . It is required that $\mathbb{P}[(pk, sk) \leftarrow \mathcal{K}(1^\eta); c \leftarrow \mathcal{E}(pk, m); m' \leftarrow \mathcal{D}(sk, c) : m = m'] = 1$. We write $\{m\}_{pk}$ to denote $\mathcal{E}(pk, m)$.*

Definition 4.2 (IND-CCA-Game).

IND-CCA-Game $_{\Gamma, \mathcal{A}}(\eta)$:
 $(sk, pk) \leftarrow \mathcal{K}(1^\eta)$
 $p_0, p_1 \leftarrow \mathcal{A}_0^{\mathcal{D}}(pk)$
 $b \leftarrow \{0, 1\}$
 $b' \leftarrow \mathcal{A}_1^{\mathcal{D}}(\{p_b\}_{pk})$
winif if $b = b'$.

Adversaries implicitly pass state i.e., from \mathcal{A}_0 to \mathcal{A}_1 .

Definition 4.3 (IND-CCA). *An encryption scheme Γ is said to be IND-CCA secure if for all probabilistic polynomial-time adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$*

$$\mathbb{P}[\mathbf{IND-CCA-Game}_{\Gamma, \mathcal{A}}(\eta)] - 1/2$$

is a negligible function of η . This adversary has access to a decryption oracle \mathcal{D} that on input a bitstring c' outputs $\mathcal{D}(sk, c')$ with the only restriction that c is not equal to the challenge ciphertext $\{p_b\}_{pk}$.

We recall the definition of signature scheme and the notion of strong existential unforgeability under chosen message attacks.

Definition 4.4. *A signature scheme is a triple $(\text{Gen}, \text{Sign}, \text{Vrfy})$ of probabilistic polynomial-time algorithms. Gen takes as input the security parameter 1^η and produces a key pair (vk, sk) where vk is the signature verification key and sk is the secret signing key. Sign takes as input sk and a message m and produces a signature s of m . Vrfy takes as input vk , a message m and a signature s and outputs whether or not s is a valid signature of m .*

<p>SEU$_{\Sigma}(A)$: $(vk, sk) \leftarrow \text{Gen}(1^\eta)$ $m, \sigma \leftarrow A^{S_{sk}}(vk)$ winif Vrfy($vk, (m, \sigma)$)</p>

We recall the standard notion of strong existential unforgeability under chosen message attacks [5].

Definition 4.5 (Strong existential unforgeability). *A signature scheme $\Sigma = (\text{Gen}, \text{Sign}, \text{Vrfy})$ is a strong, existential unforgeable if the success probability of any probabilistic polynomial-time adversary A in the game $\mathbf{SEU}_{\Sigma, \mathcal{A}}(\eta)$ is negligible in the security parameter η .*

Next we introduce a new security notion for ETP protocols. Here an adversary is able to navigate a number of vehicles at will and it is also able to communicate with a number of secure elements and with the TSP. The adversary wins the game if it is able to get a toll fee reduction for any vehicle and manages to do so with low probability of being detected. Low probability here means lower probability than that of simply driving by through a random cell without declaring it. Such an adversary models, for instance, the situation where a number of users collude in order to avoid toll charges. This threat has not been considered by Balasch et al [3].

Definition 4.6 (Security-Game).

Security-Game $_{\Pi, \mathcal{A}}(\eta)$:

$params \leftarrow Setup(1^\eta)$

$v \leftarrow \mathcal{A}^{\mathcal{O}, \mathcal{D}}(params)$

$(\mathbf{l}, \mathbf{t}) \leftarrow Trajectory(v)$

$p, b \leftarrow Result(v)$

winif if $b = 1$ and $p < \mathcal{P}(\mathbf{l}, \mathbf{t})$

were the adversary \mathcal{A} has access to an oracle \mathcal{D} which on input a vehicle identifier v and a GPS coordinate g , models the (physical) movement of vehicle v from the current location to GPS coordinates g . If g corresponds to a cell $c \in \text{CHECKPOINTS}$ then the oracle provides the corresponding evidence to the TSP. \mathcal{A} has also access to an oracle \mathcal{O} which allows her to communicate to the other parties (in this case the SE and the TSP) in the ETP protocol Π . The function $Trajectory$, on input a vehicle identifier v returns a vector of pairs $Location \times Time$, corresponding to the trajectory of vehicle v that has been submitted to the oracle \mathcal{D} during the previous billing period. The function $Result$, on input a vehicle identifier v returns the toll charge p which v has to pay as a result of the last Update protocol run, together with a bit b which equals one when no fraud has been detected for vehicle v . The function \mathcal{P} is overloaded in the definition to vectors of $Location \times Time$ pairs and computes the correct price of this trajectory.

Definition 4.7 (Security). An ETP protocol $\Pi = \langle \mathcal{S}, \mathcal{U}, \mathcal{D} \rangle$ is said to be secure if for all probabilistic polynomial-time adversaries \mathcal{A}

$$\mathbb{P}[\mathbf{Security-Game}_{\Pi, \mathcal{A}}(\eta)] < N_{chk}/N_{cells} + \text{negl}(\eta)$$

where negl is a negligible function on its input η .

5 Security and privacy of the protocol

Theorem 5.1 (Security). If the encryption scheme Γ is IND-CCA secure and the signature scheme Σ is strong, existential unforgeable then the protocol proposed in Section 3 is secure with respect to Definition 4.7.

Proof. Assume by the contrary that there is an adversary \mathcal{A} that wins the **Security-Game** with probability significantly larger than N_{chk}/N_{cells} . In order for \mathcal{A} to be able to win the Security-Game, it must be able to drive a vehicle through a cell without declaring it or do so but with a lower price. In order to win with probability significantly higher than N_{chk}/N_{cells} , it must either be able to construct valid tickets for the checkpoint cells or be able to avoid checkpoint cells with non-negligible advantage. In the first case we build the following adversary \mathcal{B} against the IND-CCA security of the encryption scheme. In the second we build the following adversary \mathcal{C} against the strong unforgeability of the signature scheme.

The adversary \mathcal{B} will first simulate the environment for \mathcal{A} , this is, it will proceed as in System Setup, creating all public key pairs by itself, except for pk_{SE} of a randomly chosen vehicle v . For this key \mathcal{B} will use the challenge key from the IND-CCA game to \mathcal{A} . Then, \mathcal{B} creates a checkpoint list containing one single cell c and sets the challenge plaintext $p_0 = \{c\}$ and sets p_1 equal to the empty checkpoint list. \mathcal{B} receives then a challenge ciphertext $\{p_b\}_{\text{pk}_{\text{SE}}}$ which it uses in step 4 of the Update protocol for v . At some point \mathcal{A} stops and outputs a vehicle identifier v' . If $v \neq v'$ then \mathcal{B} stops and outputs a random bit b' . Otherwise \mathcal{B} picks a random cell c' and compare how many times the vehicle v drives through c and c' . If v drives through c' more than through c then \mathcal{B} stops and outputs zero. By the contrary, if v drives through c more (or equally) often than through c' then it outputs one.

Next we construct an adversary \mathcal{C} against the strong unforgeability of the signature scheme. Just as before, \mathcal{C} will simulate the environment for \mathcal{A} , creating all public key pairs by herself, except for pk_{SE} of a randomly chosen vehicle v . For this key \mathcal{C} will use the challenge key from the SEU game. Whenever \mathcal{C} needs a signature using sk_{SE} it will simply invoke the signing oracle \mathcal{S}_{sk} . At some point \mathcal{A} stops and outputs a vehicle identifier v' . If $v \neq v'$ then \mathcal{C} stops as well and outputs \perp . If $v = v'$ then \mathcal{C} will analyze the transcript of the Update protocol for vehicle v submitted by \mathcal{A} to the oracle \mathcal{O} , in particular to message 5. \mathcal{C} decrypts this message (since it has possession of sk_{TSP}) and all tickets in the ticket list T . As a result \mathcal{C} obtains a number of signatures (ignoring those tickets that decrypt to zero). From these signatures \mathcal{C} ignores those that it has submitted to the signing oracle \mathcal{S}_{sk} . From the remaining signatures it chooses one at random and stops. In case that there is no such a signature, then it outputs \perp . \square

Theorem 5.2 (Privacy). *If the checkpoint list matches the actual positioning of surveillance cameras, then the scheme proposed in Section 3 does not leak any extra location information about the vehicles.*

Proof. It is straightforward to see that whenever the current position of a vehicle is not a checkpoint cell, then the only information received by the TSP is an encryption of zero. \square

6 Practical considerations and extensions

This section elaborates on practical issues concerning getting public trust in the SE, regarding memory, communication and computational complexity of the protocol and it discusses possible optimizations.

With respect to getting public trust in the SE we suggest that it is certified against an appropriate common criteria profile and that its source code (*e.g.* JavaCard applet) is disclosed. In addition, we suggest that after a certain period time the checkpoint lists are also disclosed and that subscribers can get access to cryptographic keys through the TSP allowing them to inspect the actual tickets sent in the update protocol. To further convince subscribers that they are not

sent checkpoints specifically tailored for them one could send the checkpoints in through broadcast (e.g., through the Radio Data System (RDS) which is commonly used for dissemination of road traffic information). Using RDS would also allow for regular checkpoint updates without the need for the subscriber to be connected to a two way communication network such as GSM, GPRS or UMTS that also allow for pinpointing the subscriber’s location. Actually, it might also be cheaper to use RDS broadcasts than point-to-point communication.

If we consider cells of 100 by 100 meters then the protocol described in Section 3 generates a ticket list of approximately 600 KB per 1000 km. Given that the average car usage in Europe is in the order of 15000 km per year, this accounts for 9 MB of data per car per year, which is manageable but it can be improved.

It is possible to reduce the computational and bit complexity of the protocol by a small modification in the Declare Segment protocol \mathcal{D} . This modification consists of accumulating a number of tickets at the SE and issuing only one ticket every N cells. This ticket contains the same information as the earlier N tickets, but it requires only one signature and one encryption. This modification does not affect the security of the scheme as long as the resulting ticket is of constant size, *i.e.*, padding should be used for those tickets that are just zero. For $N = 10$ this optimization reduces the size of the transcript file to just 150 KB per 1000 km which accounts to roughly 2Mb of data per year for an average user. If we consider monthly reports, the transcript file that needs to be sent will be on average of size 180 KB.

If this modest communication complexity is considered unacceptable, it can further be reduced by only auditing a small random set of users. In Step (3) in the Update protocol \mathcal{U} the vehicle can first commit to a certain balance and then the TSP decides probabilistically whether it will request the ticket list T or not. This comes of course as a trade off for security but it can be a viable option when the sanctions for fraud are severe enough.

Both optimizations reduce the amount of data that needs to be transmitted and the computational complexity of the protocol. This is also important for the TSP. In order to search for the ticket for a particular spot check, the TSP needs to perform a number of decryptions. The timestamp in the tickets can be used to search through the ticket list in (probabilistic) logarithmic time.

We remark that the scheme can obviously support more than one balance registers corresponding with cell cost categories instead of actual monetary values. In this way the OBU does not need to be updated with actual cost information; the transformation from cell cost categories to actual cost can be done by the TSP. Also the scheme can support various checkpoint lists varying from more locally oriented lists to list for subscribers who have been caught on actual fraud and get larger checkpoint list as a penalty. We also remark that the scheme trivially supports toll roads by just letting toll road segments correspond to cells. We finally remark that the system can also support parking fee payment, by introducing a status to cells, e.g “driving” and “parking”. The transition from one status to another would give rise to the Declare protocol. In a parking en-

vironment this would give rise to two declarations: one from driving to parking and one from parking to driving, which can be associated with a parking fee. Of course, the OBU would somehow need to know how the status of the vehicle; for parking this could be induced by turning of the vehicle ignition key. If the parking cell is in the checklist then - as before - both declarations will be submitted to the TSP, allowing for inspection.

7 Conclusions

We have proposed a novel protocol for Electronic Traffic Pricing. This protocol has fairly relaxed accuracy requirements for positioning hardware, making it easier to implement in practice, in an open manner. The protocol is resilient to colluding adversaries where the positioning of the checkpoints remain secret, even after the reporting of road usage to the TSP. In contrast to other systems from the literature, this reporting procedure does not require user interaction which would drastically improve its acceptability. Additionally, the current approach only involves interaction with the user when fraud is detected, and not during honest usage.

Acknowledgements

We want to thank Gert Maneschijn, corporate security officer of the Dutch vehicle authority RDW for many stimulating discussions on road-pricing and its security and privacy concerns. The present scheme can be considered as an enhancement of a scheme developed by him and Eric Verheul in 2008.

References

1. Directive 200452EC of the European Parliament and of the Council of 29 april 2004 on the interoperability of electronic road toll systems in the Community, 2004.
2. Commission decision of 6 october 2009 on the definition of the European Electronic Toll Service and its technical elements, 2009.
3. J. Balasch, A. Rial, C. Troncoso, C. Geuens, B. Preneel, and I. Verbauwhede. PrETP: Privacy-preserving electronic toll pricing. In *USENIX Security*, 2010.
4. D. Chaum and T. Pederson. Wallet databases with observers. In E.F. Brickell, editor, *Advances in Cryptology: Proceedings of Crypto'92*, pages 89–105. Springer-Verlag, London, 1985.
5. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17:281–308, April 1988.
6. Jaap-Henk Hoepman and George Huitema. Privacy enhanced fraud resistant road pricing. In Jacques Berleur, Magda Hercheui, and Lorenz Hilty, editors, *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience*, volume 328 of *IFIP Advances in Information and Communication Technology*, pages 202–213. Springer, 2010.

7. W. de Jonge and B. Jacobs. Privacy-friendly electronic traffic pricing via commits. In P. Degano, J. Guttman, and F. Martinelli, editors, *Formal Aspects in Security and Trust*, number 5491 in Lecture Notes in Computer Science, pages 143–161. Springer, Berlin, 2009.