Privacy-friendly Electronic Traffic Pricing via Commits*

Wiebren de Jonge
TIP Systems BV, and
Vrije Universiteit Amsterdam
wiebren@cs.vu.nl

Bart Jacobs
Inst. for Computing and Information Sciences
Radboud Universiteit Nijmegen
bart@cs.ru.nl

November 11, 2008

Abstract This paper introduces a novel approach or architecture for fraud-resistant and privacy-friendly Electronic Traffic Pricing (ETP). One salient contribution is that it can satisfy the seemingly incompatible requirements of a privacy-friendly and so-called "thin" solution.

The proposed approach relies on regularly sending to the traffic Pricing Authority (PA) only hashes of travelled trajectories and hashes of the corresponding fees due. This makes it possible to achieve that users keep almost all data on the trajectories they travel and on the amounts they should pay completely hidden from the PA, without having to rely for their privacy protection on a so-called Trusted Third Party (TTP). Only a very small percentage of all these privacy-sensitive data requires that the pre-image trajectories and pre-image fees are revealed to the PA for spot-checking purposes (to detect cheating).

The calculations of the amounts due for trajectories travelled can be done—at desire—inside or outside the vehicle. Thus, seamless integration of "thin" and "thick" in one ETP system with one and the same spot-checking approach is made possible and easy. The calculations can be performed in a privacy-friendly way, since they do not require any vehicle or On-Board Equipment (OBE) identification.

The proposal can, for example, be used as a declaration-based approach much in line with current tax declaration traditions in which the individual citizen is personally responsible. However, the proposal allows for much individual variation (including delegation) and many additional (commercial) services. For example, it is also possible to reduce user responsibility and/or user involvement to an absolute minimum.

1 Introduction

After years of discussion the Dutch government has decided to introduce distance-related Electronic Traffic Pricing (ETP) for all vehicles on all roads by means of modern satellite technology, such as GPS or Galileo. Particularly the inclusion of personal vehicles, requiring an appropriate level of privacy protection, and the choice for time, location and vehicle category dependent kilometre tariffs make this approach ambitious and new in the world (see also [3]). For each individual vehicle detailed time and location information must be collected and processed without endangering privacy. The correct amounts due can be calculated with the help of a digital tariff and/or road map. Now and then—for example, once per three months—the total amount due for the period (the "fee") in question must be revealed to the Pricing Authority (PA) and then collected. Clearly, the shorter these fee reporting periods, the greater the impact on privacy.

In the Netherlands, this new ETP should replace—in about five years time—the current (flat) road tax and the existing special purchase tax for personal vehicles and motorcycles. The main aims of introducing ETP are:

- fairness: the fee one has to pay will depend on one's actual vehicle use;
- congestion reduction: traffic supply can be influenced via flexible pricing policies;

* Version of the Springer LNCS proceedings of the Workshop *Formal Aspects in Security and Trust* (FAST), Malaga, oct. 2008.

– environmental impact reduction: kilometre tariffs will partly depend on (environmental) vehicle characteristics.

The techniques for such a form of ETP, like GPS and GSM, are all available. The challenge is to integrate them in such a way that the system will be reliable, privacy-friendly, cost-effective, transparent and easy to use, and will allow easy enforcement and dispute resolution. It may be expected that some of the intended users of the system—drivers / holders / owners of vehicles that are registered in the Netherlands—are hostile users and may try to obstruct or abuse the system. At the same time, the system should be trusted, by the various stakeholders involved.

This paper is not about general requirements for ETP, but focuses on privacy and security aspects. So far this topic has received relatively little attention in the computer security community. Our aim is to design a system that is both secure and privacy-friendly, in which privacy is not treated as a *post hoc* add-on, but as an essential property that needs to be built deeply into the architecture of the system. We adopt Mitch Kapor's slogan "architecture is politics" (see *e.g.* also [6]) and wish to design ICT-systems in such a manner that individual autonomy and control over one's own user data is offered and can be ensured, contributing to public trust in the system. After all, centralised informational control supports centralised societal control. This is a highly relevant issue, also in ETP.

This paper presents only the main lines of a novel solution and is organised as follows. Sections 2 and 3 give an informal introduction to the issues in this area via two possible solutions, as opposite extremes. Sections 4 and 6 describe the main ideas of the proposed solution and protocol essentials. Section 5 discusses cryptographic techniques used. Sections 7 and 8 discuss some advantages and possible use scenarios. Finally, Section 9 discusses the proposed solution from a broader perspective.

The main idea of this article is due to the first author (WdJ), see also [4]. The current elaboration and presentation is the result of joint work.

2 Context

For ETP, vehicles will contain so-called On-Board Equipment (OBE). What this OBE should do precisely depends on the architecture chosen, but we assume that it can at least:

– determine its own location, *e.g.* via a Global Navigation Satellite System (GNSS), such as GPS or (in the future) Galileo;
– communicate with the outside world, *e.g.* via GSM or WiFi on specific locations;
– store information locally and perform elementary computations.

One must take into account many aspects, of which we mention only a few here. First, no physical protection measure can prevent a user from sending false signals to the GNSS receiver in a vehicle or from blocking the true signals originating from the navigation satellites. Second, the OBE should not only do the right things, but also be prevented from doing any wrong things, like surreptitiously leaking location data, *e.g.* via a hidden/covert channel. Third, frequent data transmission from the vehicle may endanger privacy.

Although the OBE must satisfy certain minimal requirements, it can vary much in type and in additional functionality offered (see Sections 7.1 and 8). We call the OBE:

– "fat" or "thick" when it performs itself the calculation of the fees due[1] for registered road use;

[1] Actually, it might be better to use the more general term 'usage' instead of the more specific 'fee due', since usage can also be expressed in other ways, for instance as readings of one or more counters that each represent the cumulative number of kilometres travelled in a certain category. For example, one might use three categories: 1) 'outside rush hours' or 'low price', 2) 'during rush hour in a moderately congested area' or 'normal price', and 3) 'during rush hour in a highly congested area' or 'high price'. For simplicity and without loss of generality, we will focus on the case of fee calculation and not explicitly treat the very similar case of usage calculation.

– "thin" if this calculation is performed outside the vehicle (by another device or organisation).

Thin OBE must be trusted by the parties involved to register correctly. Fat OBE must additionally calculate correctly. Both are sensitive operations.

In our model we also assume that there is a (traffic) Pricing Authority (PA) that collects relevant information in its back office and takes care of the collection of fees. This PA may be subdivided, but is, for our purposes, best regarded as a single unit. We assume that the (national, road tax) authorities are responsible for the PA.

We also assume that there will be an open standard for the representation of "Traffic data Parts" or "Trajectory Parts" (TP). In this text a TP is an elementary data structure with location data aggregated to a path of a certain duration (in our examples: 1 minute), comprising a number of positions (*e.g.* 61; one per second, including an endpoint) together with a time stamp marking the time of the first position.

Road use fees will be calculated on the basis of the relevant TPs. The process to collect payments and the precise (internal) organisation of TPs are not relevant for this paper.

3 Two extremes

In order to further set the scene we shall sketch in this section two possible architectures for ETP. We shall call them "centralised" and "decentralised". This aspect of (de)centralisation refers to the place where the actual location data of vehicles will be stored: in the back office of the PA or in individual vehicles. In general, central storage implies that individuals loose control over their location data. For example, at a certain moment these data could be made available for marketing and surveillance/datamining (*e.g.* for criminal investigations). Hence, in the end the choice between central or decentralised storage is a political one, involving societal issues of power and control. Here we focus on the technical aspects.

In the centralised architecture the OBE is thin and all intelligence resides with the PA. The OBE frequently sends, say at least once every day, its collected location data to the PA. At the end of each period, say each quarter year, the PA calculates the total fee due.

This architecture is simple, but also rather naive. It will be unacceptable to many that the PA gets detailed travel information about every vehicle and that the central database with location data is vulnerable. This database will be an attractive target for individuals or organisations with unfriendly intentions, like terrorists or blackmailers. The system administrators who control this database may not always behave according to the rules, voluntarily or unvoluntarily. In short, the main weak point concerns privacy and security.

In this approach one needs to have confidence that the thin OBE registers and transfers all actual road use correctly. The PA may enforce this by "spot-checks" based on observations (*e.g.* photographs of vehicles and their licence plates) made at random locations and times. These observations can be compared with the transferred registrations. A fine can be imposed in case of discrepancy. Notice that these spot-checks can in principle take place without drivers or vehicle equipment noticing. This has advantages, because it prevents drivers/vehicles from notifying and warning each other about where to expect spot-checks.

In the decentralised architecture that we sketch next, we assume that the OBE is fat and thus contains enough intelligence to calculate the fee itself. The main problems with this architecture have to do with the OBE and its complexity. For example:

– The OBE must contain the tariff and/or road map data to perform the calculations. Since these crucial data change over time, there must also be a way to update them both securely and timely. The combination of security and timeliness here is a critical factor involving serious problems.
– The OBE must now also be trusted to make the right calculations. Hence it requires more security measures. For example, the OBE uses a separate communication channel for enforcement of correct road use registration and fee calculation, see below.

– The OBE, and particularly its software, becomes complex. This makes the OBE fragile and requires an option to securely update its firmware.

Clearly, the OBE will be more costly due to extra hardware and software required for the additional functionality and for the additional security measures.

In the decentralised approach the road-side checks involve interrogation of OBE in order to be able to check that the last few registrations and associated fee calculations have been performed correctly. For this request-response communication one usually uses Dedicated Short Range Communication (DSRC). Due to the two-way communication, spot-checking locations can easily be noticed by vehicle equipment, and then automatically passed on as warnings to other vehicles. This has a substantially negative effect on spot-checking effectiveness and thus costs. On the positive side, possible discrepancies—such as between the actual (spot-checked) vehicle location and the vehicle locations registered in the most recent (requested) entries of the OBE—may be observed directly on the spot, and may result in immediate reaction of the authorities at the spot-checking location.

As extremes, we are thus faced with a simple centralised solution that is highly privacy-unfriendly and vulnerable to data abuse, and with a complicated and fragile decentralised solution that offers good privacy protection, at least potentially (if well-designed and well-implemented). Our novel approach makes it possible to integrate 'fat' and 'thin' and also to combine the best of these two approaches. It can offer good privacy protection, even when realised with thin OBE, and it makes it possible to keep many advantages of the thin approach, even when choosing for fat OBE. In particular, decentralised and 'thin' do not conflict anymore. Hence, the strong relations suggested (by others and in our text above) between centralised and 'thin' and between decentralised and 'fat' are no longer valid.

4 Underlying ideas

The solution of this paper depends on a number of basic ideas and observations.

– The basic traffic data registration (*i.e.* the TPs) can be protected against fraud by using 'non-revealing' commits and remote spot-checking (*i.e.* remote from the vehicle). Indeed, commits can be performed without revealing any (privacy-sensitive) data contents. For example, by sending to the PA only the results of hashing the data with a secure hash function. Such non-revealing commits can also be used for committing to fees calculated. Thus, it is not necessary to reveal any privacy-sensitive data at first.
– Based on a remote (*e.g.* road-side) observation of a vehicle, the vehicle's OBE (or the user's PC or a party enlisted by the user; see Section 7.1) must later reveal the actual data concerning a short period around the time of observation. Note that these actual data (*i.e.* the TPs) are the pre-images of the hash values that have been transferred to the PA earlier. In other words, cheating can be detected. All in all, the only privacy-sensitive traffic and fee data that must be revealed to the PA are those involved in a spot-check. In fact, the privacy-sensitive data to be revealed for a vehicle can be limited to a very small percentage (*e.g.* $< 1\%$ or even $\ll 1\%$) of all fee and traffic data. Note that one can still apply to the spot-checking process many usual (or, say, 'more traditional') privacy protection measures in order to protect even this very small percentage as much as possible.
– The identity of a vehicle or of OBE involved is not required for calculating the fee due for a trajectory part (TP). Hence, traffic fee calculation can be done anonymously.
– Traffic fee calculations can be done anywhere (inside or outside the vehicle) and even by parties not trusted by the PA. For example, calculations can be performed by the vehicle user's PC or by one or more parties enlisted by the vehicle user. That parties not trusted by the PA can be used for the fee calculations stems from the fact that the fee is derived information. If the basic traffic data, *i.e.* the TPs, are protected against fraud, then it is easy to check later whether calculations have been performed correctly.

– Non-revealing fee commits can be organised in such a way that the PA only needs "local" spot-checks to convince itself of the correctness of the total fee reported. Spot-checks to verify the correctness of "subfees" calculated for individual TPs and spot-checks to verify that subfees committed are also included in the total sum reported.

5 Background about hashes

A (secure) hash is a function that turns a digital message of arbitrary length into a garbled message of fixed length (usually 160 or 256 bits). This output value is called the hash (value) of that message. Other names are 'digital fingerprint' or 'message digest'. Hashing is a basic operation in cryptology and computer security and is described in any textbook (see *e.g.* [7,5]). A (secure) hash function, usually written as h, has two basic properties:

– it is not feasible, given only an output value $v = h(m)$, to find the "pre-image" m;
– it is not feasible, given a message m, to find a different m' with $h(m') = h(m)$.

However, if a value v is given (first) it is easy to check that it is the hash value of a (later) given message m, simply by calculating $h(m)$ and checking if $v = h(m)$. A hash value $v = h(m)$ is thus a bit-pattern that is closely related to its pre-image m, but keeps (the contents of) m excellently concealed. There are standard implementations for such a function h, such as SHA-256. But here we shall abstract from such concrete functions and shall simply write h for an arbitrary secure hash function.

5.1 Use of hashes for commitment

Hashes (*i.e.* results $h(m)$ of hash function applications) can thus be used for early commitment to a piece of information without revealing its contents. In our context, this can be explained in more detail as follows. Suppose the OBE of a vehicle sends to the PA at time t_1 the hash value $v = h(m)$ of a certain piece of information m (*e.g.* a trajectory part TP or the subfee due for a TP) that is confidential in the sense that the OBE (or the vehicle's user) does not wish to reveal it to the PA, at least not without the need to do so. Furthermore, suppose that at some later time t_2 this OBE must reveal the piece of information m (*i.e.* the pre-image of v) to the PA for spot-checking purposes and does so by sending to the PA bit-pattern x pretending that x is exactly the same as the bit-pattern m committed earlier at time t_1. Then the PA can easily verify whether this is really true (*i.e.* that the PA is not cheated) by computing $h(x)$ and checking whether indeed $h(x) = v$. Thus, when spot-checked by the PA (say at time t_2) the OBE or vehicle user cannot cheat the PA by sending a message (*e.g.* trajectory or fee) different from the one committed earlier. In other words, as soon as the PA has received the hash $v = h(m)$, the message m (and thus its information contents) becomes 'frozen' and 'irreversible' (more or less: unchangeable/immutable).

5.2 Omission of cryptographic details

Finally, we warn the reader that in our presentation many details are omitted, including many cryptographic details. For example, if party A must supply hashes of confidential bit-patterns to B with a very short maximum length (in our context *e.g.* the fee due for a TP), then A should first concatenate a fresh random number to each original bit-pattern[2] in order not to endanger its secrecy. The incorporation of a random number in the pre-image prevents the receiver B from being able to construct a 'deciphering' table by brute force, that is, by computing the hash of all possible pre-images.

[2] Another detail omitted is that A has to keep the relationship between the original short bit-pattern and the random number, because otherwise A cannot reveal the correct pre-image later on.

6 Approach and protocol essentials

This section will elaborate some technical details in order to explain the essence of the proposed approach. We shall concentrate on the main lines, which are actually quite simple. Several variations are possible, some of which will also be discussed. We shall at first assume minimal OBE as described in Section 2, which can only determine its own location, communicate with the traffic Pricing Authority (PA), and store Trajectory Parts (TPs).

6.1 Road use reporting & verification

In the approach proposed, commit messages must be sent to the PA regularly. Here we assume that the OBE of each vehicle (say, with identifier $\mathsf{veh\text{-}id}$) daily sends a commit:

$$\mathsf{OBE} \longrightarrow \mathsf{PA} : \langle \mathsf{veh\text{-}id}, \mathsf{day}, \mathsf{hash}_{\mathsf{day}} \rangle \tag{1}$$

where the "hash of the day" is a two-level nested hash defined as the hash of $24 \times 60 = 1440$ concatenated hashes of one minute length trajectory parts, *i.e.*:

$$\mathsf{hash}_{\mathsf{day}} = h\big(h(\mathsf{TP}_{\mathsf{day},1}) \parallel \cdots \parallel h(\mathsf{TP}_{\mathsf{day},1440}) \big) \tag{2}$$

Notice that (1) is a very short message, typically in the order of 40 bytes, that completely freezes a vehicle's movements and whereabouts (*i.e.* parking and/or travelling) of a particular day (indicated by the variable day) without revealing anything about the actual vehicle locations (the contents of the TPs of that day). The OBE stores all these trajectory parts $\mathsf{TP}_{\mathsf{day},i}$ forming the pre-images of the hash function h. It does so for all the reports it sends, until it can safely drop them (see Section 6.4).

It is important to understand that the PA can use observations for spot-checking the underlying book-keeping. Suppose that the PA has legal proof that a specific vehicle has been at location ℓ between 8:42 and 8:43 AM on February 13th (*i.e.* in minute 523 of day 44). Within some reasonable period after that day the PA can demand that both the pre-image (say, x) of the (outer hash of) hash_{44} and $\mathsf{TP}_{44,523}$ (say, y) must be sent in. After receiving x and y, the PA verifies:

- whether x really corresponds to (*i.e.* is really the pre-image of) the fingerprint hash_{44} earlier received as commit (2);
- whether y indeed corresponds to (*i.e.* is really the pre-image of) the 523rd fingerprint present in x—using that hashes have a fixed length;
- whether the trajectory data in y is in correspondence with the observation, that is, whether location ℓ is covered by trajectory part $y = \mathsf{TP}_{44,523}$.

If all three verifications are successful, then the book-keeping regarding the whereabouts in said minute, as frozen at the time of commit, is in accordance with the observed reality. If one of the three verifications fails, this indicates a possible fraud attempt. Of course, more investigation may be needed to exclude certain exceptional causes, such as an equipment failure that has been reported earlier (and in accordance with the rules). We will not digress on such issues further.

Reasons for using the nested, two-level hash. In the next few paragraphs we digress on the two-level fingerprint $\mathsf{hash}_{\mathsf{day}}$ as described in (2). Instead of this nested hash, one could simply transfer the fingerprint of the concatenation of all TPs of the day in question:

$$h\big(\mathsf{TP}_{\mathsf{day},1} \parallel \cdots \parallel \mathsf{TP}_{\mathsf{day},1440}\big) \tag{3}$$

However, then a spot-check based on car-location-time evidence would require revealing all TPs of the day in question. Obviously, this would make privacy protection considerably worse. So, our main reason for using nested hashes is the considerably better privacy

protection that can be achieved without changing to a higher frequency of sending commit messages to the PA.

A second reason is that the spot-checking as described—the spot-checking based on two-level hashes (2)—requires less data to be communicated. For, the pre-image x of hash $\mathsf{hash}_{\mathsf{day}}$ consists of 1440 hashes while the pre-image would consist of 1440 TPs in case of a single-level hash (3) of all TPs of the day in question. Assuming hashes of 32 bytes (256 bits), the 1440 hashes take up 45 KByte. Assuming the 61 positions in a trajectory part require an average of four bytes each, the 1440 TPs would require about 340 Kbyte.

A third reason is that one might use the hash of each TP to improve fraud resistance or to reduce the intensity of the spot-checking required, particularly by storing these inner hashes $h(\mathsf{TP}_{\mathsf{day},i})$ given in (2) more or less safely into an Authority's Trusted Element (ATE), inside the OBE. If such is done, we say that the inner hashes are used for "internal commits", while the outer hash given in (2) is said to be used for "external commit". Of course, the degree of safety offered by internal commits depends on the quality of the ATE's physical protection and will never be 100%.

Actually, the above three reasons explain why the first-level (bottom-level or inner) hashes are present, but do not explain yet why also the outer hash is used in (2). For, one also could drop this outer hash and simply transfer the concatenation of the hashes of all TPs of the day in question. However, the concatenation of 1440 hashes takes up 1440 times the number of bytes of one hash. Thus, the outer hashes are only present in order to reduce the size of the commit messages. Indeed, this comes at the price of having to (request for and) transfer during each spot-check an extra pre-image consisting of the concatenation of the (in our example: 1440) hashes. But spot-checks are performed for only a small percentage of all commit messages, so the net savings are considerable. In short, the outer hashes are there for efficiency reasons, that is, for reducing the communication costs.

6.2 Fee calculation

The subfee due for each individual TP (trajectory part) can be calculated by publicly available software that uses a publicly available tariff and road map. This software may be run on the user's own PC or on computers of many independent Calculation Service Providers (CSPs), that is, organisations offering such calculations as a service. CSPs only have to run the calculation software and are supposed to prevent that this software—which may have been produced and distributed on behalf of the PA—leaks in some way any information to the PA or to others in the outside world. CSPs do not have to be trusted by the PA. Of course this software may also be run inside fat OBE.

The crucial point regarding privacy protection is that fee calculation need not involve any identity. Actually, one can organise things such that even the vehicle's category does not have to be revealed to the CSP.

Sending a TP to a CSP and then receiving back the subfee due can be done via a number of anonymity guaranteeing servers. (See *e.g.* Chaum's mixes [1]). If one fully trusts one particular CSP—one's own PC may act as such—all subfee calculations can be performed by that particular CSP. However, one can also organise that for each TP the CSP to be used will be chosen randomly from a set of independent (less trusted or even non-trusted) CSPs. Here we assume that 'dossier linking' (*i.e.* conspiracy) between a CSP and the PA via the hash of each TP will be hindered by a little trick/variation: for committing a particular TP one sends to the PA the hash of that TP concatenated with a random number. All in all, privacy can be protected as long as a sufficient percentage of the chosen CSPs do not cheat. More countermeasures exist, but are outside the scope of this article.

6.3 Fee reporting & verification

In order to enable the PA to collect payment, for each vehicle the total traffic fee due must be reported regularly, but—for privacy reasons—not too often. Here we assume that the

OBE quarterly sends a fee report:

$$\mathsf{OBE} \longrightarrow \mathsf{PA} : \langle \mathsf{veh\text{-}id}, \mathsf{quarter}, \mathsf{fee}_{\mathsf{quarter}} \rangle \tag{4}$$

The PA must be able to check for each vehicle that a) subfees of individual TPs (*i.e.* $\mathsf{fee}_{d,i}$) have been calculated correctly, and b) all these subfees add up to the reported total fee (*i.e.* $\mathsf{fee}_{\mathsf{quarter}} = \sum_{d \leq N \,\&\, i \leq 1440} \mathsf{fee}_{d,i}$where N denotes the number of days in the quarter). These checks must be carried out in a privacy-friendly way, revealing as few subfees (and subtotals) as possible. After all, subfees (and subtotals) show a little bit about an individual's behaviour, for instance whether or not the vehicle has been used or not. There are several possible ways to organise such fee reporting and verification. For illustrative purposes, we will first sketch an interactive way with a game-theoretic flavour. Then we will sketch our main solution using non-revealing commits via hashes. Finally, we will suggest possible use of homomorphic encryption for the hashing.

Interactive verification The PA may communicate as follows with an owner of a particular vehicle (or with a software agent acting on this owner's behalf).

- The PA says: "well, so your quarter amount is $\mathsf{fee}_{\mathsf{quarter}}$". Tell me the three amounts of the months that are in this quarter. Of course, the owner should produce amounts that add up to $\mathsf{fee}_{\mathsf{quarter}}$.
- The PA then picks one particular month from this quarter and proceeds to ask the amounts for the weeks in that month. Again they should add up correctly.
- Now the PA picks one particular week from the chosen month and asks the amounts for the days in that week. Again they should add up correctly.
- The PA continues to ask the amounts of the four quarters of a day, picks one, asks for the six hour amounts of that quarter day, picks one hour, asks for the four quarter (of an hour) amounts of that hour, picks one quarter, and asks for the three five-minute amounts of that quarter, and finally picks one five-minute period and asks for the minute amounts of that period. Of course the questions of the PA are organised in such a way that the pre-chosen day-minute pair (day, i) is in this five-minute period.
- Now the PA asks for $\mathsf{TP}_{\mathsf{day},i}$ and for the pre-image of the "hash of the day" as described in (2). The PA performs the checks from Section 6.1 to verify that $\mathsf{TP}_{\mathsf{day},i}$ is indeed the version committed earlier, computes the fee due for $\mathsf{TP}_{\mathsf{day},i}$ and checks whether this amount is indeed equal to the minute amount reported in the previous step.

By breaking up the path to the pre-chosen day-minute pair in many small substeps the PA learns relatively little about the fees of all other trajectory parts. In this verification method it is essential that the questions are posed and answered interactively, because otherwise the vehicle owner could successfully cheat and adjust amounts outside the path chosen by the PA (which are not checked) so that (sub)totals still add up correctly.

Non-interactive verification via hashes Suppose that during the quarter (see also Section 7.6) the PA also receives for each day d a "fee hash of the day":

$$\mathsf{fee\text{-}hash}_d = h\big(\, h(\mathsf{fee}_{d,1}) \parallel \cdots \parallel h(\mathsf{fee}_{d,1440}) \,\big) \tag{5}$$

Then checking the correctness of an individual subfee $\mathsf{fee}_{d,i}$ is easy and very similar to the spot-checking described in Section 6.1. In this case the PA also asks for both $\mathsf{fee}_{d,i}$ and the pre-image of $\mathsf{fee\text{-}hash}_d$. The spot-check now includes verifying whether the latter is indeed the pre-image of $\mathsf{fee\text{-}hash}_d$, verifying whether $\mathsf{fee}_{d,i}$ is indeed the pre-image of the i-th hash in the concatenated string of 1440 hashes, computing itself the fee due for $\mathsf{TP}_{d,i}$ and verifying that this amount is indeed equal to $\mathsf{fee}_{d,i}$.

However, this is not sufficient yet. For one could cheat by committing correct subfees and reporting a false sum as total fee. Our solution is to change the list of all $h(\mathsf{fee}_{d,i})$ of

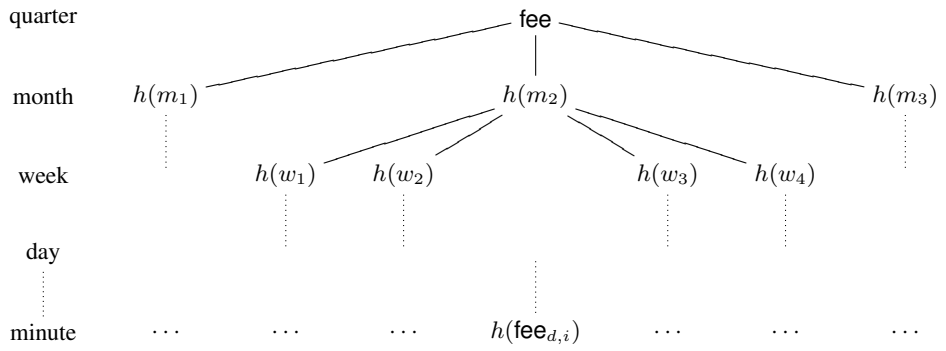
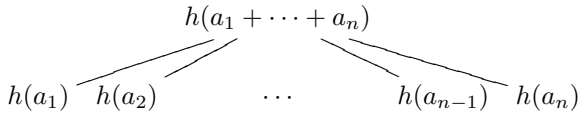


Figure1. Tree representation of hashes of subtotals of subfees, in which for instance the quarterly fee is the sum $m_1 + m_2 + m_3$ of the month amounts, and the (second) month amount m_2 is the sum $w_1 + w_2 + w_3 + w_4$ of the week amounts, *etc.*

a quarter into an "enriched" list representing (in post-order tree walk) the tree of hashes given in Figure 1, which 'freezes' all calculation steps involved. Note that a) the interactive verification that we described above, implicitly also involves a tree structure, and b) our 'freezing' allows the interactivity to be removed. Now the PA can spot-check the summation by selecting and checking a number of "triangles" consisting of an internal node and its children. Hereto one must reveal to the PA the pre-images of the hashes in these triangles of the form:



All in all, the PA can convince itself in a privacy-friendly manner of the correctness of the fee report. The sketched approach for committing (sub)fees via a tree of hashes requires a certain amount of elementary bookkeeping and communication that can be automated easily. It is not difficult to arrange that, for example, the OBE automatically does all the work required (without user involvement, if such is desired).

Possible use of homomorphic hashing Our third approach to fee reporting does not need a tree of hashes, but is computationally more involved. We will only sketch it very rudimentarily. Let G be a suitable finite group with modular multiplication and generator $g \in G$. The discrete log problem refers to the infeasibility of calculating $n \in \mathbb{N}$ when $g^n \in G$ is given. Hence, we can use the function $x \mapsto g^x$ as a homomorphic hash, since $g^x \cdot g^y = g^{x+y}$. The homomorphism property is often useful, for instance in counting protected votes via multiplication in e-voting, see *e.g.* [2]. In a similar way one may use homomorphic hashing in the current setting. Subfees $\mathsf{fee}_{d,i}$ of trajectory parts $\mathsf{TP}_{d,i}$ can be sent to the PA as $g^{\mathsf{fee}_{d,i}}$. The PA can then multiply these hashed values and check that $\prod_{d \leq N \,\&\, i \leq 1440} g^{\mathsf{fee}_{d,i}} = g^{\mathsf{fee}_{\mathsf{quarter}}}$.

There are a number of subtle points that need to be addressed, among which the following. The amounts $\mathsf{fee}_{d,i}$ are typically small numbers that should be "blinded" to prevent that $\mathsf{fee}_{d,i}$ can be obtainded from $g^{\mathsf{fee}_{d,i}}$ by trying a limited number of values. Blinding can occur by multiplying $g^{\mathsf{fee}_{d,i}}$ with $g_0^{R_{d,i}}$, where $R_{d,i}$ is a random value (or actually a well-chosen hash value that also acts as binder) and $g_0 \in G$ is coprime with g. Use of g_0 (instead of g) hinders interfering with the fee by "shifting" between the exponents in the product. The sum R of the random values must be submitted together with $\mathsf{fee}_{\mathsf{quarter}}$ so that the PA can check the equation $\prod_{d \leq N \,\&\, i \leq 1440} g^{\mathsf{fee}_{d,i}} g_0^{R_{d,i}} = g^{\mathsf{fee}_{\mathsf{quarter}}} g_0^R$.

6.4 Confirmations

At various stages a user or his/her OBE needs to receive (digitally) signed information from the PA. For example:

- confirmations of receipt of the trajectory commit messages (2) and of the fee reports that have been submitted to the PA;
- requests for disclosure of certain TPs or (sub)fees;
- clearance messages stating that all book-keeping (such as pre-images/TPs) can be dropped up to a certain day.

Confirmation messages typically involve return messages by the PA comprising a time-stamped and digitally signed copy of the data submitted to and received by the PA. If such a confirmation message does not arrive within a certain time frame, the OBE may notify the user. Clearly, OBEs need to be able to check digital signatures of the PA. This requires that they contain a certificate for the public key of the PA. This public key may also be used to encrypt messages to the PA. However, as already has been mentioned in Section 5.2, such cryptographic details are outside the scope of the current paper.

7 Some properties

This section will explicitly discuss some of the properties of the proposed approach using non-revealing commits. One main point is that this approach makes privacy-friendly (decentralised) and fraud-resistant solutions possible, even when using thin OBE. Another main point is that existing fat solutions can be improved substantially by adding the use of non-revealing commits, thus making a number of advantages of 'thin' (*e.g.* related to spot-checking, costs, monitoring ability and system continuity) also available for 'fat'. For example, fat solutions can be made less vulnerable to compromise of the OBE's physical protection (*i.e.* to tampering). Below we treat several aspects in more detail.

7.1 Wide range of realisation options

The proposed approach allows much implementation freedom. The only two tasks that certainly must be performed in the vehicle are determining the relevant traffic data (*e.g.* trajectory data) and temporarily registering these data piece by piece (*e.g.* per minute) locally. All other work—except the optional 'internal commits' (see Section 6.1)—can be performed, at desire, inside or outside the vehicle.

Clearly, for doing all work (including subfee calculations) inside the vehicle fat OBE is required. In case of minimal work inside the vehicle, the (thin) OBE must transfer the relevant traffic data to equipment outside the vehicle taking care of all other work. This latter equipment may be the user's own PC or the processing equipment of a party chosen by the user. However, thin OBE may also do all work with only the exception of subfee calculations outside the vehicle. The thin OBE then takes care of committing to the traffic data, distributing automatically and anonymously fee calculations to selected Calculation Service Providers, collecting the results, committing to these results, reporting the quarterly fee to the PA and reacting to messages from the PA, such as verification requests. Note that all this processing can be fully automated and can be performed by OBE, if desired.

In short, users have a wide range of OBE options to choose from. Their choice may depend on additional services offered and on how much control they wish to have themselves. That is, on whether they wish to trust one or more other parties and, if so, how much.

7.2 'Thin' and 'fat' can be integrated gracefully

The proposed approach makes ETP systems possible in which some vehicles use thin and others fat OBE and in which the same spot-checking approach is used for all vehicles.

7.3 Simple and effective spot-checks

Spot-checks in our approach can be based on random observations, just as in case of 'conventional thin' (*i.e.* the centralised approach from Section 3). During an observation no real-time communication with the vehicle is required, which greatly reduces the complexity (and costs) of spot-checking. Furthermore, unnoticed spot-checking is made possible, at least during daylight. Without further explanation we mention that unnoticed spot-checks can be much more effective than detectable ones and thus can be used to (further) reduce the spot-checking costs (or to achieve better fraud resistance at the same costs).

The simple and effective observation-based spot-checks can be used to monitor the actual fraud resistance level (see below) and also to replace either all or only part of spot-checks based on real-time interrogation of the OBE (*e.g.* via DSRC). If one chooses for exclusively making use of observation-based spot-checks, one saves the costs for the hardware and software required for the real-time communication channel and gives up the ability to stop a vehicle on the spot immediately after an unsatisfactory interrogation.

An advantage of the proposed approach is that these simple, effective and cost-efficient observation-based spot-checks can also be used for fat OBE.

7.4 Spot-checking and physical protection can work 'in parallel'

In our approach (as well as in 'conventional thin') spot-checks can produce effect even if the OBE is not protected at all against manipulation of trajectory data. Indeed, manipulating the contents of a trajectory data part (TP) does not make much sense as long as that TP is committed (in case of 'conventional thin': transferred to the PA) before the forger could find out at which locations and times the probability of his vehicle having been observed is sufficiently low to make the risk of being caught acceptable. Thus, if the PA manages to keep the times and locations of a considerable part of all random observations secret until the TPs have been committed (in case of 'conventional thin': have been transferred), then fraudulent TPs may be uncovered by spot-checks. In other words, the effectiveness of observation-based spot-checks depends on the amount of increased knowledge that potential forgers can timely acquire on the likelihood of having been observed, but not on the physical protection of OBE (against trajectory data manipulation).

The property just described is very important. It implies that the protection achieved by spot-checking (say, the logical protection) and the physical protection can work 'in parallel' and thus provide for two independent layers of protection. In other words, the total level of fraud resistance is equal to the sum of the fraud resistance achieved by physical protection and of the fraud resistance achieved by spot-checking. This has the advantage that one can get rid of the risks that adhere to full reliance on physical protection measures.

Note that in the decentralised fat approach from Section 3 the logical and physical protection are 'serial', since the (effectiveness of) spot-checking by real-time interrogation of the OBE depends on the physical protection of the OBE.

The 'parallelism' property of the proposed approach (and of 'conventional thin') leads to important advantages. In the following we will treat three such advantages (related to costs, system continuity and monitoring ability). Note that the proposed approach makes these advantages now also available for fat OBE.

Cost optimal balance between spot-checks and physical protection In general, a really high level of physical protection is expensive and in the long run (often) not sufficient to prevent successful manipulation. One problem is that perfect physical protection does not exist. Another problem is that almost perfect physical protection—fully in accordance with the law of diminishing returns—probably results in high or even prohibitive costs. In our context a third problem is that—as far as we know—no physical protection measure can prevent one from sending false signals to the GNSS receiver in a vehicle and/or altogether blocking the true signals originating from the navigation satellites.

If spot-checking and physical protection of OBE work in parallel, then the desired level of fraud resistance can be achieved by a combination of both. This offers as advantage that one can head for a cost optimal balance between spot-checking and physical protection measures. If the marginal costs for additional physical protection measures (required for achieving the last few percent of the required level of fraud resistance) are higher than the marginal costs for the additional spot-checking (required for achieving that same last few percent), then one can choose for increasing the intensity of spot-checking and for not applying additional physical protection measures. And if not, then one can increase the physical protection instead of increasing the spot-checking intensity.

The proposed approach makes this balancing now also possible for 'fat'.

System continuity is less vulnerable Some level of (hardware and software) protection will be used in OBE implementations. For instance, to make manipulation of trajectory data sufficiently difficult. But when that protection gets broken at some stage in the future, this event does not undermine the essence of the system and disrupt it fundamentally, at a large scale. After all, one can temporarily increase the intensity of spot-checking to keep the level of fraud resistance (roughly) intact. As soon as the problems with the physical protection have been solved (which may take quite some time), one can decrease the intensity of spot-checking to an appropriate level.

In short, the system continuity is less vulnerable, since there is less (vulnerable) dependence on physical protection measures. This now also works for 'fat'.

Ability to monitor the actual fraud resistance level Since the observation-based spot-checking that we have described, works independent of the OBE's physical protection, it can be used to monitor the fraud resistance level actually achieved, that is, the real percentage of violators.

Suppose that the PA allows multiple traffic pricing Service Providers (SPs) that each make use of a different type of equipment. Suppose also that each SP guarantees to the PA a certain level of fraud resistance (*e.g.* by physical protection only, or *e.g.* by a combination of physical protection for the OBE and of interrogation-based spot-checking by or on behalf of the SP). Then the PA can use the observation-based spot-checking to monitor in the field whether the SPs really succeed in keeping fraud below the level agreed upon.

The proposed approach makes such monitoring now also possible for 'fat'.

7.5 Privacy and data protection

Sending messages need not be done continually while driving and can be limited to, say, once a day. Thus, one can allow users to influence the moments and places of transmission. This is beneficial for privacy protection, because, for instance, a GSM provider might determine the vehicle's location at the time of transmission.

Furthermore, privacy-sensitive travel and fee data can be stored decentralised, under control of participants, instead of in some massive central database of the PA, where they might be misused in various ways, for instance as result of function creep.

Apart from the total fee due and from the location and fee data involved in spot-checks, no privacy-sensitive data needs to be revealed to the PA or a TTP. This amount of data seems to be optimal (for privacy). Note that spot-checks are always necessary, at least if one does not wish to fully rely on physical protection measures.

Assuming that a certain fixed level of fraud resistance must be achieved, one can reduce the spot-checking—and thus increase the privacy protection—in proportion as one increases the physical protection applied to the OBE.

All this is true both for the decentralised fat approach from Section 3 and for our approach, even when the latter makes use of thin OBE. As a consequence, our 'thin' (which is decentralised) offers important advantages over 'conventional thin' (*i.e.* centralised 'thin').

In order to prevent the PA from spot-checking individual vehicles too much, a limit (*i.e.* maximum) can be set to the number of spot-checks allowed per vehicle per period. Furthermore, the PA can also be kept from asking detailed whereabouts (*i.e.* TPs) of particular vehicles without having a corresponding observation, by obligating the PA to specify in requests for TPs both the time of observation and the location of the observed vehicle. Based on the time and location specified, it is easy (*e.g.* for the OBE) to automatically detect possible abuse attempts by the PA.

7.6 Communication and critical time paths

In case of 'conventional thin', the OBE commits to traffic data parts by transferring them to the PA. As suggested in Section 7.4, this transferring preferably[3] should be performed before users can find out at which locations and times the probability of their vehicle having been observed is sufficiently low to make the risk of being caught acceptable. Similarly, in our approach the committing to trajectory parts is (to a certain extent) 'time critical'.

If a request for details (*i.e.* for a TP) is only allowed if the PA has observed the vehicle at the corresponding time and location (see Section 7.5), then calculating the fee due for a traffic data part and committing to the result has also 'time critical' aspects[4].

All communication and other work afterward is not 'time critical'. Indeed, the communication required for spot-checking is not sensitive for, say, a substantial break down of the communication system, such as a breakdown of several days. Note that much or most of the communication can be done at specific moments or places, when a cheap connection (*e.g.* WiFi) is available, for instance at home or at a fueling station offering such connectivity. All this may be used to reduce the communication costs. We do not wish to discussion the communication costs of different approaches further, because these costs are rather unclear at this stage. For example, they depend much on the type of communication channel(s) used. Furthermore, this issue is not crucial for the purpose of our presentation.

7.7 Individual responsibility

With our approach it is possible (but not necessary) to give users individual autonomy by allowing them to take maximal responsibility. This is to a certain extent comparable to the current responsibility of individual citizens for the submittal and correctness of the contents of tax forms for income and revenue. Indeed, there are some developments that tax authorities support citizens by providing partly pre-filled forms, but in the end the responsibility still lies with the citizen. The role of the state is to (statistically/randomly) check these tax reports, to collect the associated fees and to punish those individuals (or organisations) that do not fulfil their duties.

A system in which the state takes the full responsibility—that is, determines all by itself (without user involvement) the amount of taxes due—is completely different. Citizens then are turned into passive subjects whose behaviour is being monitored almost constantly in order to obtain the relevant data for calculating fees. Such a system may seem more convenient for users, but is definitely also more threatening than the traditional declaration-based one. It constitutes a fundamental change in the balance of power and responsibilities.

In the end it is of course a political decision in which direction our societies are moving. Our approach at least provides a technical basis to uphold individual autonomy a bit longer.

[3] Otherwise, the logical protection (*i.e.* the additional independent layer of protection) will be weak and one must rely (almost) fully on the OBE's physical protection, which we do not advocate.

[4] If ample time is available, one might succeed in acquiring almost complete knowledge on where and when observation teams have been active and then committing to a zero fee for all TPs where the risk of having been observed is negligible. Note that: a) the PA may provide for a sufficient number of unnoticed observations as countermeasure, b) external commits may be seen as less 'time critical' if internal commits (see Section 6.1) are used, and c) internal commits rely on physical protection (just as fat OBE does).

8 Use scenarios: granny, gadget & geek

This section will elaborate, to some extent, three different use scenarios of the proposed ETP approach, which we shall (respectfully) label 'granny', 'gadget' and 'geek'.

'Granny' is well-aware of painful periods in history and is not happy with the idea that others (in particular, the state) know her car movements, but she definitely does not want much ado. She uses computers, in a limited way, but does not (wish to) understand the internal workings. She simply buys a black box that handles everything for her. Our 'granny' chooses for thin OBE that computes and sends the trajectory hashes itself, distributes fee calculations to selected Calculation Service Providers, sends hashes of the results (see Section 6.3) to the PA and also automatically handles the verification requests from the PA. After each quarter the device informs her via a display (or an SMS or e-mail) how much she has to pay for that quarter. On her request, the device will show her other aggregations of fee calculations. For example, the fee due for a particular trip, day or week.

The 'gadget' person does not care very much about his privacy. He is willing to exchange it for extra services. He chooses some organisation that he trusts and that sells fancy car navigation systems (including for instance a car assistance or breakdown service) with embedded traffic pricing functionality. He buys such a device and signs a service contract so that the company will take care of all road fee submissions and checks on his behalf. The device sends his location information (trajectory parts) to the company, which handles the hash and fee submissions and the answers to spot-checks. The company to which he has delegated his road pricing duties thus knows his whereabouts, but offers additional services in return, like safety surveillance and tailored real-time congestion information with personalised suggestions for alternative routes.

Our 'geek' does not trust anyone. She wants a minimal system in her car that only stores trajectory parts and communicates their daily hashes to the PA. She frequently transfers her trajectory parts (pre-images) to her PC, *e.g.* via WiFi or perhaps even via a dump on a USB memory stick or on her Bluetooth cell-phone. She uses open source software to do all the work required. Her software calculates the (sub)fees on the basis of publicly available map information, sends their hashes (see Section 6.3) as well as the fee due for each quarter to the PA via the web, and handles all spot-checking requests from the PA. With every spot-check request concerning a trajectory part, the software on her PC first checks whether the time and location as specified by the PA are correct (see Section 7.5). If not, she asks for the photograph to find out whether this may have been an understandable error of the PA or an abuse attempt. She uses the additional functionality of her software package to keep a personal record of all her travels and can visualise them in Google maps (via Tor). She also keeps them to show to her boss, if needed, to substantiate her occasional reclaims for business trips. Note that a reasonable possibility is that the open source software package and the required map information are produced and published on behalf of the PA, say via a web site.

All these three fictitious individuals fulfil, in quite different ways, the duties associated with a system for ETP as proposed here. It shows that there is ample room for individual variation and for contributions and additional services by commercial organisations.

9 Final remarks

The main idea in this paper is simple and general. It may be described as follows. Consumers use certain 'goods'. Examples are use of transport infrastructure (such as a whole road network as described in previous sections, or toll roads, or parking lots, or a public transportation system) or consumption of, say, electricity. Each consumer's usage is measured by equipment in the consumer's environment, which is 'potentially hostile' for the goods provider. Correct functioning of and reporting by the equipment may be the responsibility of the consumer, of a party chosen by the consumer (*e.g.* an independent equipment

provider), of the goods provider (*e.g.* in certain cases that it also provides for the equipment or parts thereof) or of any combination of these. Our approach is useful and suitable for all these cases of individual or shared responsibility. For the following, let us assume the user is (mainly) responsible. Then the consumer (or actually equipment on his behalf) commits himself to the measurements by transferring hashes (fingerprints) of the measured values to the goods provider (or a pricing authority), while keeping secret the measured values. The measured values (*i.e.* the pre-images of the hashes) are used for the calculation of fees due (for short periods). These calculations can be done separately and in a privacy-friendly way and hashes of their results must also be transferred to the goods provider. Only the total fee due for a longer period (*i.e.* the sum of the fees due for many short periods) is reported to the goods provider in 'readable' form. The goods provider can guarantee fraud resistance by spot-checking in a way similar to what has been described in previous sections.

Underlying such an architecture is a certain view on the organisation of our society in which individuals remain responsible for what they do and their behaviour is not constantly monitored and checked. To make this view more concrete, consider a toll gate, for instance at the entrance of a bridge or of a congestion fee area. The traditional way to organise such a fee is to identify (for instance via license plate recognition or via some DSRC-tag) each vehicle passing by and to charge a fee on the basis of such observations. This is in a sense the most obvious solution. It is rather privacy-unfriendly however, because all passages of individual vehicles are—at least temporarily—registered in some database (of the authority in question) that is open to various forms of secondary use. A different solution, in line with the approach presented in this paper, is the following. The gate constantly broadcasts messages of the form "you are passing this-and-this gate at this-and-this time and this-and-this tariff table must be applied", which are recorded by the OBE of vehicles that pass by. The OBE of these vehicles regularly transfer hashes of these records to a central authority and also hashes of the fees due for such passages. Vehicles may be photographed now and then in order to randomly check the correctness of the total fee reported for a longer period. Thus, only a small subset of all passages is recorded (temporarily) by the authority.

Which approach do you prefer? In the end this is a societal issue. This paper provides a technical framework for more privacy-friendly (but also more fraud-resistant) solutions than are currently being employed.

Acknowledgments

Thanks are due to Eric Verheul, Michel Oey, Gert Maneschijn, Gerke Paulusma, Gerhard de Koning Gans, Erik Poll, Jaap-Henk Hoepman, Marko van Eekelen and Engelbert Hubbers for helpful discussions and suggestions.

References

1. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), pages 84–88, 1981.
2. R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In W. Fumy, editor, *Advances in Cryptology — EUROCRYPT'97*, number 1233 in Lect. Notes Comp. Sci., pages 103–118. Springer, Berlin, 1997.
3. S. Eisses, W. de Jonge, and V. Habers. Privacy and distance-based charging for all vehicles on all roads. In *Proceedings of the 13th ITS World Congress*, 2006. Available from URL: `www.rapp.ch/documents/papers/Privacy_and_RUC_ITSLondon-doc.pdf`
4. W. de Jonge. Kilometerheffing op basis van elektronische aangifte. *Informatie*, pages 22–27, 2003. Available from URL: `www.cs.vu.nl/~wiebren/TIP/ArtikelInformatie.pdf`
5. C. Kaufman, R. Perlman, and M. Speciner. *Network Security. Private Communication in a Public World*. Prentice Hall, 2002.
6. L. Lessig. *The Future of Ideas*. Vintage, 2001.
7. B. Schneier. *Applied cryptography*. John Wiley & Sons, 2nd rev. edition, 1996.