Automated Traffic Enforcement Which Respects “Driver Privacy”

Andrew J. Blumberg
University of Chicago
Mathematics Department
blumberg@math.uchicago.edu

Lauren S. Keeler
University of Chicago
Anthropology Department
lkeeler@uchicago.edu

abhi shelat[†]
IBM Zurich Research Lab
Rüschlikon, Switzerland
abs@zurich.ibm.com

***Abstract*— At many intersections in downtown Los Angeles, cameras take pictures of the license plates of vehicles which run red lights. This information is used to automatically mail tickets to offenders [1]. It seems likely that such systems will become increasingly widespread in the near future, as they are perceived to be effective in cutting down on traffic infractions, boosting governmental revenues, and freeing law enforcement officials for other tasks. However, a network of cameras poses dramatic threats to the privacy of motorists — if kept continuously running, such a network could be used to track detailed driving habits of specific individuals. Speculative proposals involving state-monitored GPS transmitters in every registered vehicles pose even greater threats to “driver privacy” [2]. This situation is an example of the standard tension between citizen privacy and law enforcement.**

In this paper, we discuss the design of a novel camera-free protocol for traffic monitoring involving EZ-pass-like transponders. Our system uses cryptographic algorithms to guarantee that the state is able to detect and identify violators of a wide variety of traffic laws (light violations, speeding, illegal turns, and so forth) *without* having the ability to track the movements of motorists. That is, we design a traffic monitoring system with the property that no matter how it is misused, it is impossible to reconstruct the driving paths of specific cars. On the other hand, the system also guarantees that specific information about motorists who commit traffic infractions can be recovered. In addition, this protocol can be easily adapted for sophisticated and nuanced anonymous toll-collection. The system is fairly easy to implement and is immune to a variety of attacks and cheats.

I. Introduction

In the next decade, we predict there will be increasing pressure for municipal governments to monitor traffic on public roads. In England, such monitoring is already carried out in many places via cameras. Camera-based systems are also beginning to be adopted in American urban areas, and have become essential components of automated toll collection methods as well as automated traffic law enforcement devices [1]. Massive expansions of these systems are already under discussion, including systems involving the use of GPS tracking devices in every registered vehicle [2].

The prospect of ubiquitous surveillance of public driving spaces makes it important to study and develop a notion of acceptable motorist privacy. There are now reasonably comprehensive ideas about informational privacy, developed both from a legal standpoint and also from a cryptographic perspective. However, little work has been done on characterizing how these should apply in the context of driving.

Such characterizations are hampered to some degree by the fact that the privacy of a motorist is a somewhat ill-specified notion. The implicit expectation of privacy depends on a tacit assumption about the density of observers — after all, nothing prevents a police officer from trailing a particular vehicle on the roads. Moreover, the standards of suspicion necessary to stop and search a vehicle are much more lax than those required to enter and search a private residence.

In this paper, we wish to articulate a notion of transportation privacy which allows municipal governments to perform automated traffic law enforcement and toll collection while maintaining the existing “implicit privacy” of the motorist. In fact, we hold our protocols to a considerably higher standard than the police are currently held to. Specifically, we attempt to characterize protocols which permit a regulator to collect precisely the information it needs to enforce traffic laws, but no more. In particular, it is not possible to use the information collected by our system to reconstruct the specific paths of a given vehicle.

The first and most central piece of information of value in this discussion is the *location* of a particular vehicle at a particular time t. However, merely protecting the location of a vehicle is not enough. We must also safeguard *partial information* about the location during a timespan $[t_1, t_2]$. For example, we want also to keep private information that “a particular vehicle has crossed an intersection near the owner’s house an odd number of times between 9am and 2pm” since such information indicates that the vehicle owner is not home at 2pm. In formal terms, partial information about a vehicle can be modeled by considering functions of the location of a vehicle and the time. Thus, we must also seek to limit the class of functions which can be learned by the system.

On the other hand, unlike many standard cryptographic protocols, the context of driver privacy requires issues of forced disclosure to be addressed. We accept on both legal and social grounds that the state has reason and right to inspect and register every vehicle on public roads, and that it is necessary for a police officer to be able to determine the registered owner of a given vehicle at any time. Moreover, we stipulate that a vehicle which commits an infraction must be compelled to temporarily surrender some of its privacy — we must be able to record the location and registration information of such a vehicle, perhaps by automatic device. In general, these surrenderings of privacy must not depend on the discretion of the vehicle (or vehicle owner)— rather, it must be possible to compel externally the release of that vehicle’s information.

[†] This work was completed while at MIT CSAIL in Cambridge, Massachusetts.

Of course, what we are describing here is the classic tension between a desire to maintain complete privacy for the individual and a desire to enable law enforcement officials to have complete information in order to aid in criminal investigations. Our goal is to strike a balance between these two competing needs in a way which cannot be abused either by motorists or by the state.

Combining these considerations, we can formalize the notion of driver privacy using a standard cryptographic tool: the *idealized* model [3]. We first consider an *ideal* model of the world, in which, with the aid of a trusted external party, we can easily argue that driver privacy is maintained and simultaneously, law enforcement is possible when an infraction occurs. Next, we present a *real world* implementation with no trusted party in which the same functionality is captured. Finally, we argue that the two worlds are *similar* in that sense that it is computationally infeasible to distinguish an interaction in an ideal world from one in the real world. Informally, what this shows is that the *real world* implementation does not leak any information that the *ideal work* implementation does not leak (since otherwise, the two worlds *would* be distinguishable). In this short abstract, we defer from providing a formal definition and formal proof of this type of security, and merely present an informal ideal world interaction as the golden standard for privacy and correctness.

II. Ideal Privacy

The goal of this section is to capture the semantics of the system that we want to build. We reiterate that the following is an idealized model, and does not reflect how our actual protocol works since there are no trusted parties in the real world. By focusing on the semantics of an ideal system, it shall be easier for us to identify potential problems with any real implementation of the system.

In the *ideal system*, there are three parties: a vehicle V, a traffic enforcer, D, and a trusted party T. In the beginning, each vehicle registers its identification id with the trusted party T, which forwards the information to D. Then, D registers a set of locations, $S = \{s_1, \ldots, s_n\}$ where it wishes to place a traffic sensor. A traffic sensor is a program which takes as input, a vehicle, a location, and a speed, say, and determines whether the vehicle is abiding by the traffic laws or not. When a vehicle uses a road, it gives the trusted party T its identification id, its location l , and its speed v, at every point of its path. The trusted party determines if the path intersects a sensor s_j, and if so, it evaluates $s_j(id, l, v)$ and notifies V. If the sensor indicates a violation has occurred, then the trusted party informs the vehicle that an infraction has occurred, and asks whether the vehicle would like to disclose its identity, or forever forfeit its right to use the roads. If the vehicle agrees to disclose its identity, then the trusted party sends V's identity and location to the traffic enforcer, D. Otherwise, it sends D the location at which the violation has occurred, and never accepts a request to use the road again from V. Finally, at the end of each day, T publishes the number of violations which have occured that day.

The above game illustrates the semantics of privacy and correctness which we hope to provide. Notice that D learns the identity of vehicle V only after the vehicle *detectably* violated a traffic law. Notice that a vehicle may refuse to acknowledge an infraction which it has committed, but upon refusal, it also forfeits its right to use the public roads.

III. Infrastructure and protocols

In the next sections, we describe our actual protocol for traffic monitoring. Our system should intuitively capture the semantics of the ideal system described above.

There are two components to our design for a secure privacy-preserving protocol for traffic enforcement. The first important component is a rejection of cameras in favor of radio transponders, modeled on the EZ-pass transmitters in common use in the northeastern United States.[1]. Extensive camera networks are simply not compatible with the kinds of privacy we demand since they collect too much information. If misused, they can provide adequate data for real-time tracking of vehicles. Relying on assurances that the cameras only record infractions is not a satisfactory solution.

The second component in our design is the use of cryptographic algorithms (e.g. to provide secure signatures and encrypted channels) as a foundation on which we can base protocols that offer strong security guarantees.[2]

A. An intuitive sketch of our protocol

The core idea of our solution is reasonably simple. Currently, when an individual registers a car they receive a license plate. Automobile registration is authenticated using official identification of the individual as well as a vehicle identification code. Subsequently, when the individual commits a traffic infraction, their license plate number can be recorded and a ticket issued. A privacy concern arises from the fact that a ubiquitous network of cameras can record the license plate all the time, not just when an infraction occurs, and thereby track a vehicle's motion.

Our proposed solution bootstraps from the same procedural base. An individual goes to register a car. When they do so, they commit to a very long list of “digital license plates”. Note that the list itself is not disclosed to the registering authority — only a ”digital commitment” to the list is revealed. The individual purchases a radio transponder (akin to an EZ-pass transmitter) at a consumer electronics store and places it in the registered vehicle. This transponder is programmed with the list of “digital license plates”. The transponder cycles through the list of license plates, maintaining a different one as the “current license plate” each second. When the motorist runs a red light, a logging device at the intersection requests the current license

[1] Note however, although the EZ-pass system uses radio transponders, it does not preserve driver privacy. Our transponders are somewhat more sophisticated

[2] Our constructions rely on making computational assumptions about the difficulty of certain kinds of problems, like factoring large integers. These are standard assumptions that have been well-studied.

plate from the transponder. In response to this query, the transponder sends the current number to the logging device and alerts the driver of the vehicle that an infraction has been recorded.

Since the motorist is rapidly cycling their digital license plate number, it is not possible for the network of loggers to track the motion of the motorist through the city. At the end of the year, in order to renew their registration, the motorist must engage in a special protocol with the registering authority to compute the intersection of the motorist's list of digital license plate numbers and the authority's list of offending numbers which have been collected. This special protocol enables this computation to be performed in a "zero-knowledge" fashion, so that the motorist does not disclose any information about their list of numbers other than those in the intersection.

Of course, in order to make this work in a fashion which is resistant to tampering on either side, there are a number of cryptographic techniques that must be used. We have alluded to some of them — for instance, the digital commitment the motorist discloses at the beginning of the year prevents her from changing her list later or disavowing her list of numbers at the end of the year. Similarly, we require "zero-knowledge proofs" to perform the computation at the end of the year. We begin the formal description of our protocol by reviewing the cryptographic supports we require.

B. *Cryptographic infrastructure*

Unless otherwise noted, all of the algorithms listed are efficient probabilistic algorithms. Certain common inputs, such as *security parameters* which indicate how long keys should be, for example, are omitted for clarity. The overall introduction to these primitives is cursory and informal due to limited space.

Blind Signature Scheme. A *blind signature scheme*, first introduced by Chaum [4], consists of a four-tuple of probabilistic algorithms, GEN, SIGN, REQUEST, and VERIFY. The GEN algorithm generates a public key, pk, and a secret key sk. The $\textsc{Sign}(sk)$ and $\textsc{Request}(pk, m)$ algorithms together form a protocol in which a signer with input sk and signature requester, with input pk and message m interact with one another in order to generate a signature σ. At the end of the protocol, (a) the signer has no knowledge about m or σ, and (b) the requester receives σ, and nothing more. In particular, upon receiving σ at a later point, it is infeasible for the signer to associate σ with the requester; likewise, it is infeasiable for the requester to forge signatures on any other message $m' \neq m$. Finally, the $\textsc{Verify}(\sigma', m_i, pk)$ algorithm runs on input, a pair of strings σ', m' and public key pk and returns either accept or reject. The VERIFY algorithm accepts all signatures generated by the signing algorithm (when run with corresponding public and secret keys). The blind signature scheme in [5] is suitable for our application.

Commitment Scheme. A *commitment scheme* consists of two algorithms, LOCK and OPEN, which are run during two separate phases. Often the following analogy with a metal safe is used: in the commitment phase, the sender locks a message m into a safe and sends the entire safe to a receiver. The message is *hidden* since the receiver cannot crack the safe, and it is *bound* since the sender cannot change the contents of the safe after having given it to the receiver. In the second, *decommitment* phase, the sender gives the safe's key to the receiver, thereby allowing the receiver to read the message. Thus, the $\textsc{Lock}(x) \to (\alpha, \beta)$ algorithm takes input, a message x, and produces a pair of strings (α, β) which consists of a commitment α (safe) , and a secret opening value β (key). The recipient, upon receiving β, can open the value by running $\textsc{Open}(\alpha, \beta) \to x$. See [6] for a detailed description of this primitive.

Zero-knowledge Proof System. A *zero-knowledge proof system* consists of two programs, a PROVER and a VERIFIER, which interact in a way that allows the Prover to convince the Verifier of a true statement without revealing any extra information about the statement. The proof system guarantees three properties: (a) any true statement can be proven so that an honest Verifier will accept the proof, (b) a malicious Prover, no matter how it tries to cheat, has a vanishingly small chance of convincing a Verifier of a false statement, (c) a malicious Verifier, no matter how it tries to cheat, will not "learn" anything other than the truth of the statement. For the purpose of this paper, we shall only attempt to prove statements which have short "witnesses" (i.e. NP languages). Such a statement might be, for example, "The value committed in c does not appear in the list $a_1, a_2, \ldots, a_n$." This statement has a short witness since the opening to the commitment c, i.e. β, allows anyone to run $\textsc{Open}(c, \beta)$ and verify the statement. The salient point, then is that, using zero-knowledge proof systems, a Prover can prove the same statement to a verifier, without revealing β, m or anything else other than the verity of the statement!

Since their discovery by Goldreich, Micali and Widgerson [7], zero-knowledge proofs have become an essential tool for designing modern cryptographic protocols. A good introduction is presented in [6].

C. *Hardware infrastructure*

Our system involves three physical components.

Car Transponder. Each car is equipped with a *transponder* which can be requested to broadcast its identification string. As mentioned above, a transponder's identification string can periodically change. We say that at time t, the transponder broadcasts a string ρ_t. In addition, the transponder is capable of performing standard authentication procedures to verify that the request for identification is being made by an authorized party.

Traffic Logger. A *traffic logger* L monitors the roads. Once a traffic logger has detected that a vehicle has violated a regulation, it authenticates itself to the vehicles transponder, requests the transponder's ID and logs the string which it receives. In order to authenticate itself, the logger has a signing key which is signed by the registering authority. In our system, as with current traffic monitoring systems, drivers might eventually learn the location of the traffic loggers, and modify their driving behavior. (We consider this a positive quality.)

Registration server. The registering authority D maintains a public signing key, pk for a blind signature scheme, a public signing key pk' to sign certificates, and a website.

D. The protocol

There are three simple protocols in our system.

Registration Protocol

Common Input: D's public token signing key pk, for a blind signature scheme. D's public logger signing key pk' for a signature scheme. Let n be the number of minutes in the registration period (e.g., $n = 60 * 24 * 365$ if the registration period is one year).

At the beginning of each registration period, the Transponder and D perform the following steps.

1) **(Generate tokens)** The Transponder randomly generates a sequence of k-bit token strings, $\{m'_{1,a}, m'_{1,b}, \ldots, m'_{n,a}, m'_{n,b}\}$, two for each minute of the time period. These are the "digital license plates" the motorist uses during the registration period. The Transponder computes a commitment, $\text{LOCK}(m'_{i,j}) = (\alpha_{i,j}, \beta_{i,j})$ of each token, and then sends the sequence of commitments $(\alpha_{1,a}, \ldots, \alpha_{n,b})$ to D.
2) **(Proof Challenge)** For each index $i = 1, \ldots, n$ of the registration period, D randomly assigns the variable c_i to a or b and then sends to the Transponder a list of requests, $(c_1, \ldots, c_n)$.
3) **(Proof Response)** The Transponder reveals the commitment m'_{i,c_i} for each challenge c_i requested by sending β_{i,c_i}. This is a proof that the Transponder actually knows each of the tokens that it is registering. Let the sequence of remaining, unopened tokens be re-labelled $\mathcal{M} = \{m_1, m_2, \ldots, m_n\}$, and let the sequence of corresponding commitments be labelled $\mathcal{C} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$.
4) **(Blind Signature)** The Transponder and D engage in a blind signature protocol in order to generate signatures for each pair (i, m_i). Let σ_i denote the signature for (i, m_i) which the Transponder receives.

Traffic Event

Common Input: D's public token signing key pk, D's public logger signing key pk' for a signature scheme.

A traffic logger L records an event at time t involving a vehicle v as follows:

1) **(Authentication)** L authenticates itself by presenting a certificate signed by pk' and requests v to send its current token.
2) **(Reply)** The vehicle v sends (t, σ_t, m_t) to L, encrypted in L's public key pk.
3) **(Verify)** The Logger verifies that σ_t is a valid signature of m_t generated from D's token-signing key pk, and records (t, σ_t, m_t) as well as a short reason for the event (e.g., speeding, red light). If the signature does not verify, then L calls the local police office to indicate that a malfunctioning Transponder has passed through its location.
4) **(Post)** At the end of each day, D posts and timestamps a list, l_d of all of the events that have been logged by all traffic loggers for that day d.

Renewal

Common Input: D's public token-signing key pk, for a blind signature scheme. D's public logger-signing key pk' for a signature scheme.

At the end of the registration period, in order to renew registration, a Transponder and D engage in the following steps:

1) **(Determine Fines)** The Transponder determines the intersection between its private sequence of tokens $\mathcal{M} = \{m_1, \ldots, m_n\}$ and the public list of traffic events, $\mathcal{Y} = \{l_1, \ldots, l_{365}\}$. Suppose this intersection is the list $(m_{e_1}, \ldots, m_{e_k})$. The Transponder reveals these tokens to D by opening the commitments to these events, i.e., by sending the sequence of strings $(\beta_{e_1}, \ldots, \beta_{e_k})$ to D.
2) **(Prove consistency)** The Transponder then proves that there are no more events in the intersection of $\mathcal{M}$ and $\mathcal{Y}$, other than those revealed in the previous step. This is done by proving in zero-knowledge that the openings of the commitments in list $\mathcal{C}$ only intersect with $\mathcal{Y}$ at $(m_{e_1}, \ldots, m_{e_k})$.
3) If D accepts the proof, the Transponder pays the traffic fines and then repeats the Registration process. If any step of the protocol is not completed, then the registration renewal is considered to have failed.

We emphasize that our protocols are to be executed sequentially. In other words, D must perform the Registration and Renewal process one-at-a-time with each of the Transponders, and the Logger and Transponder should also engage in only one interaction at a time. [3]

E. Why Privacy is Preserved

Intuitively, the information recorded by a traffic logger does not identify a vehicle *per se*. During the initial registration phase, the registering authority does not learn the values of the tokens which are sent, nor the its own signatures of those tokens. It only learns a *cryptographic commitment* to the token which hides its value; this value, however, can be used later to guarantee that a motorist pays all of his traffic tickets.

F. How Fines are Paid

Throughout the year, we imagine that a motorist is required to reconcile any outstanding traffic tickets. During

[3] While there are various approaches to proving that a protocol is secure in a concurrent setting when D interacts with several Transponders at the same time, this type of discussion is beyond the scope of this paper.

this phase, a vehicle must disclose the tokens (which he originally committed to during the registration phase) that appear among the list of traffic events which have been posted by D every day on its traffic violations website. In order to be complete, the motorist must prove via a zero-knowledge interaction that all such violations have been acknowledged. A driver who refuses to complete the renewal protocol can be barred from using the public roads through traditional mechanisms (i.e., license suspension).

IV. Malicious Attacks

Our system is secure against a variety of malicious attacks. It is important, however, to be clear about the kinds of security guarantees we provide. We can prove that the mathematical protocols we describe achieve some notion of privacy and guarantee correctness. However, as we are describing a system which must be reified in a physical and bureaucratic implementation, there are of course limits to the kinds of guarantees we can expect simply from analysis of the protocols.

On the other hand, in our situation, we can leverage the power of the "real world" to protect against certain kinds of attacks — for instance, the threat of police enforcement is an important protection against motorists simply removing their transponders altogether. Moreover, by restricting the frequency bands on which transponders and loggers transmit their information, we can prevent various spoofing attacks (see below). To avoid committing to a particular implementation technology at this point, our suggestions of this nature should be regarded as "qualitative" and necessarily requiring interpretation in the specific technical context of implementation.

Outstanding Tickets. We may incentivize traffic violators to pay their tickets early as opposed to later by using the public website to allow vehicle owners to query whether they have any unpaid traffic violations oustanding. Recall that drivers receive "unofficial" notification from their transponder when a logger has recorded its plate number, and so we imagine that in standard usage following such an event the driver would log on to confirm and pay.

Spoofing and "man-in-the-middle" attacks. One might be concerned that a rogue transponder v_1, while passing another transponder v_2, could attempt to read v_2's token, and then passes it off as its own token for that time period. Indeed, this type of attack is a serious one.

For this reason, we require any Logger to authenticate itself to the transponder and send the token over an encrypted channel. Hence, a rogue reader will be unable to convince a Transponder to send its token.

Additionally, we may legally prohibit Transponders from broadcasting outside a certain frequency band, and design the Loggers to send their requests on a different frequency band. (See the next point for how this can be enforced.) Restricting transmission bands also provides protection against attacks in which a malicious transponder sits between a logger and another (innocent) transponder, ferrying messages back and forth.[4]

Broken Transponder. Suppose the owner of a vehicle tampers with or disconnects the car transponder when using the roads. In such a situation, the Traffic logger has no information to record about this vehicle when it commits an infraction. We consider this type of attack no different than if an individual removes or obscures his license plates. It is a serious crime to drive without a license plate; it can be just as serious to drive without a transponder. We rely on the classical (and indispensible) presence of highway patrol in order to solve this problem. Highway patrol officers can be equipped with devices that monitor whether a vehicle has a functioning transponder.

Out-of-State Driver. An out-of-state driver might not be equipped with the Transponder used in this system. Such situations can only be handled by routine highway patrol, or by inter-state cooperation for sharing traffic log records across neighboring states.

Challenges to an Infraction. In the current system, disputing a traffic violation requires considerable proof on the part of the accused that the violation was issued incorrectly. For speeding tickets, for example, a court almost always sides with the officer issuing the ticket. Our system is no different in this respect. One might, however, require the Traffic logger to post calibration information to a website every day, or to record, along with the event, calibration data which can be used to support the alleged violation.

Corrupt registering authority. Since the recorded tokens do not contain any private information, every day, each of the traffic loggers sends its daily event log to a central server. This server timestamps the log using a third party time-stamping service, and posts the list to a publicly available bulletin board (or internet site). This prevents D from generating false events, *after the fact*, in order to gather information about the location of a vehicle, say at the behest of a district attorney involved in a case against a vehicle's owner.

V. Implementation

Various modifications to existing bureaucratic, legal, and technical infrastructure are necessary to implement the scheme we describe. In our presentation of the protocol, we have outlined the new regulatory behaviors of the registry of motor vehicles and the technical behavior of the transponders and loggers. Implicit in this are certain required legal modifications.

Our general view is that all of the required changes are eminently reasonable and achievable based on observation of the experiences of various municipalities in rapidly constructing infrastructure for automated toll collection. Just as in our proposed system, modifications to registration policies, technical infrastructure, and enforcement behaviors were necessary to implement these toll collection systems. In

[4] We should note that traditional solutions to this type of man-in-the-middle attack do not apply because, although the Logger is "authenticated" to the Transponder, the requirement for Transponder anonymity makes the converse impossible.

fact, the obstacles to creating such systems were significantly greater than the ones our system faces precisely because of the lack of prior examples.

One consequence of the adoption of a system such as ours is that it initially reduces the availability of certain kinds of diagnostic information about traffic networks. For instance, precisely because the state cannot track vehicle paths it will lose the ability to estimate congestion by computing individual path durations. However, we believe that the required information is useful primarily as aggregate measures (e.g. the interest is in average time spent in traffic, not in particular vehicles experience of congestion), and so it should still be possible to design ways to compute these measures while preserving driver privacy. We intend to explore such measurement techniques in future work.

VI. Relationship to Other Cryptographic Protocols for Transportation

Many cryptographic notions have been applied to problems in transportation. As early as 1992, David Chaum et.al. [4] proposed and built a prototype anonymous electronic toll-system, called Dynacash, and installed it in Holland and in Japan. The critical element of his system involved smartcard technology which could be "charged up" with digital cash, and automatically debited as a vehicle passed a toll-booth. This system is far more respectful of driver privacy than a system like EZ-Pass, which must record a *static* special-purpose vehicle ID every time the vehicle passes through a toll-booth. Almost all of the "electronic cash" technologies that have subsequently been developed could be employed in this fashion. In such systems, enforcement of toll violations must be handled by an external mechanism, typically a camera.

Chaum has also proposed a trusted-hardware model in which each smartcard contains a tamper-proof "observer" chip which is installed and certified by the registering authority. In this model, credentials can be stored on the smartcard. A credential is simply an authorization which has been assigned to its holder. In our contexts, credentials can model the right to use a highway at a specific time (only granted to vehicles traveling under a set speed limit), or the right to cross an intersection (only granted when the light is green). Verifiers along the entire route can check whether each vehicle is authorized to travel on that route. The problem, however, the process of presenting credentials, when repeated a few times, allows the verifier to ”link” the credentials of a single vehicle to one another.

Thus, neither of these classes of protocols solve the problem we have identified of enabling the driver to preserve anonymity under normal circumstances while permitting compulsory disclosure of identity to the monitoring agent in the event of infractions.

More recently, the authors have learned of a paper by Bangerter, Camenisch, and Lysyanskaya [8] describing a framework for anonymous and unlinkable releasing of ”credential information." Their protocols, which can perhaps be used for certain parts of our system, suggest promising directions for efficient implementations.

VII. Conclusions and future work

We envision a future in which every car has a signal transponder and there are virtually ubiquitous state-managed monitoring devices spread throughout public road space. The massive success and adoption of EZ-pass and related toll systems suggests the plausibility of the former; the current push to install red-light cameras at every intersection presages the latter.

There are grave and obvious threats to the privacy of the individual in such a situation. We believe that there is an essential right to "locational privacy" for individuals. But unlike purely communications-based situations, there are necessarily pragmatic constraints on the kind of privacy available and reasonable.

In this paper, we have achieved two major goals. First, we have attempted to outline a reasonable notion of driver privacy which is compatible with pragmatic constraints and the need for law enforcement but also guarantees a suitable degree of anonymity for the individual. Secondly, we have presented a detailed protocol which enables cities to achieve the ends of the red-light camera systems (and in fact considerably more) in a way which is compatible with our notion of driver privacy. This protocol is relatively easy to implement and secure against a variety of attacks.

In future work, we intend to present a broad family of protocols which enable the state to carry out a wide variety of traffic monitoring tasks while preserving driver privacy. We are hopeful that one can imagine a future in which technology permits increased capabilities for law enforcement and revenue collection without compromising the privacy of the individual.

VIII. Acknowledgements

We would like to thank Ben Lee for helpful discussions and comments. We would also like to express our gratitude to Razvan Surdulescu for a careful reading of the paper which improved the exposition significantly.

References

[1] J. Miller, "With cameras on every corner, your ticket is in the mail," New York Times, January 06 2005.

[2] R. Salladay, "DMV chief backs tax by mile," Los Angeles Times, November 16 2004.

[3] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proc. 19th STOC*. ACM, 1987, pp. 218–229.

[4] D. Chaum, "Blind signatures for untraceable payments," in *CRYPTO '82*, 1982, pp. 199–203.

[5] J. Camenisch, M. Koprowski, and B. Warinschi, "Efficient blind signatures without random oracles," in *In Forth Conference on Security in Communication Networks - SCN '04*, 2004.

[6] O. Goldreich, *Foundations of Cryptography*. Cambridge University Press, 2004, vol. 2, ch. 7 (General Cryptographic Protocols), pp. 599–759.

[7] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proofs," *Journal of the ACM*, vol. 38, no. 3, 1991.

[8] E. Bangerter, J. Camenisch, and A. Lysyanskaya, "A cryptographic framework for the controlled release of certified data," in *Twelfth International Workshop on Security Protocols*, Apr 2004, cambridge, England.