Privacy-Preserving Electronic Toll System with Dynamic Pricing for Low Emission Zones

Roger Jardí-Cedó(✉), Jordi Castellà-Roca, and Alexandre Viejo

Dpt. d'Enginyeria Informàtica i Matemàtiques, UNESCO Chair in Data Privacy, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Spain
{roger.jardi,jordi.castella,alexandre.viejo}@urv.cat

Abstract. Low emission zones (LEZs) aim to reduce pollution and traffic congestion in cities. Current proposals for managing LEZs introduce a significant error percentage in the detection of fraudulent drivers and represent a serious privacy threat for the honest ones. In this article, a new electronic toll system to improve both issues is proposed.

Keywords: Electronic road pricing · Low emission zone · Driver privacy · Security

1 Introduction

Traffic congestion has become a significant problem for almost all major cities and governments have introduced toll systems to solve it. These systems have received a lot of attention [1–8]. The main reason behind the success of this approach is that it enables an authority to restrict the access to drivers willing to pay a certain amount of money. In this way, a Low Emission Zone (*LEZ*) is a restricted area that vehicles can access in exchange for a payment according to the vehicles' carbon emissions. LEZs can adjust the variable prices to manage the flow of vehicles by increasing toll taxes in congested roads and suggesting drivers to take cheaper routes (i.e., they are dynamic).

In general, Electronic Road Pricing (ERP) systems calculate road usage pricing by considering the vehicles' itinerary. Vehicles are equipped with e on-board units (OBU) to record their paths. Each OBU is enabled with GPS and wireless communication capabilities; they periodically collect their geographical position; and send them to a service provider. To avoid fraud, current proposals adopt control mechanisms with the use of checkpoints (*Chp*s), which are equipped with cameras and are randomly located in the LEZs. *Chp*s take pictures of all vehicles and, hence, their number plates are stored together with the corresponding geopositions and time. These three items allow the ERP to build a partial path of all the vehicles moving around the restricted area and verify that a certain driver has not altered the set of positions recorded by her car's OBU and provided to the *SP* during the billing period.

This fraud detection mechanism has a certain failure probability that directly depends on the number of *Chp*s deployed in the restricted area. In addition,

J. Garcia-Alfaro et al. (Eds.): DPM/SETOP/QASA 2014, LNCS 8872, pp. 327–334, 2015.
DOI: 10.1007/978-3-319-17016-9_22

increasing the number of checkpoints directly affects drivers' privacy due to the fact that, the more checkpoints there are, the bigger the set of registered real drivers' locations will be; and therefore, the more accurate the drivers' paths will be.

In this article, a new prepayment *ERP* system for Low Emission Zones (*LEZ*s) is proposed. This system provides: (1) a non-probabilistic fraud control and (2) honest drivers' privacy through revocable anonymity. The system is presented in Sect. 2. The protocol is introduced in Sect. 3. Security is evaluated in Sect. 4, and conclusions are presented in Sect. 5.

2 System Model

Driver D is the person who drives. *Vehicle V* is the means of transport registered by a unique *D*. *V* has an identifier, the vehicle plate. Each *V* has a *Secure element SE* (tamper-proof module) and an *On-board unit OBU* (which has location capabilities and wireless connections).

A *LEZ* is divided into a set of street stretches. A *stretch* is a one-way section of street where *V*s have to pay every time they drive through it. Each stretch is divided into a *payment area* and a traffic *restricted area*. The prices of each stretch are dynamically set according to its traffic density. A *Beacon* is a device, placed at the *payment area*, which constantly warns the *V*s entering the stretch. A *Checkpoint Chp*, placed at the entrance of the *restricted area* of a *stretch*, aims to control the access of vehicles that enter the stretch. *Service Provider SP*, which manages both component types, offers an ERP service for urban areas. *Ticket Provider TP* issues tickets to *V*s.

A *Vehicle Certification Authority VCA* provides keys and certificates to *V*s. A *Payment Service PS* enables *D*s to pay. The electronic payment system is out of scope of this article. Finally, a *Punisher authority PA* knows the identity of the *V* owner and reveals it in case of fraud.

Anti-fraud Requirements. When a *V* enters a *stretch* of a *LEZ* through a *Chp*, it obtains a *ticket* ζ^*. This ζ^* contains information to prove that a specific *V* has the right to enter at a specific time. This *proof* is considered **valid** when it has the following properties: integrity, authenticity, non-repudiation, single-use and temporality. **Fraud** is commited when a *D* drives in a *stretch* without a ζ^*, with an invalid ζ^* or with a valid ζ^* associated with another *V*/*stretch*. A *SP* cannot **falsely accuse** an honest *D* of fraud.

Authenticity Requirements. At the entrance of a *stretch*, *V* and *Chp* must prove their identity to the other part.

Privacy Requirements. The system must (1) assure the privacy (the identity of *D* or *V* cannot be linked to any itinerary); (2) avoid the linkability between itineraries; and (3) provide revocable anonymity to *D*.

3 Protocol Description

3.1 Setup

Before starting the system, the next entities are initialized as follows:

1. *PA* obtains from authorities: (1) An asymmetric key pair (Pk_{PA}, Sk_{PA}), (2) its public key certificate $cert_{PA}$, and (3) a certificate repository of the authorities.
2. *SP* and *VCA* obtain from authorities: (1) An asymmetric key pair (Pk_{SP}, Sk_{SP}) and (Pk_{VCA}, Sk_{VCA}), (2) its public key certificate $cert_{SP}$ and $cert_{VCA}$, and (3) a certificate repository of the authorities.
3. *VCA*:
   i. Defines: (1) A set of vehicles $V = \{v_1, ..., v_{n_V}\}$, where $n_V = |V|$; (2) a collection of sets $K = \{C_1, ..., C_{n_K}\}$ partition of V, where $n_K = |K|$, with $|C_i| = n_C$, $\forall i$
   ii. Generates and associates a certification entity VCA_{C_i} to each element of the subset K $(C_1, ..., C_{n_K})$: (1) An asymmetric key pair ($Pk_{VCA_{C_i}}$, $Sk_{VCA_{C_i}}$), $\forall i \in \{1, ..., n_K\}$ and, (2) a CA certificate $cert_{VCA_{C_i}}$, $\forall i \in \{1, ..., n_K\}$, which has an expiration time c_{exp}
4. *TP* and each *Chp* apply the following steps:
   i. Obtain a certificate repository of the authorities and entities
   ii. Generate an asymmetric key pair (Pk_{TP}, Sk_{TP}) and (Pk_{Chp}, Sk_{Chp})
   iii. Securely obtain a public key certificate $cert_{TP}$ and $cert_{Chp}$ from *SP*. $cert_{Chp}$ contains an extension $cert_{Chp}.loc$ with its location coordinates and a stretch identifier $cert_{Chp}.str$
5. Each *Beacon* is initialized by *SP* with a warning advise *information-of-stretch* ${\beta_{str}}^* = (\beta_{str}, \overline{\beta_{str}})$, where $\overline{\beta_{str}}$ is the signature of β_{str} $(Sign_{SP}(\beta_{str}))$, and where β_{str} contains information of: (1) the street stretch (*str* and GPS cord.); and (2) the *TP* connection (it defines how to access *TP*)
6. *VCA* certificates the *V*s (it is assumed that the *SE* of each *V* has been previously initialized with a certificate repository of the certification authorities, identifying information of the vehicle V_{id} and its technical specifications): (1) Register *V* in an element of the subset *K* (in a C_i) and (2) download the certification entity VCA_{C_i} ($Pk_{VCA_{C_i}}$, $Sk_{VCA_{C_i}}$ and $cert_{VCA_{C_i}}$) associated to C_i by a secure channel in the *SE*.

3.2 Price Generation

Every fixed period of time τ, *SP* establishes the prices of each stretch *str*, depending on its traffic density, by performing the next operations:

1. Set the *prices* per emission category (i.e. European Emission Standards), searching a balance between supply and demand.
2. Compose *information-of-prices* $\alpha_{str} = (str, prices, p_{exp}, acc_d)$, where p_{exp} is the expiration time of the prices and acc_d identifies the *SP* destination account of the electronic payment system assumed.
3. Sign α_{str}: $Sign_{SP}(\alpha_{str}) = \overline{\alpha_{str}}$, send ${\alpha_{str}}^* = (\alpha_{str}, \overline{\alpha_{str}})$ to *TP*.

3.3 Certificate Generation

When V enters a *LEZ*, its *SE* generates new credentials. Its *SE*:

1. Computes an asymmetric key pair (Pk_{V_q}, Sk_{V_q})
2. Generates a public key certificate $cert_{V_q}$ with the next attributes: (1) An extension $cert_{V_q}.idS$ containing the probabilistic encryption of the vehicle identifier V_{id} with the public key of *PA*: $Enc_{Pk_{VCA}}(V_{id})$; and (2) an extension $cert_{V_q}.em$ containing its pollutant emission category.

3.4 Purchase

When a V enters a payment area, the purchase protocol is applied:

1. *Beacons* send *information-of-stretch* ${\beta_{str}}^*$
2. The SE of the V, with the help of the *OBU*, has to:
   i. Verify the signature $\overline{\beta_{str}}$: $Verif_{SP}(\beta_{str}, \overline{\beta_{str}})$ and its GPS location
   ii. Establish with *TP* a secure and secret communication channel
   iii. Extract *str* from β_{str} and send *TP* a request for the prices of *str*
3. *TP* sends ${\alpha_{str}}^*$ to V
4. The SE of the V, with the help of the *OBU*, has to:
   i. Verify the signature $\overline{\alpha_{str}}$: $Verif_{SP}(str, prices, p_{exp}\ acc_d, \overline{\alpha_{str}})$, and the freshness of ${\alpha_{str}}^*$:$|p_{exp}\text{-}current\ time| < \tau'$ (a fixed time)
   ii. Obtain the *amount* to pay, according to its pollutant emissions.
   iii. Compose a *payment order* $\gamma = (acc_s, amount, acc_d)$, where acc_s is the source account of the user, and acc_d is the destination account
   iv. Sign γ: $Sign_{V_q}(\gamma) = \overline{\gamma}$ and send $\gamma^* = (\gamma, \overline{\gamma})$ and its certificate $cert_{V_q}$ to *PS*. γ includes additional information, which indicates the source account and authenticates its owner in front of *PS*.
5. *PS* has to: (1) Perform the actions belonging to the used payment system; and (2) compose $\delta = (trans_{id}, hash(\overline{\gamma}))$, sign δ: $Sign_{PS}(\delta) = \overline{\delta}$, and send $trans_{id}$ and $\overline{\delta}$ to V;
6. The SE of the V, with the help of the *OBU*, has to:
   i. Compute $hash(\overline{\gamma})$, recompose δ and verify $\overline{\delta}$: $Verif_{PS}(\delta, \overline{\delta})$
   ii. Compute $hash(\overline{\delta})$ and compose a *ticket request* $\epsilon = (str, ts, hash(\overline{\delta}))$, where ts is the current time
   iii. Sign ϵ: $Sign_{V_q}(\epsilon) = \overline{\epsilon}$, and send $\epsilon^* = (\epsilon, \overline{\epsilon})$ and its $cert_{V_q}$ to *TP*
7. *TP* has to:
   i. Verify the certificate $cert_{V_q}$ and the signature $\overline{\epsilon}$: $Verif_{V_q}(\epsilon, \overline{\epsilon})$
   ii. Verify the freshness of ts:$|ts - current\ time| < \tau'$, where τ' is a fixed time
   iii. Compute $hash(\overline{\epsilon})$ and the fingerprint $fing_{V_q}$ of $cert_{V_q}$
   iv. Compose a *ticket* $\zeta = (str, ts, fing_{V_q}, hash(\overline{\epsilon}))$, sign it: $Sign_{Chp}(\zeta) = \overline{\zeta}$, send ζ^*=$(\zeta, \overline{\zeta})$ to V, and send ϵ^* and ζ^* to *SP*
8. The SE of the V, with the help of the *OBU*, computes $hash(\overline{\epsilon})$, and verifies the signature $\overline{\zeta}$: $Verif_{TP}(str, ts, fing_{V_q}, hash(\overline{\epsilon}), \overline{\zeta})$

3.5 Entrance

When a *Chp* detects the V, the protocol is applied:

1. *Chp* generates a nonce N_A, and sends N_A and the $cert_{Chp}$ to V
2. *SE* of the V, with the help of the *OBU*, has to:
   i. Verify the certificate $cert_{Chp}$, $cert_{Chp}.loc$ and $cert_{Chp}.str = str$
   ii. Generate a nonce N_B and compute the $fing_{Chp}$ of $cert_{Chp}$
   iii. Compose an *entrance request* $\eta = (N_A, fing_{Chp}, N_B, \zeta^*)$ and sign it: $Sign_{V_q}(\eta) = \overline{\eta}$
   iv. Generate a digital envelope of $\eta^* = (\eta, \overline{\eta})$ with the *Chp*'s public key
   v. Send the digital envelope and its $cert_{V_q}$ to *Chp*
3. *Chp* has to:
   i. Open the digital envelope with its secret key obtaining η^*
   ii. Verify the certificate $cert_{V_q}$ and the signature $\overline{\eta}$: $Verif_{V_q}(N_A, fing_{Chp}, N_B, \zeta^*, \overline{\eta})$ and generate timestamp ts'
   iii. Verify the signature $\overline{\zeta}$: $Verif_{TP}(\zeta, \overline{\zeta})$
   iv. Verify $cert_{Chp}.str = str$ (extracted from ϵ, included in ζ) and verify the freshness of ζ^*:$|ts - ts'| < \tau''$, where τ'' is a time which is fixed according to the traffic volume of the stretch.
   v. Compute the $fing'_{V_q}$ of $cert_{V_q}$ and verify $fing'_{V_q} = fing_{V_q}$
   vi. If one of the verifications fails or ζ^* has not sent, *Chp* performs the following operations: (1) Generate an incidence number of entrance in_i; (2) Take a photo ph of V and extract the plate number plt; (3) Compose a *proof-of-entrance incidence* $\theta_i = (in_o, plt, ph, ts', \eta^*, cert_{V_q})$; (4) Sign θ_i: $Sign_{Chp}(\theta_i) = \overline{\theta_i}$ and send ${\theta_i}^* = (\theta_i, \overline{\theta_i})$ to *SP*.
   vii. If the verifications performed in 3ii–3v are correct, the *Chp* has to: (1) Compose *proof-of-entrance* $\iota = (ts', \eta^*, cert_{V_q})$; (2) Sign ι: $Sign_{Chp}(\iota) = \overline{\iota}$, and send ts' and $\overline{\iota}$ to the V
4. If the verifications performed in 3ii–3v are correct, *SE* of the V, with the help of the *OBU*, verifies the certificate $cert_{Chp}$ and the signature $\overline{\iota}$: $Verif_{Chp}(ts', N_A, fing_{Chp}, N_B, \zeta^*, \overline{\eta}, cert_{V_q}, \overline{\iota})$

3.6 Payment Verification

TP sends *SP* *ticket request*s ϵ^* and *ticket*s ζ^* periodically. Each *Chp* sends *proof-of-entrance*s ι^* and *proof-of-entrance incidence*s ${\theta_i}^*$. *SP* then forwards the incidences ${\theta_i}^*$ to *PA*. Moreover, *SP* verifies a posteriori the payment performed by each V according to the following operations:

1. Define a set of *proof-of-entrance* $I = \{{\iota_1}^*, ..., {\iota_{n_\iota}}^*\}$ and a set of *ticket request* $E = \{{\epsilon_1}^*, {\epsilon_2}^*, ..., {\epsilon_{n_\epsilon}}^*\}$, where n_ι is the number of *proof-of-entrance* and n_ϵ is the number of *ticket request* sent to *TP*
2. Select a subset E' from E (each ${\epsilon_i}^*$ has been used by some D)
3. Verify that there is a unique ϵ in the set E' with the same $hash(\overline{\delta})$
4. For each *proof-of-entrance* ι^*:

i. Extract str, ts and $hash(\overline{\epsilon})$ from *ticket* ζ
ii. Recover the prevailing *information-of-prices* ${\alpha_{str}}^*$ of the stretch str at time ts, and extract *prices* from α_{str} and $cert_{V_q}.em$ from ι
iii. Obtain from *prices* the $amount'$ to pay according to $cert_{V_q}.em$
iv. Verify whether the transfer, referenced by $hash(\overline{\epsilon})$, was successful, and recover the *amount* of money paid
v. Verify that $amount = amount'$
vi. Verify that ζ^* is unique in the set I

5. If one of the verifications fails, SP then needs to:
   i. Generate an incidence number of verification in_v
   ii. Compose *proof-of-verification incidence* θ_v, including ι^* concerned: θ_v= (in_v, ι^*). In the case of a reused ζ^*, add ${\iota^*}'$ to $\theta_v = (in_v, \iota^*, {\iota^*}')$. Moreover, when a $hash(\overline{\delta})$ is reused, θ_v is supplemented with both *ticket request*s of the *proof-of-entrance*s proving it: $\theta_v = (in_v, \iota^*, {\iota^*}', \epsilon^*, {\epsilon^*}')$
   iii. Sign θ_v: $Sign_{SP}(\theta_v)$=$\overline{\theta_v}$ and send ${\theta_v}^*$=$(\theta_v, \overline{\theta_v})$ to PA.

3.7 Sanction

For each received θ, PA performs the following operations:

1. In the case of θ_i, PA verifies the signatures and extracts the number plate plt from the photograph ph, included in θ_i
2. In the case of θ_v PA has to:
   i. Verify all the signatures included in θ_v and the signatory of ι
   ii. Verify the right payment by repeating steps 4i–4v of phase Sect. 3.6
   iii. In case of a reused ζ^*, verify that it is the same in ι and ι'
   iv. In case of a reused $hash(\overline{\delta})$, verify the $hash(\overline{\epsilon})$ of ζ^* (included in ι references ϵ^*) and the $hash(\overline{\epsilon})'$ of ${\zeta^*}'$ (included in ι' references ${\epsilon^*}'$), and verify that both ϵ^* and ${\epsilon^*}'$ have the same $hash(\overline{\delta})$
   v. If the incidence is confirmed, recover the identifier V_{id} of V_q by opening the extension $certV_q.idS$ of the certificate $certV_q$, included in ι: Dec_{PA} $(certV_q.idS) = V_{id}$. In the case of a reused $hash(\overline{\delta})$, the identifier V'_{id} of the second vehicle ${V_q}'$ is also recovered in the same way from ι'
3. Notify the owner of V, using plt or V_{id}, about the sanctioning procedure and request her contrary evidences to refute her accusation.
4. Verify the contrary evidences presented by the owner of the V. In the case of a reused $hash(\overline{\delta})$, the evidences should include the hashes, which allow to evaluate the pre-images of the hash chain, and the first value of the hash chain $\overline{\gamma}$, which proves the signature authorship.
5. Fine the owner of the V if the presented evidences are not valid.

4 Security and Requirement Analysis

The system preserves authenticity, non-repudiation, integrity, single-use and temporality for the *tickets* to be considered **valid** since:

- The creation of fraudulent *tickets* is computationally unfeasible nowadays without the knowledge of Sk_{TP} in the signature.
- TP cannot deny their emission, the issuer's identity is linked to the proofs, and for the properties of the electronic signature scheme, it cannot deny its authorship.
- The modification of the content of the *ticket*s by Vs is computationally unfeasible nowadays since the hash summary function used in the signature scheme is collision-resistant, and without the knowledge of the TP secret key.
- A *ticket* cannot be reused to enter a stretch without being detected because it is considered unique. A ticket can only be created by TP, only one ticket at the same time.
- *Tickets* cannot be used to enter a stretch after their expiration because each ticket cannot be modified and contains the time ts of its emission, which is verified by the TP.

The toll system is resistant to **fraud** since users who enter a stretch without a ticket (in step 3vi of Sect. 3.5), with no valid ticket, or with a valid ticket associated with another user or stretch, are detected (in step 3v of Sect. 3.5). Otherwise, the system protects users against **false accusations** since the protocol execution generates records signed by the involved entities, which prove the user has entered the stretch without committing fraud. She will then be able to retrieve some of these records and provide them to PA in order to prove its own honesty.

The system preserves **anonymity** and **traceability** between user itineraries (an itinerary starts each time V enters a LEZ) to honest drivers since: (1) The information that can identify a user ($cert_{V_q}.idS$, which contains V_{id}) does not reveal the user's identity because this information is encrypted using the public key of PA. The certificate $cert_{V_q}$ can be neither identify them thanks to the fact that the CA, is shared with several users and because each user is registered in an element of the subset K together with other users; and (2) the SE generates a new $cert_{V_q}$ for the vehicle in each new LEZ entrance. Nobody can neither relate the identity of the V of this itinerary with any other nor know whether two certificates belong to the same user, as there are K different CAs.

The system provides **anonymity revocation** for dishonest drivers according to how fraud is detected:

- In case of a reused ticket, the amount paid does not correspond to the tax determined for τ, or emissions of V, or a reused transfer reference $hash(\bar{\delta})$ are detected, SP sends θ_v to PA. PA verifies the incidence and identifies the user by opening the field $certV_q.idS$ of the certificate with its private key. Obtaining V_{id} allows the identification and punishment of the dishonest user.
- In other cases of fraud, the user is identified when she is photographed by the Chp at the entrance of a stretch. The user then loses her anonymity as the vehicle number plate is captured.

5 Conclusions

This paper has presented an *ERP* system for urban areas with an enhanced dynamic pricing, which provides a robust fraud control system and a high level of privacy. The entrance process of a *LEZ* is controlled so that the legitimate tax is dynamically computed depending on the traffic volume while the anonymity of the user is preserved. However, if a user commits fraud, she will then be identified by the picture of the number plate taken by the checkpoint in conjunction with the anonymity revocation system of the protocol.

Acknowledgments and disclaimer. This work was partially supported by the Government of Catalonia under grant 2009 SGR 1135 and by the Spanish Government under CO-PRIVACY TIN2011-27076-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004, BallotNext IPT-2012-0603-430000 and MOBILE KEY RTC-2014-2552-7 projects and the FPI grant BES-2012-054780.

References

1. Balasch, J., Rial, A., Troncoso, C., Preneel, B., Verbauwhede, I., Geuens, C.: Pretp: Privacy-preserving electronic toll pricing. In: USENIX Security Symposium, pp. 63–78 (2010)
2. Chen, X., Lenzini, G., Mauw, S., Pang, J.: A group signature based electronic toll pricing system. In: ARES, pp. 85–93. IEEE Computer Society (2012)
3. Day, J., Huang, Y., Knapp, E., Goldberg, I.: Spectre: spot-checked private ecash tolling at roadside. In: WPES, pp. 61–68. ACM (2011)
4. Garcia, F.D., Verheul, E.R., Jacobs, B.: Cell-based privacy-friendly roadpricing. Comput. Math. Appl. **65**(5), 774–785 (2013)
5. Hoepman, J.-H., Huitema, G.: Privacy enhanced fraud resistant road pricing. In: Berleur, J., Hercheui, M.D., Hilty, L.M. (eds.) HCC9 2010. IFIP AICT, vol. 328, pp. 202–213. Springer, Heidelberg (2010)
6. de Jonge, W., Jacobs, B.: Privacy-friendly electronic traffic pricing via commits. In: Degano, P., Guttman, J., Martinelli, F. (eds.) FAST 2008. LNCS, vol. 5491, pp. 143–161. Springer, Heidelberg (2009)
7. Meiklejohn, S., Mowery, K., Checkoway, S., Shacham, H.: The phantom tollbooth: Privacy-preserving electronic toll collection in the presence of driver collusion. In: USENIX Security Symposium, pp. 32–32 (2011)
8. Popa, R.A., Balakrishnan, H., Blumberg, A.J.: Vpriv: Protecting privacy in location-based vehicular services. In: USENIX Security Symposium, pp. 335–350. USENIX Association (2009)