Berner Fachhochschule - Technik und Informatik - RISIS

# On Road Pricing

## E-Voting Seminar

Eric Dubuis

June 25th, 2012

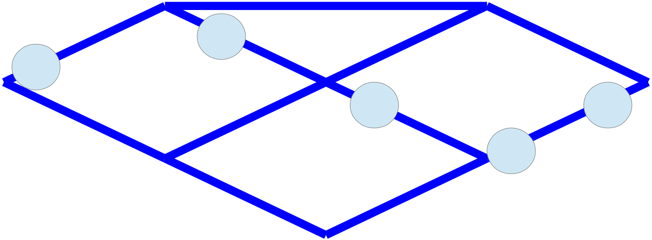
# Problem Statement

# Outline

# Outline

# Traffic Network

Basic concepts:



- "point tuple": $\langle tag, time, location \rangle$
- path of car $p_c$: $\{\langle tag, time, location \rangle\}$
- cost function: $f(p_c)$

If location privacy were no concern then the tags would uniquely identify cars.

# Kinds of Functions

We want functions $f(p_c)$ such as:

- **Usage-based tolls**
  Assessing path-dependant toll
- **Speed surveillance**
  Detecting speed limit violations
- **"Pay-as-you-go" insurance premiums**
  Individualizing insurance premiums depending on, for example, acceleration

# Outline

# Participants

The system model is composed of drivers, cars, and a (logical) server

- **Drivers**
  Driver drive cars, but run also *client software*
- **Cars**
  Every car has a transponder obtaining point tuples (GPS, roadside devices)
- **Logical server**
  Collects point tuples; participates in a cryptographic protocol

# Threat Model

It is obvious that participants may want to misbehave:

1. The driver runs a modified client software to change the result of $f(p_c)$
2. The driver manipulates the transponder
   - → by turning it off
   - → by letting it upload synthesized data
   - → by masquerading another car
3. The server guesses the path from the uploads point tuples
4. The server attempts to change the result of $f(p_c)$
5. Some intermediate device in the data network synthesizes false point tuples or modifies point tuples in transit

# Design Goals

The following three design goals are envisaged:

- **Correctness**
  For every car $c$ having path $p_c$, the server computes the correct value $f(p_c)$
- **Efficiency**
  The protocol must be sufficiently efficient allowing inexpensive in-car devices
- **Location privacy**
  See next slide...

# Location Privacy

Let

- $\mathcal{S}$ be the server's database of point tuples $\langle tag, time, location \rangle$;
- $\mathcal{S}'$ be the server's database of point tuples $\langle time, location \rangle$ such that for every $\langle tag, time, location \rangle \in \mathcal{S}$ there exists a tuple $\langle time, location \rangle \in \mathcal{S}'$;
- $c$ be an arbitrary car;
- $\mathcal{V}$ denote all information to the server;
- $\mathcal{V}'$ denote all information contained in $\mathcal{S}'$, the result of $f(p_c)$ of car $c$, and any other side information.

Then the computation of $f(p_c)$ preserves the *location privacy* of $c$ if the server's information about the tuples of $c$ is insignificantly larger in $\mathcal{V}$ than in $\mathcal{V}'$.

# Outline

# Different Phases (1/3)

The participants' interactions occur in three *phases*

1. **Registration**
   - → Driver registers identifying information *id* to the server
   - → Driver generates *random tags*
   - → Driver transfers random tags to transponder (the car)
   - → Driver transfers *commitments* of tags to the server
   - → Server binds commitments to driver/car
2. **Driving**
   See next slide...
3. **Reconciliation**
   See next slides...

# Different Phases (2/3)

The participants' interactions occur in three *phases*

1. **Registration**

   See previous slide. . .

2. **Driving**

   → Transponder collects point tuples $\langle \textit{time}, \textit{location} \rangle$
   → Transponder sends point tuples $\langle \textit{tag}, \textit{time}, \textit{location} \rangle$ to the server (continuously or in batch mode); random tags are never reused
   → Random *spot checks* send sporadic *identifying* point tuples $\langle \textit{id}, \textit{time}, \textit{location} \rangle$ to the server

3. **Reconciliation**

   See next slide. . .

# Different Phases (3/3)

The participants' interactions occur in three *phases*

1. **Registration**
   See previous slides. . .
2. **Driving**
   See previous slide. . .
3. **Reconciliation**
   At the end of the tax interval, the sever computes $f$.

# Outline

# Notation

The following notation will be used:

- Let $v_i \in_R V$ be the a *random (vehicle) tag*
- Let $f_k$, $k$ chosen at random, be a *random function*
- Let $c(.)$ be a *commitment**
- Let $d(.)$ be a *decommitment key* of commitment $c(.)$
- Let $s_j$ be a *random (vehicle) tag* received at the server
- Let $t_j$ be a *tolling cost* associated with $s_j$

*) Homomorphic commitment having the property
$c(v) \cdot c(v') = c(v + v')$.

# Three Phases of the Protocol

| Client | | Server |
|---|---|---|
| Chooses $v_i$, $k$ | | |
| Encrypts $f_k(v_i)$ | | |
| Stores $d(k)$, $d(f_k(v_i))$ | | |
| Sends... | $-c(k),\ c(f_k(v_i))\rightarrow$ | Binds values to C. |
| Produces p.t. using $v_i$ | | |
| Sends *anonymously*... | $-$p.t. with $v_i\rightarrow$ | Stores $v_i$ as $s_j$ |
| | | $\forall s_j$ computes $t_j$ |
| | $\leftarrow (s_j, t_j)-$ | Sends... |
| Computes $T = \sum_{v_i=s_j} t_j$ | | |
| Sends... | $-T\rightarrow$ | |
| | Round protocol begins | |

p.t. = path tuples

# Round Protocol ($b = 0$)

| Client | | Server |
|---|---|---|
| S. $(s_j, t_j) \rightarrow (s_j, t_j)^*$ | | |
| Encrypts $f_k(s_j)$ | | |
| Computes $c(t_j)$ | | |
| Stores $d(f_k(s_j))$, $d(t_j)$ | | |
| Sends... | $-c(f_k(s_j)),\ c(t_j)\rightarrow$ | |
| | | Choose $b$ a.r. |
| | $\leftarrow b -$ | Challenge $b$ |
| If $b = 0$: | | |
| Sends... | $-k,\ (s_j, t_j)^*,\ d(k),\ d(t_j) \rightarrow$ | |
| | | If $b = 0$: |
| | | Verifies $(s_j, t_j)^*$, |
| | | $\exists i, j$ : |
| | | $f_k(s_j) = f_k(v_i)$ |

S. = shuffles, a.r. = at random, r.o. = random order

# Round Protocol ($b = 1$)

| Client | | Server |
|---|---|---|
| ... | | |
| | $\leftarrow b -$ | ... |
| If $b = 1$:<br>Computes $D$<br>Sends... | $-D,\ d(f_k(v_i)), \rightarrow$ | |
| | | If $b = 1$:<br>Computes<br>$\prod_{j,k,f_k(v_i)=f_k(s_j)} c(t_j)$ |

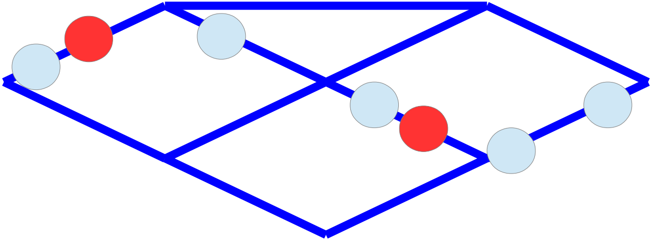
Let $I = \{t_j \ : s_j \in \{v_i\} \cap \{s_j\}\}$. By the homomorphic of the commitment scheme: $\prod_{t_j \in I} c(t_j)$ is the cyphertext of the total tolling cost $T$ whose decommitment key is $D = \sum_{t_j \in I} d(t_j)$.

# Outline

# Enforcement

Client may cheat by turning off the transponder or by providing "invented" path tuples.

Random spot checks



Client must prove that, for each random spot check, she provided a tuple "close enough" to each spot check.

# Outline

# (Some) Security Analysis

Client and network attacks:

- Point tuples should be encrypted with server's public key
- Point tuples should be anonymously signed (e.g., via group signature scheme)
- Spot checks reduce client misbehavior likelihood
- If two clients commit the same tags then they pay the sum of tolling amounts

Server misbehavior:

- Point tuples should be sent anonymously
- Collect p.t. of areas with high traffic density only
- Little changes to the protocol make server more resilient to other attacks

# Outline

# Summary of Talk

- Talk scratched the surface of the problem domain only
- Presented protocol can be used for tolling, speeding tickets, insurance premium computation
- Spot checking can be abandoned if tamper-resistent transponders are used
- Performance is said to be good enough. Could be improved if location privacy is compromised a little by forming *tag clusters*
- Location privacy-preserving solutions can be built using building blocks similar to the ones used for e-voting
- I'm tempted to say that the same is true for e-ticketing systems

# Bibliography (1/3)

This talk is based on the following paper:


- (*) R. A. Popa, H. Balakrishnan, A. Blumberg: VPriv: Protecting Privacy in Location-Based Vehicular Services. 18th USENIX Security Symposium, Montreal, Canada, 2009.


(*) Available in ./bibliography folder

# Bibliography (2/3)

Other papers and/or articles related to this subject:


- (*) B. Jacobs: Architecture is Politics: Security and Privacy Issues in Transport and Beyond. Based on key note talk, Privacy, and Data Protection (CPDP) conference, Brussels, 2009.
- (*) W. de Jonge, B. Jacobs: Privacy-friendly Electronic Traffic Pricing via Commits. In: LNCS, workshop of Formal Aspects in Security and Trust, Malaga, Spain, 2008.
- (*) J. Balasch, C. Troncoso, B. Preneel, I. Verbauwhede, C. Geuens: PrETP: Privacy-Preserving Electronic Toll Pricing (extended version). Normal version presented on: 19th USENIX Security Symposium, Washington, USA, 2010.
- (*) X. Chen, G. Lenzini, S. Mauw, J. Pang: A Group Signature Based Electronic Toll Pricing System. In: Clinical Orthopaedics and Related Research, Vol. ABS/1108.0574, 2011.
- (*) J. H. Hoepman, B. Jacobs, P. Vullers: Privacy and Security Issues in e-Ticketing. Unpublished.


(*) Available in ./bibliography folder

# Bibliography (3/3)

Other papers and/or articles related to this subject:


- (*) M. Langheinrich: Privacy by Design — Principles of Privacy-Aware Ubiquitous Systems. In: Ubicomp 2001: Ubiquitous Computing, LNCS, Vol. 2201, 2001.
- (*) A.-R. Sadeghi, I. Visconti, C. Wachsmann: User Privacy in Transport Systems Based on RFID E-Tickets. Unpublished.
- J. Balasch, I. Verbauwhede, B. Preneel: An embedded platform for privacy-friendly road charging applications. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010.
- J.-H. Hoepman, G. Huitema, J. Berleur, M. Hercheui, L. Hilty: Privacy Enhanced Fraud Resistant Road Pricing. In: What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience; IFIP Advances in Information and Communication Technology Springer, 2010, ISBN 978-3-642-15478-2.


(*) Available in ./bibliography folder