Privacy in Mobile Networks

Erman Ayday

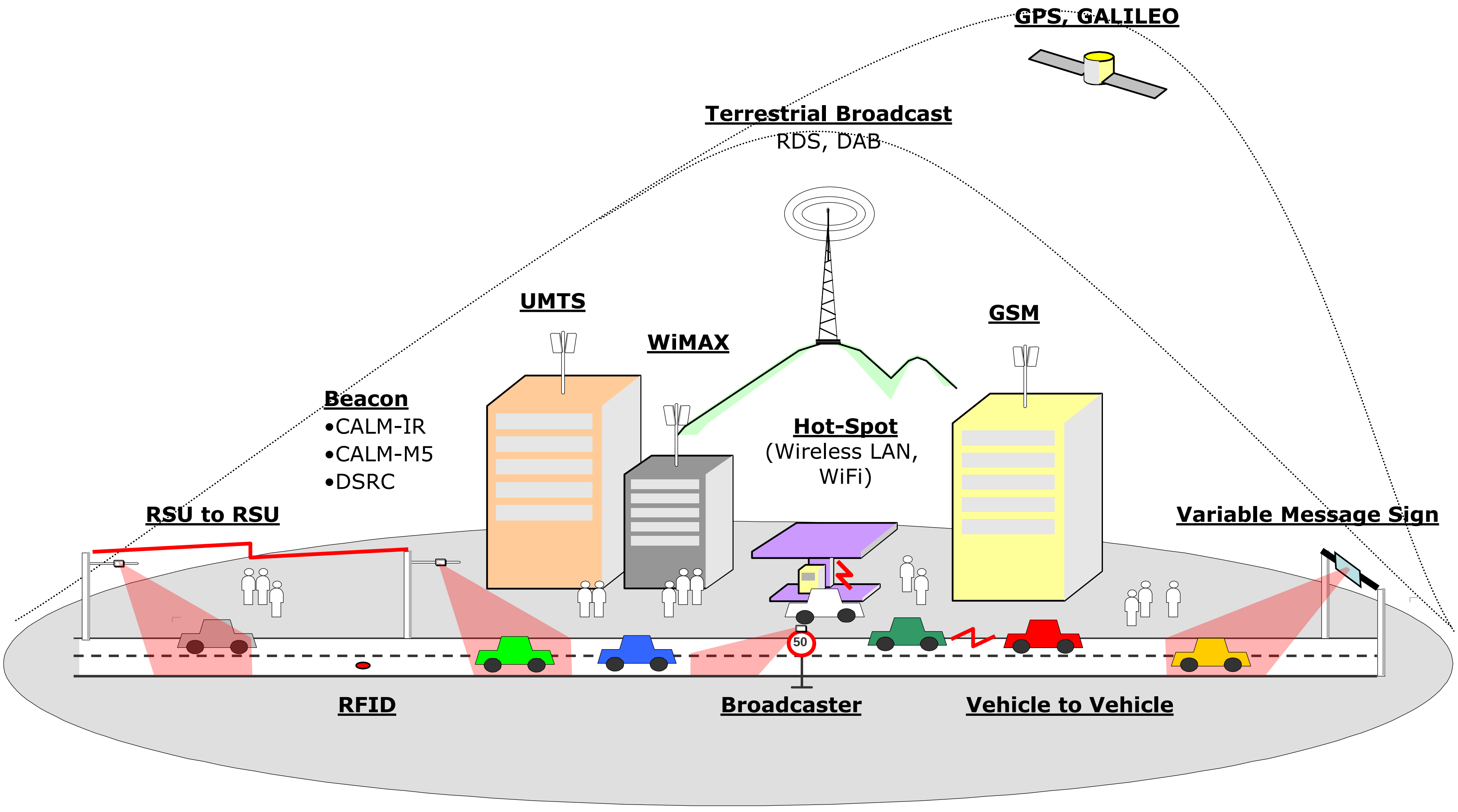
Some of the slides are adapted from the book by Buttyan and Hubaux: “Security and Cooperation in Wireless Networks, Chapter 8: Privacy Protection”

Location privacy

A location trace is not only a set of positions on a map

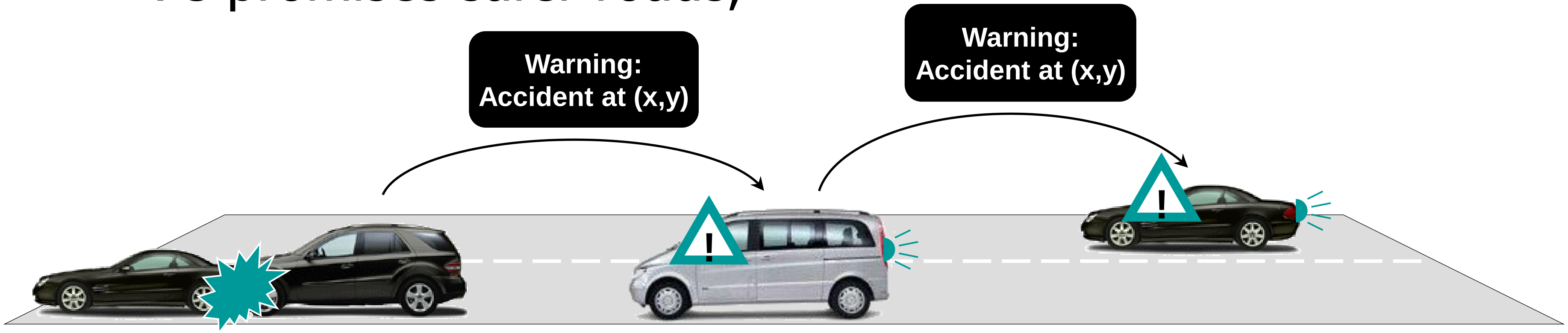
The contextual information attached to a trace tells much about our habits, interests, activities, and relationships

A first example: Vehicular networks

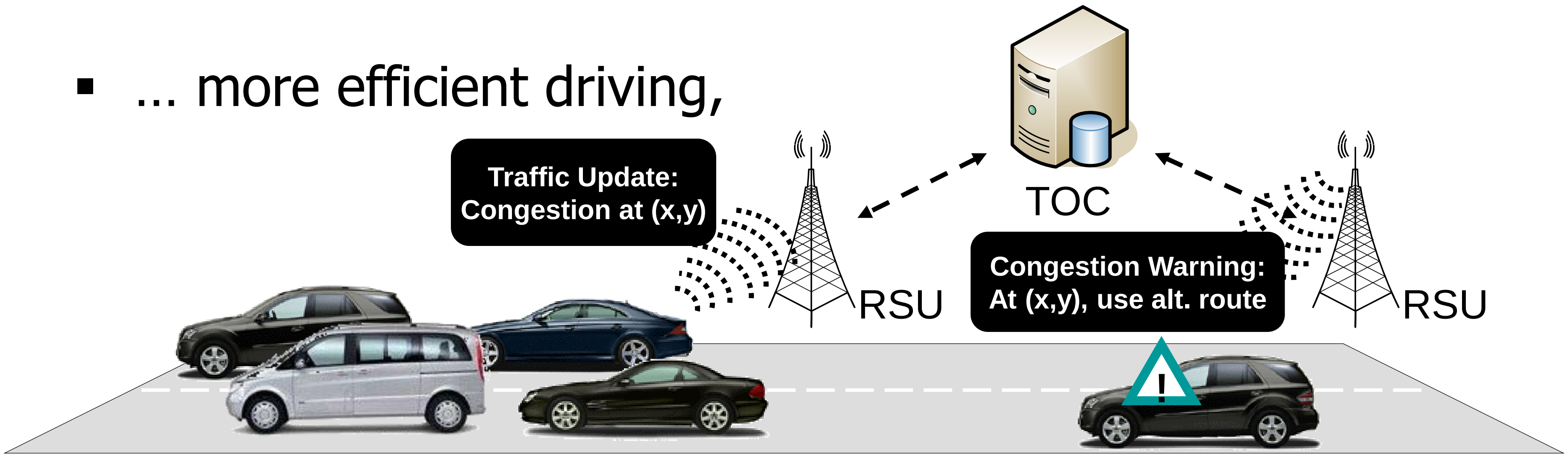


Vehicle Communication (VC)

- VC promises safer roads,

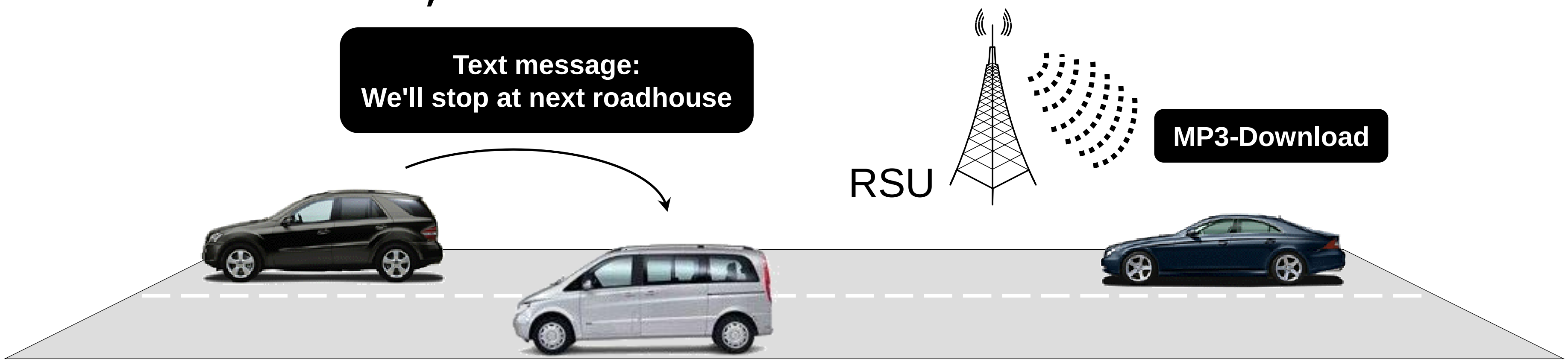


- … more efficient driving,

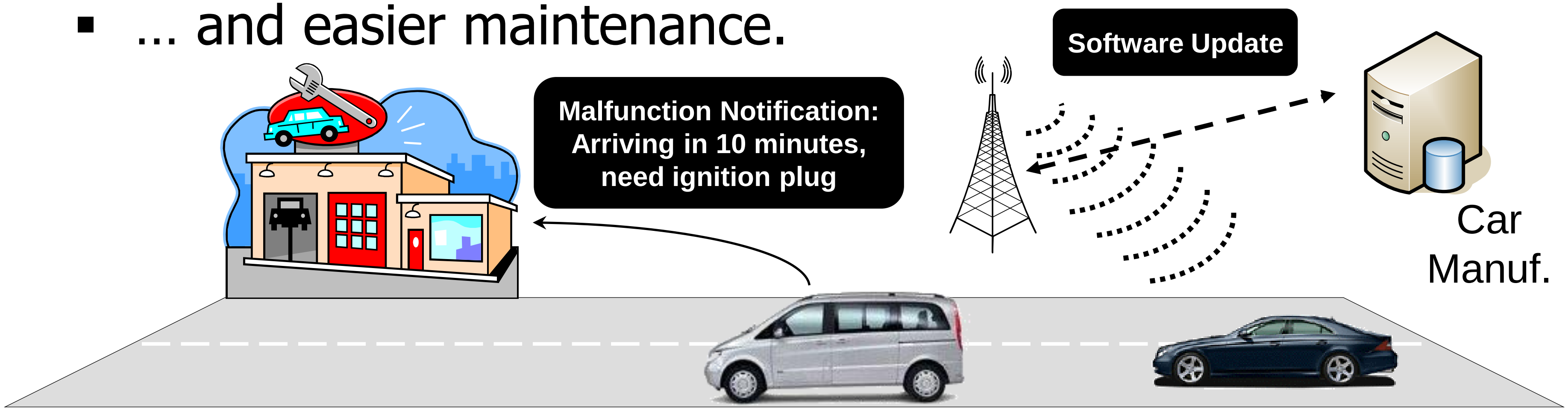


Vehicle Communication (VC)

- … more fun,

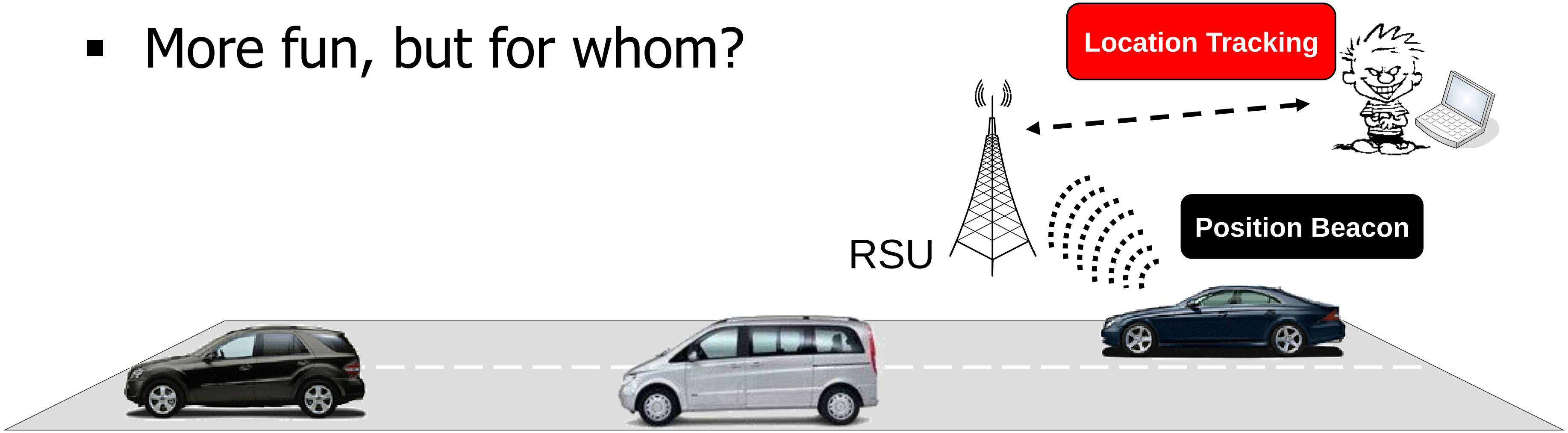


- … and easier maintenance.

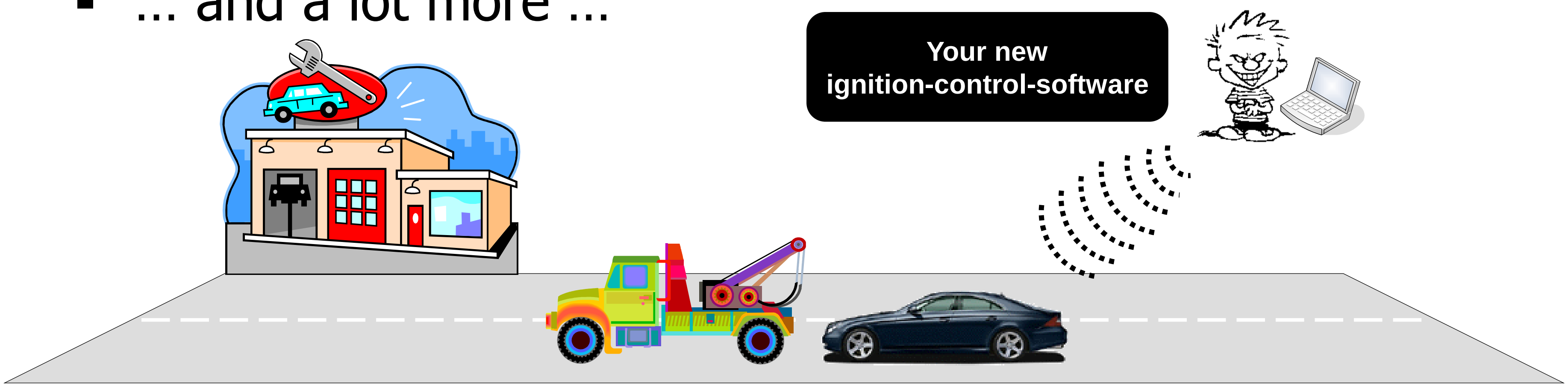


Security and Privacy

- More fun, but for whom?



- … and a lot more …



The location privacy problem and a solution

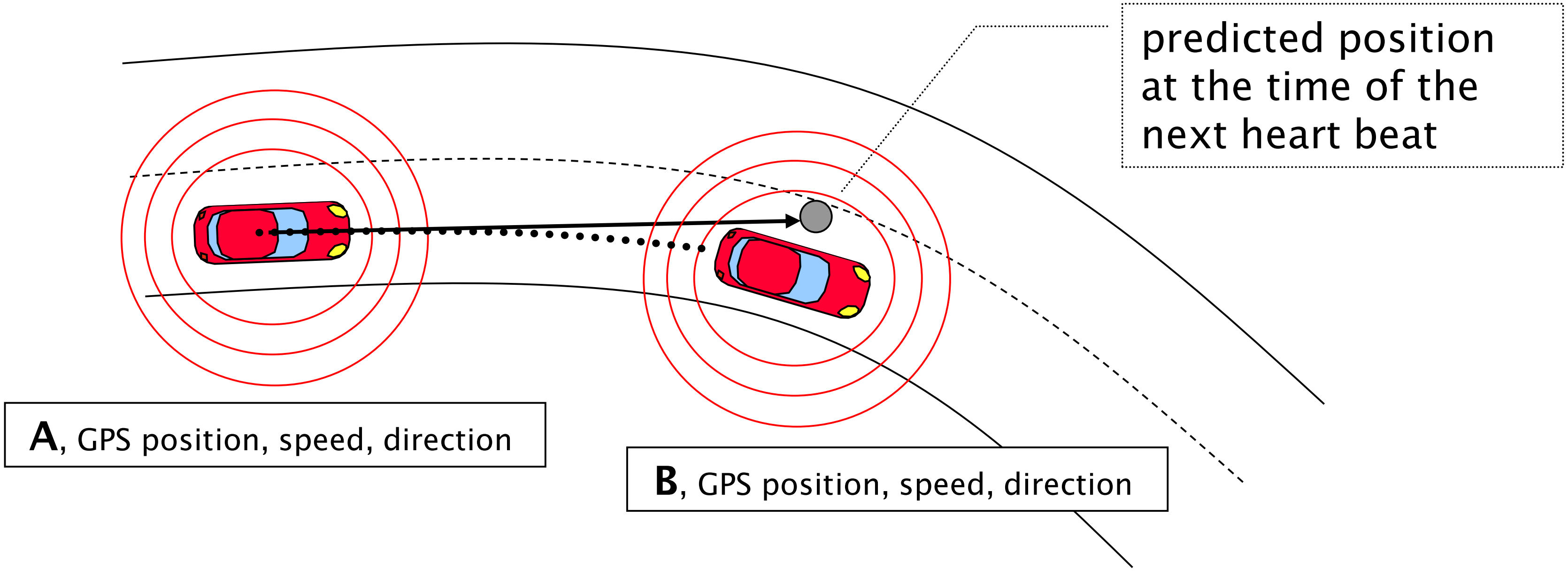
- vehicles continuously broadcast *heart beat* messages, containing their ID, position, speed, etc.

- tracking the physical location of vehicles is easy just by eavesdropping on the wireless channel

- one possible solution is to change the vehicle identifier, or in other words, to use *pseudonyms*

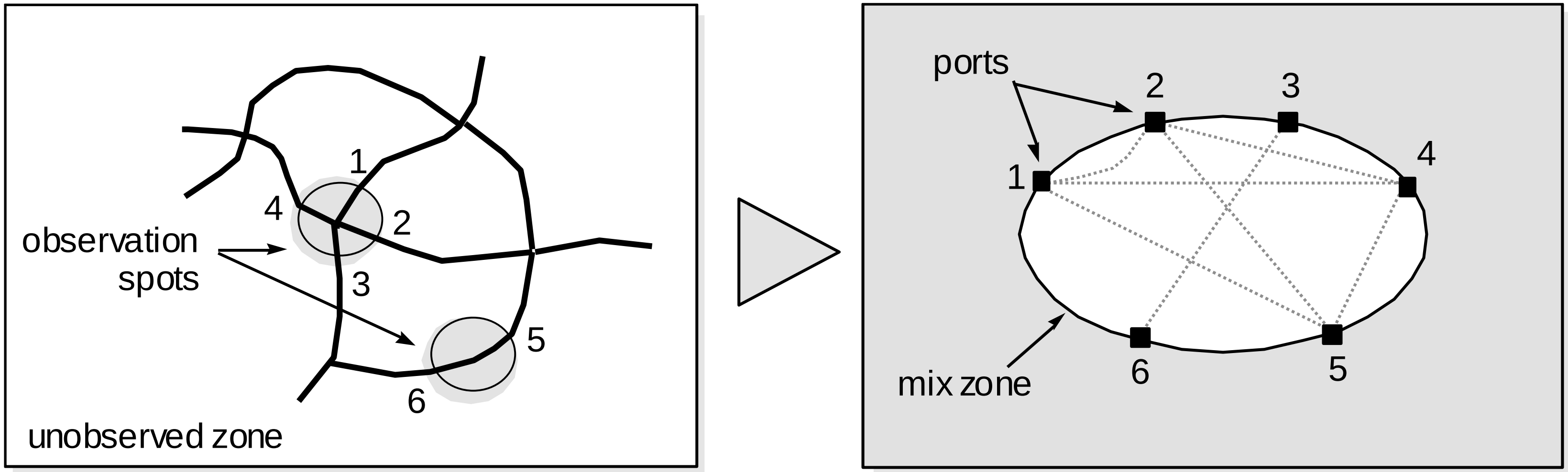
Adversary model

- changing pseudonyms is ineffective against a global eavesdropper



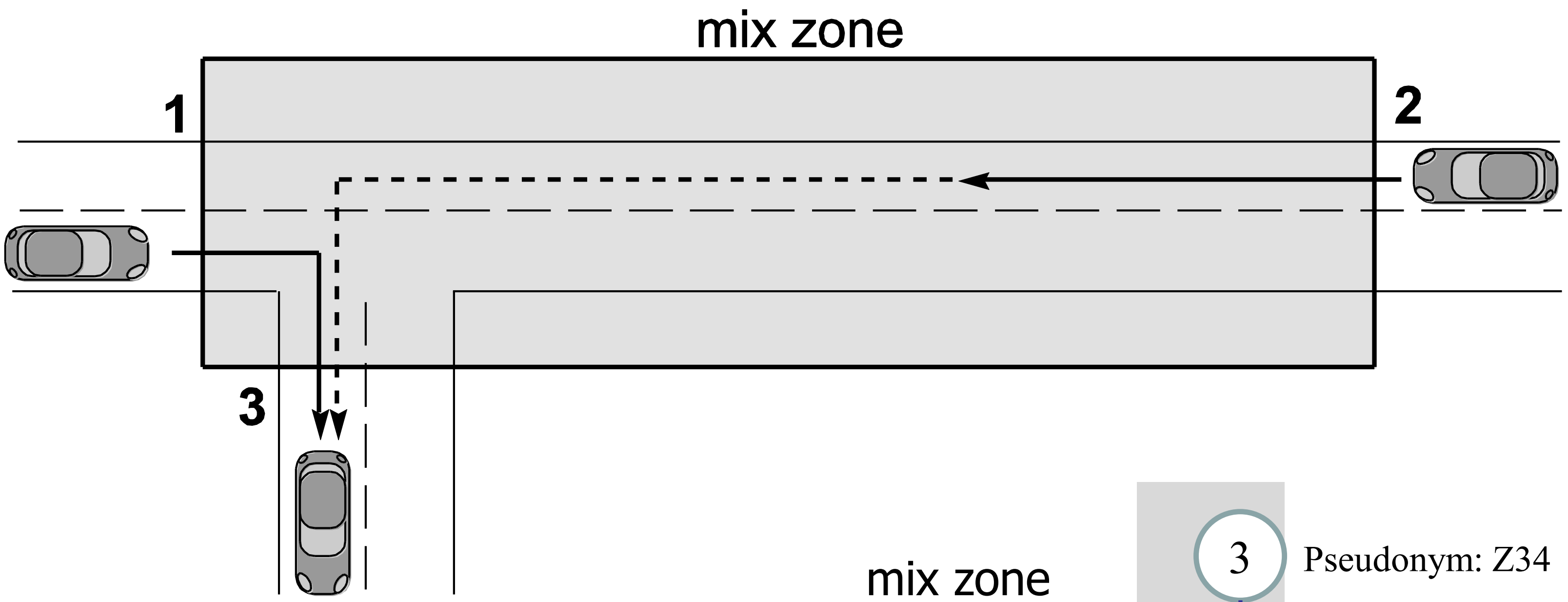
- hence, the adversary is assumed to be able to monitor the communications only at a limited number of places and in a limited range

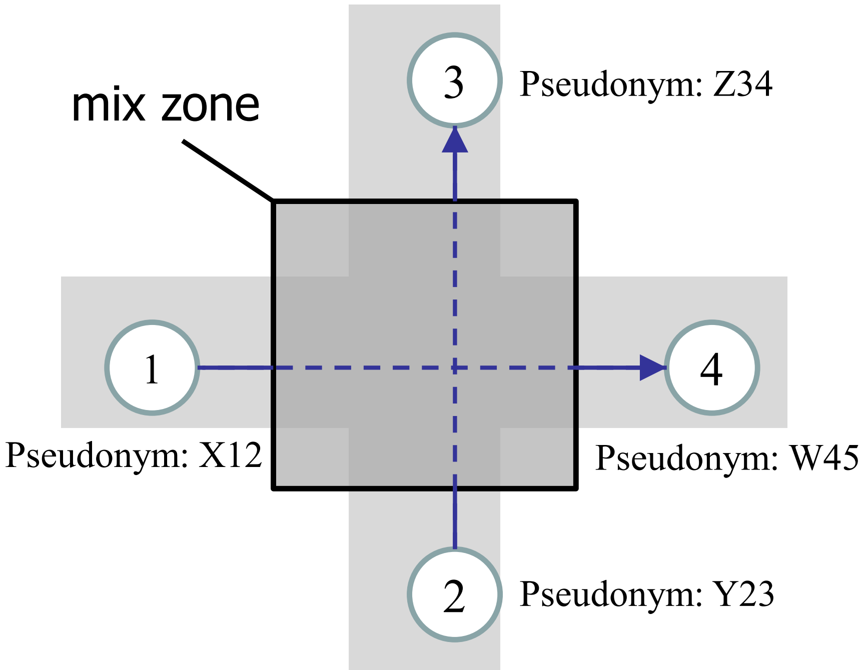
The mix zone concept



- the unobserved zone functions as a *mix zone* where the vehicles change pseudonym and mix with each other
- vehicles do not know where the mix zone is (this depends on where the adversary installs observation spots)
- vehicles change pseudonyms frequently s.t. each vehicle changes pseudonym while in the mix zone

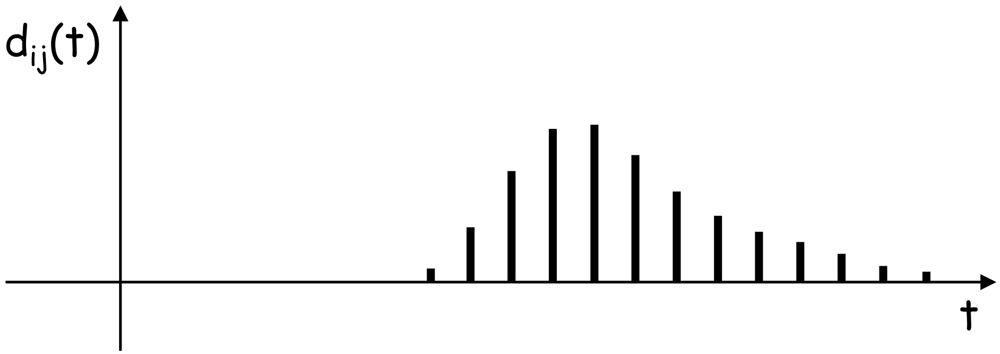
Example of mix zone





Model of the mix zone

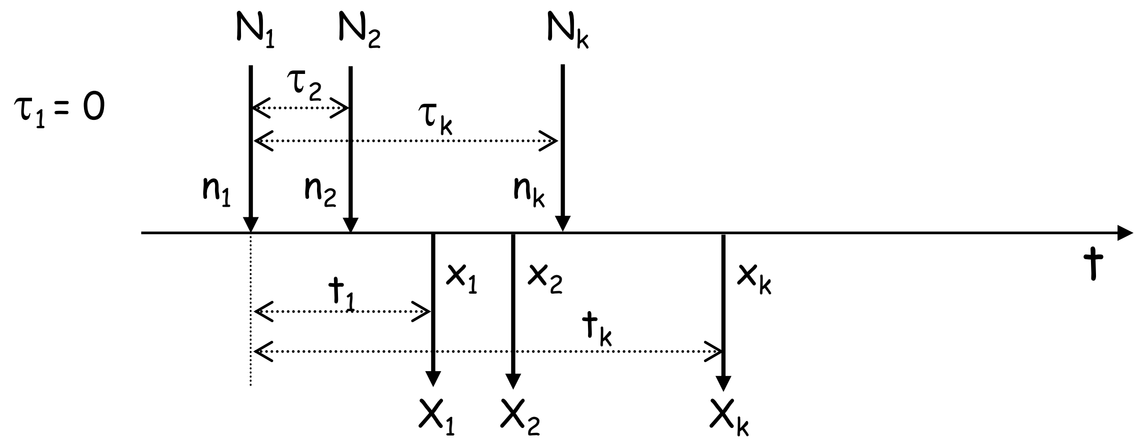
- time is divided into discrete steps
- p_{ij} = Pr{ exiting at j | entering at i }
- D_{ij} is a random variable (delay) that represents the time that elapses between entering at i and exiting at j
- $d_{ij}(t) = Pr\{ D_{ij} = t \}$



- Pr{ exiting at j at t | entering at i at τ } = $p_{ij}\, d_{ij}(t-\tau)$

Observations

- the adversary can observe the points (n_i, x_i) and the times (τ_i, t_i) of enter and exit events (N_i, X_i)



- nodes change pseudonyms inside the mix zone → no easy way to determine which exit event corresponds to which enter event
- each possible mapping between exit and enter events is represented by a permutation π of {1, 2, …, k}:

$$m_\pi = (N_1 \sim X_{\pi[1]}, N_2 \sim X_{\pi[2]}, \ldots, N_k \sim X_{\pi[k]})$$

  where $\pi[i]$ is the i-th element of the permutation
- we want to determine $\Pr\{ m_\pi \mid \overline{N}, \overline{X} \}$

Computing the level of privacy

$$\Pr\{m_\pi | \bar{N}, \bar{X}\} = \frac{\Pr\{m_\pi, \bar{X} | \bar{N}\}}{\Pr\{\bar{X} | \bar{N}\}}$$

where m_π is the mapping described by the permutation π

$$\Pr\{m_\pi, \bar{X} | \bar{N}\} = \prod_{i=1}^{k} p_{n_i x_{\pi(i)}} d_{n_i x_{\pi(i)}} (t_{\pi(i)} - \tau_i) = q_\pi$$

where p_{ij} is a cell of the matrix P of size nxn, where n is the number of gates of the mix zone and $d_{ij}(t)$ describes the probability distribution of the delay when crossing the mix zone from gate i to gate j.

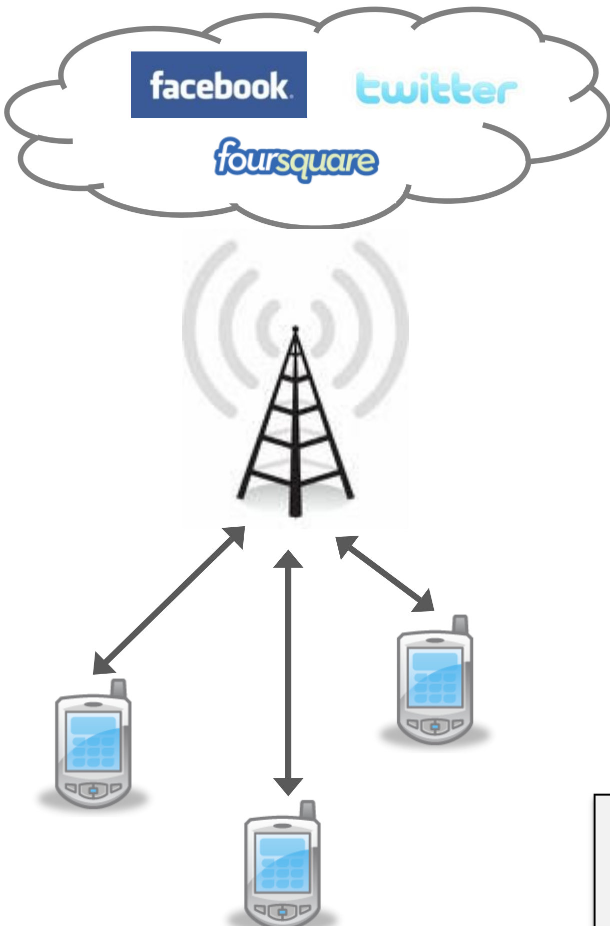
$$\Pr\{\bar{X} | \bar{N}\} = \sum_{\pi'} \Pr\{m_{\pi'}, \bar{X} | \bar{N}\} = \sum_{\pi'} q_{\pi'}$$

$$H(\bar{N}, \bar{X}) = -\sum_{\pi} \frac{q_\pi}{\sum_{\pi'} q_{\pi'}} \log\left(\frac{q_\pi}{\sum_{\pi'} q_{\pi'}}\right)$$

Location-Based Services

- People share their location on-line
  - Social purposes
  - Contextual services

Location-Based Services



Users upload location episodically through WiFi or cellular networks

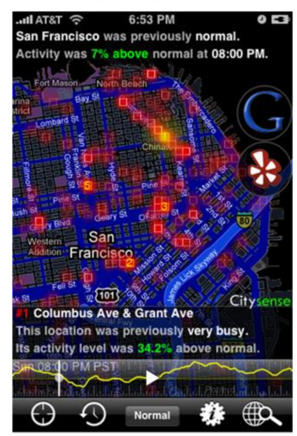
Many possible scenarios, see:

M. Wernke, P. Skvortov, F. Dürr and K. Rothermel. A Classification of Location Privacy Attacks and Approaches. Pers. Ubiquitous Computing (2014)

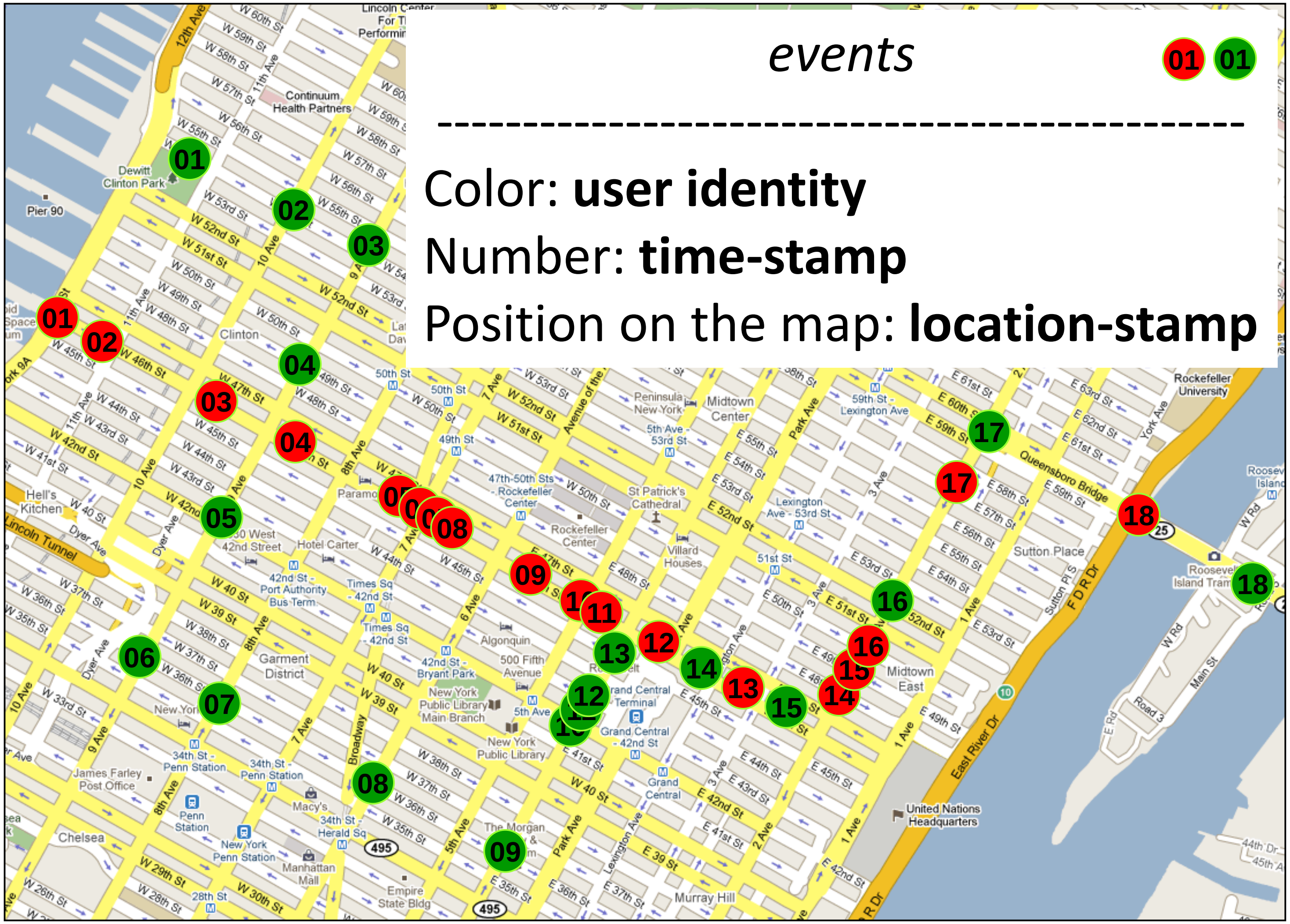
Query, Location, Time

Why Reveal Your Location?

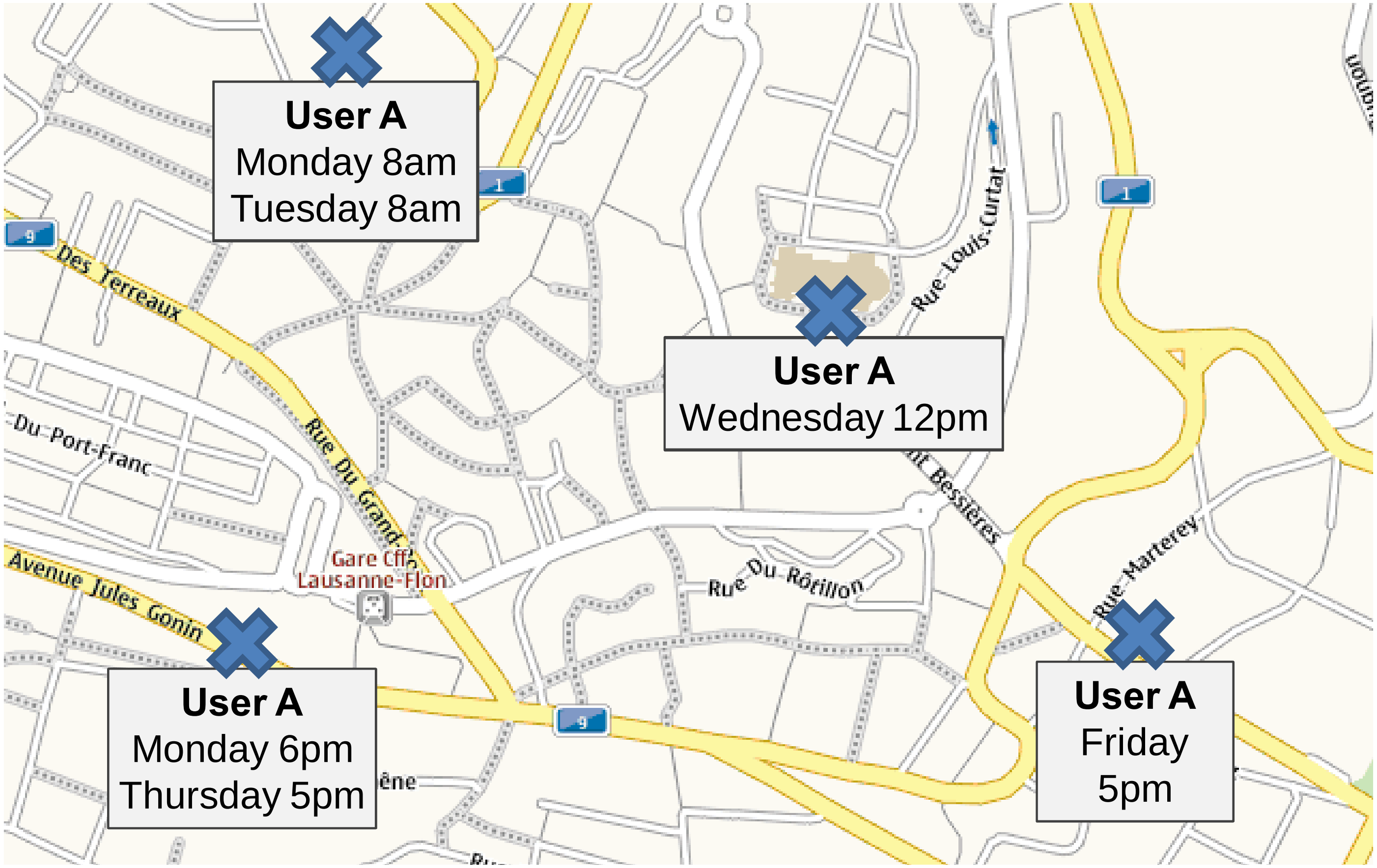
- To use service
  - Cellular connectivity
  - Location-based services
  - Local recommendations
  - Road toll payment
  - …
- For social benefits
  - Find friends



Can You Clean up Your Digital Trace?



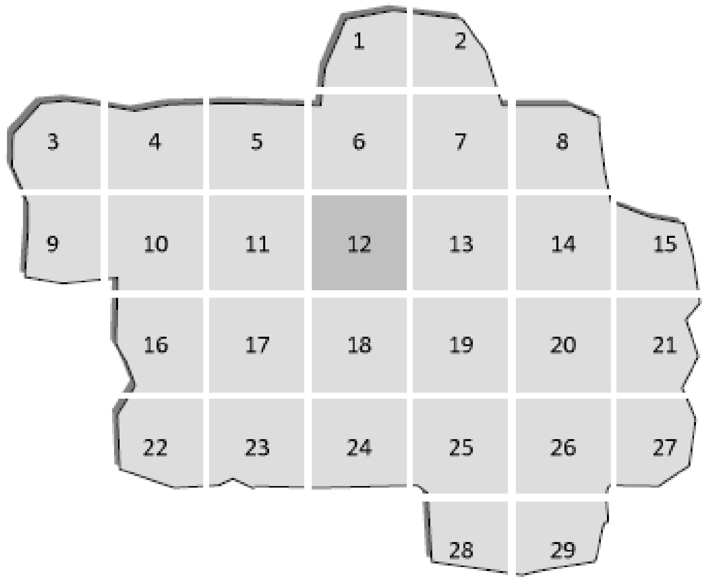
Threat



The contextual information attached to a trace tells much about our habits, interests, activities, beliefs and relationships

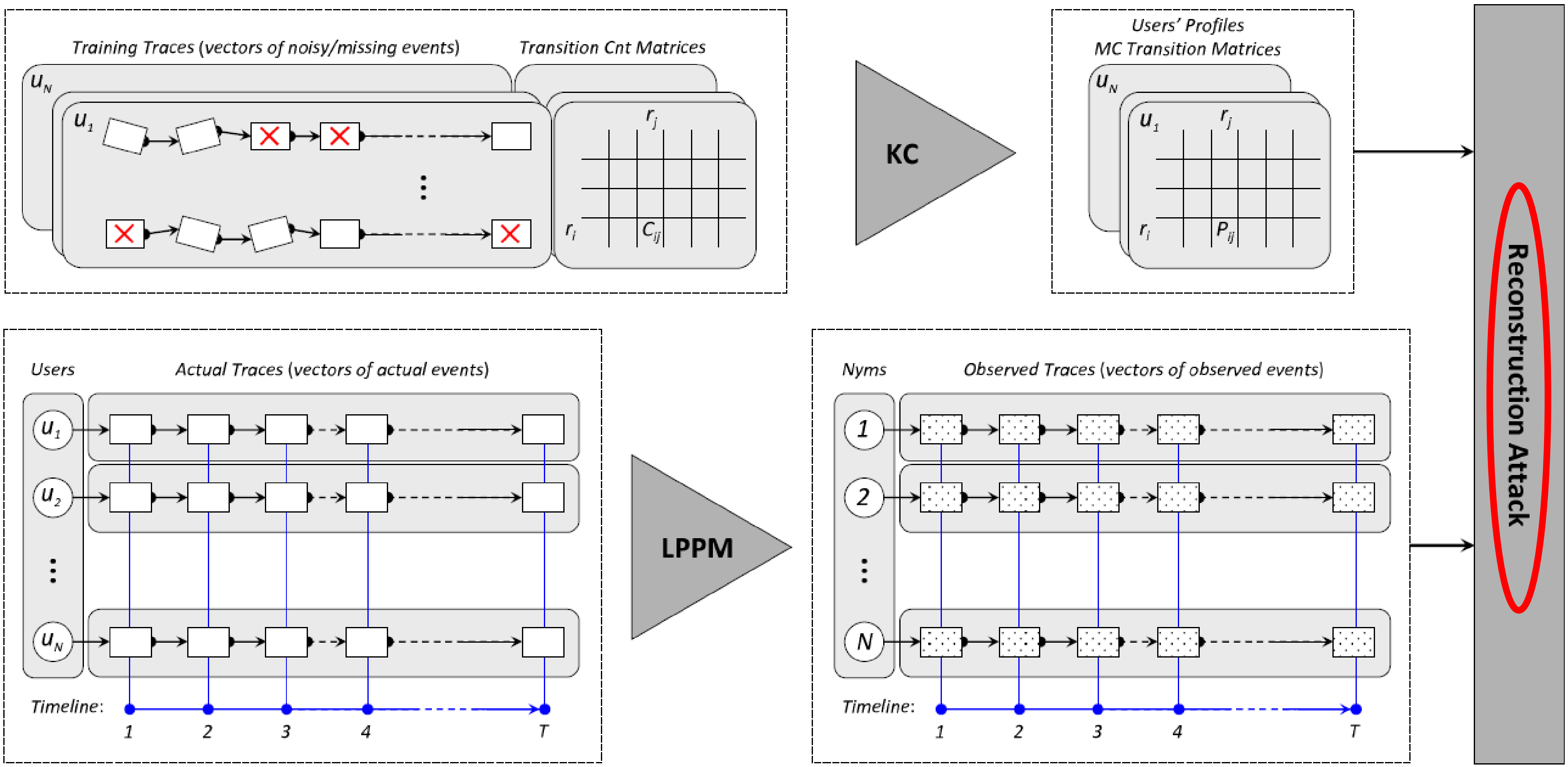
Time and Space

- Consider discrete time and space



- Attacker: service provider (``honest but curious´´)

Quantifying Location Privacy



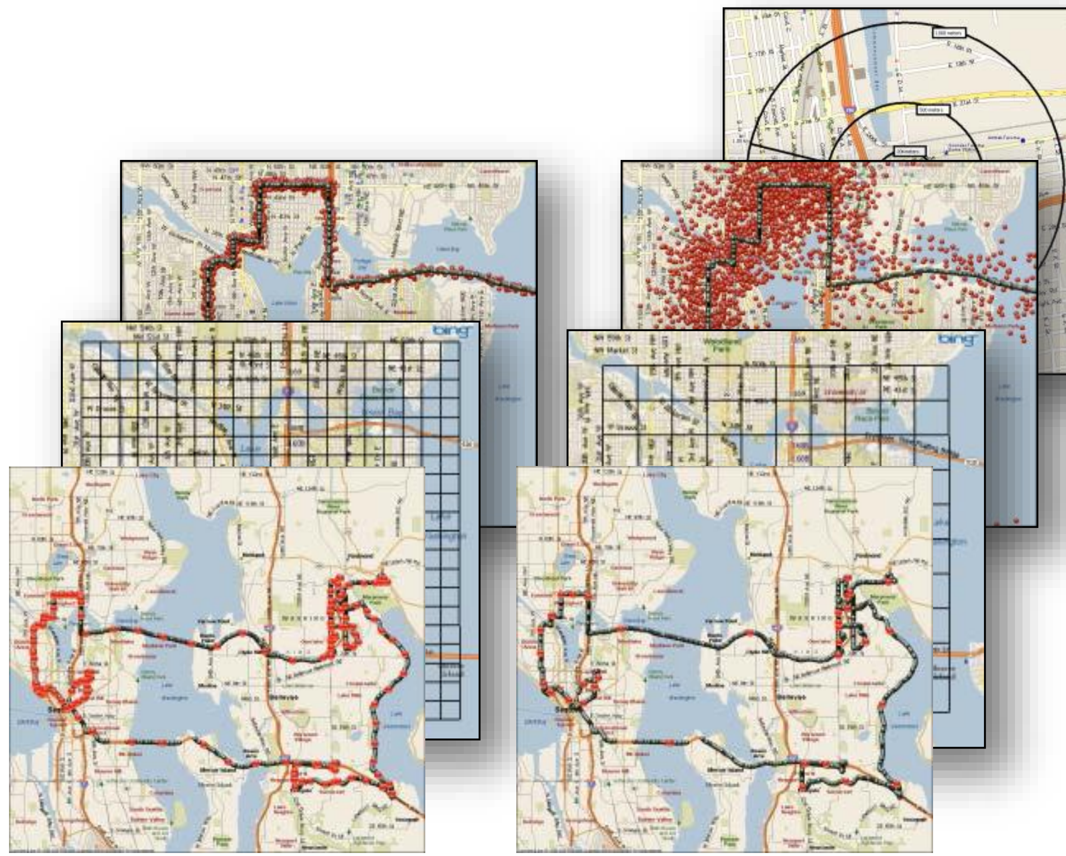
KC: Knowledge Constructor

LPPM: Location Privacy Protection Mechanism:

- deliberately imprecise coordinate reports (e.g., drop some of the least significant bits)
- Swap user identifiers

Protecting location privacy

- Anonymization
  - Pseudonyms
- Obfuscation
  - Deleting
  - Randomizing
  - Discretizing
  - Sub-sampling



All we have seen so far in this module is wonderful… but can it be implemented?

PETs on Android

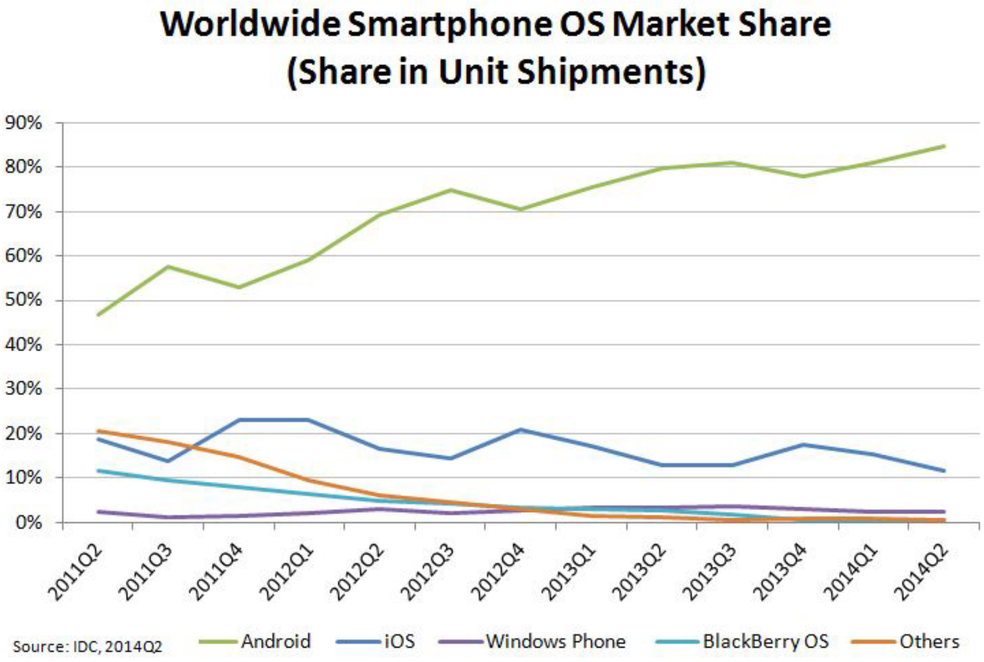
Smartphones

- Mobile phones with multiple computing and communication capabilities
- Increasingly popular – "*Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013*" [1]
- Gather, process and store lots of personal information
  - Location, photos, contacts, emails, etc.
  - New trend: health and fitness data
- *The most personal computing device today!*

[1] http://www.gartner.com/newsroom/id/2665715

Android OS

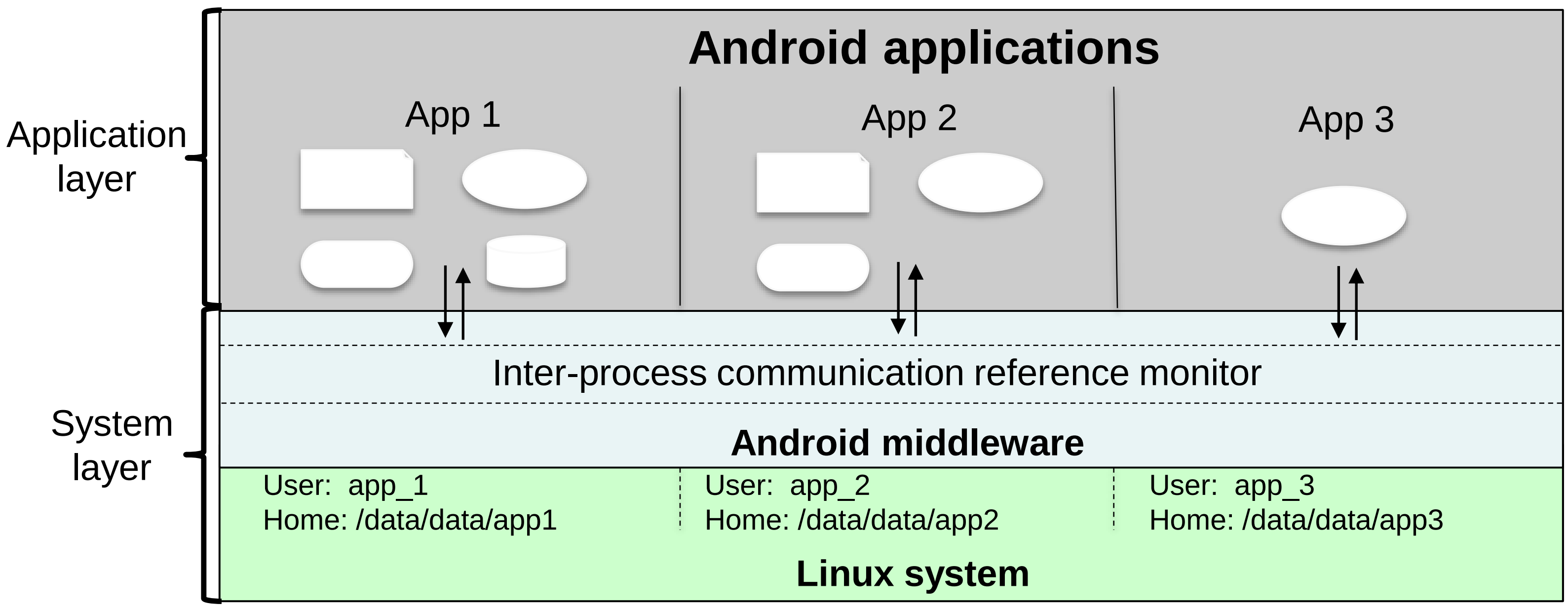
- Released in 2008 by Google
- Open source + some proprietary code
- Java middleware + Linux kernel
- 85% worldwide market share (20142Q) [1]



[1] http://www.idc.com/prodserv/smartphone-os-market-share.jsp

Android's Security Architecture

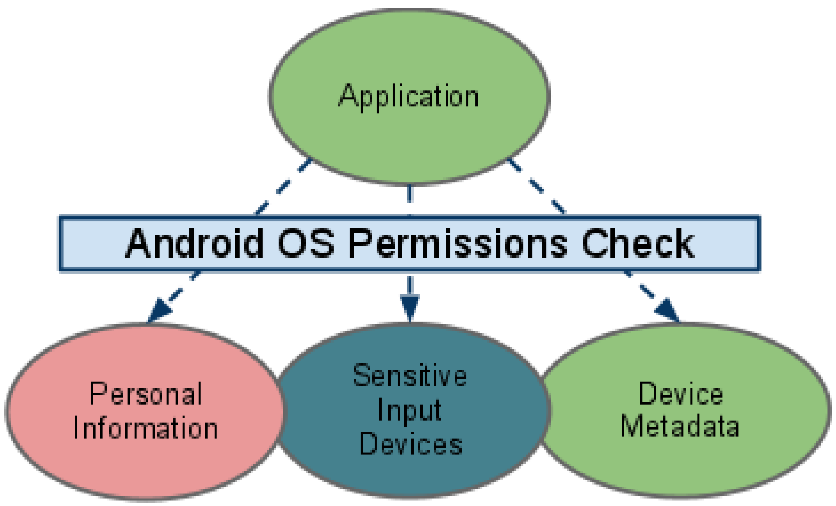
- Application isolation (sandbox)
- Secure inter-process communication
- Application-defined and user-granted permissions



[1] Enck, W., Ongtang, M., & McDaniel, P. Understanding Android Security. Security & Privacy, IEEE, 7, 50–57. 2009

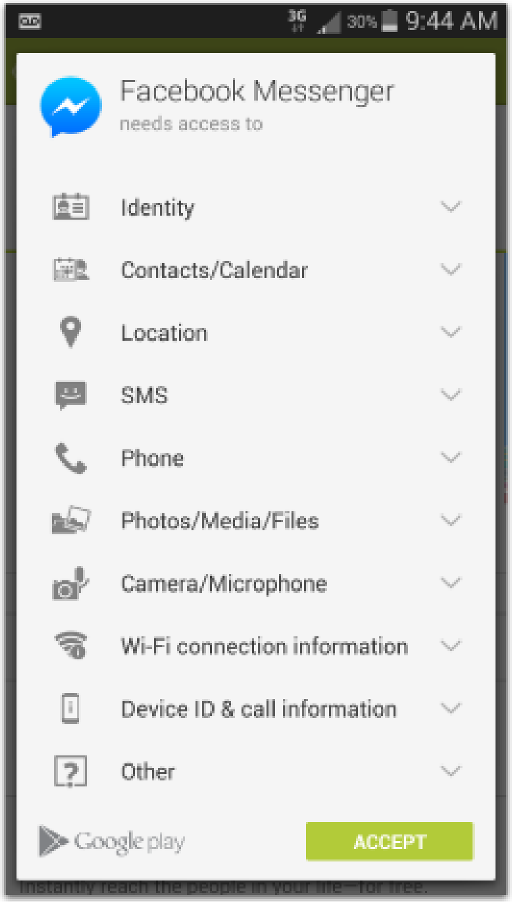
Android Permissions

- Required to access sensitive APIs
- Defined at installation time



Types of Sensitive User Data

[1] https://source.android.com/devices/tech/security/



Problems with Permissions

- Can not be changed after installation (static)
- Coarse-grained (e.g., Internet access)
- Apps keep asking for more
- Users do not understand them well

Permissions and Privacy

- Many apps and third-party libraries (e.g., ads libraries) abuse permissions to collect personal information

Global Privacy Enforcement Network (GPEN) Survey (September 2014)

85% 85% of the apps surveyed failed to clearly explain how they were collecting, using and disclosing personal information.

59% More than half (59%) of the apps left users struggling to find basic privacy information.

1/3 Almost 1 in 3 apps appeared to request an excessive number of permissions to access additional personal information.

http://ico.org.uk/news/latest_news/2014/global-survey-finds-85-percent-of-mobile-apps-fail-to-provide-basic-privacy-information-20140910

PETs on Android

- Goal: to provide users with dynamic, finer-grained and more usable controls to mediate access to their personal information
  - Enforcement of the user's privacy policy
  - Defense against permission-hungry apps
- Main research area:
  - Where to intercept apps' requests (hooks[1])

[1] Hooks: code that handles the interception of function calls, events or messages in an OS, application or other software components

Approaches for Intercepting Requests

	Description	Pros	Cons
App modification	Modify and repackage the app to include interception code	• Easier to deploy (no rooting or OS modification needed)	• Breaks apps' signature/updates • Copyright issues • Every apps needs to be modified • Problems with native code
Rooted device	Use root privileges to dynamically inject interception code in the app	• No modifications to apps or OS required • Rooting is easier than flashing a firmware • Sizeable number of users with rooted phones	• Rooting is not supported by network operators • Rooting breaks OS security model • Most users do not root their phones
OS modification	Modify OS to monitor and intercept requests	• Most robust approach • Apps do not need modifications	• Difficult to deploy as it requires flashing a new firmware (complex operation)

TaintDroid (OSDI 2010)

- TaintDroid [1] is a framework that allow users to monitor how apps handle their private data in real-time
  - It tracks the flow of privacy-sensitive data
- It relies on a system-wide integration of taint tracking into the Android platform

[1] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N. Sheth. **TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones**, Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2010.

Dynamic Taint Analysis

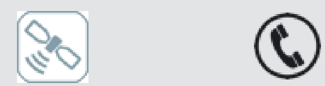
- Dynamic taint analysis is a technique that tracks information dependencies from an origin
- Conceptual idea:
  - Taint source
  - Taint propagation
  - Taint sink
- Tradeoff between performance and granularity

```
c = taint_source()
...
a = b + c
...
network_send(a)
```

Slides adapted from https://www.usenix.org/legacy/events/osdi10/tech/slides/enck.pdf

TaintDroid Application Study

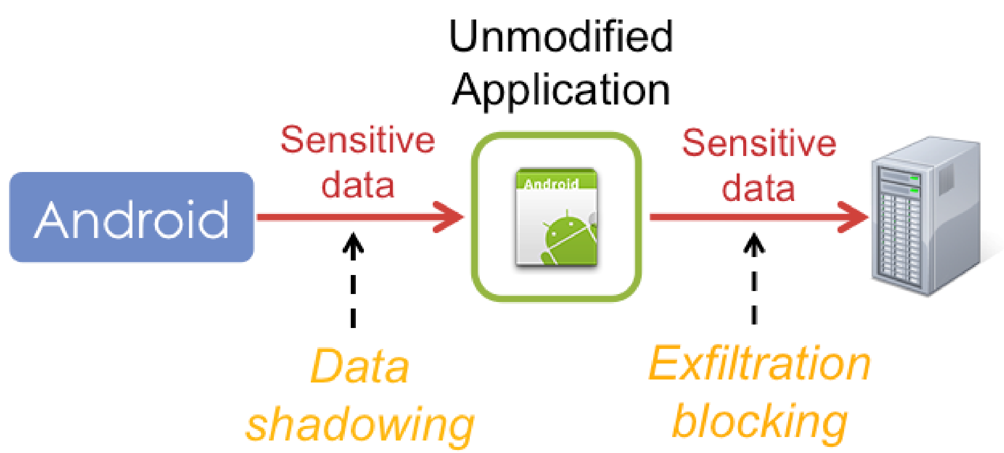
- Selected 30 applications with bias on popularity and access to *Internet*, *location*, *microphone*, and *camera*

applications	#	permissions
The Weather Channel, Cetos, Solitarie, Movies, Babble, Manga Browser	6	
Bump, Wertago, Antivirus, ABC --- Animals, Traffic Jam, Hearts, Blackjack, Horoscope, 3001 Wisdom Quotes Lite, Yellow Pages, Datelefonbuch, Astrid, BBC News Live Stream, Ringtones	14	
Layer, Knocking, Coupons, Trapster, Spongebot Slide, ProBasketBall	6	
MySpace, Barcode Scanner, ixMAT	3	
Evernote	1	

- *Of 105 flagged connections, only 37 clearly legitimate*

AppFence (CCS 2011)

- AppFence [1] extends TaintDroid to include data shadowing and exfiltration blocking
  - Shadowing: app doesn't get sensitive data at all
  - Blocking: app gets sensitive data, but can't send it out



[1] Peter Hornyack, Seungyeop Han, Jaeyeon Jung, Stuart Schechter, and David Wetherall. "**These Aren't the Droids You're Looking For": Retrofitting Android to Protect Data from Imperious Applications**. In Proc. of ACM CCS, October 2011

AppFence – Sensitive Data

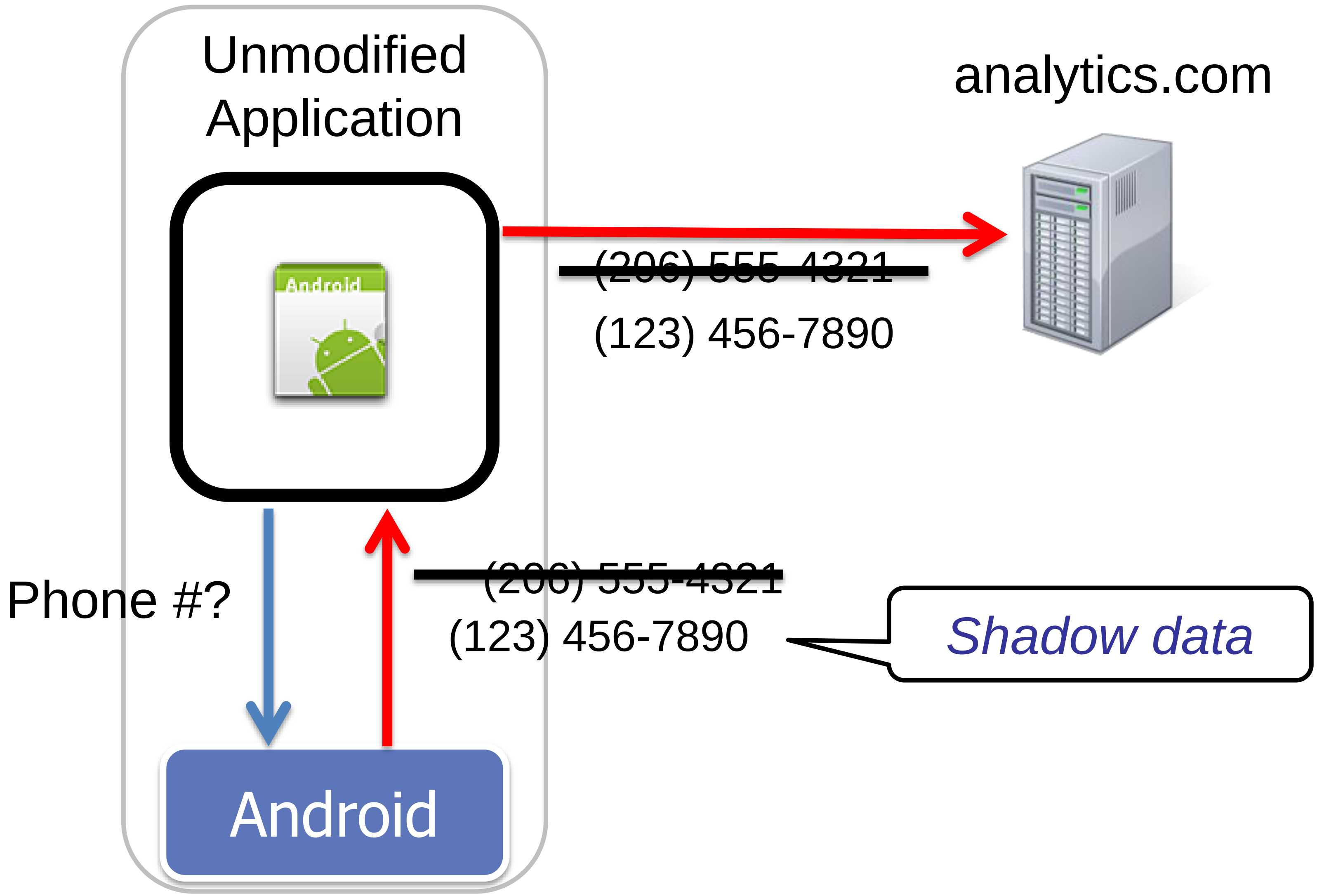
- Authors identified 12 types of privacy-sensitive data on Android

device id
location
phone number
contacts
camera
accounts
logs
microphone
SMS messages
history & bookmarks
calendar
subscribed feeds

Slides adapted from http://homes.cs.washington.edu/~pjh/pres/hornyack_appfence_ccs2011_pres.pptx

How data shadowing works

Without data shadowing:
With data shadowing:

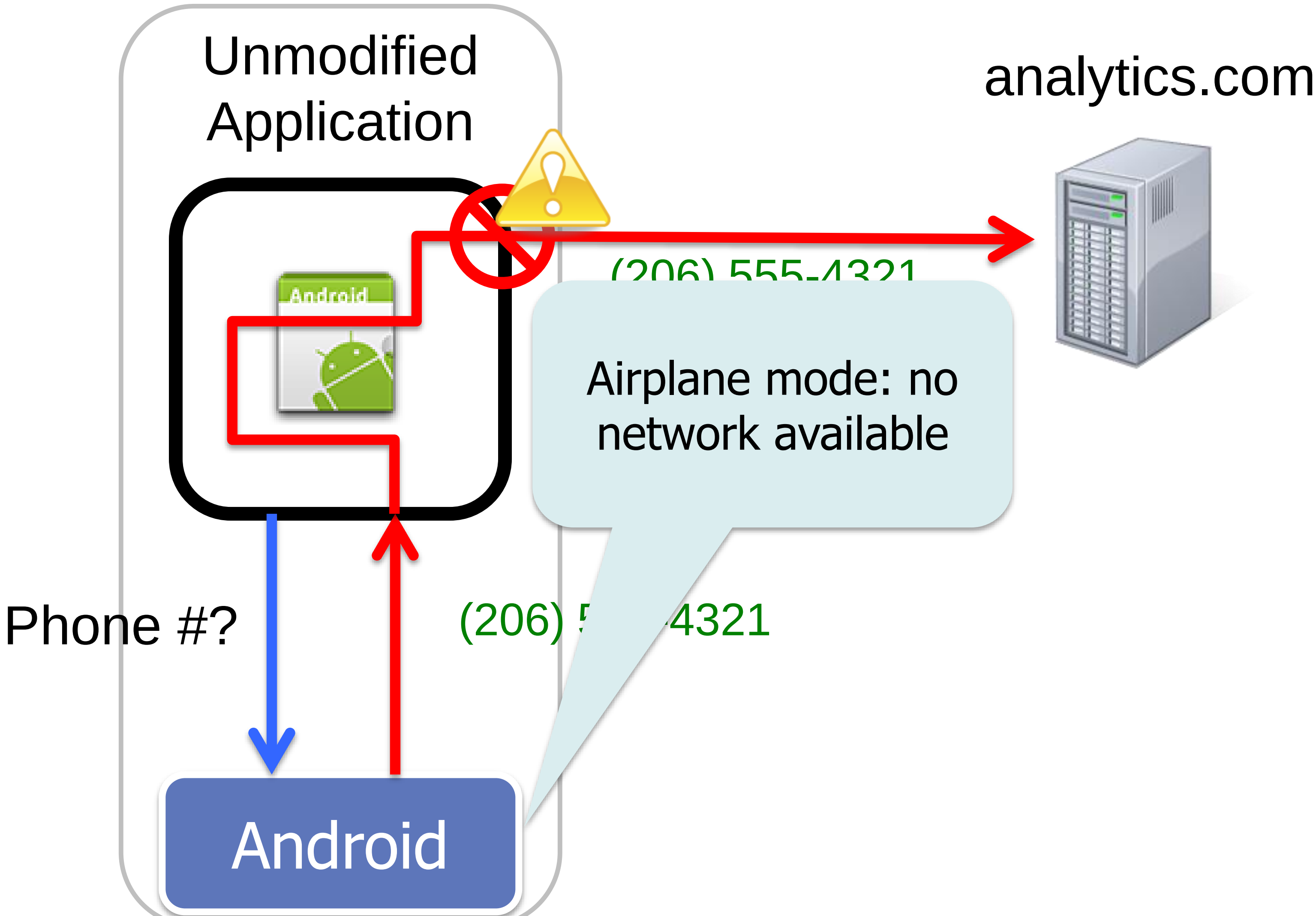


Three Kinds of Shadow Data

- Blank data
  - e.g. contacts: {S. Han, 206-555-4321} ➔ {}
- Fake data
  - e.g. location: {47.653,-122.306} ➔ {41.887,-87.619}
- Constructed data
  - e.g. device ID = *hash*(app name, true device ID)
    - Consistent for each application, but different across applications

How exfiltration blocking works

Without exfiltration blocking:



AppFence Evaluation

- Framework for evaluating impact on user's experience
  - Detecting side effects by combining automated GUI testing with visual highlighting of differences between application screenshots
- Evaluation of AppFence on 50 apps that sent out sensitive data
  - AppFence reduced the effective permissions of 66% of the apps without side effects
  - *Protecting sensitive data will always cause side effects for some apps*

Summary on Location Privacy

- Protecting location privacy is a major challenge
- Quantification of privacy can be expressed as adversary's expected estimation error (incorrectness)
- Techniques to protect location privacy: introduce imprecision in the reported location, reduce location report frequency, make use of pseudonyms,…
- Privacy (similarly to any security property) is adversary-dependent
  - Neglecting adversary's strategy and knowledge limits the privacy protection
- Implementing PETs on smartphones is an unsolved challenge

References

- **M. Wernke, P. Skvortov, F. Dürr and K. Rothermel. A Classification of Location Privacy Attacks and Approaches. Pers. Ubiquitous Computing (2014) 18:163 – 175**
- **R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying Location Privacy. In *Proc. of the IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, USA, 2011**
- **P. Hornyack, S. Han, J. Jungy, S. Schechtery and D. Wetherall. "These Aren't the Droids You're Looking For": Retrofitting Android to Protect Data from Imperious Applications. ACM CCS 2011**