Security and Privacy in Wireless Networks

Mohammad Hossein Manshaei
manshaei@gmail.com

Chapter 7: (secowinet.epfl.ch)

privacy notions and metrics, privacy in RFID systems, location privacy in vehicular networks, privacy preserving routing in MANET

PRIVACY PROTECTION

Chapter outline

8.1 Important privacy related notions and metrics

8.2 Privacy in RFID systems

8.3 Location privacy in vehicular networks

8.4 Privacy preserving routing in ad hoc networks

Privacy related notions

- **Anonymity:** hiding who performed a given action
- **Untraceability:** making difficult for an adversary to identify that a given set of actions were performed by the same subject
- **Unlinkability:** generalization of the two former notions: hiding information about the relationships between *any* item (e.g., subjects, messages, actions, …)
- **Unobservability:** hiding of the items themselves (e.g., hide the fact that a message was sent all)
- **Pseudonymity:** making use of a pseudonym instead of the real identity

Privacy metrics (1/2)

- **Anonymity set:** set of subjects that might have performed the observed action
  - Is a good measure only if all the members of the set are equally likely to have performed the observed action
- **Entropy-based measure of anonymity:**

$$-\sum_{\forall x \in A} p_x . \log p_x$$

where

A is the anonymity set

p_x is the probability (for the adversary)

that the observed action has been performed by subject $x \in A$

Privacy metrics (2/2)

- **Entropy-based measure for unlinkability:**

$$-\sum_{\forall R \subseteq I_1 \times I_2} p_R . \log p_R$$

where

I_1 and I_2 are the sets of items that the adversary wants to relate

p_R is the probability (for the adversary) that the real relationship between the elements in I_1 and in I_2 is captured by relation $R \subseteq I_1 \times I_2$

Chapter outline

8.1 Important privacy related notions and metrics

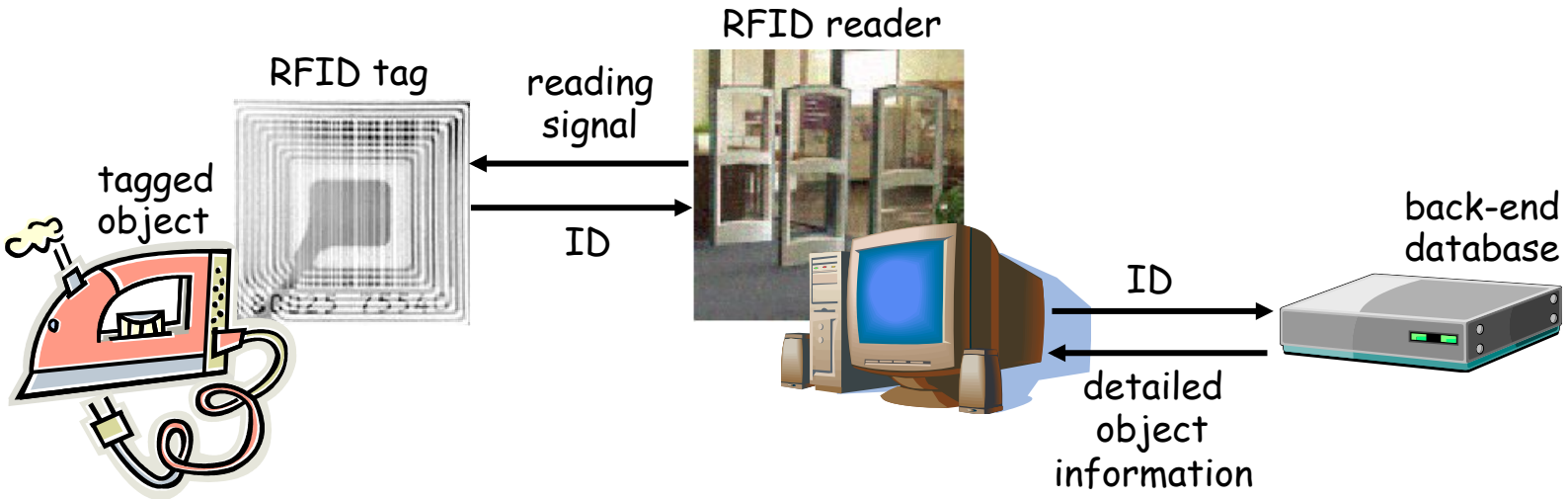
8.2 Privacy in RFID systems

8.3 Location privacy in vehicular networks

8.4 Privacy preserving routing in ad hoc networks

What is RFID?

- RFID = Radio-Frequency Identification
- RFID system elements
  - RFID tag + RFID reader + back-end database
- RFID tag = microchip + RF antenna
  - microchip stores data (few hundred bits)
  - tags can be active
    - have their own battery → expensive
  - or passive
    - powered up by the reader's signal
    - reflect the RF signal of the reader modulated with stored data



RFID applications today

- proximity cards
  - electronic tickets for public transport systems (AFC)
  - access control to buildings
- automated toll-payment transponders
- anti-theft systems for cars
  - RFID transponder in ignition keys
- payment tokens
  - contactless credit cards (e.g., Mastercard PayPass™)
- identification of animals
- identification of books in libraries

- …

RFID applications in the near future

- replacement of barcodes
  - advantages
    - no need for line-of-sight
    - hundreds of tags can be read in a second
    - unique identification of objects
    - easy management of objects throughout the entire supply chain (manufacturer → retailer → consumer)
  - standardization is on the way
    - EPC (Electronic Product Code) tag
  - main issue is price
    - today an EPC tag costs 13 cents
    - massive deployment is expected when price goes below 5 cents
- e-passports
- embedding RFID tags in Euro banknotes
  - anti-counterfeiting
  - detection of money laundering

RFID applications in the future (perhaps)

- **Shopping**
  - fast check-out at point-of-sale terminals
    - terminal reads all tags in the shopping cart in a few seconds
    - payment can be speeded up using contactless credit cards
  - return items without receipt
    - no need to keep receipts of purchased items
  - tracking faulty or contaminated products
    - object IDs can serve as indices into purchase records
    - one can easily list all records that contain IDs belonging to a particular set of products and identify consumers that bought those products

- **Smart household appliances**
  - washing machine can select the appropriate program by reading the tags attached to the clothes
  - refrigerator can print shopping lists automatically or even order food on-line

- **Interactive objects**
  - consumers can interact with tagged objects through their mobile phones (acting as an RFID reader)
  - the mobile phone can download and display information about scanned objects (e.g., movie poster, furniture, etc.)

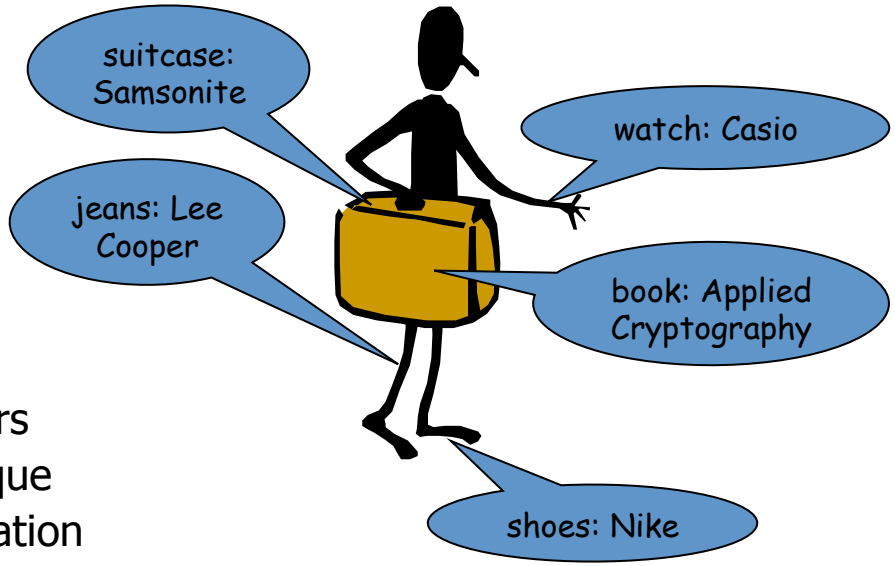
RFID privacy problems

- RFID tags respond to reader's query automatically, without authenticating the reader

  → **clandestine scanning of tags is a plausible threat**

- two particular problems:

1. **inventorying:** a reader can silently determine what objects a person is carrying
   - books
   - medicaments
   - banknotes
   - underwear
   - …
2. **tracking:** set of readers can determine where a given person is located
   - tags emit fixed unique identifiers
   - even if tag response is not unique it is possible to track a constellation of a set of particular tags



RFID read ranges

- **nominal read range**
  - max distance at which a normally operating reader can reliably scan tags
  - e.g., ISO 14443 specifies 10 cm for contactless smart cards
- **rogue scanning range**
  - rogue reader can emit stronger signal and read tags from a larger distance than the nominal range
  - e.g., ISO 14443 cards can possibly be read from 50-100 cm
- **tag-to-reader eavesdropping range**
  - read-range limitations result from the requirement that the reader powers the tag
  - however, one reader can power the tag, while another one can monitor its emission (eavesdrop)
  - e.g., RFID enabled passports can be eavesdropped from a few meters
- **reader-to-tag eavesdropping range**
  - readers transmit at much higher power than tags
  - readers can be eavesdropped form much further (kilometers?)
  - readers may reveal tag specific information

Classification of privacy protection approaches

- **standard tags**
  - "kill" command
  - "sleep" command
  - renaming
  - blocking
  - legislation

- **crypto enabled tags**
  - tree-approach
  - synchronization approach
  - hash chain based approach

Dead tags tell no tales

- **idea: permanently disable tags with a special "kill" command**
- part of the EPC specification

- **advantages:**
  - simple
  - effective

- **disadvantages:**
  - eliminates all post-purchase benefits of RFID for the consumer and for society
    - no return of items without receipt
    - no smart house-hold appliances
    - …
  - cannot be applied in some applications
    - library
    - e-passports
    - banknotes
    - …

"Sleep" command

- **idea**:
  - instead of killing the tag put it in sleep mode
  - tag can be re-activated if needed

- **advantages:**
  - simple
  - effective

- **disadvantages:**
  - difficult to manage in practice
    - tag re-activation must be password protected
    - how the consumers will manage hundreds of passwords for their tags?
    - passwords can be printed on tags, but then they need to be scanned optically or typed in by the consumer

Renaming (1/3)

- **Idea:**
  - get rid of fixed names (identifiers)
  - use random pseudonyms and change them frequently

- **Requirements:**
  - only authorized readers should be able to determine the real identifier behind a pseudonym
  - standard tags cannot perform computations → next pseudonym to be used must be set by an authorized reader

Renaming (2/3)

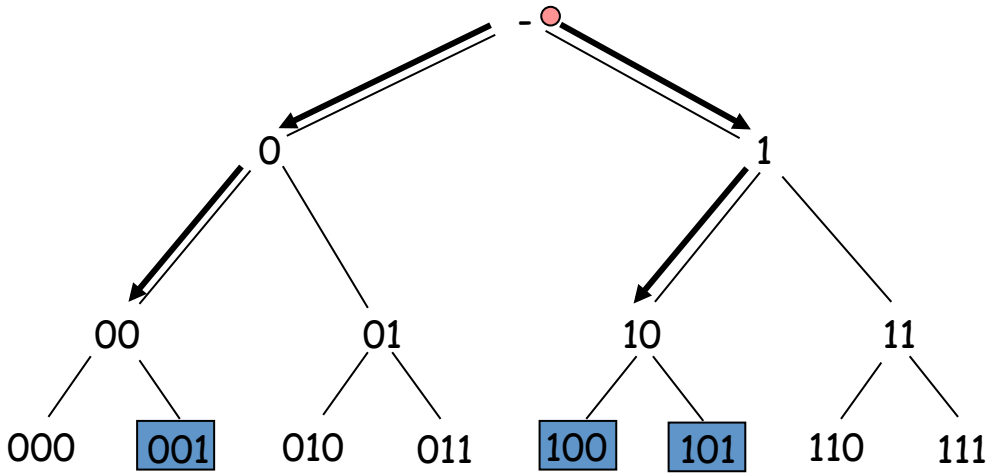
- **A possible implementation**
  - pseudonym = $\{R|ID\}_K$
    - R is a random number
    - K is a key shared by all authorized readers
  - authorized readers can decrypt pseudonyms and determine real ID
  - authorized readers can generate new pseudonyms
  - for unauthorized readers, pseudonyms look like random bit strings
- **Potential problems**
  - tracking is still possible between two renaming operations
  - if someone can eavesdrop during the renaming operation, then she may be able to link the new pseudonym to the old one
  - no reader authentication → rogue reader can overwrite pseudonyms in tags (tags will be erroneously identified by authorized readers)

Renaming (3/3)

- **A public key based implementation:**
  - El Gamal scheme:
    - public key is (p, g, A), the cleartext is m
      - p large prime
      - g is a generator of the multiplicative group Z^*_p
      - $A=g^a \pmod p$, where a is a secret value known only to Alice
    - select a random integer r, and compute $R = g^r \bmod p$
    - compute $C = m \cdot A^r \bmod p$
    - the ciphertext is the pair (R, C)
  - *one can re-encrypt a ciphertext (R, C) without decryption:*
    - select a random integer r', and compute $R' = Rg^{r'} \bmod p$ ($= g^{r+r'} \bmod p$)
    - compute $C' = CA^{r'} \bmod p$ ($= mA^{r+r'} \bmod p$)
    - (R', C') is a valid ciphertext of m
  - new tag pseudonyms can be computed by readers that know the public key
  - real tag ID can be computed only by readers that know the private key

Blocking (1/2)

- **binary tree walking**
  - a mechanism to determine which tags are present (singulation procedure)
  - IDs are leaves of a binary tree
  - reader performs a depth first search in the tree as follows
    - reader asks for the next bit of the ID starting with a given prefix
    - if every tag's ID starts with that prefix, then no collision will occur, and the reader can extend the prefix with the response
    - if there's a collision, then the reader recurses on both possible extensions of the prefix



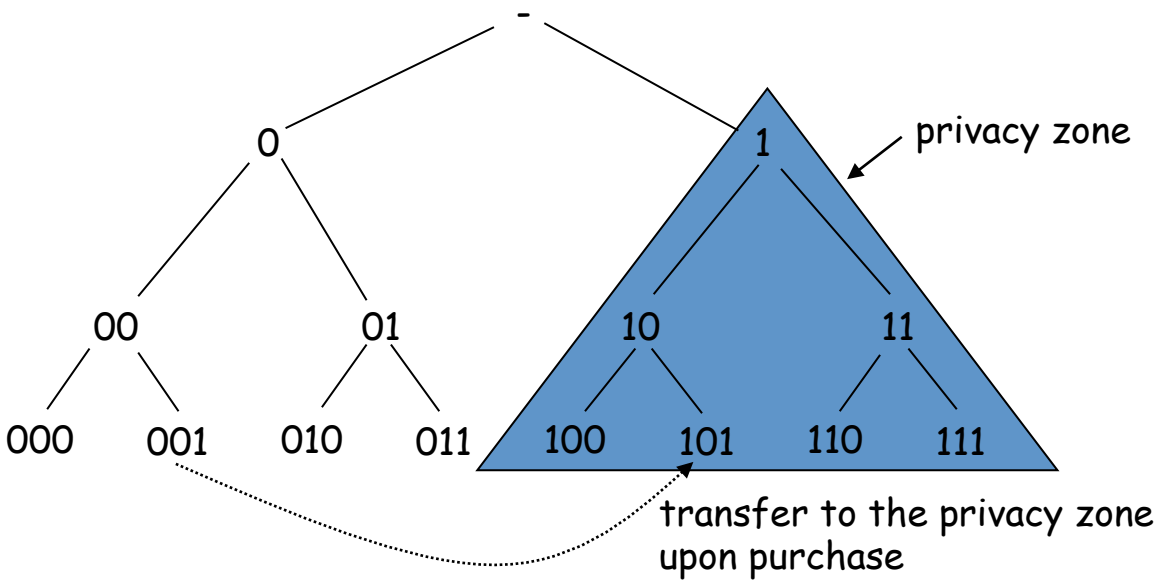
reader: prefix "-" ?
tags: collision
reader: prefix "0" ?
tags: 0
reader: prefix "00" ?
tags: 1 → 001
reader: prefix "1" ?
tags: 0
reader: prefix "10" ?
tags: collision → 100
101

Note: real tag sizes are much larger (e.g., 96 bits for EPC)

Blocking (2/2)

- **Privacy zone**
  - tree is divided into two zones
    - privacy zone: all IDs starting with 1
  - upon purchase of a product, its tag is transferred into the privacy zone by setting the leading bit

- **The blocker tag (special device carried by the user)**
  - when the prefix in the reader's query starts with 1, it simulates a collision
  - when the blocker tag is present, all IDs in the privacy zone will appear to be present for the reader
  - when the blocker tag is not present, everything works normally



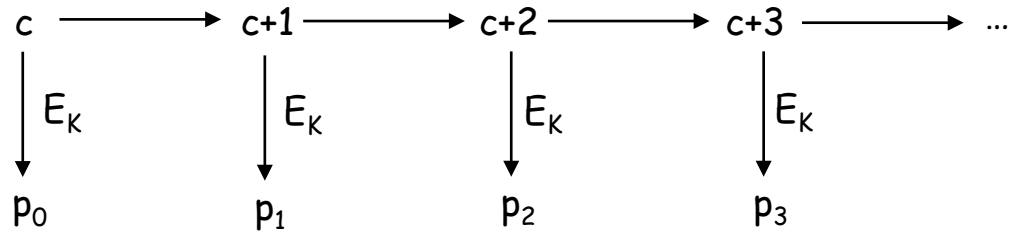
Crypto Enabled Tags

- Assume that tags can perform some crypto operations

→ tags can compute their own pseudonyms !

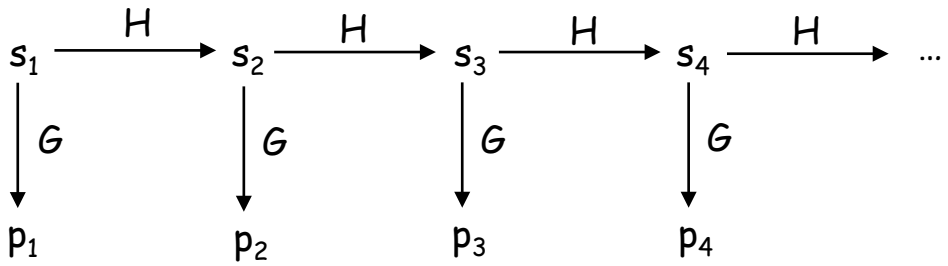
- A solution that doesn't scale:
  - next pseudonym = $\{R, S, ID\}_K$
    - R is a random number generated by the tag (ensures that pseudonyms look random and they are different)
    - S is some redundancy (ensures that the reader can determine if it used the right key to decrypt the pseudonym)
    - ID is the real identifier
    - K is a key shared by the tag and the reader
  - the reader tries all possible keys until it finds the right one
  - if there are many tags, then the verification may be too slow

Synchronization Approach



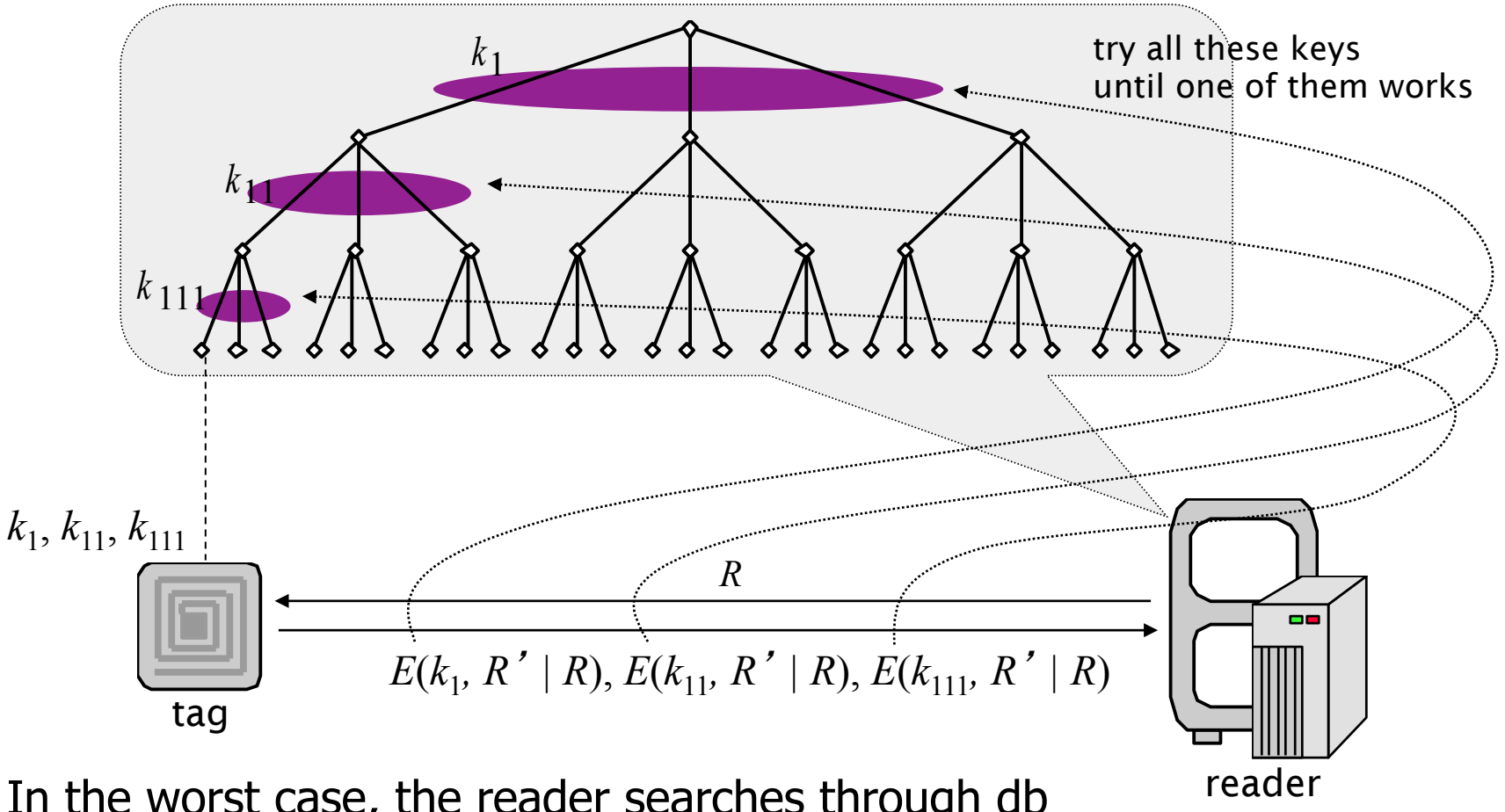
- c is a counter, K is a key shared by the tag and the reader
- **Operation of tag:**
  - when queried by the reader, the tag responds with its current pseudonym $p = E_K(c)$ and increments the counter
- **Operation of the reader:**
  - reader must know approximate current counter value
  - for each tag, it maintains a table with the most likely current counters and corresponding pseudonyms $(c+1, p_1)\ldots(c+d, p_d)$
  - when a tag responds with a pseudonym p, it finds p in any of its tables, identifies the tag, and updates the table corresponding to the tag
- one-wayness of $E_K()$ ensures that current counter value cannot be computed from observed pseudonym

Hash-chain Based Approach



- H and G are one-way functions (e.g., hash functions)
- **Operation of the tag:**
  - current state is s_i
  - when queried the tag responds with the current pseudonym $p_i = G(s_i)$ and computes its new state $s_{i+1} = H(s_i)$
- **Operation of the reader is similar to the previous approach**
- one-wayness of H ensures *forward secrecy* :
  - even if a disposed tag is broken and its current state is determined, previous states (and pseudonyms) cannot be computed

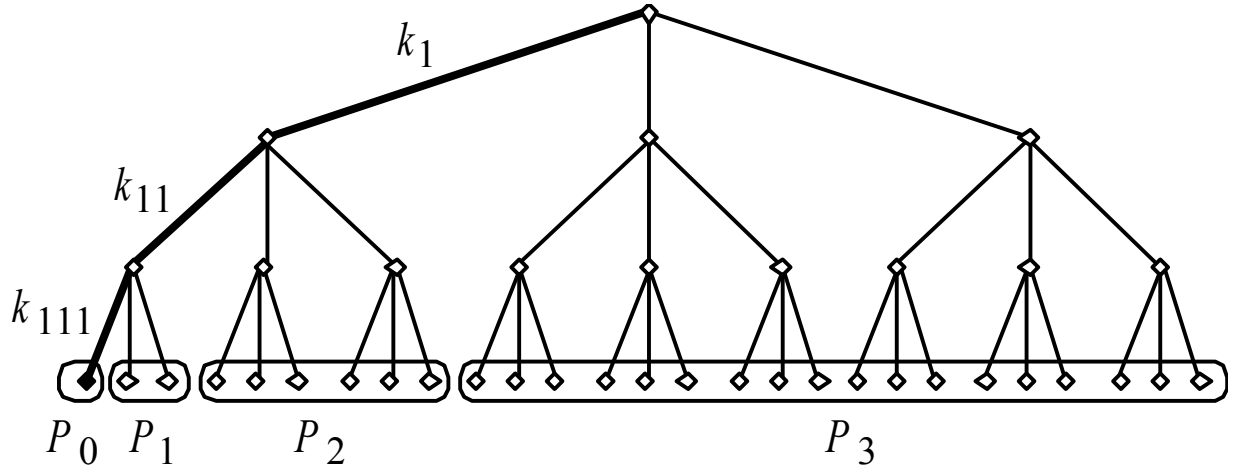
The tree-based approach



- In the worst case, the reader searches through db keys, where d is the depth of the tree, and b is the branching factor
- Complexity is *O(log n)*
- compare this to b^d, which is the total number of tags !

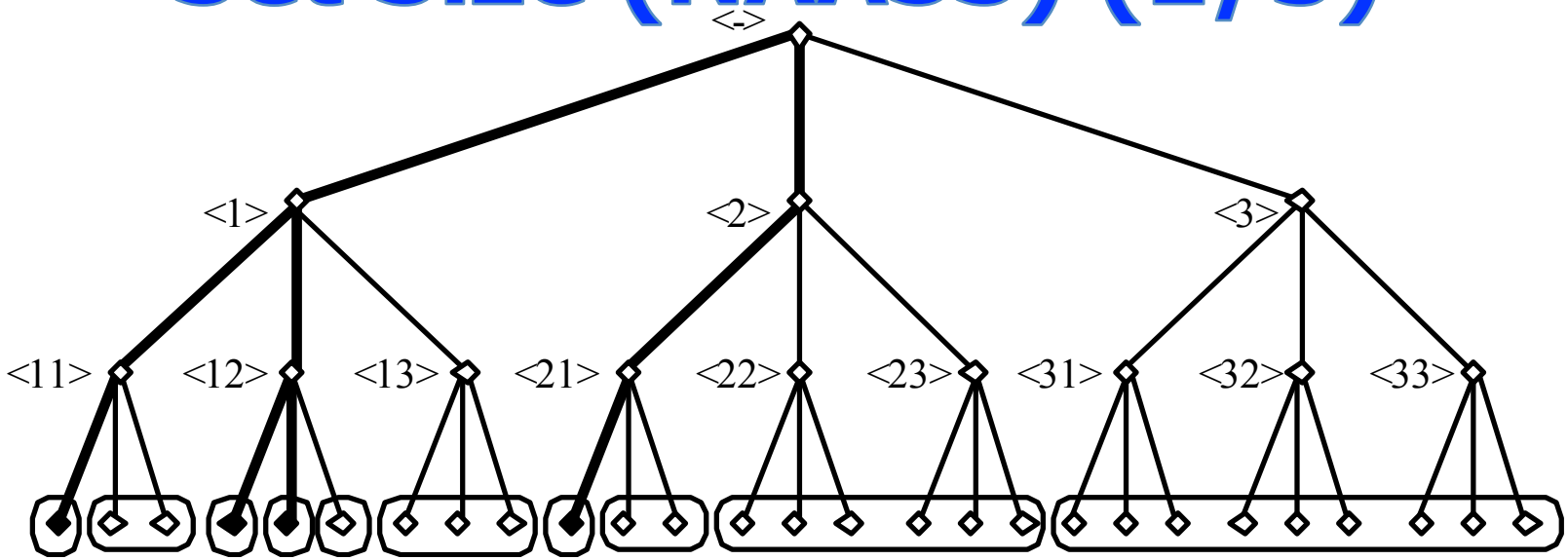
k_1, k_{11}, k_{111} → tag ID

Optimal key-trees



- If tags get compromised, then the level of privacy provided decreases
- This loss of privacy can be minimized by careful design of the tree
- Problem can be formalized as an optimization problem:
  - given the number N of tags to be supported and an upper bound D on the maximum authentication delay allowed
  - determine tree parameters (branching factor at each level) such that
    - loss of privacy is minimized
    - bound on authentication delay is respected
- The solution is:
  - **one should maximize the branching factor at the first level of the tree**

Normalized Average Anonymity Set Size (NAASS) (1/3)



- compromised tags partition the set of all tags
  - tags in a given partition are indistinguishable
  - tags in different partitions can be distinguished

Normalized Average Anonymity Set Size (NAASS) (2/3)

- The level of privacy provided by the system to a randomly selected tag is characterized by the *average anonymity set size*:

$$\bar{S} = \sum_{i=0}^{\ell} \frac{|P_i|}{N} |P_i| = \sum_{i=0}^{\ell} \frac{|P_i|^2}{N}$$

where N is the total number of tags, P_i is a partition, and the sum is computed over all the partitions

- This can be normalized to obtain a metric value between 0 and 1:

$$R = \frac{\bar{S}}{N} = \sum_{i=0}^{\ell} \frac{|P_i|^2}{N^2}$$

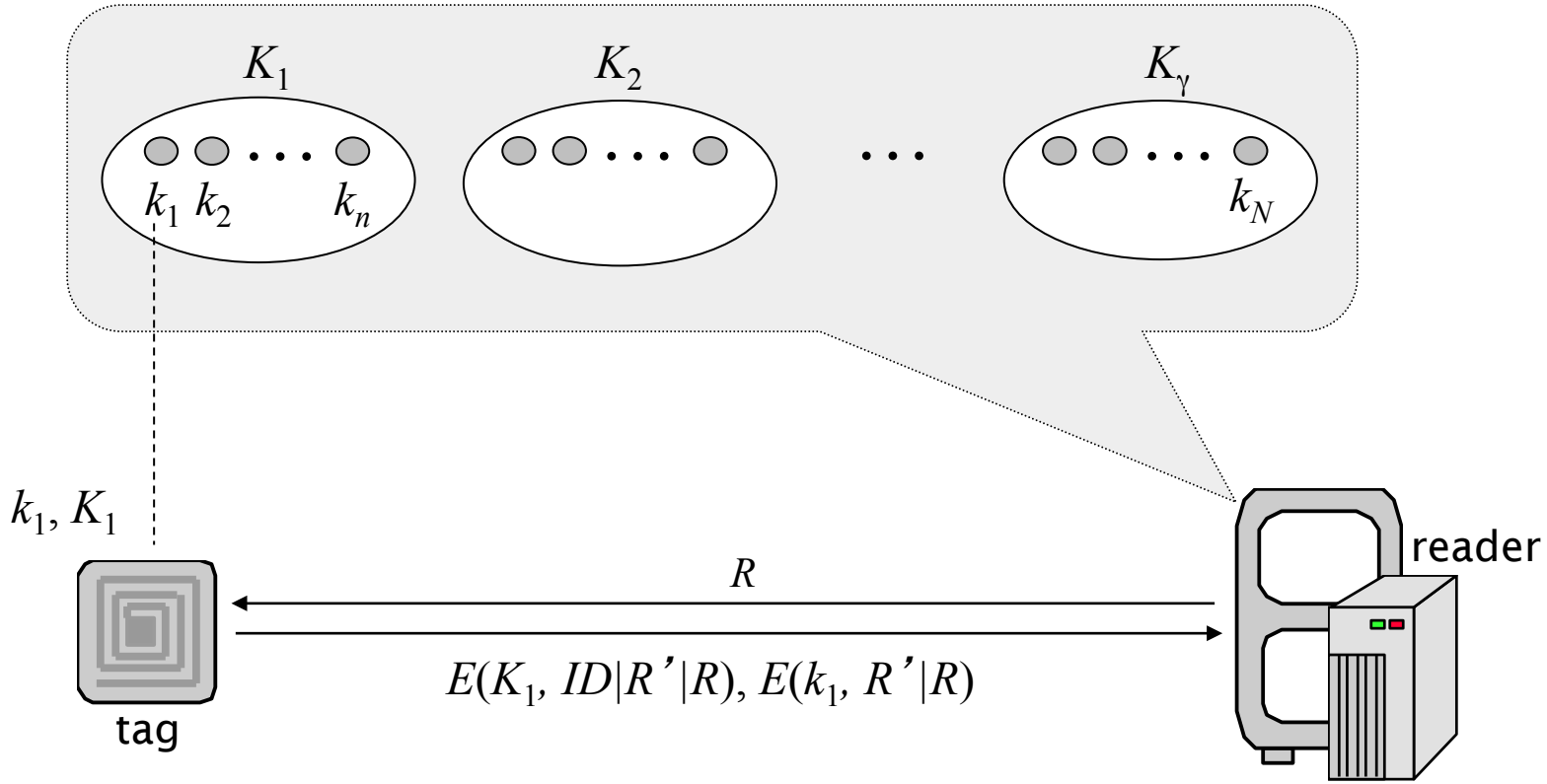
Normalized Average Anonymity Set Size (NAASS) (3/3)

- Computing NAASS for regular trees (same branching factor at each level) when a single tag is compromised:

$$
\begin{aligned}
\frac{\bar{S}}{n} &= \sum_{i=0}^{\ell} \frac{|P_i|^2}{n^2} \\
&= \frac{1}{n^2}\left(1 + (b-1)^2 + ((b-1)b)^2 + \ldots + ((b-1)b^{\ell-1})^2\right) \\
&= \frac{1}{n^2}\left(1 + (b-1)^2\left(1 + b^2 + (b^2)^2 + \ldots + (b^2)^{\ell-1}\right)\right) \\
&= \frac{1}{n^2}\left(1 + (b-1)^2 \cdot \frac{b^{2\ell}-1}{b^2-1}\right) \\
&= \frac{b-1}{b+1} + \frac{2}{(b+1)n^2}
\end{aligned}
$$

$$
\begin{aligned}
& n = b^{\ell} \\
|P_0| &= 1 \\
|P_1| &= b-1 \\
|P_2| &= (b-1)b \\
|P_3| &= (b-1)b^2 \\
\ldots &\quad \ldots \\
|P_{\ell}| &= (b-1)b^{\ell-1}
\end{aligned}
$$

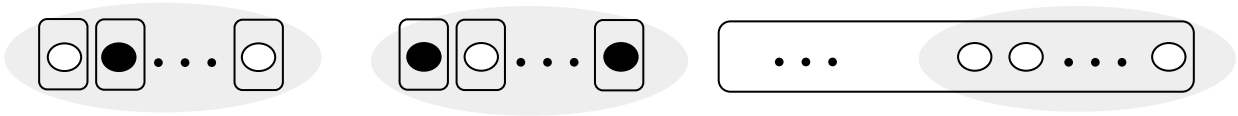
The group-based approach



immediate advantage:
each tag stores and uses only only two keys

1.) try all group keys until one of them works
2.) authenticate the tag by using its individual key

Computing NAASS for Groups



- partitioning depends on the number *C* of compromised *groups*
- NAASS can be computed as:

$$\frac{\bar{S}}{N} = \sum_{\forall i} \frac{|P_i|^2}{N^2} = \frac{nC + (n(\gamma - C))^2}{N^2}$$

- if tags are compromised randomly, then *C* is a random variable
  - we are interested in the expected value of S/N
  - for this we need to compute E[C] and E[C^2]

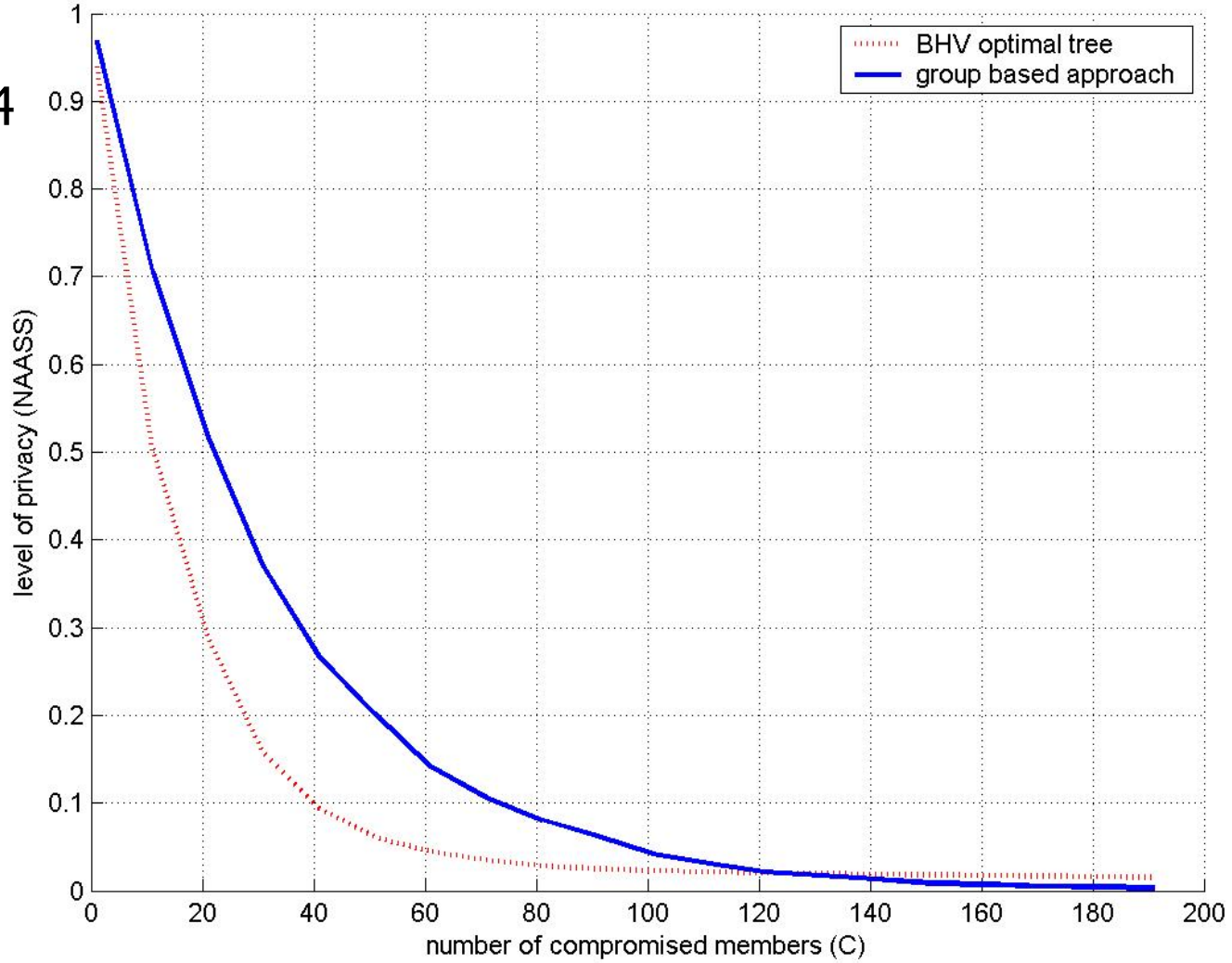
Comparison of Trees and Groups

- select a privacy metric (e.g., NAASS)
- for a given set of parameters (number N of tags, max authentication delay D), determine the optimal key-tree
- compute the privacy metric for the optimal tree (as a function of the number c of compromised tags)
- determine the corresponding parameters for the group based approach (γ = D-1)
- compute the privacy metric for the groups (as function of c)

Comparison in NAASS for a specific N and D pair

$N = 2^{14}$

$D = 65$



[32 16 8 4]
64 x 256

Chapter outline

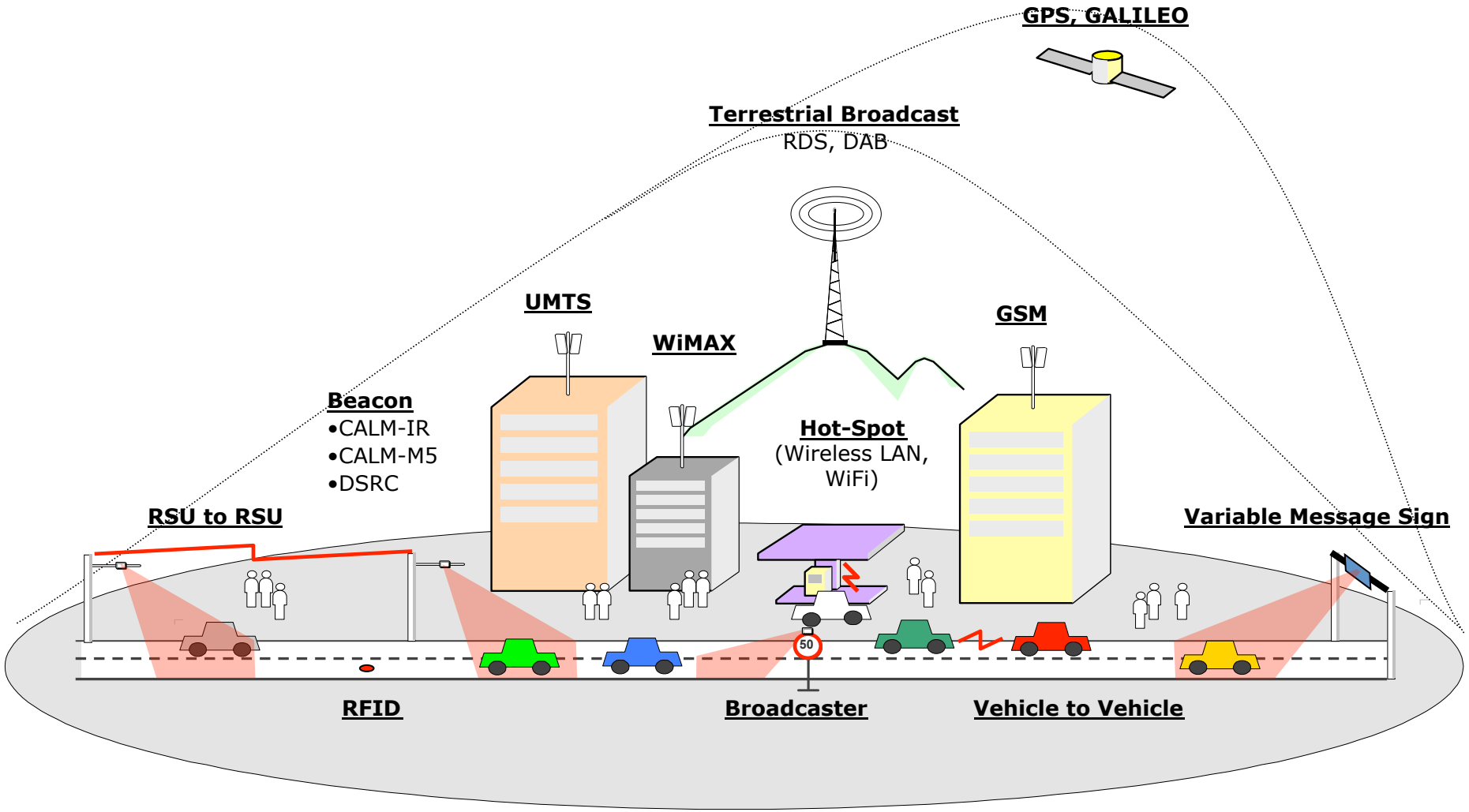
8.1 Important privacy related notions and metrics

8.2 Privacy in RFID systems

8.3 Location privacy in vehicular networks

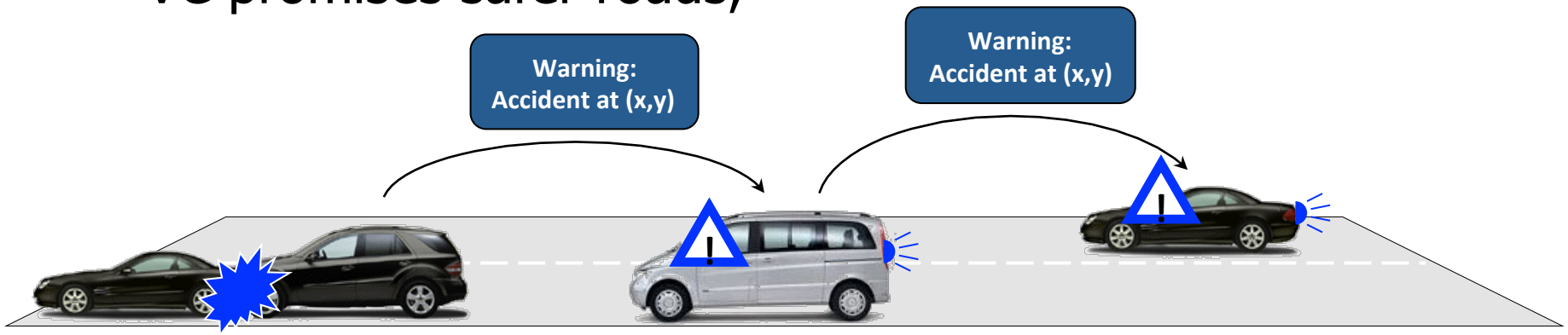
8.4 Privacy preserving routing in ad hoc networks

Vehicular Networks

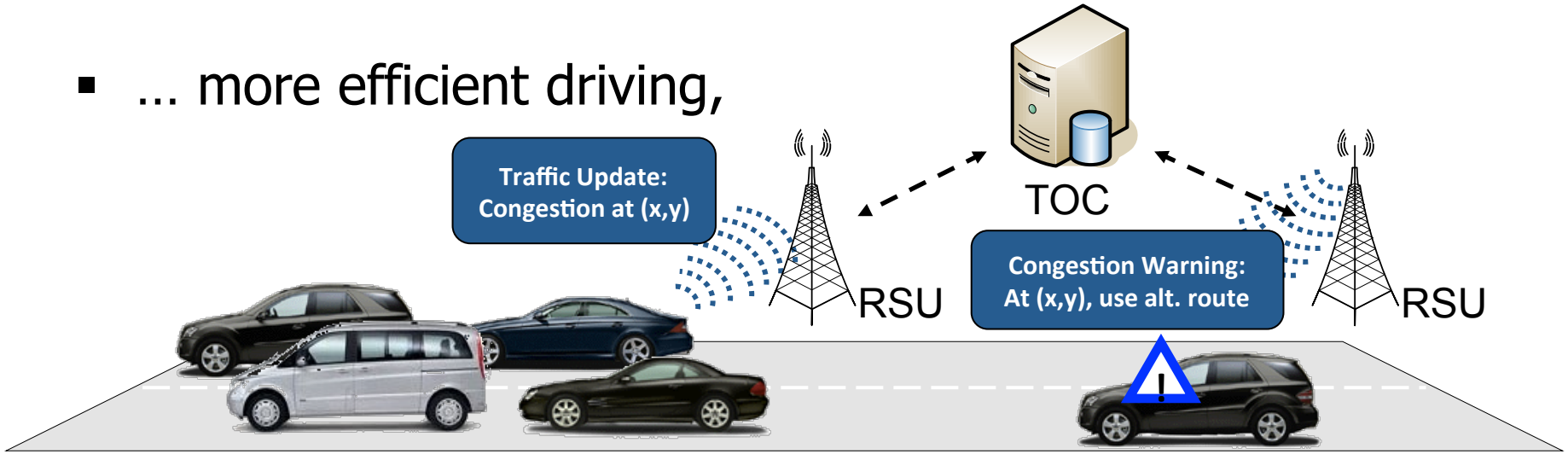


Vehicle Communication (VC)

- VC promises safer roads,

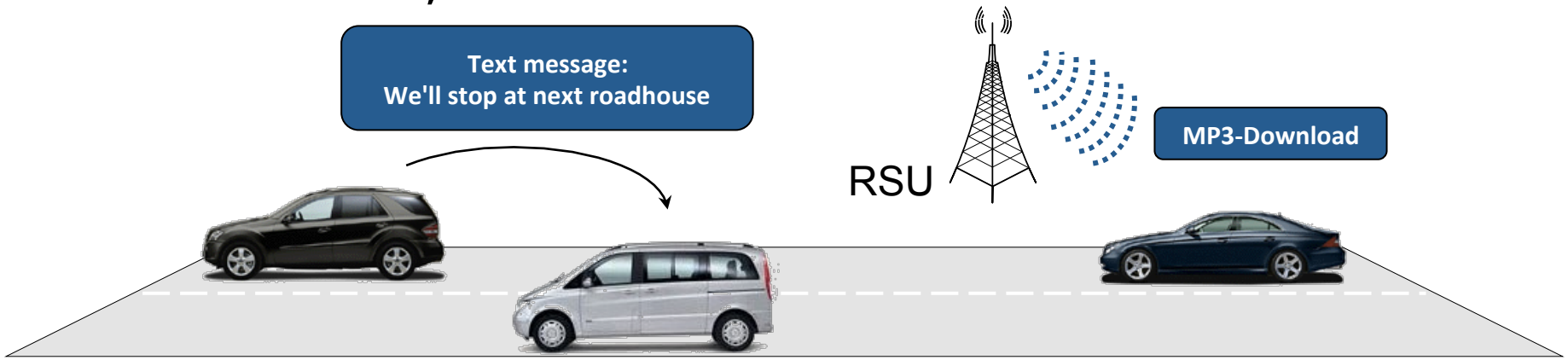


- … more efficient driving,

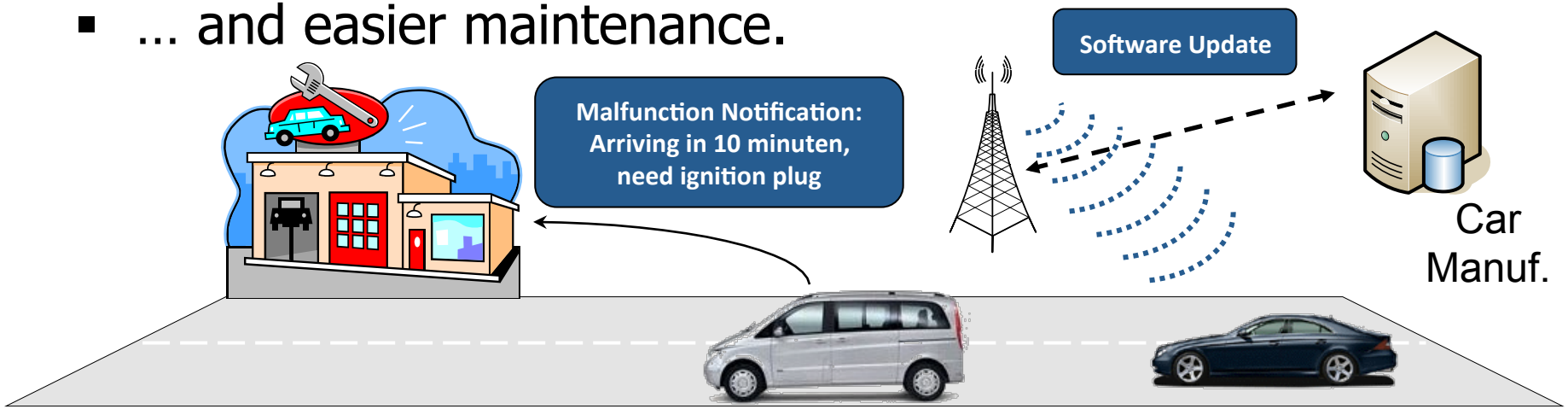


Vehicle Communication (VC)

- … more fun,

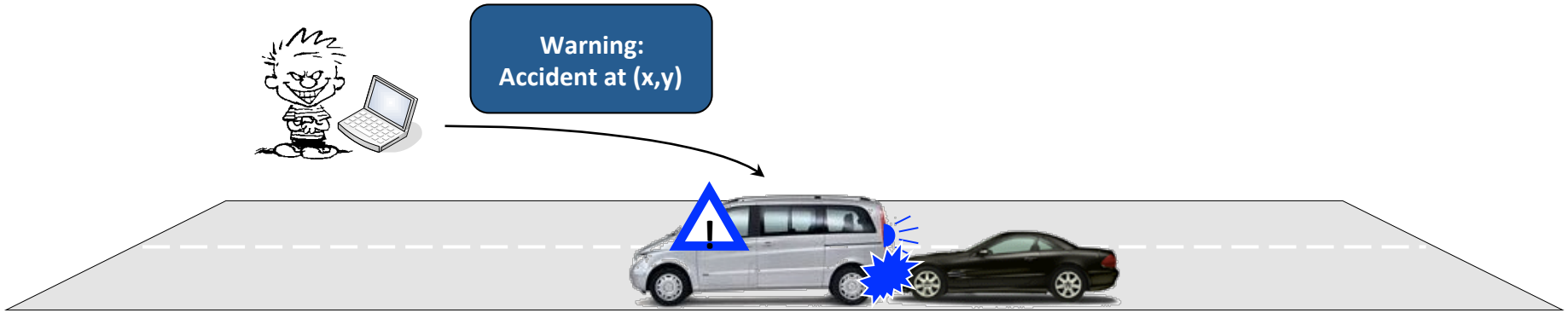


- … and easier maintenance.

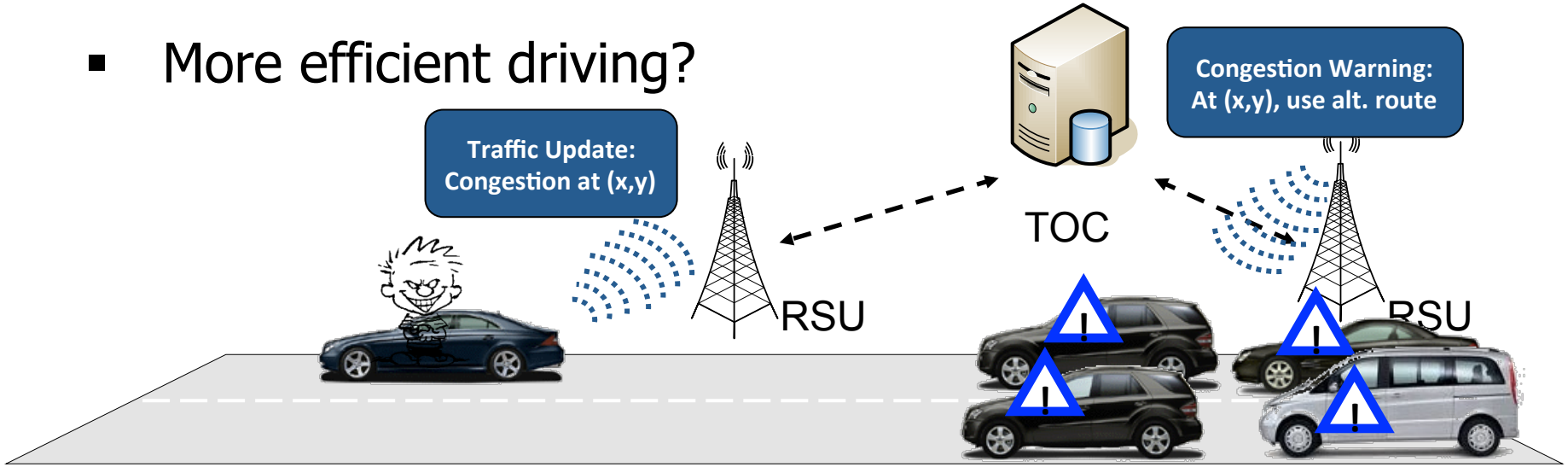


Security and Privacy???

- Safer roads?

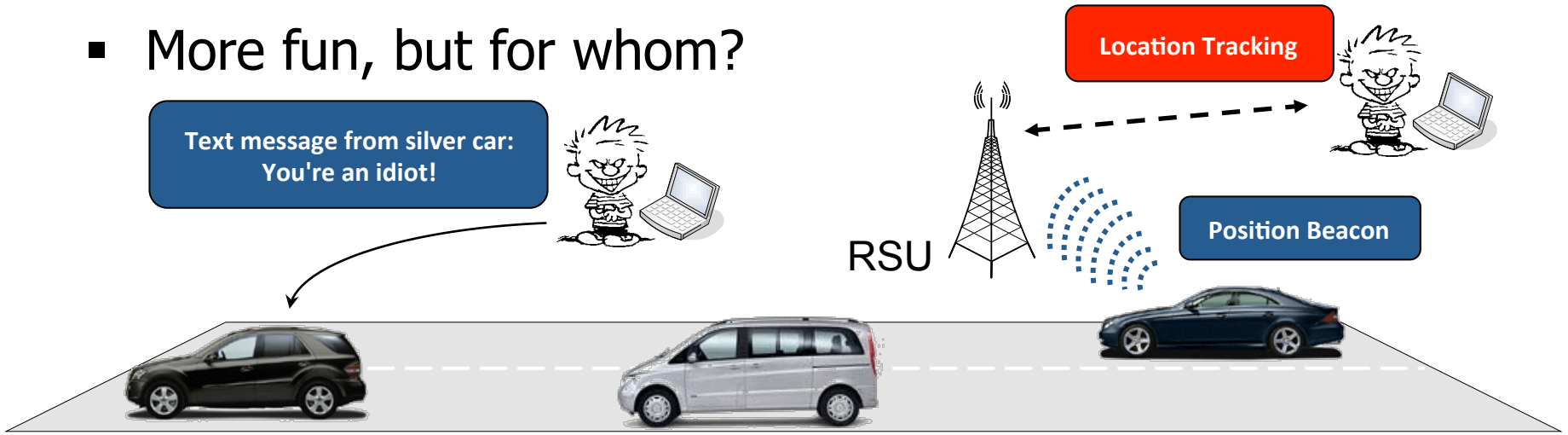


- More efficient driving?

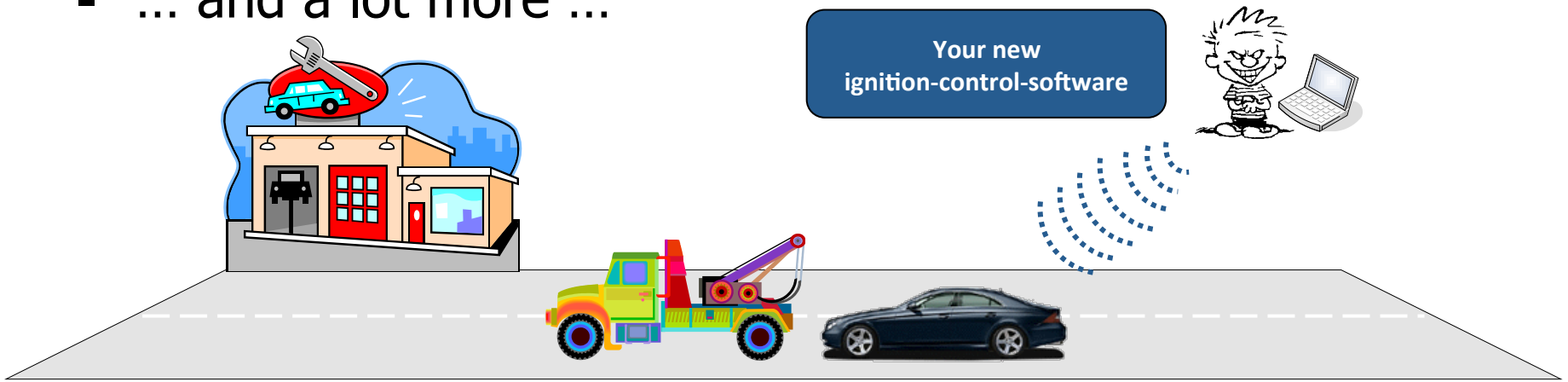


Security and Privacy???

- More fun, but for whom?



- … and a lot more …

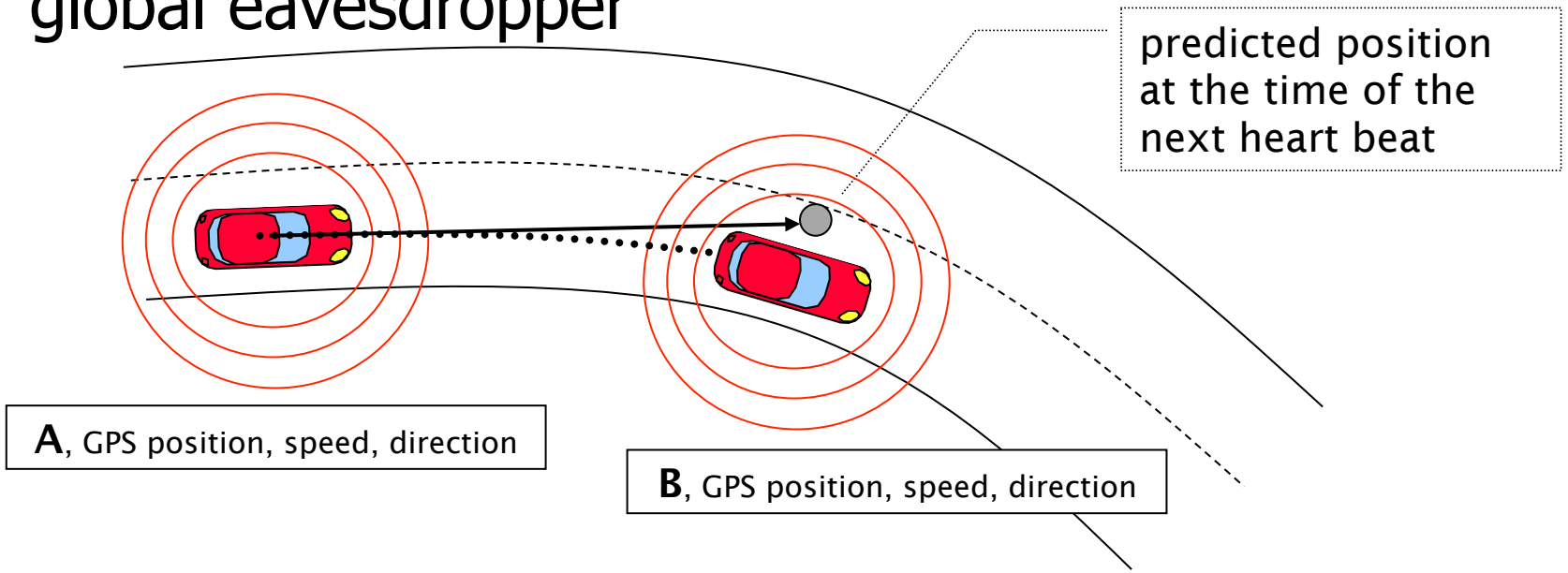


The location privacy problem and a solution

- 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*

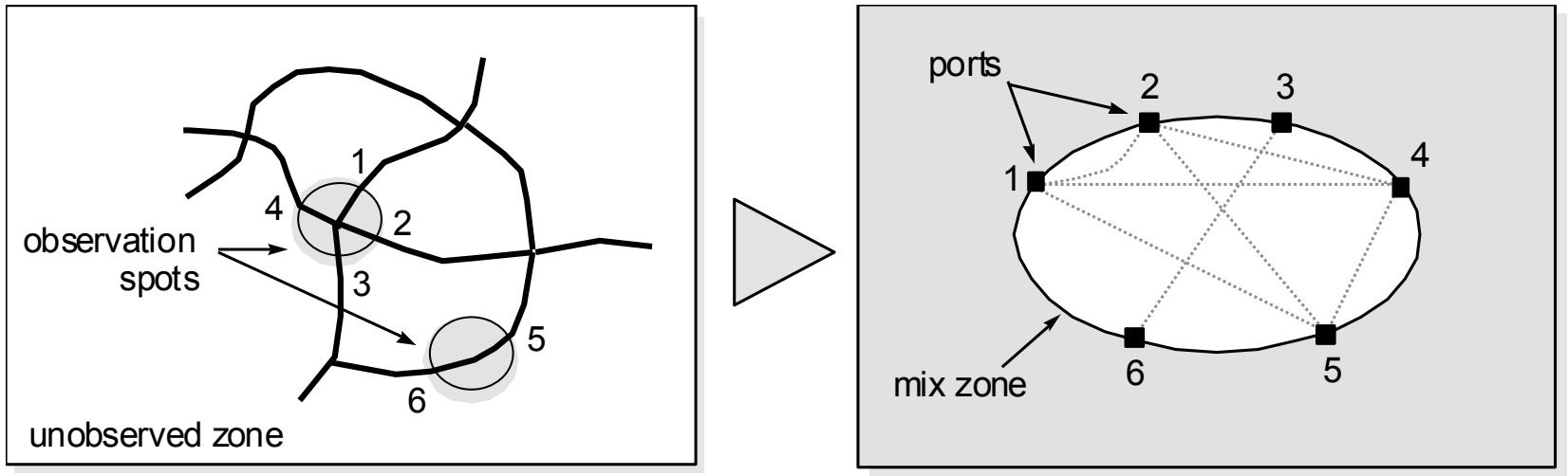
Adversary Model

- Changing pseudonyms is ineffective against a global eavesdropper



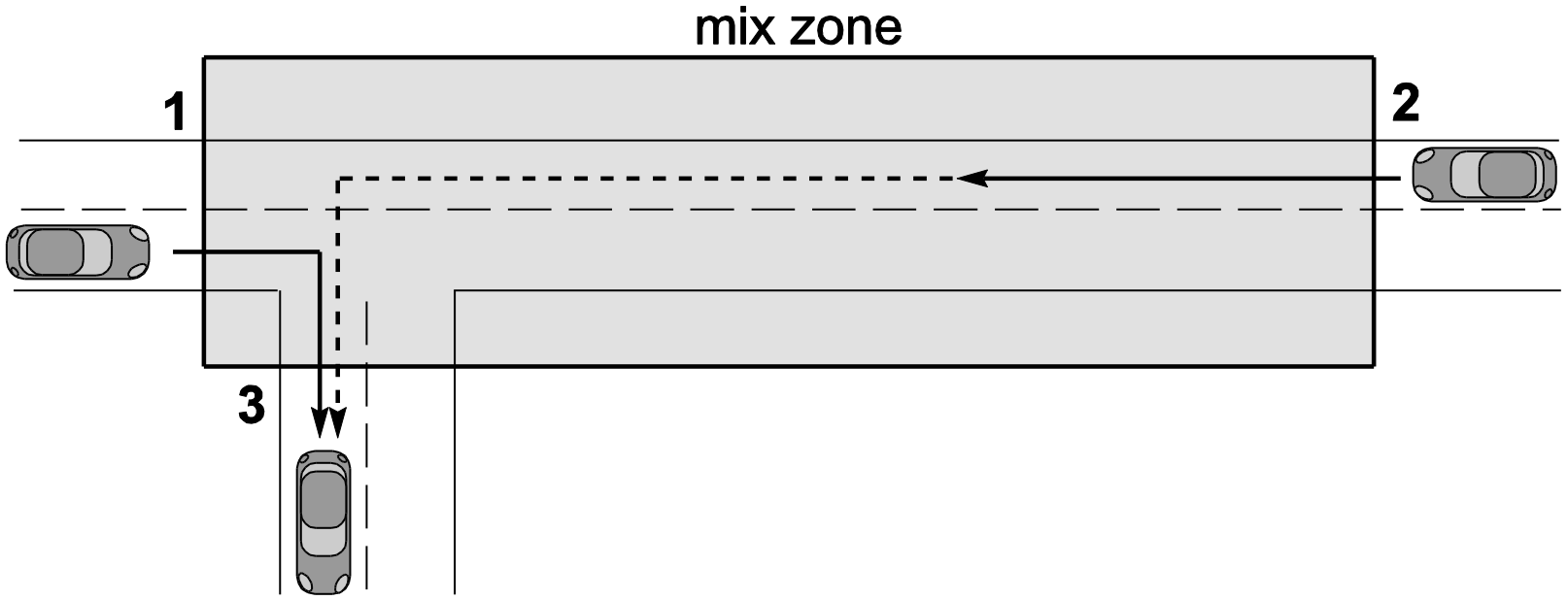
- Hence, the adversary is assumed to be able to monitor the communications only at a limited number of places and in a limited range

The mix zone concept



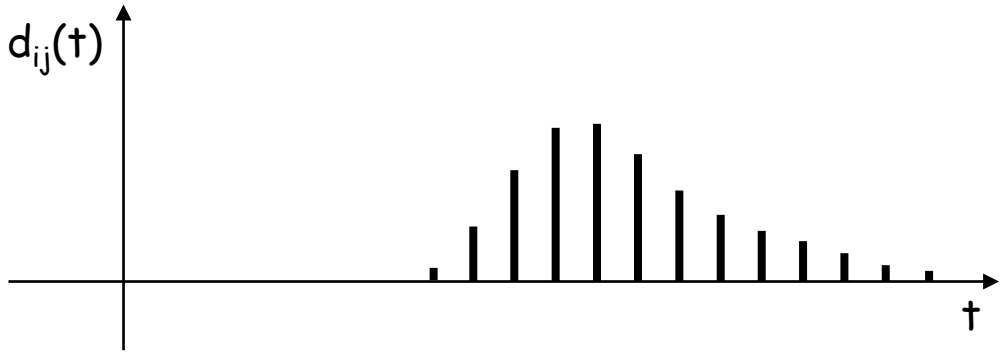
- The unobserved zone functions as a *mix zone* where the vehicles change pseudonym and mix with each other
- Note that the vehicles may not know where the mix zone is (this depends on where the adversary installs observation spots)
- We can assume that the vehicles change pseudonyms frequently so that each vehicle changes pseudonym while in the mix zone

Example of mix zone



Model of the mix zone

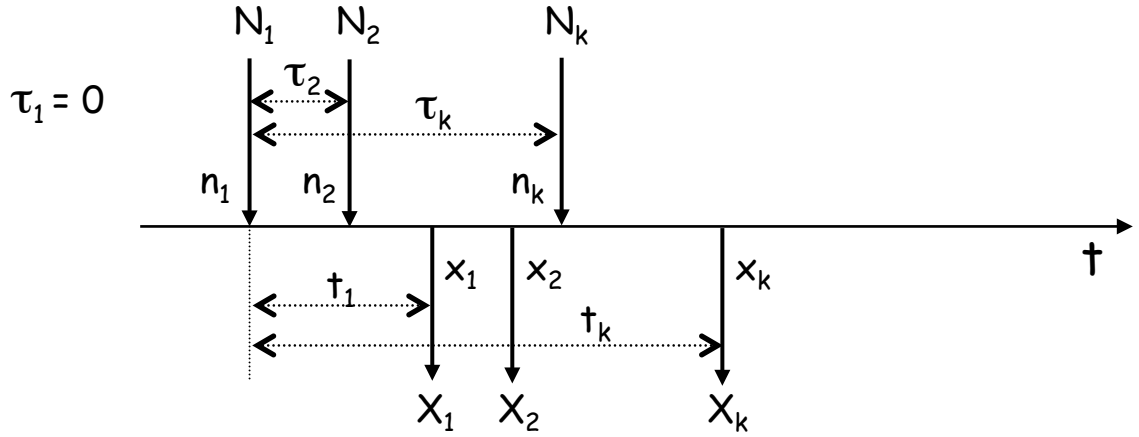
- time is divided into discrete steps
- p_{ij} = Pr{ exiting at j | entering at i }
- D_{ij} is a random variable (delay) that represents the time that elapses between entering at i and exiting at j
- $d_{ij}(t)$ = Pr{ D_{ij} = t }



- 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)



- By assumption, the nodes change pseudonyms inside the mix zone → there's 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 $Pr\{ m_\pi \mid \underline{N}, \underline{X} \}$

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

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

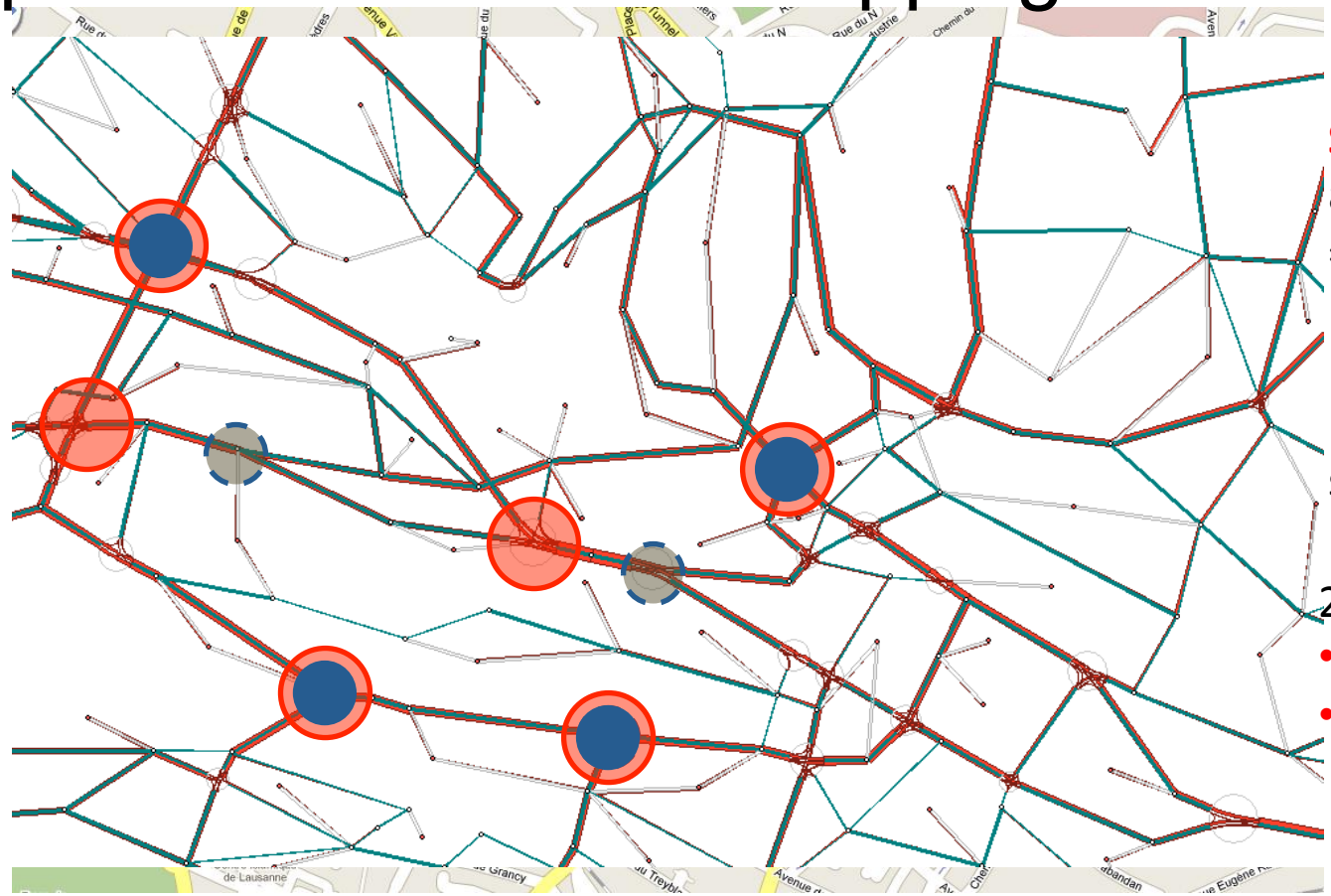
where p_{ij} is a cell of the matrix P of size nxn, where n is the number of gates of the mix zone and $d_{ij}(t)$ describes the probability distribution of the delay when crossing the mix zone from gate i to gate j.

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

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

Tracking Games

Placement of active/passive mix zones versus placement of eavesdropping stations



Strategic behaviors of attacker and defenders => **game theory** to model the interactions between players and predict their best strategies

2 knowledge levels
- complete information
- incomplete information

: Eavesdropping station (E) : Active mix zone (M) : Passive mix zone (P)

Chapter outline

8.1 Important privacy related notions and metrics

8.2 Privacy in RFID systems

8.3 Location privacy in vehicular networks

8.4 Privacy preserving routing in ad hoc networks

8.4 Privacy preserving routing in ad hoc networks

- **Goal: unlinkability** (make it very hard for a global observer to know who communicates with whom)

- Some nodes may be compromised ➔ even the forwarding nodes should not know who the source and the destination are

- We also want to hide the identity of the forwarding nodes from each other (because this information would be useful for the attacker)

Effective but inefficient solution

- **Route establishment:** flooding the network with a route request

- Source:
  - generates an asymmetric key-pair (K,K^{-1}), a secret key k_0, and a nonce n_0
  - Encrypts D, S, and K^{-1} with the public key K_D of the destination $E_{K_D}(D||S||K^{-1}) \;||\; K \;||\; E_K(k_0||n_0)$
  - Encrypts k_0 and n_0 with K
  - Broadcasts the route request:

Effective but inefficient solution

- F1 receives this route request $E_{K_D}(D||S||K^{-1})$
- It verifies if it is the target of the request:
  - decrypts $E_K(k_0||n_0)$ with its K^{-1}

- If F1 is not the target:
  - Generates a secret key k_1 and a nonce n_1
  - Concatenates them to
  - Encrypts the result with K
  - Broadcasts

$$E_{K_D}(D||S||K^{-1}) \;||\; K \;||\; E_K(k_1||n_1||E_K(k_0||n_0))$$

- General format of the route request message:

$$E_{K_D}(D||S||K^{-1}) \;||\; K \;||\; E_K(k_i||n_i||\ldots E_K(k_0||n_0)\ldots)$$

Effective but inefficient solution

- D attempts to decrypt $E_{K_D}(D||S||K^{-1})$ and it succeeds
- D broadcasts a dummy request:

$$E_{K_D}(D||S||K^{-1}) \;||\; K \;||\; garbage$$

- It decrypts $E_K(k_\ell||n_\ell||\ldots E_K(k_0||n_0)\ldots)$ and obtains the secret keys and the nonces of the forwarding nodes

- It generates a link key for each link and sends a route reply:

$$E_{k_\ell}(n_\ell||k_\ell^{in}||k_\ell^{out} \;||\; E_{k_{\ell-1}}(n_{\ell-1}||k_{\ell-1}^{in}||k_{\ell-1}^{out} \;||\; \ldots E_{k_0}(n_0||-||k_0^{out})\ldots))$$

Effective but inefficient solution

- F_i receives route reply: decrypts it with k_i
- If k_i works: checks if it received back its n_i
- If this is the case:
  - F_i peels the outer layer off the route reply
  - Applies some padding to retain its original length
  - Re-broadcasts

- Sending data:
  - Source encrypts the packet with k^{out}_0 and broadcasts it
  - Each node tries to decrypt it with its incoming link keys
  - If F_i succeeds to decrypt the packet with k_i^{in}: it re-encrypts it with k_i^{out}, and re-broadcasts it
  - Until the packet arrives to the destination

Improving efficiency

- Much computation from the nodes:
  - Solution: replace the public key encryption with symmetric key encryption

- Source and destination share a secret key k_{SD} and a counter c_{SD}
- Source computes a one-time hint for the destination: $h(k_{SD},c_{SD})$
- Each node can pre-compute the hint of each possible source:
  - only a table lookup when processing route request messages

Improving efficiency

- Modified route request:

$$h(k_{SD}, c_{SD}) \;||\; E_{k_{SD}}(D||S||K^{-1}) \;||\; K \;||\; E_K(k_i||n_i|| \ldots E_K(k_0||n_0) \ldots)$$

- Modified route reply:

$$g(n_\ell) \;||\; E_{k_\ell}(n_\ell||k_\ell^{in}||k_\ell^{out}||g(n_{\ell-1}) \;||\; E_{k_{\ell-1}}(n_{\ell-1}||k_{\ell-1}^{in}||k_{\ell-1}^{out}|| \ldots g(n_0) \;||\; E_{k_0}(n_0|| - ||k_0^{out}) \ldots))$$

- Hint for F_i: hashing n_i with g
- When processing route reply:
  - Only a table lookup to determine which key should be used to decrypt the route reply

Summary

- Privacy problems and solutions in RFID:
  - Privacy problems: clandestine reading and eavesdropping
  - Low-cost RFID tags: resource constrained, any privacy protecting solution must be carefully designed and optimized
- Location privacy in vehicular networks:
  - Adversary model: monitored zones and unmonitored zones
  - The level of location privacy can be quantified using an entropy based metric
- Privacy in ad hoc network routing protocols:
  - A routing protocol that make it very hard for a global observer to know who communicates with whom