Security and Privacy in Upcoming Wireless Networks

Efficient Symmetric-key Private Authentication

Private authentication – the problem

- authentication protocols often reveal the identity of the authenticating party (prover) to an eavesdropper
- when devices move around and authenticate themselves frequently, the location of them can be tracked
- typical examples are RFID tags and contactless smart card based systems

An example – ISO 9798-2

- the protocol:

  (1) B → A: r_B
  (2) A → B: E(K, r_B | B*)

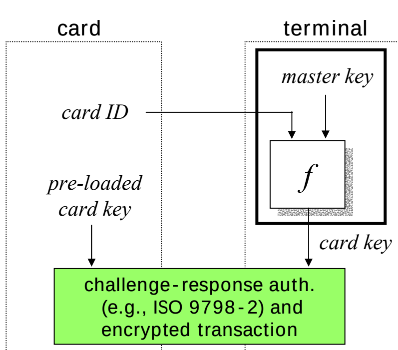
  where K is a shared key between A and B, and E(.) denotes encryption

- "it is assumed that the parties are aware of the claimed identity of the other either by context or *by additional cleartext data fields*"

  (0) A → B : A

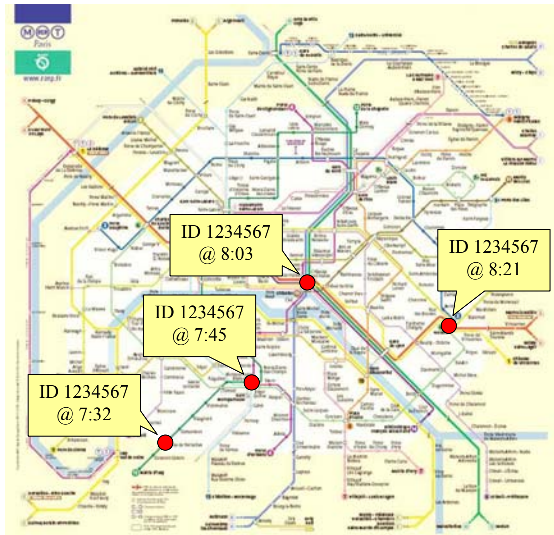
Authentication in AFC systems

- off-line terminals should be able to authenticate any transport card



- key diversification is used
  - each card has its own key
  - card key is generated from the card ID and a master key using a one-way function
  - terminals store only a few master keys, and compute card keys on-the-fly when they are needed
- *this requires transmitting the ID of the card at the beginning of the transaction*

Private authentication – the problem (cont'd)

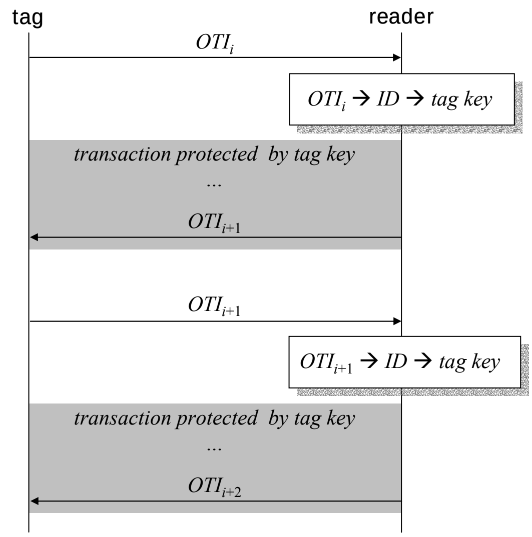


Solutions based on public-key cryptography

- encrypt identity information of the authenticating party with the public key of the verifier
- setup a confidential channel between the parties using the basic Diffie-Hellman protocol and send identity information through that channel
  - IKE in main mode works in this way
- <u>common disadvantage</u>: public key operations may not be affordable in devices with limited resources (e.g., public transport cards, RFID tags)

One-time identifiers – a solution for high-end tags

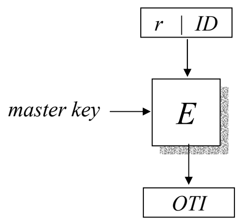
[Buttyan et al., 2006]



Assumptions and requirements for OTIs

- tags must have some writable memory
- OTIs should be unlinkable
- mapping an OTI to the real ID should be fast
- OTI generated by one reader should be decodable by another
  - since readers may be off-line, OTIs must be generated from a master key known by every reader
- it should be difficult to break the master key even if many OTIs are observed

Possible implementations

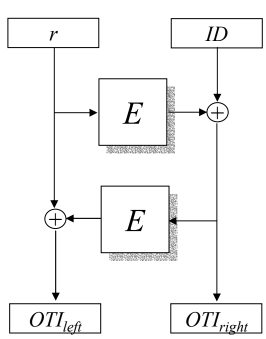


- let's use the block cipher already implemented on many readers (e.g., DES)
- if IDs are not too long, then a random number and an ID may fit in a single block
- if r is n bit long, then first repeating OTI is expected after about $2^{n/2}$ transactions (birthday paradox)
- typical ID lengths
  - Mifare Classic: 32 bits

    $n = 32$, $2^{32/2} = \sim 60000$
  - Mifare DESfire: 56 bits

    $n = 8$, $2^{8/2} = 16$ (too small!)

Possible implementations (cont'd)



- if IDs are long, then we need two blocks
- one could use the block cipher in a 2-round Feistel structure to effectively double its block length
- $n = 64$ should be enough

Naïve solutions for low-cost tags

- encrypt (hash) identity information with a single common key
  drawback:
  – compromise of a single member of the system has fatal consequences
- encrypt (hash) identity information with a unique key
  drawback:
  – number of keys need to be tested by the verifier grows linearly with the number of potential provers
  → doesn't scale (potentially long authentication delay in large systems)

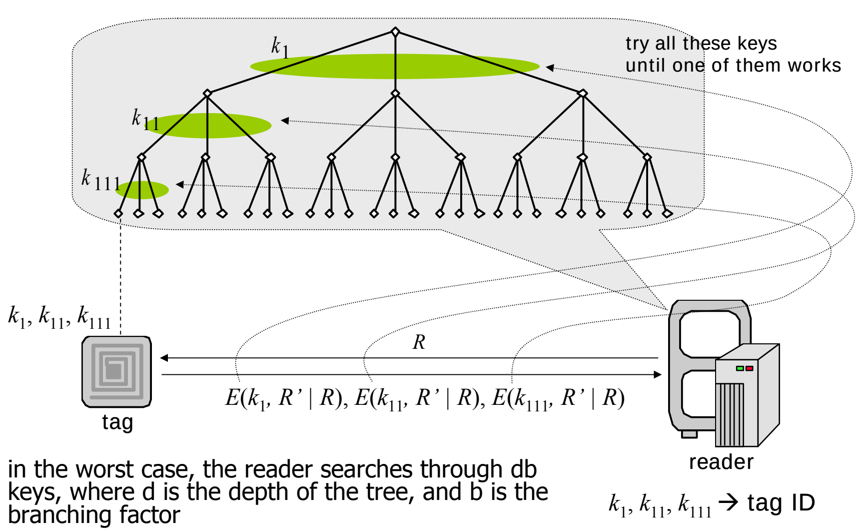
Better solutions for low cost tags

- tree-based approach
  – proposed by Molnar and Wagner in 2004
  – improved by Buttyan, Holczer, and Vajda in 2006
  advantage:
  – authentication delay is logarithmic in the number of members
  drawback:
  – increased overhead (at the prover's side)
  – level of privacy quickly decreasing as the number of compromised members increases
- group-based approach
  – proposed by Avoine, Buttyan, Holczer, Vajda in 2007
  advantage:
  – higher level of privacy and smaller overhead than in the tree-based approach
  drawback: ???

Outline

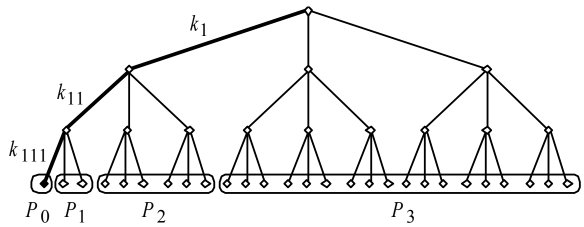
- introduction to private authentication
- overview of the tree-based approach
  - original proposal
  - optimization
- privacy metrics
  - normalized average anonymity set size [Buttyan et al., 2006]
  - probability of traceability [Avoine et al., 2005]
  - entropy based anonymity set size [Nohl and Evens, 2006]
- description and analysis of the group-based approach
- comparison of the tree-based and the group-based approaches
- conclusion and open problems

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
- compare this to b^d, which is the total number of tags !

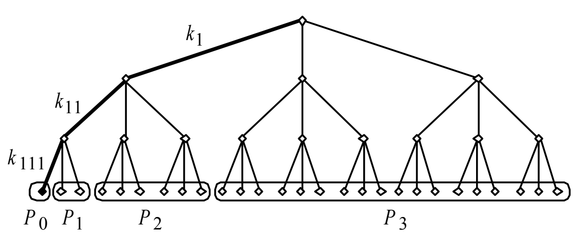
Tree-based private authentication – a problem



- if a member is compromised, its keys are learned by the adversary
- however, most of those keys are used by other members too
- the adversary can recognize the usage of those compromised keys
- consequently, the level of privacy provided by the system to *non-compromised* members is decreased

but:
this decrease can be minimized by careful design of the tree!

Resistance to single member compromise (RSMC)



- the level of privacy provided by the system to a randomly selected member is characterized by the *average anonymity set size when a single member is compromised*:

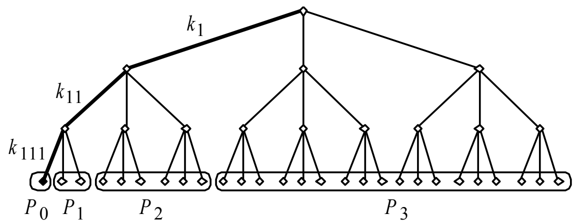
$$\bar{S} = \sum_{i=0}^{\ell} \frac{|P_i|}{N} |P_i| = \sum_{i=0}^{\ell} \frac{|P_i|^2}{N}$$

where N is the total number of members

- we normalize this to obtain the benchmark metric called *resistance to single member compromise*:

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

Computing RSMC



$|P_0| = 1$
$|P_1| = b_\ell - 1$
$|P_2| = (b_{\ell-1} - 1)b_\ell$
$|P_3| = (b_{\ell-2} - 1)b_{\ell-1}b_\ell$
$\ldots \quad \ldots$
$|P_\ell| = (b_1 - 1)b_2 b_3 \ldots b_\ell$

$$R = \frac{\bar{S}}{N} = \sum_{i=0}^{\ell} \frac{|P_i|^2}{N^2}$$
$$= \frac{1}{N^2}\left(1 + (b_\ell - 1)^2 + ((b_{\ell-1} - 1)b_\ell)^2 + \ldots + ((b_1 - 1)b_2 b_3 \ldots b_\ell)^2\right)$$
$$= \frac{1}{N^2}\left(1 + (b_\ell - 1)^2 + \sum_{i=1}^{\ell-1}(b_i - 1)^2 \prod_{j=i+1}^{\ell} b_j^2\right)$$

A trade-off between privacy and efficiency

- efficiency of the system is characterized by the *maximum authentication delay*:

$$D = \sum_{i=1}^{\ell} b_i$$

- examples:
  - naïve linear key search (l = 1)
    - R = 1 – 2(N+1)/N² ≈ 1 – 2/N ≈ 1 (if N is large)
    - D = N
  - binary key-tree (l = log N)
    - R = 1/3 + 2/(2N²) ≈ 1/3 (if N is large)
    - D = 2 log N
- how to maximize R while keeping D below a threshold?

The optimization problem

Given the total number N of members and the upper bound D_{max} on the maximum authentication delay, find a branching factor vector B = (b_1, b_2, . . . b_l) such that

$$R = \frac{1}{N^2}\left(1 + (b_\ell - 1)^2 + \sum_{i=1}^{\ell-1}(b_i - 1)^2 \prod_{j=i+1}^{\ell} b_j^2\right)$$

is maximal, subject to the following constraints:

$$\prod_{i=1}^{\ell} b_i = N$$

$$\sum_{i=1}^{\ell} b_i \leq D_{max}$$

Analysis of the optimization problem

Lemma 1: we can always improve a branching factor vector by ordering its elements in decreasing order

Lemma 2: lower and upper bounds on R(B) (where B is ordered):

$$\left(1 - \frac{1}{b_1}\right)^2 \leq R(B) \leq \left(1 - \frac{1}{b_1}\right)^2 + \frac{4}{3b_1^2}$$

Lemma 3: given two branching factor vectors (that satisfy the constraints), the one with the larger first element is always at least as good as the other

Lemma 4: given two branching factor vectors the first j elements of which are equal, the vector with the larger (j+1)-st element is always at least as good as the other

A solution

- let P be the ordered vector of prime factors of N
- if P doesn't satisfy the conditions, then no solution exists
- otherwise, let P' be a subset of P such that
  - if we multiply the prime factors in P' (let the product be Q), then the vector (Q, P\P') still satisfy the constraints, and
  - Q is maximal
- the first element of the optimal branching factor vector is Q
- if all prime factors are used (P\P' = ∅), then stop
- else repeat the procedure recursively with the remaining primes

An example for the operation of the algorithm

- let N = 27000 and D_{max} = 90

recursion level	P	d	P'	Q
1	$(5,5,5,3,3,3,2,2,2)$	90	$(3,3,2,2,2)$	72
2	$(5,5,5,3)$	18	(5)	5
3	$(5,5,3)$	13	(5)	5
4	$(5,3)$	8	(5)	5
5	(3)	3	(3)	3

- the optimal tree for these parameters is (72, 5, 5, 5, 3)
  - R ≈ 0.9725
  - D = 90

Proof sketch of the algorithm

- let $B^* = (b^*_1, \ldots, b^*_L)$ be the output of the algorithm
- assume that there's a $B' = (b'_1, \ldots, b'_K) \neq B^*$ such that $R(B') > R(B^*)$

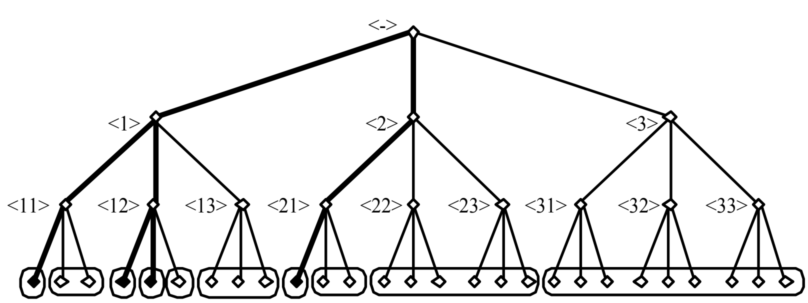
- B^* is obtained by maximizing $b^*_1 \rightarrow b^*_1 \geq b'_1$
- if $b^*_1 > b'_1$ then $R(B^*) \geq R(B')$ by Lemma 3 $\rightarrow b^*_1 = b'_1$ must hold

- B^* is obtained by maximizing b^*_2 (once b^*_1 is determined) $\rightarrow b^*_2 \geq b'_2$
- if $b^*_2 > b'_2$ then $R(B^*) \geq R(B')$ by Lemma 4 $\rightarrow b^*_2 = b'_2$ must hold

…

- $B^* = B'$ must hold, which is a contradiction

General case



- compromised tags partition the set of all tags
  - tags in a given partition are indistinguishable
  - tags in different partitions can be distinguished
- we measure the level of privacy again as the normalized average anonymity set size (NAASS)

Approximation of NAASS for key-trees

- let the branching factors of the tree be $b_1, b_2, \ldots, b_L$
- select a tag *T* randomly (without loss of generality, we assume that the left most tag of the tree is selected)
- we want to compute the expected size of *T*'s anonymity set when *some* tags are compromised
- we assume that each tag is compromised with probability

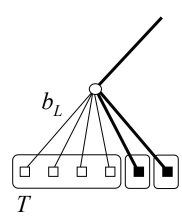
$$p = C/N$$

- the probability that a given edge (key) is compromised at level i is

$$q_i = 1 - (1 - p)^{N_i}$$

where $N_i = N/(b_1 b_2 \ldots b_i)$ is the number of tags below that edge

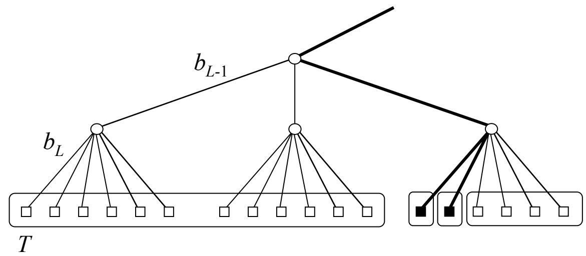
Approximation of NAASS for key-trees



- the probability that *T*'s anonymity set size is exactly *k* ($k = 1, 2, \ldots, b_L-1$) is:

$$(1-q_L)\binom{b_L-1}{k-1}(1-q_L)^{k-1}q_L^{b_L-k} = \binom{b_L-1}{k-1}(1-q_L)^k q_L^{b_L-k}$$

Approximation of NAASS for key-trees



- the probability that T's anonymity set size is kb_L (k = 1, 2, …, b_{L-1}-1) is:

$$\binom{b_{L-1}-1}{k-1}(1-q_{L-1})^k q_{L-1}^{b_{L-1}-k}$$

Approximation of NAASS for key-trees

- in general, the probability that T's anonymity set size is $kb_Lb_{L-1}\ldots b_{i+1} = kN_i$ (i = 1, 2, …, L and k = 1, 2, …, b_i-1) is:

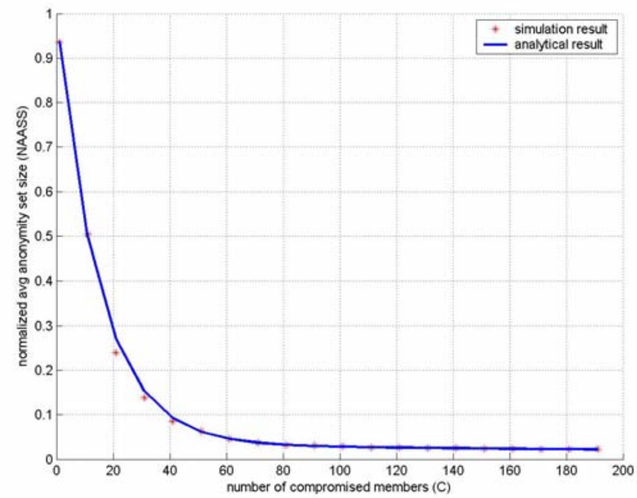
$$\binom{b_i-1}{k-1}(1-q_i)^k q_i^{b_i-k}$$

- from this, the expected size of T's anonymity set is:

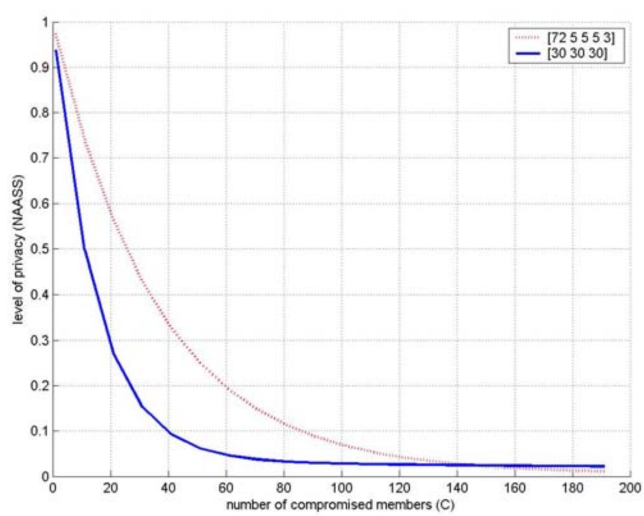
$$\bar{S} = \sum_{i=1}^{L}\sum_{k=1}^{b_i-1} kN_i \binom{b_i-1}{k-1}(1-q_i)^k q_i^{b_i-k} + 1\cdot p + N\cdot(1-p)^N$$

Verification of the approximation

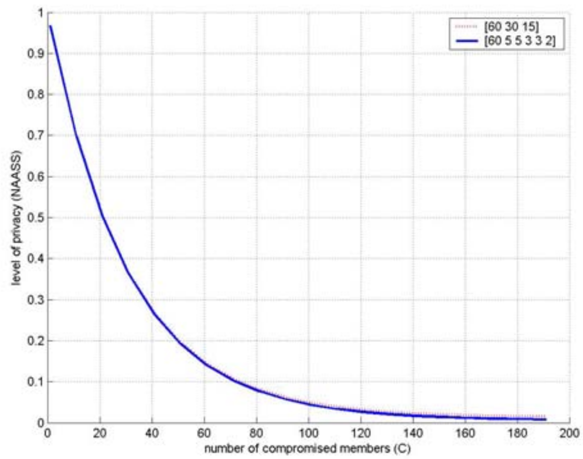
B = [30 30 30]



Comparison of key-trees in the general case



Comparison of key-trees in the general case



conclusion:

first element of the branching factor vector determines the level of privacy in the general case too

Probability of traceability as a privacy metric

- proposed by Avoine et al. at SAC 2005
- traceability game:
  1. The adversary can tamper with a certain number C of tags (Compromised tags are put back in circulation)
  2. The adversary chooses a tag T and queries it as much as she wants (but she cannot compromise T)
  3. The adversary is presented two tags T_1 and T_2 such that T is in $\{T_1, T_2\}$. The adversary can query both T_1 and T_2 as much as she wants, and she has to decide which one is T.
- the success probability of the adversary in the traceability game is a measure of privacy

Relation to NAASS

- if T_1 and T_2 are in the same partition, then the adversary cannot distinguish them → she cannot tell which one is T
- otherwise, she can distinguish T_1 and T_2 → she can decide which one is T
- prob. of success = 1 – Pr{T_1 and T_2 are in the same partition}

$$\sum_{\forall |P|} \Pr\{\text{size of } T\text{'s partition is } |P|\} \cdot \frac{|P|}{N} =$$

$$\sum_{i=1}^{L} \sum_{k=1}^{b_i-1} \binom{b_i-1}{k-1} (1-q_i)^k q_i^{b_i-k} \cdot \frac{kN_i}{N} + p \cdot \frac{1}{N} + (1-p)^N \cdot \frac{N}{N} = \frac{\bar{S}}{N}$$

Normalized entropy based anonymity set size (NEASS)

- proposed by Nohl and Evans in 2006 (tech report)
- main idea:
  - assume that a tag is compromised and this results in two equal size (N/2) partitions
  - the adversary can tell each tag in either one of the partitions → 1 bit of information has been disclosed
  - in general, the amount of information that is disclosed due to tag compromise is

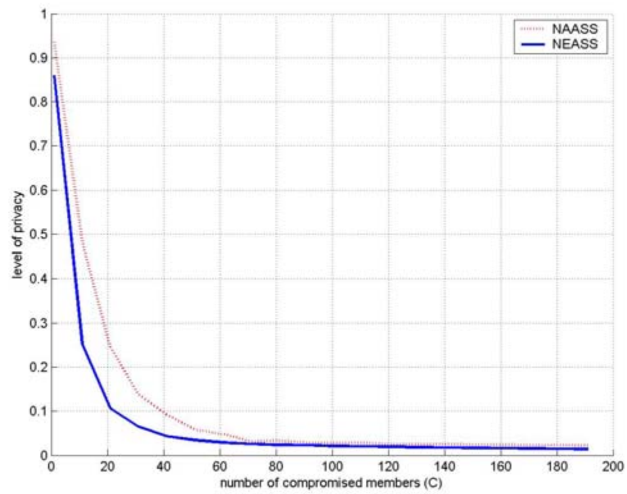
$$I = \sum_{\forall P} \frac{|P|}{N} \log_2 \frac{N}{|P|}$$

- (normalized) entropy based anonymity set size:

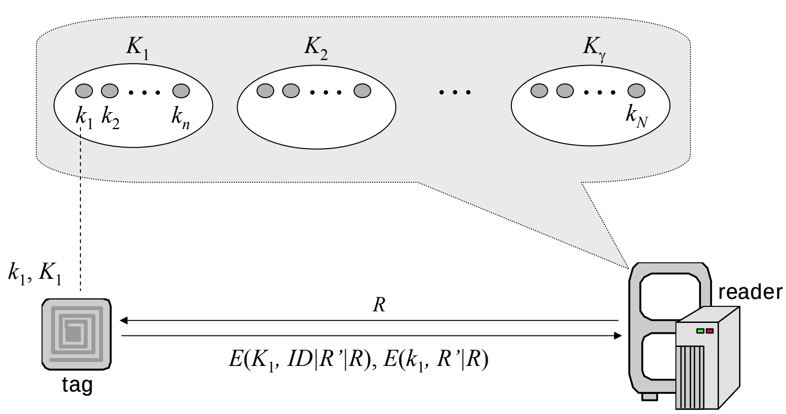
$$\bar{S}_{entropy} = \frac{N}{2^I} \qquad \frac{\bar{S}_{entropy}}{N} = \frac{1}{2^I}$$

Comparison of NAASS and NEASS (simulation)

B = [30 30 30]



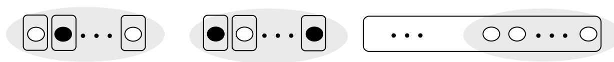
The group-based approach



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]

Computing NAASS for the group-based approach

- let A_i be the event that at least one tag is compromised from the i-th group
- let I_{Ai} be the indicator function of A_i
- the probability of A_i can be computed as:

$$P(A_i) = 1 - \frac{\binom{N-n}{c}}{\binom{N}{c}} = 1 - \prod_{j=0}^{c-1}\left(1 - \frac{n}{N-j}\right)$$

Computing NAASS for the group-based approach

$$E[C] = E\left[\sum_{i=1}^{\gamma} I_{A_i}\right] = \sum_{i=1}^{\gamma} P(A_i) = \gamma\left(1 - \prod_{j=0}^{c-1}\left(1 - \frac{n}{N-j}\right)\right)$$

$$E\left[C^2\right] = E\left[\sum_{i=1}^{\gamma} I_{A_i}\right]^2 = E\left[\sum_{i=1}^{\gamma} I_{A_i}\right] + E\left[\sum_{i\neq j} I_{A_i\cap A_j}\right] =$$

$$= E[C] + \left(\gamma^2 - \gamma\right) P\left(A_i \cap A_j\right)$$

Computing NAASS for the group-based approach

$$P\left(A_i \cap A_j\right) = 1 - P\left(\overline{A_i} \cap \overline{A_j}\right) - 2P\left(A_i \cap \overline{A_j}\right)$$

$$P\left(\overline{A_i} \cap \overline{A_j}\right) = \frac{\binom{N-2n}{c}}{\binom{N}{c}} = \prod_{j=0}^{c-1}\left(1 - \frac{2n}{N-j}\right)$$

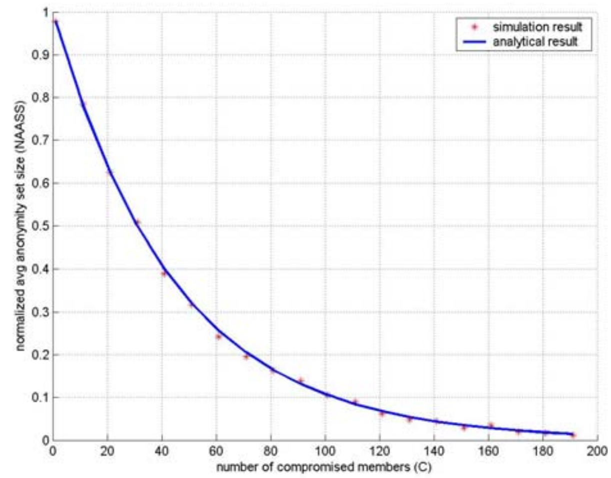
$$P\left(A_i \cap \overline{A_j}\right) = P\left(A_i | \overline{A_j}\right) P\left(\overline{A_j}\right) =$$

$$= \left[1 - \prod_{j=0}^{c-1}\left(1 - \frac{n}{N-n-j}\right)\right] \cdot \prod_{j=0}^{c-1}\left(1 - \frac{n}{N-j}\right)$$

Verification of the approximation

N = 27000

γ = 90



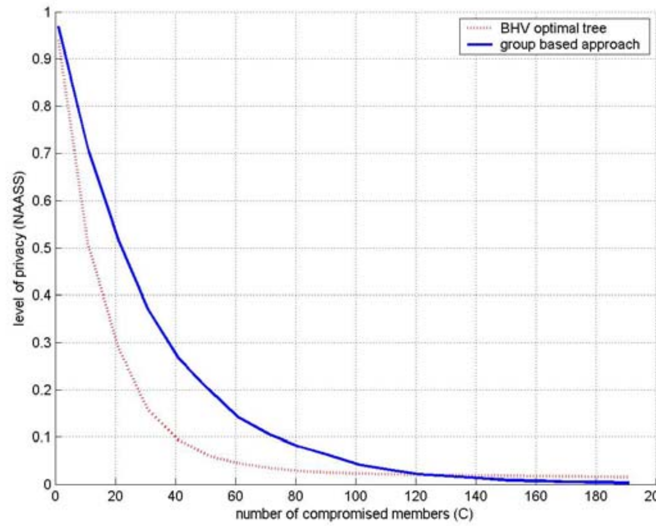
Comparison of trees and groups

- 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

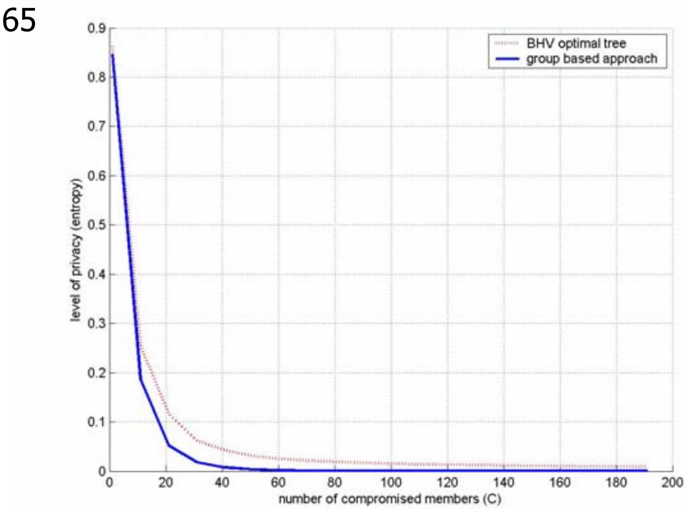


[32 16 8 4]
64 x 256

Comparison in NEASS

N = 2^{14}

D = 65



[32 16 8 4]
64 x 256

Summary

- we studied the problem of (efficient) symmetric-key private authentication
- we gave an overview of the tree-based and the group-based approaches
- we gave an overview of proposed privacy metrics
  - NAASS, NEASS, prob. of traceability
- we showed some relationships between the metrics
  - prob. of traceability ~ NAASS, NEASS < NAASS
- we gave precise approximations of the NAASS for trees and for groups
- we compared the tree and the group based approaches using NAASS and NEASS

Conclusions

- we obtained controversial results
  - group-based approach achieves better privacy if we use NAASS
  - tree-based approach achieves better privacy if we use NEASS

→ be cautious which metric you use!

- yet, the difference between trees and groups does not seem to be large in terms of privacy
- groups may be a better trade-off, due to the smaller overhead

→ the group-based approach could be a serious alternative to the tree-based approach

Open problems

1. Closed form approximation of the NEASS (both for trees and groups) ?
2. How to find the optimal tree when the metric is the NEASS ?
3. How to preserve the efficiency of the tree and the group-based approaches *and* eliminate the exponential decrease of the level of privacy at the same time ???

Security and Privacy in Upcoming Wireless Networks

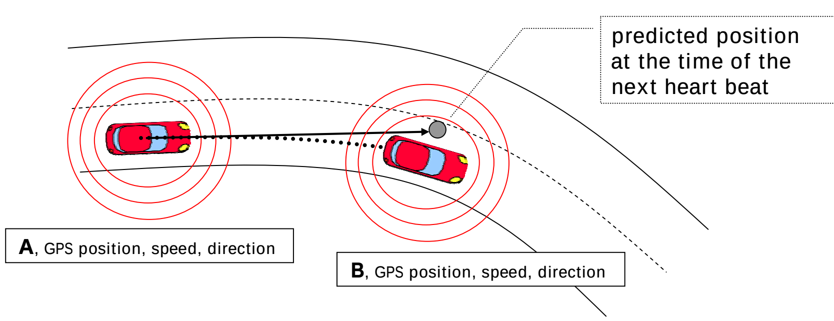
Location privacy in vehicular communication systems

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*

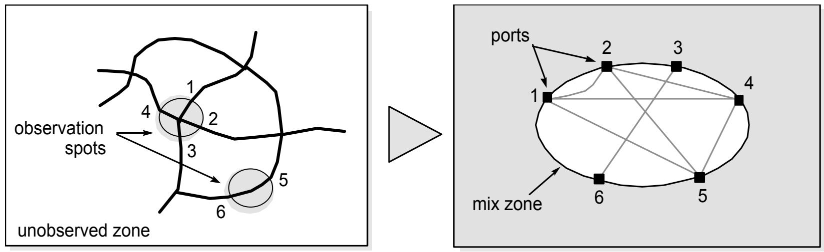
Adversary model

- 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

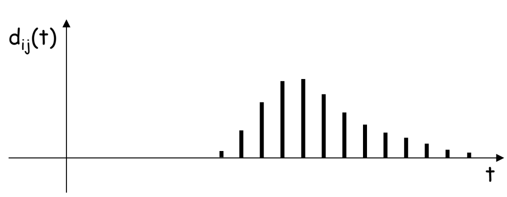
The mix zone concept



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

Model of the mix zone

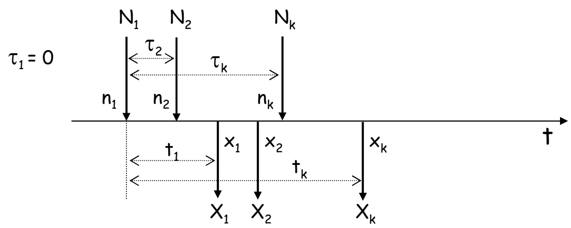
- time is divided into discrete steps
- p_{ij} = Pr{ exiting at j | entering at i }
- D_{ij} is a random variable (delay) that represents the time that elapses between entering at i and exiting at j
- $d_{ij}(t)$ = Pr{ D_{ij} = t }



- 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_π | $\underline{N}$, $\underline{X}$ }

Computing the level of privacy

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

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

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

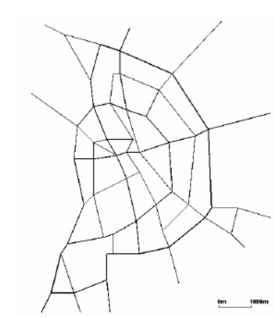
$$H(\bar{N}, \bar{X}) = -\sum_{\pi} \frac{q_\pi}{\sum_{\pi'} q_{\pi'}} \log\left(\frac{q_\pi}{\sum_{\pi'} q_{\pi'}}\right)$$

Another privacy metric

- tracking game:
  - the adversary picks a vehicle v in the observed zone
  - she tracks v until it enters the mix zone at port s
  - then, she observes the exiting events until time T (where the probability that v leaves the mix zone until T is close to one)
  - for each exiting vehicle at port j and time t, the adversary computes $q_{jt} = p_{sj} d_{sj}(t)$
  - the adversary decides to the exiting vehicle v' for which q_{jt} is maximal
    - this realizes a Bayesian decision (minimizes the error probability of the decision)
  - the adversary wins if v' = v
- the level of privacy achieved is characterized by the success probability of the adversary
  - if success probability is high, then level of privacy is low

Simulation settings

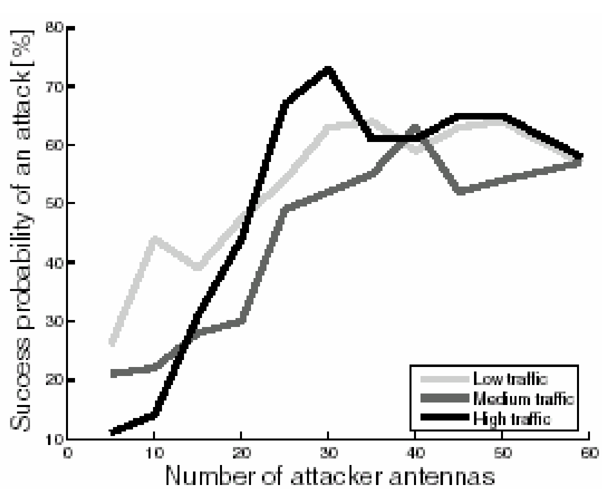
- we generated a simplified map of Budapest with MOVE
- we generated movement of the vehicles on the map with SUMO
  - low traffic: 250 new vehicles / time step
  - medium traffic: 500 new vehicles / time step
  - high traffic: 750 new vehicles / time step
- we selected the adversary's observation spots in intersections of roads
  - number of observation spots were varied from 5 to 59 with a step size of 5



Simulation settings

- we let the adversary build her model of the mix zone by letting her fully tracking vehicles for some time
- after that, we let the adversary pick a vehicle, track it until it enters the mix zone, observe exiting vehicles, and make a decision
- we run 100 simulations for each simulation setting
- we look at the percentage of the simulation runs where the adversary is successful

Simulation results



Summary

- changing pseudonyms has been proposed as a mechanism to provide location privacy in vehicular networks
- we studied the effectiveness of this approach
  - a model based on the concept of the mix zone
  - characterization of the adversary's tracking strategy
  - privacy metric
  - simulation results using realistic settings
- how about the frequency of the pseudonym change?