University of Stuttgart
Institute of Parallel and Distributed Systems (IPVS)
Universitätsstraße 38
D-70569 Stuttgart

On the Privacy of Frequently Visited User Locations

Zohaib Riaz, Frank Dürr, Kurt Rothermel

International Conference on Mobile Data Management 2016 (MDM'16)

15th June 2016

Motivation

- Today's Location-sharing apps exploit ***Location Semantics***
  - **App Examples:** Family locators, Friend finders, Geo-social networks
  - **Meaningful sharing:** (48.778786, 9.177867) → *Starbucks (Coffee Shop)*
  - Trivial to label any unlabeled trips (via Foursquare, Yelp etc.)

Motivation

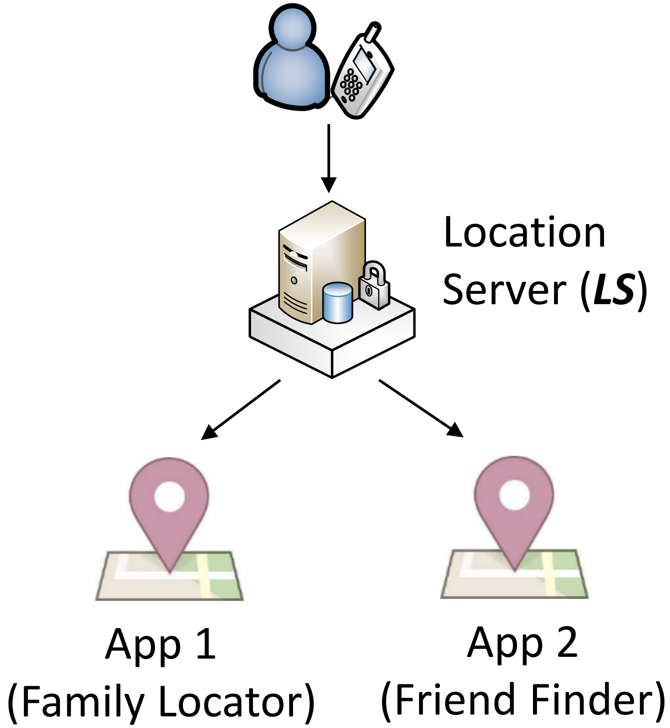
- Today's Location-sharing apps exploit ***Location Semantics***
  - **App Examples:** Family locators, Friend finders, Geo-social networks
  - **Meaningful sharing:** (48.778786, 9.177867) → *Starbucks (Coffee Shop)*
  - Trivial to label any unlabeled trips (via Foursquare, Yelp etc.)

- ***Privacy threat:*** Prolonged location sharing reveals <u>visit-frequency profile</u>!
  - *"A person who knows **all of another's travels** can deduce whether he is **a weekly church goer, a heavy drinker, a regular at the gym**,…— <u>and not just one such fact about a person, but all such facts</u>."* [United States v. Jones]

Motivation

- Typical Location-sharing apps rely on backend ***Location Server Infrastructure***
  - LSs store and manage user positions
  - Applications query user positions from LSs

- **Can the LSs provider be trusted for the security of users' location data?**



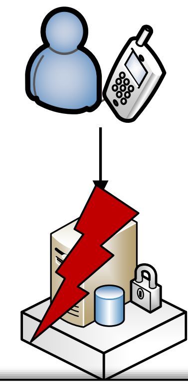
Motivation

- Typical Location-sharing apps rely on backend ***Location Server Infrastructure***
  - LSs store and manage user positions
  - Applications query user positions from LSs

- **Can the LSs provider be trusted for the security of users' location data?**

→ **No service provider can guarantee that personal information is safe!**
  - LS become <u>single-point-of-failure</u> w.r.t. privacy

"The alarming part is that the information is so concentrated,"



The Washington Post

*<u>**eBay**</u> asks **145 million users** to change passwords after data breach **(2014)***

REUTERS

*Database of <u>**191 million U.S. voters**</u> exposed on Internet: <u>researcher</u>* **(2015)**

THE WALL STREET JOURNAL.

*<u>**Twitter**</u>: Passwords Leaked for **Millions of Accounts*** **(6 days ago!)**

Contributions

- *A study of real-world check-ins dataset* to show that ***frequent locations*** pose a serious privacy threat (***next 3 slides …***)

- *An approach to protect frequent locations* while avoiding a single-point-of-failure in the LS infrastructure

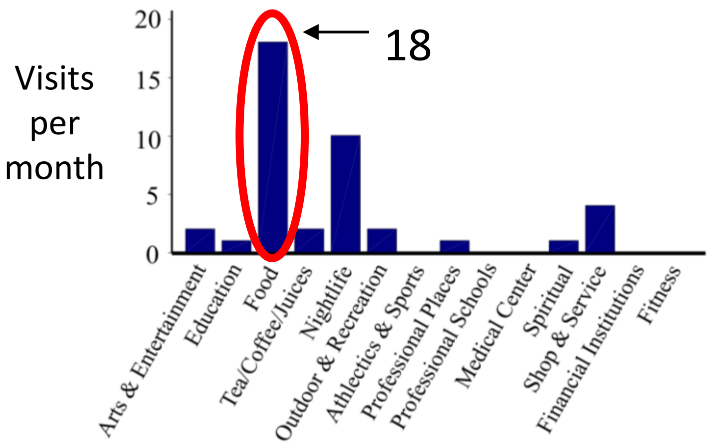
- *Evaluation of the approach* for achieved Privacy and Quality-of-Service (QoS) for location-sharing apps.

Study of Check-in Dataset: *Preprocessing*

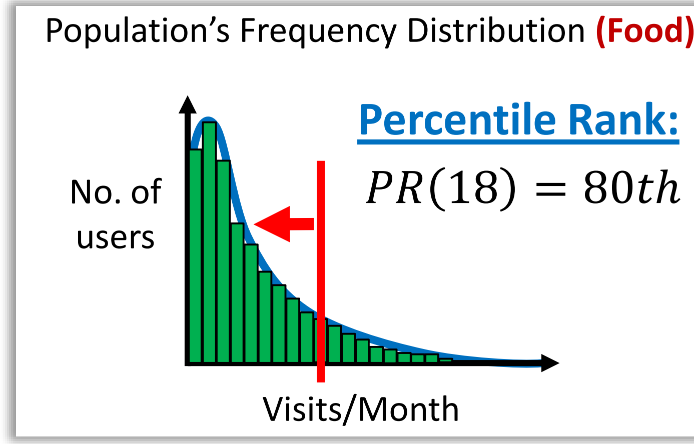
- **Goal:** Show that visit-frequency information poses a privacy threat
- **Dataset:** *22,506,721* **Geo-tagged tweets** provided by Cheng et al. 2011
- **Selected user *Population*:** criteria
  - >= 1 location check-in per day
  - >= 30 days of reported location data
  - **<u>10,306</u> users selected**
- Venue information, e.g., category, retrieved using Foursquare's free API

No.	Category
1	Arts & Entertainment
2	Education
3	Food
4	Tea/Coffee/Juices
5	Nightlife
6	Outdoor & Recreation
7	Athletics & Sports
8	Professional Places
9	Professional Schools
10	Medical Center
11	Spiritual
12	Shop & Service
13	Financial Institutions
14	Fitness

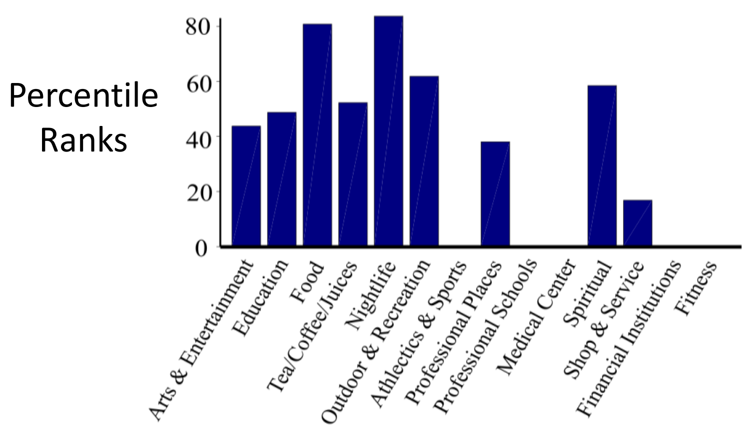
Study of Check-in Dataset: *Analysis*



Step 1: Get frequency-profile

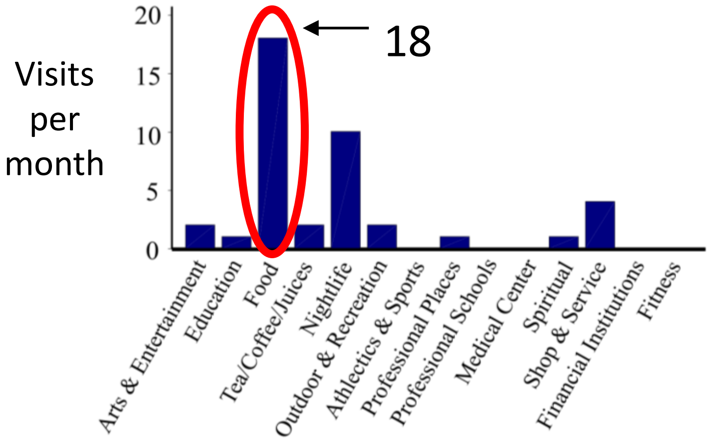


Step 2: Determine Significance of visit-frequency

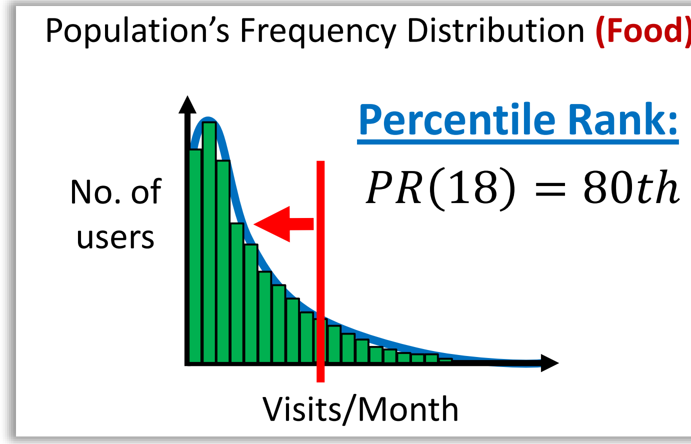


Output: Frequency rank-profile

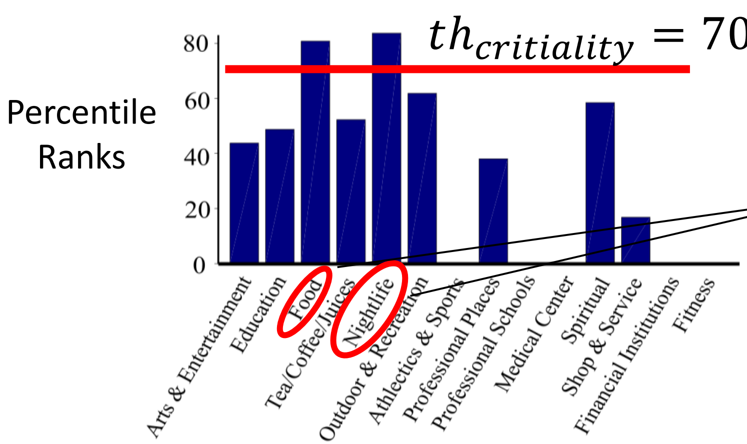
Study of Check-in Dataset: *Analysis*



Step 1: Get frequency-profile



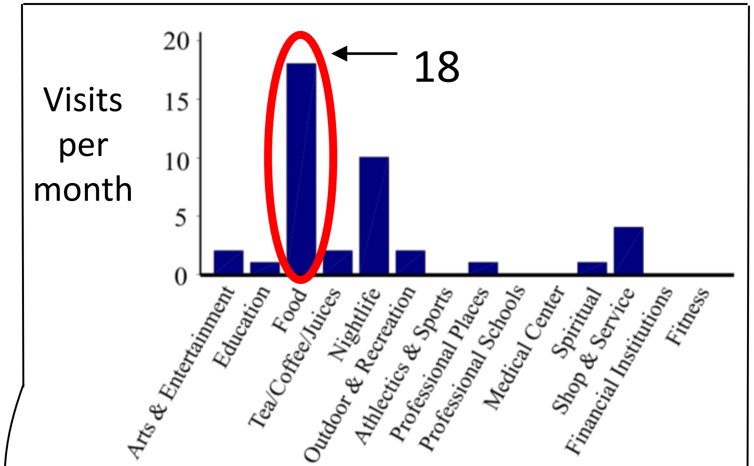
Step 2: Determine Significance of visit-frequency



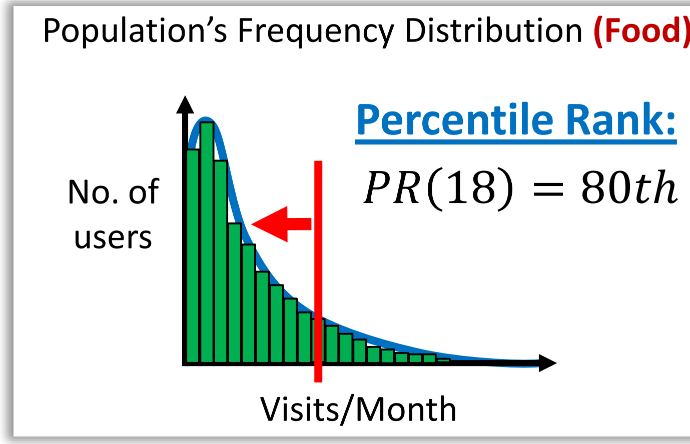
Critical locations:
visited with critically high frequencies

Output: Frequency rank-profile

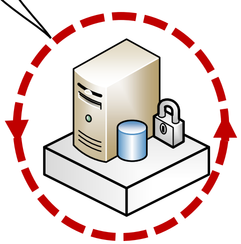
Study of Check-in Dataset: *Analysis*

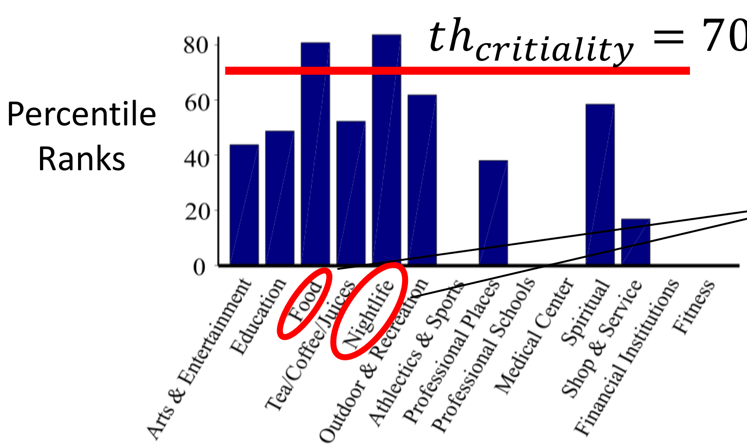


Step 1: Get frequency-profile



Step 2: Determine Significance of visit-frequency

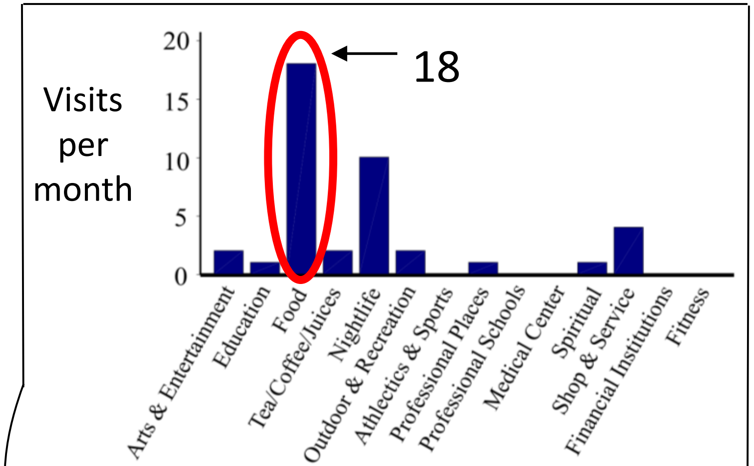




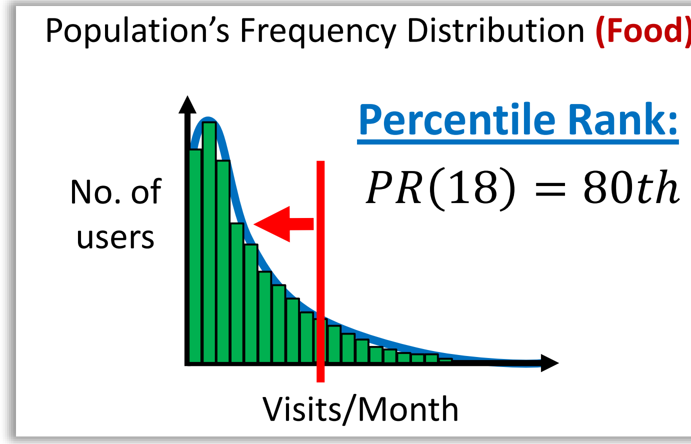
Critical locations:
visited with <u>critically high frequencies</u>

Output: Frequency rank-profile

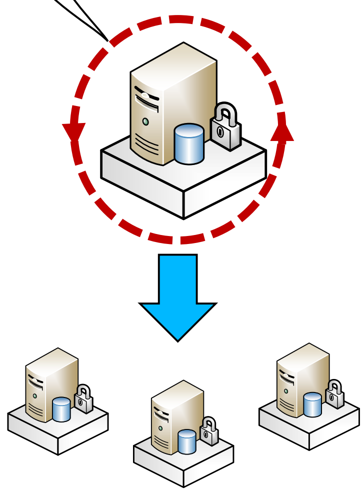
Study of Check-in Dataset: *Analysis*

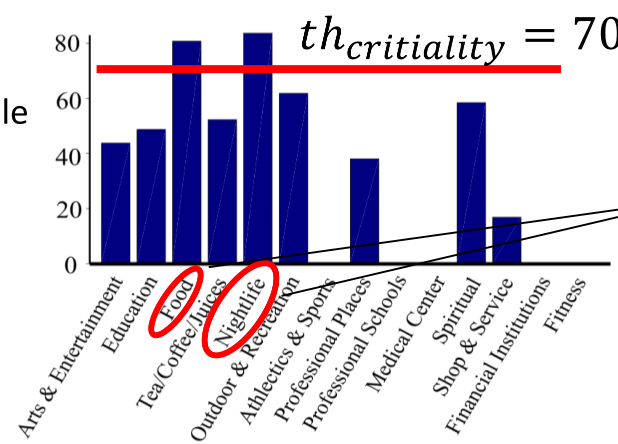


Step 1: Get frequency-profile



Step 2: Determine Significance of visit-frequency



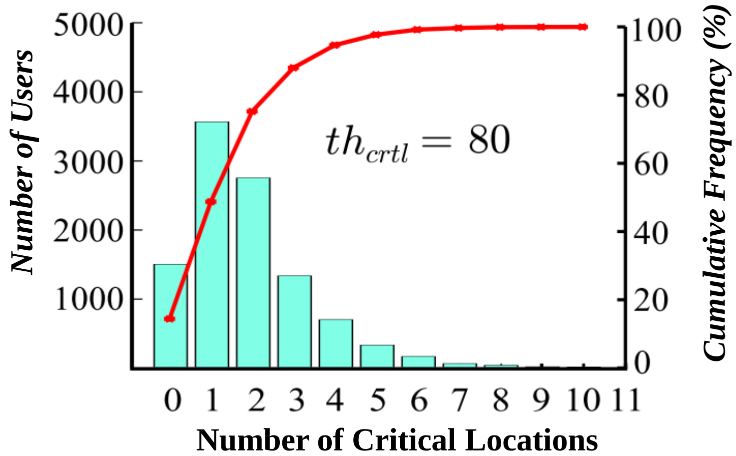


Critical locations:
visited with *critically high frequencies*

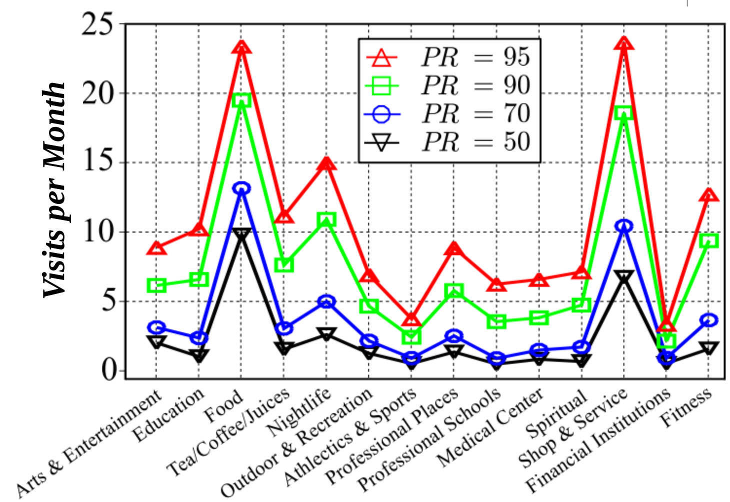
Output: Frequency rank-profile

Study of Check-in Dataset: *Evidence of Privacy Threat!*

- **Critical locations are prevalent!**
  - ~85% users have <u>at least 1</u> critical location
  - ~50% users have <u>2 or more</u> critical locations

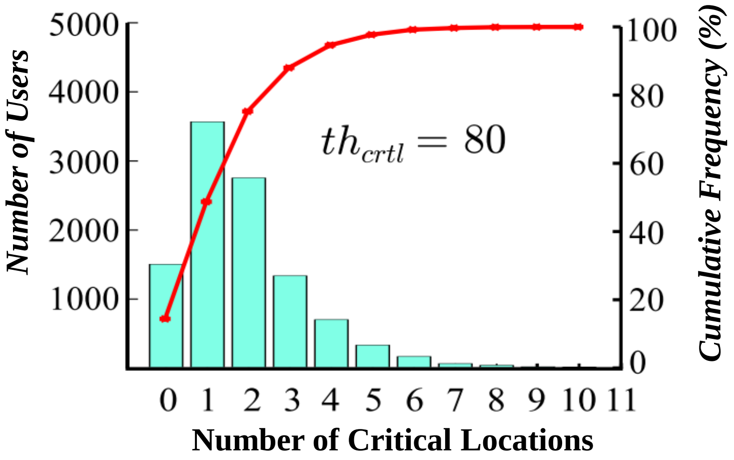


- **Visiting characteristics of locations**
  - Same percentile rank → different frequencies for diff. categories
  - High percentile-rank → a reasonable measure of user interest

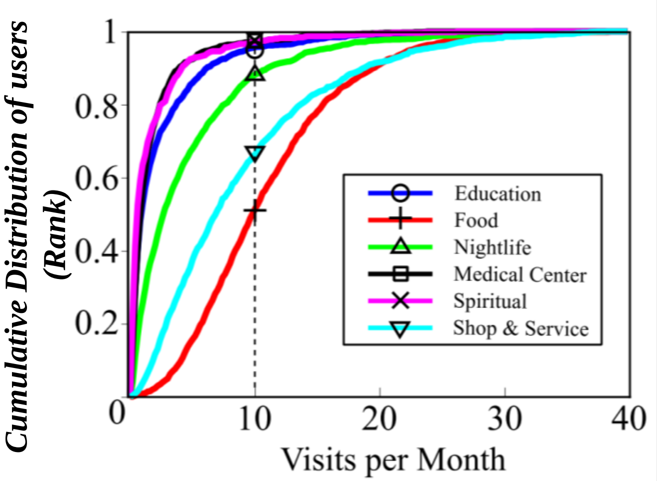


Study of Check-in Dataset: *Evidence of Privacy Threat!*

- **Critical locations are prevalent!**
  - ~85% users have <u>at least 1</u> critical location
  - ~50% users have <u>2 or more</u> critical locations



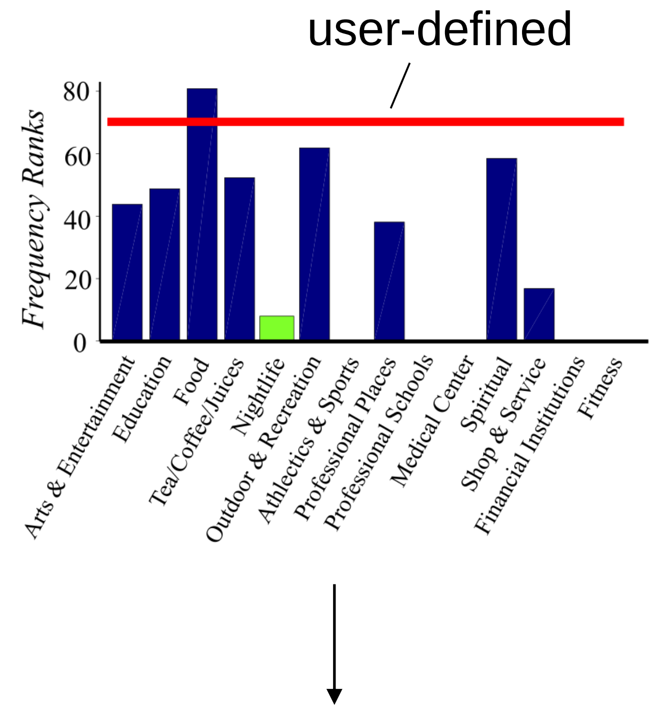
- **Visiting characteristics of locations**
  - Same percentile rank → different frequencies for diff. categories
  - High percentile-rank → a reasonable measure of user interest



→ We assume that "Frequency ⇔ Rank" relationship is publicly known

Problem Statement

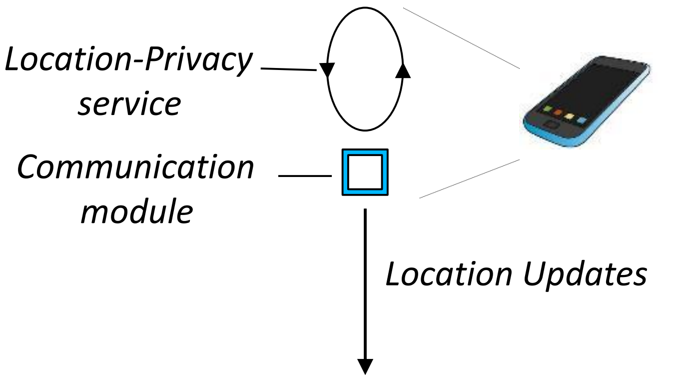
- **Privacy-preferences:** *(Persona, App)* pairs
- **User-Personas:** Define location categories whose criticality may be revealed e.g.:
  - *"Friends"* → {all} \ {Medical}
  - *"Colleagues"* → {all} \ {Spiritual, Medical, Night-life}
- **Privacy Requirement:** Implement privacy-preferences in location-sharing
  - Reveal unshared critical locations such that they *appear non-critical*
  - Avoid attacks to reveal critical locations
    - Attacker know our algorithm + additional knowledge
    - Baseline attack: ***random guess!***



$\alpha_{rand} = 1/8$

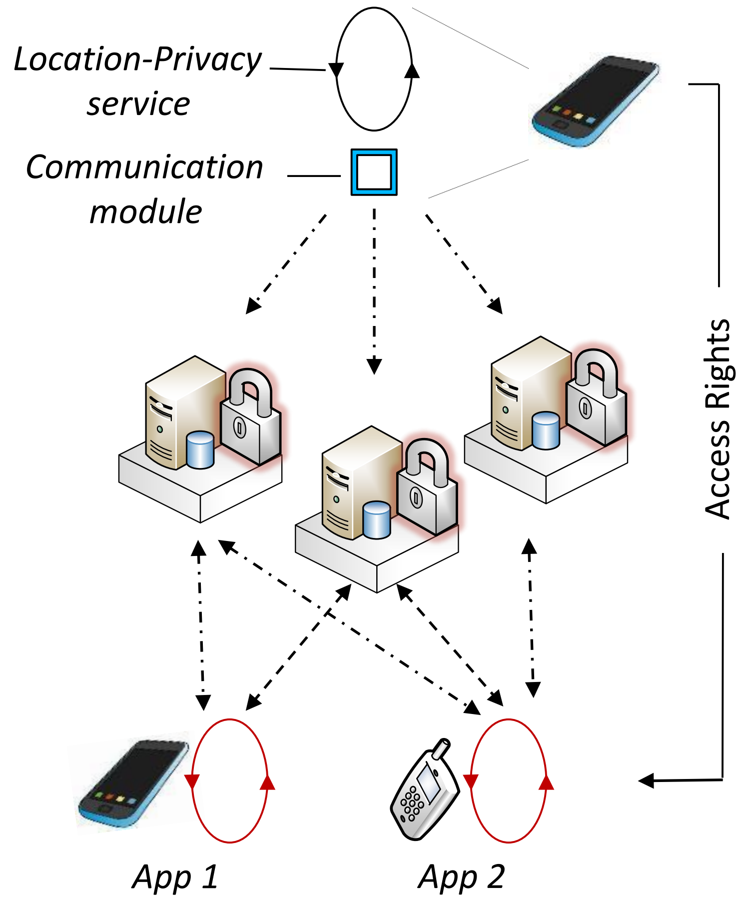
Architecture (1)

- **User's device:** a smart-phone
  - Runs **Location-Privacy service:**
    - Executes our privacy algorithm
    - Performs location updates to LSs
  - **Assumption:** Encrypted communication channel



Architecture (2)

- **A set of Location Servers (<u>LSs</u>)**
  - from different *third-party providers*
    - Example: *Backendless, App42, Heroku etc.*
  - manage location updates
  - implement Access-control mechanism
- **Location Based Applications (<u>Apps</u>)**
  - Get access authorization to LSs from users
  - Access user location from LSs or subscribe for update notifications
  - May aggregate frequency-profile of user
  - User-profile' precision ∝ no. of accessible LSs

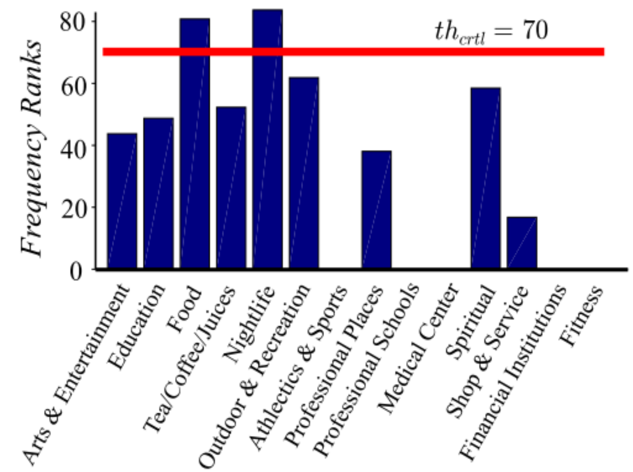


Basic Privacy Algorithm (1)

1. On-device determination of critical locations:

- $\boldsymbol{S} = \{s_1, \dots, s_{14}\}$, set of location categories
- $f_u = \{f_{s_1}, f_{s_2}, \dots, f_{s_{14}})$ and $r_u = \{r_{s_1}, r_{s_2}, \dots, r_{s_{14}}\}$
- Critical locations: $\boldsymbol{C}_u = s_i | \, r_{s_i} > th_{crtl}$

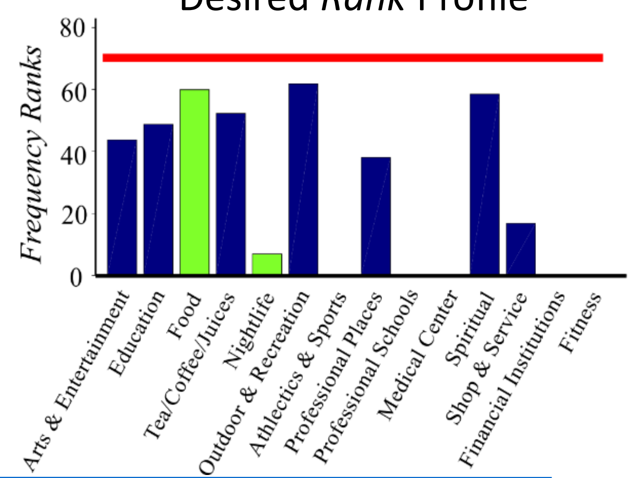
Actual *Rank* Profile



2. Determine desired ranks

- Deterministic new ranks → reversible by attacker
- **Randomized selection** of desired ranks
  - Avoids advanced attacks!

Desired *Rank* Profile

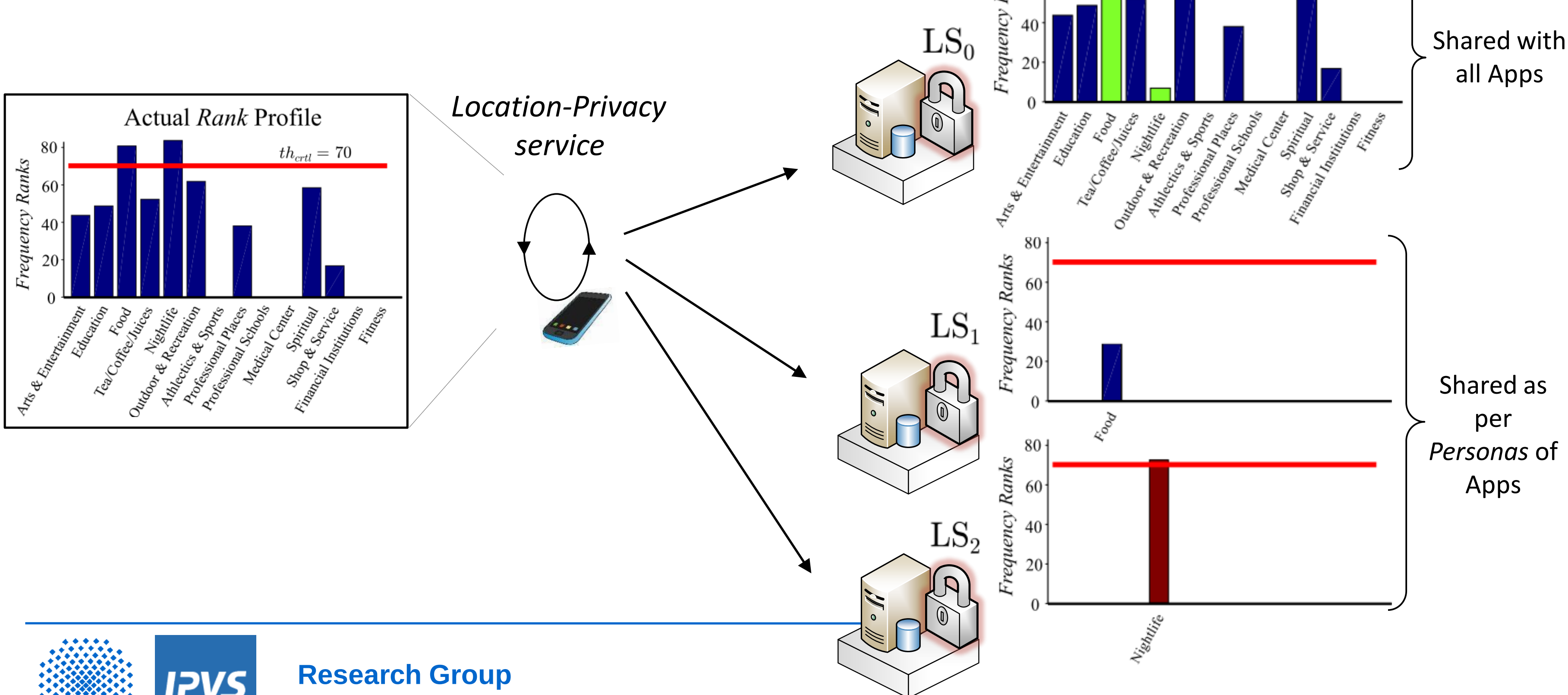


Basic Privacy Algorithm (2)

3. Enforce desired ranks for all $s_i \in C_u$:

- Divide trips for s_i among LSs
- LS_0 hosts a <u>safe-profile</u> of user

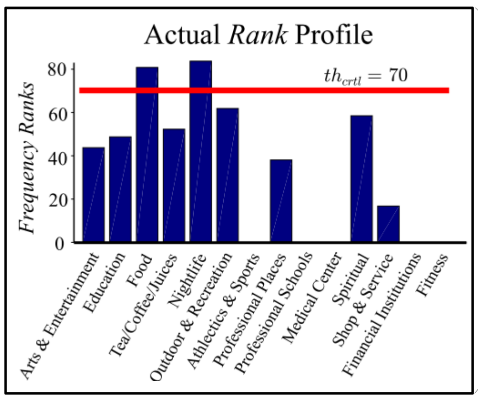
Protected *Rank* profiles at LSs

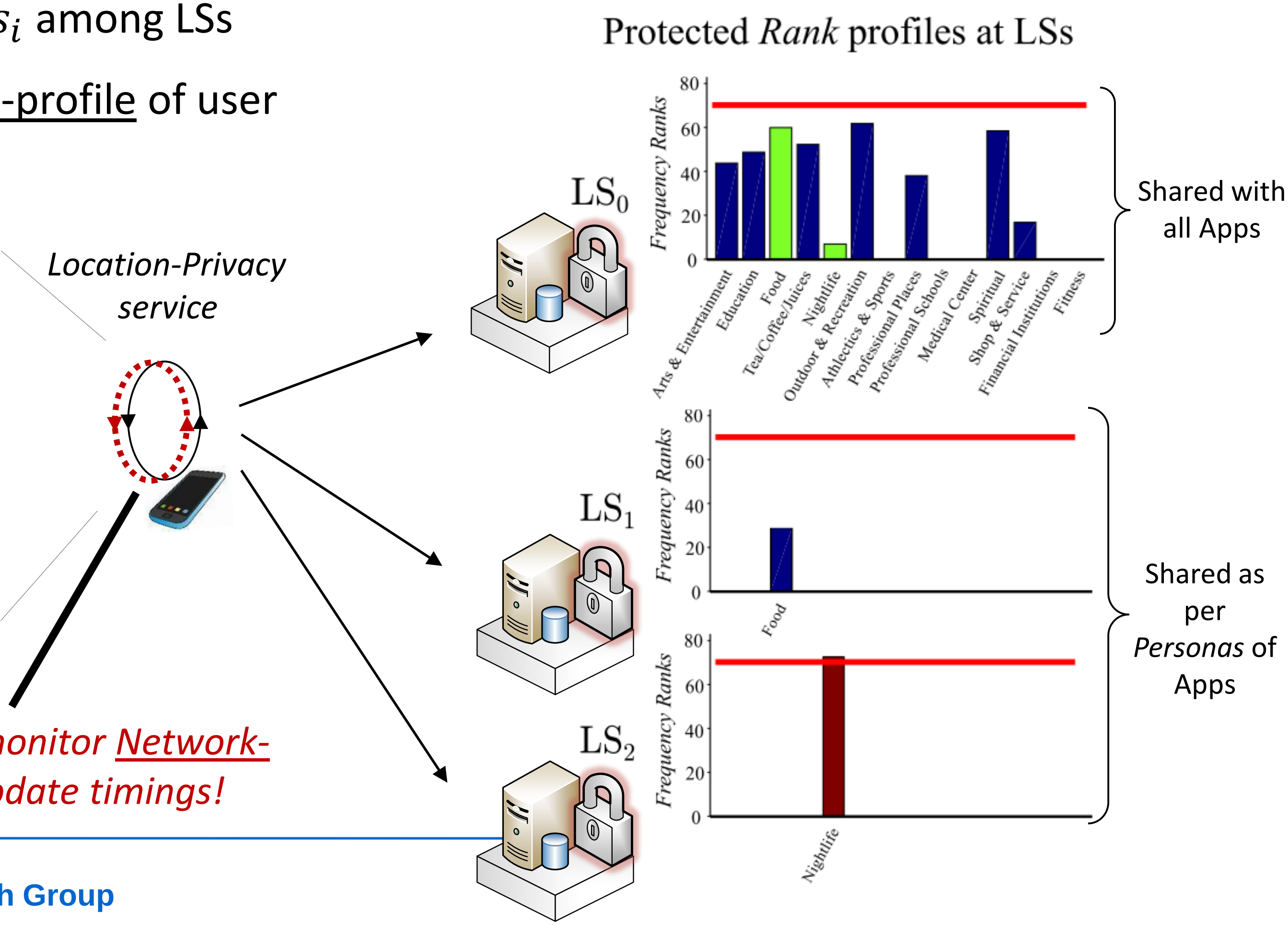


Basic Privacy Algorithm (2)

3. Enforce desired ranks for all $s_i \in C_u$:

- Divide trips for s_i among LSs
- LS_0 hosts a <u>safe-profile</u> of user





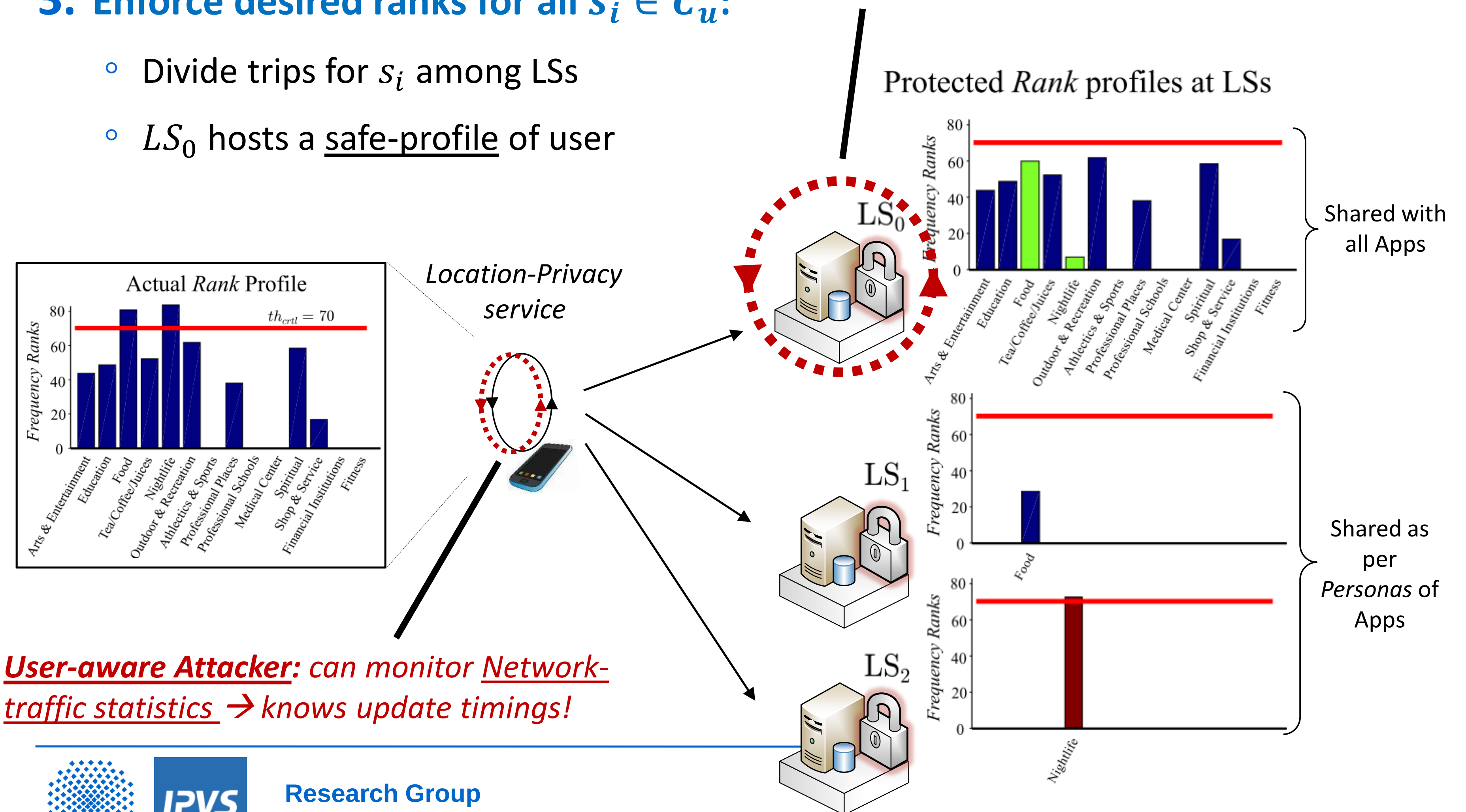
***<u>User-aware Attacker</u>:** can monitor <u>Network-traffic statistics</u> → knows update timings!*

Basic Privacy Algorithm (2)

3. Enforce desired ranks for all $s_i \in C_u$:

- Divide trips for s_i among LSs
- LS_0 hosts a <u>safe-profile</u> of user

***<u>Population-aware Attacker</u>:** Also possesses <u>location information from the other users</u>*



***<u>User-aware Attacker</u>:** can monitor <u>Network-traffic statistics</u> → knows update timings!*

Advanced Privacy Algorithm (1): Against *user-aware* Attacker

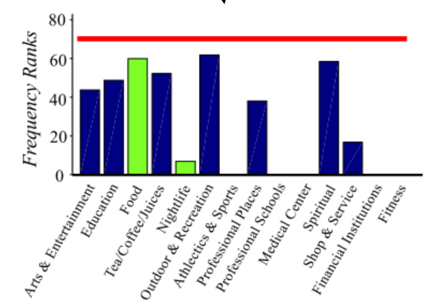
- *Attacker*: has access to a few LSs
  - Knows timings of inaccessible updates
- Trail of location updates → ***Mobility Model*** Ω

t *Attacker*

s_i =?

Inaccessible updates

t_1	t_2	t_3	t_4	$\ldots$	t_{150}	t_{151}	t_{152}	t_{153}	...
A	F	E	N		?	E	A	?	



- **Attack inaccessible updates:** Maximize $P(s_i|t)$ for s_i to predict visited location using Ω

- Bayes theorem: $P(s_i|t) = \dfrac{P(t|s_i)\,P(s_i)}{P(t)}$

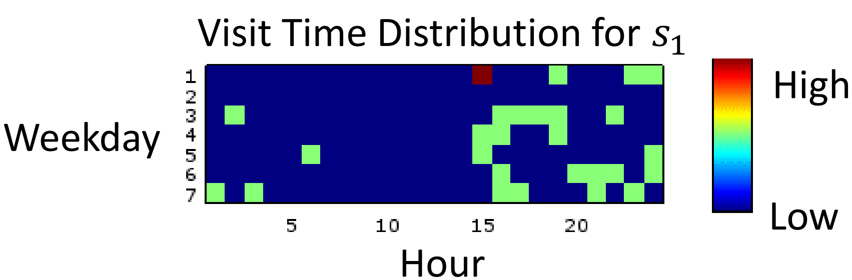
Prior: Changed by our algorithm (unreliable)

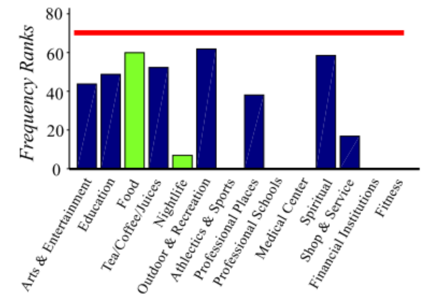
Normalizer: constant for all s_i (unimportant)

Likelihood of visiting time s_i at time t over all possible times T

Advanced Privacy Algorithm (2): Defense

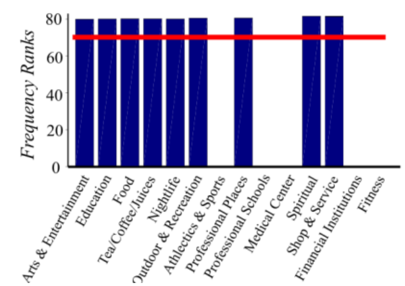
- **Defense:** Generate ***fake events*** for each location as if it were critical!
  - Fake events → ***garbage data → discarded by LSs!***
  - **Desired effect:** Rank of all locations should “appear” equal
- **Algorithmic steps:**
  1. Keep track of temporal likelihood of each category
  2. Accordingly schedule enough fake events to meet maximum rank in the rank-profile





+

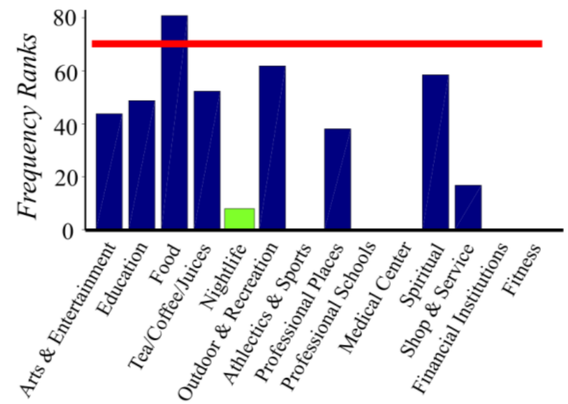
fake/inaccessible updates



“Attacker’s View”

Evaluation: ***Population-aware*** Attacker Model

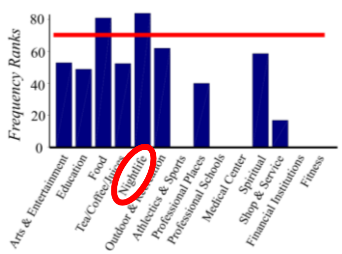
- **Attacker:** Aims to find all critical locations
  1. knows 'k' out of 'n' critical locations from **authorized or compromised LSs**
  2. Knows **correlations among visit-frequencies of different location categories** *(Acquired from the population)*



Classification

Regression

Nightlife

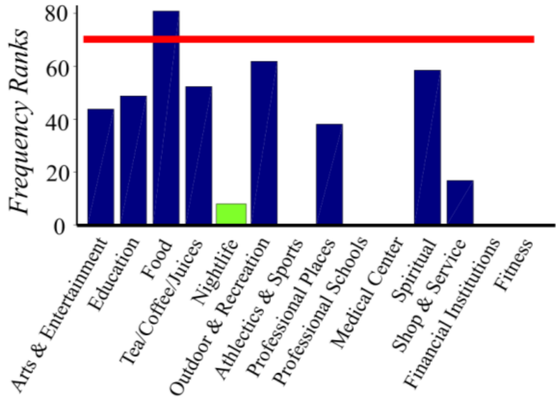


- **Frequency-correlation attack**
  - Learn correlations using **Machine Learning techniques**
  - **Data:** Frequency-profiles of 10,036 users

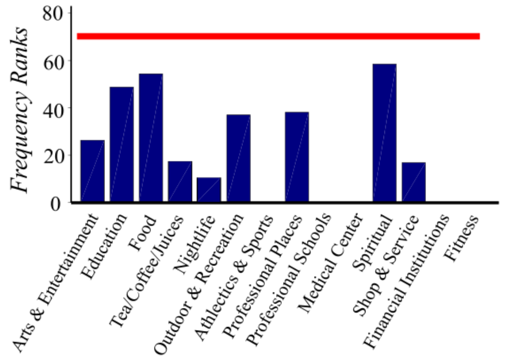
Evaluation: Privacy results for Classification Attacks

- **Classifiers:** *Random Forest* (RF) & *Support-Vector Machine* (SVM)
- **Training:**
  - On frequency-profiles with *one critical location*
  - 10-fold cross-validation
- **Results:**
  - Low classification accuracy: ***25%***
- **Repeated experiment:**
  - Added frequency-profiles with no critical locations!
  - Again, low accuracy for critical locations: ***22%***
  - ***High accuracy for non-critical: 87%***

Protected profile (altered)



non-critical profile (unaltered)



Evaluation: Privacy results for Classification Attacks

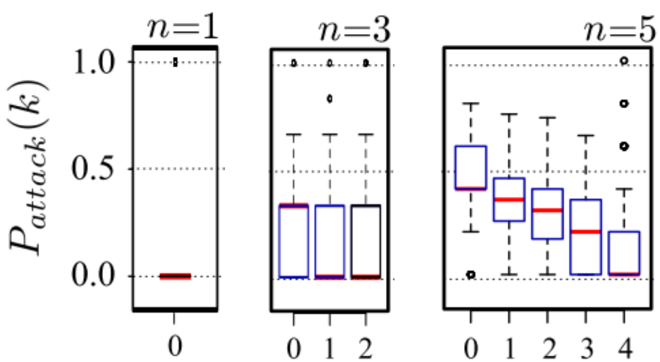
Actual Classes \ Predicted Classes	non-critical	Arts & Entertainment	Education	Food	Tea/Coffee/Juices	Nightlife	Outdoor & Recreation	Athletics & Sports	Professional Places	Professional Schools	Medical Center	Spiritual	Shop & Service	Financial Institutions	Fitness	***Recall***
Precision	0.73	0.28	0.20	0.32	0.13	0.35	0.15	0.08	0.18	0.00	0.14	0.12	0.26	0.00	0.22	
Fitness	0.03	0.10	0.08	0.13	0.06	0.19	0.03	0.00	0.08	0.00	0.02	0.01	0.15	0.00	0.12	0.12
Financial Institutions	0.48	0.07	0.07	0.05	0.06	0.01	0.01	0.00	0.11	0.00	0.07	0.01	0.02	0.00	0.01	0.00
Shop & Service	0.01	0.04	0.12	0.12	0.05	0.17	0.01	0.00	0.06	0.00	0.02	0.00	0.36	0.00	0.04	0.36
Spiritual	0.20	0.10	0.14	0.02	0.11	0.06	0.05	0.01	0.09	0.00	0.04	0.05	0.11	0.00	0.01	0.05
Medical Center	0.27	0.10	0.07	0.06	0.08	0.07	0.03	0.00	0.11	0.00	0.08	0.01	0.07	0.00	0.03	0.08
Professional Schools	0.30	0.04	0.04	0.04	0.00	0.04	0.11	0.00	0.19	0.00	0.11	0.07	0.07	0.00	0.00	0.00
Professional Places	0.15	0.11	0.13	0.06	0.06	0.08	0.04	0.01	0.17	0.00	0.05	0.01	0.10	0.00	0.03	0.17
Athletics & Sports	0.52	0.09	0.07	0.02	0.05	0.02	0.03	0.02	0.09	0.00	0.04	0.02	0.03	0.00	0.00	0.02
Outdoor & Recreation	0.32	0.11	0.06	0.06	0.05	0.07	0.10	0.01	0.10	0.00	0.02	0.02	0.05	0.00	0.02	0.10
Nightlife	0.03	0.06	0.08	0.09	0.07	0.39	0.02	0.00	0.06	0.00	0.01	0.01	0.16	0.00	0.02	0.39
Tea/Coffee/Juices	0.08	0.15	0.09	0.10	0.11	0.13	0.06	0.00	0.07	0.00	0.06	0.01	0.09	0.00	0.04	0.11
Food	0.03	0.07	0.08	0.39	0.08	0.11	0.02	0.00	0.05	0.00	0.01	0.01	0.13	0.00	0.01	0.39
Education	0.08	0.08	0.21	0.08	0.08	0.13	0.04	0.01	0.10	0.00	0.05	0.02	0.13	0.00	0.01	0.21
Arts & Entertainment	0.16	0.32	0.06	0.05	0.06	0.09	0.04	0.00	0.06	0.00	0.02	0.01	0.08	0.00	0.03	0.32
non-critical	0.87	0.03	0.01	0.00	0.01	0.00	0.02	0.01	0.03	0.00	0.01	0.00	0.00	0.00	0.00	0.87

Low accuracy

High Confusion

Evaluation: Privacy results for Regression Attacks

- **Regression Models:** *RF, SVM and Gaussian Mixture Regression* (GMR)
  - Percentage prediction error: < 5% for each semantic location
  - Attack performance on protected frequency-profiles:

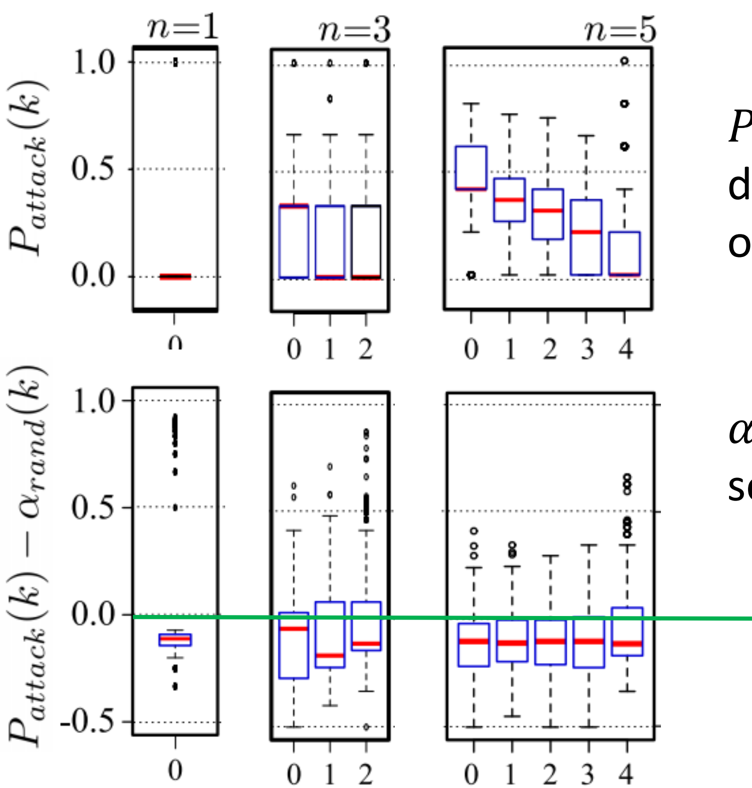


k = *known Critical Locations*

$P_{attack}(k)$ - probability of correct detection of a critical location when k out of n critical locations are already known

Evaluation: Privacy results for Regression Attacks

- **Regression Models:** *RF, SVM and Gaussian Mixture Regression* (GMR)
  - Percentage prediction error: < 5% for each semantic location
  - Attack performance on protected frequency-profiles:



$P_{attack}(k)$ - probability of correct detection of a critical location when k out of n critical locations are already known

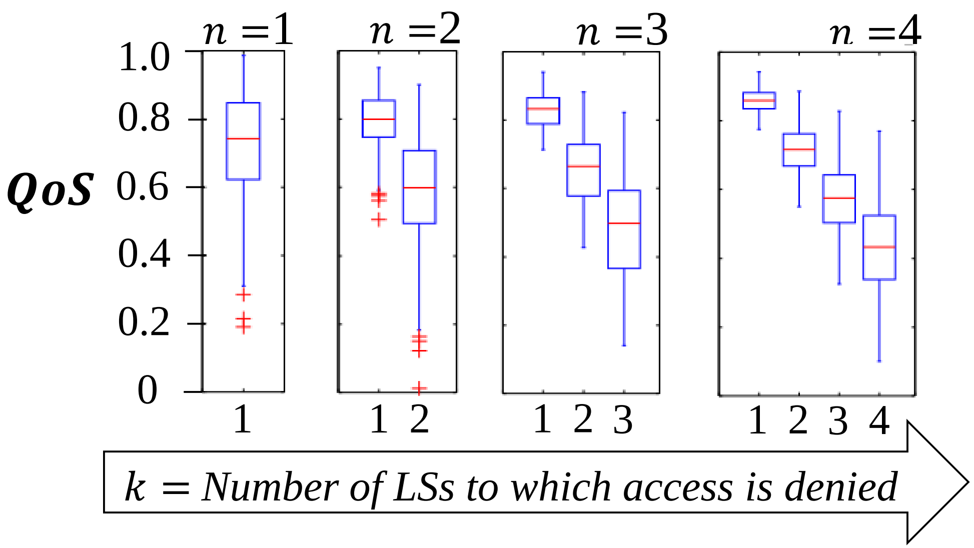
$\alpha_{rand}(k)$ - probability of randomly selecting a critical location

← Zero knowledge gain

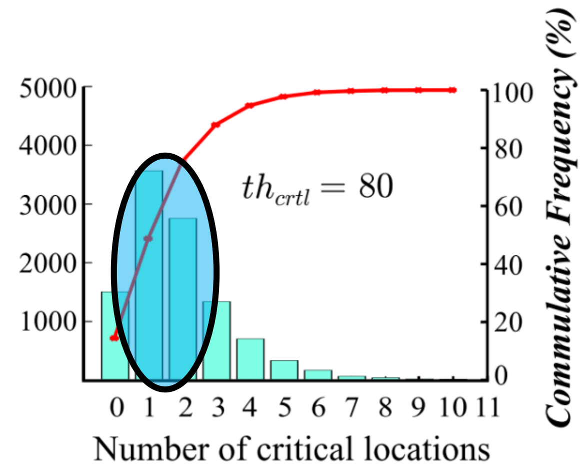
k = *known Critical Locations*

Evaluation: QoS and Communication Overhead

- **QoS** = proportion of available location updates



$k = 1, QoS \sim 80\%$
$k = 2, QoS \sim 70\%$
$k = 3, QoS \sim 60\%$



  - QoS is reasonably high given 60% population has 1 or 2 critical locations

- **Communication Cost** = no. of fake message per day
  - 1-2 messages a day for most users!

Related Work

- **Semantic location obfuscation (PROBE framework by Damiani et al. 2010)**
  - \+ Cloak individual sensitive visits with neighboring non-sensitive venues
  - – Sensitivity of location categories is not related to an individual's visit-frequency
- **Venue Recommendation techniques (Riboni et al. 2014, Zhang et al. 2014)**
  - \+ Offline publishing of check-in history statistics in a differentially private manner
  - – Require Trusted parties for implementing the privacy algorithm
  - – Cannot be used for online location sharing
- **Distributed Location Management (Duerr et al. at Percom 2011)**
  - \+ No *single-point-of-failure*
  - – For single locations without considering location semantics

Conclusion & Future Work

- ***Frequent locations*** naturally pose a privacy threat by revealing user interests
- ***Distributing location information*** in LS infrastructure → ***promising privacy solution***

- Proposed an algorithm for controlled sharing of frequent locations
  - Hides frequent locations from:
    - ***User-aware attackers***
    - ***Population-aware attackers***

- **Future Work**
  - Integrate existing *single-location* semantic obfuscation approaches for forming a comprehensive privacy mechanism

Contact and Discussion

PriLoc

www.priloc.de

Zohaib Riaz

Institute for Parallel and Distributed Systems,

University of Stuttgart, Germany

zohaib.riaz@ipvs.uni-stuttgart.de