Wireless Network Security and Privacy

Autumn 2023

Xiaoyu Ji

Location Service Security

Agenda

- GPS and its security
- LBS services and its security
- Attacks on location services

GPS

- Global Position System was developed by the US DoD initially in the 1970s and completely operational in 1994
  - Similar to other systems deployed by Russia, EU, China（北斗）, India, and others

- Satellites broadcast current time and their location to allow receivers on Earth (and elsewhere) to localize

Things using GPS

- GPS is used for:
  - Automobile navigation (and autonomous driving)
  - Mobile geo-location (for LBS, etc.)
  - Livestock / wildlife tracking
  - Aircraft and ship navigation and autopilot

  - Power grid synchronization
  - Financial transactions & trading
  - Telecom system operations
  - …

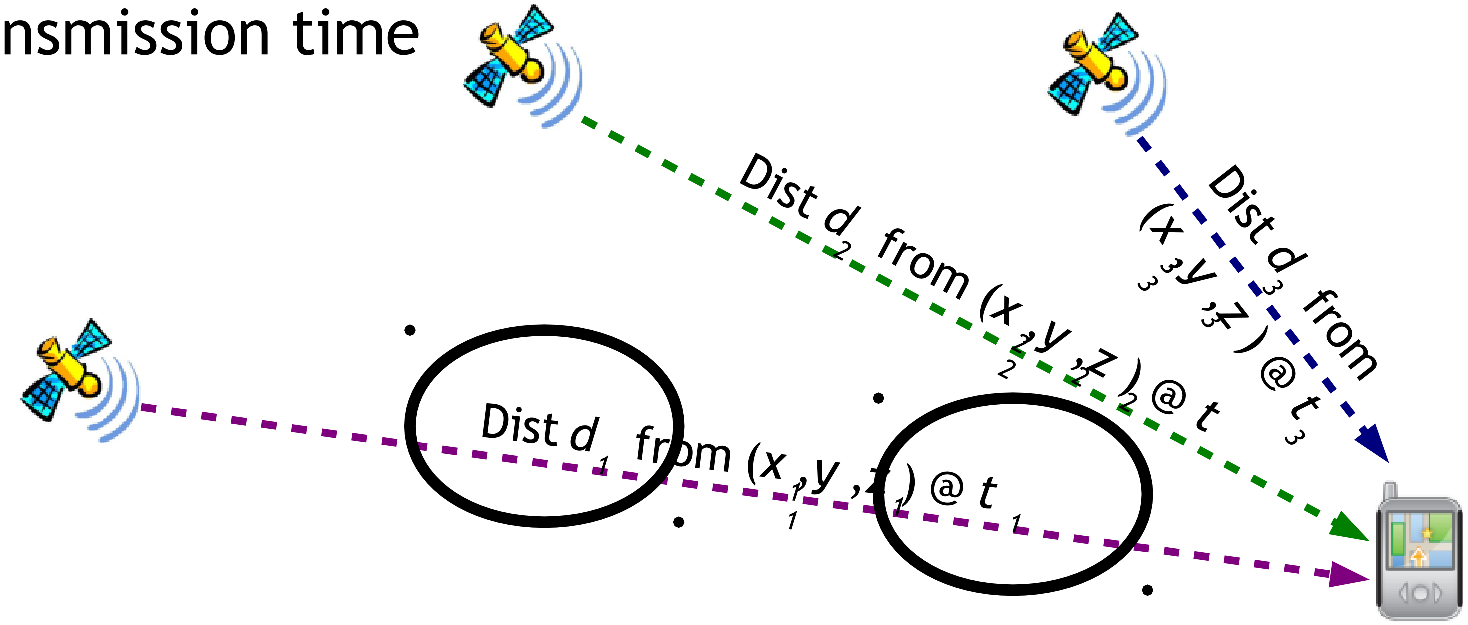
So, how does GPS actually work?

GPS Signals

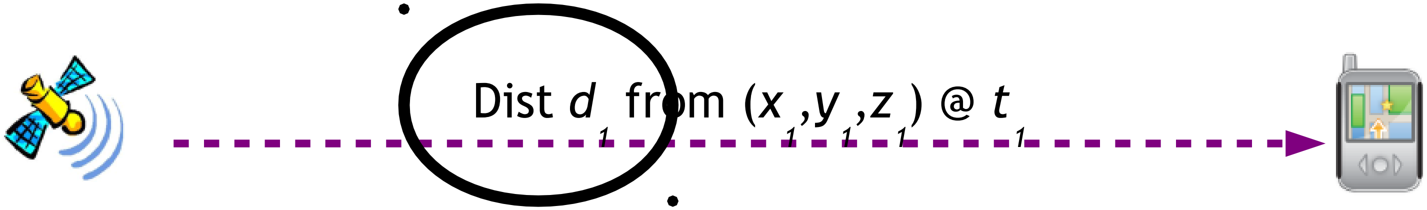
- GPS satellites send several different signals
  - On the L1 band (1575.42 MHz), coarse-acquisition (C/A) signal, encrypted precision (P(Y)) signal, L1 civilian (L1C) and military (M) codes
  - On the L2 band (1227.60 MHz), P(Y) code, L2C and M
  - Three other bands (L3, L4, L5) used for other purposes
    - Nuclear detonation detection, atmospheric correction, civilian safety-of-life

Multilateration

- GPS satellites serve as mobile reference points for Earth-based receivers
  - All satellites have high-precision, tightly synchronized clocks and precisely known locations
  - Each receiver hears a coordinate and timestamp from each transmitter, measures the distance based on the transmission time


Dist d_2 from (x_2,y_2,z_2) @ t_2
Dist d_3 from (x_3,y_3,z_3) @ t_3
Dist d_1 from (x_1,y_1,z_1) @ t_1

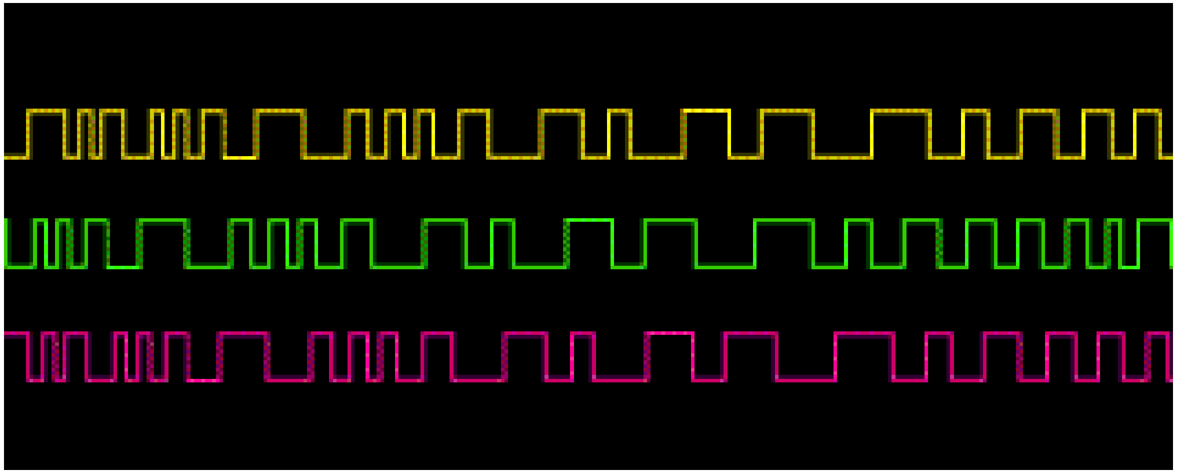
Measuring Distance



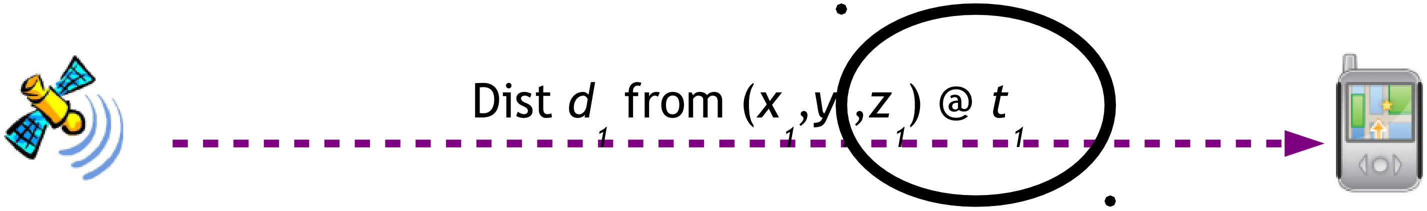
Dist d_1 from (x_1,y_1,z_1) @ t_1

- How to measure distance from the satellite?

- Well, *distance* = *speed of light* * *time*, so just measure time...



Receiver Timing



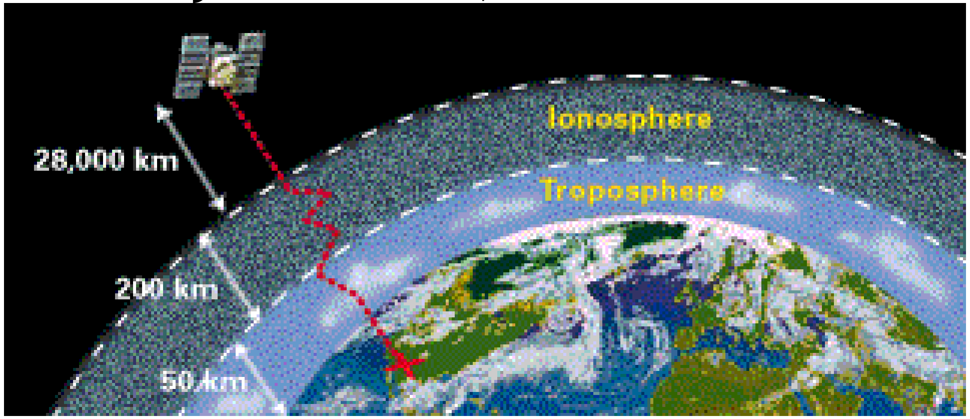
Dist d_1 from (x_1,y_1,z_1) @ t_1

- Satellites themselves use atomic clocks to maintain ground truth
  - Receivers have to synchronize with the satellites
  - Remember,1ns time error → 1ft distance error

- With clever processing, an extra satellite signal provides required synchronization
  - 3 satellites for space, 4 for space+time

Errors

- Errors arise for many different reasons
  - Scattering through Earth's atmosphere, reflection off buildings, time sync errors, etc.



  - Much of this can be handled by incorporating proper models in the distance estimation process
    - But, no longer just *distance* = *speed* * *time*
  - Some receivers get diversity from using military & civilian signals

Military v. Civilian GPS

- Civilian GPS uses an unencrypted and unauthenticated signal for location and time synchronization

- Military GPS devices can be keyed to use an encrypted and authenticated signal for high assurance location and timing
  - Military GPS requires key management, often in the form of manually entering long keys into handsets
  - Use of the military signal can provide much higher accuracy, error correction, etc.

Military GPS Rumors

- Since manual key management is often an impediment to mission-critical activities, there have been reports that a large number of soldiers use GPS in civilian mode

Selective Availability

- When GPS was originally designed, it was intended to provide coarse-grained location for civilians and fine-grained location for military
  - Does anyone remember when GPS accuracy was 20-30 meters and that was good enough for most things?
- Selective Availability was eliminated around 2000 to provide higher accuracy for civilian applications
  - Usually, we can get <5 meter accuracy

Differential GPS

- For applications that require even better accuracy
  - Differential GPS uses an additional signal sent from a ground station to compensate for errors in data sent by satellites
  - E.g., DGPS stations can send difference between location claimed by satellite and its observed location
  - Accuracy of **~10cm** can be achieved using DGPS
    - Appropriate for autonomous / swarm vehicle applications

Jamming

- GPS is based on wireless communication, so it's subject to interference
- GPS RSS is on the order of femtowatts (~10^{-15} W or -120 dBm) [some sources say .1fW or 100 attoWatts]
  - Jamming is pretty easy

What are the possible security issues with GPS?

Replay Attacks

- Replay of GPS transmissions would involve stale timestamps and location information

- The content of the message would be good

- But the time sync step would fail and most likely give unreasonable results
  - Unless the timing is precisely controlled...more in a minute

GPS Spoofing

- Instead of replaying old GPS signals, fabricate new ones and pretend to be a satellite
  - Spoofing leverages lack of authentication in civilian GPS signals
- Provides invalid information to the receiver to force it to compute an incorrect location
- Practical spoofers have been demonstrated

Timed Replay Spoofing

- Todd Humphreys's team built a spoofer (see [Humphreys et al., ION GNSS 2008])
  - It receives signals, analyzes them, and replays them after a precise delay
  - The delay affects the distance measurement, thereby affecting the location result
  - Precise control of delay allows gradual error accumulation or “drifting”, so detection is difficult

Many More Attacks

- GPS receivers are also vulnerable to a number of signal- and software-based attacks
  - e.g., Middle-of-the-Earth attack
  - See [Nighswander et al., CCS 2012]

How could you protect against these GPS attacks / threats...

without replacing or upgrading the satellite systems?

Deployment Constraints

- Because of the deployment cost, upgrading or replacing satellites is not really an option
  - Maybe very slowly over time, but not any time soon
  - So authentication is out

- GPS receivers have to respect what the GPS transmitters are sending even if they cannot authenticate them

Alternatives

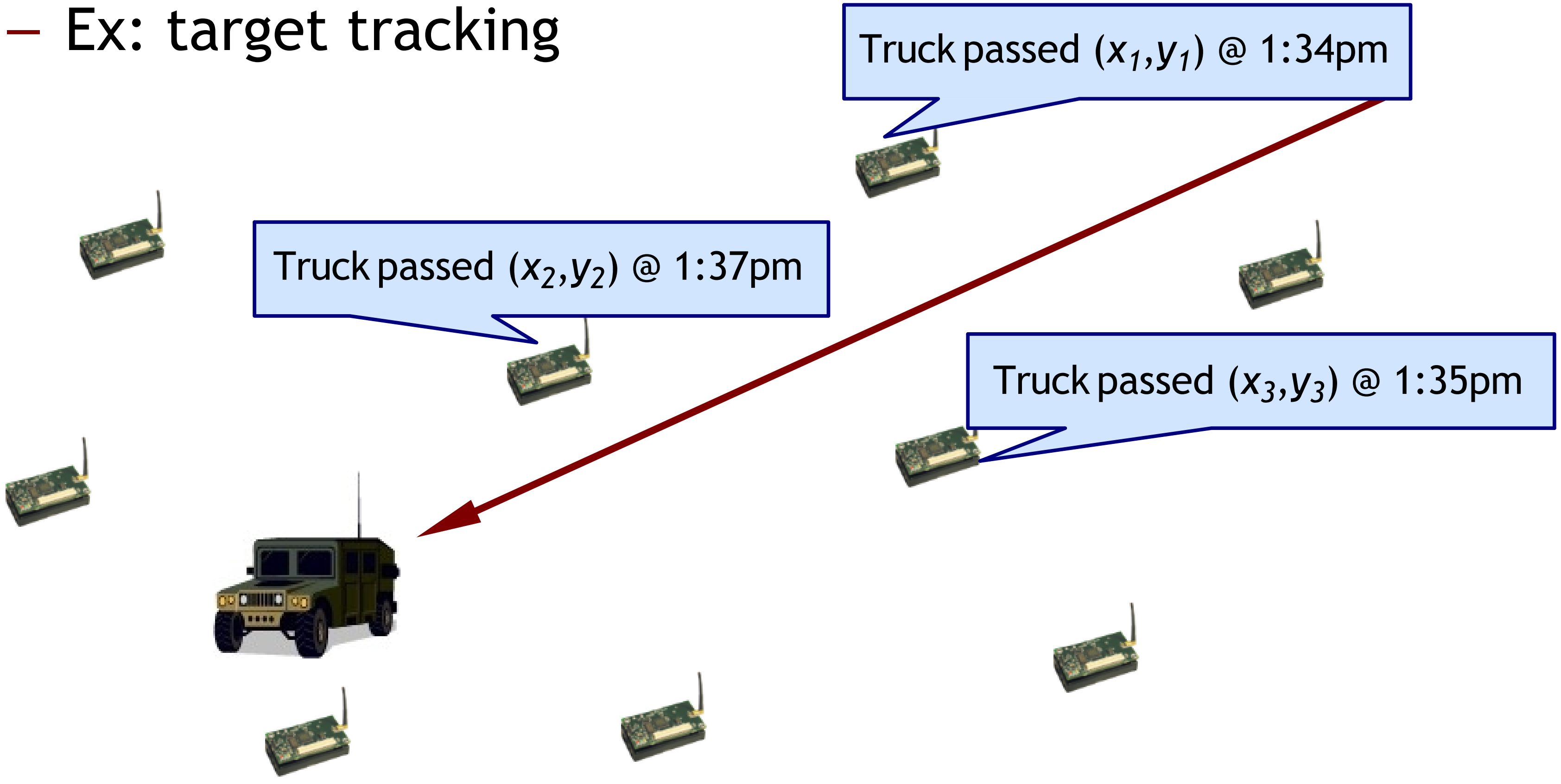
- Several defense / mitigation strategies have been proposed by the GNSS community
  - Modifying GPS receivers to use multiple antennas to verify angle of arrival consistency
  - Augment receiver software to compare changes in location over time
  - Incorporate sensor data (GPS says you're moving but gyro says you're not → ?)
  - Incorporate other GNSS systems for diversity

What about devices or scenarios where GPS is inappropriate?

Sensor networks, underground, etc.

Time & Location in WSN

- Many applications and protocols require fine-grained node location and event timestamping
  - Ex: target tracking



Truck passed (x_1,y_1) @ 1:34pm

Truck passed (x_2,y_2) @ 1:37pm

Truck passed (x_3,y_3) @ 1:35pm

WSN Sync & Loc

- Time and location services for WSN must be:
  - **Energy efficient** – energy spent for sync & loc should be minimized (noting significant cost for continuous CPU use and radio listening)
  - **Scalable** – large networks should be supportable
  - **Robust** – adaptable to network dynamics
  - **Ad hoc** – functionality without prior configuration or infrastructure

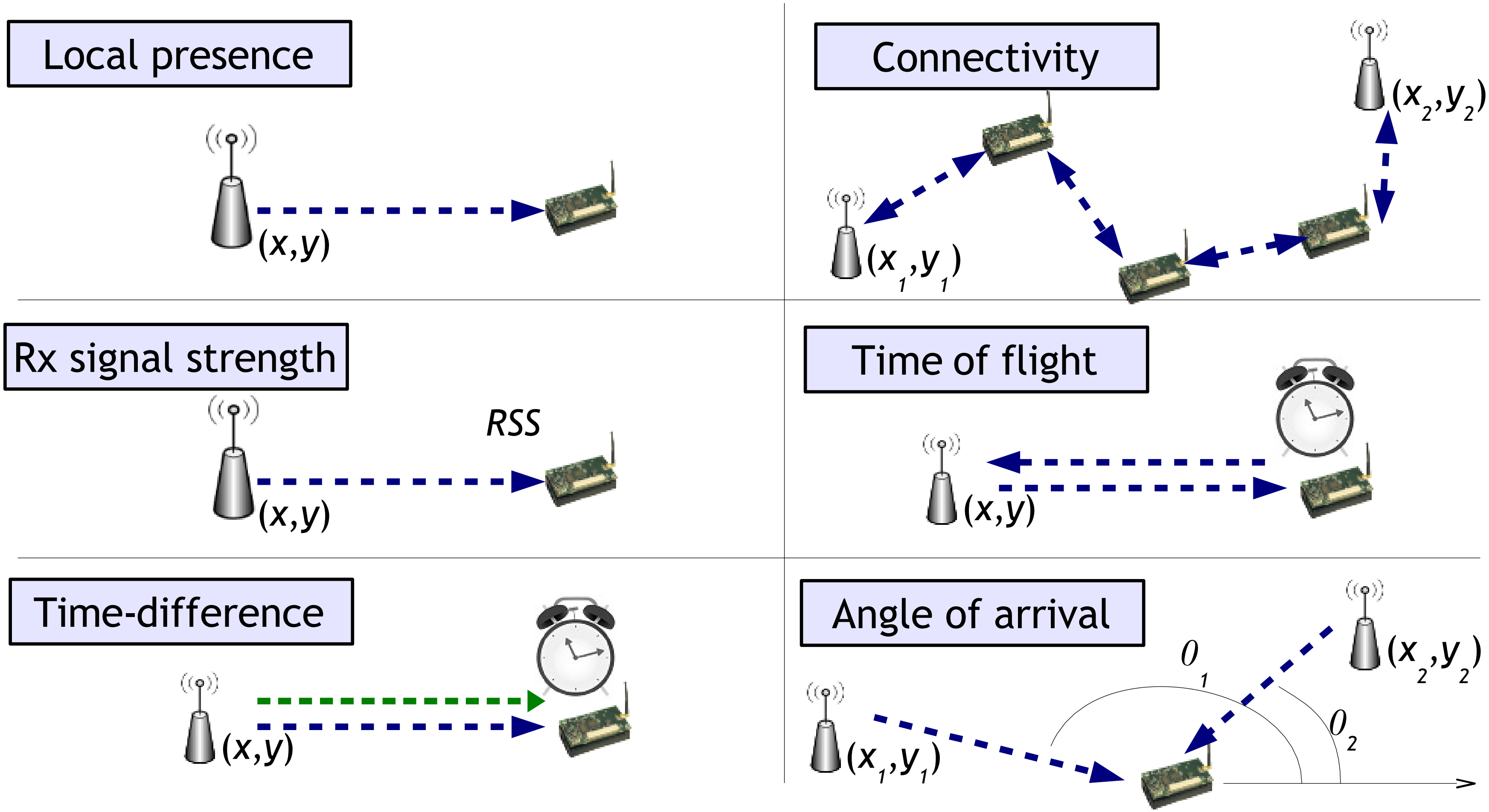
Many of the techniques used in synchronization and localization are similar

Relative Measurements

- Just as in GPS, time sync and localization mechanisms for wireless networks are based on relative measurements from others:
  - Receive a message from a neighbor
  - Content in message gives some information
  - Measurement about signal reception gives more

Relative Measurements

Each localizing device collects geometric relationships relative to several reference points (x_i, y_i)



Local presence

Connectivity

Rx signal strength

Time of flight

Time-difference

Angle of arrival

Securing Relative Measurements

- Measurements taken with respect to reference points should be:
  - **Authentic**
    - Measurements from authorized reference points only
  - **Verifiable**
    - Integrity of measurement should be guaranteed
    - If possible, physical measurement should be unforgeable
  - **Highly available**
    - Location information should be ready when needed
  - **Protected from various forms of attack**

Threats to Loc & Sync

- Most of the threats are related to lying
  - In both services, nodes act as references for each other
  - Malicious nodes can simply give false reference information
  - Bad information may be worse than no information, so this can be more serious than DoS

Secure Sync & Loc

Is it possible to secure the sync and loc processes?

- Processes are based on reference data
  - Is the source trustworthy?
  - Can the data be verified?
  - Is the data reliable?
- Reference data may be noisy or imprecise
  - How to incorporate redundancy for reliable estimation?
- Sync & loc estimation services can be attacked
  - Vulnerabilities?
  - How to mitigate them?
- System or devices may be tightly constrained
  - How efficient is the estimation algorithm?
  - What are the trade-offs?

Location Based Services (LBS):

In an abstract way

A certain *service* that is offered to the users *based* on their *locations*

LBS: Then

- Limited to fixed traffic signs

How many years we have used these signs as the ONLY source for LBS

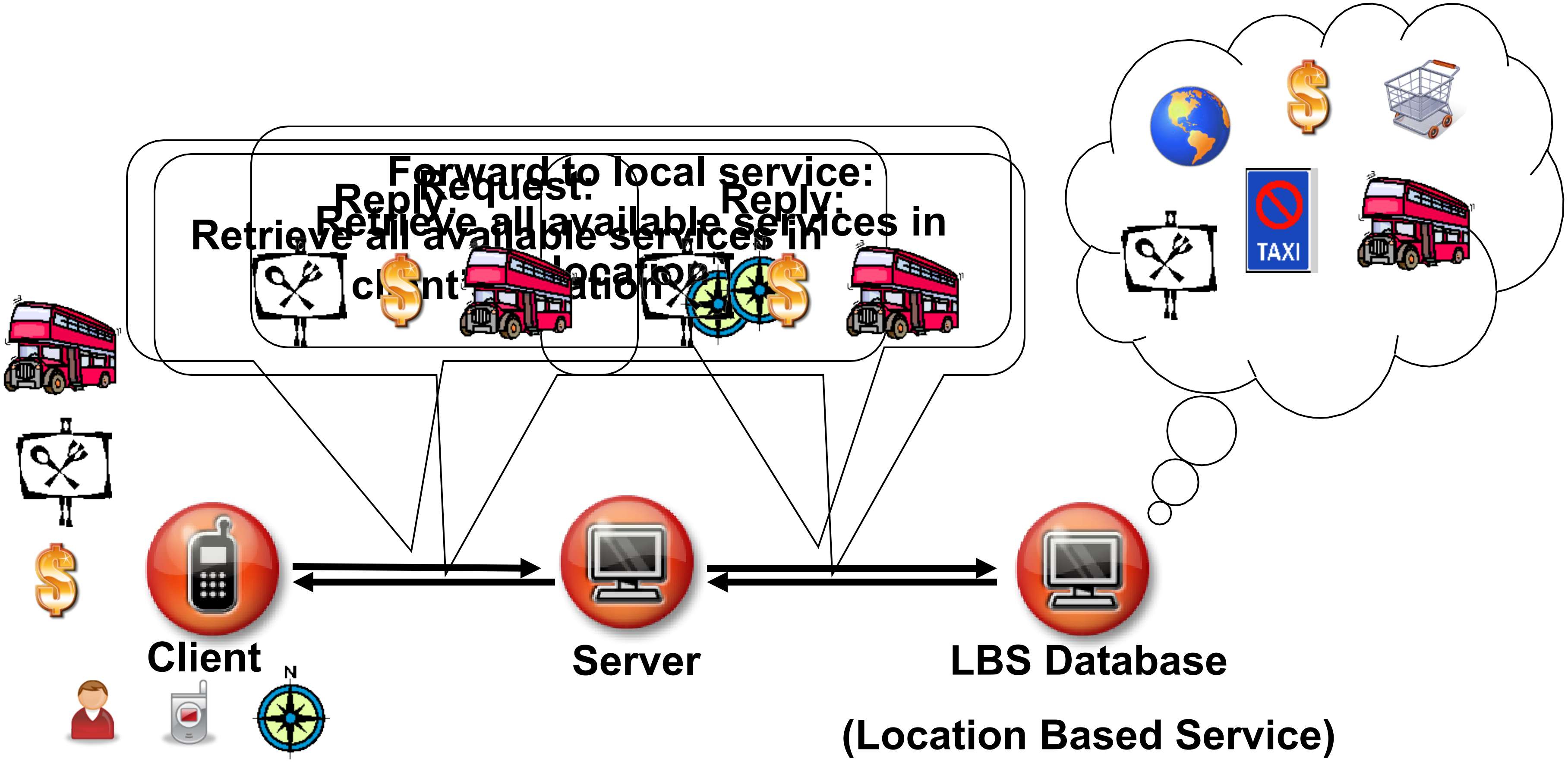
LBS: Now

- Location-based traffic reports:
  - ***Range query:*** How many cars in the free way
  - ***Shortest path query***: What is the estimated travel time to reach my destination

- **Location-based store finder:**
  - ***Range query:*** What are the restaurants within five miles of my location
  - ***Nearest-neighbor query***: Where is my nearest fast (junk) food restaurant

- **Location-based advertisement:**
  - ***Range query:*** Send E-coupons to all customers within five miles of my store

The LBS Workflow



Request:
Retrieve all available services in client's location

Forward to local service:
Retrieve all available services in location

Reply:

Reply:

Client

Server

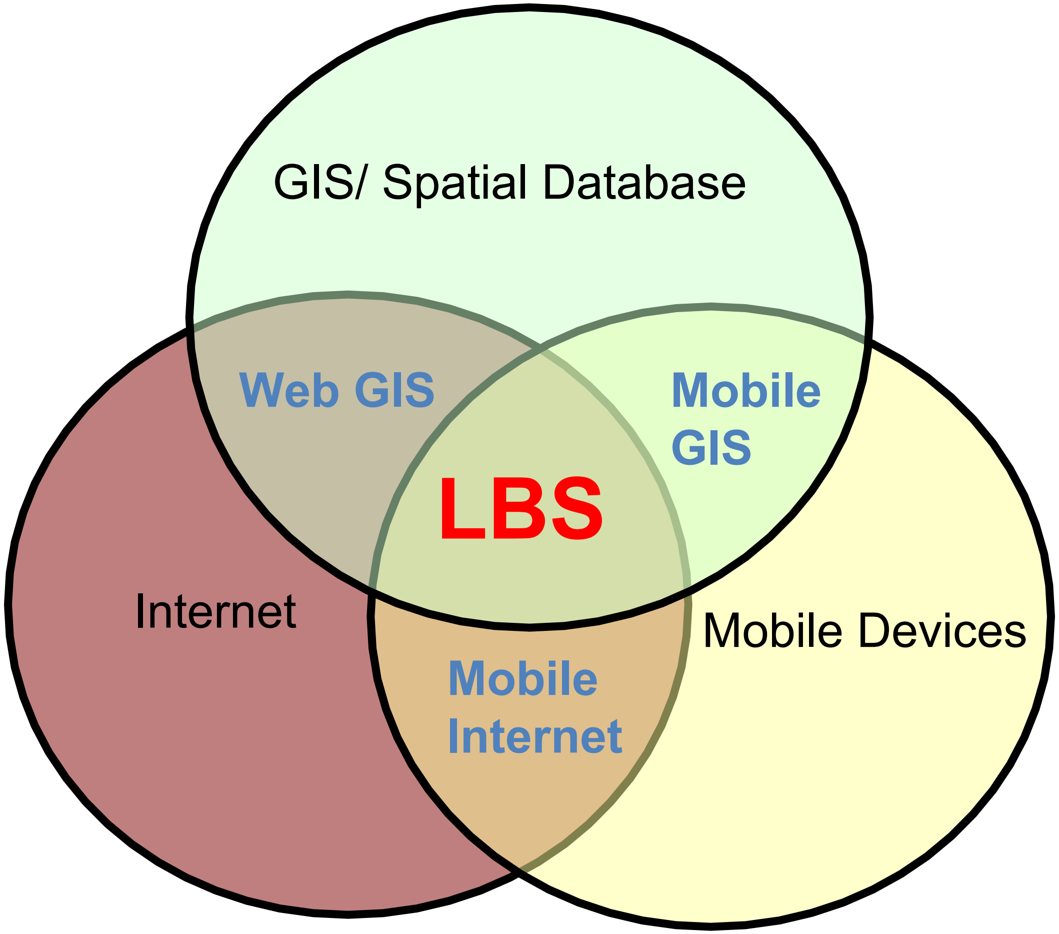
LBS Database

(Location Based Service)

LBS: Why Now ?

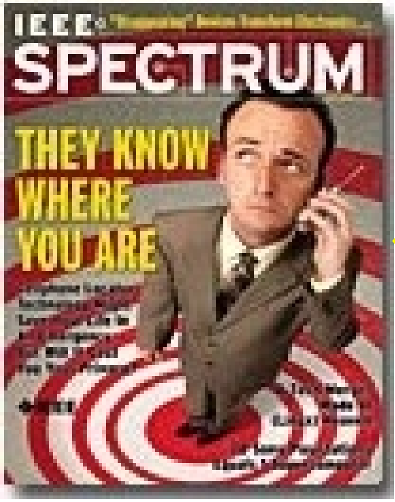
Keep left on High Street
Maps on WAP phones and PDAs
finanzen.net

LBS: Why Now ?

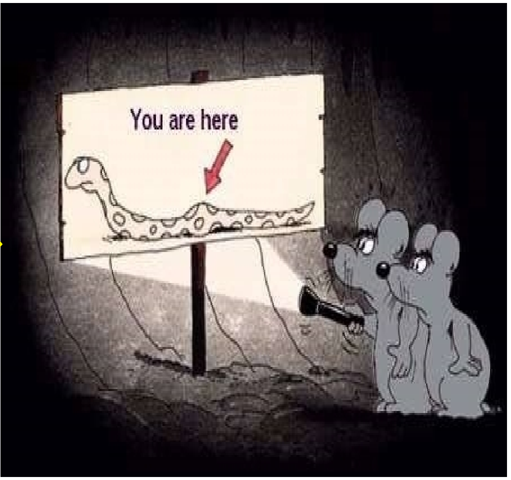


LBS is a convergence of technologies

Major Privacy Threats



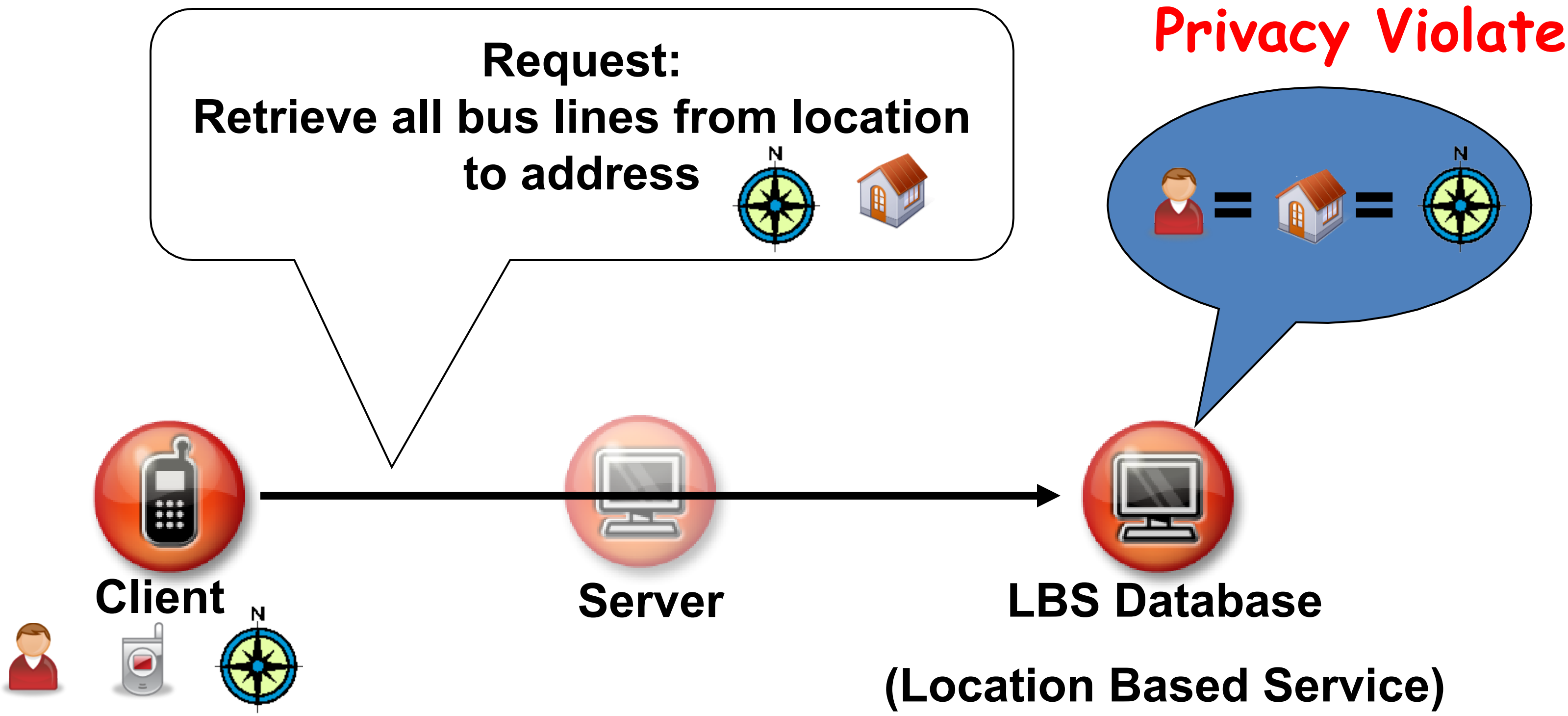
YOU ARE TRACKED…!!!!



"New technologies can pinpoint your location at any time and place. They promise safety and convenience but threaten privacy and security"

Cover story, IEEE Spectrum, July 2003

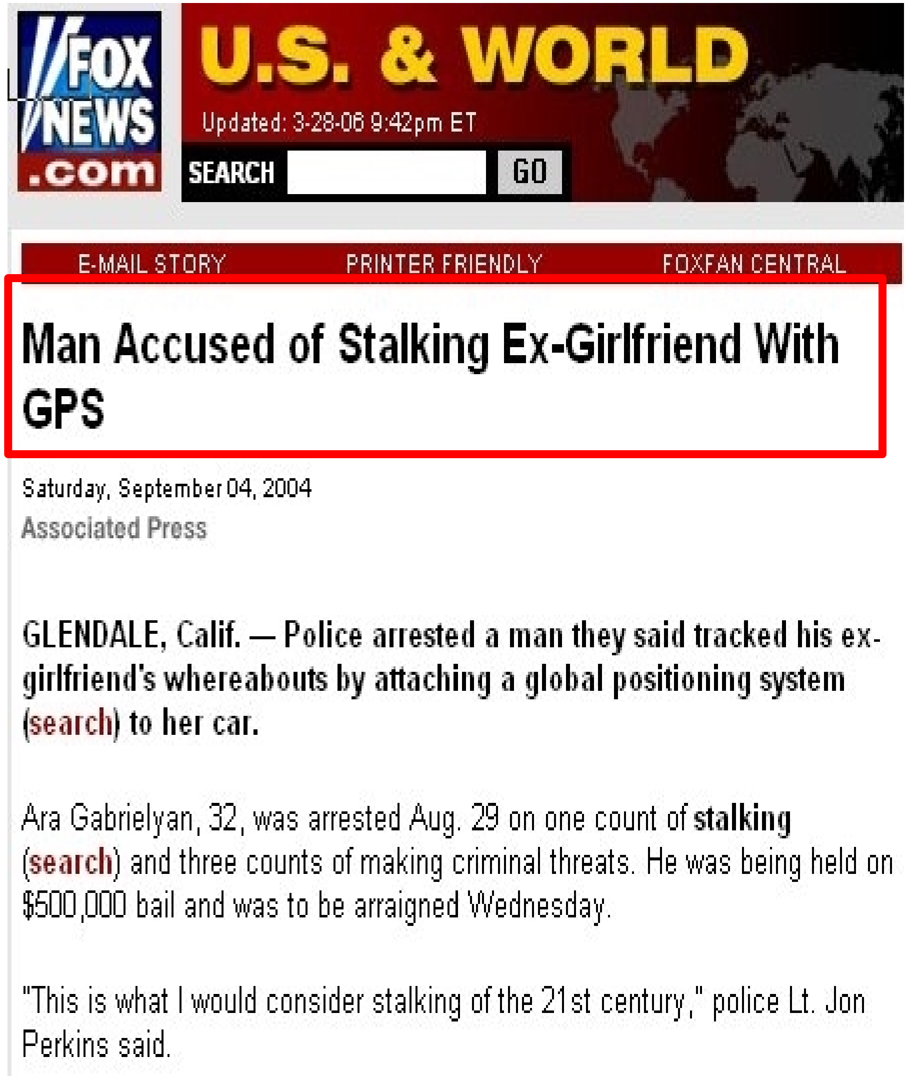
The Location Anonymity Problem



Request:
Retrieve all bus lines from location to address

Privacy Violated

Major Privacy Threats



Updated: 3-28-06 9:42pm ET

Man Accused of Stalking Ex-Girlfriend With GPS

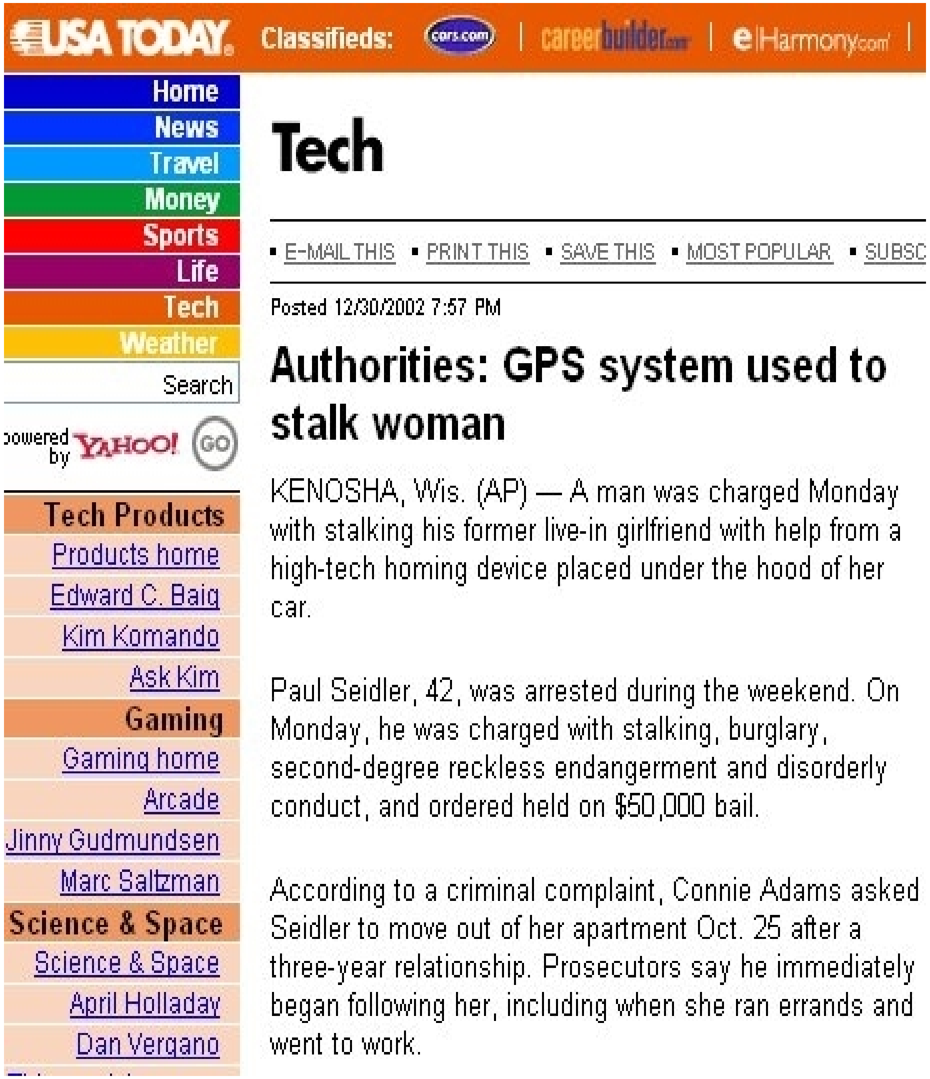
Saturday, September 04, 2004
Associated Press

GLENDALE, Calif. — Police arrested a man they said tracked his ex-girlfriend's whereabouts by attaching a global positioning system (search) to her car.

Ara Gabrielyan, 32, was arrested Aug. 29 on one count of **stalking** (search) and three counts of making criminal threats. He was being held on $500,000 bail and was to be arraigned Wednesday.

"This is what I would consider stalking of the 21st century," police Lt. Jon Perkins said.

http://www.foxnews.com/story/0,2933,131487,00.html



Posted 12/30/2002 7:57 PM

Authorities: GPS system used to stalk woman

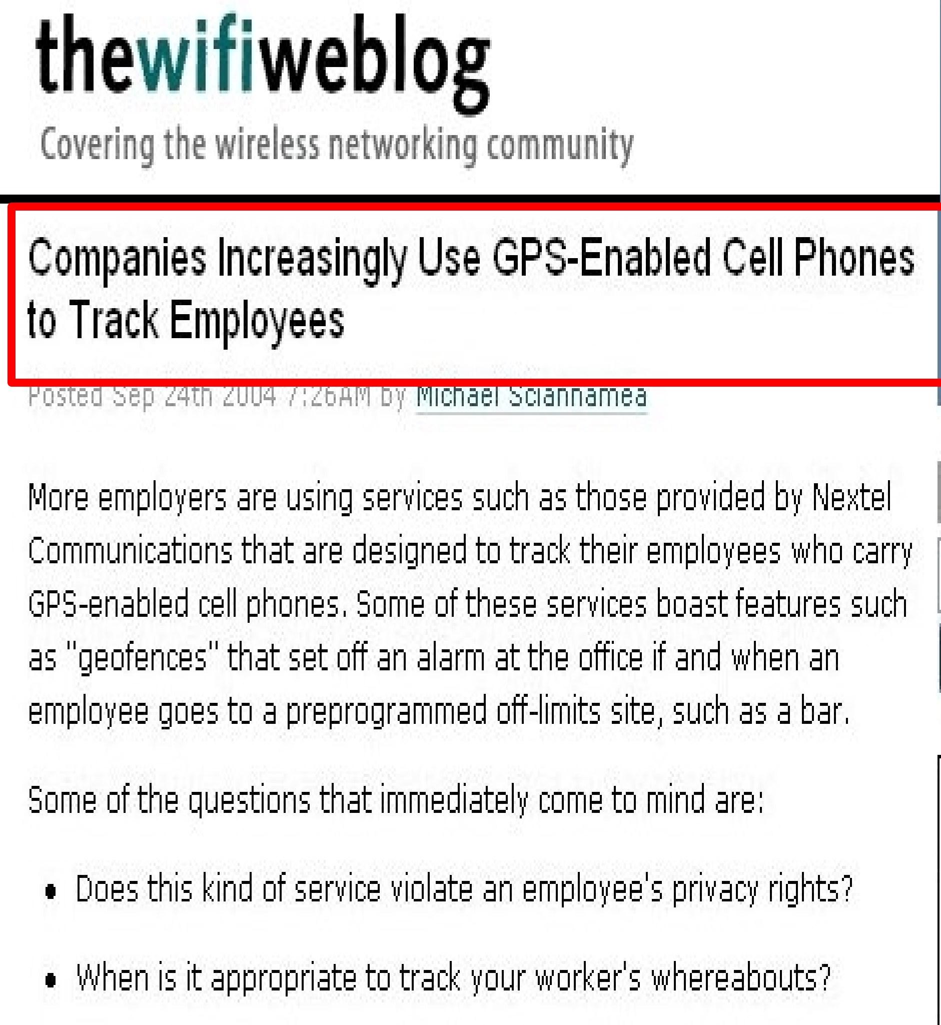
KENOSHA, Wis. (AP) — A man was charged Monday with stalking his former live-in girlfriend with help from a high-tech homing device placed under the hood of her car.

Paul Seidler, 42, was arrested during the weekend. On Monday, he was charged with stalking, burglary, second-degree reckless endangerment and disorderly conduct, and ordered held on $50,000 bail.

According to a criminal complaint, Connie Adams asked Seidler to move out of her apartment Oct. 25 after a three-year relationship. Prosecutors say he immediately began following her, including when she ran errands and went to work.

http://www.usatoday.com/tech/news/2002-12-30-gps-stalker_x.htm

Major Privacy Threats



thewifiweblog

Covering the wireless networking community

Companies Increasingly Use GPS-Enabled Cell Phones to Track Employees

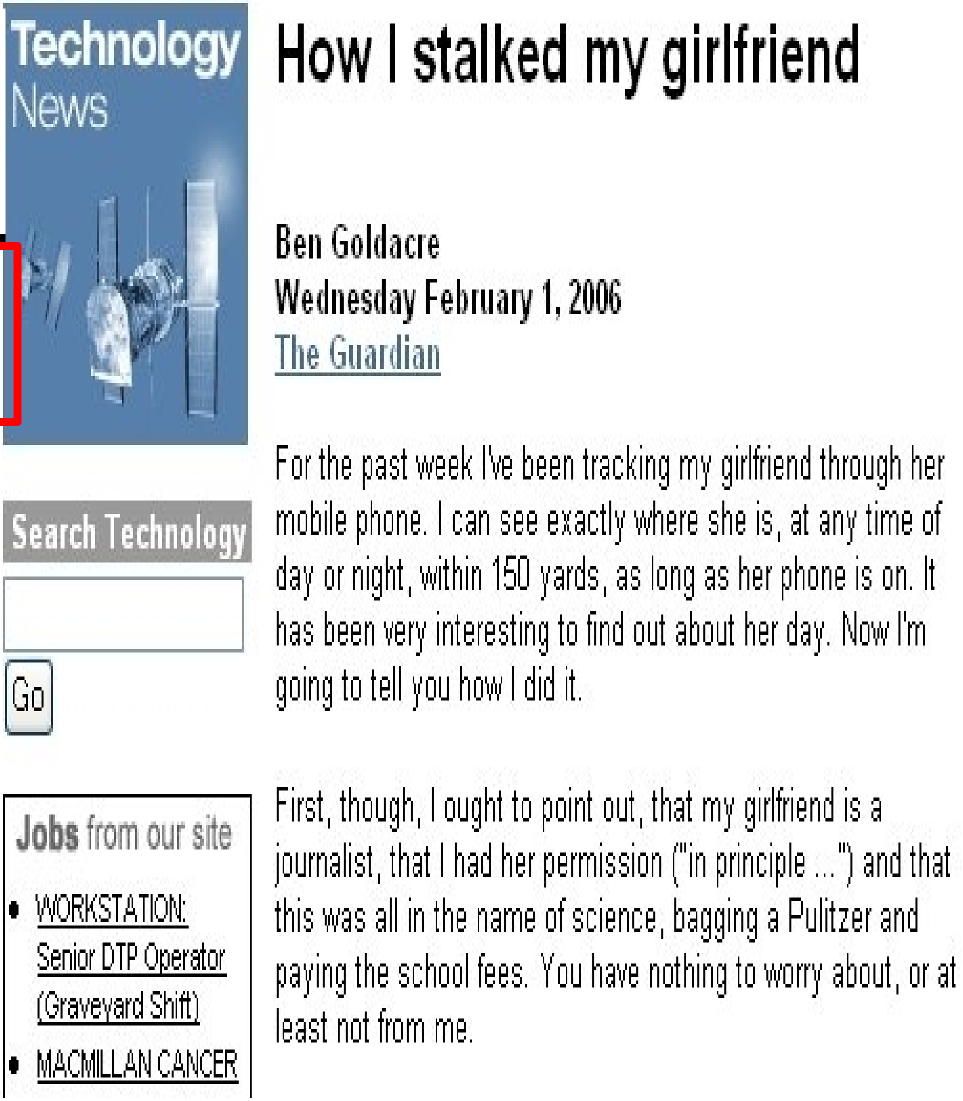
Posted Sep 24th 2004 7:26AM by Michael Sciannamea

More employers are using services such as those provided by Nextel Communications that are designed to track their employees who carry GPS-enabled cell phones. Some of these services boast features such as "geofences" that set off an alarm at the office if and when an employee goes to a preprogrammed off-limits site, such as a bar.

Some of the questions that immediately come to mind are:

- Does this kind of service violate an employee's privacy rights?
- When is it appropriate to track your worker's whereabouts?

http://wifi.weblogsinc.com/2004/09/24/companies-increasingly-use-gps-enabled-cell-phones-to-track/



How I stalked my girlfriend

Ben Goldacre
Wednesday February 1, 2006
The Guardian

For the past week I've been tracking my girlfriend through her mobile phone. I can see exactly where she is, at any time of day or night, within 150 yards, as long as her phone is on. It has been very interesting to find out about her day. Now I'm going to tell you how I did it.

First, though, I ought to point out, that my girlfriend is a journalist, that I had her permission ("in principle ...") and that this was all in the name of science, bagging a Pulitzer and paying the school fees. You have nothing to worry about, or at least not from me.

http://technology.guardian.co.uk/news/story/0,,1699156,00.html

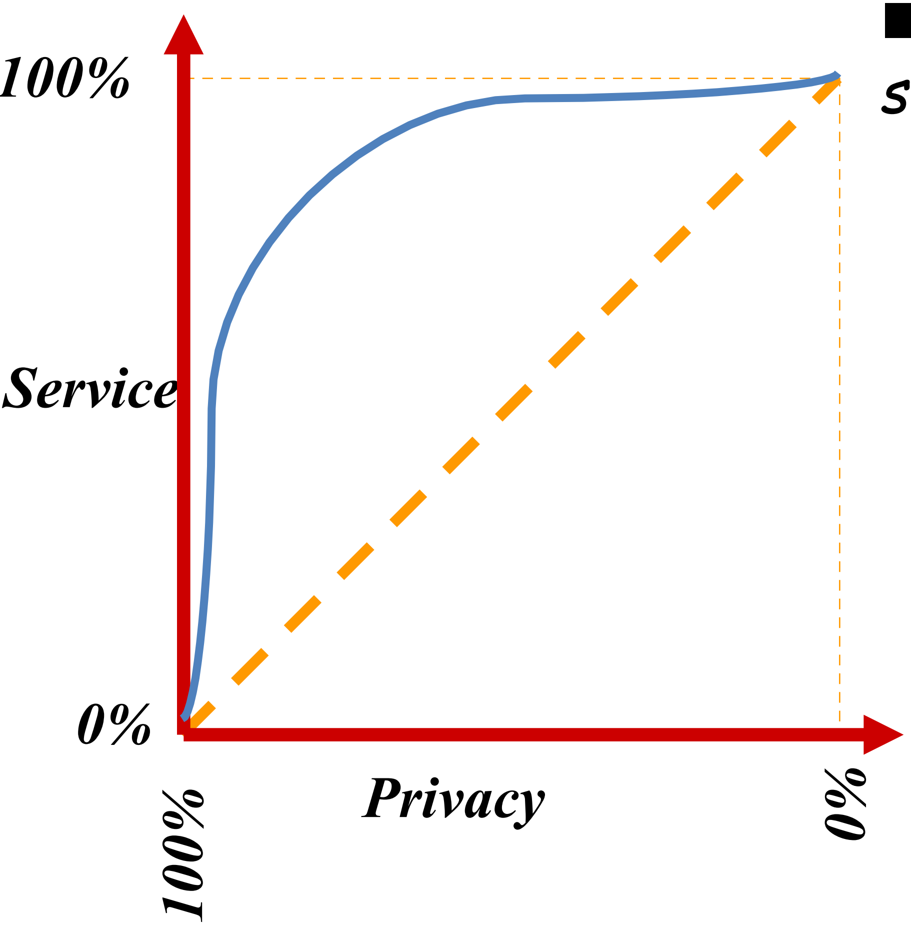
Service-Privacy Trade-off

- First extreme:
  - A user reports her exact location ➔ 100% service

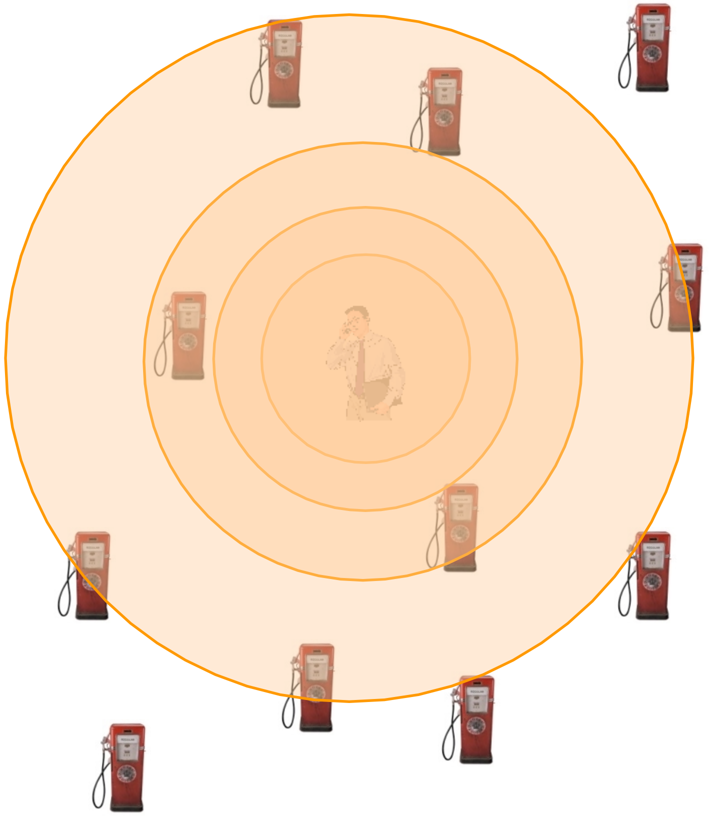
- Second extreme:
  - A user does NOT report her location ➔ 0% service

Desired Trade-off: A user reports a perturbed version of her location ➔ *x*% service

Service-Privacy Trade-off



- Example:: *What is my nearest gas station*



What is Special About Location Privacy

•Queries need to be private (e.g., location -based queries)

•Should tolerate the high frequency of location updates

Continuous location exposure a serious threat to privacy

•Privacy requirements are personalized

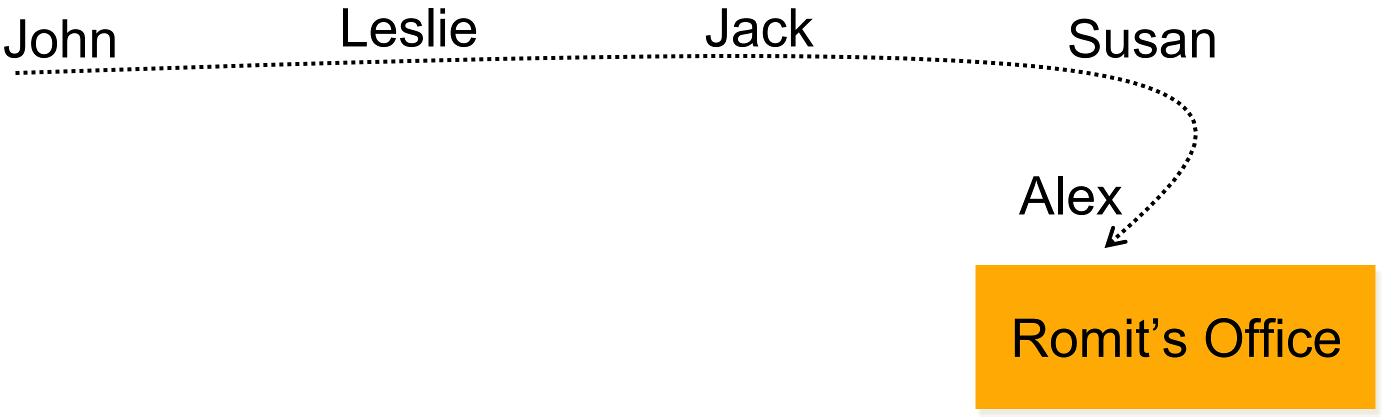
Discussion: Solutions?

Entertain location based services
without
revealing their private location information

Just Call Yourself ` `Freddy”

Pseudonymns [Gruteser04]

- Effective only when infrequent location exposure
- Else, spatio-temporal patterns enough to de-anonymize

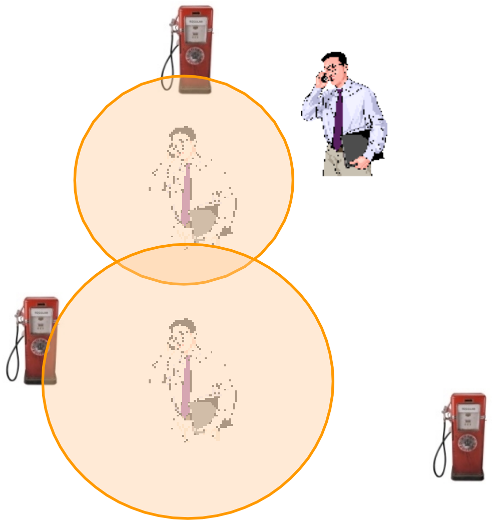


Location Perturbation

- The user location is represented with a wrong value

- The privacy is achieved from the fact that the reported location is false

- The accuracy and the amount of privacy mainly depends on how far the reported location form the exact location



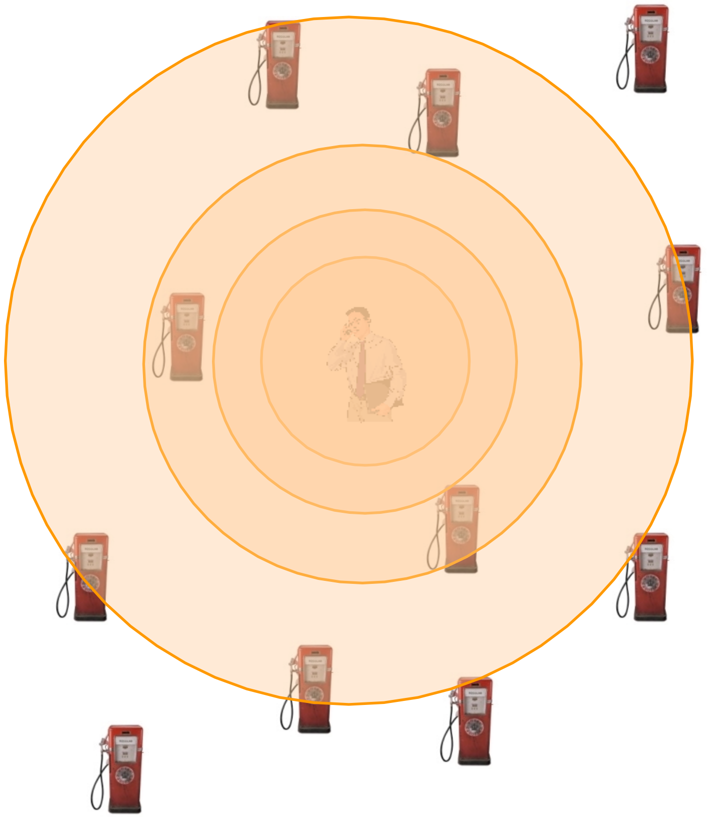
Spatial Cloaking

- Location *cloaking*, location *blurring*, location *obfuscation*

- The user exact location is represented as a region that includes the exact user location

- An adversary does know that the user is located in the *cloaked* region, but has no clue where the user is exactly located

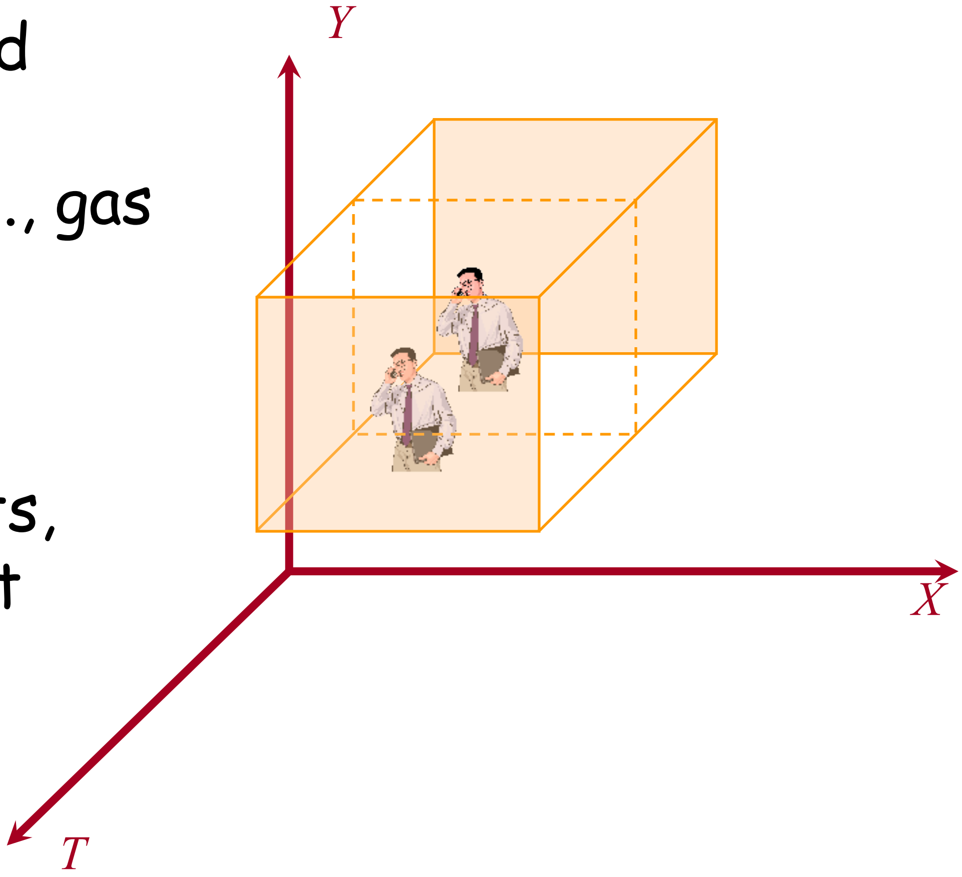
- The area of the *cloaked* region achieves a trade-off between the user privacy and the service



Temporal Cloaking

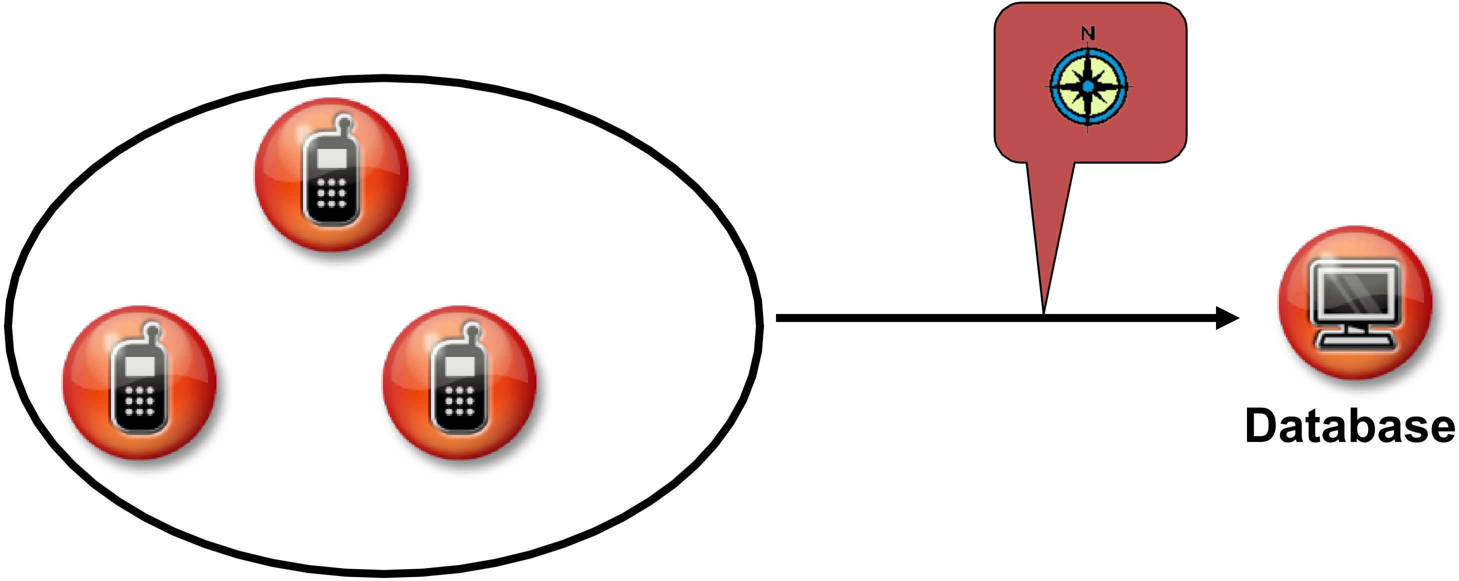
- Temporal cloaking could tolerate asking about stationary objects (e.g., gas stations)

- Challenging to support querying moving objects, e.g., what is my nearest police car

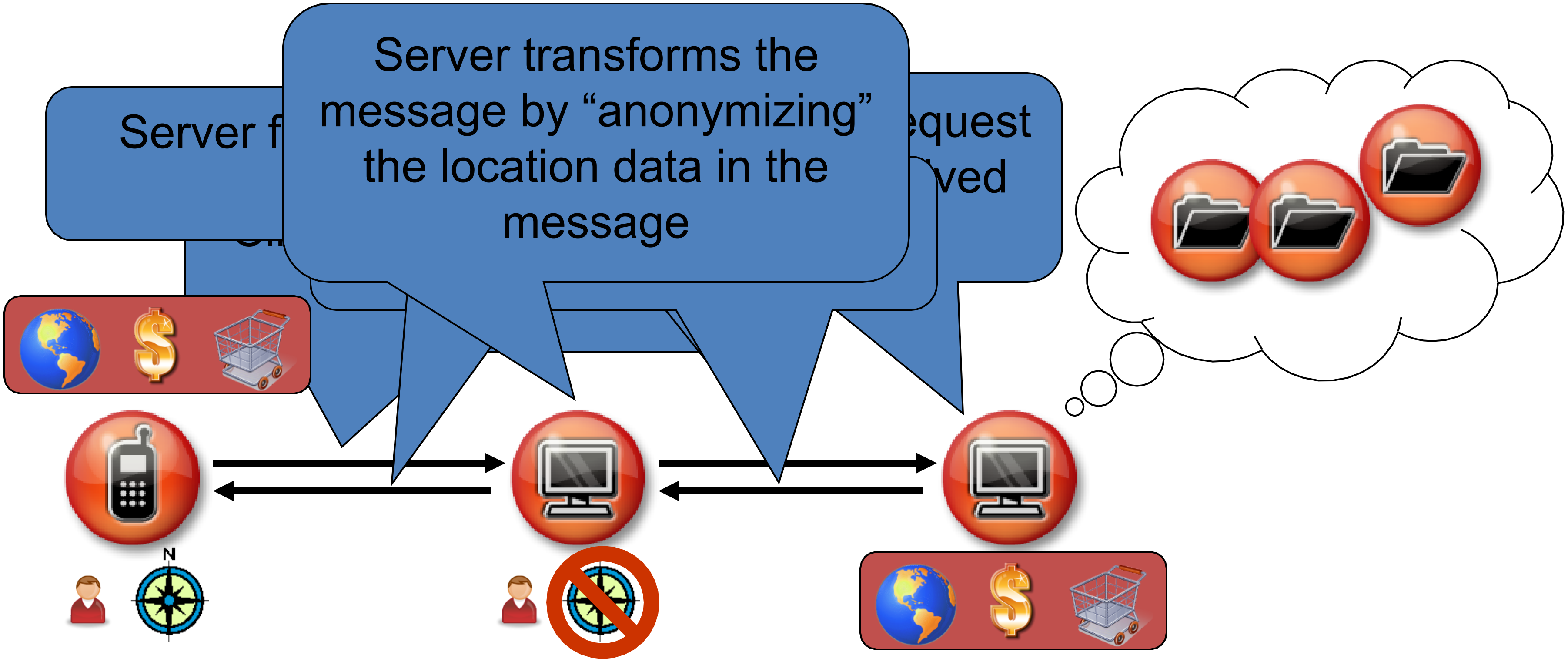


Location Anonymity

*"A message from a **client** to a **database** is called **<u>location anonymous</u>** if the **client's identity** cannot be distinguished from other users based on the **client's** location information."*



Implementation of Location Anonymity



Server transforms the message by "anonymizing" the location data in the message

k-Anonymity

*"A message from a **client** to a **database** is called **location k-anonymous** if the **client** cannot be identified by the **database** based on the **client's** location from other k-1 **clients**."*

k-Anonymity

- The *cloaked* region contains at least *k* users

- The user is indistinguishable among other *k* users

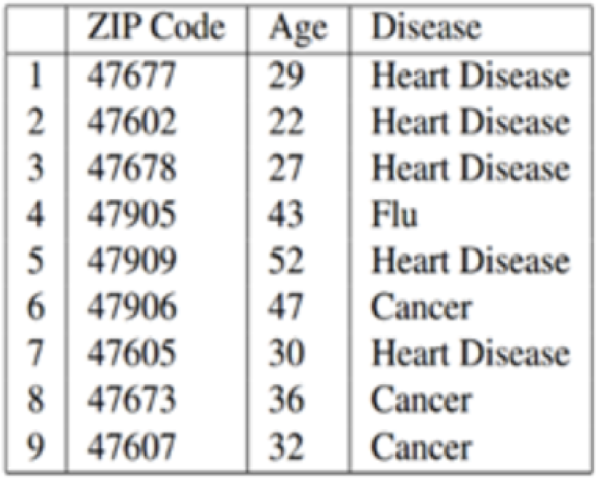
- The cloaked area largely depends on the surrounding environment.

- A value of *k* =100 may result in a very small area if a user is located in the stadium or may result in a very large area if the user in the desert.

10-anonymity

k-Anonymity for data privacy protection

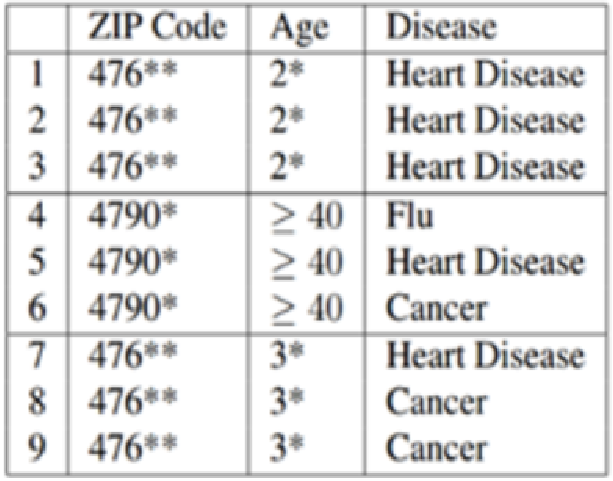
Each data tuple is cloaked by other K-1 tuple, such that it is not distinguishable from the k tuples



	ZIP Code	Age	Disease
1	47677	29	Heart Disease
2	47602	22	Heart Disease
3	47678	27	Heart Disease
4	47905	43	Flu
5	47909	52	Heart Disease
6	47906	47	Cancer
7	47605	30	Heart Disease
8	47673	36	Cancer
9	47607	32	Cancer

Table 1. Original Patients Table

After 3-Anonymity

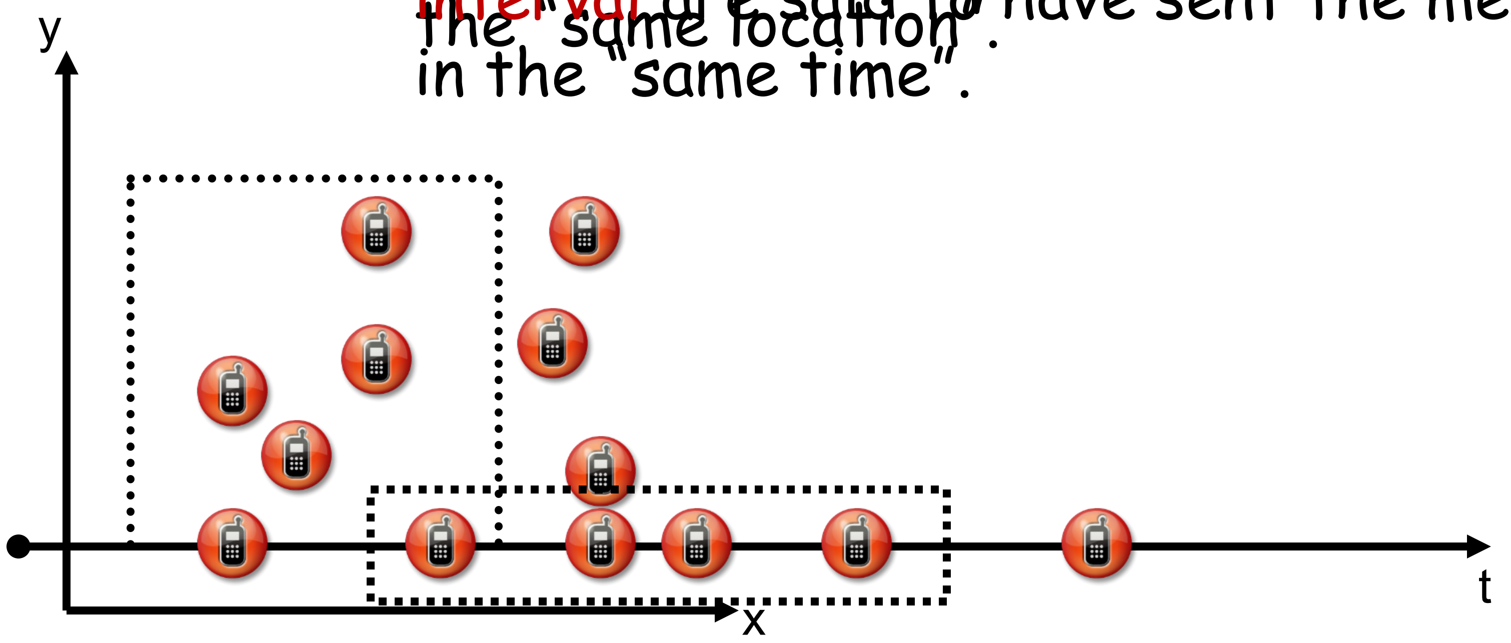


	ZIP Code	Age	Disease
1	476**	2*	Heart Disease
2	476**	2*	Heart Disease
3	476**	2*	Heart Disease
4	4790*	≥ 40	Flu
5	4790*	≥ 40	Heart Disease
6	4790*	≥ 40	Cancer
7	476**	3*	Heart Disease
8	476**	3*	Cancer
9	476**	3*	Cancer

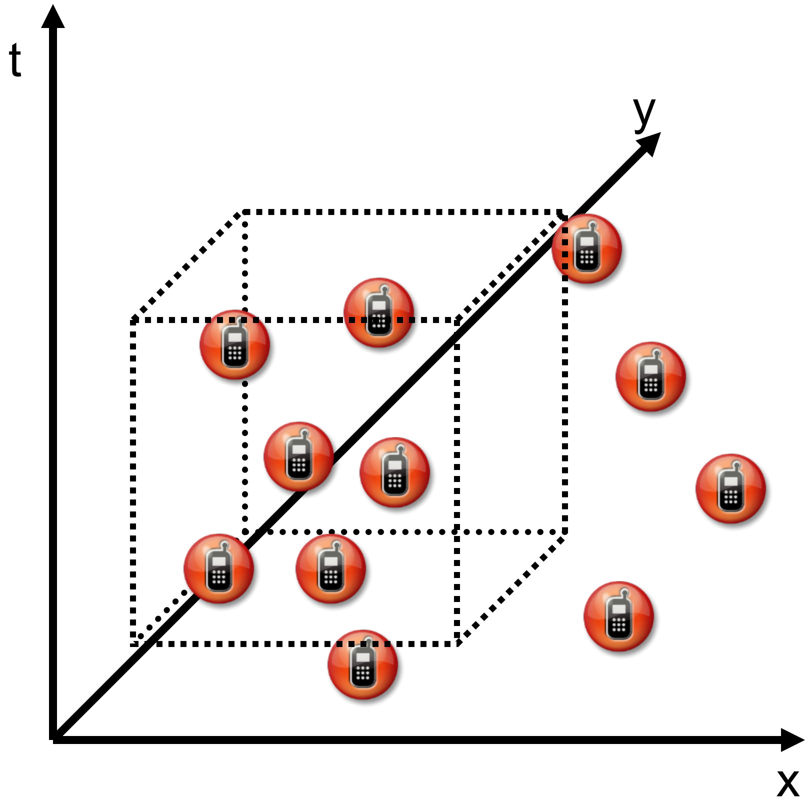
Implementation of Location k-Anonymity

Spatial Cloaking – Setting a range of space to be a single box, where all clients located within the range are said to be in the "same location".

Temporal Cloaking – Setting a time interval, where all the clients in a specific location sending a message in that time interval are said to have sent the message in the "same time".



Spatial-Temporal Cloaking



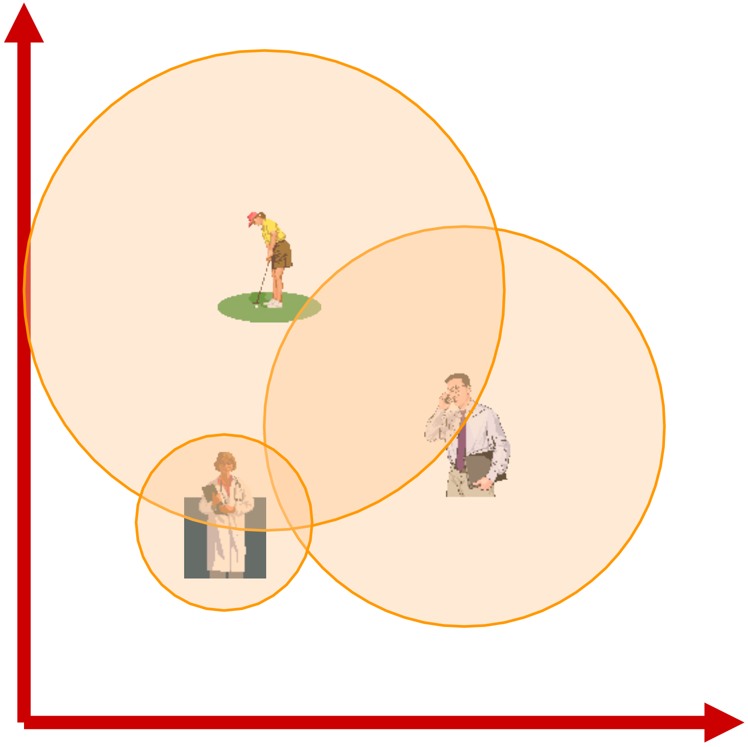
- Setting a range of space and a time interval, where all the messages sent by client inside the range in that time interval. This spatial and temporal area is called a "cloaking box".

Summary

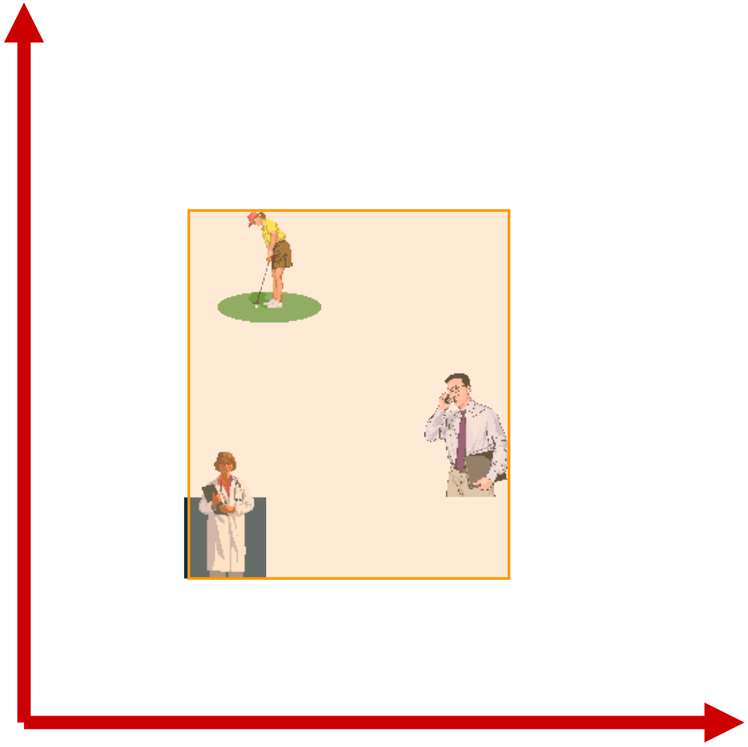
- Secure localization and synchronization are still open research areas, especially in some aspects of BSN, VANET,etc.

- K-Anoymity is a practical solution for LBS services

Data-Dependent Cloaking



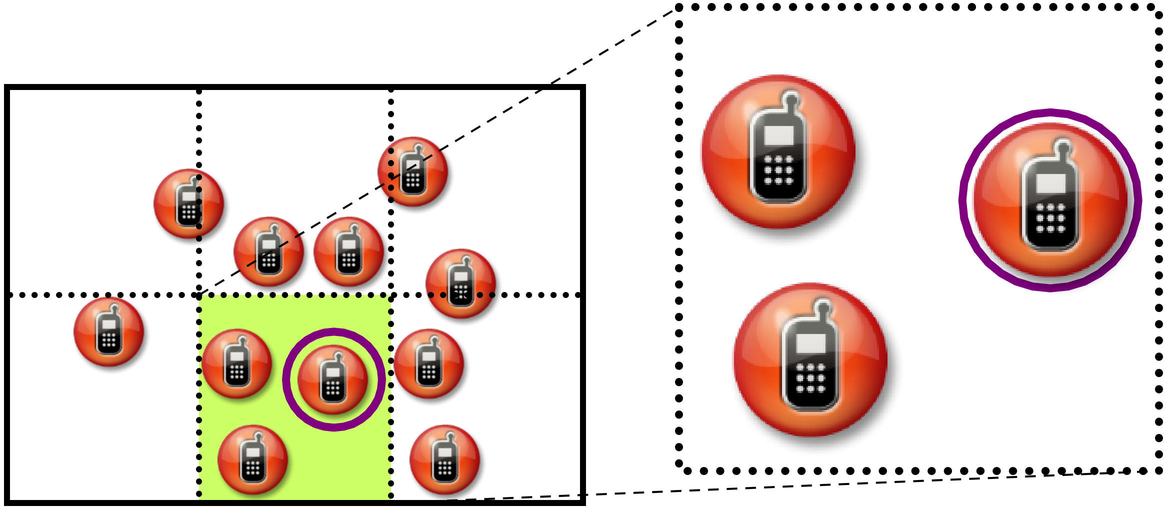
Naïve cloaking



MBR (minimum bounding rectangle) cloaking

Previous solutions: MBR

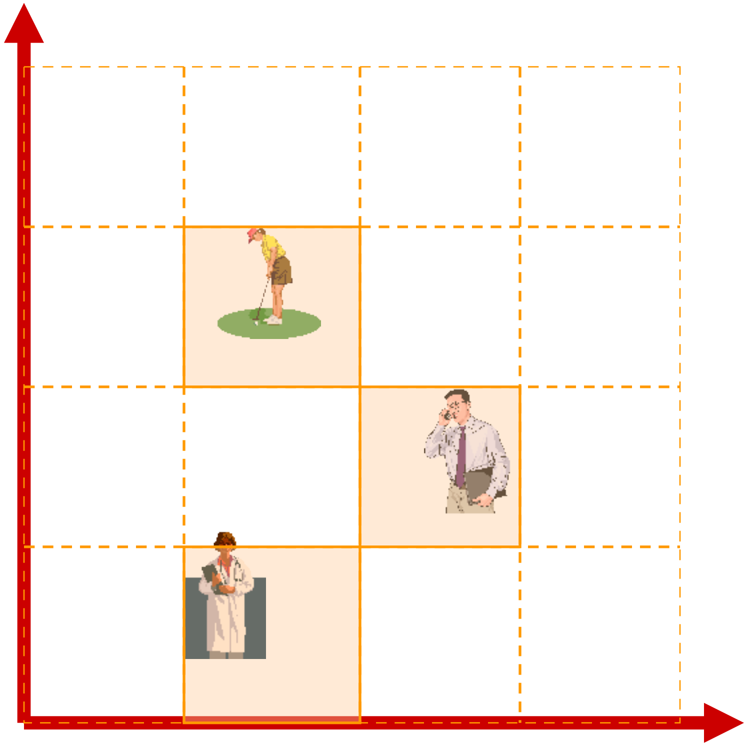
M. Gruteser, D Grunwald (2003) – For a fixed *k* value, the server finds the smallest area around the client's location that potentially contains *k-1* different other clients, and monitoring that area over time until such *k-1* clients are found.



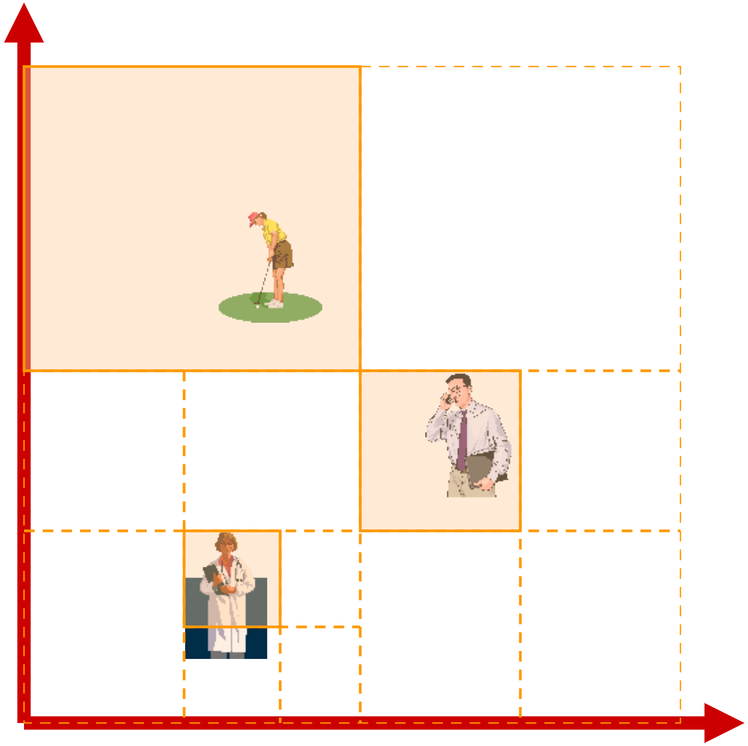
Drawback:

Fixed anonymity value for all clients (service dependent)

Space-Dependent Cloaking



Fixed grid cloaking



Adaptive grid cloaking

Privacy Profile

- Each mobile user will have her own *privacy-profile* that includes:
  - *K*. A user wants to be *k*-anonymous
  - A_{min}. The minimum required area of the blurred area
  - A_{max}. The maximum required area of the blurred area
  - Multiple instances of the above parameters to indicate different privacy profiles at different times

Time	*k*	A_{min}	A_{max}
8:00 AM -	1	—	—
5:00 PM -	100	1 mile	3 miles
10:00 PM -	1000	5 miles	—

Query Types

- *Private* Queries over *Public* Data
  - *What is my nearest gas station*
  - The user location is private while the objects of interest are public

- *Public* Queries over *Private* Data
  - *How many cars in the downtown area*
  - The query location is public while the objects of interest is private

- *Private* Queries over *Private* Data
  - *Where is my nearest friend*
  - Both the query location and objects of interest are private

Modes of Privacy

- User Location Privacy
  - Users want to hide their location information and their query information
- User Query Privacy
  - Users do not mind or obligated to reveal their locations, however, users want to hide their queries
- Trajectory Privacy
  - Users do not mind to reveal few locations, however, they want to avoid linking these locations together to form a trajectory

Requirements of the Location Anonymization Process

- Accuracy.
  - The anonymization process should satisfy and be as close as possible to the user requirements (expressed as privacy profile)
- Quality.
  - An adversary cannot infer any information about the exact user location from the reported location
- Efficiency.
  - Calculating the anonymized location should be computationally efficient and scalable
- Flexibility.
  - Each user has the ability to change her privacy profile at any time

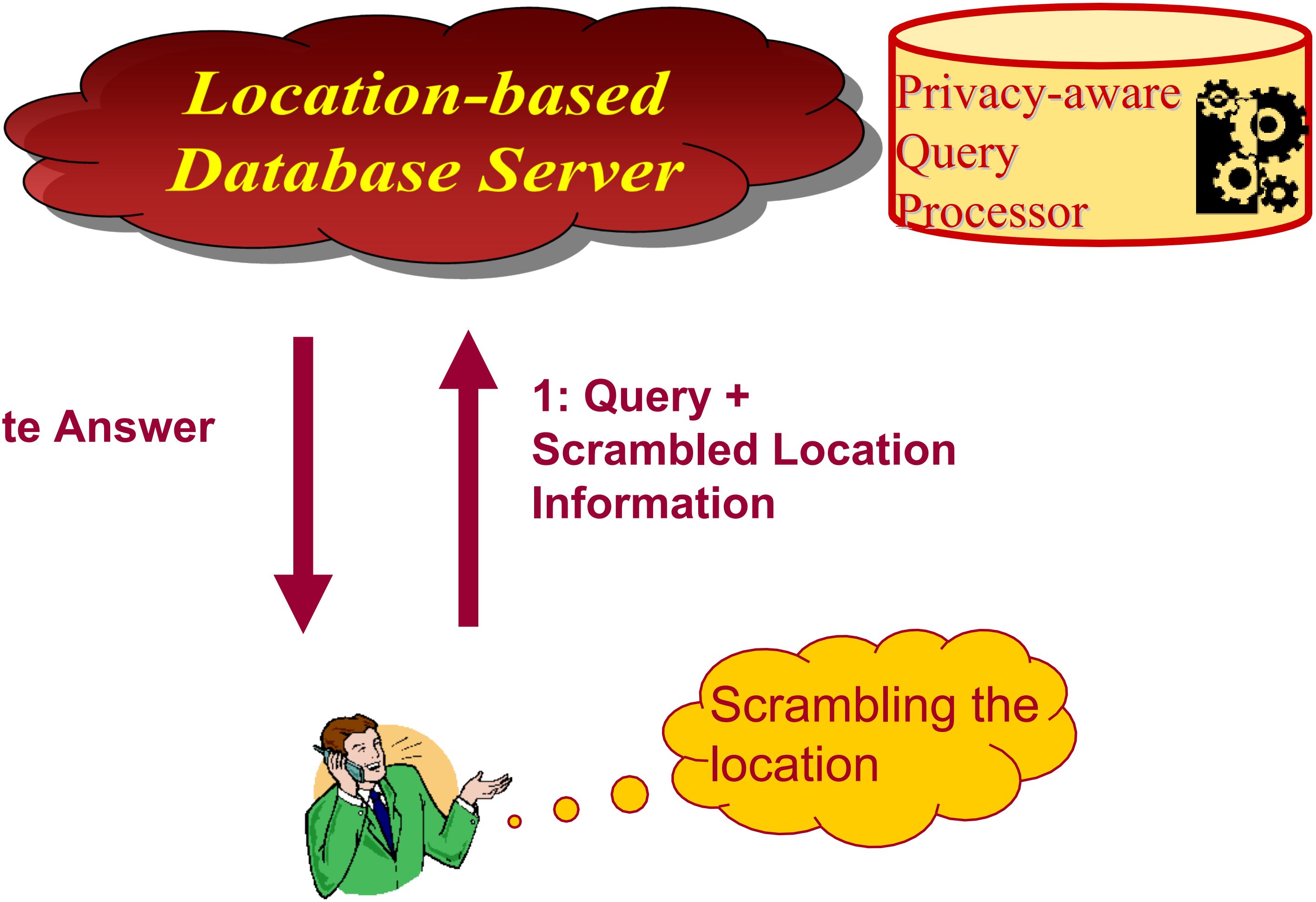
System Architectures for Preserving Location Privacy

- Client-Server Architecture
- Third Trusted Party Architecture
- Peer-to-peer Architecture

System Architectures

- *Client-Server architecture*
  - Users communicated directly with the sever to do the anonymization process. Possibly employing an offline phase with a trusted entity
- *Third trusted party architecture*
  - A centralized trusted entity is responsible for gathering information and providing the required privacy for each user
- *Peer-to-Peer cooperative architecture*
  - Users collaborate with each other without the interleaving of a centralized entity to provide customized privacy for each single user

Client-Server Architecture



Location-based Database Server

Privacy-aware Query Processor

2: Candidate Answer

1: Query + Scrambled Location Information

Scrambling the location

Client-Server Architecture

- Clients try to *cheat* the server using either fake locations or fake space

- Simple to implement, easy to integrate with existing technologies

- Lower quality of service

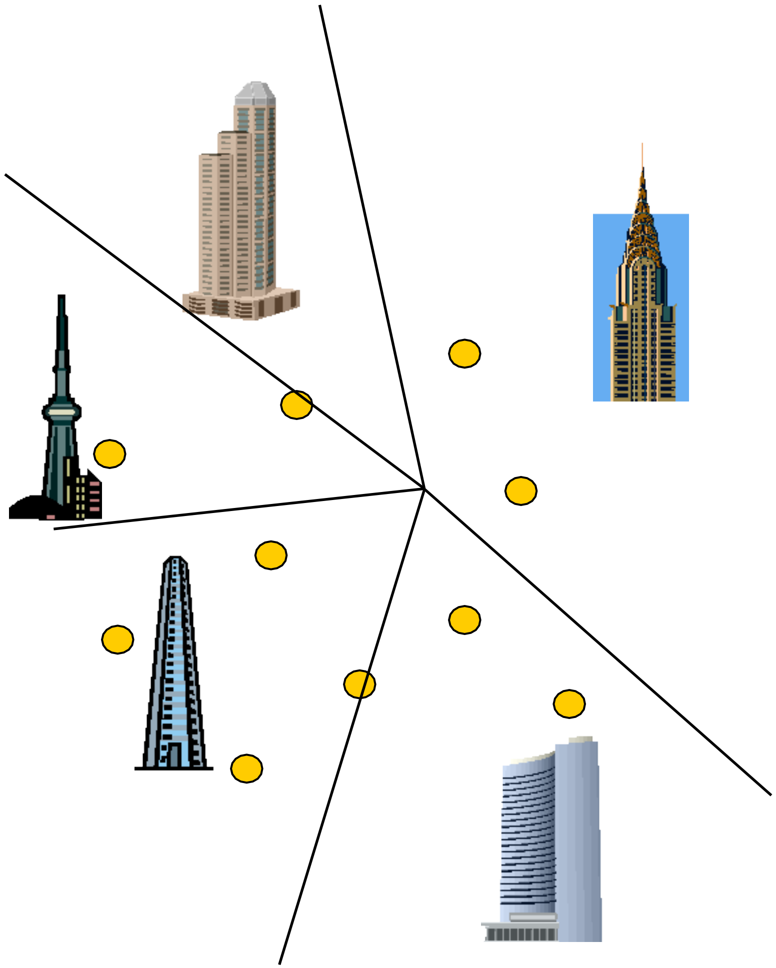
- *Examples*: Landmark objects, false dummies, and space transformation

Client-Server Architecture:
Landmark objects

- Instead of reporting the exact location, report the location of a closest landmark

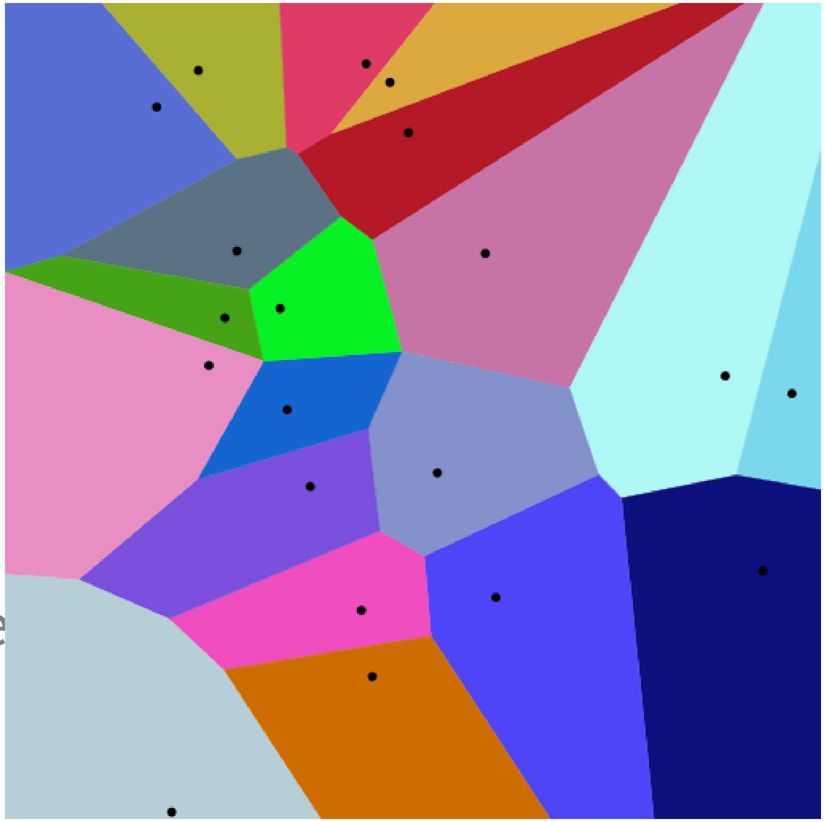
- The query answer will be based on the landmark

- Voronoi diagrams can be used to identify the closest landmark



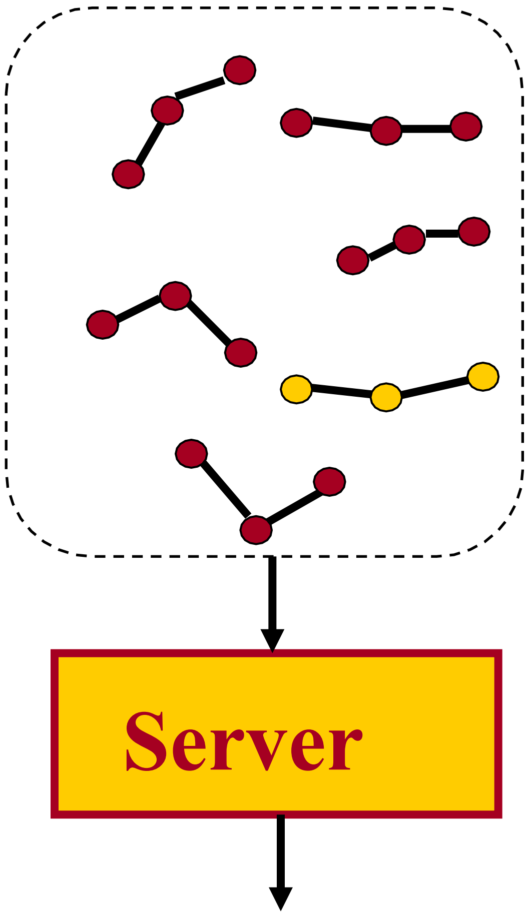
Voronoi diagrams

- **Voronoi diagram** is a partitioning of a plane into regions based on distance to points in a specific subset of the plane. That set of points (called seeds, sites, or generators) is specified beforehand, and for each seed there is a corresponding region consisting of all points closer to that seed than to any other. These regions are called Voronoi cells.



Client-Server Architecture: False Dummies

- A user sends *m* locations, only one of them is true while *m-1* are false dummies
- The server replies with a service for each received location
- The user is the only one who knows the true location, and hence the true answer
- Generating false dummies should follow a certain pattern similar to a user pattern but with different locations



Server

A separate answer for each received location

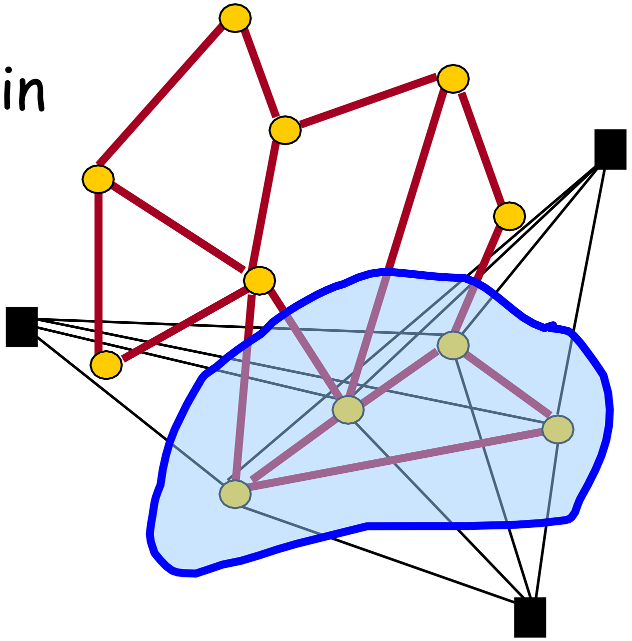
Client-Server Architecture: Location Obfuscation

- All locations are represented as vertices in a graph with edges correspond to the distance between each two vertices

- A user represents her location as an imprecise location (e.g., I am within the central park)

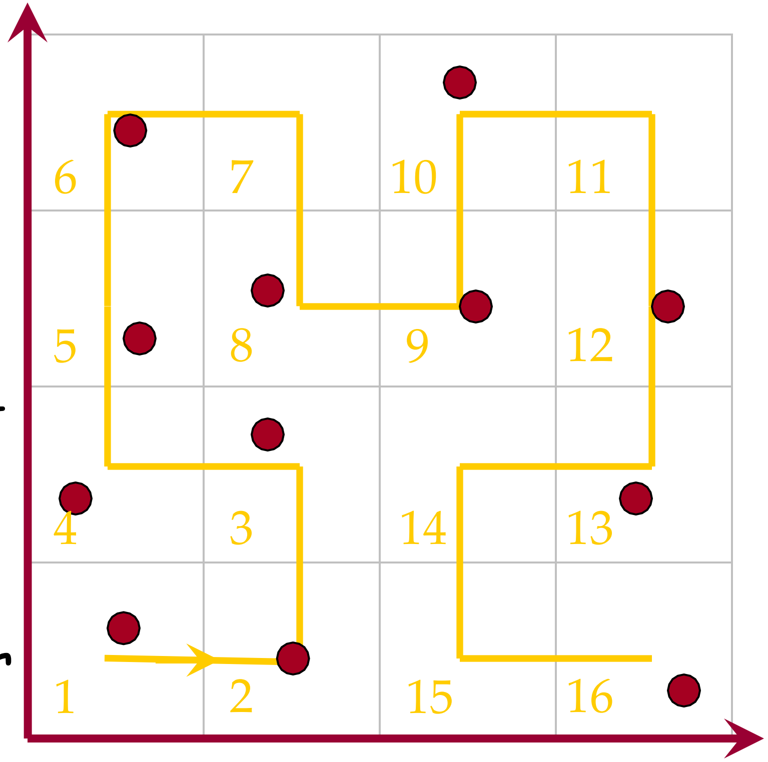
- The imprecise location is abstracted as a set of vertices

- The server evaluates the query based on the distance to each vertex of imprecise locations



Client-Server Architecture: Space Transformation

- Users transform their locations from the two-dimensional space to another space using a reversible transformation
- The new space does not have to have the same dimensionality as the original space.
- The database server answers location-based queries in the new space. This could result in an approximate answer
- The user apply a reverse transformation to transform the answer to the original space

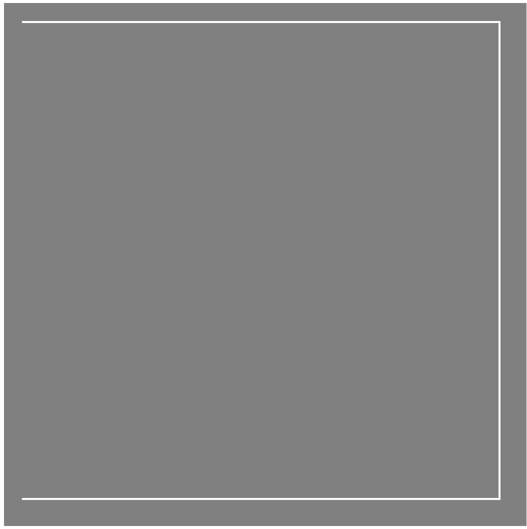


Hilbert curve

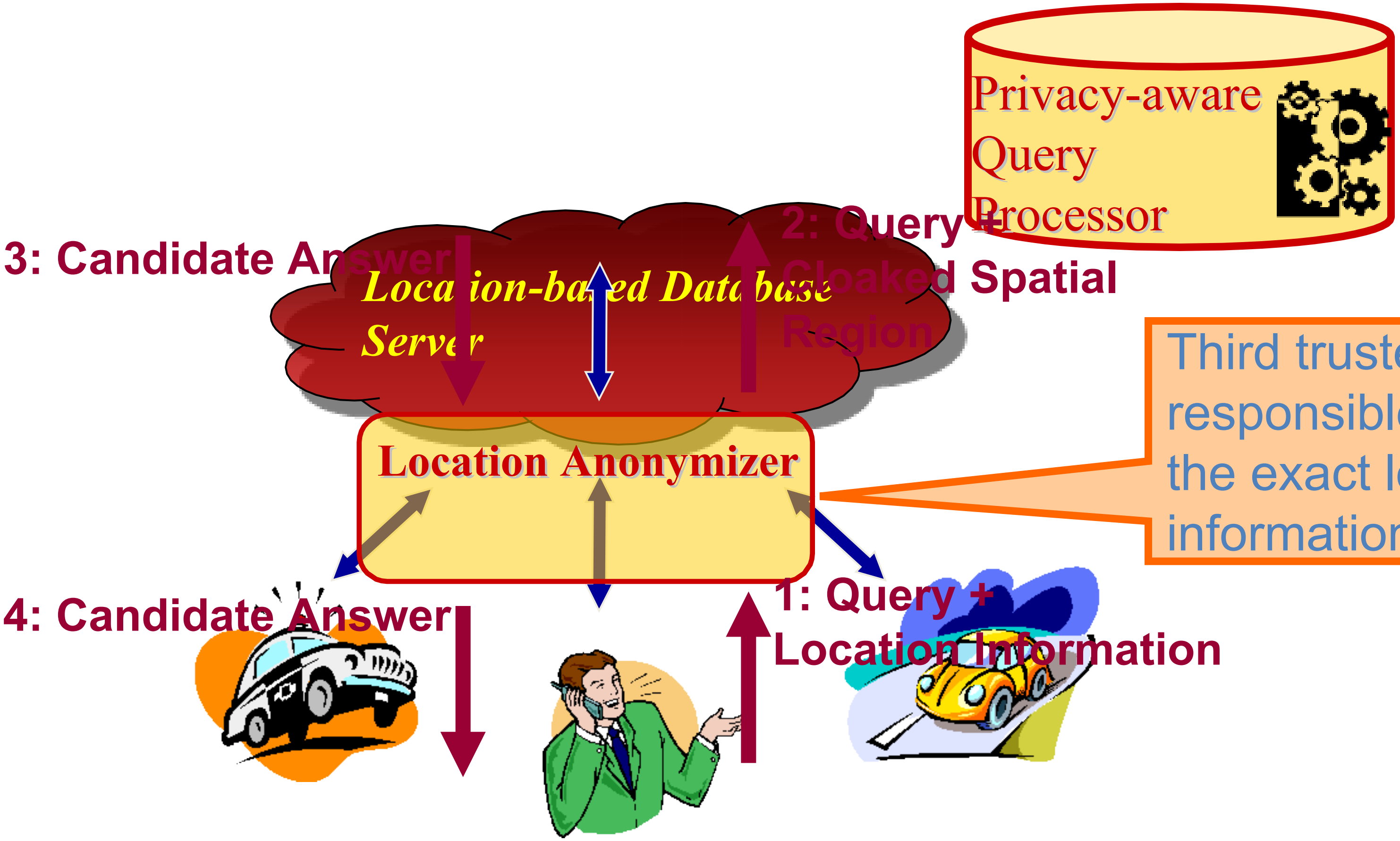
A continuous fractal space-filling curve

Hilbert curve and its discrete approximations are useful

- A mapping between 1D and 2D space that fairly well preserves locality.
- If (*x*,*y*) are the coordinates of a point within the unit square, and *d* is the distance along the curve when it reaches that point, then points that have nearby *d* values will also have nearby (*x*,*y*) values.



Third Trusted Party Architecture



Privacy-aware Query Processor

2: Query + Cloaked Spatial Region

3: Candidate Answer

Location-based Database Server

Location Anonymizer

4: Candidate Answer

1: Query + Location Information

Third trusted party that is responsible on *blurring* the exact location information.

Third Trusted Party Architecture

- A *trusted third party* receives the exact locations from clients, blurs the locations, and sends the blurred locations to the server
- Provide powerful privacy guarantees with high-quality services
- System bottleneck and sophisticated implementations

Third Trusted Party Architecture: Mix Zones

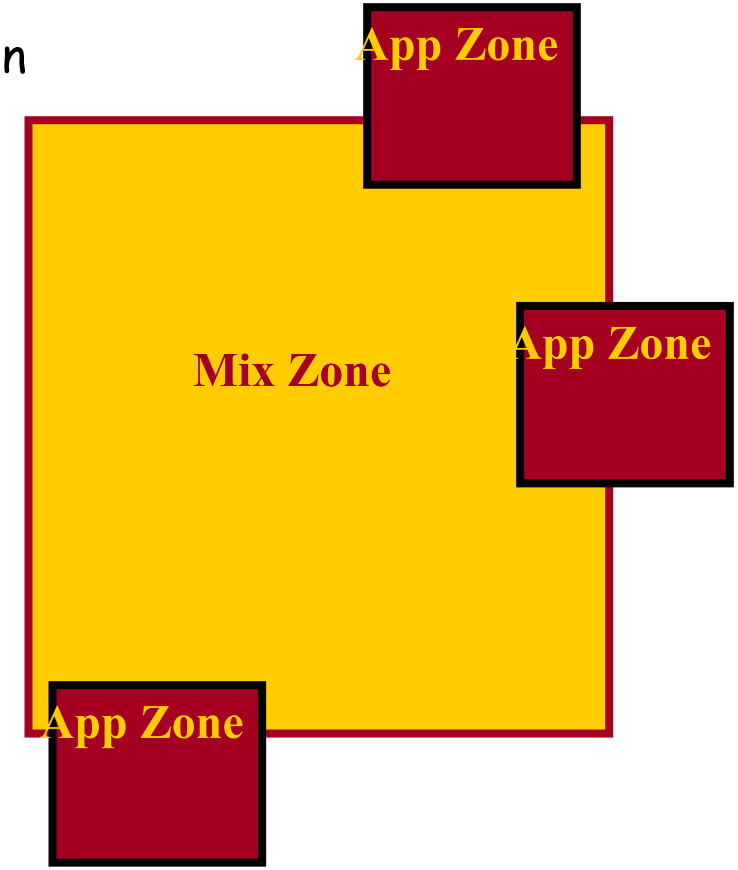
Application Zone: applications register interest in a geographic space with the middleware.

example spaces include hospital grounds, university buildings or a supermarket complex.

Users register interest in a particular set of location-aware applications.

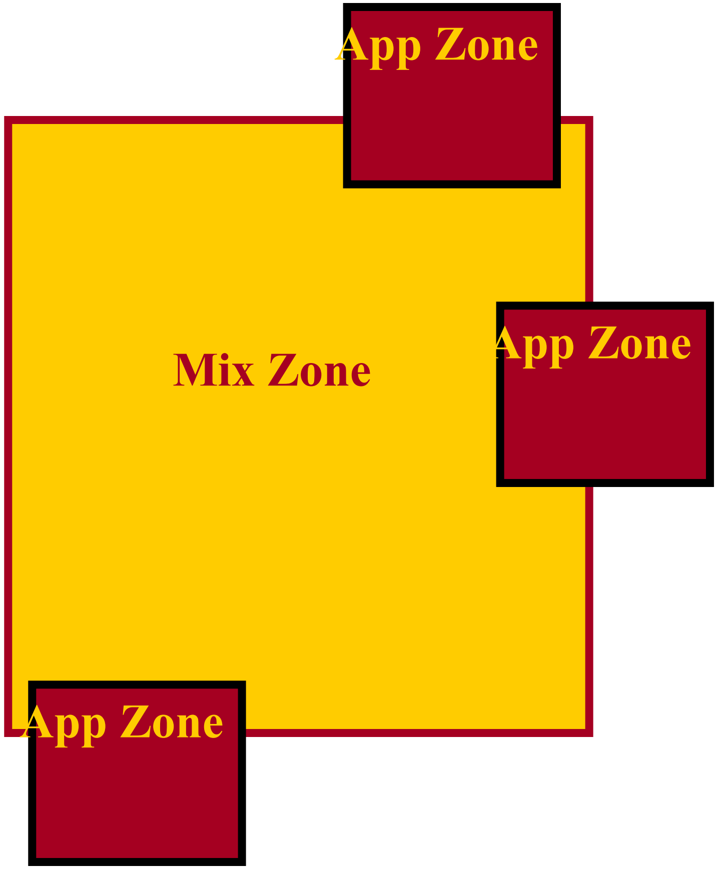
The middle are limits the location information received by applications to location sightings of registered users located inside the application zone.

Mix zones: each user has one or more unregistered geographical regions where no application can trace user movements.



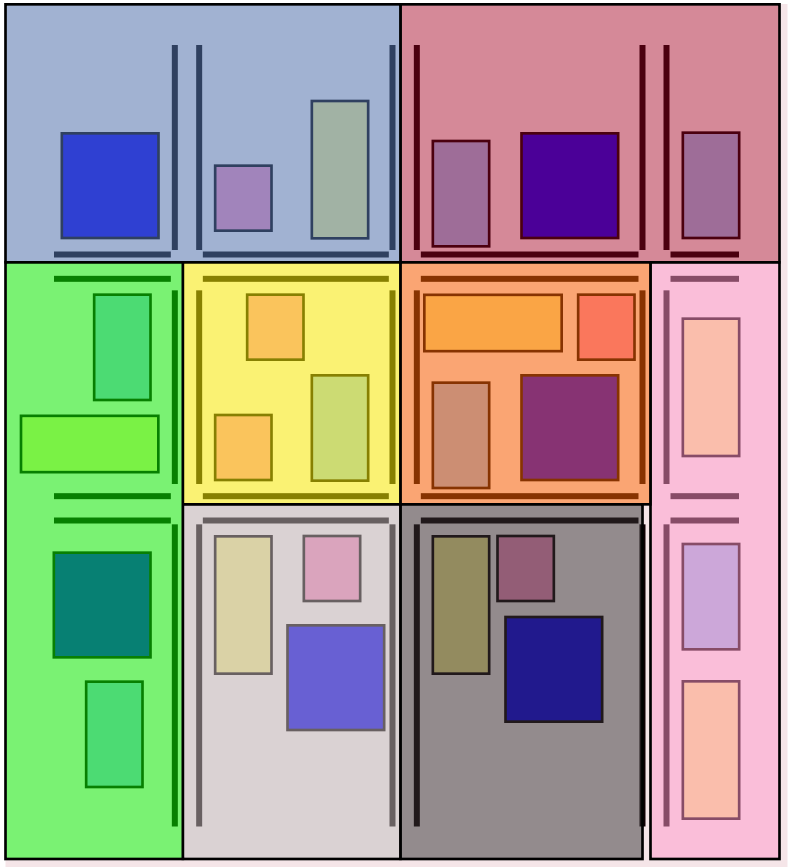
Third Trusted Party Architecture: Mix Zones

- Users can change their pseudonyms once they enter the *mix zone*
- *A* user may refuse to send any location update if the *mix zone* has less than *k* users
- Upon emerging from the *mix zone*, an adversary cannot know which one of the users has came out



Third Trusted Party Architecture: k-area cloaking

- *Sensitive* areas are pre-defined
- The space is divided into a set of zones where each zone has at least *k* sensitive area
- All location updates for a user within a certain zone are buffered
- Upon leaving a zone, user locations are revealed only if the users did not visit any of the sensitive areas

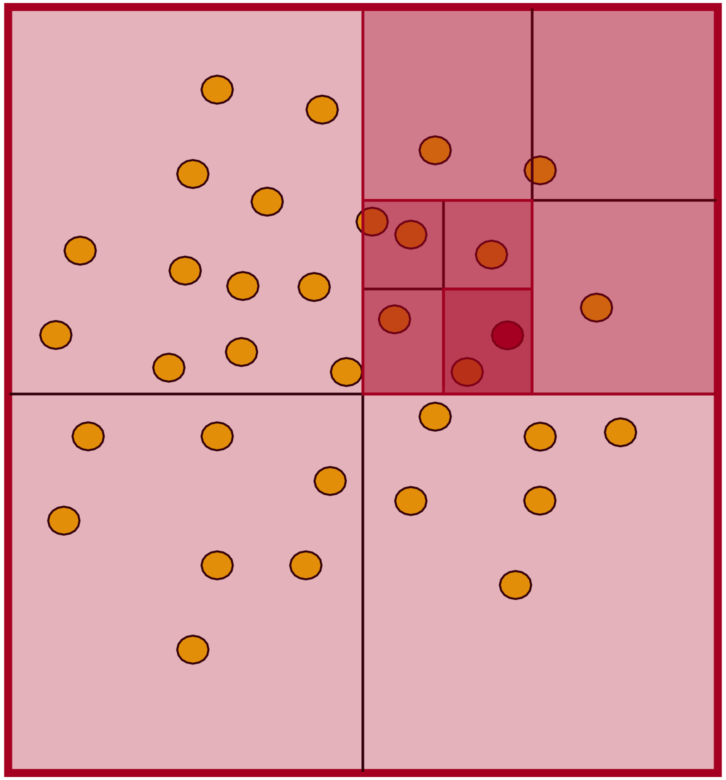


Third Trusted Party Architecture: Quadtree Spatial Cloaking

- Achieve *k-anonymity*, i.e., a user is indistinguishable from other *k-1* users

- Recursively divide the space into quadrants until a quadrant has less than *k* users.

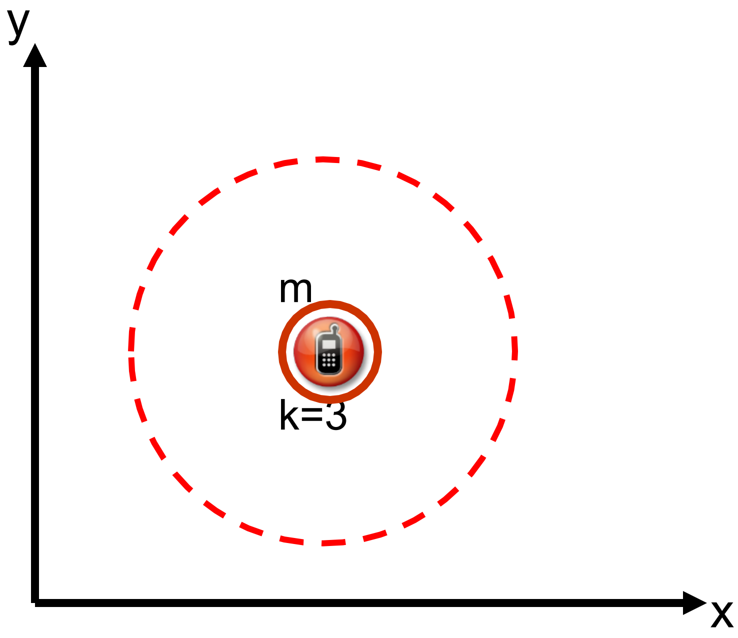
- The previous quadrant, which still meet the *k-anonymity* constraint, is returned



Achieve 5-anonmity for ●

Third Trusted Party Architecture: Bi-directional CliqueCloak

Definitions:

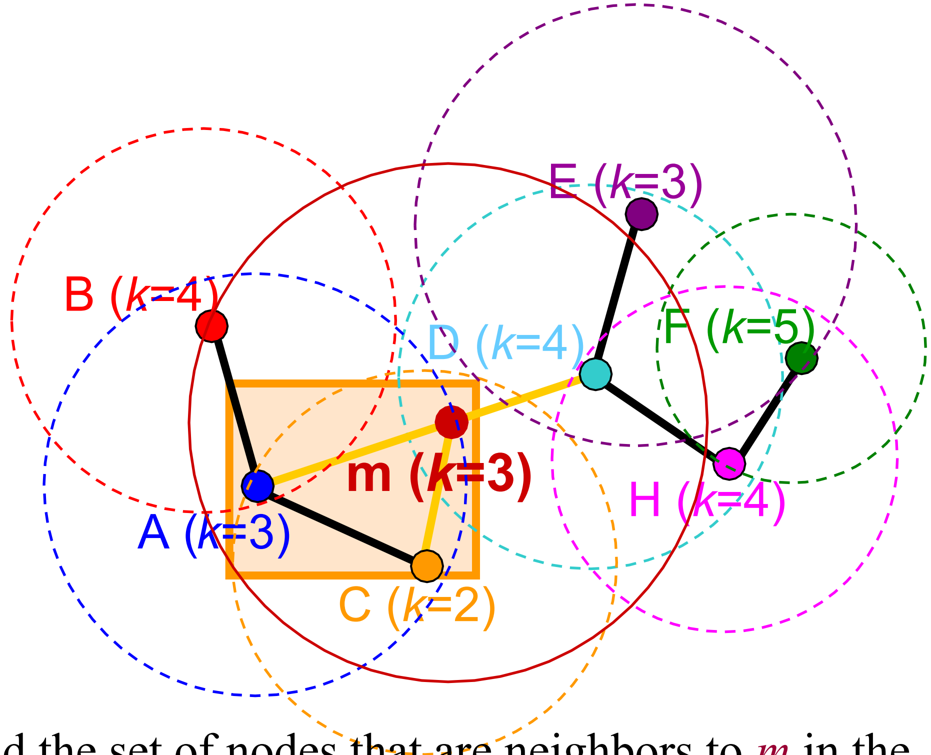


Cloaked (Constraint) Area:

For a message m, a constraint area is a spatial-temporal area that contains the sending client's location. A client sends his message along with a constraint area to prevent the database from sending the client useless information on locations outside the constraint area.

Third Trusted Party Architecture: CliqueCloak Algorithm

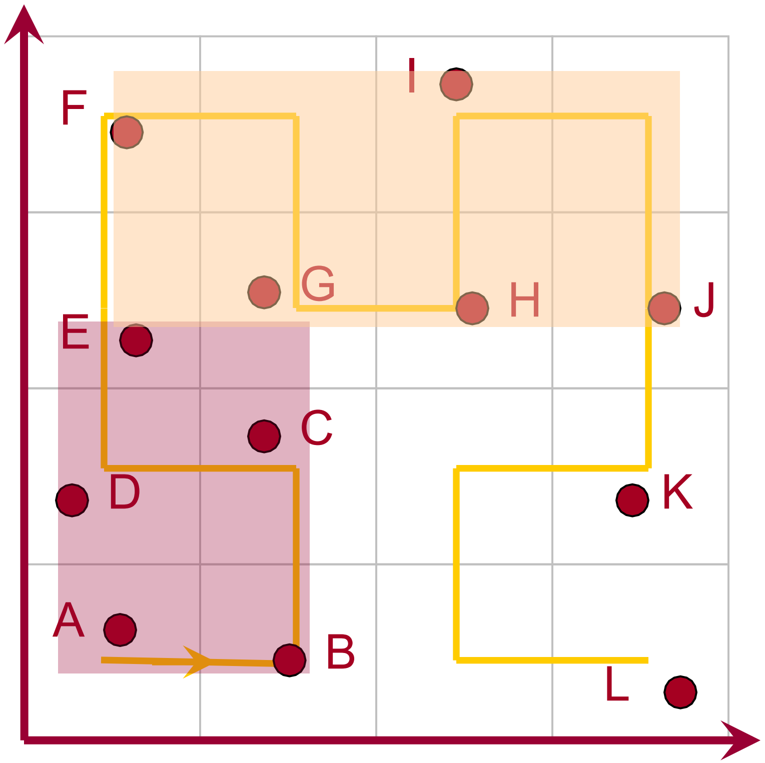
- Each user requests:
  - A level of *k* anonymity
  - A maximum cloaked area
- Build an undirected constraint graph. Two nodes are neighbors, if their maximum areas contain each other.



- For a new user *m*, add *m* to the graph. Find the set of nodes that are neighbors to *m* in the graph and has level of anonymity <= *m.k*
- The cloaked region is the MBR that includes the user and neighboring nodes. All users within an MBR use that MBR as their cloaked region

Third Trusted Party Architecture: Hilbert k-Anonymizing

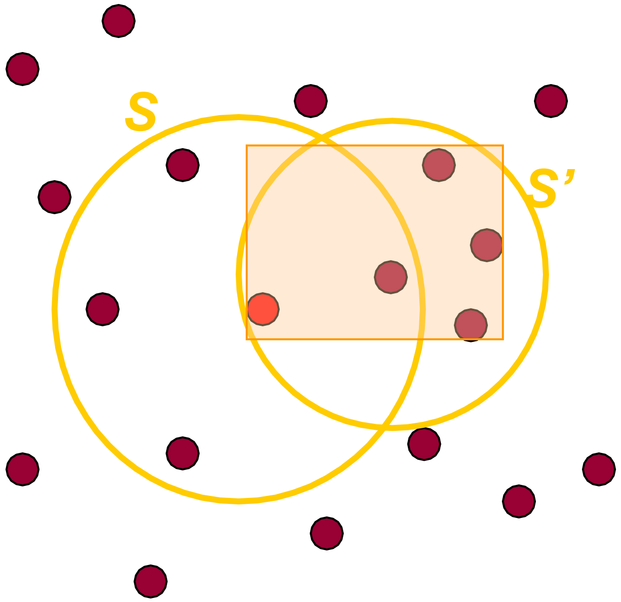
- All user locations are sorted based on their Hilbert order
- To anonymize a user, we compute *start* and *end* values as:
  - *start* = $rank_u$ - ($rank_u$ mod k_u)
  - *end* = *start* + k_u – 1
- A cloaked spatial region is an MBR of all users within the range (from *start* to *end*).
- The main idea is that it is always the case that k_u users would have the sane *[start,end]* interval



	A	B	C	D	E	F	G	H	I	J	K	L
k_u	6	5	4	5	4	5	6	5	7	4	5	4

Third Trusted Party Architecture: Nearest-Neighbor k-Anonymizing

- STEP 1: Determine a set *S* containing *u* and *k* - 1 *u*'s nearest neighbors.
- STEP 2: Randomly select *v* from *S*.
- STEP 3: Determine a set *S'* containing *v* and *v*'s *k* - 1 nearest neighbors.
- STEP 4: A cloaked spatial region is an MBR of all users in *S'* and *u*.

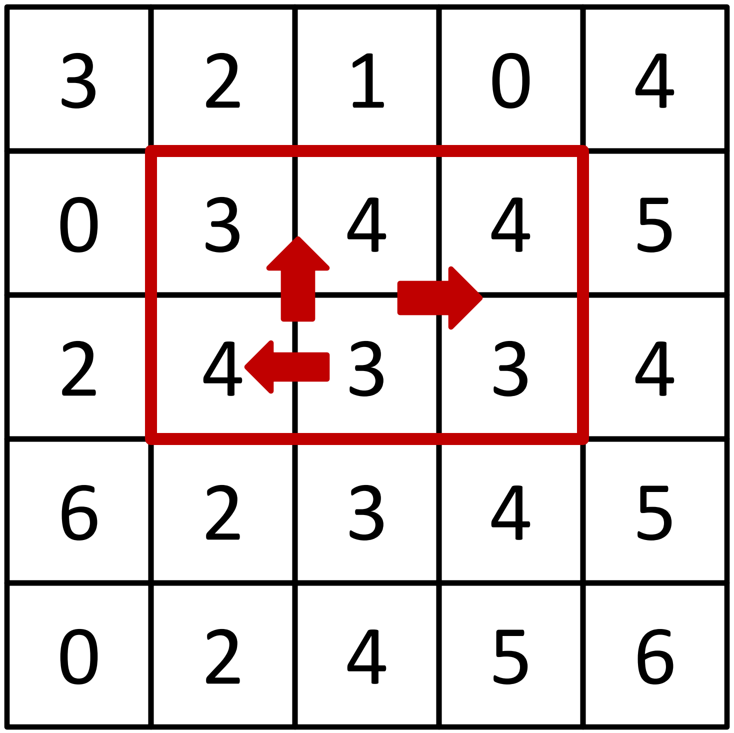


- The main idea is that randomly selecting one of the k nearest neighbors achieves the k-anonymity

Third Trusted Party Architecture: Privacy Grid

- The system space is divided into grid cells where each cell maintains the number of users in the cell

- To anonymize a user request, we start from the cell containing the user, then we expand the cell area to neighboring cells until the user privacy requirements is satisfied

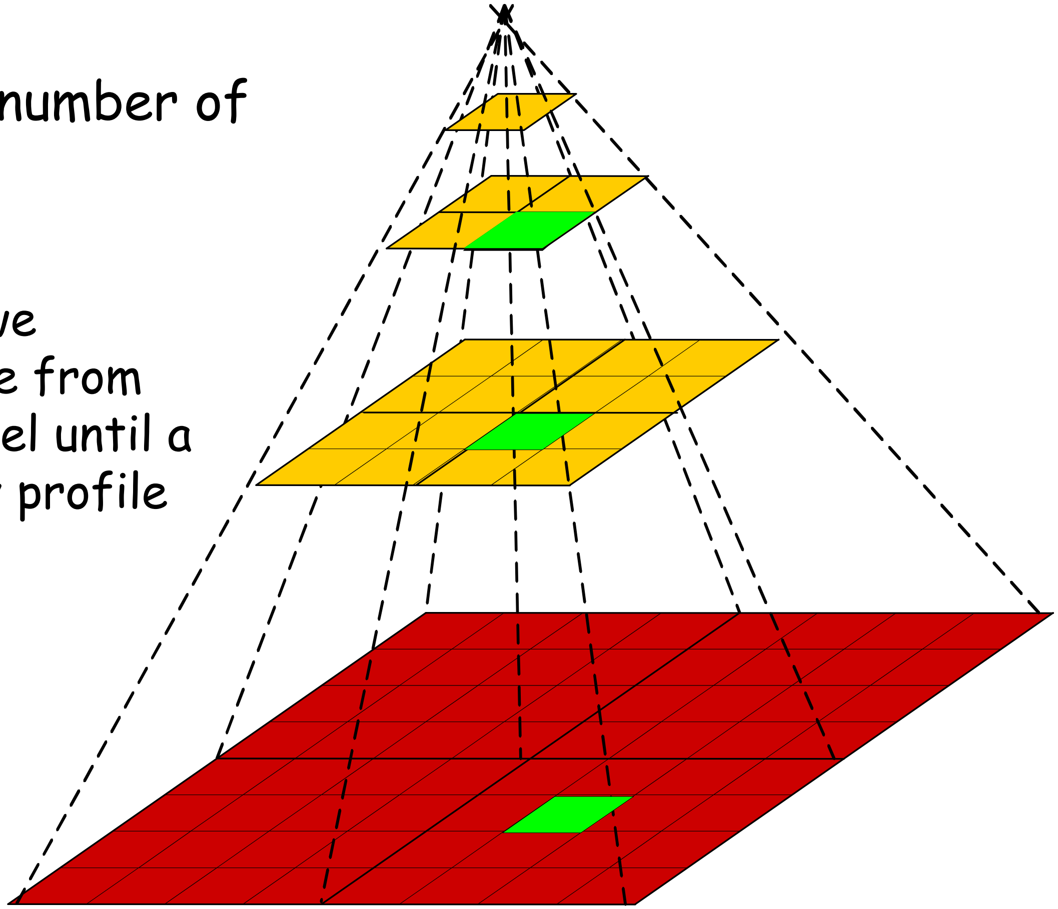


3	2	1	0	4
0	3	4	4	5
2	4	3	3	4
6	2	3	4	5
0	2	4	5	6

Anonymity level = 20

Third Trusted Party Architecture: Basic Pyramid Structure

- The entire system area is represented as a *complete pyramid* structure divided into grids at different levels of various resolution



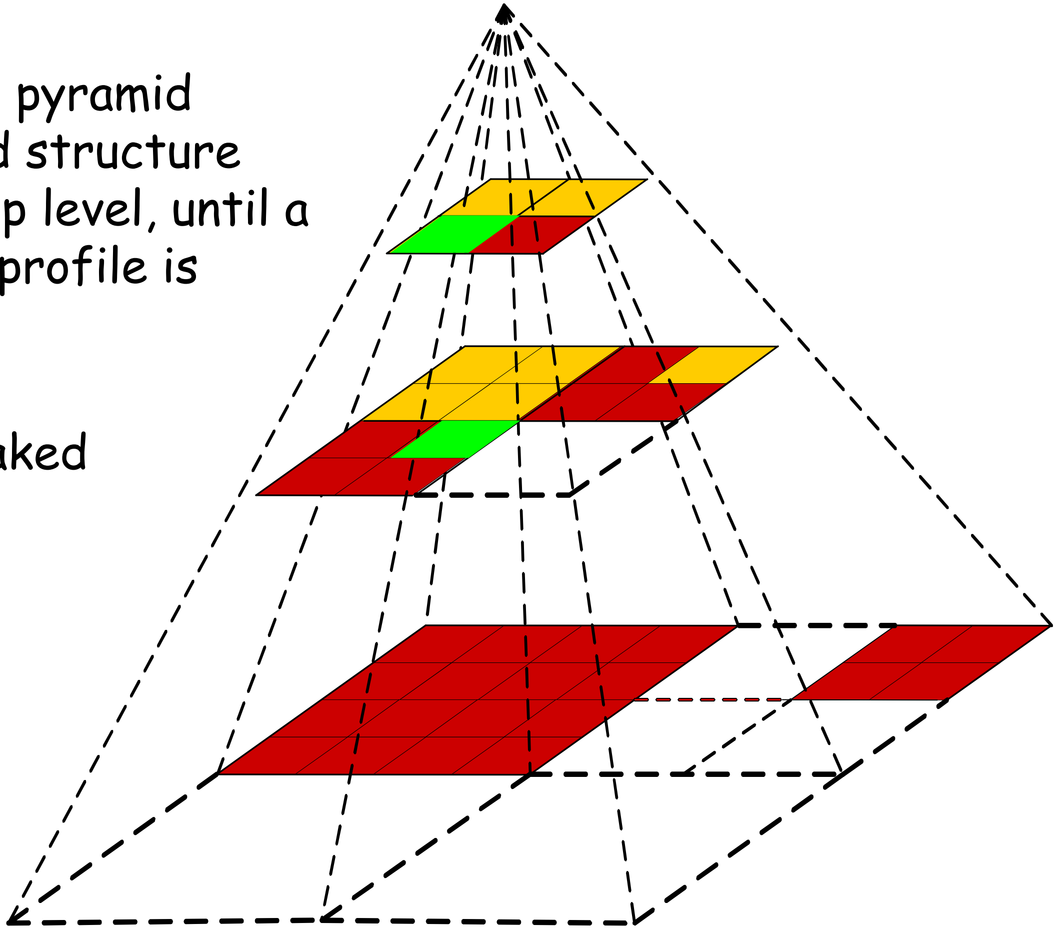
- Each grid cell maintains the number of users in that cell

- To anonymize a user request, we traverse the pyramid structure from the bottom level to the top level until a cell satisfying the user privacy profile is found.

- Scalable. Simple to implement. Overhead in maintaining all grid cells

Third Trusted Party Architecture: Adaptive Pyramid Structure

- Instead of maintaining all pyramid cells, we maintain only those cells that are potential cloaked regions
- Similar to the case of the basic pyramid structure, traverse the pyramid structure from the bottom level to the top level, until a cell satisfying the user privacy profile is found.
- Most likely we will find the cloaked region in only one hit
- Scalable. Less overhead in maintaining grid cells. Need maintenance algorithms



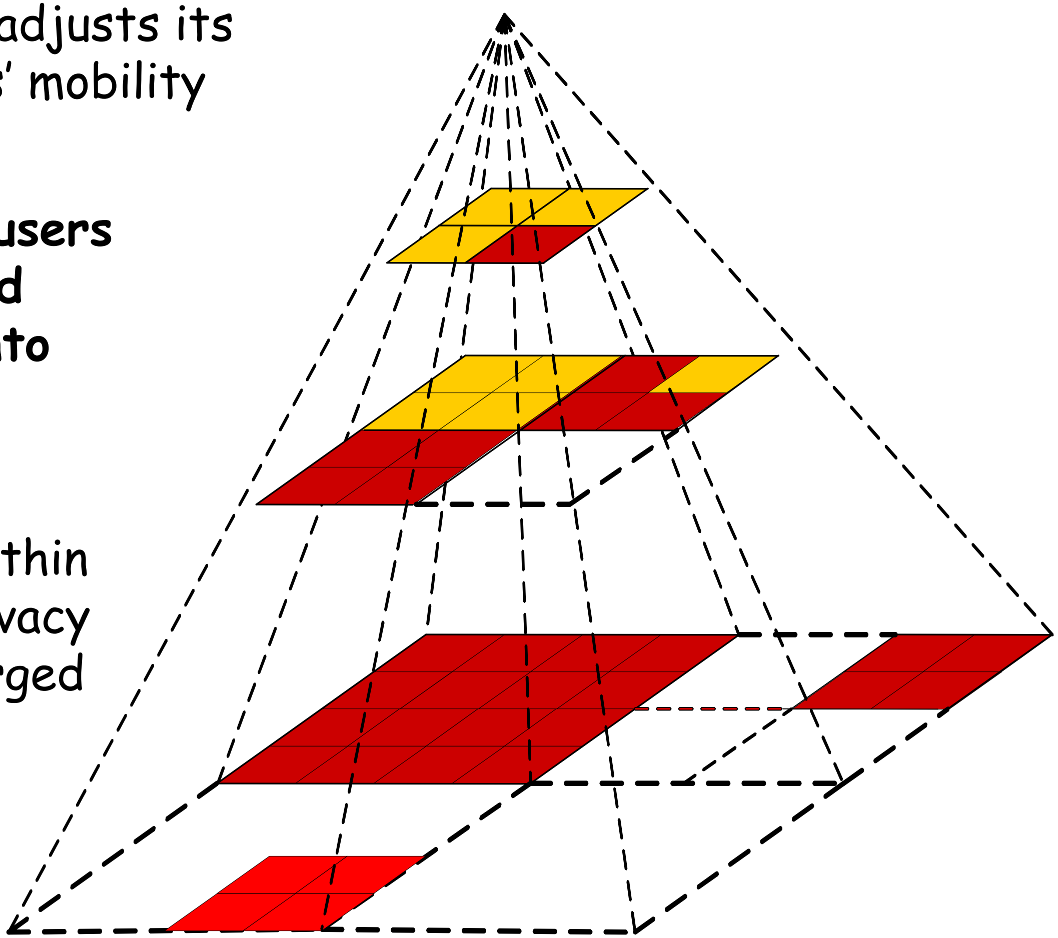
Third Trusted Party Architecture:

Adaptive Pyramid Structure: Maintenance

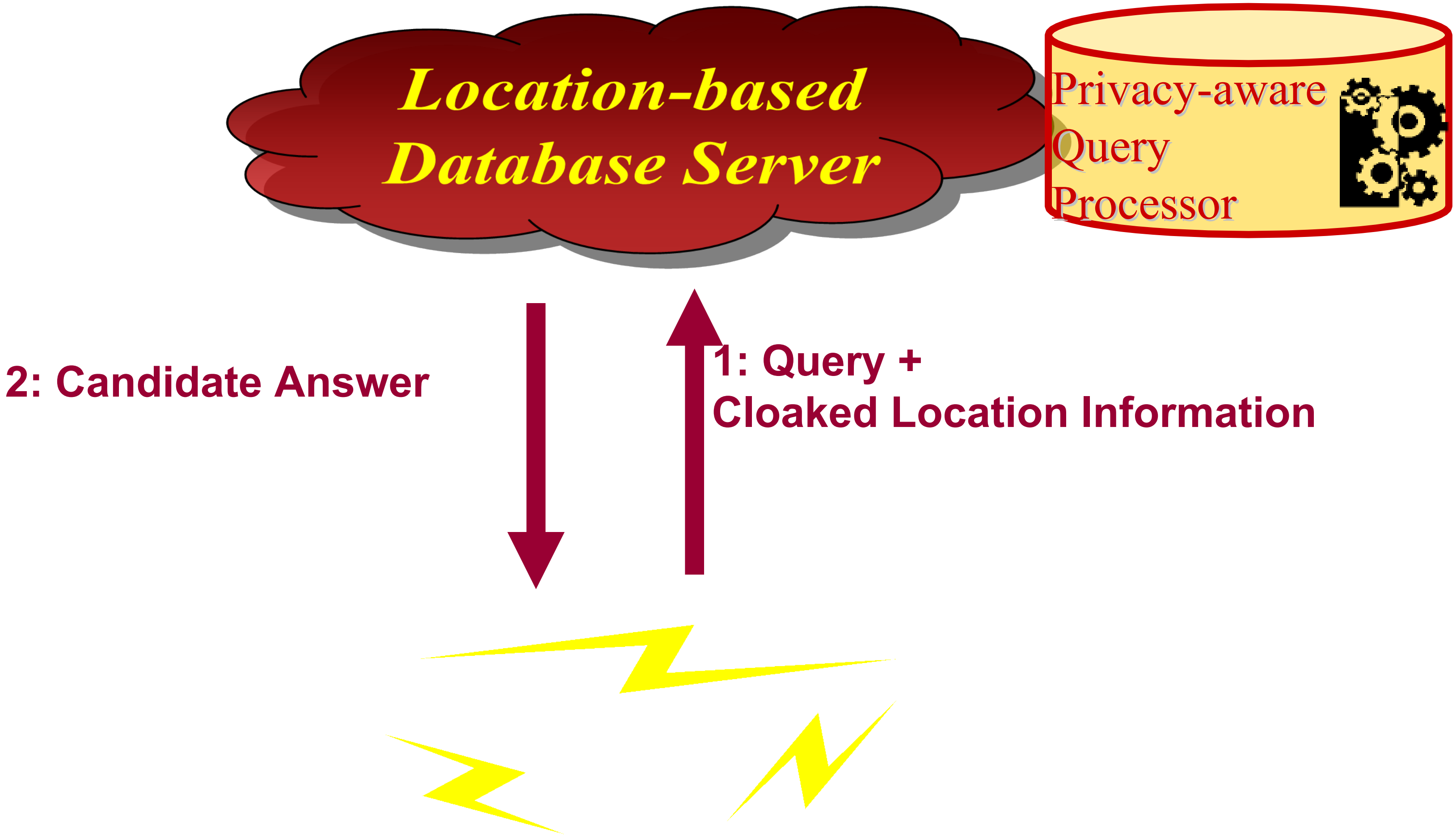
- To guarantee its efficiency, the adaptive pyramid structure dynamically adjusts its maintained cells based on users' mobility

- ***Cell Splitting:*** **Once one of the users in a certain cell expresses relaxed privacy profile, the cell is split into four lower cells**

- ***Cell Merging:*** Once all users within certain cells strength their privacy profiles, those cells can be merged together



Peer-to-Peer Architecture



Location-based Database Server

Privacy-aware Query Processor

2: Candidate Answer

1: Query +
Cloaked Location Information

Peer-to-Peer Architecture

- Peer users are collaborating with each others to keep their customized privacy information

- A result of evolving mobile peer-to-peer communication technologies

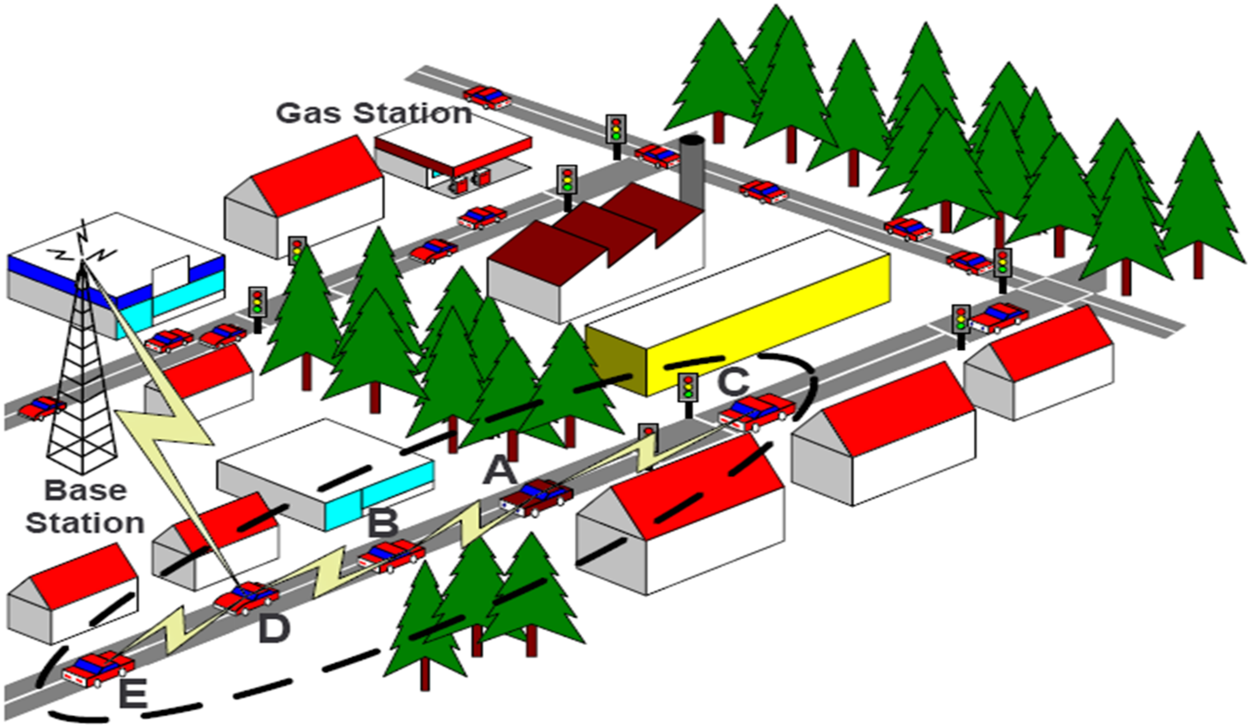
- No need for a third trusted party

- A certificate could be applied to approve trustworthy users

- *Examples:* Group Formation and PRIVE

Peer-to-Peer Architecture

Group Formation

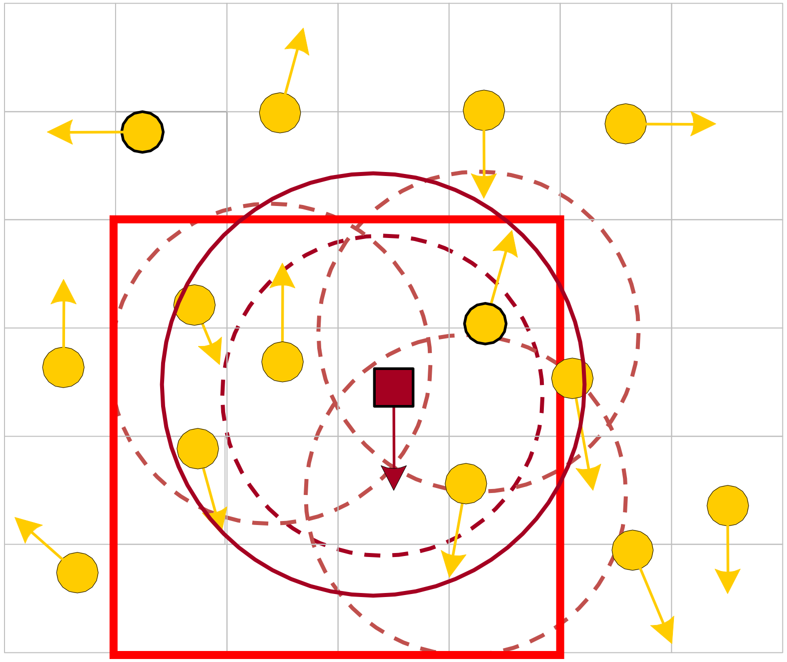


- The main idea is that whenever a user wants to issue a location-based query, the user broadcasts a request to its neighbors to form a group. Then, a random user of the group will act as the query sender.

Peer-to-Peer Cooperative Architecture

Group Formation

- Phase 1: Peer Searching
  - Broadcast a multi-hop request until at least *k*-1 peers are found
- Phase 2: Location Adjustment
  - Adjust the locations using velocity
- Phase 3: Spatial Cloaking
  - Blur user location into a region aligned to a grid that cover the *k*-1 nearest peers



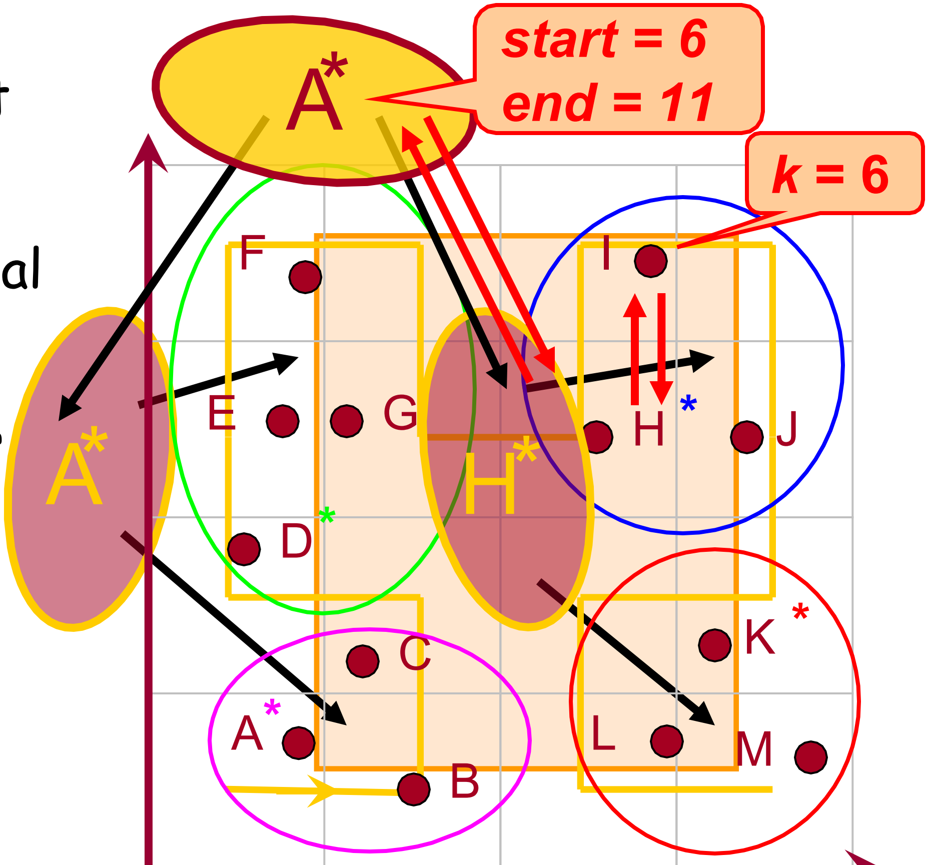
Example: *k* = 5

- ***On-demand mode***
  - A mobile user only forms an anonymous group when it needs it
- ***Proactive mode***
  - Mobile users periodically execute the on-demand approach to maintain their anonymous groups

Peer-to-Peer Cooperative Architecture

Hierarchical Hilbert Peer-to-Peer

- Users are sorted by their Hilbert values.
- Users are grouped in a hierarchical way
- Cluster heads are responsible for handling users' requests
- The root is responsible for calculating *start* and *end* values
  - *start* = *rank*$_u$ - (*rank*$_u$ mod *k*$_u$)
  - *end* = *start* + *k*$_u$ - 1

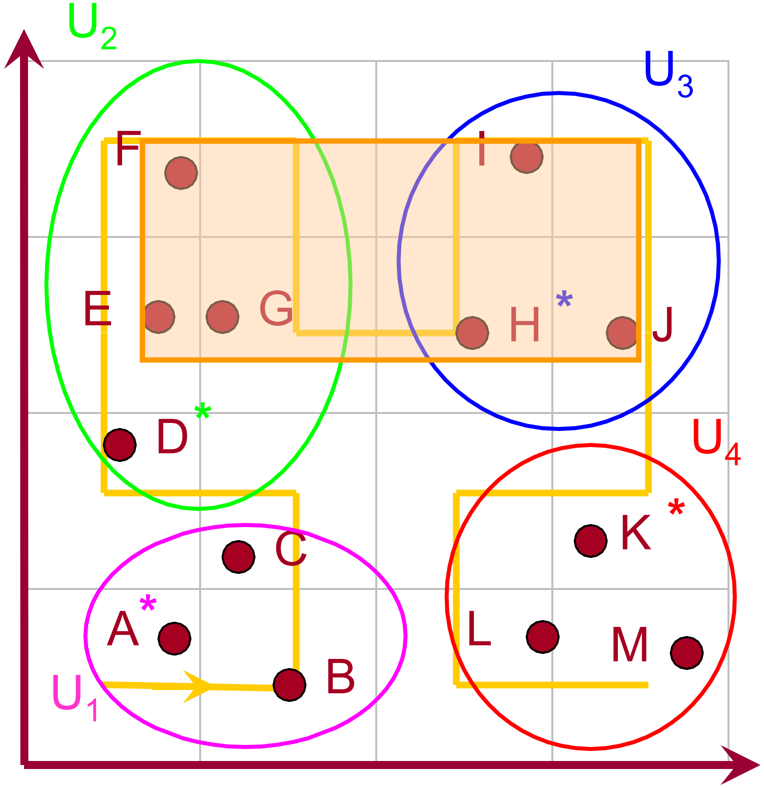


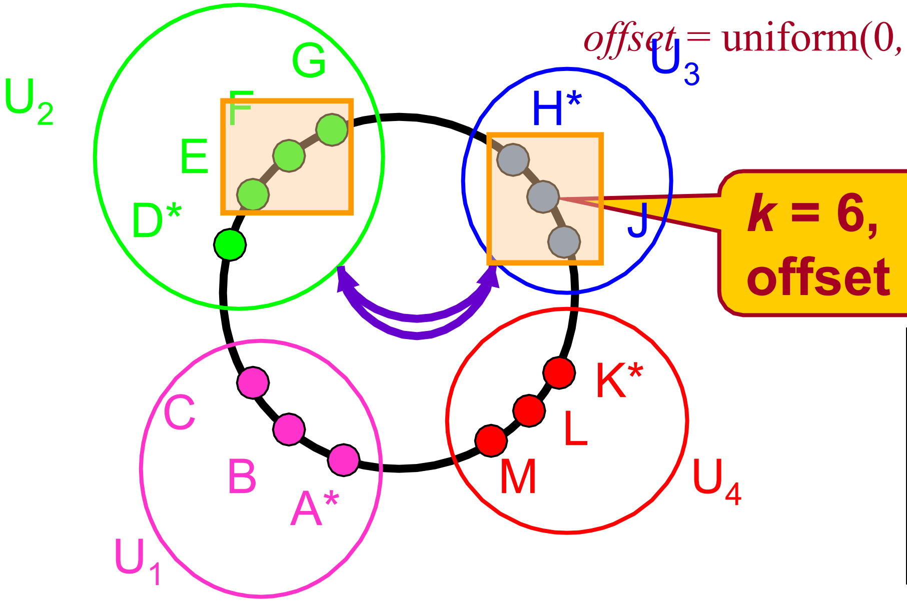
	A	B	C	D	E	F	G	H	I	J	K	L	M
k$_u$	6	5	4	5	4	5	6	5	6	4	5	4	5

Peer-to-Peer Cooperative Architecture

Non-Hierarchical Hilbert Peer-to-Peer

- Instead of organizing users on a tree, users are organized as a ring
- To get anonymized, a user generates a random *offset*
- Send to all involved clusters that involve *[offset,offset+k_u-1]*





offset = uniform(0, k_u-1)

***k* = 6, offset =4**

	A	B	C	D	E	F	G	H	I	J	K	L	M
k_u	6	5	4	5	4	5	6	5	6	4	5	4	5
H(u)	1	2	3	4	5	6	8	9	10	12	13	15	16
Rank$_u$	0	1	2	3	4	5	6	7	8	9	10	11	12