Location Privacy Protection

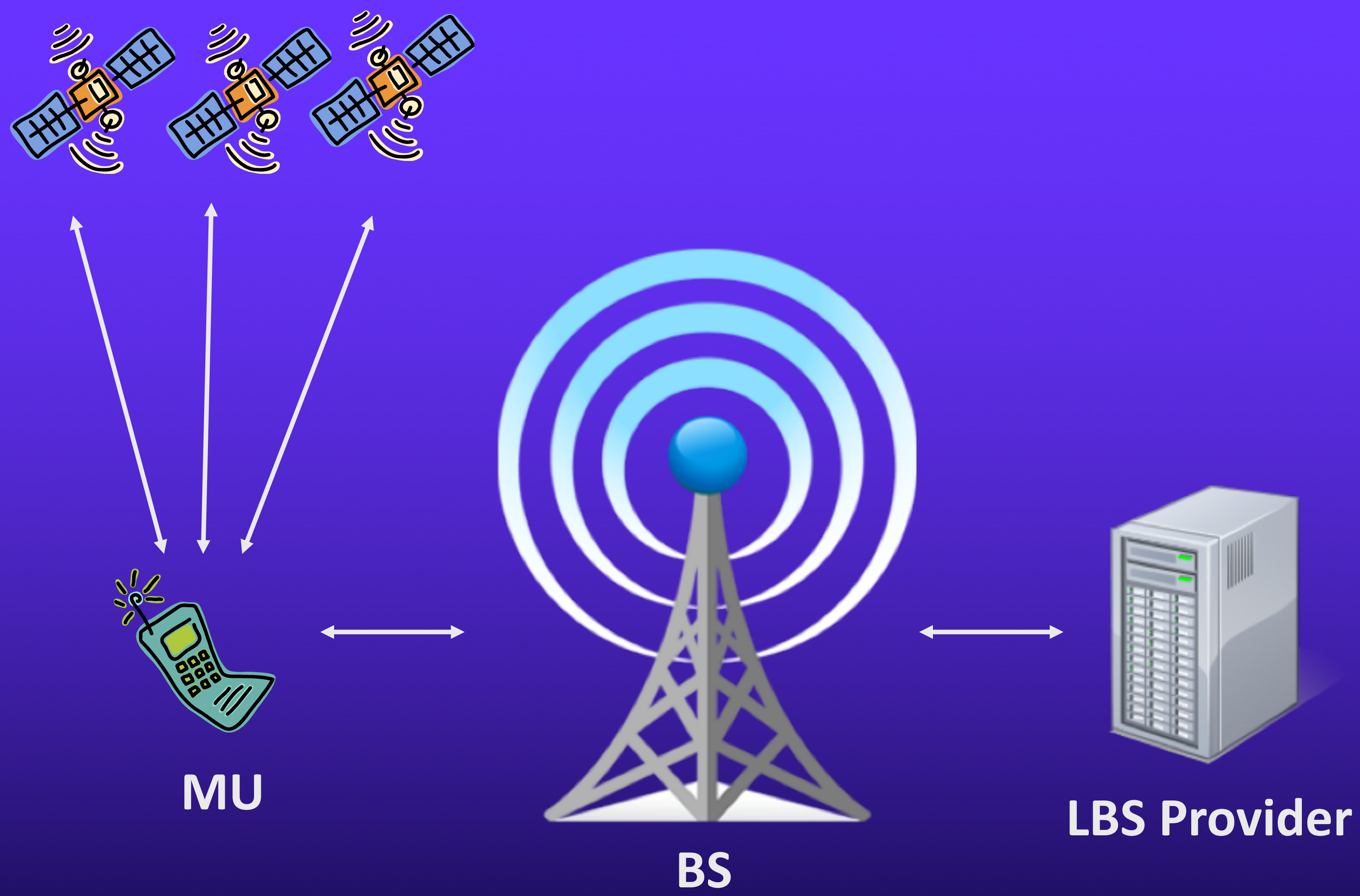
Xun Yi

RMIT University

Outlines

- privacy issues with LBS
- existing privacy-preserving solutions for LBS and problems
- our model for private location-based queries
- our solutions for private location-based queries
- security and performance analysis
- conclusions

Location-Based Service (LBS) (point-of-interest (POI) query)

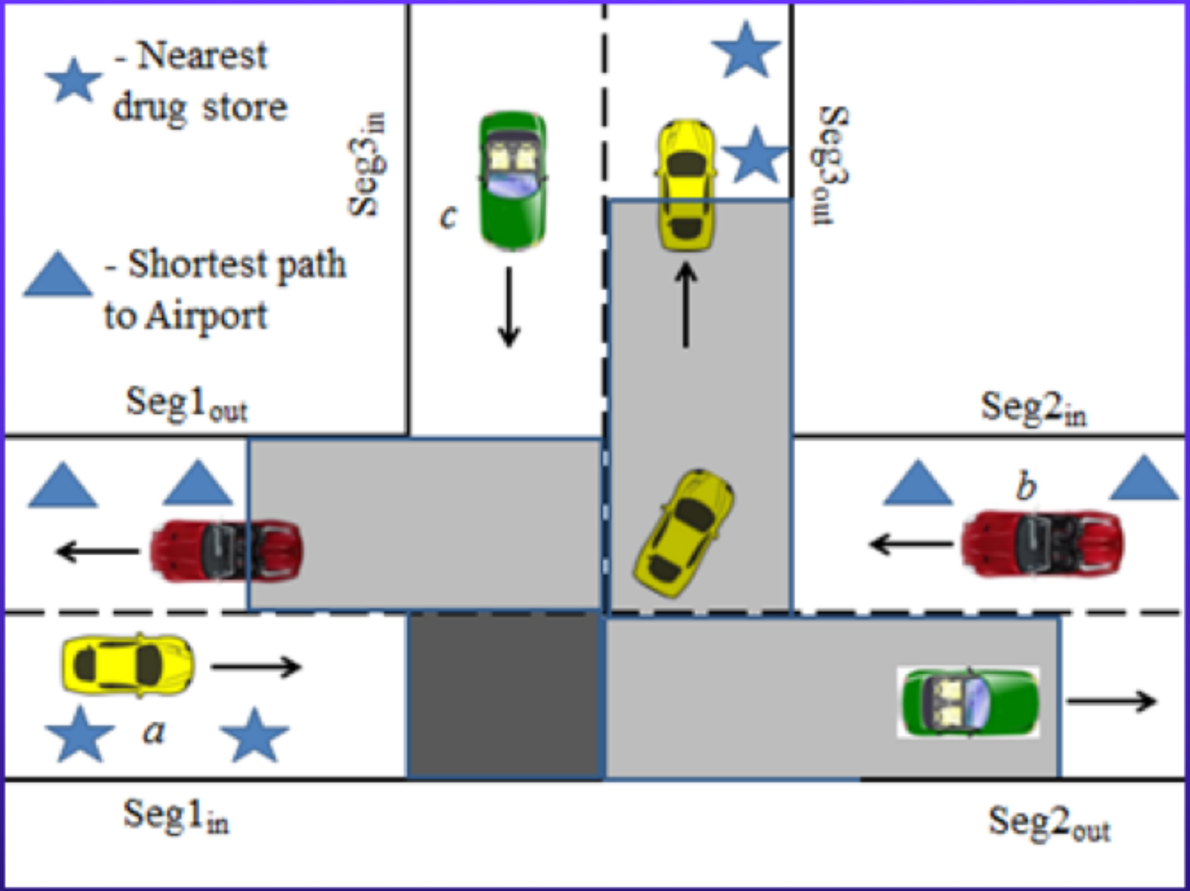


Privacy issues with LBS

- user privacy (location privacy)
  - location information collected from mobile users can reveal far more than just a user's latitude and longitude. Knowing where a mobile user is can mean knowing what he is doing
  - private location
- server privacy (data privacy)
  - server provides LBS for business purpose
  - payment per query, one record per query

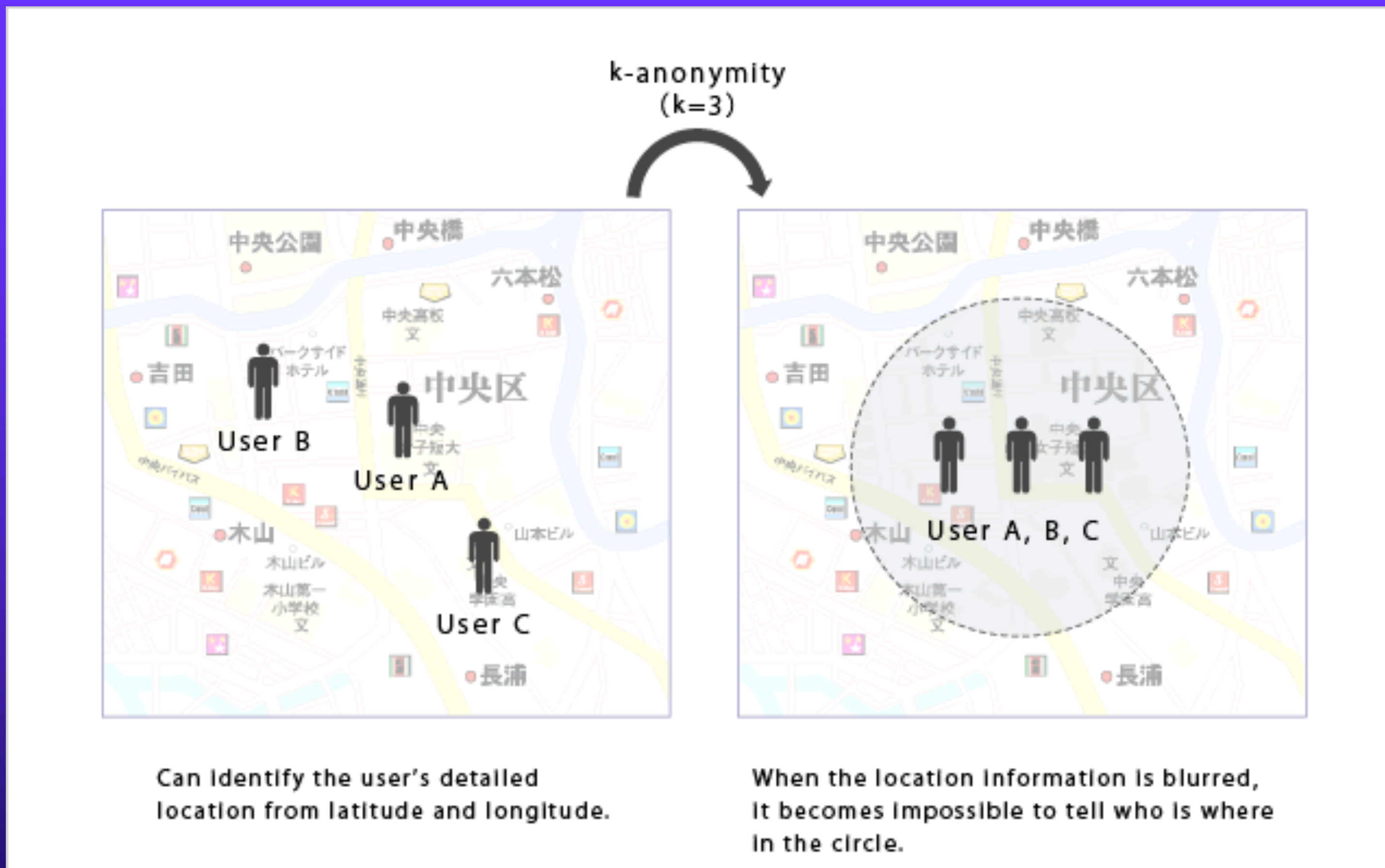
Existing solutions

- mix zone (Beresford and Stajano, IEEE Pervasive Computing 2003)



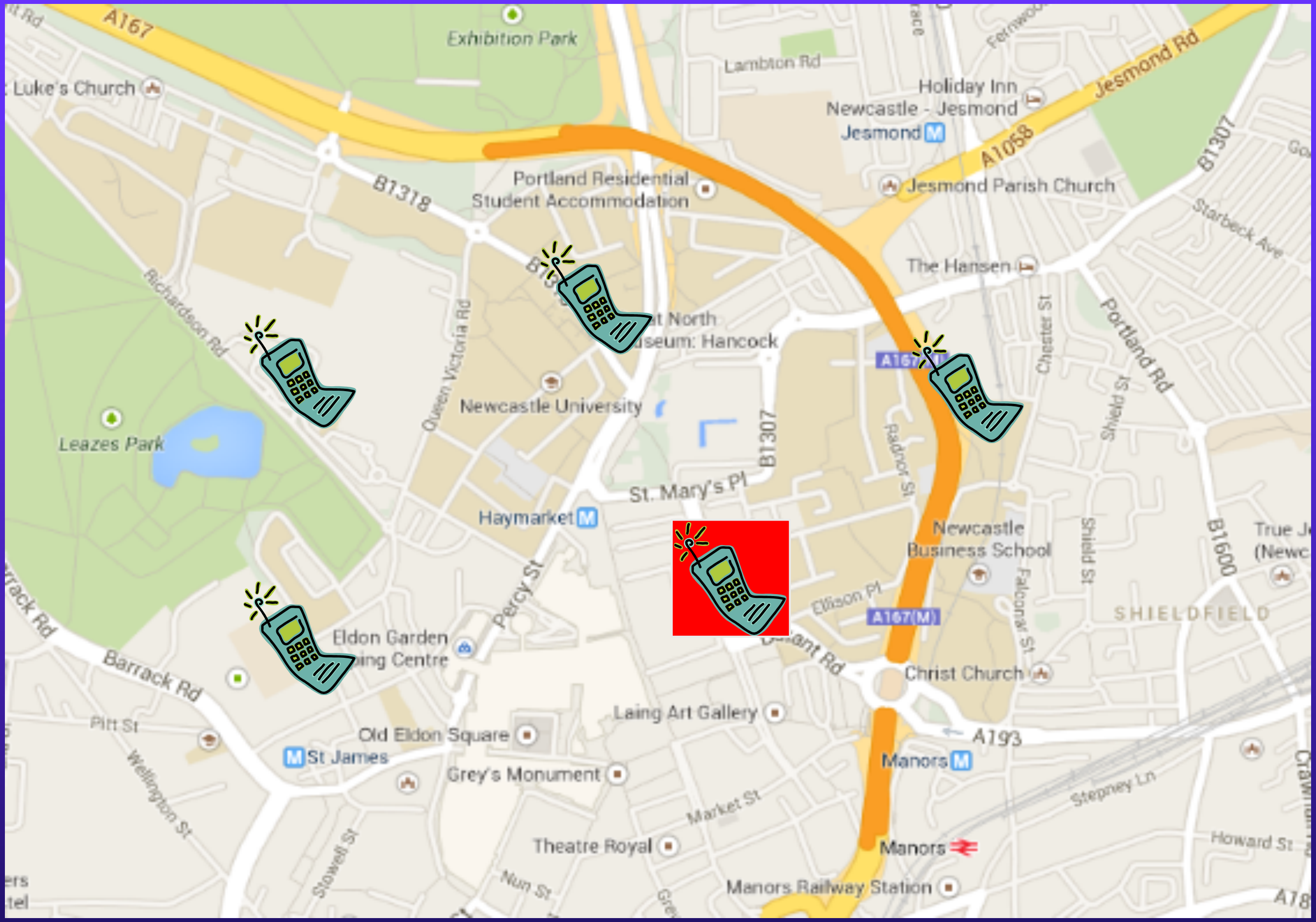
Cont.

- k-anonymity (Mokbel et al., VLDB 2006 / Bamba et al., WWW 2008)



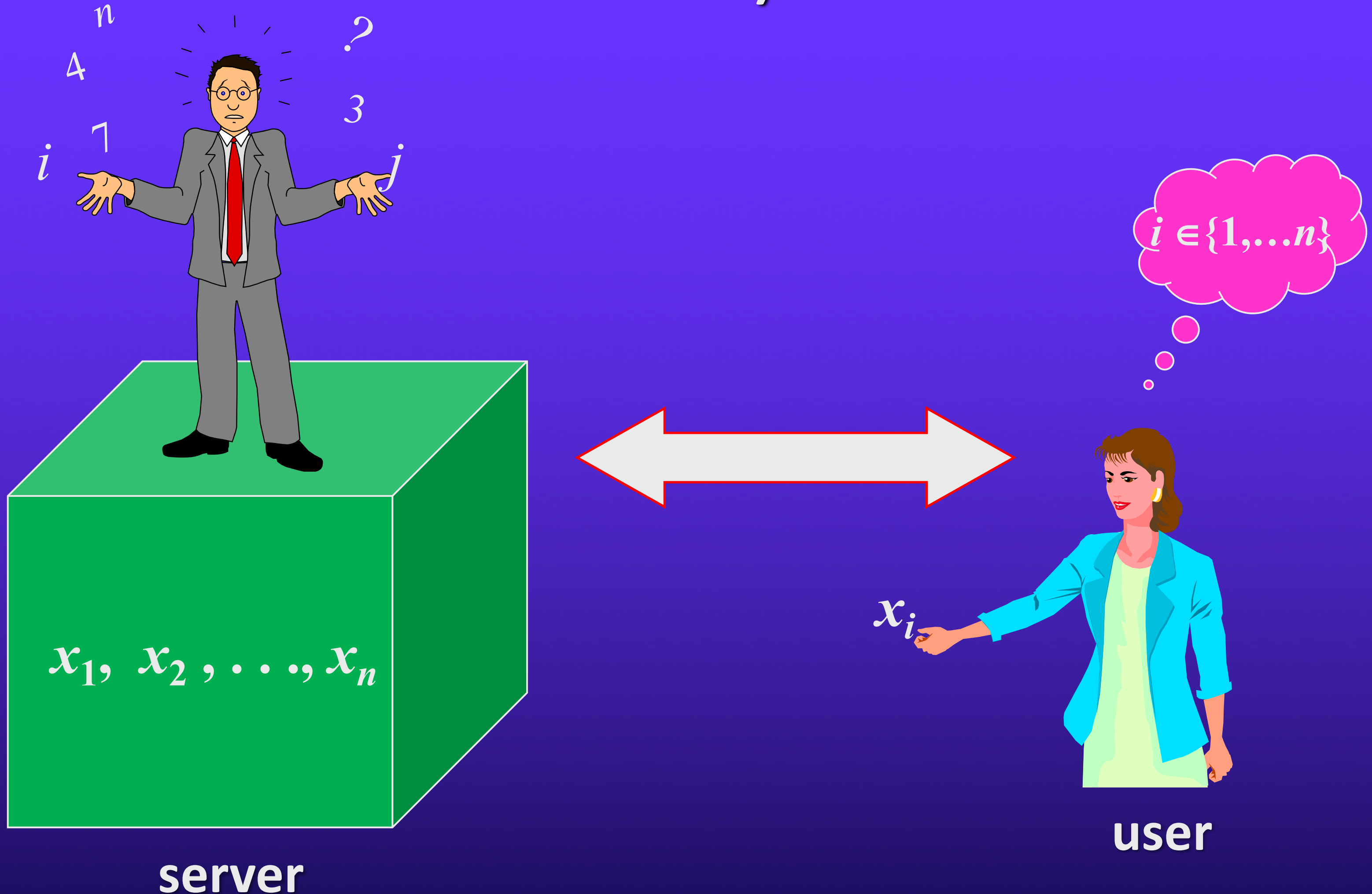
Cont.

- "dummy" locations (Kido et al., ICPS 2005 / Shankar et al., UBICOMP 2009)



Cont.

- private information retrieval (PIR) (Ghinita et al., WWW 2007 and SIGMOD 2008 / Yi et al. ICDE 2012 and IEEE TKDE)

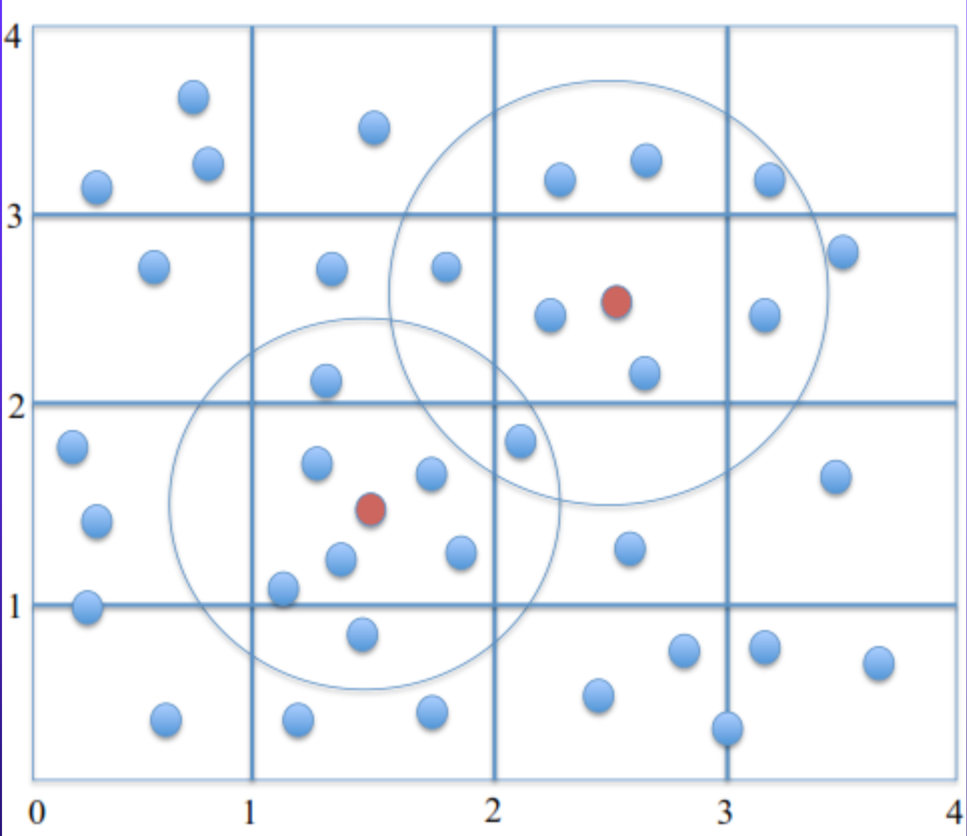


Problems

- mix zone and k-anonymity require the middleware that maintains all user locations
- k-anonymity is not suitable for location privacy protections, where the notion of distance between locations is important
- “dummy” locations require the mobile user randomly to choose and send a set of fake locations to the LBS and to receive the false reports from the LBS

Two Problems

- k nearest neighbor queries
- type of point-of-interest



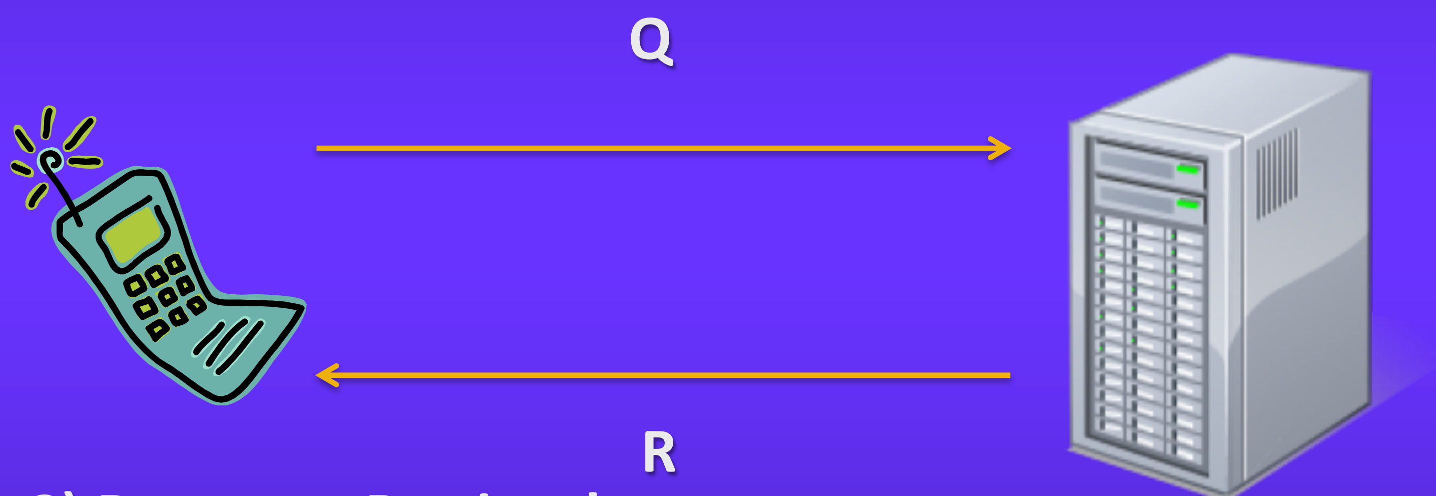
Our model

1) Query Generation

(Q,s)=QG(CR,n,m,(i, j),t)

2) Response Generation

R=RG(Q,D)



3) Response Retrieval

kNN=RR(R,s)

LBS Provider

Mobile User

Query generation [solution 1] (without server privacy) (Paillier)

Algorithm 1 Query Generation (User)

Input: $CR, n, (i, j)$

Output: Q, s

1: Randomly choose two large primes p, q such that $N = pq > M$.

2: Let $sk = \{p, q\}$ and $pk = \{g, N\}$, where g is chosen from $\mathbb{Z}_{N^2}$ and its order is a nonzero multiple of N.

3: For each $\ell \in \{1, 2, \cdots, n\}$, pick a random integer $r_\ell \in \mathbb{Z}^*_{N^2}$, compute

$$c_\ell = \begin{cases} Encrypt(1, pk) = g^1 r_\ell^N (mod\ N^2) & \text{if } \ell = i \\ Encrypt(0, pk) = g^0 r_\ell^N (mod\ N^2) & \text{otherwise} \end{cases}$$

where the encryption algorithm is described in the Paillier cryptosystem (please refer to Appendix A).

4: Let $Q = \{CR, n, c_1, c_2, \cdots, c_n, pk\}$, $s = sk$.

5: **return** Q, s

Public key cryptosystem

Key Generation → (pk,sk)

pk: encryption key

sk: decryption key

pk

C=E(m,pk)

m=D(C,sk)

Response generation [solution 1]

Algorithm 2 Response Generation RG (Server)

Input: $D, Q = \{CR, n, c_1, c_2, \cdots, c_n, (g, N)\}$

Output: $R = \{C_1, C_2, \cdots, C_n\}$

1: Based on CR and n, compute $R = \{C_1, C_2, \cdots, C_n\}$ where for $\gamma = 1, 2, \cdots, n$,

$$C_\gamma = \prod_{\ell=1}^{n} c_\ell^{d_{\ell,\gamma}} (mod\ N^2)$$

2: **return** R

Paillier cryptosystem has two homomorphic properties:
$E(m_1)E(m_2)=E(m_1+m_2)$, $E(m_1)^{m2}=E(m_1m_2)$

Location-based database

column j changes

row i changes

$$\begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,j} & \cdots & d_{1,n-1} & d_{1,n} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,j} & \cdots & d_{2,n-1} & d_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{i,1} & d_{i,2} & \cdots & d_{i,j} & \cdots & d_{i,n-1} & d_{i,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{n-1,1} & d_{n-1,2} & \cdots & d_{n-1,j} & \cdots & d_{n-1,n-1} & d_{n-1,n} \\ d_{n,1} & d_{n,2} & \cdots & d_{n,j} & \cdots & d_{n,n-1} & d_{n,n} \end{bmatrix}$$

$$c_\gamma^{d_{i,j}} : E(0)^{d_{i,j}} = E(0), E(1)^{d_{i,j}} = E(d_{i,j})$$

Response retrieval [solution 1]

Algorithm 3 Response Retrieval RR (User)

Input: $R = \{C_1, C_2, \cdots, C_n\}, sk = s$

Output: d

1: Compute

$$d = Decrypt(C_j, sk),$$

where the decryption algorithm is described in the Paillier cryptosystem (please refer to Appendix A).

2: **return** d

$$E(d_{i,1}), \cdots, E(d_{i,j}), \cdots, E(d_{i,n})$$

Query generation [solution 2] (with server privacy)

Algorithm 4 Query Generation (User)

Input: $CR, n, (i, j)$

Output: Q, s

1: Randomly choose two large primes p, q such that $N = pq > M$.

2: Let $sk = \{p, q\}$ and $pk = \{g, N\}$, where g is chosen from $\mathbb{Z}_{N^2}$ and its order is a nonzero multiple of N.

3: For each $\ell \in \{1, 2, \cdots, n\}$, pick a random integer $r_\ell \in \mathbb{Z}^*_{N^2}$, compute

$$c_\ell = \begin{cases} Encrypt(1, pk) = g^1 r_\ell^N (mod\ N^2) & \text{if } \ell = i \\ Encrypt(0, pk) = g^0 r_\ell^N (mod\ N^2) & \text{otherwise} \end{cases}$$

4: Pick a random integer $r \in \mathbb{Z}^*_{N^2}$, compute

$$c = Encrypt(j, pk) = g^j r^N (mod\ N^2)$$

5: Let $Q = \{CR, n, c_1, c_2, \cdots, c_n, c, pk\}$, $s = sk$.

6: **return** Q, s

Response generation [solution 2]

Algorithm 5 Response Generation RG (Server)

Input: $D, Q = \{CR, n, c_1, c_2, \cdots, c_n, c, (g, N)\}$

Output: $R = \{C_1, C_2, \cdots, C_n\}$

Based on CR and n, compute $R = \{C_1, C_2, \cdots, C_n\}$ where for $\gamma = 1, 2, \cdots, n$,

$$C_\gamma = (c/g^\gamma)^{w_\gamma} \prod_{\ell=1}^{n} c_\ell^{d_{\ell,\gamma}^2} (mod\ N^2),$$

where w_t is randomly chosen from $\mathbb{Z}_N^*$.

return R

Location-based database (Rabin)

column j changes

row i changes

$$\begin{bmatrix} d_{1,1}^2 & d_{1,2}^2 & \cdots & d_{1,j}^2 & \cdots & d_{1,n-1}^2 & d_{1,n}^2 \\ d_{2,1}^2 & d_{2,2}^2 & \cdots & d_{2,j}^2 & \cdots & d_{2,n-1}^2 & d_{2,n}^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{i,1}^2 & d_{i,2}^2 & \cdots & d_{i,j}^2 & \cdots & d_{i,n-1}^2 & d_{i,n}^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{n-1,1}^2 & d_{n-1,2}^2 & \cdots & d_{n-1,j}^2 & \cdots & d_{n-1,n-1}^2 & d_{n-1,n}^2 \\ d_{n,1}^2 & d_{n,2}^2 & \cdots & d_{n,j}^2 & \cdots & d_{n,n-1}^2 & d_{n,n}^2 \end{bmatrix}$$

$$c_\gamma^{d_{i,j}^2} : E(0)^{d_{i,j}^2} = E(0), E(1)^{d_{i,j}^2} = E(d_{i,j}^2)$$

Why Rabin public key encryption?

- It can prevents dishonor user from retrieving more records

$$c_1 = E(1), c_i = E(1), c_k = E(0), d_{1,j} + d_{i,j}$$
$$c_1 = E(1), c_i = E(1), c_k = E(0), d_{1,j}^2 + d_{i,j}^2$$

- Rabin public key encryption is the simplest
- Rabin and Paillier can share the same public key and private key

Response retrieval [solution 2]

Algorithm 6 Response Retrieval RR (User)

Input: $R = \{C_1, C_2, \cdots, C_n\}, sk = s$

Output: d

Compute

$$C'_j = PaillierDecrypt(C_j, sk),$$

where the decryption algorithm is described in the Paillier cryptosystem (please refer to Appendix A).

Compute

$$d = RabinDecrypt(C'_j, sk),$$

where the decryption algorithm is described in the Rabin cryptosystem (please refer to Appendix B).

return d

$$E(r_1), \cdots, E(d_{i,j}^2), \cdots, E(r_n)$$

Query generation [solution 3] (based on POI type)

Algorithm 7 Query Generation (User)

Input: $CR, n, m, (i, j), t$

Output: Q, s

1: Randomly choose two large primes p_1, q_1 such that $N_1 = p_1 q_1 > M$.

2: Randomly choose two large primes p_2, q_2 such that $N_2 = p_2 q_2$, where $N_1^2 < N_2 < N_1^4$.

3: Let $sk_1 = \{p_1, q_1\}, sk_2 = \{p_2, q_2\}, pk_1 = \{g_1, N_1\}$, $pk_2 = \{g_2, N_2\}$, where g_1 is chosen from $\mathbb{Z}_{N_1{}^2}$ and its order is a nonzero multiple of N_1 and g_2 is chosen from $\mathbb{Z}_{N_2{}^2}$ and its order is a nonzero multiple of N_2.

4: For each $\ell \in \{1, 2, \cdots, m\}$, pick a random integer $r_\ell \in \mathbb{Z}^*_{N_1^2}$, compute

$$c_\ell = \begin{cases} E(1, pk_1) = g_1{}^1 r_\ell^{N_1} (mod\ N_1{}^2) & \text{if } \ell = t \\ E(0, pk_1) = g_1{}^0 r_\ell^{N_1} (mod\ N_1{}^2) & \text{otherwise} \end{cases}$$

5: For each $\ell \in \{1, 2, \cdots, n\}$, pick a random integer $r'_\ell \in \mathbb{Z}^*_{N_2^2}$, compute

$$c'_\ell = \begin{cases} E(1, pk_2) = g_2{}^1 r'^{N_2}_\ell (mod\ N_2{}^2) & \text{if } \ell = i \\ E(0, pk_2) = g_2{}^0 r'^{N_2}_\ell (mod\ N_2{}^2) & \text{otherwise} \end{cases}$$

6: Pick a random integer $r \in \mathbb{Z}^*_{N_2{}^2}$, compute

$$c = E(j, pk_2) = g_2{}^j r^{N_2} (mod\ N_2{}^2)$$

7: Let $Q = \{CR, n, m, c_1, c_2, \cdots, c_m, c'_1, c'_2, \cdots, c'_n, c, pk_1, pk_2\}$, $s = \{sk_1, sk_2\}$.

8: **return** Q, s

Paillier 1: t

Paillier 2: (i,j)

ciphertext space
$\subseteq$
plaintext space

Response generation [solution 3]

Algorithm 8 Response Generation RG (Server)

Input: $D, Q = \{CR, m, n, c_1, c_2, \cdots, c_m, c'_1, c'_2, \cdots, c'_n, c, pk_1, pk_2)$

Output: $R = \{C_1, C_2, \cdots, C_n\}$

Based on CR and m, for each cell (α, β) in CR, compute

$$C_{\alpha,\beta} = \prod_{\ell=1}^{m} c_\ell^{d^2_{\alpha,\beta,\ell}} (mod\ {N_1}^2)$$

Based on CR and n, compute $R = \{C_1, C_2, \cdots, C_n\}$, where for $\beta \in \{1, 2, \cdots, n\}$,

$$C_\beta = (c/g^\beta)^{w_\beta} \prod_{\alpha=1}^{n} {c'_\alpha}^{C^2_{\alpha,\beta}} (mod\ {N_2}^2),$$

where w_β is randomly chosen from $\mathbb{Z}^*_{N_2}$

return R

Location-based database (POI)

$$\begin{matrix} d_{i,j,1} \\ \cdots \\ d_{i,j,t} \\ \cdots \\ d_{i,j,m} \end{matrix} \rightarrow \begin{matrix} d^2_{i,j,1} \\ \cdots \\ \boxed{d^2_{i,j,t}} \\ \cdots \\ d^2_{i,j,m} \end{matrix} \rightarrow C_{i,j}$$

column j changes

row i changes

$$\begin{bmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,j} & \cdots & C_{1,n-1} & C_{1,n} \\ C_{2,1} & C_{2,2} & \cdots & C_{2,j} & \cdots & C_{2,n-1} & C_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ C_{i,1} & C_{i,2} & \cdots & C_{i,j} & \cdots & C_{i,n-1} & C_{i,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ C_{n-1,1} & C_{n-1,2} & \cdots & C_{n-1,j} & \cdots & C_{n-1,n-1} & C_{n-1,n} \\ C_{n,1} & C_{n,2} & \cdots & C_{n,j} & \cdots & C_{n,n-1} & C_{n,n} \end{bmatrix}$$

Response retrieval [solution 3]

Algorithm 9 Response Retrieval RR (User)

Input: $R = \{C_1, C_2, \cdots, C_n\}, sk$

Output: d

1: Compute

$$C'_j = PaillierDecrypt(C_j, sk_2).$$

where the decryption algorithm is described in the Paillier cryptosystem (please refer to Appendix A)

2: Compute

$$C''_j = RabinDecrypt(C'_j, sk_2).$$

where the decryption algorithm is described in the Rabin cryptosystem (please refer to Appendix B)

3: Compute

$$C'''_j = PaillierDecrypt(C''_j, sk_1).$$

4: Compute

$$d = RabinDecrypt(C'''_j, sk_1).$$

5: **return** d

$$E(r_1), \cdots, E(C^2_{i,j}), \cdots, E(r_n)$$

Security analysis

Theorems: If the Paillier cryptosystem is semantically secure, then our kNN query protocol without data privacy / with data privacy / based on POI type has location privacy.

Location privacy definition

2) $b \in \{1,2\}$
$(Q_b, s) = QG(CR, n, m, (i_b, j_b, t_b))$

1) given CR, n,m, choose (i_1,j_1,t_1), (i_2,j_2,t_2)

CR, n,m, (i_1,j_1,t_1), (i_2,j_2,t_2) ←

Q_b →

b’ ←

4) b=b’?

Mobile User

3) guess b’

LBS Provider
(adversary A)

$Adv_A(k) = |Prob(b' = b) - 1/2|$

Performance analysis

Component	*Algorithms 1-3*	*Algorithms 4-6*	*Algorithms 7-9*
User Comp.	$O(n)$	$O(n)$	$O(n+m)$
Server Comp.	$O(n^2)$	$O(n^2)$	$O(mn^2)$
Comm.	$2n \log_2 N$	$2n \log_2 N$	$(2n+m) \log_2 N$

Component	*Ghinita et al.*	*Paulet et al.*	*Our Protocol*
User Comp.	$O(n^2)/O(n)$	$O(1)$ / generate G, g, q and solve discrete log	$O(n)$
Server Comp.	$O(n^2)/O(n^2)$	$O(n)/O(n^2)$	$O(n^2)$
Comm.	$n^2 \log_2 N / 2n \log_2 N$	$2n \log_2 N / O(1)$	$2n \log_2 N$

Component	*Paulet et. al*	*Our Protocol*
Query Gen.	0.00484s / 9.6498s	0.157726s
Res. Gen.	0.11495s / 12.6978s	8.661929s
Res. Retrieval	0.0031s / 0.25451s	0.016211s

Conclusion

- location privacy issues
- survey on existing solutions
- three private kNN query protocols
- security analysis has shown that all of our protocols have location privacy
- performance has shown that our protocols are more efficient than previous PIR-based LBS query protocols. Experiment evaluation has shown that our protocols are practical
- our future work is to solve more complicated location-based queries


Q
&
A