

위치기반 서비스에서 개인 정보 보호를 위한 K-anonymity 및 L-diversity를 지원하는 Cloaking 기법

(Cloaking Method supporting K-anonymity and L-diversity
for Privacy Protection in Location-Based Services)

김지희* 이아름* 김용기** 엄정호** 장재우***
(Jihee Kim) (Ah-reum Lee)(Yong-Ki Kim) (Jung-Ho Um) (Jae-Woo Chang)

요약 최근 무선 인터넷에서 사용자의 위치정보가 다양한 애플리케이션의 정보 요소로 활용되기 시작하였다. 이러한 애플리케이션의 하나로 위치기반 서비스(Location-Based Service)가 주목을 받고 있다. 그러나 위치기반 서비스에서는 서비스를 요청하는 사용자가 자신의 정확한 위치 정보를 데이터베이스 서버로 보내기 때문에 사용자의 개인 정보가 노출될 수 있는 취약성을 지니고 있다. 이에 모바일 사용자가 안전하고 편리하게 위치기반 서비스를 사용하기 위한 개인 정보 보호 방법이 요구된다. 이를 위해 본 논문에서는 K-anonymity 및 L-diversity를 고려한 cloaking 기법을 제안한다. 이 기법은 먼저 사용자가 요구하는 L개의 건물을 탐색하는 L-diversity를 수행한 뒤, K명의 사용자를 탐색하는 K-anonymity를 수행하여 최소한의 넓이를 가지는 cloaking 영역을 생성한다. 이를 위해, R*-트리 기반의 색인 구조를 사용하며, 특히 효율적인 cloaking 영역 생성을 위해 필터링 방법을 사용한다. 마지막으로 성능평가를 통해 본 논문에서 제안한 cloaking 기법이 기존의 그리드 기법보다 우수함을 보인다.

키워드 : 개인 정보 보호, 위치기반 서비스, Cloaking 기법, L-diversity

Abstract In wireless internet, the location information of the user is one of the important resources for many applications. One of these applications is Location-Based Services (LBSs) which are being popular. Because, in the LBS system, users request a location-based query to LBS servers by sending their exact location, the location information of the users can be misused by adversaries. In this regard, there must be a mechanism which can deal with privacy protection of the users. In this paper, we propose a cloaking method considering both features of K-anonymity and L-diversity. Our cloaking method creates a minimum cloaking region by finding L number of buildings (L-diversity) and then finding number of users (K-anonymity). To support this, we use a R*-tree based index structure and use filtering methods especially for the minimum cloaking region. Finally, we show from a performance analysis that our method outperforms the existing grid based cloaking method.

Key words : Privacy Protection, Location-Based Services, Cloaking Method, L-diversity

1. 서론

최근 무선 인터넷에서 사용자의 위치정보가 다양한 응용의 정보 요소로 활용되기 시작하였다. 이러한 응용의 하나로 위치기반 서비스(Location-Based Service:LBS) [1]가 주목을 받고 있다. 위치기반 서비스는 크게 사용자가 요청한 서비스를 전송하는 PDA 또는 휴대 전화와 같

은 모바일 기기와 요청한 서비스를 처리하는 데이터베이스 서버로 구성된다. 만약 모바일 사용자가 교통 정보, 사람 찾기, 인접한 POI(Point Of Interest) 찾기, 현재 위치의 날씨 정보 등의 서비스를 요청하면, 모바일 기기는 그것에 장착된 위치추정 시스템을 이용하여 위치 정보를 획득한다. 그리고 사용자가 요청한 서비스 내용과 사용자의 위치정보는 WiFi와 같은 프로토콜 기반의 무선 네트

† 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신 인력 양성사업으로 수행된 연구 결과임

* 전북대학교 전자정보공학부 컴퓨터공학과 석사과정, gkim@chonbuk.ac.kr, arlee@dbl.chonbuk.ac.kr

** 전북대학교 전자정보공학부 컴퓨터공학과 박사과정, {ykkim, jhum}@dbl.chonbuk.ac.kr

*** 전북대학교 전자정보공학부 컴퓨터공학과 교수, jwchang@chonbuk.ac.kr(교신저자)

워크를 통해 데이터베이스 서버로 전송된다. 그러나 이러한 서비스들은 사용자의 개인 정보가 노출될 수 있는 취약성을 지니고 있다. 그 원인은 서비스를 요청하는 사용자가 자신의 정확한 위치 정보를 데이터베이스 서버에 보내기 때문이다. 이를 통해 상대방(adversary)은 서비스 이용자들이 어떤 장소에 자주 방문하는지, 또한 이러한 방문이 어떤 시간대에 주로 이루어지는지를 파악하여, 생활스타일, 질병 정보, 종교 등의 개인 정보를 획득할 수 있다. 실제로 국외의 경우 위치 기반 서비스를 이용한 스토킹 피해 사례가 빈번히 발생하고 있다[2, 3]. 따라서 모바일 사용자의 안전하고 편리한 위치기반 서비스 사용을 위한 개인 정보 보호 방법이 요구된다.

이러한 개인 정보 보호 방법과 관련된 기존 연구 중에는 cloaking 기법을 활용한 연구가 존재한다. Cloaking 기법이란, 사용자가 데이터베이스 서버에 질의(서비스)를 요청할 때, 사용자의 좌표정보가 아닌 K-anonymity를 만족하면서 최소한의 넓이를 가지는 질의 영역(이하 cloaking 영역)을 설정하여 보내는 것을 말한다. 여기서 K-anonymity는 cloaking 영역에 질의를 요청한 사용자와 그 사용자와 구별되지 않는 K-1명의 다른 사용자의 위치 정보를 포함하는 것이다. 그러나 K-anonymity만을 만족하는 cloaking 기법은 병원, 마트 등과 같이 한 건물 내에 cloaking 영역이 설정될 경우, 쉽게 사용자의 위치 추정이 가능하다는 문제점을 지니고 있다. 따라서 K-anonymity를 만족하는 동시에 L-diversity를 만족하는 cloaking 영역을 생성하여 질의를 요청한 사용자의 위치 정보 노출 확률을 감소시켜야 한다. 여기서 L-diversity란, K-anonymity와는 보완적으로 사용되며, cloaking 영역 내에 L개의 다른 장소(우편물이 배달될 수 있는 주소지 기준)를 포함하는 것을 말한다. 현재 cloaking 기법에 L-diversity를 적용한 기존 연구로는 B. Bamba와 L. Liu의 연구[4]가 유일하다. 그러나 이 연구는 그리드를 기반으로 cloaking 영역을 생성하기 때문에, L-diversity의 특성을 충분히 고려하지 못하여 cloaking 영역의 넓이가 증가하는 문제점을 가지고 있다. 따라서 본 논문에서는 위치기반 서비스에서 사용자의 위치 정보를 보호하기 위하여 K-anonymity 및 L-diversity를 고려한 cloaking 기법을 제안한다. 이 기법은 먼저 사용자가 요구하는 L개의 건물을 탐색하는 L-diversity를 수행한 뒤, K명의 사용자를 탐색하는 K-anonymity를 수행하여 최소한의 넓이를 가지는 cloaking 영역을 생성한다. 이를 위해, R*-트리[5] 기반의 색인 구조를 사용하며, 특히 최소 넓이 cloaking 영역 생성을 위해 필터링 방법을 사용한다.

본 논문의 구성은 다음과 같다. 2장에서는 K-anonymity 및 L-diversity를 고려한 기존 cloaking 기법을 소개한다. 3장에서는 기존 연구의 문제점을 개선한 K-anonymity 및 L-diversity를 지원한 cloaking 기법을 제안한다. 4장에서는 기존 연구와 본 논문에서 제안한 기법과의 성능비교를 수행한다. 마지막으로 5장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

최근 사용자의 위치 정보를 보호하기 위하여 연구된 대부분의 cloaking 기법들은 질의를 요청한 사용자의 좌표 정보를 K-anonymity를 만족하면서 최소한의 넓이를 가지는 cloaking 영역으로 변환하는 모델을 제안하고 있다[6, 7, 8, 9]. Z. Xiao et al. [6] 연구는 공간 영역 탐색 방법을 사용하여 cloaking 영역을 설정하며, 이 때 서버에 존재하는 사용자들의 이웃(neighbor)관계를 동적 그래프로 유지하는 clique를 사용한다. C. Y. Chow et al. [7] 연구는 질의를 요청한 모바일 사용자가 자신의 통신 범위 내에 있는 K-1 명의 다른 사용자들을 찾아 cloaking 영역을 생성한다. M. F. Mokbel et al. [8] 연구는 그리드 기반 피라미드 구조(grid-based pyramid structure)를 사용하여 cloaking 영역을 생성하는 기법을 제안하였다. G. Ghinita et al. [9] 연구는 힐버트 커브(Hilbert Curve)를 사용하여 구성된 계층적인 분산 B+ 트리를 통해 cloaking 영역을 생성한다. 그러나 K-anonymity만을 만족하는 cloaking 기법은 cloaking 영역이 병원 등과 같이 한 장소 내에 생성될 경우, 사용자의 위치가 쉽게 추정될 수 있는 문제점을 가지고 있다. 이와 같은 문제점을 해결하기 위하여 cloaking 영역 내에 L개의 다른 장소를 포함시키는 L-diversity가 고려되었고, 이는 K-anonymity와는 보완적으로 사용된다.

cloaking 기법에 L-diversity를 고려한 연구로는 B. Bamba와 L. Liu의 연구[4]가 유일하다. 이 연구에서는 K-anonymity와 L-diversity를 동시에 고려한 cloaking 기법인 Privacy Grid를 제안하였다. 이를 위해 다단계 그리드를 사용하였으며, 각 그리드 셀은 셀 내에 위치한 사용자의 수와 위치 그리고 건물 개수를 저장한다. 이 기법은 그리드 셀을 이용하여 cloaking 영역을 생성하기 때문에 cloaking 시간이 매우 빠른 장점을 가진다. Privacy Grid는 셀의 확장 방법에 따라 Quad-Grid, Bottom-Up, Top-down, Hybrid의 네 가지 방법으로 분류할 수 있다. Bottom-up cloaking 기법은 사용자가 요구한 최소 cloaking 영역 크기와 사용자의 위치를 기반으로 셀을 선택한다. 사용자가 위치한 셀의 열 방향으로 셀을 먼저 증가시킨 후 요구하는 K-anonymity와 L-diversity를 만족하는지 살펴본다. 만약 만족하지 못할 경우, 행 방향으로 셀을 증가시킨다. 이와 같은 방법을 사용자의 요구사항을 만족할 때까지 반복한다. Top-down cloaking 기법은 사용자가 요구한 최대 cloaking 영역 크기를 기반으로 셀들을 선택한다. 선택된 셀들이 K-anonymity와 L-diversity를 만족하는지 살펴본 후, 이를 초과하면 선택된 셀들을 먼저 행 단위로 삭제하고, 다음에는 열 단위로 삭제한다. 이를 사용자가 요구한 K-anonymity와 L-diversity를 만족할 때까지 행열 순서로 반복 수행한다. 한편, 가장 향상된 성능을 보이는 것은 Hybrid 방법으로, 사용자가 요구한 K-anonymity와 최대 cloaking 영역의 크기에 따라 Bottom-up과 Top-down 기법 중에서 하나를 선택하여 수행한다.

3. L-diversity를 이용한 cloaking 기법

본 장에서는 L-diversity를 cloaking 기법에 적용시킨 Privacy Grid[4]의 문제점을 기술하고, 그 문제점을 보완하여 K-anonymity 및 L-diversity를 고려하는 새로운 cloaking 영역 생성 알고리즘을 제안한다.

3.1 기존 cloaking 기법의 문제점

Privacy Grid의 cloaking 기법은 L-diversity 특성을 충분히 고려하지 못하기 때문에 생기는 두 가지 문제점이 존재한다. 첫째, 하나의 건물을 여러 그리드 셀에서 중복적으로 포함시키기 때문에, 생성된 cloaking 영역이 사용자가 요구하는 실제 L개의 건물을 만족하지 못한다.

Cell ID	Mobile Object Count	Static Object Count
1	1	1
2	1	1
3	0	1
4	0	1
5	2	1
6	1	1

(a) Cell Object Count Map

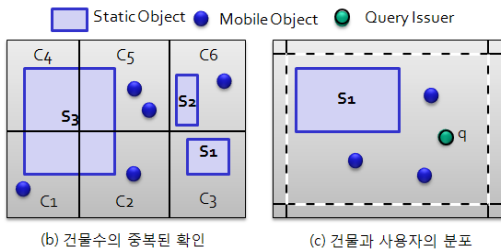


그림 1. Privacy Grid의 cloaking 기법

예를 들어, <그림 1-a>는 Privacy Grid에서 사용하고 있는 색인 구조 중 Cell Object Count Map 부분으로 각 셀 내에 위치한 사용자와 건물의 개수를 저장한다. 현재 이 Cell Object Count Map에는 <그림 1-b>의 상태가 저장되어 있으며, 각 셀 내에 사용자와 건물이 위치할 경우 그 개수(count)를 증가시킨다. 따라서 <그림 1-b>와 같이 건물 S3가 그리드 셀 C1, C2, C4, C5에 위치했을 경우, Cell Object Count Map에서 4개의 Static object count가 모두 하나씩 증가된다. 만약 어떤 사용자가 건물 4개를 포함하는 cloaking 영역을 요구했을 때 그 결과로 C1, C2, C4, C5의 셀이 선택되었다면, 이는 실제로는 하나의 건물이 포함된 cloaking 영역을 설정한 것이 된다. 따라서 L-diversity의 특성을 충분히 고려한 cloaking 영역을 설정하기 위해서는 하나의 건물을 영역 내에 중복적으로 포함시키지 않는 방법이 필요하다.

둘째, Privacy Grid는 cloaking 영역 내에 단순한 건물만 포함시키고, 건물 내에 위치한 사용자에 대해 고려하

지 않아 질의를 요청한 사용자의 위치 노출 확률을 높인다. 이러한 문제점은 상대방(adversary)이 cloaking 영역을 요구한 사용자의 정확한 위치는 모르지만, 모든 사용자의 위치를 알고 있을 경우 발생할 수 있다. 예를 들어 <그림 1-c>는 점선으로 표시된 한 셀 내에 건물 S1과 4명의 사용자가 위치한 모습을 나타낸 것이며, 건물 S1 내에는 실제 어떠한 사용자도 위치하지 않는다. 이 경우 만약 점선의 셀이 cloaking 영역으로 설정되었다면, 그 영역 내에 건물 하나가 포함되었다고 하더라도 Privacy Grid의 특성상 cloaking 영역을 요구한 사용자(q)가 건물을 제외한 도로 상에 위치한다는 것을 쉽게 추측할 수 있다. 이는 L-diversity의 특성을 충분히 고려하지 못한 것으로, 건물 내에 사용자를 분포시킨 상황을 가정함과 동시에 한명 이상의 사용자가 위치한 건물을 cloaking 영역에 포함시키는 방법이 필요하다.

3.2 설계 시 고려사항

본 논문에서 제안하는 cloaking 기법은 cloaking 영역을 생성하는 주체가 anonymizer인 중앙 집중(Centralized) 방식을 사용한다. Anonymizer란, 모바일 사용자와 LBS 서버 중간에 존재하는 신뢰할 수 있는 서버이다. 사용자는 anonymizer로 사용자의 위치 정보가 포함된 질의를 전송하고, anonymizer는 사용자의 위치 좌표를 숨기는 cloaking 영역을 생성한다. 생성된 cloaking 영역은 anonymizer에 의해 LBS 서버로 전송되며, LBS 서버는 cloaking 영역을 바탕으로 요청된 질의를 처리한다. 아울러 anonymizer는 LBS 서버에서 전송된 질의 수행 결과를 저장된 사용자의 위치 정보를 바탕으로 필터링 한 뒤, 정확한 결과를 사용자에게 전송한다. <그림 2>는 anonymizer를 사용하는 중앙 집중 방식을 나타낸다.

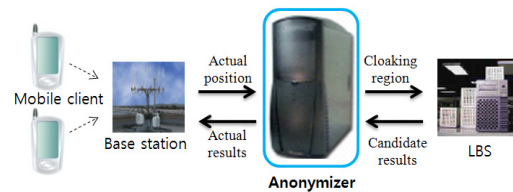


그림 2. 중앙 집중 방식

K-anonymity와 L-diversity를 cloaking 기법에 적용하는 방법으로 세 가지가 존재한다. 첫째, K-anonymity와 L-diversity를 동시에 고려하는 방법이 있다. 이는 Privacy Grid의 cloaking 기법에서 사용된 방법으로, 색인 구조에 저장된 그리드 셀 내의 사용자 수와 건물 수를 동시에 확인하면서 그리드 셀을 확장하여 cloaking 영역을 설정한다. 그러나 이러한 방법은 특히 그리드 기반 cloaking 기법 같은 경우, K와 L을 동시에 만족하지 않을 때 그리드 셀 단위로 cloaking 영역이 확장되기 때문에 넓이가 증가하는 문제점을 가지고 있다. 둘째, K-anonymity를 먼저 고려한 후 L-diversity를 고려하는 방

법이 있다. 그러나 일반적으로 질의 요청자가 cloaking 영역 내에 포함시키길 원하는 건물 수 L 보다 사용자 수 K 가 많기 때문에, 이 방법은 K 명의 사용자를 먼저 찾는데 많은 시간이 소요될 수 있다. 마지막으로 L -diversity를 먼저 고려한 후 K -anonymity를 고려하는 방법이 있다. Cloaking 영역이 L -diversity만 만족하더라도 그 안에 여러 개의 건물이 포함되어 다수의 사용자가 위치할 수 있기 때문에 이 방법은 K -anonymity까지 고려할 필요 없이 cloaking 설정이 종료되는 경우가 발생하여 cloaking 시간을 단축시킬 수 있다. 따라서 본 논문에서는 L -diversity를 먼저 고려하여 임시 cloaking 영역을 설정한 후, K -anonymity를 고려하는 방법을 사용한 cloaking 기법을 제안한다. 이를 위해 두 개의 이차원 R^* -트리를 사용하며, 한 건물의 중복된 고려 없이 정확한 L -diversity 알고리즘을 수행할 수 있다. <그림 3>에서 첫 번째 R^* -트리는 L 개의 건물을 찾기 위한 L -diversity 색인 구조로써 건물들의 위치를 저장 및 색인한다. 두 번째 R^* -트리는 K 명의 인접한 사용자를 찾기 위한 K -anonymity 색인 구조로써 전체 사용자들의 위치를 저장 및 색인한다.

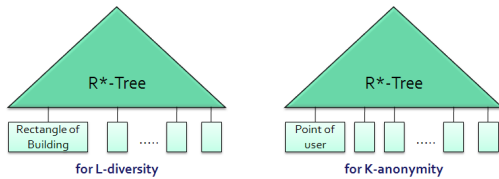


그림 3. 제안하는 방법의 색인 구조

3.3 L-diversity를 만족하는 cloaking 영역 설정 알고리즘

본 절에서는 L -diversity를 만족하는 cloaking 영역 생성 알고리즘을 위한 수행 단계를 제시한다. 먼저, 수행단계 1은 사용자가 요구하는 건물 수가 2 이상으로 cloaking 영역을 요구할 때, L -diversity를 수행하기 위하여 질의를 요청한 사용자 q 의 위치에 따라 인접한 L 개의 건물을 탐색하는 방법을 제시한다.

수행단계 1. 사용자가 요구한 L 개의 건물 선택 ($L \geq 2$)

$$\begin{cases} \text{if } q(x, y) \in R_b \\ \quad LNNsearch(L-1, q) \\ \text{otherwise} \\ \quad LNNsearch(L, q) \end{cases}$$

여기서 $q(x, y)$ 는 Cloaking 영역을 요구한 사용자 q 의 위치를, R_b 는 건물의 영역을 의미한다. $LNNsearch(L, q)$ 는 건물의 영역을 색인하는 R^* -tree에서 질의 지점 q 로부터 L 개의 인접한 건물을 탐색하는 함수이다. 수행단계 1은 Cloaking 영역을 요구한 사용자 q 가 건물 내에 위치 시에는 그 건물을 제외한 $L-1$ 개의 인접한 다른 건물들

검색하고, 만일 q 가 건물 외부에 위치 시에는 L 개의 인접한 다른 건물들 검색하는 것을 의미한다. 예를 들어 $L=3$ 일 때, <그림 4-a>는 q 가 건물 내부에 위치한 경우로 q 가 속한 건물을 제외한 2개의 인접한 건물들 검색한다. <그림 4-b>는 q 가 건물 외부에 위치한 경우로 q 와 인접한 3개의 건물들 검색한다.

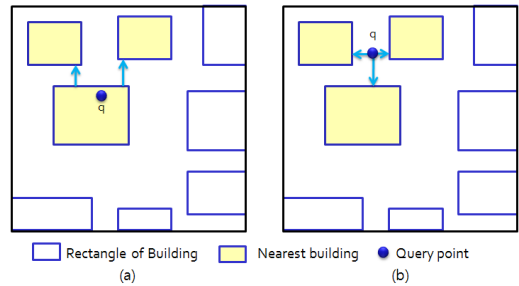


그림 4. 사용자가 요구한 L 개의 건물을 선택하는 방법

사용자가 요구한 L 개의 건물을 선택하면, 그것을 포함하는 cloaking 영역은 2개 이상의 건물을 포함하는 넓은 영역이 된다. 따라서 선택된 L 개의 건물을 포함하는 임시 cloaking 영역을 생성하여, 그 영역 내에 위치한 사용자들의 수를 확인할 필요가 있다. 또한 임시 cloaking 영역을 생성할 때는 L -diversity의 특성을 충분히 반영하지 못한 기존 연구의 두 번째 문제점을 보완해야 한다. 이를 위해, L 개의 각 건물 내에 위치한 사용자 중에서 최소한 m 이상을 cloaking 영역 내에 포함시킨다. 수행단계 2에서 임시 cloaking 영역의 설정은 L 개의 인접한 건물 내에 위치한 사용자들 중에서 cloaking 영역을 요구한 사용자 q 와 가장 가까운 거리에 있는 사용자를 각 건물에서 한명씩 검색하여 생성한다.

수행단계 2. 임시 cloaking 영역 설정

$$TempCloakArea = MBR(U)$$

$$U = \{u_i \mid \exists u_i \in R_{b_i} \text{ and } \exists u_i = NN(q) \text{ and } 1 \leq i \leq L\}$$

여기서, $MBR(U)$ 는 사용자 집합 U 를 포함하는 최소 경계사각형(Minimum Bounding Rectangle:MBR)을 생성하는 함수이고, R_{b_i} 는 i 번째 건물의 영역을 의미한다. 아울러, $NN(q)$ 는 질의지점 q 에서 최근접점을 의미한다. 수행단계 2를 통해 설정된 임시 cloaking 영역은 요구된 L 개의 건물을 포함하면서 최소한의 넓이를 가지는 영역이 된다.

를 들어, <그림 5>는 L 이 3 이고 질의 요청자 q 가 건물 내부에 위치할 경우, 임시 cloaking 영역을 설정한 상황을 나타낸다. R^* -트리를 이용한 최근접점 탐색을 통해 건물 $L2, L3$ 에서 q 와 가까운 사용자 $p1, p2$ 가 각각 선택되고, 사용자 $q, p1, p2$ 의 최소경계사각형인 점선 사각형이 임시 cloaking 영역으로 설정된다.

임시 cloaking 영역 내에 존재하는 사용자의 수의 검사

를 통해 cloaking 영역 생성 알고리즘은 확장되거나 종료된다. 수행단계 3은 사용자 수에 따른 cloaking 영역 생성 알고리즘의 종료 또는 확장 조건을 나타낸다.

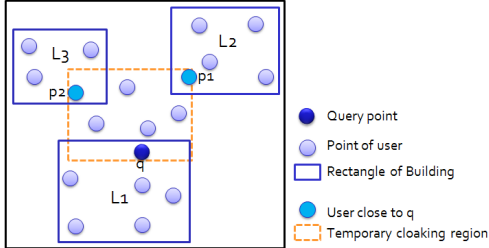


그림 5. L=3 일 때, 임시 cloaking 영역을 설정한 경우

수행단계 3. 임시 cloaking 영역 내의 사용자 수에 따른 cloaking 영역 확장

```

if (k' ≥ k)
    return TempCloakArea
otherwise
    execute ExpandCloakingArea(TempCloakArea, K - K')
    
```

여기서 ExpandCloakingArea() 함수는 K-anonymity를 만족하는 cloaking 영역 확장 알고리즘을 의미하며, 본문 3.4절에 기술하였다. 예를 들어, <그림 5>의 경우 K = 7 이라면, 영역 질의를 통한 탐색으로 임시 cloaking 영역 내에 질의의 요청자를 포함한 7명의 사용자가 확인된다. 따라서 K' = K를 만족하여, 임시 cloaking 영역이 결과 cloaking 영역으로 반환되면서 알고리즘은 종료한다. <그림 6>은 반대의 경우로 K'가 4명으로 K'가 K보다 적은 경우로써, 이 때 부족한 3명의 사용자를 찾기 위한 확장 알고리즘을 수행하게 된다.

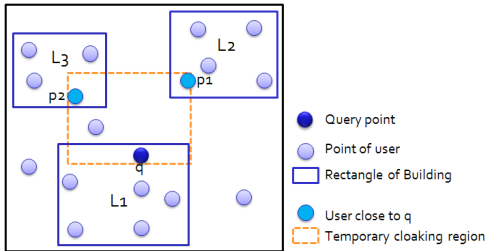


그림 6. K=7 일 때, K'=4로 알고리즘 확장이 필요한 경우

앞에서의 정의를 바탕으로 설계한 L-diversity를 만족하는 cloaking 영역 설정 알고리즘은 <그림 7>과 같다. 먼저 cloaking 영역을 요청하는 사용자의 위치정보와 cloaking 영역 내에 포함되길 원하는 건물 수 L과 사용자 수 K를 입력받는다. 다음으로 질의를 요청한 사용자가 위치한 곳이 건물 내부인지 외부인지를 확인하여, 그 상황에 맞는 L개의 인접 건물을 탐색한다. 탐색된 건물

내의 사용자 중에서 질의의 요청자와 가까운 사용자를 검색하여, 그들을 포함하는 최소경계사각형을 임시 cloaking 영역으로 설정한다. 마지막으로 임시 cloaking 영역 내의 위치한 사용자 수를 검사하여, 그 수가 요구된 K보다 많거나 같을 경우 임시 cloaking 영역을 결과 cloaking 영역으로 반환하고 알고리즘을 종료한다. 그러나 그 수가 요구된 K보다 적을 경우, K를 만족시키는데 부족한 사용자를 찾기 위한 확장 알고리즘을 수행한다.

Cloaking 알고리즘 (질의 요청자 q의 좌표정보, K, L)

//입력: L(cloaking 영역 내에 포함시킬 건물의 수), K(cloaking 영역 내에 포함시킬 사용자 수)

//출력: L과 K를 만족하는 cloaking 영역

- q가 위치한 곳이 건물 내부인지 외부인지 확인한다.
 - if, 건물 내부라면, q가 위치한 건물을 제외한 L-1개의 인접한 건물을 검색한다.
 - else, 건물 외부라면, q와 인접한 L개의 건물을 검색한다.
- 선택된 L개의 건물 내에 위치한 사용자들 중 q와 가장 가까운 사용자를 한명씩 선택한다.
 - if, q의 위치가 건물 내부라면, q와 L-1개의 건물에서 위치한 사용자를 선택한다.
 - else, L개의 건물에 위치한 사용자를 한명씩 선택한다.
- 선택된 사용자들을 포함하는 최소경계사각형을 임시 cloaking 영역으로 설정한다.
- 임시 cloaking 영역으로 범위 질의를 수행하여 그곳에 위치한 사용자 수 K'를 검사한다.
 - if, K' ≥ K일 경우, 임시 cloaking 영역을 결과로 반환하고 알고리즘을 종료한다.
 - else, K' < K일 경우, (K-K')명의 부족한 사용자를 찾는 확장 알고리즘을 수행한다.
 - 4.2.1 Cloaking 영역 확장 알고리즘(3.4절)을 호출한다.

그림 7. L-diversity를 만족하는 cloaking 영역 설정 알고리즘

3.4 K-anonymity를 만족하는 cloaking 영역 확장 알고리즘

본 절에서는 L-diversity를 만족하는 임시 cloaking 영역이 K-anonymity를 만족하지 않을 경우, 부족한 K-K'(이하 R로 명명)명의 사용자를 찾는 확장 알고리즘을 위한 수행단계를 제시한다. 먼저, 부족한 R명의 사용자를 찾기 위하여 임시 cloaking 영역을 기준으로 동서남북 방향의 영역에 대한 R명의 인접한 사용자들을 검색하여 후보 집합을 구성한다. 수행단계 4는 각 방향에 대해 R명의 인접한 사용자를 검색하는 방법을 정의한다.

수행단계 4. 각 방향에서 R명의 인접한 사용자 검색 cloaking 영역에서 8방향

$$D_i = \{N, NE, E, ES, S, SW, W, NW\}, 0 \leq i < 8$$

이라 할 때 R명의 가까운 거리를 가진 사용자의 검색은 다음과 같다.

```

if(i=0)
  execute LNNsearch(Di%2)
else(i=1)
  execute LNNsearch(Di%2)
    
```

여기에서 *LineLNNsearch(direction)*은 영역의 direction에 해당하는 방향의 선분에서 제일 가까운 사용자를 찾는 함수이고, *LNNsearch(direction)*은 영역의 direction에 해당하는 꼭지점에서 제일 가까운 사용자를 찾는 함수이다. 수행단계 4와 같이 검색하는 이유는 다음과 같다. 예를 들어 임시 cloaking 영역의 동쪽 방향에 대해 가까운 거리의 사용자를 검색할 경우, 고려해야 하는 범위는 북동, 동, 동남 방향이 된다. 이 때 사용자까지의 거리를 구하기 위한 기준을 임시 cloaking 영역의 동쪽 선분으로만 설정할 경우, 북동, 동, 동남 방향 사용자들 중에 가장 짧은 거리를 가지는 R명의 사용자가 검색된다. <그림 8>과 같은 상황에서 동쪽 방향으로 가장 가까운 사용자 한명을 구한다고 하자. 이 때 임시 cloaking 영역의 동쪽 선분만을 기준으로 북동, 동, 동남 방향을 하나의 영역으로 생각하여 탐색하면 사용자 u2보다 짧은 거리를 갖는 u1이 선택된다. 그러나 각 사용자를 선택하였을 때 확장되는 cloaking 영역의 크기는 u1은 10, u2는 6으로 u2가 작다(셀 한 칸의 크기는 1로 가정). 이와 같이 최적의 후보가 될 가능성을 가진 사용자들의 후보 집합 탈락을 방지하기 위하여, 네 선분과 네 꼭지점을 기준으로 8 방향에 대한 R명의 사용자 검색이 필요하다.

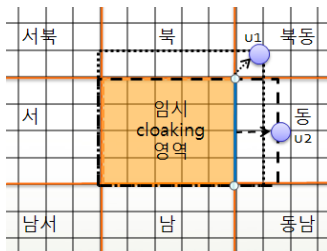


그림 8. 동쪽 방향에 대한 후보 선택

수행단계 4를 통해 구해진 후보 집합에 대한 필터링을 통해 최적의 조합을 구하기 위한 계산량을 줄일 수 있다. 수행단계 5와 6은 각 방향에 대해 후보 집합을 필터링하는 두 가지 방법을 정의한다.

수행단계 5. 각 방향의 후보 집합에 대한 길이 필터링

한 방향에서 선택할 사용자의 수가 X이고, 현재 고려하고 있는 방향의 X번째 사용자 u_X 가 기준 사용자일 때, 현재 사용자가 u_i 라면 u_i 는 다음의 식을 만족해야 한다.

단, u_i 의 위치가 cloaking 영역의 D_{cur} 방향에 위치한다고 가정한다.

$$DIST(LINE(D_{cur}), u_i) \leq DIST(LINE(D_{cur}), u_X)$$

즉, 수행단계 5는 기준 사용자 u_X 의 x, y 좌표 중에서 현재 고려하고 있는 방향(동서남북 중 한 방향)과 같은 축의 값을 한계 축으로 설정하고, 이를 이용해 같은 방향의 후보들 중 한계 축과 값이 같거나 그 범위를 넘는 사용자는 후보 집합에서 제외시키는 것을 의미한다.

수행단계 6. 각 방향의 후보 집합에 대한 넓이 필터링

현재 사용자 u_i 의 현재 고려하는 방향에 대한 영역 넓이 $Area_i$ 는 다음과 같다.

$$Area_i = DIST(LINE(D_{cur}), u_i) * (Length((LINE(D_{cur})) + Project(LINE(D_{cur}), U_i)))$$

길이 필터링의 기준 사용자 u_X 의 현재 고려하는 방향에 대한 영역의 넓이 $Area_X$ 는 다음과 같다.

$$Area_X = Length(LINE(D_{cur})) * DIST(LINE(D_{cur}), u_X)$$

현재 사용자 u_i 의 영역 넓이가 다음의 식을 만족하지 않을 경우 후보 집합에서 제외된다.

$$Area_i \leq Area_X$$

즉, 수행단계 6은 한계 넓이보다 큰 확장 넓이를 가지는 사용자를 후보 집합에서 제외하는 것을 의미한다. 여기에서 *Length()* 함수는 선분의 길이를 의미한다. 아울러 *Project()* 함수는 현재 고려하고 있는 방향 D_{cur} 에 대해 사용자 u_i 를 투영시켰을 때, *LINE(D_{cur})*와 차이가 나는 선분의 길이를 의미한다.

예를 들어, <그림 9-a>는 임시 cloaking 영역(색칠된 부분)에서 부족한 3명의 사용자를 찾기 위해 정의된 4의 방법을 통하여 8방향에서 3명의 인접 사용자들을 검색한 모습이다. 각 사용자들은 각 영역에 대해 길이에 대한 오름차순으로 정렬되어 있다. 만약, 동쪽 방향에서 3명을 모두 선택하는 경우라면, u1에서 u9까지 9명의 사용자가 최초의 후보 집합이 된다. 먼저, 후보 집합의 수를 줄이기 위하여 길이에 대한 필터링을 수행한다. 동쪽 방향의 3번째 근접 사용자인 u6가 기준으로 선택되고 고려하는 방향인 동쪽 방향과 같은 축인 u6의 x축 값이 한계 축으로 설정된다. 따라서 그 이상의 값을 갖는 사용자들은 후보 집합에서 제외되며, 이를 통해 <그림 9-b>와 같이 사용자 u2와 u8이 후보 집합에서 제외된다. 다음으로 넓이 필터링 시에는 기준 사용자 u6의 확장 넓이(색칠된 부분)가 한계 넓이로 설정된다. 현재 고려하는 방향이 동쪽이므로, 임시 cloaking 영역의 동쪽 선분을 기준으로 각 후보들의 확장 넓이(각각의 점선 사각형들)를 구한다. 후보들의 확장 넓이가 한계 넓이보다 클 경우 그 사용자가 후보 집합에서 제외되며, 그 결과 사용자 u3이 제외된다. 결과적으로 두 필터링 방법을 통하여 세 명의 후보가 제외됨으로서, 동쪽방향에서 3명의 후보 선택 시 조합하는 경우의 수가 $9C3 = 84$ 에서 $6C3 = 20$ 으로 감소된다.

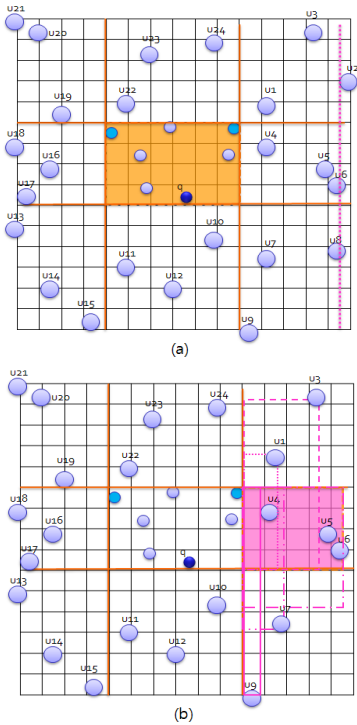


그림 9. 후보 집합에 대한 두 가지 필터링 방법

일반적으로 사각형의 넓이가 가장 작게 증가하는 경우는 <그림 10-a>와 같이 한쪽 방향으로만 넓이를 확장시킨 경우이다. 그 다음은 <그림 10-b>와 같이 한쪽의 대각선 방향으로 늘어나는 경우이며, <그림 10-c>와 같이 세면이 늘어나는 경우, 마지막으로 <그림 10-d>와 같이 네 면이 동시에 늘어나는 경우의 순서로 확장 넓이가 증가하게 된다. 이를 이용하여 수행단계 7은 각 방향에 대한 후보 집합이 구해졌을 때, 이들 중에서 가장 최소의 넓이를 가지는 R명의 조합을 구하는 방법을 정의한다.

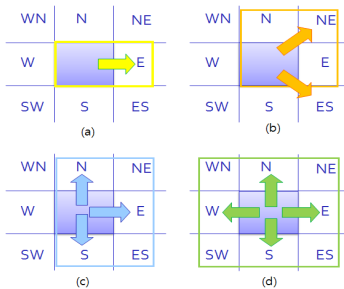


그림 10. 일반적인 넓이의 확장

수행단계 7. 가장 최소의 넓이를 가지는 R명의 조합을 구하는 방법

먼저, 한쪽 방향에서 R명을 모두 선택함을 가정하고 최소 넓이를 가지는 조합을 구한다. 그 후 한쪽 방향에서 선택하는 사용자 수를 한명 감소시키고, 대신 중복되지 않는 다른 쪽 방향으로 한명을 증가시켜 조합을 구한다. 만약 기준이 되는 방향보다 다른 방향의 사용자가 많이 선택되는 경우는 조합을 구하지 않는다. 이것을 R명에 대해 선택할 수 있는 모든 경우를 만족할 때까지 수행한다.

실제로 최적의 조합을 구하기 위해서는 후보집합에 대해 R명의 조합이 나올 수 있는 모든 경우를 고려해야 한다. 그러나 최소 한계값이 일찍 설정될수록 모든 경우를 계산하지 않고 많은 경우에 대해 가지치기를 할 수 있다. 따라서 수행단계 7을 이용하면 모든 경우를 계산하면서도 지속적으로 최소 한계 값이 갱신되기 때문에 알고리즘 속도를 향상시킬 수 있다.

A	B	C	D
4	0	0	0
3	1	0	0
2	2	0	0
2	1	1	0
1	3	0	0
1	2	1	0
1	1	1	1

동	서	남	북
4	0	0	0
0	4	0	0
0	0	4	0
0	0	0	4

그림 11. R=4 일 때, 나올 수 있는 조합

<그림 11>의 오른쪽 그림은 R이 4일 때 후보를 선택할 수 있는 경우를 나타낸다. 점선 밑줄이 그어진 경우는 기준이 되는 A방향보다 다른 방향의 사용자가 많이 선택된 경우로, 이전의 경우에서 이미 계산된 경우이기 때문에 중복 계산하지 않는다. 왼쪽 그림은 하나의 경우에 대해 기준을 바꾼 경우를 나타낸다. 결과적으로 R이 4일 때 조합을 구하는 모든 경우는 $5 \times 4 = 20$ 가지이다.

만약 검색해야 하는 사용자의 수가 10명 이상이 될 경우, 너무 많은 수의 조합을 계산해야 하기 때문에 알고리즘이 비효율적이 될 수 있다. 정의 6은 이러한 상황일 때 효율적인 cloaking 영역을 계산할 위한 알고리즘을 정의한다.

수행단계 8. R이 10이상일 경우 알고리즘

$$width_i = width_{i-1} + (width_{i-1} \times \alpha)$$

$$height_i = height_{i-1} + (height_{i-1} \times \alpha)$$

$$Area_i = width_i \times height_i$$

if (User(Area_i) ≥ k) return

수행단계 8은 임시 cloaking 영역의 가로, 세로 길이를 각각 자신의 α 배씩 (0 < α ≤ 1) 증가시켜 K'가 K' ≥ K를 만족할 때까지 영역을 확장하는 것을 의미한다. 여기에서 Area는 현재 cloaking 영역을 User(Area_i) 함수는 해당 Area에 포함된 사용자의 수를 반환한다.

지금까지의 정의를 바탕으로 설계한 K-anonymity를 만족하는 cloaking 영역 확장 알고리즘은 <그림 12>와 같

다. 먼저 L-diversity만 고려하여 설정된 임시 cloaking 영역과 부족한 사용자 수 R을 입력받는다. 만약 검색해야 하는 사용자의 수 R이 10 이상일 경우, 수행단계 8을 통하여 검색할 사용자의 수를 10미만으로 줄인다. R이 10 미만일 경우, 8 방향에 대한 R-최근접점 탐색을 사용하여 후보 집합을 생성하고, 이에 대해 길이 및 넓이 필터링을 수행한다. 필터링 되어진 후보 집합에 대해, R명을 선택될 수 있는 모든 경우를 비교하며 최소한의 확장 넓이를 가지는 최적의 사용자 조합을 구한다. 마지막으로 그들을 포함하는 최소경계사각형을 결과 cloaking 영역으로 반환한다.

```

Cloaking 영역 확장 알고리즘(임시 cloaking 영역, R)
//입력: R = K-K'
//출력: 최소한의 넓이를 가지는 cloaking 영역
1. if, R≥10일 경우, R이 10미만이 될 때까지 임시 cloaking 영역을 확장한다.
2. else, R<10일 경우,
  2.1 8방향에 대한 R-최근접점을 탐색을 통해 동서남북 방향에 대한 후보 집합을 생성한다.
  2.2 생성된 후보 집합에 대해 길이 및 넓이 필터링을 수행한다.
  2.3 필터링된 후보 집합에 대해 최적의 조합을 구한다.
  2.4 최소한의 확장 넓이를 가지는 사용자 조합을 포함하는 최소한계영역을 결과로 반환한다.
  
```

그림 12. K-anonymity를 만족하는 cloaking 영역 확장 알고리즘

4. 구현 및 성능평가

본 논문에서 제안하는 L-diversity를 지원하는 cloaking 영역 생성 알고리즘을 Microsoft Visual Studio.NET 2003으로 작성하였으며, Intel Core2 Quad CPU Q6600 2.40GHz와 Ram 2GB, Windows XP 상에서 구현하였다. 성능평가에 사용된 데이터는 GSTD(Generate Spatio Temporal Data) 알고리즘[10]을 이용하였으며, 성능평가를 위하여 1,000개의 사각형 건물과 10,000명의 사용자 포인트를 생성하였다. GSTD 알고리즘은 가로 및 세로의 범위가 1인 사각형 내에 객체를 생성하는 방식을 사용하기 때문에, 실험에서 cloaking 영역의 최대 크기는 1로 설정된다. 아울러, 제안하는 기법은 R*-트리 기반 색인구조를 사용하기 때문에 데이터를 삽입하는 단계에서 두 개의 R*-트리를 구축하는 시간이 요구된다. 건물의 영역을 위한 R*-트리 구축에는 0.078초, 사용자 위치를 위한 R*-트리 구축에는 1.125초가 소요된다. 그러나 데이터의 삽입에 따른 트리 구축은 제안하는 cloaking 기법의 수행 전에 이루어지기 때문에, cloaking 영역 생성 속도에는 영향을 미치지 않는다.

성능평가 대상은 Privacy Grid에서 우수한 성능을 보

이는 Bottom-up(이하 그리드로 명명) cloaking 기법을 구현하여 비교하였으며, 그리드는 크기는 가로와 세로가 0.01일 때와 0.005의 두 경우를 사용하였다(이하 그리드(0.01), 그리드(0.005)로 명명). 그러나 이 기법은 하나의 건물을 여러 그리드 셀에서 중복적으로 포함시키기 때문에 사용자가 요구하는 L-diversity를 정확히 만족시키지 않는다. 따라서 정확한 성능평가를 위하여 L개의 건물을 찾는 단계에서 한 건물의 중복된 확인이 발생하지 않도록 알고리즘을 수정하여 성능을 비교하였다. 아울러 임시 cloaking 영역을 확장할 때 사용하는 변수 a는 실험을 통해 0.2로 설정하였다. 성능평가 항목은 K를 20으로 고정시키고, L을 2, 4, 6, 8, 10로 증가시키면서 cloaking 영역 크기 및 cloaking 영역 생성 속도(wall time)를 측정하였다.

<그림 13>은 본 논문에서 제안하는 기법과 그리드 기법의 cloaking 영역 크기를 비교한 것이다. 두 방법 모두 L이 커질수록 cloaking 영역 크기가 증가하는 모습을 보인다. 결과적으로 제안한 기법은 모든 경우를 고려하여 K 및 L을 만족하는 최적의 cloaking 영역을 생성하기 때문에, 그리드 기법이 제안한 기법보다 cloaking 영역 크기가 크게 나타남을 확인할 수 있었다. 제안한 기법을 기준으로 그리드(0.01), 그리드(0.005)는 제안한 기법보다 L이 2일 때 각각 138%, 23% 증가된 영역을 설정하였고, L이 4일 때는 151%, 35%, L이 6일 때는 93%, 16%, L이 8일 때는 56%, 6%, 마지막으로 L이 10일 때는 각각 37%, 1% 증가된 영역을 설정하였다. 평균적으로 cloaking 영역 크기는 그리드(0.01)은 94%, 그리드(0.005)는 16%의 증가율을 보인다. 특히 그리드 기법의 경우에는 그리드 크기가 작을수록 cloaking 영역 확장 탐색 시 영역을 미세하게 확장하기 때문에 최적의 크기에 가까워진다.

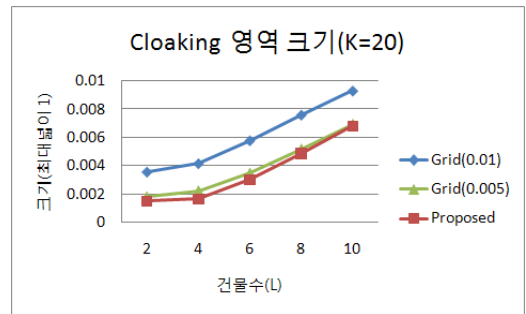


그림 13. 제안하는 기법과 그리드 기법의 cloaking 영역 크기

<그림 14>는 <그림 13>의 제안하는 기법과 그리드(0.005)의 cloaking 영역 크기를 비교한 것이다. 제안하는 기법은 항상 최적의 cloaking 영역을 생성하기 때문에, 이를 기준으로(최적의 값을 1로 설정) 그리드(0.005)와 영역 크기 증가율을 비교하였다. 즉, 그리드(0.005)의

cloaking 영역 크기는 L이 2일 때, 제안하는 기법의 1.23배, L이 4일 때는 1.35배, L이 6일 때는 1.16배, L이 8일 때는 1.06배, L이 10일 때는 1.01배 증가된 영역을 설정하였다. 따라서 L의 크기가 커질수록 cloaking 영역 크기 증가율이 감소하지만 제안하는 기법보다 cloaking 영역의 크기가 평균 16% 정도 크게 설정됨을 알 수 있다. 이러한 차이는 POI 밀도가 높은 도심지의 경우 중요한 변수로 작용될 수 있다. 즉, 도심지의 경우, LBS에서 사용자의 질의 처리 시 전송된 cloaking 영역의 작은 크기 차이에도, 질의 결과 후보 집합의 개수가 크게 달라지기 때문이다. 특히 후보 집합이 많을수록 질의 서버에서 anonymizer로 후보 집합을 전송하기 위한 통신비용이 증가하고 anonymizer에서 필터링을 위한 오버헤드가 증가하기 때문에, 이를 방지하기 위해 최적의 cloaking 영역(즉, K와 L을 만족하는 최소 크기의 영역)을 설정하는 것이 중요하다.

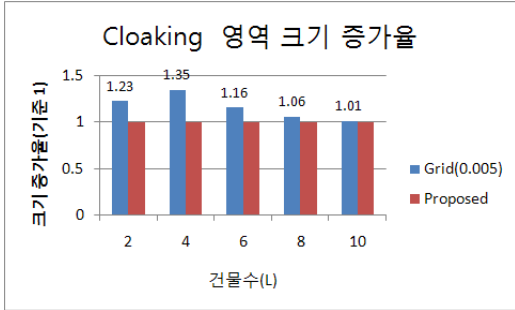


그림 14. 제안하는 기법을 기준으로 한 그리드(0.005)의 cloaking 영역 크기 증가율

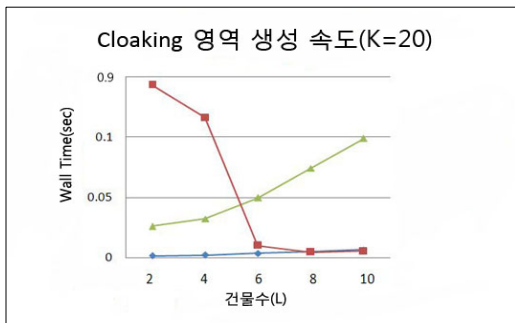


그림 15. 제안한 알고리즘과 그리드 기법의 cloaking 영역 생성 속도

<그림 15>는 제안한 기법과 그리드 기법의 cloaking 영역 생성 속도를 비교한 것이다. 제안한 기법은 L의 값이 증가할수록 영역 생성 속도가 빨라지는 모습을 보였다. 특히 L이 10 일 때는 0.005초의 시간이 걸려 그리드 방법 중 빠른 속도를 보이는 그리드(0.01)의 0.007초보다도 빠른 속도를 보였다. 이와 같은 성능을 보이는 이유

는, L의 값이 증가할수록 임시 cloaking 영역 내에 많은 건물이 포함되어 영역 크기가 커지기 때문이다. 대부분의 경우가 $K' \geq K$ 를 만족해 확장 알고리즘을 수행하지 않고 알고리즘이 종료된다. 반면에 그리드 기법은 L이 증가할수록 cloaking 속도가 저하되는 모습을 보였다. 특히, cloaking 영역 넓이 실험에서 그리드 기법 중 좋은 성능을 보였던 그리드(0.005)는 L이 6 일 때 cloaking 시간이 0.04초로, 제안한 방법의 0.008초보다 약 5배 느려지고, L이 10일 때는 제안한 기법이 0.005초, 그리드(0.005)가 0.09초로 약 18배까지 느려진다. 그러나 L이 4 이하인 경우, 제안하는 기법은 임시 cloaking 영역 내에 사용자가 원하는 K명을 포함하지 못하는 경우가 많이 발생하여, R*-트리의 탐색 및 확장 알고리즘의 수행으로 cloaking 영역 생성 시간이 증가한다. Cloaking 기법의 목적이 개인 정보의 노출을 최대한 감소시키는 것이기 때문에, 제안한 기법은 L이 6 이상일 때와 같이 높은 수준의 개인 정보 보호를 요구하는 응용에 적합하게 사용될 수 있다.

성능비교를 통해, 그리드 기법은 그리드 크기에 따라 cloaking 영역 크기와 cloaking 속도가 상충된 관계를 가지기 때문에, 두 조건을 모두 만족하면서 cloaking 영역을 생성할 수 없다. 그러나 제안한 기법은 최소 cloaking 영역 크기와 최소한의 cloaking 시간을 만족하면서 cloaking 영역을 생성할 수 있다.

5. 결론 및 향후연구

본 논문에서는 위치기반 서비스에서 개인의 위치 정보를 보호하기 위한 새로운 cloaking 기법을 제안하였다. 아울러 Privacy Grid의 L-diversity를 충분히 고려하지 못한 cloaking 기법의 문제점을 보완하였다. 제안한 기법은 먼저 사용자가 요구하는 L개의 건물을 탐색하는 L-diversity를 수행한 뒤, K명의 사용자를 탐색하는 K-anonymity를 최소한의 넓이는 가지는 cloaking 영역을 생성한다. 이를 위해, R*-트리 기반의 색인 구조를 사용하며, 특히 최소한의 cloaking 영역 설정 알고리즘을 위해 필터링 방법을 사용하였다. 기존 그리드 기법과의 성능비교를 통해 제안한 기법이 최소 cloaking 넓이와 최소한의 cloaking 시간을 만족하는 효율적인 cloaking 기법이라는 것을 보였다. 향후 연구로는 분산 환경에서 K-anonymity와 L-diversity를 고려한 cloaking 기법을 연구하는 것이다.

참고 문헌

[1] 이준석, 김서균, “위치기반서비스(LBS)의 기술 동향 및 국내외 산업 동향 분석,” 정보통신연구진흥원 계간 제 5권 제 2호 (통권 16호), 2003.

[2] Foxs News. Man Accused of Stalking Ex-Girlfriend With GPS, <http://www.foxnews.com/story/0,2933,131487,00.html>, 2004.

[3] USA TODAY News, GPS system used to stalk woman,
http://www.usatoday.com/tech/news/2002-12-30-gps-stalker_x.htm, 2002.

[4] B. Bamba and L. Liu, "PRIVACYGRID: Supporting Anonymous Location Queries in Mobile Environments", Research report in National Technical Information Service, 2007.

[5] N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," In Proc. of SIGMOD, 1990, pp. 322-231.

[6] Z. Xiao, X. Meng and J. Xu, "Quality Aware Privacy Protection for Location-based Services," In Proc. of Database Systems for Advanced Applications, vol.4443, April 2007, pp. 434-446.

[7] C. Y. Chow, M. F. Mokbel, and X. Liu. A, "Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services," In Proc. of the ACM International Symposium on Advances in Geographic Information Systems, November 2006, pp. 171 - 178.

[8] M. F. Mokbel, C. Chow, and W. Aref, "The New Casper : Query Processing for Location Services without Compromising Privacy," In Proc. of the International Conference on Very Large Data Bases, September 2006, pp. 763 - 774.

[9] G. Ghinita, P. Kalnis and S. Skiadopoulos, "PRIVE : Anonymous Location-Based Queries in Distributed Mobile Systems," In Proc. of World Wide Web, May 2007, pp. 237-246.

[10] Yannis Theodoridis, Jefferson R. O. Silva, and Mario A. Nascimento, "On the Generation of Spatiotemporal Datasets", 6th Int'l Symposium on Large Spatial Databases (SSD), 1999.

[11] 엄정호, 김용기, 김지희, 장재우, "개인 정보 보호를 위한 기존 cloaking 기법 연구의 분석," 데이터베이스연구회지 24권 1호, 2008, pp.17-32.

[12] B. Gedik and L. Liu, "Location Privacy in Mobile Systems: A Personalized Anonymization Model," In Proc. of IEEE International Conference on Distributed Computing Systems, 2005, pp. 620 - 629.



김 지 희
 2007년 전북대학교 컴퓨터공학과(공학사)
 2007년~현재 전북대학교 컴퓨터공학과 석사과정
 관심분야는 공간네트워크 데이터베이스, 질의처리 알고리즘, 정보보호, LBS



이 아 름
 2008년 전북대학교 컴퓨터공학과(공학사)
 2008년~현재 전북대학교 컴퓨터공학과 석사과정
 관심분야는 공간 데이터베이스, 질의처리, 위치 보안을 위한 클로킹



김 용 기
 2002년 전북대학교 컴퓨터공학과(공학사)
 2005년 전북대학교 대학원 컴퓨터공학과 (공학석사)
 2006년~현재 전북대학교대학원 컴퓨터 공학과 박사과정
 관심분야는 공간 데이터베이스, 질의 처리 알고리즘, 공간 색인 구조, 센서 네트워크



엄 정 호
 2004년 전북대학교 컴퓨터공학과(공학사)
 2004년~2006년 전북대학교 컴퓨터공학과 (공학석사)
 2006년~현재 전북대학교 컴퓨터 공학과 박사과정
 관심분야는 공간 데이터베이스, 공간 색인 구조, GIS



장 재 우
 1984년 서울대학교 전자계산기공학과 (공학사)
 1986년 한국과학기술원 전산학과 (공학석사)
 1991년 한국과학기술원 전산학과 (공학박사)
 1996년~1997년 Univ. of Minnesota, Visiting Scholar
 2003년~2004년 Penn State Univ., Visiting Scholar.
 1991년~현재 전북대학교 컴퓨터공학과 교수
 관심분야는 공간 네트워크 데이터베이스, 하부저장구조, 센서 네트워크