P2P 환경의 위치 기반 서비스에서 연속적인 질의 처리를 지원하는 그리드 기반 Cloaking 영역 설정 기법

Grid-based Cloaking Area Creation Scheme supporting Continuous Query Processing for Location-based Services of Peer-to-Peer Environment

김 형 일* 이 아 름** 장 재 우***
Hyeong Il Kim Ah Reum Lee Jae Woo Chang

요 약 최근 PDA, 휴대폰과 같은 모바일 기기 및 GPS와 같은 무선 통신 기술의 발달로 인하여 위치 기반 서비스의 이용이 확산되었다. 하지만 이러한 서비스는 사용자의 정확한 위치정보를 가지고 LBS 서버에 연속적으로 질의를 요청하기 때문에, 심각한 개인 정보 누출의 위협이 될 수 있다. 따라서 모바일 사용자의 안전하고 편리한 위치기반 서비스 사용을 위한 개인 정보 보호 방법이 필요하다. 이를 위해 본 논문에서는 Peer-to-Peer 환경의 위치기반 서비스에서 연속적인 질의 처리를 지원하는 그리드 기반 클로킹(Cloaking) 영역 설정 기법을 제안한다. 제안하는 기법은 Peer-to-Peer 환경에서 연속적인 위치기반 서비스를 지원하기 위하여, Chord 프로토콜을 사용하여 클로킹 영역을 설정한다. 마지막으로 성능평가를 통해서 제안하는 기법이 서비스 시간면에서 기존 연구보다 우수함을 보인다.

키워드 : 개인 정보 보호, 연속 질의 처리, 위치기반 서비스, 클로킹 기법, Peer-to-peer 환경

Abstract Recent development in wireless communication technology, like GPS, and mobile equipments, like PDA and cellular phone, make location-based services (LBSs) popular. However, because, users continuously request a query to a server in the LBSs by using their exact locations, privacy information could be in danger. Therefore, a mechanism for users' privacy protection is required for the safe and comfortable use of LBSs by mobile users. For this, we, in this paper, propose a grid-based cloaking area creation scheme supporting continuous LBSs in peer-to-peer environment. The proposed scheme creates a cloaking area by using Chord protocol, so as to support the continuous LBSs in peer-to-peer environment. Finally, we show from a performance analysis that our cloaking scheme outperforms the existing cloaking schemes, in terms of service time.

Keywords : Privacy protection, Continuous Query Processing, Location-Based Services, Cloaking scheme, Peer-to-peer Environment

1. 서 론

최근 PDA, 휴대폰과 같은 모바일 기기 및 GPS와 같은 무선 통신 기술의 발달로 인하여 위치 기반 서비스(Location-Based Service : LBS)의 이용이 확산되었다. LBS란 유무선 통신망을 통해 얻은 위치정보를 유용한 정보와 결합하여 사용자가 필요로 하는 부가적인 응용 서비스를 제공하는 것이다[1, 2]. 위치 기반 서비스에서 모바일 사용자는 자신의 위치정보를 LBS 서버에 보내어 교통 정보, 친구 찾기, 인접한 POI(Point Of Interest) 찾기 등 다양한 종류의 서비스를 이용할 수 있다. 그러나 이와 같이 사용자의 정확한 위치정보를 가지고 LBS 서버에 서비스를 요청하는 것은 심각한 개인 정보 누출의 위협이 될 수 있다. 이는 LBS 서버에 보내진 사용자의 위치정보가 유/무선 통신상에서 유출되거나,

† 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구이며(No. 2010-0000202) 아울러 교육과학기술부와 한국산업기술진흥원의 지역혁신인력양성사업으로 수행된 연구결과임.

* 전북대학교 전기전자컴퓨터공학부 컴퓨터공학 석사과정 hikim@dblab.chonbuk.ac.kr
** 전북대학교 전기전자컴퓨터공학부 컴퓨터공학 석사 arlee@dblab.chonbuk.ac.kr
*** 전북대학교 전기전자컴퓨터공학부 컴퓨터공학 교수 jwchang@dblab.chonbuk.ac.kr(교신저자)

LBS 서버의 관리자에 의해 악용될 수 있기 때문이다. 실제로 위치 기반 서비스를 이용한 스토킹이나 개인정보 유출 사례가 빈번히 발생하고 있다[3, 4]. 따라서 사용자의 안전하고 편리한 위치기반 서비스 사용을 위한 개인 정보 보호 방법이 요구된다.

위치기반 서비스에서 사용자의 위치정보 보호를 위한 연구로 K-Anonymity를 만족하는 클로킹 영역을 설정하는 연구가 존재한다. 이 기법은 LBS 서버에 질의(서비스) 전송 시 질의를 요청한 사용자의 위치정보와 k-1명의 인접한 사용자의 위치정보를 포함하는 클로킹 영역을 전송함으로써, 사용자의 위치 노출 확률을 1/k로 감소시킨다. 그러나 이와 같은 클로킹 영역 설정 기법은 사용자가 이동하면서 연속적으로 서비스를 요청하는 경우, 설정되는 모든 클로킹 영역에 질의를 요청한 사용자는 존재하지만, 이전 시간 클로킹 영역에 포함되었던 k-1명의 사용자는 영역 내에 존재하지 않을 수 있다. 이러한 경우, 각 서비스 시간별 클로킹 영역 내 사용자 비교를 통하여 질의 요청자의 신원을 유추할 수 있다는 문제점이 존재한다. 이러한 문제점을 해결하기 위해 T. Xu et al.의 연구[5]에서는 Advanced K-Anonymity Area(이하 Advanced KAA)를 제안하였다. Advanced KAA는 엔트로피(Entropy)를 사용하여 설정되는 클로킹 영역의 Anonymity Degree (이하 anonymity 정도)를 계산함으로써 질의 요청자의 노출을 방지하고, 아울러 연속적인 위치기반 질의를 처리할 수 있는 기법이다. 하지만, Advanced KAA는 anonymizer를 통해 클로킹 영역을 설정하는 중앙 집중 방식이기 때문에, anonymizer에서의 병목(bottle-neck) 현상으로 인한 성능 저하 문제 및 anonymizer 자체의 보안 위협 문제가 존재한다.

따라서 본 논문에서는 중앙 집중 방식에서의 문제점을 해결하기 위해, P2P 환경의 위치기반 서비스에서 연속적인 질의 처리를 지원하는 그리드 기반 클로킹 영역 설정 기법을 제안한다. 이를 위해, 첫째, 클로킹 영역 설정 시 필요한 정보를 분산 저장하여, 연속적인 질의를 효율적으로 지원한다. 둘째, 클로킹 영역 설정을 위한 연산을 분산적으로 수행하여, 사용자에게 빠른 응답시간을 보장한다. 마지막으로, 정보 병합 시, 각 클러스터 내 병합 노드를 통해 통신을 계층적으로 수행함으로써, 질의 요청자에게 집중되는 통신을 분산시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 K-anonymity를 고려한 기존의 클로킹 기법들을 소개한다. 3장에서는 Peer-to-Peer 환경의 위치기반 서비스에서 연속적인 질의 처리를 지원하는 그리드 기반 클로킹 영역 설정 기법을 제안하고, 4장에서는 제안하는 기법과 기존 기법과의 성능 비교를 수행한다. 마지막으로 5장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

위치기반 서비스를 이용하는 사용자의 위치정보 보호를 위한 연구 가운데, 중앙 집중 방식으로 K-anonymity를 만족하는 클로킹 영역을 생성하는 대표적인 연구로는 [6, 7, 8, 9]가 존재한다. 그러나, 중앙 집중 방식에서의 클로킹 기법들은 anonymizer에서의 병목 현상으로 인한 성능 저하 문제 및 사용자의 위치정보를 관리하고 있는 anonymizer 자체의 보안 위협 문제를 지닌다. 이와 같은 중앙 집중 방식에서의 문제점을 해결하기 위해, 클로킹 영역 설정 시 중앙의 anonymizer를 사용하지 않고, 질의 요청자가 다른 사용자들과의 통신을 통하여 직접 클로킹 영역을 설정하는 연구가 존재한다. 먼저, C. Chow et al.의 연구[10]는 질의를 요청한 모바일 사용자가 자신의 통신 범위 내에 있는 k-1명의 다른 사용자를 찾아 클로킹 영역을 설정하는 기법을 제안하였다. 또한, G. Ghinita et al.의 연구[11]는 힐버트 커브를 이용하여 사용자의 위치를 1차원으로 암호화하고 분산 해쉬 테이블 구조인 Chord를 구성하여, 이를 기반으로 클로킹 영역을 설정하는 연구를 수행하였다.

그러나 이러한 클로킹 영역 설정 기법들은 특정 시점(snapshot)에서 사용자의 위치정보를 보호하기 위한 것으로, 사용자가 연속적으로 질의를 요청할 경우 사용자의 위치가 노출될 가능성이 존재한다. 이는 매 서비스 시간 설정되는 클로킹 영역이 K-anonymity를 만족하지만, 이동 경로에 따라 처음 서비스 시간에 포함된 k-1명의 사용자가 포함되지 않을 가능성이 존재하여, 각 서비스 시간별 클로킹 영역 안의 사용자 비교를 통해 질의 요청자를 파악할 수 있기 때문이다. 예를 들어, 질의를 요청한 사용자(N)가 k=5를 요청한다고 가정할 때, 그림 1은 이러한 문제점을 보여준다. 매 서비스 시간에서의 클로킹 영역은 사용자가 요구하는 k=5를 만족한다.

하지만, 초기 클로킹 영역(T=0) 안에 포함된 사용자들 중, T=2초일 때의 클로킹 영역에 남아있는 사용자는 N 뿐이다. 이러한 경우, 상대방은 N이 질의를 요청한 사용자라는 것을 파악 할 수 있다.

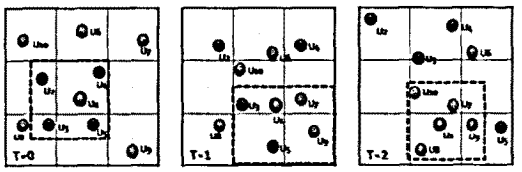


그림 1. 연속적으로 질의 요청 시 발생하는 문제

이러한 문제를 해결하기 위해, Toby Xu와 Ying Cai의 연구[5]인 Advanced KAA에서는 엔트로피(entropy)를 기반으로 클로킹 영역의 anonymity 정도를 측정하여, 이를 통해 사용자가 요구한 K-anonymity를 만족하는 최소의 원을 클로킹 영역으로 설정한다. Advanced KAA의 클로킹 영역 설정 기법은 다음과 같다. 첫째, 서비스 요청(T_0) 시, anonymizer는 질의 요청자를 기준으로 k-1명의 인접한 사용자를 찾고, 이를 모두 포함하는 원을 생성한 후, 해당 원의 2배가 되는 원을 임시 클로킹 영역으로 설정한다. 이 후, 임시 클로킹 영역 내에서 질의 요청자가 요구하는 K-anonymity를 만족하는 가장 작은 원을 찾아, 이를 클로킹 영역으로 설정한다. 둘째, 서비스 시간이 T_{i+1}일 경우, T_i 시간의 클로킹 영역에 속한 사용자(이하 참여자)의 갱신된 위치정보를 획득한 후, 이를 통해 생성할 수 있는 모든 클로킹 영역을 구한다. 이 후, 각 영역에 포함된 사용자의 확률 값을 기반으로 엔트로피를 계산하고, 아울러 영역의 anonymity 정도를 구한다. 마지막으로, 이 중 질의 요청자가 요구하는 K-anonymity를 만족하면서 최소의 크기를 갖는 영역을 최종 클로킹 영역으로 설정한다. 한편, 질의를 요청한 사용자 N을 위한 클로킹 영역을 A, 클로킹 영역 A에 포함된 사용자들의 집합을 $S(A) = \{N_1, N_2, \cdots, N_m\}$라 할 때, A의 엔트로피는 식 (1)과 같이 계산한다.

$$H(A) = -\sum_{i=1}^{m} p_i \log p_i \quad (1)$$

여기서 H(A)는 클로킹 영역 A의 엔트로피를 나타내고, $p_i(1 \le i \le m)$는 이전 시간의 사용자가 현재 시간의 각 사용자의 위치로 움직였을 확률을 의미한다. 또한, 식 (1)에 의해 구해진 엔트로피를 식 (2)에 대입하여 클로킹 영역 A의 Anonymity 정도(D(A))를 계산한다.

$$D(A) = 2^{H(A)} \quad (2)$$

그러나, Advanced KAA는 anonymizer를 통해 클로킹 영역을 설정하는 중앙 집중 방식이기 때문에, anonymizer에서의 병목 현상으로 인한 성능 저하 문제 및 사용자의 위치정보를 관리하고 있는 anonymizer 자체의 보안 위협 문제를 지닌다.

3. P2P 환경에서 연속적인 질의 처리를 지원하는 그리드 기반 클로킹 영역 설정 기법

3.1 연구 동기

연속적인 위치기반 서비스를 지원하는 기존의 연구들은 anonymizer를 이용하여 클로킹 영역을 생성하는 중앙 집중 방식의 시스템 구조를 사용한다. 이러한 중앙 집중 방식은 다음과 같은 두 가지 문제점을 지닌다. 첫째, anonymizer에서의 병목 현상으로 인한 성능 저하 문제가 존재한다. 이는 클로킹 영역을 설정하거나, LBS 서버로부터 전송받은 후보 집합 중 질의 요청자의 실제 위치를 고려한 최종 결과를 얻는 모든 과정을 anonymizer가 수행하기 때문이다. 둘째, 해커(hacker)나 악의적인 관리자에 의해 anonymizer가 저장하고 있는 사용자의 위치정보가 악용될 수 있다. 예를 들어, 이러한 위치정보를 통해 사용자가 어느 장소를 자주 방문하는지, 또한 이러한 방문이 주로 어느 시간대에 이뤄지는지를 파악할 수 있기 때문에, 이를 통해 사용자의 생활 스타일 및 질병 정보 등을 유추할 수 있다는 문제점이 존재한다. 따라서 본 논문에서는 질의 요청자 스스로 클로킹 영역을 설정하는 P2P 방식을 사용하며, 위치기반 서비스에서 연속적인 질의 처리를 지원할 수 있는 클로킹 영역 설정 기법을 제안한다. P2P 환경에서 연속적인 질의 처리를 지원하기 위해, 다음의 세 가지 사항을 고려하여 클로킹 영역 설정 기법을 제안한다. 첫째, 클로킹 영역 설정에 필요한 정보들을 분산 저장하여, 자원이 한정적인 모바일에서 정보를 효율적으로 유지할 수 있도록 한다. 둘째, 클로킹 영역 설정에 참여하는 사용자들이 이동확률 매트릭스 생성 및 사용자 확률 계산 등의 연산을 분산 수행하여, 제한적인 처리 능력을

지니는 모바일에서 빠른 응답시간을 보장하도록 한다. 마지막으로, 클로킹 영역 설정에 참여하는 사용자들의 클러스터 내 병합 노드를 선정하여, 동일한 클러스터 내 사용자들이 병합 노드를 통해 메시지를 전송하도록 한다. 이를 통해 질의 요청자에게 집중되는 통신을 분산시켜 병목현상을 방지한다.

3.2 시스템 구조

본 논문에서 사용하는 시스템은 그림 2와 같다. 시스템은 크게 인증 서버(Certification Server), 익명화(Pseudonym) 서비스, LBS 서버, Chord 프로토콜로 구성된다. 인증 서버에는 위치기반 서비스 사용자들이 등록되어 있으며, 인증 서버에 존재하는 사용자들은 주기적으로 인증을 요청한다. 익명화 서비스는 클로킹 영역을 LBS 서버에게 보내기 전에 사용자의 IP 및 식별 번호를 은닉하는 역할[12]을 하며, LBS 서버는 전송된 클로킹 영역에 대하여 질의 처리를 하고, 그에 대한 질의 결과 후보 집합을 질의 요청자에게 전송한다. Chord 프로토콜[13]은 모바일 사용자들이 Chord 라는 분산 해시 테이블(Distributed Hash Table) 구조에 기반하여 구성된 링 형태의 P2P 네트워크이며, 사용자들은 클러스터 단위로 구성된다. 본 논문에서는 사용자 간 통신 시 상대방이 감청하는 것에 대비하여, 사용자의 위치를 힐버트 커브를 이용하여 힐버트 값으로 변환하고, 이를 인덱싱 키로 고려하여 가상 링 구조를 형성하는 클러스터 구성방법[11]을 사용한다.

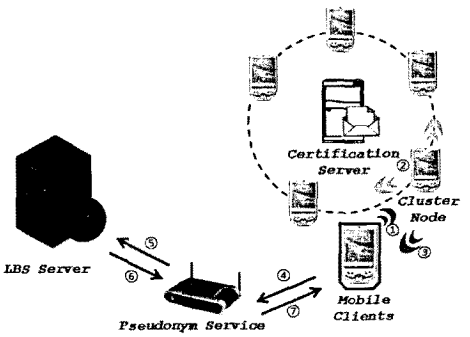


그림 2. 사용하는 분산 방식의 시스템 구조

위의 시스템 구조를 고려하여 사용자의 질의를 수행하는 과정은 다음과 같다. 인증 서버에 존재하는 질의 요청자는 자신을 포함한 클러스터의 대표 노드에 접근하여 이웃한 사용자 정보를 요청한다(①). 대표노드는 이웃한 사용자를 검색하여(②), 검색된 사용자 정보를 질의 요청자에게 전송한다(③). 위 과정은 질의 요청자가 요구한 K-anonymity를 만족하기 위해, k-1명의 사용자를 검색할 때까지 반복 수행되며, 검색된 사용자를 기반으로 클로킹 영역을 설정한 후, 이를 통해 질의를 요청한다(④). 이 때, 질의 요청자는 익명화 서비스를 통해 사용자의 IP 및 식별 번호를 은닉한 후, LBS 서버에 질의를 요청한다(⑤). LBS 서버는 전송받은 클로킹 영역을 기반으로 질의를 수행하고, 후보 결과 집합을 익명화 서비스로 전송한다. (⑥). 익명화 서비스는 질의 요청 시에 은닉했던 IP 주소를 탐색하고, 후보 결과 집합을 질의 요청자에게 전송한다(⑦). 클로킹 영역을 기반으로 얻어진 후보 결과 집합은 실제 질의 지점(exact query point)을 기반으로 한 질의 결과를 모두 포함한다. 따라서 질의 요청자는 자신의 실제 위치를 고려하여 후보 결과 집합에서 정확한 결과를 얻는다.

3.3 P2P 그리드 기반 클로킹 영역 설정 기법

본 절에서는 P2P 그리드 기반 환경에서 연속적인 위치기반 서비스를 지원하기 위해 KAA를 확장한 방법을 제안한다. 아울러 힐버트 커브로 암호화된 셀을 효율적으로 검색하기 위해, A. Lee et al.이 제안한 DAHC(Direct Accessing of adjacent cells using Hilbert Curves)[14]를 이용한다. DAHC 기법을 사용하여 힐버트 값으로 암호화된 셀 정보를 계산하고, 이를 통해 인접한 셀에 존재하는 사용자를 검색한다.

수행단계 1. 서비스 요청 시 클로킹 영역 설정

서비스 요청 시(T_0), 질의 요청자는 자신을 기준으로 k-1명의 인접한 사용자를 찾고, 이를 모두 포함하는 원을 생성한 후, 생성된 원의 2배가 되는 원을 임시 클로킹 영역으로 설정한다. 이 후, 임시 클로킹 영역 내에서 질의 요청자가 요구하는 K-anonymity를 만족하는 가장 작은 원을 찾아, 이를 클로킹 영역으로 설정한다. 서비스 수행 시간동안(T_i)에는 수행단계 2~5를 통해 사용자의 프라이버시를 보장하면서 P2P 환경에서 분산적으로 클로킹 영역을 설정한다.

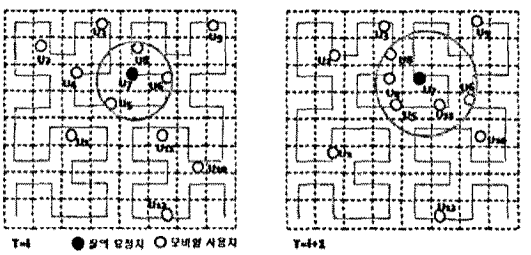
수행단계 2. 임시 클로킹 영역 설정

Ti 시간에 클로킹 영역 안에 존재했던 참여자는

각자의 이동 경로를 따라 움직인 후, T_{i+1} 시간의 위치정보를 질의 요청자에게 전송한다. 질의 요청자는 전송받은 위치정보를 바탕으로 T_i 시간에 클로킹 영역에 포함되었던 참여자들을 모두 포함하는 최소의 원을 임시 클로킹 영역으로 설정한다. 질의 요청자는 T_{i+1} 시간의 임시 클로킹 영역에 포함된 사용자를 모두 찾기 위하여, 임시 클로킹 영역에 속한 모든 셀을 DAHC 기법을 이용하여 계산한다. 다음으로, 계산된 셀 정보를 클러스터의 대표 노드에 전송하여 해당 셀에 사용자가 존재하는 지를 검색한 후, 이를 기반으로 임시 클로킹 영역에 속한 사용자의 정보를 저장한다. 그림 3은 임시 클로킹 영역 설정에 대한 예제를 보인다. 그림 3(a)는 T=i일 때, 질의 요청자(U_7)가 k=4를 요청한 경우 설정된 클로킹 영역을 나타낸다. T=i일 때 클로킹 영역에 포함되었던 참여자는 T=i+1 시간에 갱신된 자신의 위치정보를 질의 요청자에게 전송한다. 질의 요청자는 이를 기반으로 T=i 시간에 클로킹 영역에 포함되었던 참여자들을 모두 포함하는 임시 클로킹 영역을 그림 3(b)와 같이 설정한다. 이 후, 임시 클로킹 영역에 포함된 셀을 계산하고 대표노드에 셀 정보를 전송하여, 클로킹 영역에 포함된 사용자 U_4, U_5, U_6, U_7, U_8, U_{11}에 대한 정보를 얻어온다.

수행단계 3. 각 클러스터 내 병합노드의 선정

사용자들의 확률 값을 분산적으로 계산하기 위하여 이전 시간에 클로킹 영역에 속한 참여자들에게 표 1과 같은 임시 사용자 정보 테이블과 이동확률 매트릭스를 구성할 샘플의 수 α값을 생성하여 전송한다. 여기서 임시 사용자 정보 테이블은 임시 클로킹 영역에 포함된 사용자들의 정보와 병합노드, 병합할 사용자 수에 대한 정보를 저장하며, 병합노드는 해당 클러스터에 속한 참여자의 부분 확률 값을 합산하는 노드를 의미한다. 병합노드는 질의 요청자가 참여자들에게 정보 전송 시 선정되며, 병합 방식은 다음과 같다. 먼저, 참여자들을 각 클러스터에 따라 분류하고 임의로(random) 병합노드를 선정한다. 질의 요청자는 병합노드에게는 병합노드임을 알리고, 병합할 사용자의 수를 함께 전송한다. 나머지 참여자들에게는 자신을 담당하는 병합노드를 알려준다. 이러한 과정을 통해 분류된 참여자들은 선정된 병합노드에게 자신의 정보를 전송하고, 병합노드들은 병합할 참여자 수만큼의 메시지를 전송 받을 때까지 대기한다. 모든 참여자로부터 정보를 전송받으면, 정보를 합산하여 질의 요청자에게 전송한다.



(a) 사용자 분포(T=i) (b) 사용자 분포(T=i+1)

그림 3. 임시 클로킹 영역 설정 예제

예를 들어, 그림 3(b)와 같은 분포를 반영한 Chord 구조가 그림 4와 같을 때, 각 참여자들은 자신의 이전 확률 값을 유지하고 있다. 아울러, 질의 요청자인 U_7은 참여자들에게 표 1과 같은 임시 사용자 정보 테이블을 전송한다. 임시 사용자 정보 테이블을 통해 U_8은 병합노드이며, 1명의 참여자로부터 정보를 전송받아야 함을 알 수 있다. 아울러, U_8에게 정보를 전송해야하는 참여자는 U_5임을 알 수 있다.

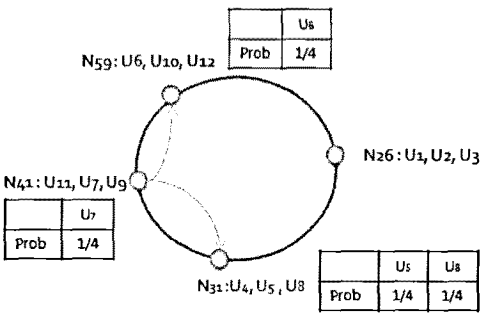


그림 4. T = i + 1일 때의 Chord 구조

표 1. 임시 사용자 정보 테이블

	U_5	U_6	U_7	U_8
병합 노드	U_8	U_6	U_7	U_8
병합할 사용자 수	0	0	0	1

수행단계 4. 임시 클로킹 영역 내 사용자의 확률 값 계산

설정되는 클로킹 영역의 엔트로피를 계산하기 위해, T_i 시간에 설정된 클로킹 영역에 포함된 참여자들이 T_{i+1} 시간에 설정되는 클로킹 영역 안에서 어떤 사용자가 될 확률을 계산한다. 이를 위해, 각 행

이 α개의 샘플로 이루어진 이동확률 매트릭스를 생성한다. 여기에서 각 행은 한 참여자가 임시 클로킹 영역에 속한 어떤 사용자가 될 확률을 의미한다. 사용자의 확률 값 계산을 참여자들과 나누어 계산하기 위해 이동확률 매트릭스의 각 행을 분할한 후, 부분적으로 확률 값을 계산한다. 즉, 각 참여자는 자신이 T_{i+1} 시간에 설정되는 클로킹 영역에서 어떤 사용자가 될 확률을 계산하기 위하여, 질의 요청자로부터 전송받은 정보와 α, 임시 사용자 정보를 기반으로 부분 이동확률 매트릭스(이하 M')를 생성한다. 아울러, 자신이 유지하고 있는 확률 값을 식 (4)에 대입하여 부분 확률 값을 계산한다.

$$Prob_{i+1} = beforeU_{prob} * M'_{i+1} / \alpha \quad (4)$$

여기서 beforeUprob는 Ti 시간에 참여자의 확률 값을 의미한다. 참여자들은 계산된 부분 확률 값을 병합노드로 전송하고, 각 클러스터 내의 병합노드는 자신이 병합해야 할 부분 확률 값을 합산하여 질의 요청자에게 전송한다. 질의 요청자는 병합노드로부터 전송받은 부분 확률 값을 합산하여 임시 클로킹 영역에 포함된 사용자의 최종 확률 값을 구한다.

수행단계 5. Anonymity 정도를 고려한 클로킹 영역 설정

임시 클로킹 영역에 속한 사용자들의 확률 값이 계산되면, 이를 기반으로 엔트로피를 계산한다. 질의 요청자는 Anonymity 정도를 만족하는 최소 클로킹 영역 설정을 위해, 임시 클로킹 영역 안에 있는 사용자를 통해 생성 가능한 모든 원을 고려한다. 이 때, 생성된 각 원 안에 속한 사용자들의 확률 값을 식 1에 대입하여 엔트로피를 계산하고, 이를 식 2에 대입하여 *anonymity* 정도인 k'을 계산한다. 이 중, $k' \geq k$를 만족하면서 가장 작은 영역을 가지는 원이 최종 클로킹 영역으로 설정된다. 이 후, 최종 클로킹 영역에 포함된 사용자에게 계산된 확률 값을 전송하여, 참여자들이 유지하고 있는 정보를 갱신하게 한다. 이는 이후 시간에 클로킹 영역을 설정할 때 이용된다.

수행단계 1~5를 고려한 알고리즘은 P2P 환경에서 분산적으로 클로킹 영역을 설정하기 때문에 질의 요청자와 참여자 측면으로 구분하여 설계한다. 먼저, 질의 요청자 측면의 알고리즘은 그림 5와 같다. 알고리즘은 크게 서비스 요청시와 수행시간동안으로 나뉜다. 첫째, 서비스 요청시, 질의 요청자의 위치정보와 k 값을 고려하여 클로킹 영역을 설정하여 반환하고, 설정된 클로킹 영역 내의 사용자들에게 확률 값을 전송한다(line 2~5). 둘째, 서비스 시간이 증가하면, 이전 시간에 설정된 클로킹 영역에 속했던 참여자들로부터 현재 갱신된 위치정보를 전송받아 임시 클로킹 영역을 원으로 설정한다(line 7). 셋째, 설정된 임시 클로킹 영역 내의 각 셀 정보를 바탕으로 클러스터에 존재하는 사용자 정보를 검색한다. 만약 사용자가 존재한다면, 사용자 정보 테이블에 검색된 사용자 정보를 추가한다(line 8~11). 넷째, 임시 클로킹 영역 내에 존재하는 사용자의 수가 참여자의 수와 같은 경우, 임시 클로킹 영역을 현재 서비스 시간의 클로킹 영역으로 설정한다(line 12). 만약 참여자의 수보다 많다면, 임시 클로킹 영역에 속한 사용자들의 확률 값을 계산한다. 질의 요청자는 참여자들이 포함된 클러스터를 고려하여 병합노드를 선정하고, 참여자들에게 임시 클로킹 영역에 대한 정보를 전송한다. 이후 모든 병합노드로부터 메시지를 기다린 후, 분산적으로 계산된 사용자의 확률 값을 합산한다(line 14~17). 다섯째,

```
Distributed advanced K-anonymity Area Algorithm
Input : <qx, qy>   //질의 요청자의 위치정보
        k          //K-anonymity
Output : CA        //Cloaking Area
1. For(t < T)      // t : 현재 서비스 시간
2.   If(t==0)
3.   |  CA = setCloakingArea(qx, qy, k);
4.   |  preNum = num; //설정된 CA 내 사용자 수
5.   |  sendUserInfo(uid, prob);
6.   else{
7.   |  bound = setLimitCloakingArea(UserInfo);
8.   |  for all cells in LimitCloakingArea
9.   |  |  uInfo = searchChord(cell);
10.  |  |  if(uInfo!=0)
11.  |  |   uTable = uInfo;   num++;
12.  |  if(preNum == num) then CA = bound;
13.  |  else
14.  |  |  decide_mergeNode(uTable);
15.  |  |  send_Usermessage(uTable);
16.  |  |  while( mergeNum< uTable.userNum)
17.  |  |    userprob += aggregatePatialprob();
18.  |  |  for all circle can be generated
19.  |  |  |  temp_radius = getRadius();
20.  |  |  |  D = calculateDegree(usernum, userprob);
21.  |  |  |  if(D ≥ k && temp_radius < radius)
22.  |  |  |    CA = setCloakingArea_t(userInfo);
23.  |  |  sendParticipantInfo(uid, prob);
End Algorithm
```

그림 5. P2P 그리드 기반 클로킹 영역 설정 기법(질의 요청자 측면)

질의 요청자는 임시 클로킹 영역 안에 있는 사용자들 중에서 2명 혹은 3명을 통해 생성할 수 있는 모든 원을 생성한다. 이 때, 각 원 안에 포함된 사용자들의 확률 값을 기반으로 Anonymity 정도를 계산한다. 이 중, 계산된 값이 k 값보다 크면서, 가장 작은 영역을 갖는 원을 최종 클로킹 영역으로 설정한다(line 18~22). 이후 설정된 클로킹 영역에 포함된 사용자들에게 계산된 확률 값을 전송하고 알고리즘을 종료한다(line 23).

참여자 측면의 알고리즘은 그림 6과 같다. 첫째, 참여자들은 질의 요청자로부터 전송받은 정보를 기반으로 부분 이동확률 매트릭스를 생성하고, 부분 확률 값을 계산한다(line 1~2). 둘째, 병합노드는 자신이 담당하는 사용자들의 부분 확률 값을 합산하여 질의 요청자에게 전송한다(line 3~6). 만약 병합노드가 아니라면, 계산된 부분 확률 값을 병합노드로 전송한다(line 7).

```
Partially user probability calculation Algorithm
1.  pTmatrix = createPartialTmatrix(uInfo);
2.  patialprob = calculatePartialprob(pTmatrix);
3.  If(uid == agg_id)
4.  |  while( agg_num < uInfo.userNum)
5.  |       patialprob = aggregatePatialprob();
6.  |  sendtoQueryNode(patialprob);
7.  else then sendtoaggNode(patialprob);
End Algorithm
```

그림 6. P2P 그리드 기반 클로킹 영역 설정 기법(참여자 측면)

4. 성능평가

본 장에서는 P2P 환경의 위치기반 서비스에서 연속적인 질의 처리를 지원하는 그리드 기반 클로킹 영역 설정 기법(이하 DKAA(Distributed Advanced K-anonymity Area)의 우수성을 검증하기 위하여 성능 평가를 수행한다. 성능 평가는 제안하는 기법이 P2P 그리드 환경에서 연속적인 질의 처리를 지원하는 첫 번째 연구이기 때문에 중앙 집중 방식에서 연구된 기존의 KAA와 비교한다. 성능 평가 항목으로는 k-anonymity, 모바일 사용자의 수, session life time 변화에 따른 클로킹 영역 크기, 클로킹 영역 설정 시간 및 질의 처리 시간을 비교한다. 성능 평가의 실험 환경은 표 2와 같다.

표 2. 실험 환경

항목	성 능
CPU	Intel Core2 Quad CPU Q6600 2.40GHz
Memory	2GB
OS	Windows XP professional
Compiler	Microsoft Visual Studio 6.0

아울러, 이동객체 데이터는 Network-based Generator[15]를 사용하여 독일 올덴버그의(15×15㎢) 실제 도로 네트워크를 기반으로 생성하였으며, 전체 영역 대비 설정된 클로킹 영역의 비율을 쉽게 파악하기 위하여 클로킹 영역의 최대 크기는 1로 설정하였다. 표 3은 성능평가에 사용된 매개변수들이다.

표 3. 실험 환경 매개변수

parameter	range	default
K-anonymity	5 ~ 20	10
모바일 사용자 수	2k ~ 8k	5000
서비스 수행 시간	3 ~ 8	5
셀 크기	-	512×512

4.1 K-anonymity(이하 k) 변화에 따른 성능평가

그림 7은 k 변화에 따른 설정되는 클로킹 영역의 크기를 비교한 것이다. 두 기법 모두 k 값이 증가함에 따라 클로킹 영역을 크게 설정한다. k가 10인 경우 KAA는 0.00045, DKAA는 0.00062의 클로킹 영역을 설정한다. DKAA가 KAA에 비해 넓은 영역을 설정하는 이유는, 사용자간의 암호화된 메시지를 주고받기 위해 좌표 정보를 헐버트 값으로 변환하면서, 사용자의 위치정보가 한 점으로 사상(mapping)되는 현상이 발생하기 때문이다. 이로 인해, 원을 생성할 수 있는 경우가 줄어들게 되고, 결과적으로 가장 먼 2명의 사용자를 지름으로 하는 원이 클로킹 영역으로 반환되는 경우가 발생하기 때문이다.

그림 8은 k 변화에 따른 총 서비스 시간을 비교한 것이다. 총 서비스 시간이란 클로킹 영역 설정 시간과 설정된 영역을 통한 질의 처리 시간을 합한 것이다. 두 기법 모두 k 값이 증가함에 따라 총 서비스 시간이 증가하였다. k가 10인 경우, KAA는 0.27, DKAA는 0.04가 걸린다. DKAA가 KAA보다

좋은 성능을 보이는 이유는 클로킹 영역 설정에 참여하는 사용자와 일을 분산적으로 수행하기 때문이다. 또한, 힐버트 커브를 사용한 DKAA는 사용자의 위치정보가 한 점으로 사상되는 현상으로 인해 KAA에 비해 생성 가능한 원의 개수가 줄어들어, 보다 적은 클로킹 영역 설정 시간이 소요되기 때문이다. 한편, 질의 처리 시간은 클로킹 영역 설정 시간에 비해 상대적으로 적은 비중을 차지하기 때문에, 위치 정보 암호화를 위해 DKAA가 KAA에 비해 조금 넓은 클로킹 영역을 설정함에도, 실제적인 서비스 처리 시간에 있어서 DKAA가 보다 우수함을 알 수 있다. 이를 통해, 힐버트 커브를 통한 위치 정보의 암호화를 수행한 DKAA가 KAA에 비해 정보 보호 측면에서 안전하며, 서비스 제공 측면에서도 효율적임을 알 수 있다.

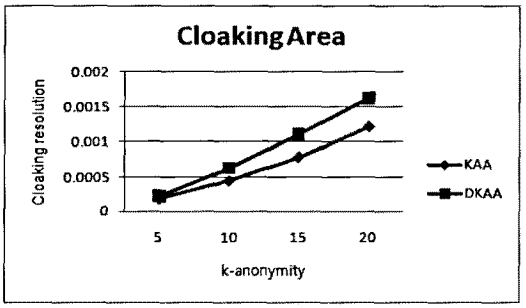


그림 7. k 변화에 따른 클로킹 영역 크기 비교

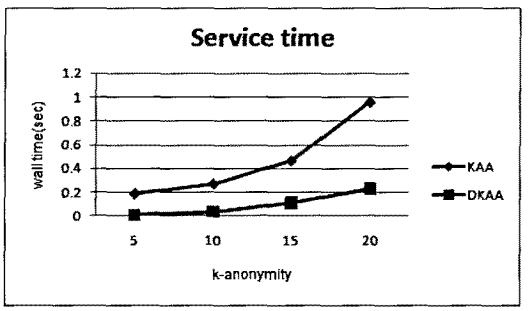


그림 8. k 변화에 따른 클로킹 영역설정 시간 비교

4.2 모바일 사용자 수 변화에 따른 성능평가

그림 9는 모바일 사용자 수 변화에 따른 클로킹 영역의 크기를 비교한 것이다. 사용자의 수가 2000인 경우, KAA는 0.00133, DKAA는 0.00214, 사용자의 수가 8000인 경우, KAA는 0.00032, DKAA는 0.00034의 Cloakng 영역을 설정하였다. 두 기법 모두 모바일 사용자 수가 증가할수록, 클로킹 영역이 감소하였다. 이는 모바일 사용자 수가 증가할수록 단위 면적 당 밀집도가 증가하여, 보다 좁은 면적에서 K-anonymity를 만족할 수 있는 충분한 모바일 사용자를 찾을 수 있기 때문이다. 특히, 모바일 사용자의 수가 5000 이상일 때는 DKAA와 KAA의 영역이 거의 유사해짐을 알 수 있다. 아울러, 성능 평가에 사용된 데이터가 실제 도시를 기반으로 한 데이터임을 고려하면, 모바일 사용자 수가 많은 경우가 실세계에 더 유사하다고 볼 수 있다. 따라서 사용자의 수가 많은 경우 보다 좋은 성능을 보이는 DKAA가 실용적임을 알 수 있다.

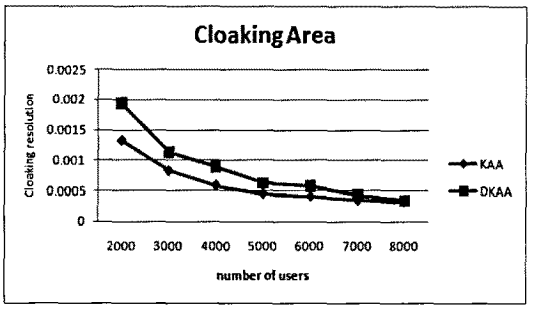


그림 9. 모바일 사용자 수 변화에 따른 클로킹 영역 크기 비교

그림 10은 모바일 사용자 수 변화에 따른 총 서비스 시간을 비교한 것이다. 사용자의 수가 2000인 경우, KAA는 0.28, DKAA는 0.05, 사용자의 수가 8000인 경우, KAA는 0.38, DKAA는 0.03의 시간이 소요되었다. 그 이유는, KAA는 모바일 사용자의 수가 증가할수록 클로킹 영역 설정 시간이 더 소요된다. 이는 KAA의 경우, 사용자의 수가 증가할수록 단위 면적 당 사용자의 밀집도가 증가하여, 모바일 사용자들로 생성 가능한 원의 개수가 증가하기 때문이다. 반면, DKAA는 모바일 사용자의 수가 증가할수록 클로킹 영역 설정 시간이 짧아진다. 이는, 사용자의 수가 늘어나면 보다 좁은 면적에서 클로킹 영역을 생성하게 되는데, DKAA의 경우는 사용자의 위치정보가 한 점으로 사상되는 현상으로 인해, 생성 가능한 원의 개수가 현저하게 줄어들기 때문이다. 한편, 질의처리 시간은 설정된 클로킹 영역의 크기에 영향을 받으며, 클로킹 영역 설정 시간에

비해 적은 비중을 차지한다. 또한 두 기법의 설정된 클로킹 영역의 크기가 큰 차이를 보이지 않기 때문에, 서비스 시간에 큰 영향을 미치지 않는다.

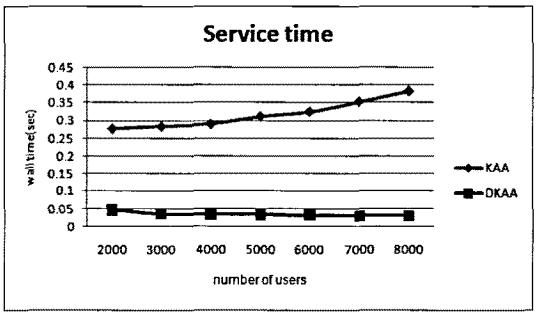


그림 10. 모바일 사용자 수 변화에 따른 클로킹 영역 설정 시간 비교

4.3 Session life time 수 변화에 따른 성능평가

그림 11은 session life time 수에 따른 클로킹 영역의 크기를 비교한 것이다. 두 기법 모두 움직이는 모바일 사용자를 대상으로 하기 때문에, 서비스 횟수가 증가할수록 클로킹 영역이 커지는 양상을 보였다. DKAA가 KAA에 비해 조금 큰 클로킹 영역을 설정하는 이유는, 힐버트 값 사용으로 인해 사용자의 위치정보가 한 점으로 사상(mapping)되는 현상이 발생하기 때문이다.

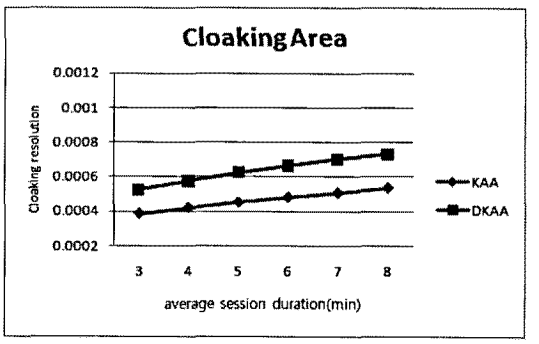


그림 11. session life time 변화에 따른 클로킹 영역 크기 비교

그림 12는 session life time 수에 따른 총 서비스 시간을 보여준다. 두 기법 모두 서비스 횟수가 증가할수록 클로킹 영역이 넓어지고, 영역에 포함되는 모바일 사용자의 수가 증가하기 때문에, 클로킹 영역 설정 시간이 증가한다. 따라서 총 서비스 시간도 증가함을 알 수 있다. 하지만, DKAA가 보다 완만한 상승 곡선을 보여 서비스 시간 측면에서 KAA에 비해 효율적임을 알 수 있다.

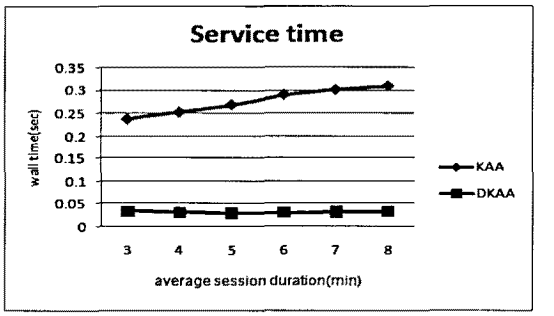


그림 12. session life time 변화에 따른 클로킹 영역 설정 시간 비교

5. 결론 및 향후연구

본 논문에서는 P2P 환경의 위치기반 서비스에서 연속적인 질의 처리를 지원하는 그리드 기반 클로킹 영역 설정 기법을 제안하였다. 제안하는 기법은 클로킹 영역 설정 시 필요한 정보들을 분산 유지하고, anonymity degree를 계산하기 위한 연산을 클로킹 영역에 포함된 사용자가 분산적으로 처리하였다. 아울러 질의 요청자에게 집중되는 통신을 오버헤드를 위해 병합노드를 사용하여 계층적으로 통신을 수행함으로써, P2P 환경에서 발생할 수 있는 문제점들을 해결하였다. 또한, 기존 연구인 Advanced KAA와의 성능 비교를 통해 제안하는 기법이 클로킹 영역 면에서 기존 기법과 유사한 성능을 보이고, 클로킹 영역 설정 시간 면에서 우수함을 검증하였다.

향후 연구는 분산 환경에서 사용자의 위치를 왜곡하지 않고 클로킹 영역을 설정하는 연구와, 본 연구를 확장하여 도로 네크워크를 고려한 연속적인 질의 처리를 지원하는 기법에 대해 연구하는 것이다.

참 고 문 헌

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

[2] 이낙훈, 박주훈, 안병익, 2004, "위치기반 응용 서비스(항법, 디렉토리, 위치추적)를 지원하는 LBS 표준 참조 시스템," 한국공간정보시스템학회 학술대회 논문집, pp.33-38.

[3] J. Voelcker, 2006, "Stalked by Satellite: An Alarming Rise in GPS-enabled Harassment," IEEE Spectrum, Vol.47 NO.7, pp.15-16.

[4] J. Warrior, E. McHenry, and K. McGee, 2003, "They Know Where You Are", IEEE Spectrum, Vol.40 No.7, pp.20-25.

[5] T. Xu and Y. Cai, 2007, "Location Anonymity in Continuous Location-based Services", In Proc. of the 15th annual ACMGIS, pp.221-238.

[6] M. Gruteser and D. Grunwald, 2003, "Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking," In Proc. of the International Conference on Mobile Systems, Applications and Services, pp.31-42.

[7] B. Gedik, L. Liu, 2005, "Location Privacy in Mobile Systems: A Personalized Anonymization Model", In Proc. of the 25th IEEE ICDCS, pp.620-629.

[8] M. Mokbel, C. Chow, W. Aref, 2006, "The New Casper: Query Processing for Location Services without Compromising Privacy," In Proc. of 32nd international conference on VLDB, pp.763-774.

[9] 엄정호, 김지희, 장재우, 2009, "위치기반 서비스에서 개인 정보 보호를 위한 그리드를 이용한 Cloaking 영역 생성 알고리즘", 한국공간정보시스템학회, 제11권, 2호, pp.151-161.

[10] C. Chow, M. Mokbel, and X. Liu. 2006, "A, Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services," In Proc. of the 14th annual ACMGIS, pp. 171 - 178

[11] G. Ghinita, P. Kalnis and S. Skiadopoulos, 2007, "MobiHide: A Mobilea Peer-to-Peer System for Anonymous Location-Based Queries," In Proc. of 10th international conference on SSTD, pp.221-238.

[12] Tor: Anonymity Online, http://tor.eff.org/

[13] I. Stoica, R. Morris, D. Karger, M. Kaashoek and H. Balakrishnan, 2003, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Application," In Proc. of IEEE/ACM TON, Vol.11 No.1, pp.17-32.

[14] 이아름, 엄정호, 장재우, 2009, "분산 그리드 환경에서 힐버트 커브를 이용한 효율적인 Cloaking 영역 설정기법", 한국공간정보시스템 학회, 제11권, 1호, pp.115-126.

[15] T. Brinkhoff, 2002, "A Framework for Generating Network-Based Moving Objects," GeoInfor- matica, Vol.6 No.2, pp.153-180.

논문접수 : 2010.04.22
수 정 일 : 2010.08.04
심사완료 : 2010.08.06

김 형 일
2009년 전북대학교 컴퓨터공학과 (공학사)
2009년~현재 전북대학교 컴퓨터공학과 석사과정
관심분야는 공간 데이터베이스, 질의처리, 위치 보안을 위한 cloaking

이 아 름
2008년 전북대학교 컴퓨터공학과 (공학사)
2010년 전북대학교 컴퓨터공학과(공학석사)
관심분야는 공간 데이터베이스, 질의처리, 위치 보안을 위한 cloaking

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, 센서네크워크