TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KONUM TABANLI UYGULAMALARDA KONUM VE KONUM ÖRÜNTÜ MAHREMİYETİNİN SAĞLANMASI

YÜKSEK LİSANS TEZİ

Cansın BAYRAK

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Osman ABUL

ARALIK 2016

Fen Bilimleri Enstitüsü Onayı

.....
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

.....
Doç. Dr. Oğuz ERGİN
Anabilimdalı Başkan Vekili

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 141111033 numaralı Yüksek Lisans öğrencisi **Cansın BAYRAK** 'nın ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"KONUM TABANLI UYGULAMALARDA KONUM VE KONUM ÖRÜNTÜ MAHREMİYETİNİN SAĞLANMASI"** başlıklı tezi 19.12.2016 tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı: **Doç. Dr. Osman ABUL** .....
TOBB Ekonomi ve Teknoloji Üniversitesi

Jüri Üyeleri: **Prof. Dr. İsmail Hakkı TOROSLU (Başkan)** .....
Orta Doğu Teknik Üniversitesi

Yrd. Doç. Dr. Mehmet TAN .....
TOBB Ekonomi ve Teknoloji Üniversitesi

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Cansın BAYRAK

ÖZET

Yüksek Lisans Tezi

KONUM TABANLI UYGULAMALARDA KONUM VE KONUM ÖRÜNTÜ MAHREMİYETİNİN SAĞLANMASI

Cansın BAYRAK

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Osman ABUL

Tarih: ARALIK 2016

Mobil teknolojideki gelişmeler ve konum tabanlı uygulamaların yaygınlaşmasıyla, konum mahremiyeti önemli bir konu haline gelmiştir. Bu sebeple, Konum Tabanlı Servisler (KTS) bağlamında bu durum kapsamlı olarak incelenmiştir. Tipik bir çözümde, kullanıcılara bir konum mahremiyeti profili atanır ve kullanıcının konum mahremiyetini ihlal etmemek için kesin konumlar perdelenir. Bu süreç anlık konum mahremiyetini sağlayabilmek için kapsamlı bir şekilde tanımlanmaktadır. Öte yandan bu tür çözümler, kullanıcı yörüngesinde birden fazla KTS isteğinde bulunulması durumunda ek olarak bazı işlemler gerektirmektedir. Böyle durumlarda saldırgan, kullanıcıyı tam olarak bulabilmek için maksimum hız da dahil olmak üzere bazı arka plan bilgilerinden yararlanabilmektedir. PROBE ve diğer sistemler, gizlilik ihlallerini saptayarak ve gerekli durumda zaman gecikmesi veya geriye dönük konum gönderme işlemini uygulayarak çalışır. Veri madenciliği daha ciddi bir sorun olarak değerlendirildiğinden, kullanıcı için mahrem olarak nitelendirilebilen örüntüler, ilgili yörünge geçmişinden çıkarılabilir. Bu tez çalışması, konum örüntü mahremiyeti problemini konum mahremiyeti probleminin tamamlayıcısı olarak tanımlamaktadır. Sonuç olarak, kullanıcılar için mahrem konum örüntülerinin çevrimiçi biçimde ayıklandığı bir durum anlatılmaktadır. Bu çözüm, örüntü ihlallerini aşamalı olarak kontrol etmek için etkili bir dinamik programlama yaklaşımı kullanmaktadır. Ayrıca, bu tez çalışması kapsamında, bu çözümü uygulayan araç ve deneysel değerlendirme sonuçları da sunulmuştur.

Anahtar Kelimeler : Konum-tabanlı servisler, Konum mahremiyeti, Örüntü mahremiyeti, Servis istek geçmişi

ABSTRACT

Master of Science

PROVIDING LOCATION AND LOCATION PATTERN PRIVACY ON LOCATION-BASED APPLICATIONS

Cansın BAYRAK

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Osman ABUL

Date: DECEMBER 2016

Location privacy is becoming an important issue with the advances in mobile technology and widespread use of location based applications. Hence, it is extensively studied in the context of Location Based Services (LBSs). In a typical solution, users are assigned a location privacy profile and the precise locations are cloaked so that the location privacy is not compromised. The process is well-defined for snapshot location privacy. On the other hand, such solutions require patches in case of multiple LBS requests on the user trajectory. The fact is that the attacker can exploit some background knowledge including maximum velocity to exactly locate the user. PROBE and other systems operate in this fashion to detect any privacy violations and apply necessary time-delaying or post-dating. Taking the data mining as a stronger adversary, user-specific private patterns can be extracted from the respective trajectory history. This thesis defines location pattern privacy problem as a complementary to location privacy problem. As a result, a framework in which user-specific sensitive location patterns are sanitized online fashion is introduced. The solution uses an efficient dynamic programming approach to check pattern violations in incremental manner. The tool implementing this solution and an experimental evaluation are presented as well.

Keywords: Location-based services, Location privacy, Pattern privacy, Service request history

TEŞEKKÜR

Yüksek lisans eğitimim, gelecek planlamalarım, makale ve tez çalışmalarım boyunca beni yönlendiren ve yardımını esirgemeyen değerli hocam Doç. Dr. Osman ABUL başta olmak üzere, bu süreçte kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi'nin değerli öğretim üyelerine, en ufak bir ihtiyacımızda, kendi ihtiyacıymış gibi neşe ve özenle koşan değerli bölüm sekreterimiz Merve BAĞCI'ya, lisans ve yüksek lisans eğitimim süresince bana burs imkanı sağlayan TOBB Ekonomi ve Teknoloji Üniversitesi'ne, bir bölümünü tez çalışması olarak yaptığım 114E132 nolu proje kapsamında destek veren TÜBİTAK'a, mutlu ve huzurlu bir çalışma ortamında birlikte çalıştığımız ve desteklerimizi birbirimizden esirgemediğimiz değerli asistan arkadaşlarıma, eğitim öğretim hayatımın tamamını borçlu olduğum, üzerimde yegâne emeği bulunan babam Sami BAYRAK'a, tez çalışmam süresince benden desteğini esirgemeyen annem Gülhis TOPÇU'ya, bu süreçte bana destek olan İbrahim Burak DİKİLİ'ye, çalışmalarım için gereken huzurlu ortamı sağlayan ev arkadaşım Sencer Baki Sanberk YİĞCİ'ye, ne zaman yardımına ihtiyacım olsa beni asla geri çevirmeyen Gamze BAYRAK'a, aldığım her kararda beni sorgusuz sualsiz destekleyen Emel ÇOLAKOĞLU'na, dost, kardeş, can, candan öte tanımlamalarının hiç birisinin aramızdaki muhabbeti tam olarak anlatmaya yetmeyeceği Mehmet Ali AKAR'a, değerli büyüğüm Necmettin ŞAHİNLER'e ve isimlerini saymayı unuttuğum üzerimde emeği olan tüm sevdiklerim ve sevenlerime teşekkürlerimi sunarım.

İÇİNDEKİLER

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

KISALTMALAR

DBMS	: Database Management System - Veritabanı Yönetim Sistemi
GPS	: Global Positioning System - Küresel Konumlama Sistemi
IDE	: Integrated Development Environment - Tümleşik Geliştirme Ortamı
KMS	: Konum Mahremiyet Sağlayıcısı
KÖMS	: Konum Örüntü Mahremiyeti Sağlayıcısı
KTS	: Konum-Tabanlı Servis
PROBE	: Privacy-preserving Obfuscation Environment - Gizlilik Koruyucu Şaşırtma Ortamı
SECONDO	: Second Order Query Processor - İkinci Düzey Sorgu İşlemcisi
SOAP	: Simple Object Access Protocol - Basit Nesne Erişim Protokolü
TTP	: Trusted Third Parties - Güvenilir Üçüncü Şahıslar

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
D	Kullanıcı tanımlı uzaklık tahammül eşik değeri
I	Destek kontrol ikili değişkeni
k	Kullanıcı tanımlı önceki konum gönderme sınır değeri
KMS	Konum mahremiyeti sağlayıcısı
KÖMS	Konum örüntü mahremiyeti sağlayıcısı
KTS_{gec}	KTS geçmişi
KTS_{ist}	KTS isteği
M	Destek kontrol değişkeni
P	Mahrem örüntü
R	Perdeleme haritası
r	Perdelenmiş bölge belirteci
$\bar{r}$	Bölge sıralaması
s	KTS isteğiyle alakalı uydu bilgisi
T	Kullanıcı tanımlı zaman tahammül eşik değeri
t	Zaman bilgisi
$\bar{t}$	Zaman sıralaması
τ	Kullanıcı tanımlı zaman eşik değeri

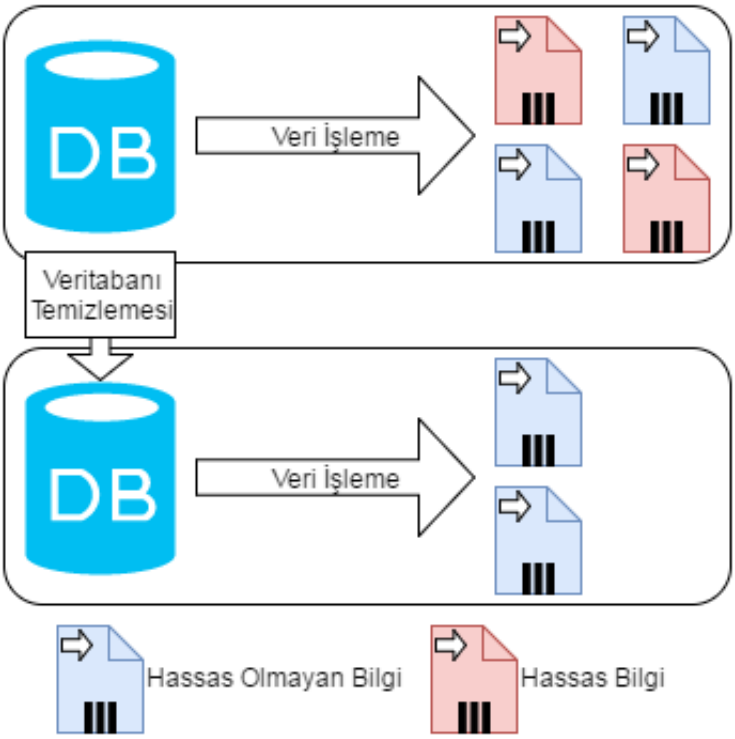
1. GİRİŞ

1965 yılında Gordon Moore tarafından yapılan bir öngörü olarak başlayan daha sonra bu kişinin ismiyle özdeşleşen Moore'nin Yasası [24], her 18 ayda bir, bir silikon levha üzerine yerleştirilebilecek olan transistör sayısı 2 katına çıkmaktadır, der. Bununla doğru orantılı olarak teknoloji de artmaktadır. Hızlı artan teknoloji önce bilgisayar adını verdiğimiz makinelerin küçülmesini sağlamış daha sonra bu makinelerin evlerimize girmesine vesile olmuştur. Bununla da kalmayarak ceplerimize kadar girerek, her bir kişiye ait taşınabilir cihazlarımız olmuştur. Bu rakam özellikle son yıllarda ciddi şekilde artarak devam etmektedir. 2007 yılından 2015 yılına gelene kadar dünyada taşınabilir cihaz kullanımı 12 kat artmış bununla beraber internet kullanım oranıysa yüzde 6.5'tan yüzde 43'e çıkmıştır [39].

Bu gelişmeler ışığında, veri çok değerli bir kavram olmaya başlamıştır. Bir çok firma, şirket ya da uygulama, kullanıcılarının verilerini tutmakta ve daha iyi hizmet verebilmek adına analiz etmektedir. İstatistiksel sonuçlara varabilmek, çeşitli çalışmalarda kullanılabilmek için ya da reklam, pazarlama gibi değişik ticari kaygılar taşıyarak analiz edilen veri, zamanlar paylaşılmaya başlamıştır. Paylaşılan veride bazı veri madenciliği teknikleri kullanılarak ulaşılabilecek hassas, kullanıcının ya da veriyi paylaşanın mahrem olarak nitelendirebileceği bilgiler bulunabilir. Bu gibi durumlarda veritabanı paylaşıma açılmadan önce hassas bilgiden arındırılmalıdır. Hassas bilgiden arındırma işlemi, veritabanından o bilgilerin temizlenmesi ya da değiştirilmesi şeklinde olabilir. Şekil 1.1, temizlenme işlemi uygulanmadan önce ve temizlenme işlemi uygulandıktan sonra, aynı veri madenciliği teknikleriyle ulaşılabilecek verileri görselleştirmektedir. Temizleme işleminden önce ulaşılabilen hassas bilgilere, temizleme işlemi sonrasında ulaşılamamaktadır.

Küresel Konumlama Sistemi (GPS) teknolojisinin de gelişmesiyle beraber artık hemen hemen herkes her gittiği yerde internete ve GPS'ye bağlı olabilmektedir. Böylelikle insanlar her an internetten bir bilgiye erişebilmekte, Google ya da Yandex gibi harita servisi veren yolbul (navigasyon) uygulamaları sayesinde bir yerden bir yere nasıl gidebileceklerine bakabilmekte ya da kendilerine en yakın eczaneyi bulabilmektedirler. İnsanların konumlarına göre farklı bilgilere erişebilme durumu, konum kullanan uygulamaların ve konum-tabanlı servislerin (KTS) sayısında bir artışa neden olmuştur.

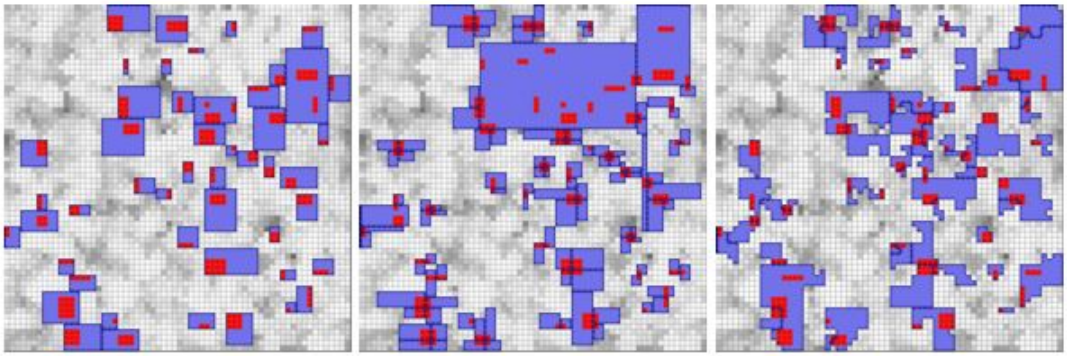
Konum-tabanlı servisler ve konum bilgisi kullanan uygulamalar çeşitlilik göstermektedir. Google Haritalar, En Yakın gibi konum-tabanlı servisler, kullanıcıların konum isteklerine göre uygun cevapları vermektedir. Google Haritalar sayesinde kullanıcı bulunduğu konumdan başka bir yere en yakın hangi yol üzerinden gidebileceğini görebilmektedir ya da En Yakın uygulamasıyla kendisine en yakın nöbetçi eczaneyi veya en yakın sinema salonunu bulabilmektedir. Twitter, Swarm, Facebook, Instagram, Snapchat, Pokemon GO gibi uygulamalar da kullanıcının



Şekil 1.1: Veritabanı temizleme öncesi ve sonrası

konum bilgilerini kullanmaktadır. Swarm, Facebook gibi uygulamalarda kullanıcılar doğrudan bulundukları konumu bildirirlerken, Twitter, Instagram gibi uygulamalarda, paylaşılan bir fotoğraf ya da yazının daha sonra nereden paylaşıldığının unutulmaması adına konum etiketi koymaktadırlar. Pokemon GO gibi uygulamalardaysa kullanıcılar bulundukları bölgede dolaşarak arttırılmış gerçeklik teknolojisiyle oyun oynayabilmektedirler. Bu uygulama ve oyunlar bir çok kişi tarafından kullanılmaktadır. Özellikle Pokemon GO, çıktığı ilk günden beri ciddi bir başarı sağlamış ve bir çok kişi tarafından kullanılmaya başlanmıştır [7]. İnsanlar her gittiği yere GPS bilgileri açık olarak gitmeye ve her gittikleri yerde bu uygulama aracılığıyla Pokemon (Pocket Monsters) adı verilen küçük canavarlardan yakalamaya çalışmaya başlamışlardır. Araştırmacıların bir süredir üzerinde çalıştığı "Konum tabanlı uygulama ve servisler güvenli midir?" sorusu, Pokemon GO uygulaması sayesinde artık herkesin merak ettiği bir konu haline gelmiştir. Konum kullanan uygulamalarda, kişinin uygulamayı kullanıp kullanmaması kişisel bir tercih olmakla beraber, gizlilik ya da mahremiyet durumunu önemseyen kişilerin kullanmaması gibi aşırı basit bir yaklaşım yapılabilir ancak bu yaklaşım konum-tabanlı uygulamalarda işe yaramamaktadır. Bir gece herhangi bir yerde çevresinde insan bulamayan bir kişi, en yakın nöbetçi eczaneyi bulmak isteyebilir ve kaybedecek zamanı olmayabilir. Bu durumda gizlilik ve mahremiyet durumunu düşünerek servisi kullanmaktan vazgeçemez. Bu gibi durumlarda konum mahremiyeti araştırmaları devreye girmektedir.

Konum mahretini sağlayabilmek için, konum-tabanlı servisler kullanılırken gerçek konum yerine perdelenmiş konum kullanılır. PROBE [12] ve benzeri [13, 27, 46] konum perdeleme yaklaşımları şu şekilde çalışır: İlk olarak, bir mahremiyet profil havuzu oluşturulur. Bu havuzda, belli sayıda farklı mahremiyet profilleri bulunur.



Şekil 1.2: Farklı tekniklerle oluşturulmuş perdeleme haritaları [13]

Sisteme yeni bir kullanıcı geldiğinde var olan profillerden birini seçer ya da kendisi bir profil oluşturur. Her bir mahremiyet profili için tek bir konum perdeleme haritası oluşturulur. Şekil 1.2, farklı tekniklerle oluşturulmuş perdeleme haritalarını görselleştirmektedir. Aynı coğrafi bölgede bulunan ve benzer mahremiyet profiline sahip kişilerin konum perdeleme haritaları da aynıdır. Başka bir deyişle kullanılan perdeleme haritası kullanıcıları denklik sınıflarına ayırır ve her bir sınıf bir öbektir. Akla gelen ilk soru, aynı öbekteki her bir kullanıcının konumla ilgili mahremiyet kaygısının aynı olup olmadığıdır. PROBE ve benzeri yaklaşımlar aynı perdeleme haritasına sahip kişiler arasında konum mahremiyeti bakımından bir fark gözetmez. Fakat bu tez çalışmasında, tam anlamıyla kişiselleştirme için, konum mahremiyeti yanında konum örüntü mahremiyetinin de adreslenmesi gerektiği üzerinde durulmuştur. Böylelikle konum mahremiyeti hususunda tam anlamıyla bir kişiselleştirmeden söz edilebilecektir.

K_1 ve K_2 aynı perdelenmiş haritayı kullanan iki farklı kullanıcı olsun ve durum gereği aynı KTS istek sıralısına sahip olsunlar. Ayrıca, varsayalım ki bu isteklerdeki bazı bölgeler K_1 kullanıcısı için mahrem konum içersin (mesela kendisinin evi, işyeri, sık gittiği yerler vb.) ancak K_2 için kişisel bir konum içermesin. Bu durumda K_1, K_2'ye nazaran KTS isteklerinden daha rahatsız olacaktır. Bu durumda K_1, kendi "kişisel konumları için bir örüntü tanımlayıp bunu mahrem bilgi olarak sisteme verdiğinde K_1 ve K_2 perdeleme haritaları aynı olmasına rağmen farklı şekilde ele alınacaktır. Bu sorunu çözebilmek adına yine bu bölümde bahsedilen veritabanının hassas, mahrem bilgilerden arındırılması yöntemi kullanılabilmektedir ancak bu yöntem geriye dönük, çevrimdışı bir yöntem olacağı için, veritabanına sahip olan servis sağlayayıcısının kötü niyetli (saldırgan) olarak düşünüldüğü durumda yeterli kalmamaktadır.

Çevrimdışı, geriye dönük arındırma işleminden ziyade, konum-tabanlı servis kullanıcısının servis isteklerinin çevrimiçi olarak değerlendirmesi daha sağlıklı bir çözüm sunmaktadır. Kullanıcı konum-tabanlı servislerden yararlanmak istediği anda, konum mahremiyetiyle beraber, konum örüntü mahremiyetinin de sağlandığından emin olunmadan servis isteği, servis sağlayıcısına gönderilmemelidir. Eğer konum örüntü mahremiyetini ihlal edecek bir servis isteği yapılmamışsa gönderilmeli, aksi halde istek bazı işlemlerden geçirilmelidir. Bu işlemler sırasıyla, servis isteğinin bir süre geciktirilmesi, servis isteğinin kullanıcının daha önce bulunduğu konumlardan birisindeymişcesine gönderilmesi ya da servis isteğinden vazgeçilmesi işlemleridir. Kullanıcı servis isteğinin geciktirilme süresine, tahammül edebileceği en fazla

bekleme süresi değişkeniyle (T) ve önceki konumlarından birinin gönderilmesineyse, tahammül edebileceği en fazla uzaklık değişkeniyle (D) karar verebilmektedir. Ayrıca önceki konumlarından birinin gönderilmesi işleminde, en fazla kaç konum uzağa gidilebileceği (k) bilgisi de değişkendir. Kullanıcının bir önceki konumuna değil de bir önceki k konumuna bakılmaktaki mantık, kullanıcının son servis isteğindeki konumun şu anki konumuna en yakın konumu olmak zorunda olmaması bilgisinin kolayca anlaşılabiliyor olmasıdır. Kullanıcının yapmış olduğu servis isteğinin, kullanıcının konum örüntü mahremiyetini ihlal edip etmeyeceğine, yine kullanıcının belirlemiş olduğu mahrem örüntüler kümesindeki örüntülerle eşleme yaparak karar verilmektedir. Kullanıcının mahrem olarak belirlediği örüntülerin, kullanıcının rotasıyla eşlenmesi/kontrolü esnasında, her bir örüntü için ayrıca belirlenmiş olan τ değeri büyük bir rol oynamaktadır. τ, örüntünün süre hassasiyetini belirlemektedir. Kullanıcının yapmış olduğu istek sonucunda, kullanıcın mahrem örüntü kümesindeki örüntülerden herhangi bir tanesi bile gözüküyorsa, ihlal var demektir ancak hiç birisi gözükmüyorsa, servis isteği güvenlidir. Burada dikkat edilmesi gereken nokta, mahrem örüntü kümesi içerisindeki herhangi bir örüntünün ihlal edilmesi durumunda, servis isteği üzerinden yapılan işlemler sonucunda, istek o örüntü için güvenli hale getirilirken başka bir örüntüyü ihlal edecek hale gelmiş olabilir. Bu gibi durumlarda işlemden geçirilen istek, tekrar olarak tüm örüntüler için değerlendirilmelidir.

1.1 Tez İçeriği

Bu çalışmada kullanıcılar için perdelenmiş konum haritası atandıktan sonra kullanıcılara mahrem konum örüntüsü tanımlama imkanı sağlayarak bu örüntülerin konum-tabanlı servis isteklerinde oluşmamasını sağlayan bir yaklaşım önerilmiştir. Yani, bu çözüm PROBE ve benzeri yaklaşımlara alternatif değil tamamlayıcı niteliktedir.

Bu tez çalışması şu şekilde düzenlenmiştir: Bölüm 1'de geçmişten günümüze gelirken konum ve konum örüntü mahremiyeti konularının neden ve nasıl gündemimize girdiği incelenmiş, genel olarak tez çalışması özetlenmiş ve gerekli bilgiler verilmiştir. Bölüm 2'de, yapılan literatür taramasının ayrıntılarından bahsedilmiş, veri ve bilgi mahremiyeti konularının nasıl literatüre girdiği incelendikten sonra, değişik kapsamlarda ele alınan konum servisleri, uygulamaları ve mahremiyet konuları hakkında yapılan çalışmalara değinilmiştir. Bölüm 3'de çevrimiçi konum mahremiyetinin nasıl sağlanabileceği ve geçmiş çalışmalarda nasıl sağlandığıyla ilgili örnekler verilerek bu konu anlatılmıştır. Çevrimiçi konum mahremiyetinin, kişisel ihtiyaçlara göre yeterli olmadığı motivasyonunun sağlanabilmesi için mevcut örnek incelenmiş ve konum örüntü mahremiyeti konusunun neden önemli olduğu anlatılmıştır. Bölüm 4'te geçmişte çevrimdışı geriye yönelik temizleme şeklinde örnekleri olan konum örüntü mahremiyeti konusuna getirilen çevrimiçi çözüm tanıtılmış, formülize edilmiş ve ayrıntılarıyla anlatılmıştır. Çevrimiçi örüntü mahremiyetinin sağlanabilmesi için kullanıcı tarafından servis sağlayıcısına gönderilen isteğin, mahrem örüntü oluşturup oluşturmadığı kontrolünün nasıl yapılacağı detaylandırılmış ve sonrasında mahrem örüntü oluşturabilecek olan bir istek karşısında nasıl davranılması gerektiği adreslenmiştir. Bölüm 5'te, geliştirilen uygulamanın yapısı, sunucu, mobil uygulama ve simülatör uygulaması anlatılmış, bu

yapı gerçekleştirilirken kullanılan teknolojilere değinilmiş, uygulama test ve sonuç çıkarma aşamasında kullanılan verinin nasıl elde edildiği, hangi işlemlerle temizlenip analiz edildiğinden bahsedilmiştir. Simülatör uygulamasından alınan deneysel sonuçlar incelenmiş, görsellik sağlayabilmek adına çizelge ve şekillerle sonuçlar paylaşılmıştır. Bölüm 6'da, genel sonuçlar irdelenmiş, açıklanmış ve bu tez çalışmasına devam niteliğinde yapılabilecek çalışmalardan bahsedilerek tez çalışması sonlandırılmıştır.

2. İLGİLİ ÇALIŞMALAR

Günümüz şartlarında güvenlik, gizlilik ve mahremiyet konuları bir çok alanda mutlaka sağlanması gereken konular olarak karşımıza çıkmaktadır. Artık durum o kadar ciddi bir hal almıştır ki, bazı kişiler aldıkları hizmetin kalitesinden ödün verebilmek adına bile olsa gilzilik ve mahremiyeti en üst seviyede tutmak istemektedirler. Bu tez çalışmasına temel kabul edilebilecek olan araştırma aşamasında incelenen farklı konularda çalışmalar mevcuttur. Bu çalışmaları farklı başlıklar altında toplamak mümkündür.

2.1 Veri ve Bilgi Mahremiyetine Yönelik Çalışmalar

Kullanıcısına herhangi bir serviste bulunan bir firma, şirket ya da uygulama, değişen teknoloji ve zaman koşullarıyla beraber, kullanıcılarının bilgilerini toplamaya, tutmaya başlamışlardır. Bu verilerle beraber, veri madenciliği teknikleriyle anlamlı sonuçlar çıkarıp bir takım istatistiki sonuçlara varılabilmektedir. Bu sonuçların herkes tarafından kullanılabilmesi için, verilerin ticari ya da bilimsel kaygılarla paylaşılması, mahremiyet sıkıntılarını doğurmaya başlamıştır [35]. Bazı durumlarda verileri paylaşan kişi ya da firmanın mahrem olarak nitelendirebileceği bilgiler olabileceği gibi bazı durumlardaysa, verilerin içerisinde bilgisi bulunan kullanıcıların mahrem olarak nitelendirebileceği bilgiler bulunabilir. Bu durumlarda, verilerin yayınlanmasından önce mahrem/hassas olarak nitelendirilen verilerin mevcut veri kümesinden çıkarılma ya da değiştirilme işlemi uygulanabilir. Buna çözüm olarak yapılan bir yaklaşımda, veritabanından bilgiler dışarıya yayınlanacağı zaman, hassas kural tanımlamalarına göre, veritabanında sık görülen öğeler kümelenerek veritabanından çıkarılmıştır [5] ve teorik bir yaklaşım yaparak deneysel sonuçlar paylaşılmıştır. Çıkarılacak ya da gizlenecek mahrem bilgilere daha iyi karar verebilmek adına, önceki çalışmaya devam olarak düşünülebilecek, bir arındırma matrisi tanımlamasıyla beraber veri madenciliği tekniklerini kullanarak gizlilik kaygısıyla temizlenmiş yeni bir veritabanı elde edilebilmektedir [28]. Veritabanından bilgileri çıkarırken uygulanan tekniklerde, veritabanının bozulma durumu mevcuttur, sonuçta arındırılmış bir veritabanıyla orjinal veritabanının aynı olması beklenemez. Veritabanının arındırılması işleminde, verilerin çıkarılmasındansa değiştirilmesinin veritabanının orjinalliğini daha az seviyede bozacağı gösterilerek, daha iyi gizlilik sağlamaya çalışırken aynı zamanda mevcut veritabanına en iyi kalitede benzeyen arındırılmış veritabanı oluşturma üzerine yapılan çalışmalar mevcuttur [42]. Diğer çalışmalara temel kabul edilebilecek, bilgiyi paylaşırken gizliliğinin korunması konusuna k-anonimlik bakış açısıyla yaklaşan [38] çalışması, her bir bilgiyi en azından k varlıkla eşleştirerek minimum genelleme kavramını sunmaktadır. Ayrıca istatistiksel veritabanlarında da gizlilik konusu araştırılmış ve bazı çalışmalar yapılmıştır. İstatistiksel veritabanlarının nasıl oluşturulması gerektiği ya da güvenlik

kısıtlamaları incelenmiştir [8, 36]. Bu çalışmalara devam niteliğinde düşünülebilecek olan bir önyargı-düzeltme mekanizması geliştirilmiş ve küçük sorgu kümesinden kaynaklanabilecek problemler giderilmeye çalışılmıştır [2].

Gizlilik sağlayabilmek adına sunulan k-anonimlik modeli [43], paylaşılan bilgideki anahtar niteliği olmayan bilgileri gizler ya da genelleştirir. Bu model geliştirilerek, her bilginin tanımlayıcıları oluşturulmuş ve kümelenmiştir [3]. Bu yaklaşımda, bir bilginin bir kümeye ait olabilmesi için önceden belirlenmiş tanımlayıcılardan en az değişken olarak belirtilen miktarda tanımlayıcı içermesi gerekmektedir. Ayrıca bu kümeleme yöntemine ek olarak sabit faktörlü yaklaşım algoritmaları sunulmuştur. Bu çalışma sayesinde bir bilgi, verilen tanımlayıcıların niteliklerine göre kendisinden başka en az k-1 bilgiden ayırt edilememektedir. K-anonimlik algoritmaları bazı saldırı durumlarına karşı yeterli kalamamaktadır. Bunlar, bilgideki çeşitliliğin az olduğu durum ya da kümelemeyle ilgili daha önceden elde edilebilen arka plan bilgileriyle beraber saldırılma durumlarıdır. Bu sorunları çözebilmek adına l-çeşitlilik çalışması önerilmiştir [33]. L-çeşitlilik, konum tabanlı servislerde de gizlilik sağlayabilmek adına da kullanılmıştır [31].

2.2 Konum Tabanlı Servislerde Yapılan Çalışmalar

GPS hayatımıza girdiğinden beri, konum tabanlı servis ve uygulamaların sayısındaki artış giderek devam etmektedir. Bu da beraberinde bu alana yönelik yapılan çalışmaların sayısını arttırmıştır. Mekansal-zamansal veri kavramları hayatımıza dahil olmuş, konum uygulamaları bu mekansal-zamansal veri türlerini kullanarak sunucu-istemci arasında haberleşme gerçekleştirmeye başlamışlardır. Mekansal-zamansal veri türleri belli noktalar ve belli zaman için mevcut olan soyut veri türleri olarak tanımlanmışlardır [15]. İlerleyen ihtiyaçlar mekansal-zamansal veri türleriyle bir veritabanı yönetim sisteminde (DBMS) nasıl daha verimli çalışılabileceği sorularını doğurmuş ve ortaya mekansal-zamansal cebir kavramı çıkarılmıştır. Basit ilişkisel cebirin, saklama ve sorgulamadaki başarısı, mekansal-zamansal veri türlerine de uygulanabilir mi sorusuna cevap aranmış ve bu konuda çalışmalar yapılmıştır [32]. SECONDO [21, 22] kullanarak yeni türler tanımlanabilen ve yeni işlemler kümesi tanımlanabilen yeni bir veritabanı yönetim sistemi geliştirilmiştir.

Geliştirilen mekansal-zamansal veritabanı yönetim sistemleriyle beraber, konum servislerinden yararlanan nesnelerin hareketleri incelenmeye başlanmış ve hareketli nesneler için öngörülebilir durumlar araştırılmıştır. Hareketli bir nesnenin harekete başlama ve hareketi bitirme noktalarıyla beraber, nesnenin en fazla çıkabileceği hız, trafik bilgisi, yol bilgisi gibi diğer bilgiler birleştirilerek, hareketli nesnelerin rotaları öngörülebilecek olup bu yaklaşımla beraber test ve deneyler adına büyük veri kümeleri üretilmeye başlanmıştır [6]. Hareketli nesnelerin tamamı öngörülebilir şekilde hareket etmemektedir. Bazı dış etkenler, ani hareket değişimlerine sebep olabilirler. Bu yüzden hareketli nesnelerin hareketlerini açıklayabilmek adına hareketli nesne sorguları üzerinde çalışmalar yapılmış ve sorgu optimizasyonu kavramı ortaya atılmıştır. [14]. Ayrıca bu çalışmada hareketli nesneler için bir veri modeli ve sorgu dili tanıtılmıştır.

Hareketli nesnelerin rotaları üzerinde yapılan çalışmalar araştırmacıları, belli örüntüleri incelemeye yönenlendirmiştir. Büyük miktarda hareketli nesnelerin oluşturdukları rota örüntüleri, belli dış faktörleri de göz önünde bulundurarak incelenmeye başlanmış, özellikle belli örüntüleri oluşturan hayvanların hareketleri keşfedilmeye başlanmıştır [30]. MoveMine çalışmasında hareketli nesnelerin rotaları veri madenciliği teknikleriyle incelenmiş, hayvanların eğilimleri, hareket kalıpları yorumlanmaya çalışılmış ve hayvanlarla ekolojik sistem arasındaki etkileşim incelenmiştir. Böylelikle mevcut ekosistemin devamlılığı için geliştirilmesi gereken anlayışa dair sonuçlara varılmaya çalışılmıştır. Devamında, mekansal-zamansal rota örüntülerinin incelenmesi çeşitli bilgisayar bilimleri alanlarıyla birleştirilmeye başlanmış ve bazı tahmin kümeleri oluşturulabilmek ve onları keşfedebilmek adına görsel analiz tekniklerinden yararlanılmıştır [37]. Hareketli nesnelerin rotaları ve oluşturdukları örüntüler, bir başka boyutta incelenerek, insanların hareketleri üzerinde durulup bazı problemlerin çözülebileceği öngörülmüştür. Konum bilgileri kullanılarak şehirlerin trafik sorunları üzerlerinde durulmuş, trafik sorununa çözümler üretilmeye çalışılmıştır. Büyük şehirlerin karayolları için geliştirmeler yapılmıştır [4].

Konum verilerinin işlenmesi araştırmaları, hareketli nesnelerin rotalarına ve dolayısıyla insanların hareketlerine doğru yönelmeye başladıkça ortaya mahremiyet sorunları çıkmaktadır. Hareketlerinin incelenmesinden rahatsız olabilecek bir kullanıcı, farkında olmadan verdiği bir izinle konum bilgilerini paylaşabilmektedir. Bu da mahremiyeti zedelemektedir. Öte yandan, konum bilgilerini paylaşmasa bile, her servis sağlayıcısının iyi niyetli olduğunu ya da bu kullanıcının ya da servis sağlayıcısının bünyesindeki bilgilere yönelik bir saldırıda bulunulmayacağı garantisi de yoktur. Bu yüzden zamanla konum mahremiyeti kavramı oluşturulmuştur.

2.3 Konum Mahremiyetine Yönelik Çalışmalar

Hareketli nesnelerin rota bilgileri üzerinde çalışmalar yapılıp, bu bilgiler ışığında istatistiksel değerlendirmeler yapıldıkça ve bilgiler servis sağlayıcılarının sunucularında tutulmaya başlandıkça, konum gizliliği ve konum mahremiyeti kavramları ortaya çıkmıştır. Kullancıların konum ve konum rota bilgileri, servis sağlayıcılarıyla paylaşılırken, kimlikleriyle eşleştirilmemeleri bir seçenek olarak görülebileceği gibi bazı sahte konum ya da konum gizleme gibi teknikler de geliştirilmiş, araştırılmıştır. Bunlar çevrimiçi konum mahremiyeti olarak düşünülebilir. Konum rotaları üzerinde bazı çevrimdışı işlemler uygulanabilir, böylelikle istatistiksel sonuçlar çıkarılabildiği gibi geleceğe yönelik tahminler de yapılabilir. Ayrıca konum rotalarında ortaya çıkabilecek bazı konum örüntüleri, kişiden kişiye değişmekle beraber, mahrem olarak düşünülebilir. Bu durumda kişilerin konum rotalarında bu örüntülerin bulunması engellenmelidir.

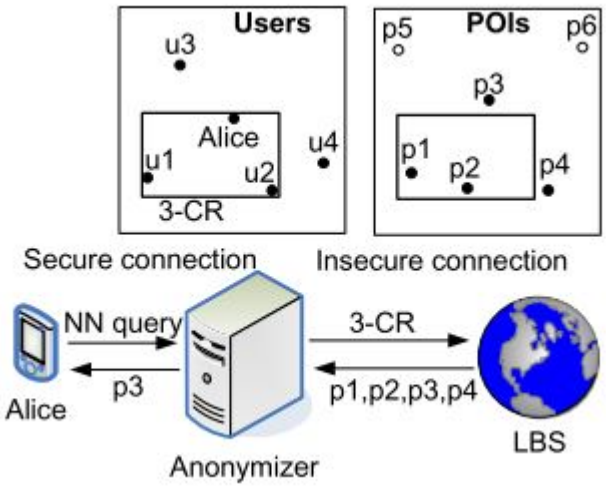
2.3.1 Çevrimiçi konum mahremiyetiyle ilgili yapılan çalışmalar

Hareketli nesnelerin konum bilgilerinin mahremiyetinin sağlanabilmesi açısından bu bilgiler anonimleştirilmeye çalışılmıştır. Buradaki düşünce, mevcut konum bilgileriyle, kişi bilgilerinin eşleştirilmesini engelleyerek kişinin mahremiyetini

sağlayabilmektir. Bu kapsamda k-anonimlik kullanan CliqueCloak algoritması tanıtılmıştır [16]. Bu algoritma sayesinde servis isteklerinin en az k adet kullanıcıdan birisinden geldiği bilinebilmekte, tam olarak kim tarafından olduğu bilinememektedir ancak burada k değerinin küçülmesine ve örüntülerin incelenmesine bağlı olarak yapılabilecek saldırılarda küme daraltma sayesinde kişi bilgilerinin eşleştirilebilme ihtimali doğmaktadır. Devamında da bu konuda çalışmalar yapılmış [34] olsa da potansiyel saldırılar için kesin çözüm vermemektedir. Anonimlik yanı sıra komple gizlilik konusu da araştırmalara konu olmuştur. Konum gizliliğinin sağlanabilmesi adına en uygun teknikler araştırılmış, TTP tabanlı güvenilir üçüncü şahıslara vurgu yapılmaktadır [41]. Ayrıca hesaplamaya dayalı gizlilik teknikleri incelenip, anonimlik gibi diğer algoritmaların üzerine şifreleme, erişim kontrolü gibi teknikler dahil edilebiliyor [27]. Bu yaklaşım, potansiyel saldırılara küçük bir çözüm olabilecek gibi dursa da, servis sağlayıcısını potansiyel saldırgan olarak düşündüğümüz durumda tam mahremiyet sağlayamamış oluyor.

Sorunların sadece belli başlı isteklerden dolayı kaynaklanmadığını, ayrıca konum sıralamalarından oluşan rotaların mahremiyetinin de sorun oluşturduğunu anlayabiliyoruz. Kişilerin konum tabanlı servislerden yararlanmasıyla beraber, kısmi güzergâhları açığa çıkmakta ve ciddi bir gizlilik ihlali olmaktadır. Bu konuya çözüm olarak olabildiğince doğru veri gönderirken, gizlilik ihlalini de önlemeye yönelik çalışacak olan bir veri süpürme tekniği tanıtılmıştır [44]. Bazı durumlarda kullanıcıların rota bilgileri, kimlikleriyle bağdaşlaştırılmadan istatistiksel sonuçlara varabilmek adına kullanılabilmektedir. Buna örnek olarak şehir içi trafik durumu verilebilir. Bu gibi durumda kullanıcıların rotalarının, kimlikleriyle bağdaşlaştırılmamasını kesin olarak önleyebilmek çok önemlidir. Bu konu hakkında, konum servis isteği gizliliği ve rota anonimleştirmesiyle ilgili genel bir bakış sunan, sonrasında yaklaşım tekniklerini gözden geçirerek zayıf ve kuvvetli yanlarını inceleyen, k-anonimlik çalışmalarıyla destekleyen çalışmalar da mevcuttur [18]. Bu yaklaşım Şekil 2.1'de görselleştirilmiştir. Bu çalışmalara ek olarak, yine şehir planlama, akıllı ulaşım, işletme analizi gibi konum rotaları üzerinden istatistiksel sonuç çıkaran, çıkarmaya çalışan uygulamalardaki, konum rota bilgilerini paylaşırken oluşabilecek gizlilik ihlallerine yönelik ek teknikler incelenmiştir [10]. Rota gizliliğine yönelik başka yaklaşımlar da bulunmaktadır. Rota eşleşmeleri üzerinde yapılan çalışmalar [23], rota gizliliğini en uygun şekilde sağlayabilmek adına optimizasyon araştırmaları [45] ve kullanıcının gerçek konumlarının değiştirilerek yerelleştirme saldırılarıyla ilgili uygun teknikleri araştıran çalışmalar da mevcuttur [40].

Kişisel konum mahremiyetinin çok daha etkili sağlanabilmesi için konum perdeleme yöntemleri sunulmuştur. PROBE çalışması bunun en güzel örneklerinden bir tanesidir [12]. Bu çalışma, kullanıcının şaşırtma veya gizleme olarak düşünülebilecek bir teknikle, tam konumundan ziyade, bulunduğu konumu çevreleyen bir perdelenmiş bölge ile servis isteklerini yapmasına yardımcı olur. Kullanıcıların kişisel mahremiyet profillerine göre her kullanıcıya perdelenmiş bir harita tahsis edilmektedir. Yine bu çalışmada, kullanıcılara, kendi mahremiyet profillerine göre en verimli şekilde perdelenmiş harita ataması yapan teknikler incelenmekte ve değerlendirilmetedir. Bu çalışmanın devamında kullanılan perdelenmiş bölgede bazı problemler oluşabilmektedir. Perdelenmiş bölgelerin alanı, coğrafi koşullar gibi bazı arka plan bilgileri göz önünde bulundurularak daraltılabilir. Daha açık bir deyişle, perdelenmiş



Şekil 2.1: Mekansal k-anonimlik [18]

bir bölgeden servis isteğinde bulunan kişi için, bulunduğu bölgenin büyük bir bölümü insanların girmesine engel teşkil edebilecek bir bölge ise, potansiyel saldırgan bölgeyi daraltabilir. Bu sorunu çözebilmek adına da yapılan çalışmalar mevcuttur [13]. Coğrafi arka plan bilgisinden ziyade, hız sınırlamaları da ciddi bir şekilde saldırılara davetiye çıkarabilmektedir. İki tane perdelenmiş bölgenin önce birisinden ve kısa bir süre sonra ikincisinde servis isteği gönderen kullanıcı için, eğer bu süre zarfında, bu bölgelerin birbirlerine en uzak noktaları arasındaki mesafeyi alabilme ihtimali yoksa, en fazla hız sınırı dikkate alınarak, bu iki bölgede de daraltma uygulanabilir ve kullanıcının bulunduğu konum açığa çıkabilir. Bu probleme çözüm aranmış ve deneysel olarak gösterilmiştir [19]. Perdeleme haritası yöntemlerini kullanan ve daha önce araştırılan potansiyel saldırılar da göz önünde bulundurularak buna ayrıca şehir yol ağı kısıtlamalarını ekleyerek daha verimli sonuçlara ulaşan farklı algoritmalar araştırılarak deneysel sonuçlara ulaşılmıştır, [46].

2.3.2 Çevrimdışı konum örüntüleriyle ilgili yapılan çalışmalar

Rotaların çevrimdışı işlenmesiyle bazı sonuçlar elde edilebilir. Mekansal-zamansal veri türü kuralları, geleneksel veri madenciliği teknikleriyle birleştirildiğinde veriler üzerinden anlamlı sonuçlar çıkarılabilir. Bir gezinti uygulaması için sık gidilen güzergâhlar ve kullanıcıların rotaları incelenerek sık güzergâhların verimli bir şekilde çıkarılabilmesi sağlanmıştır [20]. Rotaların incelenmesi, makine öğrenmesi teknikleriyle bir hareketli nesnenin ileride nasıl davranacağının tahmin edilmesi için kullanılabilir. Bu konuda bazı teknikler önerilmiş ve kıyaslamaları yapılmıştır [25]. Ayrıca rota verisi madenciliğinde, performans geliştirmeye yönelik çalışmalar da yapılmıştır. Öncelikle anlamlı bölgeleri ayıklayıp daha sonra üzerinde madencilik teknikleri uygulandığında performans artışı gözlenmiştir [26].

Öte yandan, konum örüntülerinin işlenmesinin yanı sıra, kullanıcının konum rotası arasında mahrem olarak değerlendirdiği ya da daha sonradan mahrem olarak değerlendirebileceği örüntüler mevcut olabilir. Bu örüntüleri geçmiş rota bilgilerinden

temizleyebilmek adına geliştirilen polinom zamanda bir arındırma algoritması tanıtılıyor [1]. Bu algoritma, kullanıcıların mahrem olarak belirlediği örüntüleri rota geçmişinden temizlemekte ve bunu yaparken rota bilgisini mümkün olan en az düzeyde değiştirmektedir.

3. ÇEVRİMİÇİ KONUM MAHREMİYETİ

Çevrimiçi konum mahremiyeti PROBE [12] ve benzeri çalışmalarda anlatılmıştır. Bu konu, PROBE özelinde aşağıdaki şekilde ele alınmıştır.

3.1 PROBE Çatısı

Çevrimiçi konum mahremiyeti sağlanabilmesi için literatürde tanımlanan PROBE [12] ve benzeri çalışmalarda, kullanıcılar, öncelikle kendilerine bir harita tanımlanabilmesi adına bölge seçerler. Kullanıcılar, konum mahremiyetlerinin sağlanabilmesi için yapacakları servis isteklerinde servis sağlayıcılarına, bulundukları noktasal konum yerine, önceden oluşturulmuş bir perdelenmiş bölgeyi gönderirler. Kullanıcıların seçtikleri harita üzerinde perdelenmiş bölgelerinin oluşturulabilmesi için, kullanıcıların mahrem olarak nitelendirebileceği hastane, gece kulubü ya da camii gibi semantik konumlara göre kendilerine bir mahremiyet profili oluşturmaları beklenir. Kullanıcıların oluşturdukları mahremiyet profiline göre, her bir kullanıcıya perdelenmiş bölgelerini içeren perdelenme haritası atanır. Çevrimiçi konum mahremiyeti durumu irdelendiği zaman, potansiyel saldırı durumları ve şehir yol ağı kısıtları da göz önünde bulundurulmalıdır. Örneğin, arka arkaya iki bölgeden servis isteğinde bulunan bir kullanıcının bu iki isteği arasında geçen süre, ilgili iki perdelenmiş bölgenin birbirlerine en uzak noktaları arasında seyahat edilebilecek bir süre olması gerekmektedir, aksi durumda perdelenmiş bölge daraltılarak kullanıcının bulunduğu konum hakkında daha kesin tahminler yapılabilir.

3.2 Konum Mahremiyet Profili

Kullanıcının mahremiyet gereksinimleri konum mahremiyet profilinde özetlenmiştir. FT_s (feature types) kullanıcı tanımlı mahrem semantik konum türleri ve T mahremiyet tercihleri kümesi olmak üzere, konum mahremiyet profili (FT_s, T) şeklinde gösterilir. Bir konum mahremiyet profili, kullanıcının belirli mahrem semantik konum türlerine olan hassaslığını tanımlamaktadır ayrıca $ft \in FT_S$ ve τ, ft'nin mahremiyet eşik değeri olmak üzere $(ft, \tau), \tau \in (0,1)$ şeklini alır. PROBE çatısında, $(ft,1)$ ve $(ft,0)$ durumları hariç tutulmuştur çünkü $(ft,1)$ tanımlaması ft'nin mahrem olmadığını tanımlamaktadır. Öte yandan $(ft,0)$ tanımlamasıysa çok sıkı bir tanım oluşturmaktadır ve sağlanabilmesi için ft'nin boş olması gerekir.

PROBE çatısındaki konum mahremiyet profili tanımlaması bir örnekle daha rahat anlaşılabilir. Camii gibi ibadethaneleri ve sağlık kuruluşlarını içeren, bir kullanıcıya ait mahrem olarak nitelendirilebilen yerlere ilişkin konum mahremiyet profili aşağıdaki gibi tanımlanır [12].

$FT_s = \{Hastane, Ibadethane\}$
$T = \{(Hastane, 0.4), (Ibadethane, 0.1)\}$

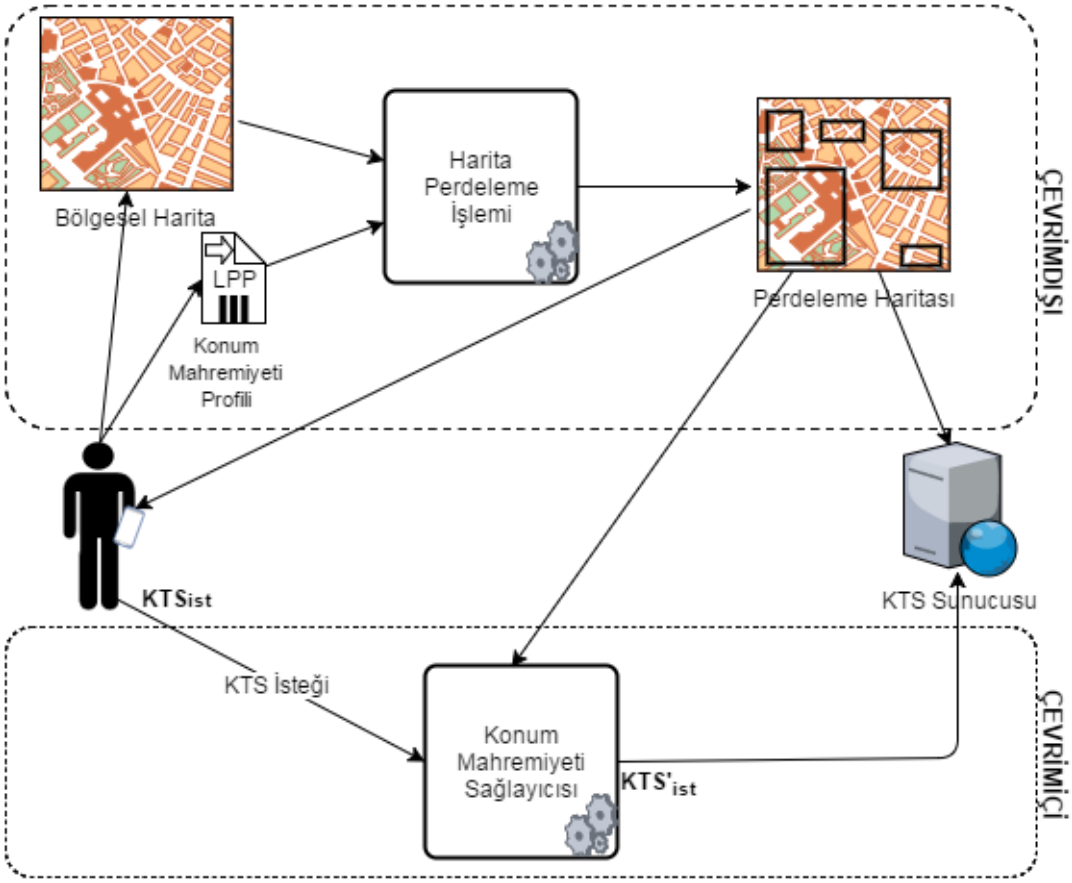
Bu örnekte, *Ibadethane* semantik konum türü için eşik değeri, *Hastane*'ye kıyasla daha düşük olup, mahremiyet ihtiyacının daha fazla olduğu anlamına gelmektedir.

3.3 Perdelenmiş Bölge

r, bir bölge olmak üzere, eğer r kullanıcı tarafından mahrem olarak nitelendirilen bir bölge ise ve T kümesindeki her bir tercih sağlanıyorsa, r'ye (FT_s, T) konum mahremiyet profili için bir perdelenmiş bölge denir ve Formül 3.1'de görüldüğü gibi formülize edilir. Buradaki P_{sens}, olasılık yoğunluk fonksiyonudur. Bu formül, mahrem olarak nitelendirilen semantik bir konumun alanının, içerisinde bulunduğu bölgedeki kapladığı alanın belli bir eşik değerinden küçük olmasını sağlamaktadır.

$$\forall (ft, \tau) \in T, P_{sens}(ft, r) \leq \tau \tag{3.1}$$

Özetle, perdelenmiş bölge, bir kısmında kullanıcı tarafından mahrem olarak nitelendirilen semantik konumları içeren ve kullanıcının mahremiyet şartlarını sağlayan bölgedir. PROBE çatısında perdelenmiş bölgeler üç farklı teknikten birisiyle oluşturulur. Bu teknikler, $Sens_{Reg}$, $Sens_{Div}$ ve $Sens_{Hil}$'dir [12].



Şekil 3.1: Konum mahremiyeti çatısı

Şekil 3.1, konum mahremiyet çatısını görselleştirmektedir. Kullanıcılar öncelikle bölgesel harita seçer ve konum mahremiyet profillerini belirlerler. Seçilen harita ve konum mahremiyet profili harita perdeleme işleminden geçirilerek perdelenme haritası oluşturulur ve kullanıcıya atanır. Kullanıcılar servis isteklerinde bulunurlarken bu perdelenmiş haritayı kullanırlar. Perdelenmiş bölge, zaman ve gerekli istek bilgilerini içeren servis isteği öncelikle konum mahremiyeti sağlayıcısına gönderilir. Konum mahremiyet sağlayıcısı, potansiyel saldırıları inceleyerek mahremiyet ihlali oluşan durumlarda servis isteğini, kullanıcının belirlediği değişkenlere göre değiştirerek konum-tabanlı servis sağlayıcısı sunucusuna iletir.

Şekil 3.2: 13 perdelenmiş bölge içeren bir perdeleme haritası

Şekil 3.2, 13 tane perdelenmiş bölge içeren bir harita kesitidir. Bu perdelenmiş haritayı kullanan bir kullanıcı, bir servis isteğinde bulunacağı zaman, perdelenmiş bölgelerden herhangi birisinde bulunmuyorsa nokta konumu gönderilmekte öte yandan, perdelenmiş bölgelerin birisinde bulunuyorsa bölgesel konumu gönderilmektedir. Örneğin, onkoloji hastanesinde bulunmasının bilinmesinden rahatsız olan bir kullanıcı R_8 bölgesi içerisinde bulunan bu hastanede servis isteğinde bulunduğu zaman nokta konum yerine perdelenmiş R_8 bölgesinin konumu gönderileceği için, kullanıcının bu bölge içerisinde bulunan, hastane, camii, gece kulübünden ya da herhangi bir evden hangisinde bulunduğu bilinememektedir. Ayrıca R_8 bölgesinde onkoloji hastanesinde servis isteğinde bulunan bu kullanıcı, çok kısa bir süre sonra R_7 bölgesindeki eczaneden servis isteğinde bulunduğunda, eğer bu eczane R_7 bölgesinin sınırında bulunuyorsa ve kullanıcının bu iki isteği arasındaki geçen süre, R_7 bölgesinin tamamının katedilebileceği bir süre değilse, R_7 bölgesi daraltılarak, kullanıcının bu servis isteğinin R_7 bölgesinin neresinden yapıldığı tahmin edilebilir. Bu yüzden PROBE [12] ve benzeri uygulamalarda, bu tarz hız saldırıları değerlendirilerek mevcut servis isteğine bir zaman gecikmesi uygulanır. Kullanıcının

belirlemiş olduğu en fazla zaman tahammül değişkeni değerine göre zaman gecikmesi uygulanamadığı durumlardaysa geriye dönük gönderme işlemi uygulanır.

3.4 Konum Örüntü Mahremiyeti

Konum mahremiyeti sağlanmış olsa da, konum örüntü mahremiyeti bu kişi için hala rahatsız edici olabilir. Şekil 3.2'teki kesikli çizgiyle gösterilen ve başlangıç noktası R_2 ve bitiş noktası R_5 olan rotanın şans eseri iki farklı kişi için de aynı şekilde oluştuğunu varsayalım. Bu kişilerden ikisi için de başlangıç noktataları evlerini temsil ediyor olsun. Bu kişiler başlangıç noktasından çıktıktan sonra R_8 bölgesinde bulunan hastanede bir süre geçirdikten sonra bitiş bölgesine devam etmişlerdir. Bitiş bölgesi de bu kişilerden ikisi için de iş yeri olarak düşünülebilir. Öte yandan, kişilerden birisi kanser hastası olmakla beraber, diğeri hamile bir kadındır. Bu örüntünün belli günlerde oluşuyor olması durumunda, kanser hastası olan kişi, çevresindekilerin hastalığını öğrenmesini istemiyor ve hastalığını saklıyorsa, belli günlerde evden çıktıktan sonra iş yerine gelmeden önce hastanede neden zaman geçirdiğini kimseye açıklamak zorunda kalmamak adına bu örüntüden rahatsız olabilir oysaki hamile kadının, zaten hamile olduğu dışarıdan da farkedilebildiği için saklayacak herhangi bir durumu yoktur.

Örnekle açıklanan örüntü mahremiyeti durumu, servis isteğinde bulunulan perdelenmiş bölgeler ve servis istekleri arasındaki geçen süreler açıklanarak daha rahat anlaşılabilir. Konum örüntü mahremiyetinin sağlanabilmesi için bir kullanıcı mahrem örüntülerini önceden belirlemelidir. Bir kullanıcı, R_5 bölgesinde servis isteğinde bulunduktan 10 birim süre sonra R_{11} bölgesinde, bundan 10 birim süre sonra R_7 bölgesinde ve bundan da 10 birim süre sonra R_8 bölgesinde servis isteğinde bulunmasını mahrem bir örüntü olarak tanımladığını düşünelim. Kullanıcılar mahrem örüntü tanımlamalarına mekan bilgisinin yanı sıra, zaman bilgisini de eklemektedirler. Zaman bilgisi düşünülmezse, kullanıcı mahrem örüntü tanımlamasında bulunan servis istek sıralamasına denk geldiği her an, mahremiyeti ihlal eden bir durum oluşur ancak bu servis istekleri arasında günler hatta aylar olabilir, dolayısıyla bu kadar uzun bir süre için mahremiyet ihlalini incelemek olası ve mantıklı bir durum değildir. Zaman bilgisi bu yüzden tanımlanmalıdır. Zaman bilgisinin yanı sıra, mahrem örüntülerin birebir oluşmasının zorluğu nedeniyle, kullanıcıların her bir mahrem örüntü için bir zaman esneklik değişkeni tanımlaması gerekmektedir. Bu kullanıcı için zaman esneklik değişkeni de 5 birim süre olarak tanımlansın. Yeni isteğin güvenli hale getirilebilmesi için izlenmesi gereken adımlarda, kullanıcının sınırlarını belirlemek adına yine kullanıcının belirleyeceği kabul edilebilir en fazla zaman gecikmesi 4 birim süre ve kabul edilebilir en fazla uzaklık tahammül mesafesi 200 birim olsun.

Kullanıcının istek geçmişi bilgisi boş olarak, Şekil 3.2'deki gibi rotayı takip ederek R_5 bölgesinden yola çıkan kullanıcı, bu bölgenin içerisinde en yakın eczane sorgusu çalıştırdığı zaman, kullanıcının tanımladığı mahrem örüntüyü kısmi destekliyor olacaktır ancak bu bir sorun teşkil etmemektedir. Devamında geçtiği her bölgede servis isteğinde bulunduğunu varsayarsak 5 birim süre sonra R_9 bölgesinde bulunduğu servis isteği herhangi bir etki etmemektedir. Bundan da 5 birim süre sonra R_{11} bölgesinde yaptığı servis isteği ile hala mahrem örüntüsünü kısmi

desteklemektedir. 8 birim süre sonra R_7 bölgesinde yapılan servis isteğinin ardından, mahrem örüntüsünün kısmi desteklenme durumu değişmemekle beraber, bundan da 12 birim süre sonra R_8 bölgesinde servis isteği yapıldığını varsayalım. Bu servis isteğiyle beraber artık kullanıcının mahrem örüntüsü desteklenmektedir çünkü R_7 bölgesindeki servis isteğinden sadece 12 birim süre sonra R_8 bölgesinde servis isteği yapılmıştır ve bu $10+5$ (kullanıcının tanımladığı zaman esneklik değişkeni) aralığının içerisindedir. Bu örüntünün oluşmasını engellemek için $10+5-12=3$ birim süre beklenmesi gerekmektedir ve bu değer de kullanıcının bekleyebileceği en fazla zaman tahammül değişkeni olan 4 birim süre değerinden küçük olduğu için bu servis isteğinde zaman gecikmesi kullanılabilir. Ancak bu değer 2 birim süre gibi bir değer olsaydı, bir önceki konumu gönderme yöntemi incelenmeliydi. Yani bu örüntünün oluşmaması için R_8 bölgesi yerine bir önceki bölge olan R_7 bölgesini göndermek düşünülebilirdi. Bunun içinse R_7 ve R_8 bölgeleri arasındaki uzaklık şartının kullanıcının belirlediği kabul edilebilir en fazla uzaklık tahammül değişkeni değeri 200 birimden küçük olması şartı sağlanmalıdır. Bu şart sağlanmıyorsa, daha da önceki bölge olan R_2 bölgesini göndermek düşünülebilir.

Bazı durumlarda daha gerideki bölgeler kullanıcının o anda bulunduğu bölgeye daha yakın olabilirler. R_7 ve R_2 bölgeleri arasındaki uzaklık 200 birimden küçük olması şartı da sağlanmıyorsa kullanıcının belirlemiş olduğu geriye dönük bakılacak bölge sayısı değeri kadar, kullanıcının istek geçmişinde o kadar bölge varsa, geriye yönelik devam eder. Geriye yönelik bölgelerden geri gitme sırasına göre herhangi birinin şartı sağlama durumunda önceki konum gönderme işlemiyle servis isteği gönderilir. Aksi halde istek reddedilir.

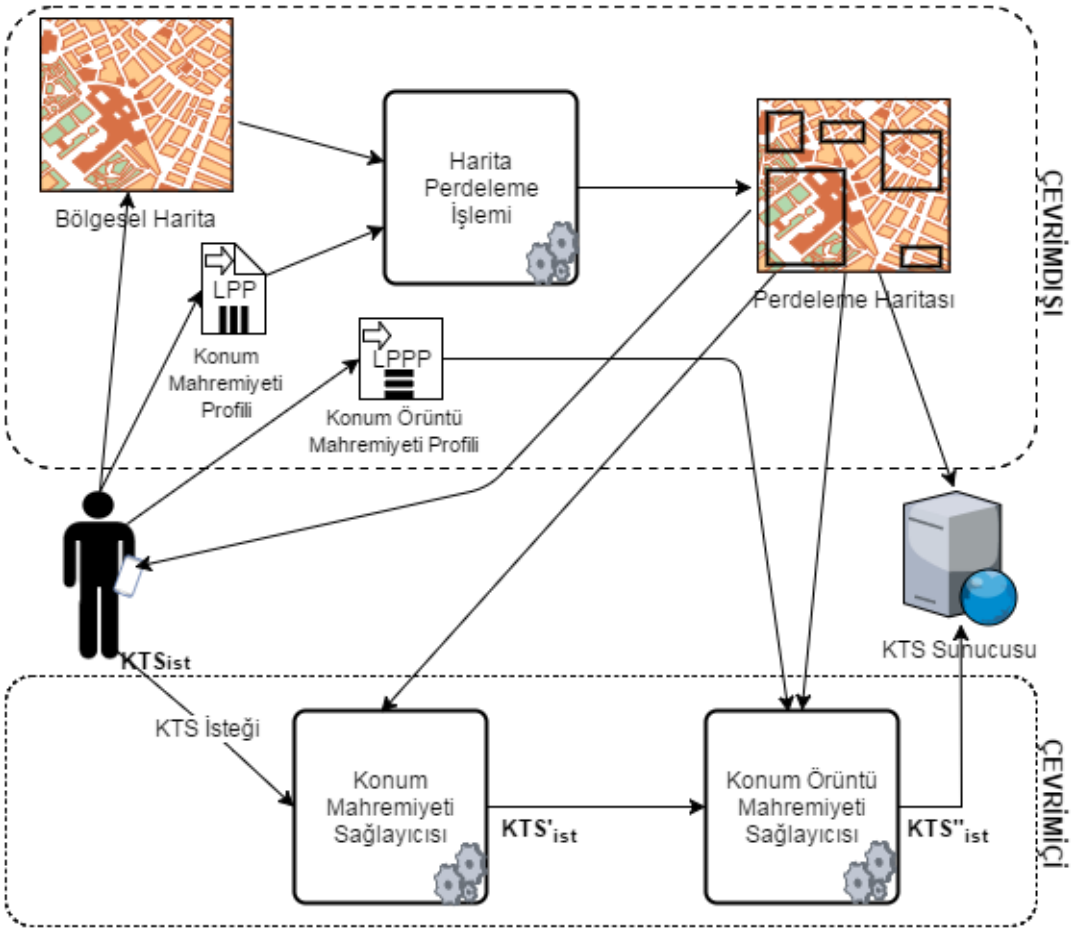
Örnek ile açık bir şekilde anlatılan bu durum tam olarak üzerinde çalışılan çevrimiçi örüntü mahremiyeti konusunu örneklemektedir. Bölüm 4'te, bu durum ayrıntılarıyla aktarılmış, çevrimiçi örüntü mahremiyetinin sağlanabilmesi için kullanıcı tarafından servis sağlayıcısına gönderilen isteğin, mahrem örüntü oluşturup oluşturmadığı kontrolünün nasıl yapılacağı detaylandırılmış ve sonrasında mahrem örüntü oluşturabilecek olan bir servis isteği karşısında nasıl davranılması gerektiği adreslenmiştir.

4. ÇEVRİMİÇİ KONUM ÖRÜNTÜ MAHREMİYETİ

Konum mahremiyeti Bölüm 3'te anlatıldığı gibi kişilerin mahremiyetlerini tam olarak sağlayamamaktadır. Bu yüzden, bu tez çalışmasında konum örüntü mahremiyeti tanımlanmaktadır. PROBE [12] veya benzeri konum perdeleme yaklaşımları tarafından üretilmiş olan bir perdeleme haritası R olarak tanımlanır. Böylelikle R içinde $|R|$ adet perdelenmiş konum belirteci vardır ve her bir istekte bu belirteçlerden birisi isteğin parçasını oluşturur. Her bir belirteç tipik olarak gerçek dünyada bir dikdörtgensel bölgeye karşılık geliyor gibi düşünebilir. Şehirsel bölgeler içinse benzer şekilde R içindeki her bir belirteç şehir yol ağının bir kesiti olabilir.

4.1 Konum ve Konum Örüntü Mahremiyeti Çatısı

Konum mahremiyeti çatısı, konum örüntü mahremiyeti de eklenerek Şekil 4.1'de görüldüğü gibi genişletilir.



Şekil 4.1: Konum ve konum örüntü mahremiyeti çatısı

Konum ve konum örüntü mahremiyeti çatısında kullanıcılar, konum mahremiyeti çatısındaki gibi, çevrimdışı olarak konum ve konum örüntü mahremiyetini sağlamak istedikleri bölgeyi ve konum mahremiyet profillerini seçerler, ayrıca konum örüntü mahremiyet profili oluşturarak mahrem örüntülerini tanımlarlar. Kullanıcıların konum mahremiyet profili ve seçtikleri harita bilgisiyle, her bir kullanıcıya perdelenmiş harita atanır ve bu harita çevrimiçi işlemler esnasında hem konum mahremiyetini hem de konum örüntü mahremiyetini sağlayabilmek için kullanılır. Çevrimiçi işlemler bölümündeki konum mahremiyeti sağlayıcısı, PROBE [12] ve benzeri şekillerde konum mahremiyeti sağlayarak kullanıcıdan aldığı isteği değiştirilmesi gerekiyorsa değiştirip, bu tez çalışmasında tanımlanan konum örüntü mahremiyeti sağlayıcısına göndermektedir. Konum örüntü mahremiyeti sağlayıcısı da, kullanıcının perdeleme haritası ve konum örüntü mahremiyet profilini göz önünde bulundurarak gerekli işlemleri yaparak, değişmesi gerekiyorsa değiştirilen servis isteğini, konum-tabanlı servis sunucusuna göndermektedir.

4.2 Problem Formülizasyonu

Tanım 1 (KTS İsteği KTS_{ist})

Bir KTS isteği $KTS_{ist} = (r,t,s)$ şeklinde bir üçlüdür. Burada $r \in R$ isteğin yapıldığı perdelenmiş bölge belirteci, t isteğin yapıldığı zaman bilgisi ve s ise istekle alakalı diğer uydu bilgileri içerir. Basitçe, s kullanıcı tanımlayıcısı, servis tanımlayıcısı ve istekle alakalı diğer parametreleri içerir.

Tanım 2 (KTS İstek Geçmişi KTS_{gec})

Bir kullanıcı aynı servis sağlayıcıdan farklı zamanlarda servis isteğinde bulunabilir. Bu istekler KTS istek geçmişini oluşturur. KTS_{gec} tanımı aşağıdaki gibidir:

$$KTS_{gec} =< (r_1,t_1,s_1),(r_2,t_2,s_2),\cdots,(r_n,t_n,s_n) > \tag{4.1}$$

Burada istek geçmişinde n adet istek bulunduğu anlaşılır ve $t_i < t_{i+1}$ ve $t_n < t_{simdi}$'dir. Servis sağlayıcılar, çevrimiçi bir süreç yönettiği ve KTS_{gec} bilgisini kullandığı için, bu bilgileri kullanıcıyla ilişkili olarak saklıyor olabilirler. Bu yüzden, bir kullanıcının konum mahremiyetine, KTS sağlayıcı üzerinden/tarafından t_{simdi} zamanında yapılabilecek potansiyel saldırıların irdelenmesi gerekir.

4.2.1 Konum mahremiyeti saldırısı

Kullanıcının KTS_{gec} bilgisine sahip olan ve saldırgan olarak düşüneceğimiz KTS sağlayıcı, bazı arka plan bilgilerinden yararlanarak kullanıcının bulunduğu bölgeyi daraltarak, tam konumu hakkında fikir sahibi olabilir. Maksimum hız gibi arka plan bilgileri kullanılarak yapılabilecek saldırılar ve önlemler Bölüm 2.3'te anlatıldığı gibi daha önce ayrıntılı olarak açıklanmıştır [12, 19, 46].

Bu tez çalışması kapsamında, isteklerin konum saldırısı mahremiyetine sebep olmayacak şekilde PROBE [12] veya benzer bir konum mahremiyeti sağlayıcısı denetiminden geçerek geldiği kabul edilmiş olup konum örüntü mahremiyeti problemi adreslenmiştir.

4.2.2 Konum örüntü mahremiyeti saldırısı

KTS_{gec} bilgisi içerisinde kullanıcının mahrem olarak düşünebileceği bazı örüntüler olabilir. Kullanıcı bu yüzden bu tip örüntülerin bu bilgi içerisinde oluşmasının önlenmesini talep edebilir. Bu problem konum örüntü mahremiyeti problemidir.

Tanım 3 (Mahrem Örüntü)

R için bir mahrem örüntü $P = (\bar{r}, \bar{t})$ ikilisidir. Burada $\bar{r}$ ziyaret edilen bölgeler sıralaması, $\bar{t}$ ise ardışık bölgeler arası geçen süredir ve bu iki ifade aşağıdaki şekilde tanımlanırlar.

$$\bar{r} = (r_0, r_1, \cdots, r_m), \forall 0 \leq i \leq m, r_i \in R \tag{4.2}$$

$$\bar{t} = (t_1, t_2, \cdots, t_m) \in R_+^m \tag{4.3}$$

Bir mahrem örüntü aslında zamansal ve mekânsal boyutları olan zaman açıklamalı mekan sıralısıdır [1]. Bir mahrem örüntü şu şekilde gösterilir:

$$P = (\bar{r}, \bar{t}) = r_0 \xrightarrow{t_1} r_1 \xrightarrow{t_2} \cdots \xrightarrow{t_m} r_m \tag{4.4}$$

Kullanıcı için mahrem olan tüm örüntüler ise mahrem örüntü kümesini $\mathbf{P} = \{P_1, P_2, \cdots, P_k\}$ oluşturur.

Tanım 4 (Destek)

Her ikisi de kullanıcı tarafından tanımlanan bir zaman eşik değeri τ ve mahrem örüntü $P = (\bar{r}, \bar{t}) = r_0 \xrightarrow{t_1} r_1 \xrightarrow{t_2} \cdots \xrightarrow{t_m} r_m$ için, $KTS_{gec} =< (r_1, t_1, s_1), (r_2, t_2, s_2), \cdots, (r_n, t_n, s_n) >$ sıralısı eğer $0 \leq i_0 < \cdots < i_m \leq n$ tamsayıları varsa ve aşağıdaki koşullar sağlanıyorsa bu örüntüyü destekler denir ve $KTS_{gec} \sqsupseteq_\tau P$ şeklinde gösterilir.

- *(mekansal eşleşme)* $\forall 0 \leq k \leq m \cdot \exists (r_{i_k}, t_{i_k}, s_{i_k}) \in KTS_{gec}, r_m = r_{i_k}$ *ve*
- *(zamansal eşleşme)* $\forall t_k \in \bar{t} \cdot |(t_{i_k} - t_{i_{k-1}}) - t_m| \leq \tau$

Eğer herhangi bir $P_i \in \mathbf{P}$ için $KTS_{gec} \sqsupseteq_\tau P_i$ sağlanıyorsa (destekleniyorsa) kullanıcının konum örüntü mahremiyeti ihlal ediliyor demektir. Aksi durumda kullanıcının KTS istek geçmişi KTS_{gec} güvenlidir, yani servis sağlayıcı tarafından bilinmesinde ve saklanmasında bir sakınca yoktur. Tanımdan anlaşılacağı üzere, τ değeri küçüldükçe zaman kısıtı daha sıkılaşır. Uç noktalarda düşünecek olursak, $\tau = 0$ için, örüntüde tanımlanan zaman boyutu hiç esnek değildir, mutlaka belirtilen zaman değeriyle eşleşme olması gerekmektedir. Öte yandan $\tau = \infty$ içinse, zaman boyutunun hiç bir önemi kalmamaktadır.

Konum örüntü mahremiyeti ile ilgili olarak iki problem tanımlanabilir: (i) *çevrimiçi*, devam eden KTS istek geçmişi üzerine yeni sorgular geldikçe ve (ii) *çevrimdışı*, KTS geçmişinin son hali üzerinden. Birincisinde ihlal tespit edildiğinde kullanıcı KTS isteğini servis sağlayıcıya gönderip göndermemek arasında bir seçim yapmak durumundadır. İkincisindeyse, geçmişte yaptığı özensiz sorgulardan ya da mahrem olduğunu düşünmediği ama sonradan mahremleşen örüntülerden dolayı geçmişinde mahrem örüntünün olup olmadığını kontrol etmek ve varsa bunların temizlenmesini

servis sağlayıcıdan talep etmek durumundadır. Çevrimiçi durumda bilgi henüz paylaşılmadığından kontrol kullanıcıda olup kullanıcı önceden önlem alırken, çevrimdışı durumda paylaşılmış olan mahrem bilginin servis sağlayıcı tarafından temizlenmesi söz konusudur.

Çevrimdışı konum örüntü mahremiyeti problemi Bölüm 2.3.2'de anlatıldığı gibi literatürde konum-zaman sıralıları bilgi gizleme/temizleme problemine indirgenebilir [1]. Çözüm olarak KTS_{gec} içinde yer alan bazı istekler mahrem örüntülerin hiçbirisini destelemeyecek şekilde silinir. Ne yazık ki bu çözüm çevrimiçi durumda uygulanamaz. Çevrimiçi durum için geliştirilen çözüm aşağıda anlatılmıştır.

Tüm servis istekleri çevrimiçinde konum örüntü mahremiyeti bakımından kontrol edilip güvenli ise gönderildiğinden KTS_{gec}'in güvenli olduğu kabulü yapılabilir. Yeni servis isteği geldiğinde bu devamlılığın sağlanması problemi aşağıda verilmiştir. Yani her bir yeni istek sonrasında istek geçmişi güvenli halde bırakılır.

Problem 1 (Çevrimiçi Konum Örüntü Mahremiyeti)

Verilen kullanıcı tanımlı mahrem örüntü kümesi $\mathbf{P} = \{(\overline{r_i}, \overline{t_i})\}$, zaman eşik değeri τ ve şimdiye kadar oluşan istek geçmişi $KTS_{gec} =< (r_1,t_1,s_1),(r_2,t_2,s_2),\cdots,(r_n,t_n,s_n) >$ olsun. Çevrimiçi örüntü mahremiyeti problemi, kullanıcı yeni bir istekte $KTS_{ist} = (r_{simdi=n+1}, t_{simdi=n+1}, s_{simdi=n+1})$ bulunduğunda aşağıda formülize edilen $KTS_{gec} \diamond KTS_{ist}$'nin herhangi bir $P \in \mathbf{P}$ örüntüsünü destekleyip desteklemediğinin belirlenmesi ve destekliyorsa desteklemeyecek hale getirilmesidir. $KTS_{gec} \diamond KTS_{ist} =< (r_1,t_1,s_1),(r_2,t_2,s_2),\cdots,(r_n,t_n,s_n),(r_{simdi},t_{simdi},s_{simdi}) >$ Eğer, yeni istek sonucunda $KTS_{gec} \diamond KTS_{ist}$ herhangi bir örüntü tarafından destekleniyorsa, yeni istek güvenli değil şeklinde tanımlanır. Yani yeni isteğin güvenli olması için $KTS_{gec} \not\sqsupseteq_{\tau} P, P \in \mathbf{P}$ şartının sağlanması gerekmektedir.

4.3 Yeni İsteğin Güvenlilik Kontrolü

Çevrimiçi konum örüntü mahremiyeti probleminde tanımlanan güvenlilik kavramı, her bir yeni istek geldiğinde kontrol edilmelidir. Bu işlem, anlık olarak yapılması gerektiği için hızlı olmalıdır. Bu probleme, hızlı bir çözüm geliştirebilmek için, dinamik programlama mantığıyla yaklaşılmıştır. Bu yaklaşımın kolay olabilmesi için $|\mathbf{P}| = 1$ şeklinde sabitlenmiş, yani $\mathbf{P}$ tek bir mahrem bölge içerecek şekilde, daha sonra bu genel bir durum olan $|\mathbf{P}| \geq 1$ olacak şekilde genişletilmiştir.

Tanım 5 (Kısmi destek)

Verilen bir mahrem örüntü P için i. öneki P^i, ve KTS istek geçmişi KTS_{gec}'in j.öneki KTS^j_{gec} ile gösterilir ve aşağıdaki şekilde tanımlanır;

$$P^i = (\overline{r}, \overline{t}) = r_0 \stackrel{t_1}{\rightarrow} r_1 \stackrel{t_3}{\rightarrow} \cdots \stackrel{t_{i-1}}{\rightarrow} r_{i-1} \tag{4.5}$$

$$KTS^j_{gec} =< (r_1,t_1,s_1),(r_2,t_2,s_2),\cdots,(r_j,t_j,s_j) > \tag{4.6}$$

$KTS^j_{gec} \sqsupseteq_{\tau} P^i$ sağlanıyorsa istek geçmişi örüntüyü kısmi destekler denir. $M(i,j), P^i$'yi destekleyen KTS^j_{gec}'nin tüm örneklerini göstersin. Yani; $M(i,j) = \{k : k \in 1 \cdots j$ ve

$KTS^j_{gec} \sqsupseteq_\tau P^i\}$ olarak tanımlansın. Ayrıca ikili değere sahip değişken $I(i,j) = KTS^j_{gec} \sqsupseteq_\tau P^i \wedge j \in M(i,j)$ tanımlayalım. Eğer $I(i,j) = true$ ise her ikisi arasında en sağda biten bir eşleşme vardır. $I(i,j)$ değeri bilindiğinde $I(i,j+1)$ değeri artımsal olarak dinamik programlama yöntemi ile kolayca hesaplanabilir. I tablosunu aşağıdaki kurallara uygun şekilde doldurabiliriz.

$$\begin{aligned}
&I(0,j) = false, \forall j \\
&I(i,0) = false, \forall i \\
&I(i,j<i) = false, \forall i \\
&I(1,j) = ayniMi(r_0, r_j) \forall j, r_0 \in P^{i'}, r_j \in KTS^j_{gec} \\
&I(i,j+1) = \vee_{k=1}^{k=j} [I(i-1,k) \wedge ayniMi(r_{i-1}, r_{j+1}) \wedge (|(t_{j+1,t_k}) - t_{i-1}| \leq \tau)], \forall j
\end{aligned} \tag{4.7}$$

Belirtmeliyiz ki, $I(i,j+1)$ formülünde, $ayniMi(x,y), x \in P^i$ ve $y \in KTS^j_{gec}$ ifadesi $x = y$ mekânsal eşleşmesini kontrol etmektedir, devamındaki $|(t_{j+1,t_k}) - t_{i-1}| \leq \tau, t_{j+1}, t_k \in KTS^j_{gec}$ ve $t_{i-1} \in P^i$ ifadesi ise $(r_{j+1}, t_{j+1}, s_{j+1})$ isteği için zamansal eşleşmeyi kontrol eder.

n, KTS_{gec}'in uzunluğu olarak ve m ise P'nin uzunluğu olarak tanımlansın. Böylelikle tüm I tablosunu doldurmanın maliyeti $O(mn^2)$ olacaktır. Öte yandan, artımsal hesaplama (her bir istek için) maliyeti $O(mn)$ olur ki bu dinamik programlama sayesinde kazanılır.

Örnek (Kısmi Destek): Şekil 3.2'de gösterilen rotada KTS istek sıralısı $KTS_{gec} =< (R_2,10,Q_1),(R_8,20,Q_2),(R_1 1,35,Q_1),(R_9,50,Q_3) >$ şeklinde tanımlansın ve şu anki servis isteği $KTS_ist = (R_5,60,Q_2)$ olsun. Kullanıcı, mahrem örüntüsünü $P = R_2 \xrightarrow{7} R_8 \xrightarrow{32} R_9 \xrightarrow{8} R_5$, $\tau = 4$ olarak tanımlasın. İlgili I tablosu Çizelge 4.1'de görülebileceği gibi, şu şekilde hesaplanır. Tablo hesaplaması dinamik olarak arttırılmalı olduğu için tabloya sütun eklenir ve mevcut KTS isteği KTS_ist'ten sonra hesaplanır. Satır 4 ve sütun 5'deki girdi *doğru (true)* olarak hesaplandığından, geçerli istek değiştirilmeden konum-tabanlı servis sağlayıcısıyla paylaşıldığı zaman, konum örüntü mahremiyeti ihlaline neden olur. Bir sonraki bölümde bunun nasıl çözüleceği gösterilmektedir.

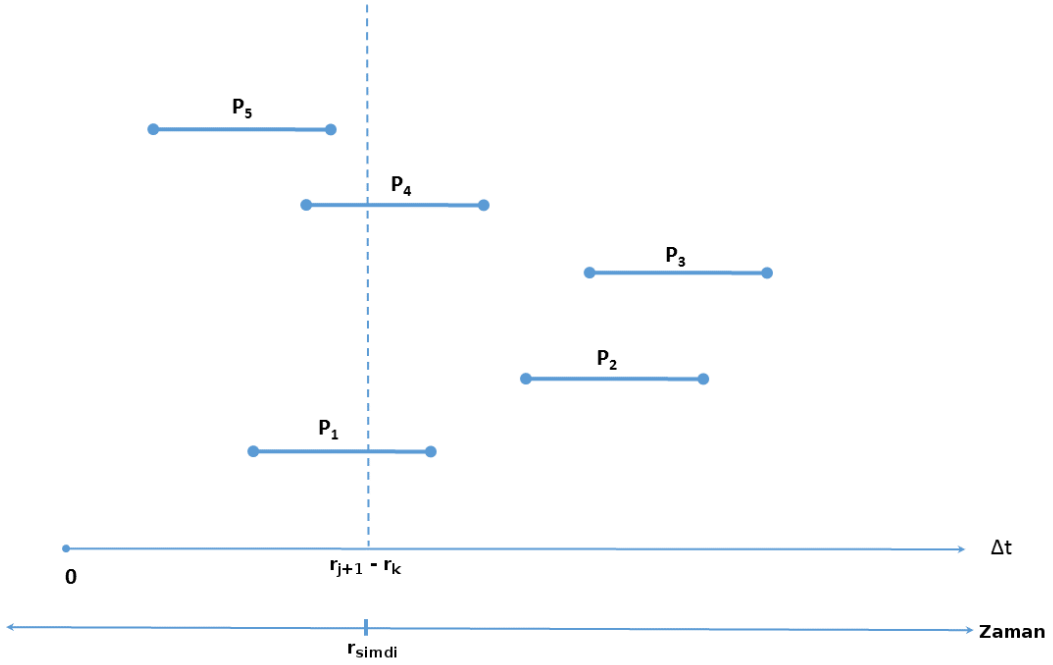
Çizelge 4.1: Kısmi destek için örnek I tablosu

I	*0*	$(R_2,10,Q_1)$ 1	$(R_8,20,Q_2)$ 2	$(R_{11},35,Q_1)$ 3	$(R_9,50,Q_3)$ 4	$(R_5,60,Q_2)$ **5**
0	False	False	False	False	False	False
1	False	True	False	False	False	False
2	False	False	True	False	False	False
3	False	False	False	False	True	False
4	False	False	False	False	False	True

Mahrem örüntü kümesinde $k > 1$ bölge varsa, Çizelge 4.1'de gösterilen I tablosu her örüntü için ayrı ayrı hesaplanır ve I_P ile gösterilir.

Zamansal eşleşmeyi görselleştirmek için Şekil 4.2'de gösterildiği gibi örüntülerin parça gösterimi oluşturulmuştur. Her örüntü için, geçen zamanın Δt ekseni üzerind,e

bitiş noktaları $t_{i-1} - \tau$ ve $t_{i-1} + \tau$ olan bir doğru parçası elde edilir ve burada geçen süre r_k'ya göre hesaplanır. Bu eksende, örüntü ile ilişkili doğru parçası $(r_{j+1} - r_k)$ çizgisi tarafından kesiliyorsa ilgili desen desteklenmektedir. Örneğin, Şekil 4.2'de, zaman kısıtı $|(r_{j+1} - r_k) - t_{i-1}| \leq \tau$ yalnızca P_1 ve P_4 örüntüleri için $true$ ve diğer örüntüler için $false$ olmaktadır.



Şekil 4.2: Örüntülerin parça temsili ve zamansal eşleşmenin grafiksel olarak gösterilmesi

4.3.1 Zaman karmaşıklığı ve iyileştirme

$n = |KTS_{gec}|$, $k = |\mathbf{P}|$ ve $m = \max\{|P| : P \in \mathbf{P}\}$ olmak üzere I_P tablosunu doldurma işleminin hesaplama karmaşıklığı $O(mn^2)$ olmaktadır. Bunun nedeni, I_P tablosu $O(mn)$ girdi içermekte ve her girdi $O(n)$ zamanda hesaplanmaktadır. Toplamda k adet tablo bulunduğundan, tüm tabloları doldurmanın karmaşıklığı, n'de ikinci dereceden olan $O(kmn^2)$'dir.

Tablonun aşamalı olarak hesaplandığı unutulmamalıdır, yani yeni KTS isteği ile $(r_{j+1}, r_{j+1}, s_{j+1})$, yalnızca tabloların $j+1$ sütunu hesaplanmaktadır. Açıkça, hesaplama $O(kmn)$ zaman alır, yani doldurulan girdilerin sayısı $O(km)$'dir ve toplamda her bir girdi $O(n)$ süre alır. KTS_{gec} sürekli akan veri olduğu için, boyut olarak çok uzayabilir. Dolayısıyla, hesaplama karmaşıklığı ve herhangi bir I tablosunun boyutu pratikte çok büyük olabilir. Bununla birlikte, aşağıdaki zamansal

sınırlar teoreminde açıklandığı gibi, zamansal sınırlayıcı özelliği dikkate alınarak (hem zaman hem de mekan) tablo boyutu sınırlandırılabilir.

Teorem 1 (Zamansal Sınırlar): $I(i, j+1)$ formülündeki $|(t_{j+1}, t_k) - t_{i-1}| \leq \tau$ eşitsizliğini sağlamayan en büyük $k = l$ ise, $k < l$ değerlerini kontrol etmemize gerek yoktur.

İspat: Açıkça, tüm $k < l$ değerleri için $|(t_{j+1}, t_k) - t_{i-1}| \leq \tau$ ifadesi isteklerin zaman ekseninde sıralı olması nedeniyle sağlanmayacaktır.

Dahası t_{i-1} ve τ değerleri her bir iterasyon için sabit olduğundan dolayı $k = l$ indisi kolaylıkla güncellenebilir. Yani, son zaman aralığı $t_{i-1} + \tau$ uzunluğunda tutulur. Bu yaklaşım, son zaman aralığının maksimum üst değerinin t_{simdi}'den geriye doğru olduğunu varsayarak, ciddi miktarda zaman kazandırır ve hesaplama maliyetini hemen hemen $O(m)$'e indirir. m değeri aslında süreç başladığında sabittir ve n gibi artmadığından her bir adımda yapılan işlem sabittir denilebilir. $I(m, simdi) = true$ olarak hesaplanırsa yeni istek P örüntüsü için güvenli değildir. Benzer şekilde I matrisi P içindeki tüm mahrem örüntüler için hesaplanır ve en az birisi için $true$ değeri bulunursa $(r_{simdi}, t_{simdi}, s_{simdi})$ isteği KTS sunucusuna gönderilmemelidir. Çünkü en az bir mahrem örüntü istek geçmişi içinde vardır. Aksi durumda bu istek KTS sunucuya gönderilebilir.

Zamansal sınırlar teoremi, Formül 4.7'teki $\bigvee_{k=1}^{k=j}$ teriminin $\bigvee_{k=l}^{k=j}$ terimi ile değiştirilebilmesini sağlamaktadır. Ayrıca zamanın monotonluğundan dolayı, yeni $KTS_ist = (r_{j+2}, t_{j+2}, s_{j+2})$ ile birlikte, k için yeni alt sınır l'den daha az olamaz. Teorem, bu gözlem ile birlikte, $I(i, k < l)$ tüm tablo girdilerinin silinmesini sağlar. Böylece, boyutu $t_{i-1} + \tau$ olan bir kayan pencere tanımlamasıyla (sabit bir örüntü parçası için sabit bir değer $r_{i-1} \xrightarrow{t_{i-1}} r_i$) yeni KTS isteği ile $k \geq l$ değerleri için tablo etkili bir şekilde güncellenebilir. Aynı mantık, diğer tüm tablo girdileri için de geçerlidir. Sonuç olarak, I tablosunun her satırının küçük sütunları benzer şekilde silinebilir. Böylece, tablonun sütun sayısı sınırlandırılmış olur, yani n değerine bağımlı değildir. Maksimum zamansal pencere teoremi, bu gerçeğin bir diğer alternatif kanıtıdır.

Teorem 2 (Maksimum Zamansal Pencere): Zaman esneklik parametresi τ ve mahrem örüntü $P = r_0 \xrightarrow{t_1} r_1 \xrightarrow{t_2} \ldots \xrightarrow{t_m} r_m$ olmak üzere, KTS istek sıralısı $KTS_{gec} =< (r_1, r_1, s_1), (r_2, r_2, s_2), \ldots, (r_n, r_n, s_n) >$ için, herhangi bir $KTS_{gec} \sqsupseteq_\tau P$ için en büyük zamansal pencere büyüklüğü, $\sum_{i=1}^{i=m} t_i + m\tau$'dan büyük olamaz.

İspat: $r_0 \xrightarrow{t_1} r_1$ üzerinde herhangi bir zamansal eşleşme, $t_1 + \tau$'dan büyük olamaz. Aynı şekilde, $r_{i-1} \xrightarrow{t_i} r_i$ parçası üzerinde herhangi bir eşleşme $t_i + \tau$'dan büyük olamaz. Sonuç olarak, bunların toplamı herhangi bir pencerenin maksimum zamanla eşleşmesinin $\sum_{i=1}^{i=m} t_i + m\tau$'dan büyük olmadığını gösterir.

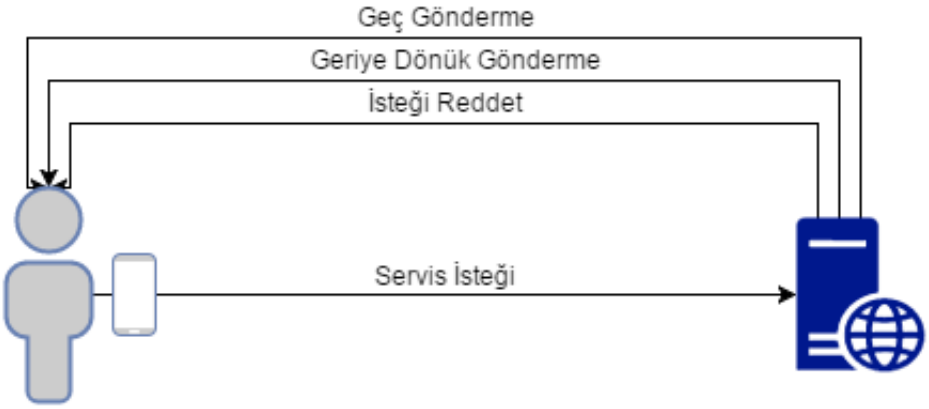
Teorem 1 veya Teorem 2'den dolayı, tek bir I tablosunu güncelleme zaman karmaşıklığı, $O(mn)$'den $O(m)$'e düşürür. Ayrıca, desen KTS istekleri sıralısında değişmediğinden, bir örüntünün uzunluğu m'nin sabit olduğunu kabul edebiliriz. Sonuç olarak, tek bir I tablosunu etkili bir şekilde güncellemek, $O(1)$ zaman

karmaşıklığına sahip olur. $|\mathbf{P}| = k$ tablo için, karmaşıklık $k \times O$ yani k sabit olduğu için, diğer bir deyişle, her kişinin sınırlı sayıda mahrem örüntüsü olduğu için $O(1)$ olur.

Özetle, Teorem 1, tüm k tablolarının her KTS servis talebinde $O(1)$ süre içerisinde güncellenebileceğini ispatlar. Ayrıca, mekân karmaşıklığı da $O(1)$'dir.

4.4 Yeni İsteğin Güvenli Hale Getirilmesi

$(r_{simdi}, t_{simdi}, s_{simdi})$ isteğinin güvenli olmadığı anlaşıldığında çözüm olarak Şekil 4.3'de görülebileceği gibi sırasıyla (i) *zaman gecikmesi*, (ii) *önceki konumu gönderme*, (iii) *servis isteğini reddetme* adımları denenmektedir. En başta kullanıcıdan kabul edilebilir maksimum zaman gecikmesi (T) ve kabul edilebilir maksimum konum hatası mesafesi (D) alınır.



Şekil 4.3: Kullanıcı servis isteği ve muhtemel cevapları

Eğer servis isteği en erken $t_{simdi} + t$ zamanda güvenli hale gelecekse ve $t \leq T$ ise servis isteği t zaman ötelenir ve istek $(r_{simdi}, t_{simdi} + t, s_{simdi})$olarak gönderilir. Aksi durumda (yani $t > T$) ise servis isteği sanki önceki konumların birisindeymiş gibi değerlendirilir ve istek $(r_{n-k}, t_{simdi}, s_{simdi})$ biçiminde düşünülür. Eğer $Uzaklik(r_n - k, r_{simdi}) \leq D$ ve $(r_n, t_{simdi}, s_{simdi})$ güvenli ise servis isteği $(r_{n-k}, t_{simdi}, s_{simdi})$ile yapılır. k değeri kullanıcının belirlediği, rotada ne kadar geriye gidebileceğine izin veren değerdir. 1'den k'ya kadar sırayla geriye giderken, uzaklık şartı sağlanılan ilk bölge, herhangi bir örüntüye de takılmıyorsa gönderilebilir. Burada ana fikir önceki konumların birisiyle şimdiki konum yakınsa ve önceki konum güvenli ise istek sanki önceki konumdan yapılmış gibi göstermektir. Bu iki seçenekten hiç birisi, konum örüntü mahremiyeti ihlal problemini çözemiyorsa istek reddedilir.

Zaman gecikmesi durumunda önemli olan bir husus bir sonraki isteğin $t_{simdi} + t$ süresi dolmadan yapılma durumudur. Bu durumda yeni yapılan isteğin zamanı $t_{simdi} + t$ olarak güncellenir ve istek geciktirilir.

Algoritma 1, çevrimiçi mahrem örüntü ihlali sorununun çözümünü özetlemektedir. Mevcut KTS isteği göz önüne alındığında, I tablolarını ve KTS geçmişini günceller.

KTS sağlayıcısına gönderilmek üzere muhtemelen zaman gecikmeli veya önceki konumlardan birisine dönüştürülmüş şekilde KTS isteğini değiştirir. Aynı zamanda, Şekil 4.1'de tanımlanan konum mahremiyet sağlayıcısına bu kararı sorar ve bildirir. Herhangi bir çözüm bulunmaması durumunda, isteği düşürür. $SpatialDistance(\cdot,\cdot)$ fonksiyonu, verilen iki bölgenin merkezleri arasındaki mesafeyi hesaplamaktadır.

Örnek (Kısmi Destek 2): Herhangi bir kullanıcının KTS istek geçmişi $KTS_{gec} =< (R_2,10,Q_1),(R_8,20,Q_2),(R_1 1,35,Q_1),(R_9,50,Q_3) >$ ve yeni servis isteği $KTS_{ist} = (R_5,60,Q_2)$ olsun. Aynı kullanıcının mahrem örüntülerinin bir tanesinin $P = R_2 \xrightarrow{7} R_8 \xrightarrow{32} R_9 \xrightarrow{8} R_5$ ve $\tau = 4$ olduğu varsayılmak üzere bir önceki kısmi destek örneğinde detaylı şekilde anlatıldığı gibi konum örüntü mahremiyeti

Algoritma 1 Çevrimiçi Mahrem Örüntü Eşleme Algoritması

Input: Table I, $\mathbf{P}$, KTS_{gec}, $KTS_{ist} = (r_{simdi}, t_{simdi}, s_{simdi})$, τ, T, D, k
Output: KTS'_{gec}

$KTS'_{gec} \leftarrow KTS_{gec} \diamond KTS_{ist}$
Compute last column for Table I_P using Equation 4.7, $\forall P \in \mathbf{P}$
if $KTS'_{gec} \not\sqsupseteq_\tau P.\ \forall P \in \mathbf{P}$ **then**
    Update Table I_P. $\forall P \in \mathbf{P}$ using Equation 4.7
    return KTS_{ist}
$t \leftarrow$ Minimum time delay so that KTS'_{gec} is pattern safe w.r.t. $(\mathbf{P}, \tau)$
if $t \leq T$ **then**
    // Time delaying
    $KTS'_{ist} \leftarrow (r_{simdi}, t_{simdi+t}, s_{simdi})$
    $KTS'_{gec} \leftarrow KTS_{gec} \diamond KTS'_{ist}$
    Update Table I_P using Equation 4.7. $\forall P \in \mathbf{P}$
    TellKMS(KTS'_{ist})
    return KTS'_{ist}
for $i \leftarrow 1$ *to* k **do**
    $kr_i \leftarrow i$'th region while doing regression on KTS_{gec}
    if $SpatialDistance(kr_i, r_{simdi}) \leq D$ **then**
        // Postdating
        $KTS'_{ist} \leftarrow (kr_i, t_{simdi}, s_{simdi})$
        if AskKMS(KTS'_{ist}) = $False$ **then**
            continue
        $KTS'_{gec} \leftarrow KTS_{gec} \diamond KTS'_{ist}$
        Compute last column for Table I_P using Equation 4.7. $\forall P \in \mathbf{P}$
        if $KTS'_{gec} \not\sqsupseteq_\tau P.\ \forall P \in \mathbf{P}$ **then**
            Update Table I_P. $\forall P \in \mathbf{P}$ using Equation 4.7
            TellKMS(KTS'_{ist})
            return KTS'_{ist}
// Drop the request
$KTS'_{gec} \leftarrow KTS_{gec}$
TellKMS(*null*)
return *null*

ihlali mevcuttur. $T = 5$ olması durumunda, mevcut istek $KTS_{ist} = (R_5, 60, Q_2)$, $t = 3$ dakika geciktirilebilir ve $KTS'_{ist} = (R_5, 60+3, Q_2)$ şeklinde servis sağlayıcısına gönderilebilir. Bu durumda I tablosu, Çizelge 4.2'de görüldüğü gibi olur.

Çizelge 4.2: Kısmi destek zaman gecikmesi işlemi için örnek I tablosu

I	*0*	$(R_2, 10, Q_1)$ 1	$(R_8, 20, Q_2)$ 2	$(R_{11}, 35, Q_1)$ 3	$(R_9, 50, Q_3)$ 4	$(R_5, 63, Q_2)$ **5**
0	False	False	False	False	False	False
1	False	True	False	False	False	False
2	False	False	True	False	False	False
3	False	False	False	False	True	False
4	False	False	False	False	False	False

Aksi durumda, $T = 1$ ise, bu isteğe zaman gecikmesi uygulanamaz. Bu durumda, önceki konum gönderme işlemi denenmesi gerekmektedir. KTS_{gec} üzerinde geriye doğru giderken ilk görülen bölge olan R_9 bölgesi için eğer $Uzaklik(R_9, R_5) \leq D$ ise, mevcut istek $KTS'_{ist} = (R_9, 60, Q_2)$ şeklinde güncellenir. Bu durumda I tablosu, Çizelge 4.3'te görüldüğü gibi olur.

4.4.1 KMS ve KÖMS arasındaki arayüz

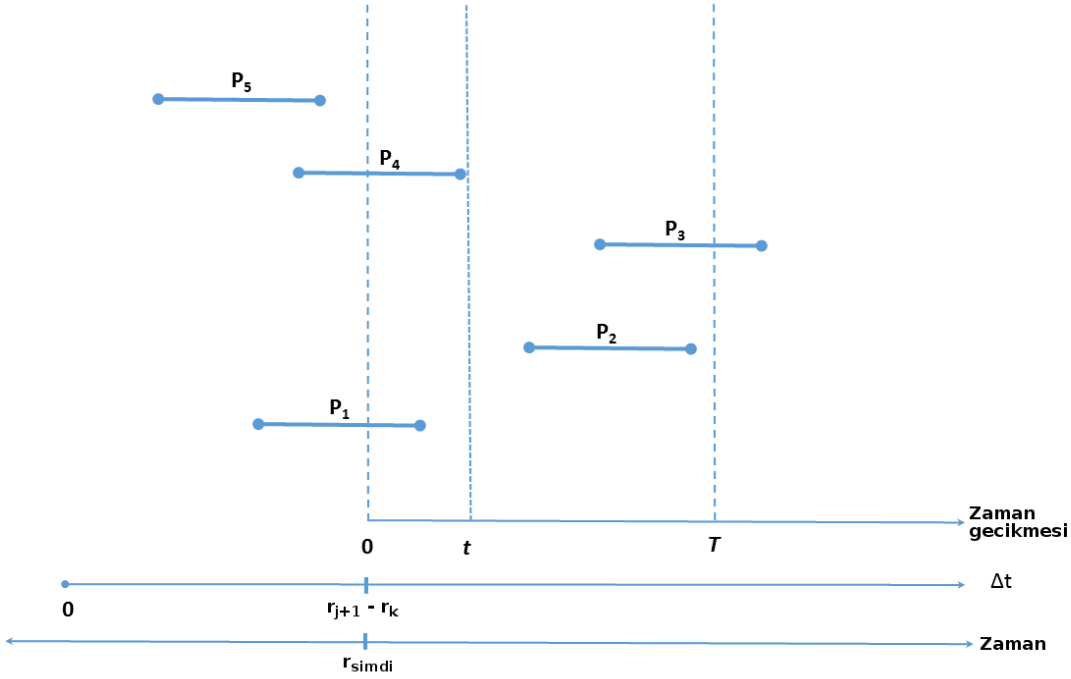
Konum mahremiyet sağlayıcısından (KMS), konum örüntü mahremiyeti sağlayıcısına (KÖMS), Şekil 4.1'de gösterildiği gibi, veri akışı arayüzüyle beraber bir de kontrol akış arayüzüne ihtiyaç duyulmaktadır (Algoritma 1'deki TellKMS ve AskKMS fonksiyonları). TellKMS, KÖMS tarafından KMS'ye, KTS sağlayıcısı ile paylaşılan bilgiler hakkında kendi kayıtlarını güncelleyebilmesi için bilgi verilmesini sağlamaktadır. Böylece KMS ve KÖMS arasındaki tutarlılık sağlanır. KÖMS tarafından uygulanacak olan herhangi bir değişikliğin KMS tarafında mahremiyet ihlaline neden olup olmadığı AskKMS fonksiyonu ile kontrol edilir. İstek yapılacak olan bölge ve daha önce istek yapılan bölge aralarında geçen süre içerisinde erişilebilir durumdaysa konum mahremiyet ihlalinin önceki konum gönderme işlemi ile giderilmesi mümkündür. Başka bir deyişle KMS'nin, en azından, maksimum hız saldırısını geçersiz kılması gerekir. AskKMS'nin cevabı *True* ise bu durumda isteğin gönderileceği mevcut konum güvenlidir, aksi takdirde daha da geride kalan bölge denenir. Zaman gecikmesi işlemi, maksimum hız saldırısı göz önünde bulundurulduğunda her zaman daha güvenli olacağı için, AskKMS'ye zaman geciktirme işlemi uygulanırken başvurulmaz.

4.4.2 Minimum zaman gecikmesinin bulunması

Algoritma 1'deki satır 6, KTS'_{gec}'i güvenli yapabilecek olan minimum zaman gecikmesi t'yi $\mathbf{P}$ ve τ'ya göre bulur. Bu adım, $|(t_{j+1} + t - t_k) - t_{i-1}| \leq \tau$ formülünü tüm $P \in \mathbf{P}$ örüntüleri için kontrol eder. Amaç, en küçük $t \in [0..T]$ değerini bulmaktır, böylece t'nin dikey çizgisi Şekil 4.4'teki herhangi bir örüntü parçasıyla kesişmez.

Çizelge 4.3: Kısmi destek önceki konum gönderme işlemi için örnek I tablosu

I	0	$(R_2,10,Q_1)$ 1	$(R_8,20,Q_2)$ 2	$(R_{11},35,Q_1)$ 3	$(R_9,50,Q_3)$ 4	$(R_9,60,Q_2)$ **5**
0	False	False	False	False	False	False
1	False	True	False	False	False	False
2	False	False	True	False	False	False
3	False	False	False	False	True	False
4	False	False	False	False	False	False



Şekil 4.4: Minimum zaman gecikmesinin bulunması

Teorem 3 (Zamansal eşleşmenin düzensizliği): Zamansal eşleşme, zaman gecikmesi değeri olan t göz önünde bulundurulduğunda monoton değildir.

İspat: Görsel bir şekilde açıklanabilmesi adına Şekil 4.4'teki P_2 örüntüsü incelenebilir. Mevcut t değerinde herhangi bir zamansal eşleşme bulunmamaktadır ancak t değerinin artış göstermesi durumunda P_2 örüntüsüyle bir zamansal eşleşme meydana gelebilir. Öte yandan bu değer, T değerine yaklaştıkça herhangi bir zamansal eşleşme meydana gelmemektedir. Teorem 3, algoritma geliştirirken, minimum zaman gecikmesi olan t değerini bulmayı sağlar.

$m = |\mathbf{P}|$ olmak üzere, minimum zaman gecikmesi t, Algoritma 2'de gösterildiği gibi basit bir algoritma ile $O(m^2)$ zaman karmaşıklığında bulunabilir. Algoritma, bütün örüntülerin mahremiyet ihlalini kontrol eder ve bir tane bulduğunda, bu örüntüyü desteklemeyecek şekilde ilgili isteği bir süre $(t_{i-1}+\tau)$ geciktirilir , yani $t = t_{i-1}+\tau$ dikey çizgisi, örüntü parçasını sol tarafında/arkasında bırakır. İşlem daima en kötü durumda olduğu gibi, tüm örüntü parçalarını t dikey çizgisi dışarısında tutacak şekilde çalışır. t değerinin, T'değerini geçmesi gerekirse, algoritma $false$ cevabı ile erken

sonlandırılabilir. Her döngüde, yeni t değerinin solundaki en az bir örüntü parçasının ihlali giderildiği için, zaman karmaşıklığı $O(m^2)$ olur. m değerinin küçük bir sabit olması beklendiğinden, zaman karmaşıklığı kesinlikle büyük bir değer değildir.

Bir parçalı ağaç veri yapısı [11] kullanarak zaman karmaşıklığının kolayca $O(mlogm)$'e düşürülebilir. Ağaç, $O(mlogm)$ zamanında oluşturulabilir ve örüntüler parçalarının sağ uç noktaları azalan düzende $O(mlogm)$ zamanda sıralanabilir. Ardından, doğru parçası ağacı, her biri $O(logm)$ zamanı ile her uç nokta için en fazla $2m$ kez sorgulanabilir. Dolayısıyla, bu yaklaşımın toplam zaman karmaşıklığı $O(mlogm)$'dir.

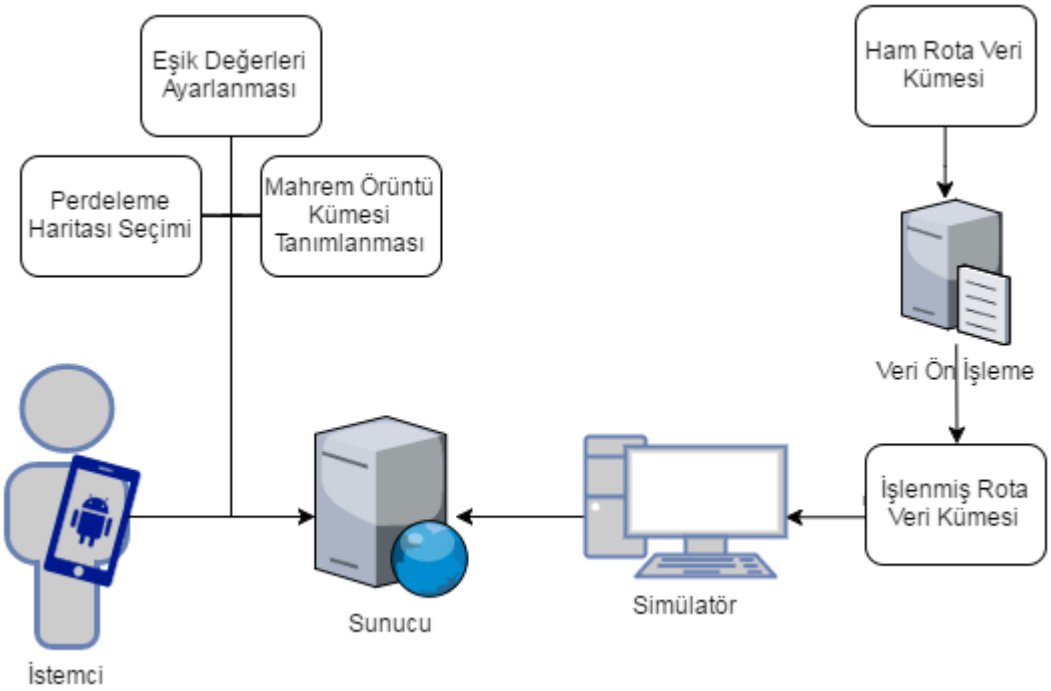
Algoritma 2 Minimum Zaman Gecikmesinin Bulunması

Input: **P**, KTS'_{gec}, τ
Output: t

```
1:  t ← 0
2:  while true do
3:      violation ← false
4:      for P ∈ P do
5:          if |(t_{j+1} + t − t_k) − t_{i−1}| ≤ τ then
6:              violation ← true
7:              t ← t_{i−1} + τ
8:      if violation = false then
9:          break
10: return t
```

5. UYGULAMA ve DENEYSEL SONUÇLAR

Bölüm 4'te anlatılan çevrimiçi konum örüntü mahremiyeti konusunun sağlanabilmesi için bir sunucu, bir istemci ve simülasyon yapısına ihtiyaç duyulmaktadır. Sunucu, konum-tabanlı servis sunucusu olarak, istemci, konum-tabanlı servis uygulamasını kullanan herhangi bir kullanıcı olarak ve simülatör ise test ortamı olarak düşünülebilir. Bu yapı Şekil 5.1'de gösterilmiştir. İstemci kullanıcısı için bir perdeleme haritası bulunmaktadır ve yine bu kullanıcı için mahrem örüntülerinden oluşan $\mathbf{P} = (\bar{r}, \bar{t}) = r_0 \xrightarrow{t_1} r_1 \xrightarrow{t_2} \cdots \xrightarrow{t_m} r_m$ kümesi tanımlanmaktadır. Kullanıcının mevcut durumda yaptığı bir KTS isteği, sunucuya gönderilmekte ve sunucu bu isteğin güvenilir olup olmadığının kontrolünü yaparak uygun cevabı istemciye iletmektedir. Sunucu, Bölüm 4'teki Şekil 4.3'de görüldüğü gibi, mevcut durumdaki istek güvenli değilse istek zaman gecikmesi, önceki konumlarından birini gönderme ya da isteğin reddedilmesi/düşürülmesi gibi durumları sırasıyla değerlendirerek sonucu istemci ile paylaşmaktadır.



Şekil 5.1: Uygulama bileşenleri

5.1 Konum Örüntü Mahremiyeti Sağlama Sunucusu

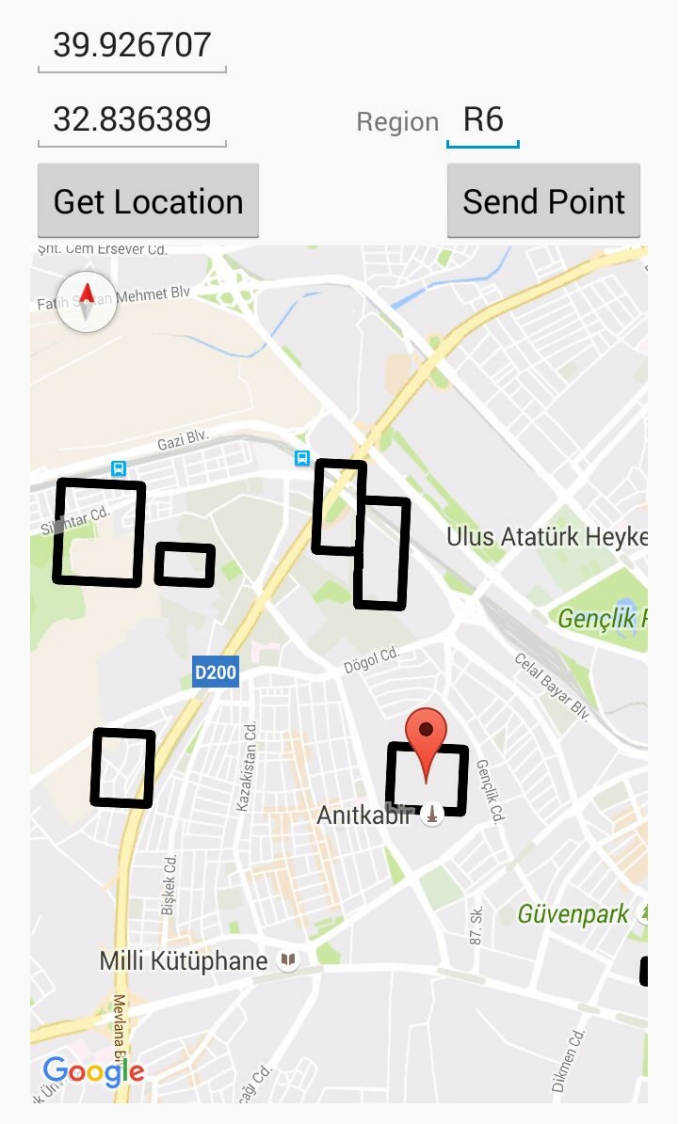
Uygulama sunucusu, istemciden konum istek bilgilerini ve kullanıcı tanımlı D ve T eşik değişkenlerini alarak, mevcut isteğin güvenli olup olmadığına ve güvenli değilse bu istek karşısında nasıl davranılacağına karar vermektedir. Sunucu SOAP (Simple Object Access Protocol) web servis yapısı kullanılarak Java programlama dili ile kodlanmış ve Oracle Glassfish Server üzerinde çalıştırılmaktadır.

Web servis sunucusu, kullanıcıya bir perdeleme haritası atama işlemi gerçekleştirir. Perdeleme haritaları çevrimdışı süreçte hazırlanır. Daha önce Bölüm 2.3.1'de anlatılan ve özellikle [12]'de üzerinde çalışılan tarzda bir perdeleme haritası oluşturulur. Konum mahremiyetiyle ilgili daha önce yapılan çalışmalar mevcut olduğu için ve bu tez çalışmasında daha çok konum örüntü mahremiyeti inceleneceği için önceki çalışmaların bire bir gerçeklemeleri yapılmamış buna karşın perdeleme haritası rastgele oluşturulmuştur. Simülasyon bölümünde kullanacağımız veri, İtalya'nın Milano şehrini içeren bir coğrafi bölgede yapılan konum isteklerini içerdiği için, sunucunun oluşturduğu perdelenmiş harita o bölgeyi içermektedir. Perdeleme haritası çevrimdışı süreçte hazır edildikten sonra, istemciye ulaştırılır.

Sunucu, ayrıca, kullanıcının mahrem örüntülerini de barındırmaktadır, istemci üzerinden kullanıcının bu mahrem örüntü kümesini değiştirmesine izin vermektedir. Kullanıcının konum örüntü mahremiyetini sağlamak demek, kullanıcının KTS geçmişinde, mahrem olarak tanımlanan hiçbir örüntünün bulunmamasını garanti altına almak demek olduğu için, sunucu kullanıcının KTS geçmişi bilgisine de sahip olmalıdır. KTS geçmişine sahip olan sunucu, kendisine istemci tarafından bir konum isteği gönderildiğinde, isteği yapan kullanıcının KTS geçmişini inceler ve mevcut mahrem örüntülerin oluşup oluşmadığının kontrolünü dinamik programlama ile yapar. Algoritma 1, Tanım 5'te anlatıldığı gibi $I(i, j)$ hesaplamasından sonra, buna karar verir.

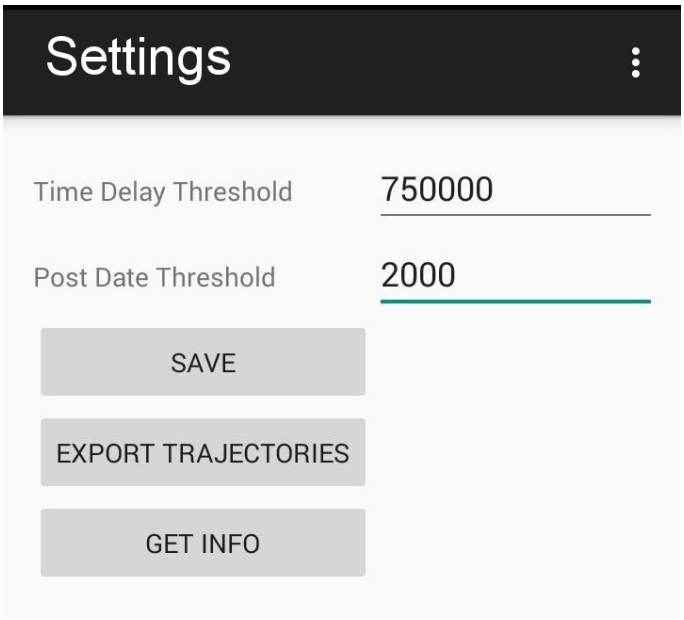
5.2 Konum Tabanlı Servis İstemcisi

İstemci olarak kullanılan, her kullanıcının kendi kişisel değişkenlerini ayarlayabildiği ve konum isteklerini sunucuya gönderebildiği bir mobil uygulama geliştirilmiştir. Bu istemci uygulaması Android işletim sistemli cihazlar üzerinde çalışmaktadır ve Android Studio ortamında geliştirilmiştir. Sunucudan alınan perdeleme haritasındaki perdeleme bölgelerini görsel olarak harita üzerinde göstermektedir. Şekil 5.2 mobil uygulamanın ana ekran görüntüsünü ve perdeleme haritasını göstermektedir. Mevcut harita Google Maps Android API ile sağlanmakta olup, kullanıcıya mevcut isteğini bulunduğu konumdan yollamasına izin vermekle beraber, haritadan seçerek başka bir konum göndermesine de olanak sağlamaktadır. Ayrıca kullanıcının mevcut perdeleme haritasına sonradan perdelenmiş bölge ekleme seçeneği de vardır. Kullanıcı perdelenmiş bölge oluşturmak istediğinde, perdelemek istediği bölgeyi içerisine alan bir dikdörtgenin köşe konum bilgilerini girerek yeni bir perdelenmiş bölge oluşturabilmektedir.



Şekil 5.2: Mobil uygulama perdeleme haritası ve istek gönderme ekranı

Mobil uygulamanın ayarlar bölümünde, kullanıcıya kişisel değişkenlerini değiştirme ve ayarlama imkanı sunulmaktadır. Kullanıcı bu ekrandan kişisel zaman tahammül eşik değeri (T) ve kişisel uzaklık tahammül eşik değerini (D) ayarlayabilmektedir. Bu değerlerden, T, istekte bulunan kullanıcının isteğinin güvenli olmadığının anlaşılması durumunda, zaman gecikmesi yapılıp yapılamayacağını kontrol ederken bir eşik değeri oluşturmaktadır. Bir başka deyişle, kullanıcı T değeri ile, bir isteğin geç gönderilmesinde en fazla ne kadar süre bekleyebileceğini belirler. Eğer istek T'ye eşit ya da daha kısa bir süre bekleyerek güvenli hale getirilebiliyorsa, bu seçenek kullanıcıya iletilerek kullanıcı bilgilendirilir ve kullanıcının istek geçmişinde istek zamanı değeri güncellenir. D değeri ise bir isteğin güvenli hale getirilebilmesi için önceki konumu gönderme işlemi yapılması gerekiyorsa, kullanıcının kabul edebileceği en fazla mesafe bilgisini belirtmektedir. Daha açık bir deyişle, kullanıcının bir isteği güvenli hale getirilebilmek için, kullanıcının daha önce bulunduğu bir konumdan yapılmalıysa, kullanıcının mevcut konumuyla o konum arasındaki uzaklık ölçüsü D'ye eşit ya da daha kısa olmalıdır. Bu durumda önceki konumu gönderme işlemi uygulanabilir.



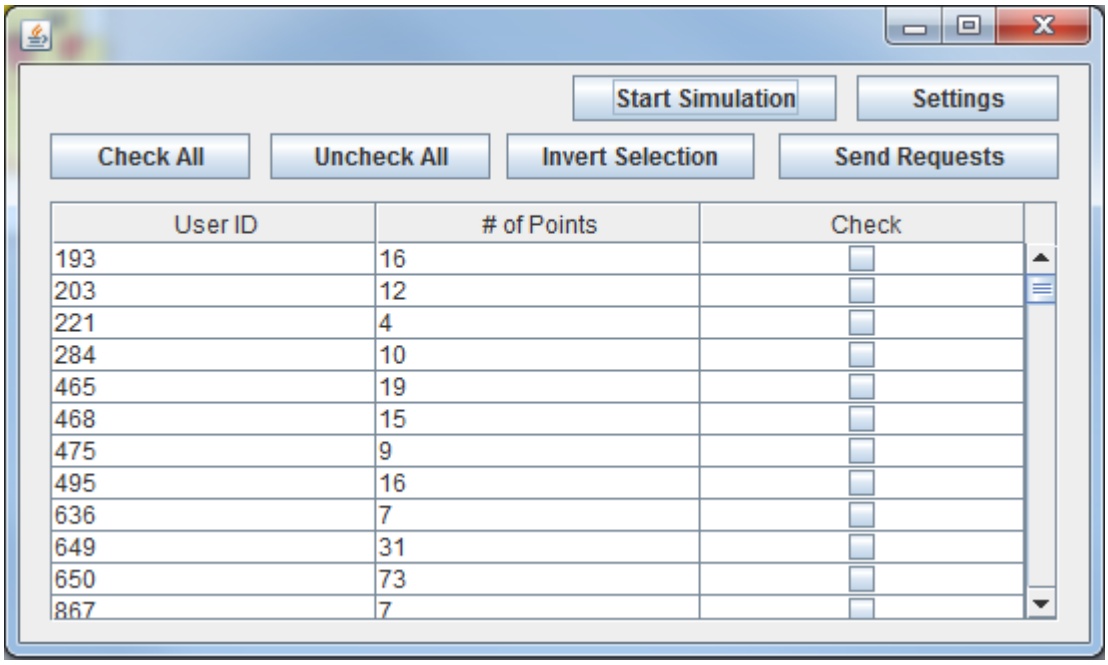
Şekil 5.3: Mobil uygulama ayarlar ekranı

Mobil uygulamanın ayarlar bölümünde kullanıcı, daha önceki istekleriyle ilgili istatistiksel bilgilere ulaşabildiği gibi, KTS geçmişi rota bilgisine de ulaşabilmektedir. Mobil uygulamanın ayarlar ekranına ait ekran görüntüsü Şekil 5.3'de görülebilmektedir. Burada Time Delay Threshold olarak gösterilen değişken, kullanıcı tanımlı T tahammül edilebilen en fazla zaman gecikmesi değişkeniyken, Post Date Threshold ise D tahammül edilebilen en fazla mesafe değişkenidir. τ değişkeni kullanıcının mahrem örüntüsünü tanımlarken her örüntü için ayrı ayrı belirlediği bir değişken olduğu için bu ekranda değiştirilememektedir.

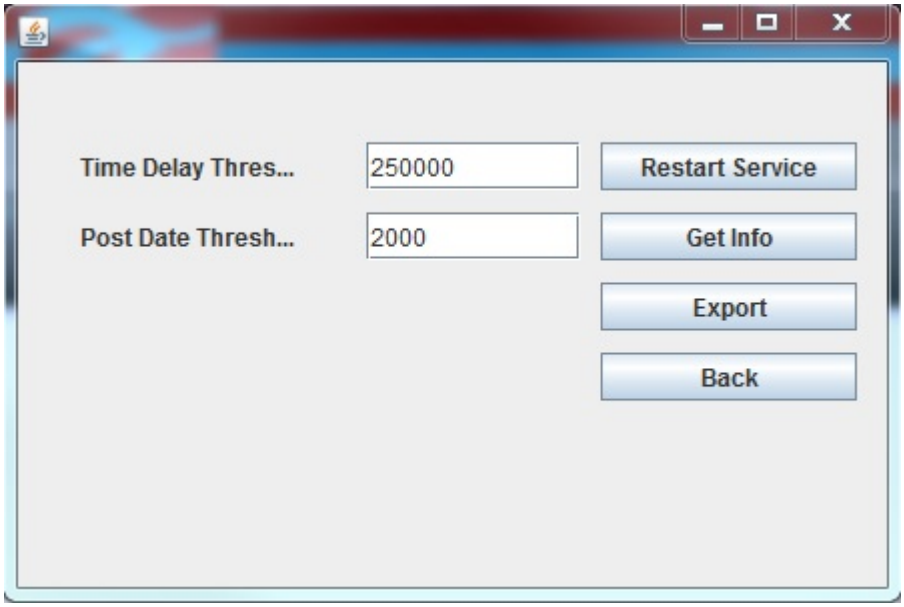
Mobil uygulama ayrıca kullanıcıya mahrem örüntü kümesini görüntüleme ve değiştirme imkânı da vermektedir. Perdeleme haritasındaki perdelenmiş bölgelere hakim olan kullanıcı bu bölgelerin tanımlayıcı numaralarıyla beraber kendisine göre zaman aralıkları vererek, mevcut mahrem örüntü kümesine ekleme yapabilmektedir. Kullanıcının yaptığı değişiklikler sunucuya iletilmekte ve bir sonraki servis isteğinde değerlendirilmek üzere kullanıcının mahrem örüntü kümesine eklenmektedir.

5.3 Simülatör

Mobil uygulama istemcisinin yanı sıra, büyük ölçekli test ve sonuç değerlendirme amaçlı bir simülatör yapılmıştır. Simülatör uygulaması Java programlama dili kullanılarak Eclipse IDE ortamında geliştirilmiştir. Simülatörde, Milano GPS verisi işlenmiş ve sanki bir kullanıcı tek tek servis isteğinde bulunuyormuşcasına, bölge ve zaman bilgileriyle beraber sunucuya istekler gönderilmektedir. Simülatöre ait ana ekran görüntüsü Şekil 5.4'de görülmektedir. Milano GPS veri kümesindeki farklı kullanıcıların, farklı rotaları bulunmaktadır ve test etmek istenilen rotalar seçilerek sunucuya gönderme işlemi başlatılabilir. Bu işlem, rotanın her bir noktasını tek tek göndermektedir. Mahrem örüntü yakalanması durumunda zaman gecikmesi, bir önceki konumu gönderme ya da isteğin düşürülmesi gibi işlemler yapılacağı için, sunucu bünyesindeki rota bilgisi, simülatördeki rota bilgisinden farklı olabilir.



Şekil 5.4: Simülator ana ekranı



Şekil 5.5: Simülator ayarlar ekranı

Ayrıca simülatörde, mobil istemcide kullanıcının zaman ve uzaklık tahammül eşik değerlerini değiştirebildiği gibi, bu değerler değiştirilebilmektedir. Bu, çalışma sonucunda, değerler değiştirilerek ve rota verisi farklı değerlerle simüle edilerek sonuçların daha verimli değerlendirilmesine yaramaktadır. Simülatördeki bu ekran Şekil 5.5'te görülmektedir. Buradaki hız değeriyse, verinin temizlenmesi aşamasında kullanılan bir değişkendir ve Bölüm 5.4'te anlatılacaktır. Ayrıca bu ekranda bulunan web servisi baştan başlatma, istatistiksel bilgilere erişme ve tüm kullanıcılar için sunucu büyesinde barındırılan KTS geçmişi bilgisinin dışa aktarılması gibi seçenekler test ve değerlendirme aşamasının daha verimli olmasını sağlamaktadır.

5.4 Veri Elde Edilmesi ve İşlenmesi

Simülatör uygulamasında kullanılmak üzere elde edilen iki farklı veri kümesi mevcuttur. Bunlar, GeoPKDD [17] projesinden elde edilen Milano GPS veri kümesi ve Stanford Network Analysis Project (SNAP)'ten [29] elde edilen Gowalla veri kümesidir [9]. Bu veri kümeleri çeşitli işlemlerden geçirilerek kullanılmışlardır.

5.4.1 Milano veri kümesi

Oluşturulan yapıda değerlendirilmek üzere elde edilen veri, İtalya'nın Milano şehrinde toplanmış/oluşturulmuş Milano GPS veri kümesi kullanılmıştır. Bu veri kümesinde toplamda 15.800 rota bulunmaktadır. Rotaların hepsinde toplamda 2.075.216 servis isteğinde bulunulmuştur. Milano GPS verisinin her bir satırında, TAB karakteriyle ayrılmış, sırasıyla, rota numarası, servis istek zamanı, boylam bilgisi ve enlem bilgisi bulunmaktadır. Bu veri kümesinin zaman bilgisi 01/01/2007 tarihinden itibaren saniye cinsinden olup, enlem ve boylam bilgileriyse 10^6 değeriyle çarpılmış durumdadır.

Milano GPS veri kümesi olduğu gibi kullanılamamaktadır çünkü mevcut yapı enlem ve boylam bilgilerinden ziyade perdelenmiş bölge numaralarıyla çalışmaktadır. Öncelikle veri işlenmesi adımında, her bir servis isteğinin yapıldığı enlem ve boylam bilgisi, perdeleme haritasında bulunduğu bölgeyle eşlenmelidir.

Mevcut veri kümesi Milano bölgesine ait olduğu için, Milano bölgesini kapsayan bir perdeleme haritası oluşturulmuş, 45.56 - 45.37 kuzey paralelleri ve 9.05, 9.28 doğu meridyenleri arasındaki bölge $2x10^{-4}$ birimlik eşit aralıklarla toplamda 1.090.401 birimlik bir ızgara oluşturulmuştur. Perdeleme haritası literatürde bir çok çalışmada kullanıldığı ya da oluşturulduğu için bu problem kişiselleştirilmeden, rastgele olarak irili ufaklı toplamda 1.000 tane perdelenmiş bölge oluşturulmuştur. Milano GPS verisindeki her bir servis isteğinin enlem ve boylam bilgileri bu 1.000 perdelenmiş bölge ile eşlenmeye çalışılmış, mahrem bölgelere denk gelen veriler saklanmış, herhangi bir mahrem bölgeye gelmeyen veriler, veri kümesinden çıkarılmıştır. Buradaki amaç, herhangi bir perdelenmiş bölgeye denk gelmeyen noktada yapılan bir servis isteğinin mahrem olma ihtimali yoktur ve herhangi bir işlem gerektirmemektedir. Eğer o nokta da mahrem olmuş olsaydı, o noktayı içerecek bir perdelenmiş bölge olurdu. Bu temizleme işleminden sonra, perdelenmiş bölgelerle eşleşmeyi başarabilen toplamda 580.692 servis isteği kalmaktadır. Ayrıca 15.800 rotanın her biri farklı kullanıcı olarak düşünülmüş ve her bir kullanıcı için mahrem örüntü kümesi oluşturulmuştur. Mahrem örüntü kümesi, kullanıcının rota bilgisini dikkate alarak rastgele bir şekilde oluşturulmuş olup her bir kullanıcıya 10 tane mahrem örüntü tanımlanmıştır. Mahrem örüntüleri rastgele şekilde oluştururken, kullanıcının rotası üzerinde bulunan bir noktanın mahrem örüntüye dahil edilip edilmemesine rastgele bir şekilde karar verilmiş ve aynı şekilde rotada iki istek arasındaki süre, mahrem örüntüye aktarılırken belli bir komşuluk değeriyle aktarılmıştır ve bu değere de rastgele karar verilmiştir.

5.4.2 Gowalla veri kümesi

Gowalla veri kümesi, arkadaşlık ve yer bildirimleri veri kümesidir. Veri kümesiyle aynı isimli bir sosyal ağ uygulamasındaki arkadaşlık ve yer bildirimleri verilerinin Şubat 2009 - Ekim 2010 tarihleri arasında toplanıp düzenlemiş halidir. Bu tez çalışmasında konum istekleriyle çalışıldığı için arkadaşlık bilgileri kullanılmamış sadece her bir kişinin yer bildirimleri, konum servis isteğiymişcesine düşünülmüştür. Veri setinin içerisinde toplamda 196.561 kişi ve 6.442.890 yer bildirimi mevcuttur. Tüm bu yer bildirimleri dünya genelinden toplanmış olduğu için bu tez çalışmasında kullanılmaya uygun değildir. Bu yüzden yine İtalya'nın Milano şehrine denk gelen 8.250 kilometrekarelik bir alan rastgele seçilmiştir ve bu alanın dışında yapılan istekler veri kümesinden temizlenmiştir. Küçültme işlemi sonucunda 6.933 istek kalmıştır.

Geri kalan istekler incelenmiş, bir kişinin yaptığı iki istek arasında bazen birkaç ay gibi uzun süreler varolduğu görülmüştür. Konum ya da konum mahremiyeti çalışmalarında, mahrem olarak nitelendirilebilecek durumlar birbirlerine yakın isteklerdir. Bu yüzden birkaç ay hatta birkaç günlük aralıklar bile mahrem örüntüleri desteklemeyecek şekilde olacaktır. Bu nedenle bu veri setindeki her bir kullanıcının istek sıralıları için, ilk istekle son istek arasında en fazla bir gün süre olacak şekilde düzenlenmiş yani, her bir kullanıcının günlük istekleri farklı kullanıcıların istekleriymişcesine ayrılmıştır. Öte yandan, bazı kullanıcılar bazı günlerde sadece bir ya da iki istekte bulundukları için, farklı kullanıcıymış gibi davranılan bu günler için rotalardaki istek sayısı az kalmaktadır. Bu sorunu çözebilmek adına, rotasında beşten az istek bulunan kullanıcılar da veri kümesinden çıkarılmıştır.

Son aşamada veri kümesinde kalan isteklerin bulunduğu alan 45.999 - 45.014 kuzey paralelleriyle 9.010 - 9.696 doğu meridyenleri arasında kalan alandır. Bu alan, çevrimdışı hazırlık aşamasında 100 x 100 olmak üzere toplamda 10.000 alanlık bir ızgaraya bölünmüş ve bu alanda boyutlar 10 ile 20 ızgara hücresinden oluşan 500 farklı perdelenmiş bölge oluşturulmuştur. Perdelenmiş bölgeler hiçbir şekilde birbirleriyle çakışmamaktadır. Oluşturulan alandan sonra, veri setindeki her bir isteğin konum bilgileri, içlerinde bulundukları bölge bilgileriyle değiştirilmiştir. Bazı istekler herhangi bir bölgeye denk gelmemektedir. Bir bölgeye denk gelmeyen isteklerin konum mahremiyetini tehdit etmeyeceği düşünüldüğünden üzerlerinde hiçbir işlem yapmayı gerektirmemektedir. Bu yüzden herhangi bir bölgeye denk gelmeyen istekler de veri kümesinden çıkarılmıştır. Sonuç olarak veri kümesinde her birinin sadece bir adet rotası olan 102 kullanıcı ve toplamda 1101 adet servis isteği kalmıştır. Ayrıca her bir kullanıcı için mahrem örüntüler, Bölüm 5.4.1'de anlatılan Milano veri kümesindeki gibi hazırlanmış ancak her bir kullanıcıya 1 - 10 arasında rastgele sayıda mahrem örüntü tanımlanmıştır.

5.5 Deneysel Sonuçlar

Simülatör uygulaması kullanılarak deneysel sonuçlara varabilmek adına Bölüm 5.4'te anlatılan iki farklı veri kümesi de ayrı ayrı kullanılmıştır. Milano veri kümesi, konum-tabanlı servis isteklerine yeterince uygun olmadığı için, τ ve T değişkenlerinin değeri dakika cinsinden çok yüksek ve gerçekçi olmayan değerlerdir.

Bunun sebebi, bu veri kümesinde bulunan sorguların bazılarının arasında bir kaç ay gibi ciddi zaman farklılıkları olmasından dolayıdır. Bu yüzden Milano veri kümesinden elde ettiğimiz sonuçlar, geniş bir bakış açısıyla bakabilmek adına yorumlanmış sonuçlardır. Öte yandan Gowalla veri kümesinden elde edilen sonuçlarsa, konum-tabanlı servis isteklerine daha uygun ve daha gerçekçi değişken değerleriyle daha iyi sonuçlar vermektedir.

5.5.1 Milano verisi sonuçları

Simülatör uygulaması çalıştırılmadan önce rastgele 200 kullanıcı seçilmiş, bu 200 kullanıcı ve rotalarında bulunan toplamda 4269 istekle çalıştırılmıştır. Kullanıcı değişlenleri olan τ, T ve D değerleri değiştirilerek, farklı değerler için farklı sonuçlar elde edildiği teoride olduğu gibi pratikte de gözlemlenmiştir. Yine bir kullanıcı değişkeni olan, önceki konumu gönderme durumunda ne kadar geriye gidileceğini belirten k değişkeni varsayılan olarak 5 kabul edilmiştir. τ, T ve D değişkenlerinin her birini on beşer kere değiştirerek toplamda kırk beş farklı değerle sonuçlar incelenmiştir. Simülatörün her bir çalıştırılma esnasında değişkenlerin aldığı değerler çizelgelerde gösterilmiştir.

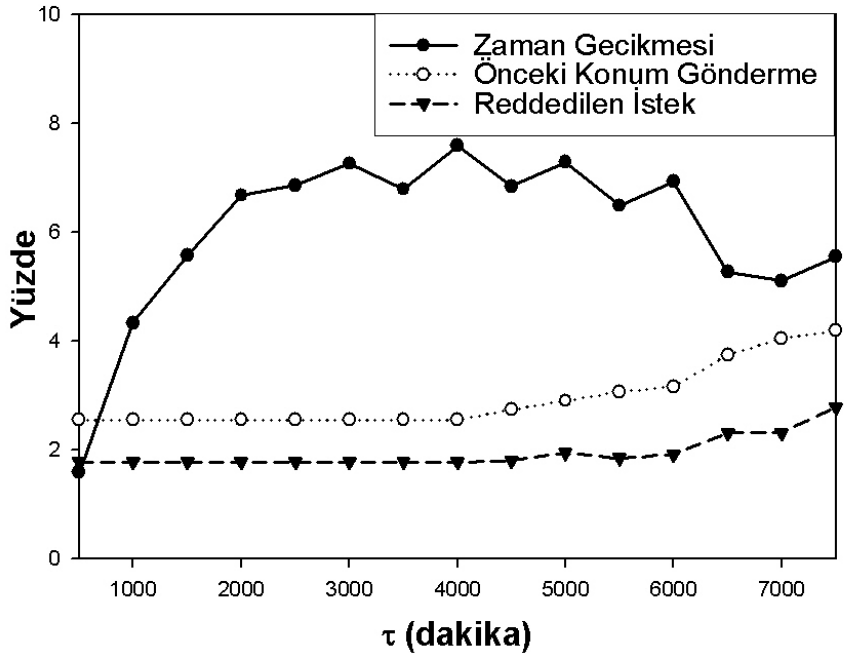
Simülatör çalışmalarının ilk on beşinde τ değeri değişkendir. Tanım 4'te anlatıldığı gibi τ değerinin değişmesi, kullanıcının mahrem örüntü kümesi içerisindeki örüntülerin desteklenme hassasiyetini belirlemektedir. τ değeri arttıkça, desteklenen örüntü sayısının artması beklenmektedir. τ değeri değiştirilerek alınan sonuçlar Çizelge 5.1'de görülebilmektedir. Bu sonuçlarda, beklendiği üzere τ değerinin artmasıyla beraber desteklenen örüntülerin sayısında bir artış meydana gelmiştir. Çizelge 5.1'de verilen değerlerin yüzdelik değişimi Şekil 5.6'da verilmiştir. Bu grafik aracılığıyla artan τ değerine göre herhangi bir işlemden geçirilen toplam isteklerin artışı görülmektedir.

Devamında yapılan on beş çalışmada, T değeri değişkendir. Bu çalışma sonucunda elde edilen değerler, Çizelge 5.2'de verilmiştir. İkinci on beş çalışmada, τ değeri sabit olduğu, sadece T değeri artış gösterdiği için, desteklenen örüntü sayısında ciddi bir değişim olması beklenmemektedir, sadece T değerine bağlı olarak, bir isteğe uygulanacak işlem değişmektedir. T değerinin artışına bağlı olarak, zaman gecikmesi uygulanan isteklerin sayısında bir artış meydana geldiği sonuçlardan anlaşılabilir. Şekil 5.7'deki değerlerin yüzdelik değişimi de Çizelge 5.2'deki sonuçlara aittir. Burada kullanıcı tanımlı zaman tahammül değişkeni T'nin artışına bağlı olarak uygulanan zaman gecikmesi işlemindeki artış farkedilir düzeydedir.

Aynı şekilde son on beş çalışma sonucunda elde edilen değerler, Çizelge 5.3'te verilmiştir. Son on beş çalışmada da, τ değeri sabit olduğu, sadece D değeri artış gösterdiği için, desteklenen örüntü sayısında ikinci on beş çalışmadaki gibi bir değişim olması beklenmemektedir, sadece D değerine bağlı olarak, bir isteğe uygulanacak işlem değişmektedir. D değerinin artışına bağlı olarak, önceki konum gönderme işlemi uygulanan isteklerin sayısında bir artış meydana geldiği sonuçlardan anlaşılabilir ve bu sonuçların yüzdelik değişimi Şekil 5.8'den takip edilebilir.

Çizelge 5.1: Değişen τ değerlerine göre Milano verisi simülasyon çalışma sonuçları. ($T = 25.000$ ve $D = 2.000$ değerleri sabit)

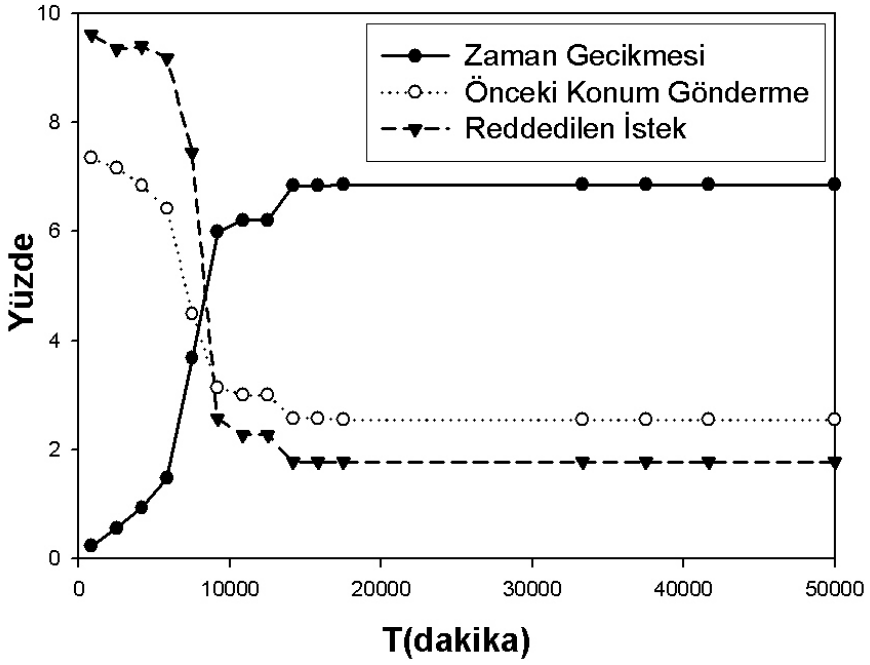
τ (Dakika)	Zaman Gecikmesi	Önceki Konum Gönderme	Reddedilen İstek	Değiştirilmeyen İstek
500	68	109	76	4.016
1.000	185	109	76	3.899
1.500	238	109	76	3.846
2.000	285	109	76	3.799
2.500	293	109	76	3.791
3.000	310	109	76	3.774
3.500	290	109	76	3.794
4.000	324	109	76	3.760
4.500	292	117	77	3.783
5.000	311	124	83	3.751
5.500	277	131	79	3.782
6.000	296	135	82	3.756
6.500	225	160	99	3.785
7.000	218	173	99	3.779
7.500	237	179	119	3.734



Şekil 5.6: Mahrem olarak belirtilen örüntülerin $T = 25.000$ ve $D = 2.000$ değerleri sabitken farklı τ değerleriyle birlikte verdiği yüzdesel oranlar

Çizelge 5.2: Değişen T değerlerine göre Milano verisi simülasyon çalışma sonuçları. ($\tau = 2.500$ ve $D = 2.000$ değerleri sabit)

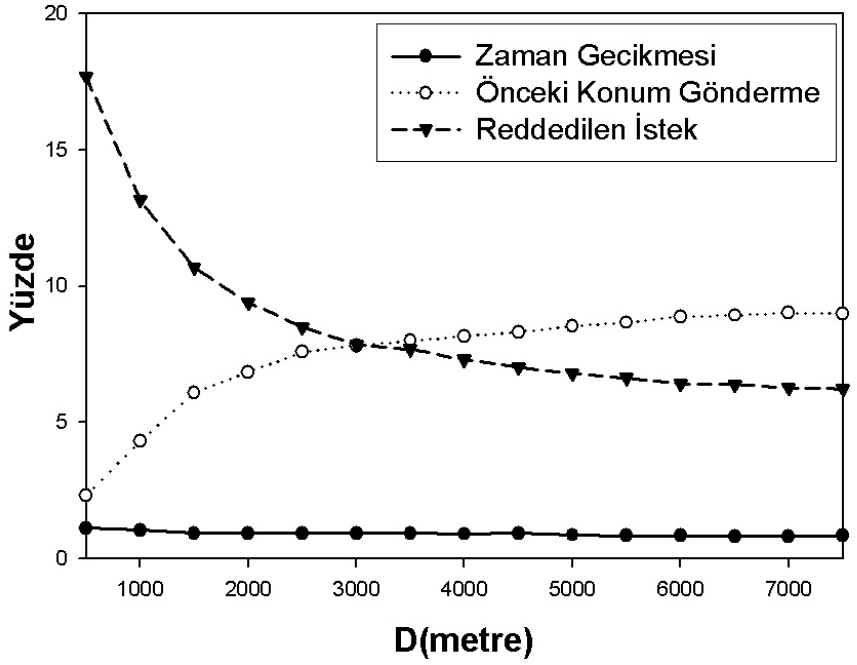
T (Dakika)	Zaman Gecikmesi	Önceki Konum Gönderme	Reddedilen İstek	Değiştirilmeyen İstek
833	10	314	410	3.535
2.500	24	306	399	3.540
4.166	40	292	401	3.536
5.833	63	274	392	3.540
7.500	157	192	318	3.602
9.166	256	134	110	3.769
10.833	265	128	97	3.779
12.500	265	128	97	3.779
14.166	292	110	76	3.791
15.833	292	110	76	3.791
17.500	293	109	76	3.791
33.333	293	109	76	3.791
37.500	293	109	76	3.791
41.666	293	109	76	3.791
50.000	293	109	76	3.791



Şekil 5.7: Mahrem olarak belirtilen örüntülerin $\tau = 2.500$ ve $D = 2.000$ değerleri sabitken farklı T değerleriyle birlikte verdiği yüzdesel oranlar

Çizelge 5.3: Değişen D değerlerine göre Milano verisi simülasyon çalışma sonuçları. ($\tau = 2.500$ ve $T = 4.166$ değerleri sabit)

D (Metre)	Zaman Gecikmesi	Önceki Konum Gönderme	Reddedilen İstek	Değiştirilmeyen İstek
500	48	99	755	3.367
1.000	45	184	562	3.478
1.500	40	260	456	3.513
2.000	40	292	401	3.536
2.500	40	324	363	3.542
3.000	40	334	335	3.560
3.500	40	341	328	3.560
4.000	39	348	312	3.570
4.500	40	355	300	3.574
5.000	37	363	289	3.568
5.500	36	370	283	3.580
6.000	36	379	274	3.580
6.500	35	381	273	3.580
7.000	35	385	268	3.579
7.500	36	384	266	3.583



Şekil 5.8: Mahrem olarak belirtilen örüntülerin $\tau = 2.500$ ve $T = 4.166$ değerleri sabitken farklı D değerleriyle birlikte verdiği yüzdesel oranlar

Çizelge 5.1'de bulunan ilk on beş çalışmada, τ değerinin artmasına bağlı olarak mahrem olarak değerlendirilen örüntü sayısının arttığını görüyoruz. Bu artış "Değiştirilmeyen İstek" sütunundaki azalma aracılığıyla takip edilebilmektedir. Görülen bu artışın nedeni, Tanım 4'te anlatıldığı gibi, τ değeri 0'a yaklaştıkça zaman kısıtı sıkılaşacağı için mahrem örüntü destekleme şansı azalacak, sonsuza(∞) yaklaştıkça da esnekleşeceğinden dolayı, mahrem örüntüleri daha çok destekleyecektir. Desteklenen mahrem örüntü sayısının artmasıyla birlikte, uygulanan zaman gecikmesi, önceki konum gönderme ve isteği reddetme işlemlerinde de artış görülmektedir. Desteklenen her mahrem örüntüye bir işlem yapılacağından dolayı bu normal bir durumdur. Öte yandan τ değerinin artışına bağlı olarak artış sağlayan desteklenen toplam örüntü mahremiyeti sayısındaki dalgalanma, desteklenen örüntülere uygulanan işlemler sonrasında KTS istek sıralısının değişiyor olmasıdır.

Çizelge 5.2'de bulunan ikinci on beş çalışmada, değişen T değerine bağlı olarak, uygulanan zaman gecikmesi işlemindeki artış gözle görülebilir düzeydedir. Zaman gecikmesindeki artışa bağlı olarak önceki konumu gönderme işleminin sayısı azalmaktadır. Desteklenen toplam örüntü sayısındaki azalma, uygulanan zaman gecikmesinin yeni sorguların uygulanma zamanını da ötelemesinden kaynaklanan kullanıcının istek sıralısındaki zaman sapmasıdır. Yani, τ değeri sabit olduğu için desteklenen örüntü mahremiyeti sayısındaki artış, τ kaynaklı değil, uygulanan zaman gecikmesi işleminin kullanıcının KTS istek sıralısındaki zaman boyutunu değiştirmesinden kaynaklıdır.

Aynı şekilde, Çizelge 5.3'te bulunan son on beş çalışmada, değişen D değerine bağlı olarak, uygulanan önceki konumu gönderme işleminde artış söz konusudur. Buna bağlı olarak, reddedilen isteklerin sayısında azalma görülmektedir. Desteklenen toplam mahrem örüntü sayısındaki azalma, aynı şekilde, uygulanan önceki konum gönderme işleminin, yeni sorguların konum bilgilerini değiştirmesiyle beraber oluşabilecek olan yeni mahrem örüntülerdir.

Çizelge 5.2 ve Çizelge 5.3'deki sonuçlarda, sırasıyla sadece T değişmesine ve sadece D değişmesine rağmen aynı sırayla önceki konum gönderme ve zaman gecikmesi değerlerinde de değişmeler gözükmektedir. İlk başta, D değişkeninin değerinin değişmesine bağlı olarak zaman gecikmesi uygulanan isteklerin sayısında bir değişme beklenmemesi düşünülebilir ancak değişen D değeri sonucunda önceki konum gönderme işlemi uygulanan istek sayısı arttıkça, kullanıcının konum istek geçmişi değişmektedir buna bağlı olarak yeni desteklenen örüntüler ortaya çıkabilir, böylelikle D değişimine bağlı olarak küçük de olsa zaman gecikmesi uygulanan isteklerin sayısının değişmesi bu kapsamda değerlendirilir. Aynı şey T değeri için de geçerlidir.

5.5.2 Gowalla verisi sonuçları

Gowalla veri kümesiyle simülatör uygulaması çalıştırılırken Bölüm 5.4.2'de anlatılan veri işleme işlemlerinin ardından kalan 102 kullanıcı ve toplamda 1101 isteğin tamamı kullanılmıştır. Simülatör Bölüm 5.5.1'de anlatılan şekilde τ, T ve D değişkenleri yirmi kere değiştirilerek toplamda altmış kere çalıştırılmıştır. Yine aynı şekilde k değişkeni varsayılan olarak 5 kabul edilmiştir.

Farklı değişken değerleriyle ne gibi sonuçların alındığını irdeleyebilmek için, öncelikle T ve D değerleri sabit tutularak farklı τ değerleriyle çalıştırılmıştır. İlk aşamada $T = 70$ dakika ve $D = 5.000$ metre seçilmiştir. Bu seçimler sırasıyla zaman gecikmesi ve önceki konum gönderme işlemlerinin yapılmasını etkilediği için, toplam işlem uygulanan istek sayısına direkt etkileri bulunmamaktadır ancak τ'daki değişim, mahrem örüntü desteklenmesiyle doğrudan ilişkili olduğu için, τ değişkenini değiştirmekteki amaç toplam işlem yapılan isteklerin sayısının τ ile doğru orantılı olduğunu görebilmektir. Bu yüzden T ve D'nin değerlerinden ziyade, sabit oluyor olmaları önemlidir. τ, Tanım 4'te anlatıldığı üzere, mahrem örüntülerin desteklenebilmesi için tanımlanan zaman farkıdır. Mahrem örüntüler, bir günlük mekan zaman sıralılarından çıkarıldığı için, tanımlanan süre kısıtları küçüktür. Bu süre kısıtlarını çok fazla esnetmemek adına, τ da olabildiğince küçük değerler arasından seçilerek denemeler yapılmalıdır.

İlk yirmi çalışmada τ değerleri 5 ila 100 dakika arasında artacak sırayla seçilmiştir. Bu çalışmalar sonucunda elde edilen sonuçlar Çizelge 5.4'te görülebilmektedir. "Değiştirilmeyen İstek" sütunundaki τ'nun artışına bağlı olarak meydana gelen gözle görülür azalma, τ arttıkça işlemden geçirilen istek sayısı artacaktır öngörüsünü doğrular niteliktedir. τ arttıkça uygulanan zaman gecikmesi işleminin azaldığı gözlemlenmiştir. Özellikle $\tau = 35$ değeri ve sonrasında ciddi düşme söz konusudur. Bunun sebebi seçilen T değişkeninin değerinin 70 olmasıdır. τ, bir artı-eksi komşuluk belirlediği için, alt sınırdan desteklenen bir örüntünün, desteklenmeyecek şekilde zaman gecikmesi yapılabilmesi için bekletilmesi gereken en az süre $2 * \tau$ kadardır. Bu durumda $2 * \tau$ değeri T değerinden büyük olursa bu işleme zaman gecikmesi uygulanması imkansızdır. Dolayısıyla, τ değeri, $T/2$ değerini geçtikten sonra uygulanan zaman gecikmesi sayısındaki azalma bu şekilde açıklabilir. Öte yandan τ değeri arttıkça meydana gelen önceki konum gönderme işlemindeki artışsa, zaman gecikmesi işlemi uygulanamayan istekler arttıkça bu isteklere önceki konum gönderme işleminin yapılabiliyor olmasından kaynaklanmaktadır.

Reddedilen isteklerdeki ciddi artışsa, T ve D değerlerinin yeterince büyük olmamasından kaynaklanmaktadır. Çizelde 5.4'ten elde edilen sonuçların yüzdesel dağılımı Şekil 5.9'da grafiksel olarak gösterilmiştir. τ değerine bağlı olarak değişen yapılan işlem ya da değiştirilmeyen istek sayısı bu grafikten de takip edilebilmektedir.

Değişen τ değerlerine göre verilen tepkinin beklendiği şekilde olduğu görüldükten sonra diğer çalıştırılmalarda öncelikle T değerinin değişimine göre sonuçların nasıl değiştiği incelenmiştir. Yirmi farklı çalışmada, τ ve D değerleri sabit tutularak kullanıcı tanımlı zaman tahammül değişkeni olan T değiştirilmiştir. Değişen T değerleri 15 dakika ile 240 dakika arasında artan sıradadır ve bu esnada $\tau = 30$ dakika ve $D = 5.000$ metre olmak üzere sabitlenmişlerdir. Amaç sadece değişen T değerine göre alınan sonuçları incelemek olduğu için τ ve D değişkenlerinin, değerlerinden ziyade sabit olmaları daha önemlidir öte yandan önceki çalışmalarda desteklenen toplam mahrem örüntü sayısında tatmin edici bir sonuç veren $\tau = 30$ dakika değeri seçilmiştir.

T değerinin, τ değişkeninin önceki çalışmalarda aldığı değerlere göre (5 - 100 dakika arası) daha büyük olmasının nedeni (15 - 240 dakika) T'nin zaman gecikmesine karar veren değişken olmasıdır. Yani τ, mahrem örüntülere bir komşuluk yaklaşımı

Çizelge 5.4: Değişen τ değerlerine göre Gowalla verisi simülasyon çalışma sonuçları. ($T = 70$ ve $D = 5.000$ değerleri sabit)

τ (Dakika)	Zaman Gecikmesi	Önceki Konum Gönderme	Reddedilen İstek	Değiştirilmeyen İstek
5	187	0	0	914
10	214	0	0	887
15	232	0	0	869
20	237	0	10	854
25	232	2	91	776
30	164	3	196	738
35	36	12	344	709
40	28	25	365	683
45	26	26	386	663
50	23	27	398	653
55	20	27	414	640
60	17	30	421	633
65	11	33	433	624
70	11	35	461	594
75	10	34	467	590
80	6	37	473	585
85	5	36	478	582
90	6	38	478	579
95	9	37	481	574
100	8	37	482	574

yapacağı için küçük olması daha gerçekçiyken, kullanıcıların mahrem örüntüleri desteklememek adına bekleyebilecekleri süre çok daha büyük olabilir. Çizelge 5.5, değişen T değerleriyle birlikte alınan sonuçları listelemektedir. Görüldüğü üzere T'nin artmasına bağlı olarak uygulanan zaman gecikmesi işlemi artış göstermektedir.

$T = 120$ dakikadan sonra herhangi bir artış olmamaktadır, bunun sebebiyse mahrem olarak nitelendirilebilecek tüm istekler için, daha açık bir deyişle işlem yapılması gereken tüm isteklere zaman gecikmesi uygulanabildiği için bu değerden sonra herhangi bir artış görülmemektedir ki bunu, değiştirilmeyen isteklerin sayısının sabit kalmasından da anlayabilmekteyiz. T değerinin artışına bağlı olarak meydana gelen önceki konum gönderme ve isteği reddetme işlemlerindeki azalma da yine zaman gecikmesi yapılan isteklerin sayısının artmasıyla açıklanmaktadır. İşlem uygulanması gereken istek sayısı hemen hemen aynı olduğu için zaman gecikmesi yapılan işlemlerin sayısı arttıkça diğerlerinin düşmesi beklenen bir durumdur. Şekil 5.10, T değeri değişirken alınan sonuçların yüzdesel dağılımını göstermektedir. Zaman gecikmesi uygulanan isteklerin artışı ve sonrasında tüm işlemlerin sabit kalışı buradan takip edilebilmektedir.

$T = 120$ dakika değerine kadar meydana gelen değiştirilmeyen istek sayısındaki artış, yani herhangi bir işlem yapılmasına gerek duyulan işlemlerin sayısındaki azalma, uygulanan zaman gecikmesi işlemlerinin, kullanıcı rotalarını değiştirmesinden dolayı

Çizelge 5.5: Değişen T değerlerine göre Gowalla verisi simülasyon çalışma sonuçları. ($\tau = 30$ ve $D = 5.000$ değerleri sabit)

T (Dakika)	Zaman Gecikmesi	Önceki Konum Gönderme	Reddedilen İstek	Değiştirilmeyen İstek
15	8	27	365	701
30	14	25	358	704
45	23	20	351	707
60	37	12	323	729
65	98	9	267	727
70	164	3	196	738
75	230	2	105	764
80	231	2	98	770
85	241	0	22	838
90	246	0	10	845
105	250	0	4	847
120	252	0	0	849
135	252	0	0	849
150	252	0	0	849
165	252	0	0	849
180	252	0	0	849
195	252	0	0	849
210	252	0	0	849
225	252	0	0	849
240	252	0	0	849

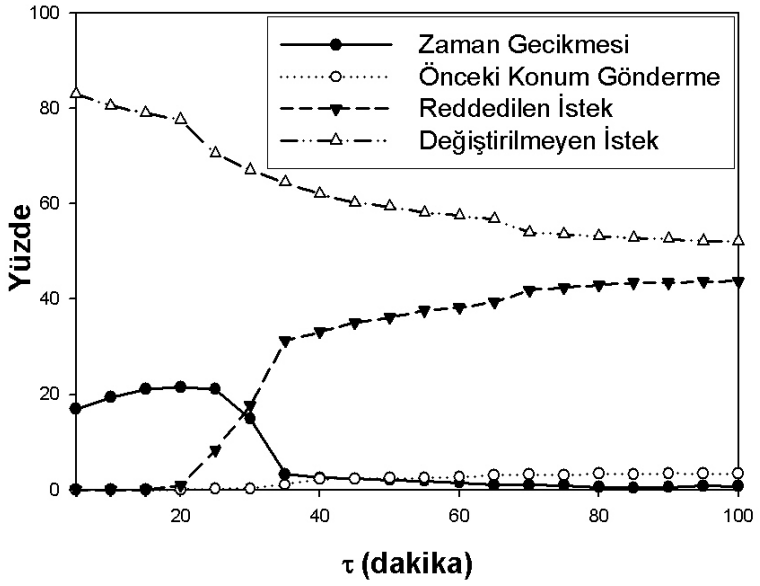
meydana gelmektedir. Daha açık bir deyişle, bir kullanıcı bir servis isteğinde bulunması sonrası o servis isteğine zaman gecikmesi uygulanacaksa o istek en fazla T dakika ötelenecektir bu da kullanıcının rotasında en fazla T dakikalık bir sapma meydana getirecek ve sapma olmasa desteklenecek örüntülerin desteklenmesini engelleyecektir. Aynı şekilde sapma yüzünden, önceden desteklenmeyecek örüntüler artık desteklenir hale de gelebilir ancak kullanıcılara mahrem örüntü kümesi atanırken ilk rotaları referans olarak alındığı için bunun örneği yok denecek kadar azdır.

Son olarak kullanıcı uzaklık tahammül değişkeni olan D değerinin değişimine göre uygulanan önceki konum gönderme işlemi sayısındaki değişimi inceleyebilmek adına, τ ve T değerleri sabitken D değeri 500 metre ile 10.000 metre arasında artacak şekilde değiştirilmiştir. Aynı şekilde yapılan yirmi çalışmada, τ ve T değişkenlerinin ikisi de 60 dakika olacak şekilde şeçilmiştir. Bu şekilde seçilmesinin sebebi, bu değerlerle, uygulanan zaman gecikmesi işleminin sayısının hemen hemen sabit kalmasıdır, bu işlem sayısı sabit kaldıkça, esas değişim önceki konum gönderme ya da isteği reddetme işlemlerinin sayısından rahatlıkla takip edilebilecektir.

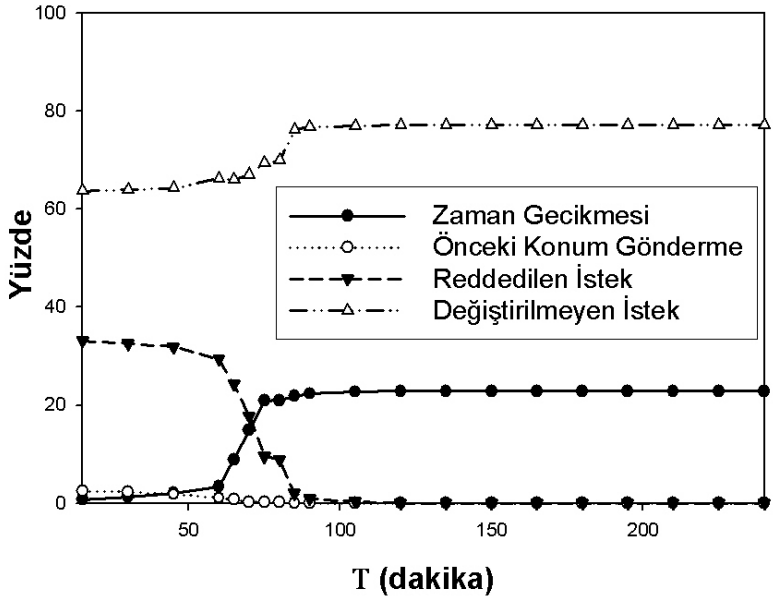
D değerinin artışıyla doğru orantılı olarak artan önceki konum gönderme işleminin sayısındaki artış ve diğer tüm sonuçlar, Çizelge 5.6'da verilmiştir. Aynı şekilde Şekil 5.11 ise, bu değerlerin yüzdelik dağılımını göstermektedir. Uygulanan önceki konum gönderme işleminde, D değerine bağlı olarak meydana gelen artış,

Çizelge 5.6: Değişen D değerlerine göre Gowalla verisi simülasyon çalışma sonuçları. ($\tau = 60$ ve $T = 60$ değerleri sabit)

D (Metre)	Zaman Gecikmesi	Önceki Konum Gönderme	Reddedilen İstek	Değiştirilmeyen İstek
500	12	7	495	587
1.000	12	16	454	619
1.500	12	17	453	619
2.000	12	24	444	621
2.500	12	24	444	621
3.000	12	25	443	621
3.500	12	25	443	621
4.000	12	25	443	621
4.500	12	30	432	627
5.000	12	30	432	627
5.500	12	35	422	632
6.000	12	35	422	632
6.500	12	26	421	642
7.000	13	36	406	646
7.500	13	36	406	646
8.000	11	48	378	664
8.500	11	48	378	664
9.000	11	48	378	664
9.500	11	48	378	664
10.000	10	51	375	665

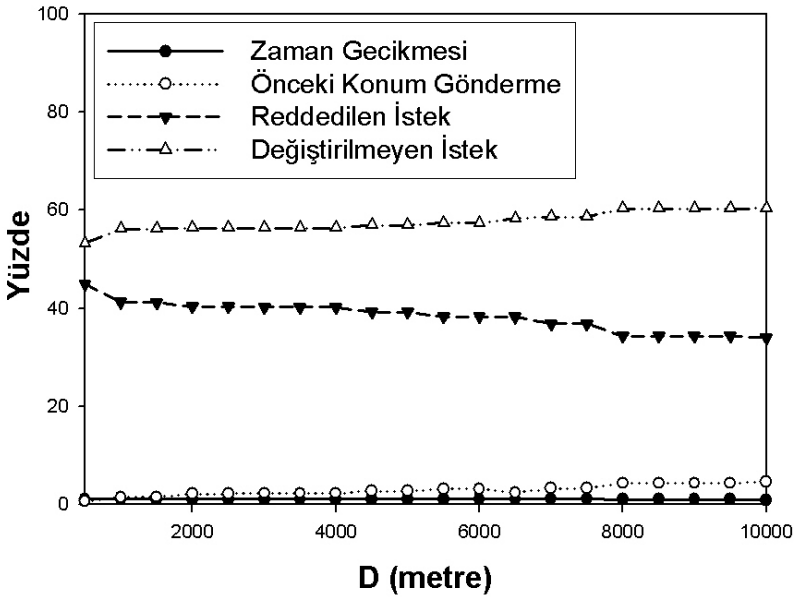


Şekil 5.9: Mahrem olarak belirtilen örüntülerin $T = 70$ ve $D = 5.000$ değerleri sabitken farklı τ değerleriyle birlikte verdiği yüzdesel oranlar



Şekil 5.10: Mahrem olarak belirtilen örüntülerin $\tau = 30$ ve $D = 5.000$ değerleri sabitken farklı T değerleriyle birlikte verdiği yüzdesel oranlar

Çizelge 5.5'teki T değerine bağlı uygulanan zaman gecikmesi sayısındaki artış kadar ciddi değildir. Bunun sebebiyse mahrem örüntü kümesinde tanımlanan ve desteklenen örüntülerin bir kısmının önceki konum gönderme işlemiyle uygun hale getirilemeyecek olmasından kaynaklıdır. Ayrıca, τ sabit olmasına rağmen değiştirilmeyen istek sayısında meydana gelen azalma da yine uygulanan işlemlerden sonra rotanın değişmesinden kaynaklıdır.



Şekil 5.11: Mahrem olarak belirtilen örüntülerin $\tau = 60$ ve $T = 60$ değerleri sabitken farklı D değerleriyle birlikte verdiği yüzdesel oranlar

6. SONUÇ

Bu çalışmada konum mahremiyetinin gerçek anlamda kişiselleştirilmesi için konum örüntü mahremiyeti problemi tanıtılmıştır. Bu modelde kullanıcı konum mahremiyeti sağlayıcısı tarafından üretilen perdeleme haritası üzerinden kendine özgü mahrem örüntüleri tanımlar. Geliştirdiğimiz sistem her konum tabanlı istek sorgusunda bu isteğin güvenli olup olmadığını kontrol ederek güvenli değilse güvenli hale getirecek şekilde zaman gecikmesi ya da önceki konumu gönderme seçeneklerini değerlendirir.

İnternet ve GPS teknolojisinin kullanılmasıyla beraber, taşınabilir cihazlarla etkileşimi artan insanlar, sürekli çevrimiçi kalmaktadırlar. Yararlanılan servis ve uygulamalar kullanıcılara daha iyi hizmet verebilmek adına kullanıcılarının bazı bilgilerini toplamakta ve depolamaktadır. Aynı şekilde konum-tabanlı, kullanıcının bulunduğu konuma göre bir hizmette bulunan servisler de, kullanıcılarının konum ve zaman bilgilerini toplamakta, zaman sıralısı olarak bakıldığında kullanıcının hangi zamanda nerede olduğu bilgisine erişebilmektedir. Kullanıcıların, hangi zamanda nerede olduğundan da öte, sık gittiği yerler ve her gün düzenli kullandığı güzergahların bilgisi bile bulunabilmektedir. Toplanan bu bilgiler, bazı istatistiksel sonuçlara ulaşabilmek adına analiz edilmekte ve bunlardan veri madenciliği teknikleriyle anlamlar çıkarılmaktadır. Bu çıkarılan sonuçların ya da kullanıcının bilgilerinin arasında, kullanıcı için mahrem olarak düşünülebilecek bilgiler varolabilir. Bu çalışmada, kullanıcıların mahrem olarak nitelendirebileceği örüntülerin, konum istek geçmişlerinde oluşmasını önlemeye yönelik, çevrimiçi uygulanabilecek bir model oluşturulmuş, hayata geçirilmiş ve deneysel sonuçlar incelenmiştir. Bilindiği kadarıyla, lüteratürde anlık olarak konum örüntü mahremiyetini sağlamaya yönelik bir çalışma yoktur ve ilk kez bu çalışmada ele alınmıştır. Öncelikle kullanıcı, konum mahremiyetini sağlayabilmek adına, Bölüm 3'te de bahsedildiği şekilde, mahremiyet profilini oluşturur ve bir perdelenmiş haritaya sahip olur. Bölüm 2.3.1'de de anlatılan, hız kısıtları gibi gerekli arka plan bilgilerine uyum sağlayan bir sistem sayesinde kullanıcının konum mahremiyeti garanti altına alınır.

Kullanıcı, konum örüntü mahremiyetini sağlayabilmek adınaysa, mahrem olarak nitelendirdiği örüntülerin tanımlarını yapar. Konum ve zaman ikililerinden oluşan örüntü sıralıları bir küme olarak tanımlanır. Bu aşamadan sonra, kullanıcının yaptığı her bir servis isteğinde, mevcut örüntülerden herhangi birisinin desteklenip desteklenmediği kontrol edilir. Hiç bir örüntü desteklenmiyorsa, servis isteği güvenli olarak düşünülebilir, aksi halde, herhangi bir örüntüyü bile destekleme durumunda, sırasıyla (*i*) zaman gecikmesi, (*ii*) önceki konumu gönderme ve (*iii*) isteği reddetme işlemlerinden birisi uygulanır. Zaman gecikmesi ve önceki konumu gönderme işlemi uygulandıktan sonra servis isteği gönderilmeden, tekrar mahrem örüntü kümesiyle eşleşmesine bakılır, çünkü daha önceden kontrol edilmiş ancak desteklemeyen bir

örüntü, yapılan işlem sonrasında destekleyecek duruma gelmiş olabilir. Sonuç olarak kullanıcının belirlediği zaman tahammül eşik değeri (T) ve uzaklık tahammül eşik değeri (D) değişkenlerinin değerine göre isteğe yapılacak işlem belirlenir.

Konum örüntü mahremiyeti sağlayabilmek adına geliştirilen model, bir sunucu, istemci ve simülatör uygulaması olarak hayata geçirilmiştir. İstemci, Android işletim sistemi üzerinde çalışan ve konum örüntü mahremiyetini garanti altına almak isteyen kullanıcıların kullandıkları kişisel uygulamadır. Mahrem örüntü ve sistem tahammül değişkenleri tanımlamaları burada yapılır. Sunucu, istemcinin servis isteklerini gönderdiği ve Oracle Glassfish üzerinde çalışan bir web servisdir. İstemciden gelen isteklere göre Bölüm 4.3 ve 4.4'da anlatılan kontrol ve güvenli hale getirilme işlemlerini uygulayarak uygun cevabı istemci ile paylaşır. Simülatör uygulamasıysa deneysel sonuçlara ulaşabilmek adına, büyük hacimde iki farklı veriyi, farklı servis istekleriymiş gibi sunucuya göndermeyi ve istatistiksel sonuçlara ulaşabilmeyi sağlamaktadır. Simülatör uygulamasında kullanılan veriler Bölüm 5.4'te anlatıldığı gibi Milano GPS ve Gowalla veri kümeleridir ve simülatör Milano GPS veri kümesi için her bir çalışmada 4269 tane servis isteğinde bulunulmak üzere toplamda kırk beş kere farklı değişken değerleriyle, Gowalla veri kümesi içinse her bir çalışmada 1101 tane servis isteğinde bulunumak üzere toplamda altmış kere farklı değişken değerleriyle çalıştırılmıştır.

Deneysel sonuçlar incelendiğinde, kullanıcıların belirlediği değişken değerlerine göre, beklenen sonuçlar elde edilmiş, mahrem örüntülerin zaman kısıtını sıkılaştırmak ya da esnetmek adına kullanılan τ değişkeninin değeriyle doğru orantılı olarak desteklenen ve işlem gören mahrem örüntülerin sayısının değiştiği deneyimlenmiştir. Ayrıca zaman ve uzaklık tahammül değişkenlerinin değişimlerine göre de, uygulanan zaman gecikmesi ve önceki konum gönderme işlemlerinin sayılarında da beklenilen değişimler görülmüştür.

Bu çalışmada oluşturulan istemci, tam olarak, kullanıcının mahremiyetini engelleyecek bir durum olup olmadığını sorgulayabilmesi adına kullandığı bir yan uygulamadır. Gelecekte yapılabilecek çalışmalar arasında, bu çalışmayı temel alarak, kullanıcıların servis istekleri sırasında, servis isteğinin uygulama sunucularına gönderilmeden hemen önce araya bir katman oluşturulabilir ve bu katman sayesinde kullanıcı ayrıca bir uygulama kullanmadan mahremiyetini garanti altına almış olur. Ek olarak, konum-tabanlı servis sağlayıcıları için bir standard geliştirilip, geçmişte yapılan konum mahremiyeti ve geriye dönük çevrimdışı konum örüntü verileri temizlenmesi seçeneklerinin yanı sıra bu çalışmadaki çevrimiçi konum örüntü mahremiyeti de eklenebilir. Böylelikle bu model sayesinde her bir kullanıcının ek bir katman ya da uygulamaya ihtiyacı kalmadan mahremiyetleri sağlanabilir.

Özetle, deneysel sonuçlardan da anlaşılabileceği üzere bu tez çalışmasının hedeflendiği şekilde tamamlandığı ve amacına ulaştığı ve daha önce üzerinde çalışma yapılmamış olan çevrimiçi konum örüntü mahremiyetinin sağlanması konusunda literatüre önemli katkıların yapıldığı söylenebilir.

KAYNAKLAR

[1] **Abul, O., Bonchi, F., and Giannotti, F.** Hiding sequential and spatiotemporal patterns. *IEEE Transactions on Knowledge and Data Engineering 22*, 12 (2010), 1709–1723.

[2] **Adam, N. R., and Worthmann, J. C.** Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys 21*, 4 (1989), 515–556.

[3] **Aggarwal, G., Panigrahy, R., Feder, T., Thomas, D., Kenthapadi, K., Khuller, S., and Zhu, A.** Achieving anonymity via clustering. *ACM Transactions on Algorithms 6*, 3 (2010), 1–19.

[4] **Anwar, T., Liu, C., Vu, H., and Leckie, C.** Spatial Partitioning of Large Urban Road Networks. *EDBT2014*, c (2014), 343–354.

[5] **Atallah, M., Bertino, E., Elmagarmid, A., Ibrahim, M., and Verykios, V.** Disclosure limitation of sensitive rules. *Proceedings 1999 Workshop on Knowledge and Data Engineering Exchange (KDEX'99) (Cat. No.PR00453)* (1999), 45–52.

[6] **Brinkhoff, T.** A framework for generating network-based moving objects. *GeoInformatica 6*, 2 (2002), 153–180.

[7] **Capt, G., Gupta, A., and Fellow, S.** 'POKÉMON GO' MANIA: AN INFLECTION POINT FOR AUGMENTED REALITY?

[8] **Chin, F. Y., and Ozsoyoglu, G.** Statistical database design. *ACM Transactions on Database Systems 6*, 1 (1981), 113–139.

[9] **Cho, E., Myers, S. A., and Leskovec, J.** Friendship and mobility: User Movement in Location-Based Social Networks. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (2011), 1082–1090.

[10] **Chow, C.-Y., and Mokbel, M. F.** Trajectory privacy in location-based services and data publication. *ACM SIGKDD Explorations Newsletter 13*, 1 (2011), 19.

[11] **Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E.** Introduction to Algorithms. McGraw-Hill Higher Education, 2nd edition, 2001.

[12] **Damiani, M. L., Bertino, E., and Silvestri, C.** The PROBE framework for the personalized cloaking of private locations. *Transactions on Data Privacy 3*, 2 (2010), 123–148.

[13] **Damiani, M. L., Silvestri, C., and Bertino, E.** Fine Grained Cloaking Of Sensitive Positions In Location Sharing Applications. *IEEE Pervasive Computing 10*, 4 (2011), 64–72.

[14] **De Almeida, V. T., Güting, R. H., and Behr, T.** Querying moving objects in SECONDO. *Proceedings - IEEE International Conference on Mobile Data Management 2006* (2006).

[15] **Erwig, M., Guting, R. H., Schneider, M., and Vazirgiannis, M.** Spatio-temporal data types: An approach to modelling and querying moving objects in databases. *Geoinformatica 3* (1999), 269–296.

[16] **Gedik, B., and Liu, L.** Location privacy in mobile systems: A personalized approach. *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems* (2005).

[17] **GeoPKDD**. Geographic Privacy-aware Knowledge Discovery and Delivery. Available at `http://www.geopkdd.eu`.

[18] **Ghinita, G.** Private queries and trajectory anonymization: A dual perspective on location privacy. *Transactions on Data Privacy 2*, 1 (2009), 3–19.

[19] **Ghinita, G., Damiani, M. L., Silvestri, C., and Bertino, E.** Preventing Velocity based Linkage Attacks in Location Aware Applications Categories and Subject Descriptors. *Gis* (2009), 246–255.

[20] **Gidófalvi, G., and Pedersen, T. B.** Mining long, sharable patterns in trajectories of moving objects. *CEUR Workshop Proceedings 174* (2006), 49–58.

[21] **Güting, R. H.** Second-order signature: a tool for specifying data models, query processing, and optimization. *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data* (1993), 277–286.

[22] **Güting, R. H., Dieker, S., Freundorfer, C., Becker, L., and Schenk, H.** Secondo/qp: Implementation of a generic query processor. In *International Conference on Database and Expert Systems Applications* (1999), Springer, pp. 66–87.

[23] **Haghnegahdar, A., Khabbazian, M., and Bhargava, V. K.** Privacy risks in publishing mobile device trajectories. *IEEE Wireless Communications Letters 3*, 3 (2014), 241–244.

[24] **inc. Encyclopædia Britannica**. Moore's law. Available at `https://global.britannica.com/topic/Moores-law`.

[25] **Jeung, H., Liu, Q., Shen, H. T., and Zhou, X.** A hybrid prediction model for moving objects. *Proceedings - International Conference on Data Engineering* (2008), 70–79.

[26] **Kang, J., and Yong, H.-s.** Mining Spatio Temporal Patterns In Trajectory Data. *Journal of Information Processing Systems 6*, 4 (2010), 521–536.

[27] **Krumm, J.** A survey of computational location privacy. *Personal and Ubiquitous Computing 13*, 6 (2009), 391–399.

[28] **Lee, G., Chang, C.-Y. C., and Chen, A. A. L. P.** Hiding sensitive patterns in association rules mining. *Computer Software and ...* (2004), 0–5.

[29] **Leskovec, J., and Krevl, A.** SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[30] **Li, Z., Han, J., Ji, M., Tang, L., and Yu, Y.** MoveMine: Mining moving object data for discovery of animal movement patterns. *ACM Transactions on Intelligent Systems and Technology (TIST) 2*, 4 (2011), 32.

[31] **Liu, F., Hua, K. A., and Cai, Y.** Query l-diversity in location-based services. *Proceedings - IEEE International Conference on Mobile Data Management* (2009), 436–442.

[32] **Luaces, M. R.** A spatio-temporal algebra implementation. In *Proceedings of the Fifth World Conference on Integrated Design and Process Technology* (2000), Citeseer.

[33] **Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M.** L-diversity. *ACM Transactions on Knowledge Discovery from Data 1*, 1 (2007), 3–es.

[34] **Nergiz, M. E., Atzori, M., Saygin, Y., and Bariş, G.** Towards trajectory anonymization: A generalization-based approach. *Transactions on Data Privacy 2*, 1 (2009), 47–75.

[35] **O'Leary, D. E.** Knowledge discovery as a threat to database security. *Knowledge discovery in databases 9* (1991), 507–516.

[36] **Ozsoyoglu, G., and Ozsoyoglu, Z.** Statistical database query languages. *Software Engineering, IEEE Transactions on 10*, 10 (1985), 1071–1081.

[37] **Sakr, M., Andrienko, G., Behr, T., Andrienko, N., Güting, R. H., and Hurter, C.** Exploring spatiotemporal patterns by integrating visual analytics with a moving objects database system. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '11* (2011), 505.

[38] **Samarati, P., and Sweeney, L.** Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement Through Generalization and Suppresion. *Proc of the IEEE Symposium on Research in Security and Privacy* (1998), 384–393.

[39] **Sanou, B.** Ict facts and figures 2015. *International Telecommunication Union (ITU) Fact Sheet* (2015).

[40] **Shokri, R., Theodorakopoulos, G., Troncoso, C., Hubaux, J.-P., and Le Boudec, J.-Y.** Protecting Location Privacy: Optimal Strategy against Localization Attacks. *Proceedings of the 2012 ACM conference on Computer and communications security* (2012), 617–627.

[41] **Solanas, A., Domingo-Ferrer, J., and Martínez-Ballesté, A.** Location privacy in location-based services: Beyond TTP-based schemes. *CEUR Workshop Proceedings 397* (2008), 12–23.

[42] **Sun, X., and Yu, P. S.** Hiding Sensitive Frequent Itemsets by a Border-Based Approach. *Journal of Computing Science and Engineering 1*, 1 (2007), 74–94.

[43] **Sweeney, L.** k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10*, 05 (2002), 557–570.

[44] **Terrovitis, M., and Mamoulis, N.** Privacy preservation in the publication of trajectories. *Proceedings - IEEE International Conference on Mobile Data Management* (2008), 65–72.

[45] **Theodorakopoulos, G., Shokri, R., Troncoso, C., Hubaux, J.-p., and Le Boudec, J.-Y.** Prolonging the Hide-and-Seek Game: Optimal Trajectory Privacy for Location-Based Services. *Wpes'14*, 4 (2014), 73–82.

[46] **Yigitoglu, E., Damiani, M. L., Abul, O., and Silvestri, C.** Privacy-preserving sharing of sensitive semantic locations under road-network constraints. *Proceedings - 2012 IEEE 13th International Conference on Mobile Data Management, MDM 2012* (2012), 186–195.

ÖZGEÇMİŞ

Ad-Soyad : Cansın Bayrak

Uyruğu : T.C.

Doğum Tarihi ve Yeri : 24.05.1991 - Antalya

E-posta : c.bayrak@etu.edu.tr

ÖĞRENİM DURUMU:

- **Yüksek Lisans** : 2016, TOBB ETU, Bilgisayar Mühendisliği
- **Lisans** : 2014, TOBB ETU, Bilgisayar Mühendisliği
- **Lisans** : 2014, TOBB ETU, Endüstri Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	Yer	Görev
2014-2016	TOBB ETU	Tam Burslu Yüksek Lisans Öğrencisi
2014-2014	Innova IT Solutions	Stajyer
2013-2013	Medical Fly GmbH	Stajyer
2012-2012	Athena Computer	Stajyer
2009-2014	TOBB ETU	Üstün Başarı Burslu Lisans Öğrencisi

YABANCI DİL: İNGİLİZCE

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Bayrak, Cansın** ve Abul, Osman, "Konum Tabanlı Servislerde Konum Örüntü Mahremiyeti". *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu 2016* (2016).