

**ŞİFRELİ BULUT VERİLERİ ÜZERİNDE GÜVENLİ VE
EFEKTİF ÇOKLU ANAHTAR KELİME İLE SIRALI ARAMA
YÖNTEMİ**

**EFFICIENT PRIVACY PRESERVING MULTI KEYWORD
RANKED SEARCH METHOD OVER ENCRYPTED CLOUD
DATA**

KUBRA MAMMADOVA

Yrd.Doç.Dr. Murat Aydos

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin

Bilgisayar Mühendisliği Anabilim Dalı İçin Öngördüğü

YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

KUBRA MAMMADOVA'nın hazırladığı “Şifreli Bulut Verileri Üzerinde Güvenli ve Etkatif Çoklu Anahtar Kelime İle Sıralı Arama Yöntemi” adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Mehmet Önder Efe

Başkan



Yrd. Doç. Dr. Murat Aydos

Danışman



Yrd. Doç. Dr. Erhan Mengüşoğlu

Üye



Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Salih Bülent ALTEN
Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, yüksek lisans tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada

- Tez çalışmamda kullandığım bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başka araştırmacıların çalışmalarından veya eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

15/07/2016



KUBRA MAMMADOVA

ÖZET

ŞİFRELİ BULUT VERİLERİ ÜZERİNDE GÜVENLİ VE EFEKTİF ÇOKLU ANAHTAR KELİME İLE SIRALI ARAMA YÖNTEMİ

Kubra MAMMADOVA

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Yrd. Doç. Dr. Murat AYDOS

Temmuz 2016, 93 Sayfa

Sürekli yeni donanım ve yazılım gereksinimlerinin piyasaya sunulduğu ve geliştirildiği aşırı bilgi yoğunluğu döneminde, bulut bilişimi kullanıcıların ihtiyaç duydukları verilere zamanında ulaşmalarını sağlayan ve bilgi trafiğine çözüm olabilecek güçlü bilgi işlem mimarisidir. Bulut yapısı, kullanıcılarına sunduğu esneklik, yüksek veri paylaşımı üst sınırı, basit altyapı, basit veri yönetim sistemi, ölçeklenebilirlik, ayrıca ekonomik açıdan tasarruflu, kullanıcı dostu hizmetleri nedeniyle geniş kullanıcı kitlesi tarafından tercih edilmektedir. Kullanıcıların (kişiler, kurumlar, şirketler) dış kaynakta depolamak istedikleri hassas bilgilerinin (kimlik bilgileri, hasta kayıtları, finansal bilgiler vb.) güvenliğini ve gizliliğini sağlamak için kullandıkları şifreleme yöntemleri, arama metotları da dahil olmak üzere düz metinler üzerinde yapılan ve efektif sonuçlar elde edilmesini sağlayan işlemleri etkisiz hale getirmiştir. Bu sorunu çözmek ve verimli sonuçlar elde etmek için dış kaynağa aktarılan verilerin aranabilir şifreleme yöntemleri kullanılarak şifrelenmesi arama yapılmasını mümkün kılmaktadır. Şifreli bulut verileri üzerinde arama algoritmaları büyük veri belgeleri koleksiyonlarının güvenliği ve gizliliğini sağlarken, yetkili kullanıcıların ihtiyaç duydukları dosyalara efektif ve verimli şekilde erişimini sağlar, ayrıca yetkisiz kullanıcıların ve bulut sunucusunun verilerin içeriğine istatistiksel veya tahmin yoluyla erişmesini engellemektedir.

Sunulan şifreli bulut verileri üzerinde arama yöntemlerinden bazıları efektif sonuçların döndürülmesini sağlarken veri güvenliğınden taviz vermektedir. Diğer metotlarda ise, güçlü güvenlik ve gizlilik gereksinimleri sunulurken, döndürülen sonuçların verimliliğı ve efektifliğı yetersiz düzeydedir. Arama yöntemlerinin bir kısmı ise matematiksel içeriğı nedeniyle veriler kısıtlı yetkili veya yetkisiz kullanıcıların ve sunucunun istatistiksel vb. saldırılarına karşı savunmasızdır.

Tez kapsamında, bulut sunucularına aktarılmış şifreli veriler üzerinde yapılan aramalarda verimli sonuçlar döndürürken verilerin güvenliğınden taviz vermeyen gizlilik korumalı çoklu anahtar kelime ile sorgu oluşturulmasına izin veren daha verimli bir algoritma geliştirilmiştir. Şifreli bulut verileri üzerinde ilk defa uygulanan ORE aranabilir şifreleme yöntemi ve iki katmanlı Bloom filtresi ile bu algoritma sunulmuş diğer arama algoritmalarından ciddi şekilde farklıdır ve daha efektif sonuçlar döndürmektedir. Bu yöntem yerel sunuculara aktarılmış deneme veri belgeleri üzerinde uygulanmış ve optimum sürede daha verimli sonuçlar döndürdüğü gözlemlenmiştir.

Anahtar Kelimeler: bulut bilişimi, aranabilir şifreleme, çoklu kelimeli arama, sıralama mantığı, benzerlik ölçekleri, şifreli bulut veriler, arama sorguları.

ABSTRACT

EFFICIENT PRIVACY PRESERVING MULTI KEYWORD RANKED SEARCH METHOD OVER ENCRYPTED CLOUD DATA

Kubra MAMMADOVA

Master of Science, Department of Computer Engineering

Supervisor: Asst. Pr. Murat AYDOS

July 2016, 93 Pages

With its offered user-friendly services such as flexibility, high data sharing limit, simple infrastructure, simple data management system, scalability, efficiency in the term of economy, cloud computing is preferred by user audience and data owners motivated to transfer their data collection from local sites to cloud storage server. User audience and data owners who are motivated to transfer their data collection from local sites to a cloud storage server prefer cloud computing because of the user-friendly services it offers, such as its flexibility, high data sharing limit, simple infrastructure, simple data management system, scalability and efficiency in the terms of economy. Encryption techniques used for relieving user's (i.e. individuals', institutions', companies') concerns about the provision of security and privacy of outsourced sensitive data (identification information, patient records, financial information, etc.) disable procedures that are used to get needed documents and for active information searching. To ensure access to the data that clients need and to improve the usage and efficiency of cloud servers, various search methods that allow for generating query request are proposed while maintaining the security and confidentiality of encrypted cloud data. While ensuring the security and confidentiality of large data document collections, search methods allow authorized users to get needed data efficiently. In addition, these techniques protect content of data documents against unauthorized users and cloud servers that want to get access with statistical or estimate attacks. Some of the presented

search methods on the encrypted cloud data made concession from data security, while returning effective search results. Several methods return search results with low level efficiency while providing strong security and privacy requirements. Part of the developed methods are vulnerable against statistical attacks by limited authorized or unauthorized users and servers because of its mathematical content.

In this thesis, search method that allows to return search result with efficient privacy preserving multi keyword query request generated by client and which does not compromise the security of the data while returning effective search results are presented. This method differs from other presented techniques with its efficient searchable encryption and by using two-tier Bloom filter. This algorithm tested on the local servers and returned effective results on the test document collections.

Keywords: Cloud computing, searchable encryption, multi keyword search, ranking search, similarity measures, encrypted cloud data, privacy preserving.

TEŞEKKÜR

Tez konusunun belirlenmesini sağlayan, bu tezin geliştirilmesi ve tamamlanması esnasında yol göstermesi ve yapıcı eleştirileri için, ayrıca tez metninin yazılmasına ve tez çalışmasının hazırlanmasına yaptığı yardımlar için danışmanım Sayın Yrd. Doç. Dr Murat Aydos'a sonsuz teşekkürlerimi sunarım.

Ayrıca tez metnini inceleyerek önemli yorum ve değerlendirmeleri ile biçim ve içerik bakımından son halini almasına yardımcı olan Sayın Prof. Dr. Mehmet Önder Efe'ye, Sayın Yrd. Doç. Dr Erhan Mengüçoğlu'ya teşekkürlerimi sunarım.

Tez çalışmalarında yaptığı yardım ve verdiği desteklerden dolayı Aydın Kaya'ya, çalışma boyunca desteklerini eksik etmeyen ve fikirlerini benimle paylaşan, Hacettepe Üniversitesi Bilgisayar Mühendisliği'ndeki değerli arkadaşlarıma teşekkürlerimi sunmayı kendime borç bilirim.

Hem bilimsel konularda, hem de manevi olarak her daim yanımda olan, en zor ve gergin zamanlarımda desteğini ve yardımını esirgemeyen hayat arkadaşım Özgün Bursalıoğlu'na, hayatım boyunca bana inanan ve her konuda yanımda olup beni destekleyen sevgili aileme ve kardeşlerime sonsuz sevgi ve teşekkürlerimi sunmayı kendime borç bilirim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	I
ABSTRACT	III
TEŞEKKÜR	V
İÇİNDEKİLER	VI
ŞEKİLLER	viii
ÇİZELGELER	ix
SİMGELER ve KISALTMALAR	x
SÖZLÜK	xii
1. GİRİŞ	1
2. BULUT BİLİŞİMİ	4
2.1. Genel Tanım	4
2.2. Bulut Bilişiminin Kısa Tarihçesi	5
2.3. Bulut Modelleri	6
2.4. Bulut Hizmetleri	7
2.5. Bulut Bilişimi ve Grid Paradigması	7
3. ŞİFRELE BULUT VERİLERİ: GÜVENLİK	9
3.1. İlgili Çalışmalar	11
3.2. Veri Güvenliği	12
3.2.1. Güvenlik ve Gizlilik Koşulları	12
3.2.2. Aranabilir Şifreleme	14
3.2.3. Aranabilir Açık Anahtarlı Şifreleme	16
3.2.4. Veritabanlarında Özel Bilgi Erişimi (PIR)	18
3.2.5. Aranabilir Simetrik Şifreleme	18
3.2.5.1. Simetrik Şifreleme	18
3.2.5.2. Simetrik Aranabilir Şifreleme	19
3.2.6. Özellik Korumalı Şifreleme	23
3.2.6.1. Düzen Açıklayıcı Şifreleme	25
3.3. Tam Anahtar Kelime Arama	25
3.3.1. Açık Anahtar Şifreleme Tabanlı Arama	26
3.3.2. Simetrik Anahtar Şifreleme Tabanlı Arama	27
3.4. Bulanık Anahtar Kelime Arama	28

4. ARAMA YÖNTEMLERİ	30
4.1. Genel Tanım	30
4.2. Tek Anahtar Kelime ile Arama	30
4.3. Boole Anahtar Kelime ile Arama	31
4.4. Birden Fazla Anahtar Kelime ile Arama	31
5. İNDİS OLUŞTURMA YÖNTEMLERİ	32
5.1. İndeks Mimarisi	33
5.2. Bloom Filtresi	34
5.2.1. Bloom Filtresinin Çalışma Prensipi	36
6. BENZERLİK TABANLI ARAMA	38
6.1. Benzerlik Ölçüsü	38
6.2. Sıralama Mantığı	42
7. DÜZEN AÇIKLAMA ŞİFRELEME YÖNTEMİNE DAYALI ŞİFRELİ BULUT VERİLERİ ÜZERİNDE ARAMA ALGORİTMASI	44
7.1. Algoritmanın Genel Yapısı	44
7.2. Pozisyona Duyarlı Özetleme Fonksiyonu	44
7.3. Algoritmada Bloom Filtresinin Yeri	45
7.4. Düzen Açıklayıcı Şifreleme	47
7.5. Sıralama Fonksiyonu	47
7.6. Algoritmanın Aşamaları	48
7.7. Birinci Aşama	48
7.8. İkinci Aşama	50
7.9. Yöntemin Analizi ve Karşılaştırma	50
7.10. Verimlilik	51
7.10.1. Farklı Şartlarda Arama Süresi	52
SONUÇ	53
KAYNAKLAR	54
ÖZGEÇMİŞ	70

ŞEKİLLER

	<u>Sayfa</u>
Şekil 2.1. Bulut bilişimi ve hizmetleri	5
Şekil 2.2. Bulut hizmetleri yönetimi	
Şekil 3.1. Şifreli bulut verileri üzerinde arama	10
Şekil 3.2. Kimlik doğrulama	10
Şekil 3.3. Veri erişimi	11
Şekil 3.4. Bulut bilişiminde güvenlik zorlukları	13
Şekil 3.5. Bulut bilişimi güvenlik gereksinimleri	14
Şekil 3.6. Genel güvenlik ilkeleri	15
Şekil 3.7. Bulut verileri üzerinde arama yöntemlerinde anarabilir şifreleme yöntemleri	17
Şekil 3.8. Anahtar kelime araması ile açık anahtar şifreleme	18
Şekil 3.9a. Adaptif sorgular	21
Şekil 3.9b. Adaptif olmayan sorgular	21
Şekil 3.10. Çok kullanıcı simetrik aranabilir şifreleme yönteminin genel yapısı	23
Şekil 3.11. Bulanık anahtar kelime arama	28
Şekil 3.12. Bulanık anahtar kelime arama şeması.....	32
Şekil 5.1. İndeks mimarisinin akış şeması	34
Şekil 5.2. Ağaç tabanlı indis mimarisinin genel akışı	35
Şekil 5.3. Bloom filtresinin açıklaması	36
Şekil 5.4. 4 özetleme fonksiyonu ile Bloom filtresi	37
Şekil 5.5. Yalnız pozitif oranı	37
Şekil 6.1. k – en yakın komşu	40
Şekil 7.1. İki katmanlı indis şemasının genel akış diyagramı	47
Şekil 7.2. Geçit oluşturulması süresi	51
Şekil 7.3. İndis oluşturulması süresi	51
Şekil 7.4. İndis şifreleme süresi	52
Şekil 7.5. İndis katmanlarının şifreleme süresinin karşılaştırılması	52
Şekil 7.6. Anahtar kelime sayısına bağlı olarak arama süresi	53

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 2.1. Bulut bilişimi ve Grid paradigmasının karşılaştırılması	9
Çizelge 3.1 Aranabilir şifreleme yöntemlerinin karşılaştırılması	16
Çizelge 3.2 Bazı aranabilir simetrik şifreleme ile arama yöntemlerinin karşılaştırılması	29
Çizelge 5.1 Ters indis mimarisinde dosyaların alaka puanı	33
Çizelge 5.2. Çeşitli m / n ve k kombinasyonları altında yalancı pozitiflik oranı ...	38
Çizelge 6.1 $T_f - I_{df}$ değerinin hesaplanma fonksiyonları	42



SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

$PT = \{p_i\}$	Kullanıcının n sayıda düz metin dosyaları
$C = \{c_i\}$	Bulut sunucularına aktarılan şifreli metinler
$D = \{D_i\}$	Düz metinlerden çıkarılan anahtar kelimeler sözlüğü
I	Düz metnin aranabilir indisi
k	Güvenlik Anahtarı
T	Geçit
d	Belirlenen mesafe
ID	Kullanıcı Kimliği
Q	İstemci tarafından oluşturulan sorgu
A	Saldırgan (sunucu veya üçüncü taraf)
f, r	Güvenlik parametreleri
s	Belirteç
E_r	Şifreleme algoritması
V	Talimat vektörü
H	Özetleme fonksiyonu ailesi
$a(H)$	Erişim örüntüsü
λ	Güvenlik parametresi
O	Girdi dosya etiketleri için operatörler
m, l	Bloom filtresi parametreleri
w'	Dosya etiketi

Kısaltmalar

SLA	Servis Seviyesi Anlaşması (İng. Service Level Agreement)
LSH	Mekana Duyarlı Özetleme (İng. Local Sensitive Hashing)
SSE	Simetrik Aranabilir Şifreleme (İng. Symmetric Searchable Encryption)
ORE	Düzen Açıklayıcı Şifreleme (İng. Order Revealing Encryption)
PKE	Açık Anahtar Şifreleme (İng. Public Key Encryption)
SKE	Simetrik Anahtar Şifreleme
PIR	Açık Bilgi Erişimi (İng. Private Information Retrieval)
PEKS	Açık Anahtar Şifrelemesi ile Arahtar Kelime Arama
OPE	Düzen Korumalı Şifreleme (İng. Order Preserving Encryption)
CVS	Eşzamanlı Sürüm Sistemleri
PPE	Özellik Korumalı Şifreleme (İng. Property Preserving Encryption)
PCPA	Seçilmiş Düz Metin Saldırılarına Karşı Yalancı Rastlantı (İng. Pseudorandomness against Chosen Plaintext Attacks)
TF	Terim Frekansı (İng. Term Frequency)
IDF	Ters Belge Frekansı (İng. Inverse Document Frequency)
Gen	Oluşturma algoritması
Enc	Şifreleme algoritması
Dec	Şifre çözme algoritması
AES	Gelişmiş şifreleme standardı (İng. Advanced Encryption Standard)

1. GİRİŞ

Bilgi yoğunluğunun her gün daha fazla arttığı günümüzde, büyük veri belgelerine sahip kişiler veya kurumlar tarafından bu veri belgelerinin bireysel olarak depolanması yerine uzak sunucularda depolanması tercih edilmektedir. Belgelerin uzak sunucularda depolanması verilerin daha verimli kullanımını sağlarken, kişilerin (kurumların) depolama maliyetini de önemli ölçüde azaltır. Veri belgelerinin bu şekilde uzak sunucularda depolanması, e-mail (Gmail, Yahoo Mail), vekil sunucular, CVS vb. gibi bir çok alanda kullanılmaktadır [1].

Veri sahiplerinin bilgi dosyalarını uzak sunucularda depolaması için kullanılan iki senaryo mevcuttur: açık ve özel veri tabanları. Bu iki depolama senaryosu kullanıcılara birbirinden tamamen farklı çözümler ve güvenlik şemaları sunmaktadır. Açık veritabanlarında depolanan veri dosyaları kamuya açık şekilde tutulur, yani herhangi bir kullanıcı ihtiyaç duyduğu veriler için arama sorguları oluşturarak bu verileri elde edebilir. Kullanıcılara veri dosyalarına düşük iletişim yükü ile verimli erişim sağlayan açık veritabanlarında PIR protokolleri kullanılmaktadır [2]. Bu yaklaşımla ilgili daha geniş bilgi aşağıda verilmiştir.

Özel veritabanları, sunucularda depolanan veri dosyalarına sadece veri sahipleri tarafından belirlenmiş ve yetkilendirilmiş kullanıcıların erişimini mümkün kılmaktadır. Bu çalışmada ele alınmış olan bulut sunucularında özel veritabanı senaryosu uygulanmaktadır. Uzak sunuculara aktarılan veri belgeleri bulut servis sağlayıcıları tarafından yönetilen sunucularda tutulur ve servis sağlayıcıları verilerin yetkisiz kullanıcılar tarafından kullanılmasını engellemek, yani aktarılan belgelerin güvenliği ve gizliliğini sağlamaktan sorumludur. Bulut yapısı son dönemlerde bilgi teknolojisinin gelişmesi ve ticari bakımdan etkinlik yönünde yapılan en önemli çalışmalardan biri olarak kabul edilmektedir. Esnek ve ölçeklenebilir yapısı, verilere kişisel ve esnek erişim sağlaması, kullanım kolaylığı, uygulamaya bağlı yazılım sağlaması, mimarinin kullanıcı tarafından seçilmesi vb. bulut yapısının büyük şirketler ve kuruluşlar tarafından tercih edilme nedenlerindendir. “Kullandığın kadarını öde” özelliği bulut yapısının sadece büyük şirketler ve kurumlar tarafından değil, aynı zamanda bireysel kullanıcılar tarafından da tercih edilmesine ve böylece, geniş kitleler arasında yayılmasını sağlamaktadır. Veri sahibi kendi verilerini bulut sunucusuna aktardıktan sonra verileri

sorgulayabilir veya sorgu gerçekleştirmek için istemci kimlik doğrulaması yapabilir. Arama işlemi veri sahipleri ve yetkili kullanıcılar tarafından ihtiyaç duydukları verilere ulaşabilmek için anahtar kelime tabanlı sorgular oluşturarak gerçekleştirilen bir işlemdir. Bulut sunucuları aynı anda çok sayıda kullanıcıya tek merkezli sunuculardan yararlanma imkanı sağlamaktadır. Bulut yapısı veri sahiplerinin, dosyalarına ağ bağlantısı uyumlu herhangi bir cihazdan istedikleri bir yerde erişebilmesini ve değişiklikler yapmasını sağlar [3].

Veri güvenliği, uzak sunuculara aktarılan veri belgelerinin bulut sunucusu veya yetkisiz kullanıcılar tarafından ele geçirilmesinin engellenmesi için geliştirilen güvenlik önemleridir. Veri sahiplerinin belgelerini bulut sunucularına aktarılmadan önce şifrelenmesi de bu güvenlik önlemlerinin en önemlilerinden biridir. Belgelerin şifrelenmesi veri güvenliğini sağlamasına rağmen, veriler üzerinde arama vb. gibi işlemlerin yapılmasını imkansız kılmaktadır. Bu nedenle, son yıllarda veri gizliliği ve arama hızı arasında dengeyi sağlayacak, ayrıca, sunucuya aktarılan belgelerin boyutlarının büyük olması durumları göz önünde bulundurulduğunda, güvenliği sağlarken, anahtar kelimelere dayalı arama ve benzerlik yaklaşımını benimseyen sıralama mantığı ile çalışan efektif aranabilir şifreleme yöntemleri geliştirilmektedir.

Uzak sunuculara aktarılan şifreli veriler üzerinde yapılan arama işlemlerinin genel yapısı şu şekildedir: veri sahibi bulut sunucusuna aktaracağı veri dosyalarından bu dosyaların içeriğine ilişkin anahtar kelimeleri belirler ve veri dosyalarını bulut sunucusuna göndermeden önce şifreler. Ayrıca, son dönemlerde geliştirilen arama yöntemlerinde belgelerin daha güvenli şekilde depolanmasını sağlayan indis yapısı kullanılmaktadır. Veri sahibi dosyalarını şifrelemeden önce, dosyalardan çıkarılan anahtar kelimelerin yanı sıra, bu dosyalar için indisler belirler ve bu indisleri şifreleyerek belgelerle birlikte sunucuya gönderir. Bu indisler güvenli aramalar yapılması ve sonuçların daha hızlı şekilde döndürülmesine yardım edecek dosya ile ilgili bireysel işaretçiler içerir [4].

Ayrıca, veri sahibi tarafından yetkilendirilmeyen kullanıcıların ve bulut sunucusunun dosyalara erişimini engellemek için, sunucu tarafında belgelerin döndürülmesine ilişkin “geçit” oluşturulması yöntemi geliştirilmiştir. Öncelikle, kullanıcı ihtiyaç duyduğu herhangi bir belgeye ulaşmak için bir veya bir kaç anahtar kelimeyi kullanarak sorgu oluşturur ve bulut sunucusuna gönderir. Kullanıcı bu sorguyla birlikte, veri sahibi tarafından gizli şekilde gönderilen veya

kullanıcının kendisinin gizli anahtarını kullanarak oluşturduğu “geçit”i de sunucuya gönderir. Sunucu bu “geçit”i kullanarak gereken belgeleri belirler ve optimum sürede kullanıcıya döndürür. “Geçit” oluşturulurken dikkat edilmesi gereken en önemli nokta, bu “geçit”lerin deterministik değil, bir birinden bağımsız, rastlantısal olarak oluşturulması gerektiğidir.

Bu tez çalışması kapsamında, verilere efektif şekilde erişimi sağlarken, güçlü güvenlik gereksinimleri sunan, indis tabanlı, sıralama mantığını destekleyen bulanık çoklu anahtar kelime ile arama yöntemi geliştirilmiştir. Geliştirilen bu yöntemde, önceki çalışmalarda olan güvenlik açıkları dikkate alınarak, daha güçlü veri güvenliği sağlanması için iki katmanlı Bloom filtresi kullanılmıştır [5]. Ayrıca, sunucuya aktarılan verilerin şifrelenmesi için ORE şifreleme yöntemi kullanılmıştır [6]. Kullanıcı tarafından oluşturulan sorgulara uygun veri dosyalarının döndürülmesini sağlayacak, sorgu ve dosyalar arasında benzerliği belirlemek için term frekansı ve ters belge frekansı (tf - idf) yöntemi kullanılmıştır [7]. Tf - idf benzerlik yöntemi sorgularda kullanılan anahtar kelimeler ve indis işaretçileri arasındaki benzerliği belirlemektedir. Ayrıca, koordinat eşleşmesi [8] yönteminden farklı olarak, sadece tek bir anahtar kelimenin çok sayıda tekrarlandığı dosyalar ve sorgu oluşturulurken kullanılan anahtar kelimelerin hepsini barındıran dosyalar arasındaki farkı da belirlemektedir. Bu özelliği döndürülen dosyaların eşleşme ve önem derecesine göre sıralanmasına olanak sağlamaktadır.

Veri güvenliği, dış kaynağa aktarılan belgelerin barındırdığı verilerin veya bu belgelerle ilgili bilgilerin bulut sunucusunun veya yetkisiz şahısların eline geçmesinin önüne geçilmesi için alınan önlemlerdir. Verilerin bulut sunucularına aktarılmadan önce şifrelenmesi en önemli veri güvenliği önlemlerinden biridir. Şifreleme yöntemleri veri güvenliğini sağlarken, kullanıcıların ihtiyaç duydukları verilere erişim sağlamasında zorluklara neden olur. Bu nedenle, son yıllarda veri gizliliği ve arama hızı arasındaki dengeyi sağlayacak, ayrıca, bulut sunucularına aktarılan verilerin boyutlarının büyük olması göz önünde bulundurulduğunda, güvenliği sağlarken, anahtar kelimelere dayalı ve benzerlik yaklaşımını benimseyerek sıralama mantığı ile çalışan efektif aranabilir şifreleme yöntemleri geliştirilmektedir.

Tezin devamı şu şekilde oluşturulmuştur: Bölüm 2’de bulut bilişiminin genel yapısı ve Grid paradigması ile ortak ve farklı özellikleri anlatılmıştır. Bölüm 3’te şifreli

bulut verileri, güvenlik gereksinimleri ve şifreleme yöntemleri ele alınmıştır. Bölüm 4'de, anahtar kelime ile arama teknikleri, arama algoritmalarının önemli özellikleri, avantajları ve dezavantajları belirtilmiştir. Bölüm 5'de indis oluşturulması ile geliştirilen arama yöntemleri, indis mimarisi, farklı indislerin oluşturulması metotları ve son yıllarda geliştirilmiş yöntemlerden bahsedilmiştir. Bölüm 6'da benzerlik tabanlı sıralı arama yöntemleri, sıralama mantığı, sıralama ölçüsü ve farklı sıralı arama yöntemleri ele alınmıştır. Bölüm 7'de geliştirilen güvenli ve efektif çoklu anahtar kelime ile sıralı arama yöntemi ve temel alınan metotlar anlatılmıştır. Bölüm 8'de yapılan analizler ve testlere dayanan sonuçlar verilmiştir.

2. Bulut Bilişimi

2.1 Genel Tanım

Veri belgelerinin, uygulamaların ve programların sanal bir sunucuda depolanmasını sağlayan ve internet bağlantısı olan her ortamda bu dosyalara ve uygulamalara kolayca ulaşılmasını sağlayan hizmetler bütününe bulut bilişimi veya bulut teknolojisi, diğer bir deyişle, çevrimiçi bilgi dağıtımı denir [1]. Bulut bilişimi kullanıcılara daha fazla depolama alanı, hızlı veri transferi, maliyet tasarrufu, iş gücü tasarrufu yapma olanağı sağlamaktadır. Bu yönüyle bulut bilişimi bir ürün değil bir hizmettir, diğer bir deyişle, temel kaynakların paylaşımını sağlayarak, mevcut bilişim hizmetinin bilgisayar ve diğer aygıtlarda elektrik taşıyıcılarına benzer bir biçimde bilişim ağı üzerinden kullanılmasıdır [8, 9]. Bulut sunucuları internet servisleri için temel internet protokollerine dayanan bir tamamlayıcı, hizmet dağıtıcı olarak tanımlanır. Bulut yapısı donanımsal ve yazılımsal maliyetlerin azaltılması için ağa bağlı altyapı olarak oluşturulmuştur. En çok kullanılan bulut bilişimlerine örnek olarak: Dropbox, Google Drive, SkyDrive, iCloud, YandexDisk, Turkcell Akıllı Bulut, TTNNet Bulut, Ubuntu One gösterilebilir. Ayrıca, Google, IBM, Amazon, Microsoft ve Yahoo gibi büyük yabancı şirketler bulut bilişiminde olan gelişmelere öncülük etmektedir. Myspace, Facebook, Salesforce ve YouTube gibi diğer popüler şirketler de, aynı zamanda, bulut bilişiminde [11, 12] başarı göstermektedir.

Bulut bilişiminin genel yapısı şu kısımlardan oluşur [13]: veri işlemcisi, veri doğrulayıcısı, belirteç jeneratörü ve kimlik doğrulama jeneratörü. Veri işlemcisi, veri sahibi tarafından dış kaynağa aktarılması istenen verileri aktarmadan önce bu

verileri işler. Veri doğrulayıcısı verilerin bozulup bozulmadığını denetler. Belirteç jeneratörü kullanıcı verilerinin segmentlerini almak için bulut depolama sağlayıcılarını etkinleştiren işaretçileri üretir. Kimlik doğrulama jeneratörü ise sistemde taraflara kimlikleri veren bir erişim kontrol politikası uygular. Bulut sisteminin genel yapısı, hizmetleri ve türleri Çizelge 2.1’de verilmiştir.

Bulut bilişiminin en önemli özellikleri aşağıdaki şekilde özetlenebilir:

- Bulut yapısı hızlı kullanım sağlar ve kullanıcıların teknolojik altyapı kaynaklarını tekrar tekrar sağlama kabiliyeti ile gelişir.
- Uygulama Programlama Ara Yüzü (API) bulut yazılımı ile bilgisayarın etkileşime girmesine izin veren, bilgisayarla kullanıcı arasındaki iletişimi sağlayan kullanıcı arayüzü gibi tanımlamalar bütünüdür.
- Bulut bilişiminin kullanımı daha düşük masraflıdır, genel bulut dağıtım modelinde sermaye masrafları işletimsel masrafa dönüştürülmüştür. Ayrıca, altyapı düşük masraflı yerlerde merkezleştirilir (gayrimenkul, elektrik gibi).
- Profesyonel olmayan insanlar tarafından bile rahatlıkla idare edilebilir.
- Cihaz ve yerden bağımsız olması kullanıcılara bir web tarayıcısı kullanarak bulundukları yer ve kullanılan cihazdan bağımsız olarak sistemlere erişebilmelerini sağlar.
- Sanallaştırma teknolojisi sunucuların, cihaz paylaşımının ve kullanımının artırılmasına izin verir.
- Bulut bilişimi kullanılabilecek veri depolama kapasitesini çok fazla genişletmiştir.
- Sistemlerden yararlanma ve verimlilik çoğunlukla % 10 – 20 civarında artış göstermiştir. Bunun yanı sıra, çok sayıda yedek alanlar kullanılması, güvenliği geliştirmiştir.
- Bulut bilişimi çok sayıda kullanıcının kaynaklarını ve masraflarını tek bir kullanıcı havuzunda tutmaktadır.

Bulut bilişiminin ölçeklenebilir ve esnek olması, dinamik (talep üzerine) kaynakların hazırlığı sayesinde detaylı, kendi kendine hizmet temelli, gerçek zamanlıya yakın hizmet sunmasını sağlar. Bulut için önemli olan unsurlardan biri acil ölçeklenebilirlik ve kaynak kullanım optimizasyonudur. Bu özellikler dinamik ortamda kaynak yönetim otomasyonu ile sağlanır. Bulut bilişiminde verilerin depolanma zamanı, çok sayıda yedek alanın kullanılması ve verilerin

merkezleştirilmesi güvenliğini artırır. Ancak verilerin merkezleştirilmesi bazı veriler üzerinde kontrol kaybının olmasına ve depolama zamanı sunucu ve üçüncü tarafa karşı yeterli veri güvenliğinin sağlanamamasına neden olur.

2.2 Bulut Bilişiminin Kısa Tarihçesi

Bulut bilişimi ile ilgili ilk kavramlara 1950 yıllarında rastlanmıştır. Akademik araştırma merkezlerinde ve büyük kuruluşlarda büyük boyutlu ana bilgisayar elde etmek için daha zayıf terminal bilgisayarlar kullanılmıştır. Bu sistemler bilgisayarların boş kalma süresini elimine ederek, çok maliyetli olan ana bilgisayarların en iyi şekilde kullanılmasını sağlamıştır. Bu çalışma şekli zaman paylaşımı olarak bilinmekteydi ve kullanıcıların bilgisayarları fiziksel erişimde ve CPU kullanımında paylaşmasını sağlamaktaydı [14]. 1960'larda John McCarthy bilgi işlemin kamu hizmetleri şeklinde düzenlenebileceği fikrini belirtmiştir. 1966 yılında Douglas Parkhill mevcut bulut yapısının esnek hizmet sağlama, çevrimiçi ağ üzerinde hizmet, sınırsız kaynak gibi neredeyse tüm özelliklerini genel, özel, kamu ve topluluklara aynı kullanım şeklinde incelemiştir. GE'nin GEISCO, IBM subsidiary the service bureau Corporation, Tymshare, National CSS, Dial Data, Bolt, Beranek and Newman gibi kuruluşlar güçlü bilgisayarların pahalı olması nedeniyle, kendi bilgisayarlarının zaman paylaşımı ve diğer düzenlemelerini pazarlama girişimi olarak sunmuşlardı. 1990'lı yıllarda telekomünikasyon şirketleri sanal ağ hizmetleri vermeye başlamış ve az masrafla daha kaliteli hizmetler vermeyi başarmıştı. Bant genişliğinin daha verimli şekilde kullanılması için anahtarlama yöntemi kullanılmıştır.

Bilgisayar ağlarının kapasitesinin sadece %10'nun kullanıldığı dönemde, Amazon kullanılmayan kaynakları bilgi işlem hizmeti olarak sunarak bulut bilişimi yapısının gelişmesinde anahtar rol oynamıştır. Amazon'un aktif ekipleri dışarıya pazarlamak niyetiyle yeni ürünler geliştirmeye başlamış ve 2001 yılında Amazon Web Hizmetlerini (AWS) piyasaya sürmüştür. 2006 yılında ürüne özel geliştirilmiş fiyatlandırma modeli ile ilk bulut bilişimi olan Amazon S3, kullanıcıların hizmetine sunulmuştur. Bundan başka, aynı yılda IIRMO Avrupa Komisyonu tarafından desteklenen proje kapsamında veri aktarım hızı, paket kaybolma olasılığı, paket gecikme süresi değişimi (jitter), bit hata oranı gibi konularda bulut yapısının geliştirilmesine odaklanılmış ve gerçek zamanlı bulut ortamı geliştirilmiştir [15].

Gartner 2008 yılında kullanıcılar ve tedarikçiler arasındaki ilişkinin değiştirebileceği fikrini ileri sürmüştür. Hizmete sunulan ilk bulut yapıları sadece yedekleme amacıyla kullanılmaktaydı. Amazon Web Hizmetleri - API uyumlu platform 2008 yılında sunulmuş Eucalyptus olmuştur. Aynı yılda özel, hibrid ve federasyon bulutlarını uygulayan ilk açık kaynak yazılım Avrupa komisyonu tarafından desteklenen RESERVOIR projesi – OpenNebula olmuştur [15].

Türkiye’de hizmet veren TTNET internet sağlayıcısı kendi bulut hizmetini geliştirmiş ve TTNET Bulutu olarak Türkiye piyasasına sunmuştur [16].

2.3 Bulut Modelleri

Bulut bilişiminin aşağıdaki gibi farklı modelleri geliştirilmiştir [17]:

- *Genel Bulut:* bu bulut modelinde sistemin tamamı bulut hizmetlerinden faydalanarak oluşturulur, diğer bir deyişle, kullanılacak yapı üçüncü bir şirket tarafından sağlanan kaynaklar üzerine kurulur. Bu tür sistemlere örnek olarak, kullanıcılar tarafından ödeme yapılmadan çeşitli özellikleri kullanılan elektronik postaları gösterilebilir. Genel bulut bir hizmet sağlayıcısı kullanarak depolama ve diğer kaynak hizmetlerini sağlar. Bu model ücretsiz erişimlidir ve ücretlendirme şekli olarak “kullandığın kadar öde” modelini kullanır. Amazon AWS, Google, Microsoft kendi bulut sağlayıcılarını kullanmaktadır.
- *Özel Bulut:* büyük kurumlar, şirketler ve önemli bilgilere sahip kuruluşlar tarafından kurulan bulut teknolojisidir. Örnek olarak, Microsoft şirketinin sunduğu Hyper-V ve SystemCenter ürünleri gösterilebilir. Bu şekilde olan bulut yapısı sadece tek bir organizasyon tarafından kullanılmaya uygun olarak inşa edilir.
- *Hibrit Bulut:* bu bulut modeli genel ve özel bulut yapılarının birleşmesinden ortaya çıkan altyapıdır. Veri güvenliği ve gizliliğinin ön plana çıktığı durumlarda özel bulut yapısı, diğer durumlarda ise genel bulut yapısı kullanılır. Bu bulut yapılarını tercih eden şirketlerin hacmine göre kullanılan bulut yapısı değişmektedir. Hata tolerans derecesini yükseltmiştir. Bu bulut yapısında kurum içi uygulamaların esnek, güvenli ve belirli olması gerekmektedir.
- *Topluluk Bulutu* diğer bulut modellerine göre daha az tercih edilen bir yapıya sahiptir. Bu bulut yapısına aktarılan veri dosyaları bir kaç kullanıcı

tarafından ortak olarak kullanılmaktadır. Daha düşük masraflı bir bulut yapısıdır ve kullanıcılar tarafından dahili veya harici barındırmasına olanak sağlar.

Şekil 2.1’de bulut bilişiminin bileşenleri, dağıtım modelleri ve hizmet modelleri verilmiştir.

Katmanlar	Bulut Bilişimi Bileşenleri		
Özellikler	Talep üzerine kendi kendine hizmet	Geniş ağ erişimi	
	Kaynak Havuzu	Hızlı Elastikiyet	
	Ölçülebilir Hizmet		
Temel Hizmetler	IaaS	SaaS	PaaS
Dağıtım Modelleri	Açık	Özel	Topluluk
		Hibrid	

Şekil 2.1. Bulut bilişimi ve hizmetleri

2.4 Bulut Hizmetleri

Bulut bilişimi sağlayıcıları üç temel hizmet sunmaktadır. Bu hizmetler aşağıdaki gibi ifade edilebilir: Altyapı hizmeti (IaaS), Yazılım hizmeti (SaaS) ve Platform altyapısı (PaaS) [17].

- Altyapı hizmeti (IaaS) – bu hizmet bulut bilişiminin kullanıcılara sağladığı en temel hizmettir. Bulut sağlayıcıları sunucularını fiziksel veya sanal makineler olarak sunmaktadır. IaaS bulut dahilinde bulunan depolama, güvenlik duvarları, yük dengeleyiciler, IP adresleri, sanal yerel alan ağları

(wlan) ve yazılım demetlerinden oluşur. IaaS bulut sağlayıcıları (Amies, Alex, SuimanHarm, Tang) veri merkezinde olan büyük bilgi teknolojileri havuzunda kullanıcılara talep üzerine kaynaklar sağlar, kullanıcılara sistem ve yazılım onarımı verir. Bu hizmete ait ücret, ayrılan ve tüketilen kaynak miktarına göre belirlenir [18]. IaaS bulut hizmetinin önemli özellikleri aşağıdaki şekilde özetlenebilir:

- Esneklik
- Risklerin aktarılması
- Azaltılmış işletim maliyetleri
- En son altyapı durumu
- İşbirliği ile çalışma
- Olağanüstü durumda kurtulma

Bulut sunucularında kullanıcılara altyapı hizmetlerinin sağlanmasında bazı önemli zorluklarla karşılaşılır:

- Bulut sunucuları belirli sıcaklıkta muhafaza edilmelidir.
- Sistemin elektrik kesilmelerine karşı dayanıklı olması gerekmektedir.
- Altyapı donanımının seçimi çok önemlidir.
- Bulut ve donanım arasındaki bağlantı yüksek bant genişlikli kanal ile olmalıdır.
- Bulut depolamasının büyük veri boyutlarının değişen taleplerini karşılaması gerekmektedir.
- Kontrol kaybı gözlemlenebilir.

b) Platform hizmeti (PaaS) – bulut sağlayıcıları işletim sistemi, uygulama ortamı, veri tabanı ve web sunucuları içeren, bilgi işleme platformu sağlamaktadır. Araştırmacılar ve uygulama geliştirmek isteyen kullanıcılar bir bulut platformu kullanarak kendi yazılımlarını masraflar, donanımsal ve yazılımsal zorluklar olmadan geliştirebilir ve çalıştırabilirler. Ayrıca, bulut bilişimi mevcut kaynakların ölçeklendirilmesine izin vermektedir. Platform hizmetlerinin önemli özellikleri şunlardır:

- Üst düzey altyapı erişimi
- Esneklik
- Hizmetlerin kullanımı için hazırlık
- Ölçeklenebilirlik

- Daha düşük yönetim maliyeti

Platform hizmetinin sağlanmasında gözlemlenen zorluklar aşağıdaki gibi özetlenebilir:

- Sınırlı API'ler
- Verilerin dahili kilitlenmesi
- Denetlenebilirlik
- Öngörülemez performans
- Düşük seviyeli güvenlik üzerinde kontrol eksikliği
- Uygulamalar arasında veri erişimsizliği
- Web uygulamaları ve SOA güvenlik açıkları

Bulut bilişiminin platform hizmet sağlayıcılarına örnek olarak Amazon Elastic Beanstalk Cloud Foundry, IBM Bluemix, Google Engine App, Mendix, Ms Azure vb. gösterilebilir.

c) Yazılım hizmeti (SaaS) – Bulut bilişimi, kullanıcılara uygulamaların yüklemesi ve bunların çalıştırılmasına izin verir, kullanıcılar bulut hizmetlerini kullanarak ihtiyaç duydukları yazılıma erişebilir. Kullanıcılar, üzerinde uygulamayı çalıştırdıkları platformları yönetmemektedir, bu nedenle uygulamanın kullanıcıların şahsi bilgisayarlarında kurulu olması gerekmez. Esnek yapısı, değişen iş taleplerinin çok sayıda sanal makineler üzerinde klonlamasını sağlamaktadır. Taleplerin sanal makineler üzerinde dağıtılması yük dengeleyicileri tarafından dengelenir. SaaS hizmeti aşağıda belirtilen önemli özelliklere sahiptir:

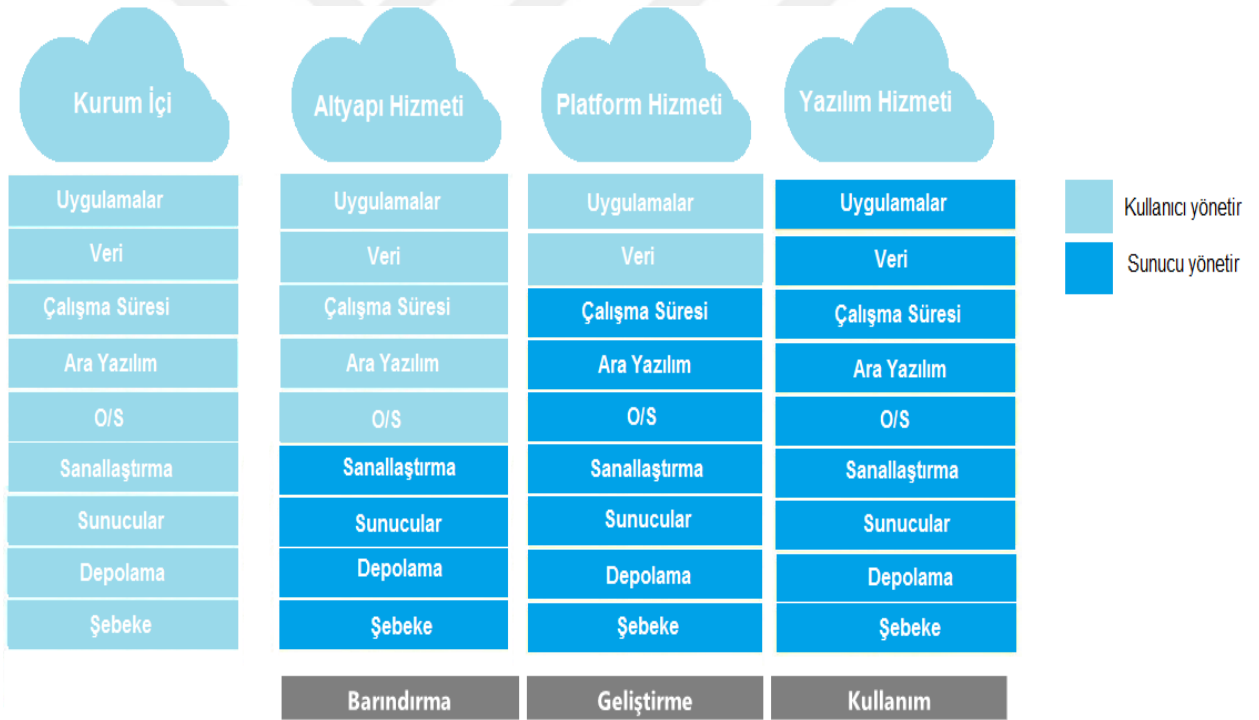
- Maliyetin azaltılması
- Yüksek verimlilik
- Zaman tasarrufu
- Yüksek teknoloji hizmetlerinin mevcut olması
- Yüksek kullanılabilirlik
- Azaltılmış yönetim maliyeti

Yazılım hizmetinin sunulmasında aşağıda belirtilen zorluklar gözlemlenmektedir:

- Veri güvenliğinin en üst düzeyde sağlanması
- Yüksek kullanılabilirlik gereksinimi
- Kimlik doğrulaması ve yetkilendirme
- Veri bütünlüğü

- Veri gizliliği
- Ağ güvenliği
- Bulut standardizasyon
- Farklı ülke sonuçlarında kuralların çatışması durumunda bulut kaynaklarının dağıtımı
- Veri yedekleme
- Web uygulama güvenliği
- Veri gizliliği
- Sanallaştırma

SaaS hizmetinin uygulamaları fiyatlandırma modeli genellikle aylık ve yıllık bazda her kullanıcıya fiyatlandırma modelidir. Bulut bilişiminin SaaS hizmetlerine örnek olarak, Google Apps, İnkeypas, Quick book online, Ms Office 365, Salesforce.com vb. gösterilebilir.



Şekil 2.2. Bulut hizmetleri yönetimi

2.5 Bulut Bilişimi ve Grid Paradigması

Günümüzde her geçen gün kullanımı daha geniş kitleler arasında yayılan bulut bilişimi bazı durumlarda benzer özelliklere sahip Grid paradigması ile karıştırılmaktadır [19]. Ancak bulut bilişimi donanımsal ve yazılımsal maliyetlerin azaltılması için ağa bağlı altyapı şeklinde oluşturulurken, Grid paradigması

bilgisayarların hesaplama ve veri depolama kapasitesini internet üzerinde paylaşılarak gittikçe daha performanslı hale getiren servis hizmeti olarak tanımlanır ve ağına bağlanan her bir kullanıcı tarafından kullanılabilir. Ayrıca, bulut bilişimi internet merkezli olup kullanıcılara kullanım kolaylığı, sanallaşma, kaynak farklılığı, otomatik adaptasyon, ölçeklenebilirlik, kaynak optimizasyonu, SLA altyapısı ve hizmeti sunarken, Grid paradigması genel amaçlı, açık protokoller ve arayüzler kullanarak kaba hizmet niteliklerini sunmak için açık, merkezi kontrole tabi olmayan kaynakları koordine etme sistemidir. Bu sistem farklı kaynakları ortak amaç için birleştirme yeteneğine sahiptir.

Bulut bilişiminde, sanallaşma izolyasyonu nedeniyle hiçbir gerçek paylaşım mevcut değildir, sunuculara aktarılan kaynaklar paylaşılmamaktadır. Grid paradigması hizmet kaynaklarının heterojenliğinin gizliliğini arayüzlerle sağlar. Bu nedenle, onun sunduğu tekil geniş alan kaynak havuzunda bölümlerin toplanmasının sanallaşma yeteneğini sağlar. Ayrıca, sanallaştırma yeteneği hem veri hem de bilgisayar kaynakları için sağlar. Bulut bilişiminde ise donanım ve yazılım kaynaklarının sanallaştırılmasının yanı sıra, izolyasyon sağlandığı için sanallaşma güvenliği de sağlamaktadır [12].

Bulut hizmetleri her kullanıcı için tekil bir erişim ve her kullanıcıya özel bireysel sanallaşmış bir ortam sağlamaktadır. Grid paradigması ise genellikle son kullanıcı güvenliği ile ilgilenmemektedir, hizmet ve iş odaklıdır. Bu paradigmaya hizmet akışı ve konum koordinasyonu için ihtiyaç duyulur. Uygulamalar hizmetlerin önceden tanımlanmış iş akışını gerektirir. Bulut bilişimi kullanıcı isteğine bağlı olarak dağıtım hizmeti sunmaktadır. Kullanılan çok sayıda uygulama için iş akışı gerekli değildir.

Bundan başka, Grid paradigması daha çok kamu finansmanını temel almaktadır, yani verilen her bir hizmet için sabit bir oranda ücret belirlenir veya farklı kuruluşlarda kaynakların paylaşılması faturalanır. Bunun aksine, bulut bilişimi kullanıcılara “kullandığın kadar öde” modeli sunarak, sadece kullanılan hizmetleri faturalamaktadır. Ayrıca, bulut sunucuları eşit dağıtımı sağlar ve kullanıcının kendi kullanacağı mimariyi seçmesine izin verir. Grid paradigmasının en önemli dezavantajı, sadece “gridified” adı verilen özel uygulamaların kullanılmasını gerektirmesidir ki, bu da, geliştiricilere sert sınırlar koyar. Grid paradigmasının

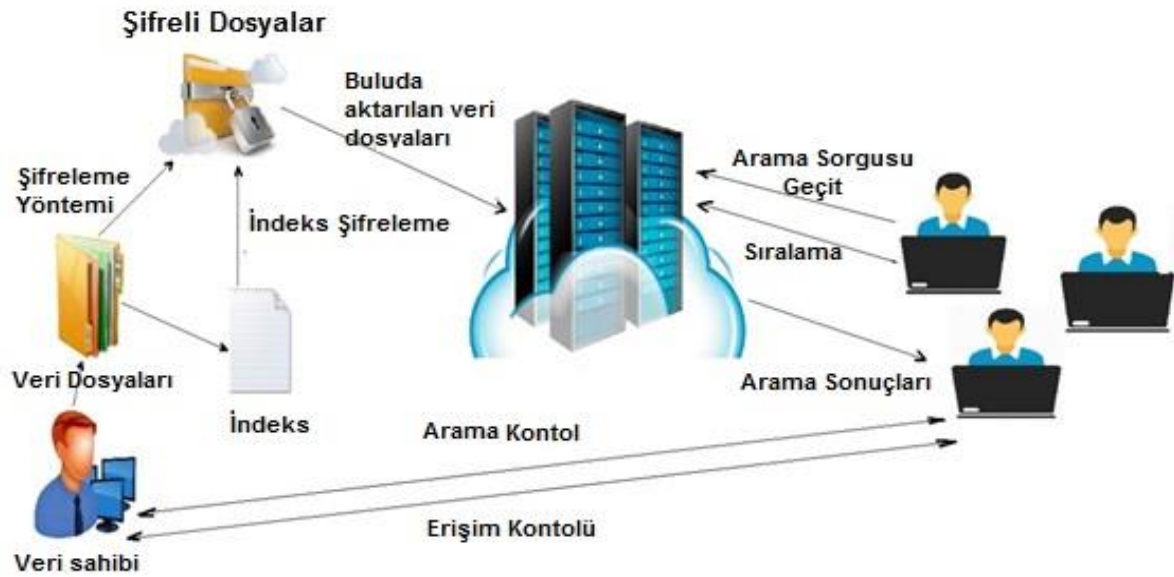
mimarisi servis odaklıdır [19]. Bulut bilişimi ve Grid paradigmasının önemli özellikleri Çizelge 2.1’de gösterilmiştir.

Özellikler	Grid paradigması	Bulut Bilişimi
Kaynak paylaşımı	İşbirliği (adil paylaşım)	Aktarılan kaynaklar paylaşılmaz
Kaynak heterojenliği	Heterojen kaynakların toplanması	Heterojen kaynakların toplanması
Sanallaştırma	Veri ve bilgi işlem kaynaklarının sanallaştırılması	Donanım ve yazılım platformunun sanallaştırılması
Güvenlik	Kimlik heyet aracılığıyla güvenlik	İzolyasyon yoluyla güvenlik
Yüksek derece hizmet	Üst düzey hizmetler	Henüz mevcut değil
Arxitektura	Servis odaklı	Kullanıcı mimariyi seçer
Yazılım bağımlılığı	Uygulama bağımlı yazılım sağlar	Uygulama bağımlı yazılım sağlar
Platform farkındalığı	Kullanıcı yazılımı Gridnabile olmalı	SP yazılımı özelleştirilmiş bir ortamda çalışır
Yazılım iş akışı	Uygulamalar hizmetlerin önceden tanımlanmış iş akışını gerektirir.	İş akışı çoğu uygulama için gerekli değildir
Ölçeklenebilirlik	Düğüm ve siteler ölçeklenebilir.	Düğüm, siteler ve donanım ölçeklenebilir.
Kendini yönetme	Yeniden konfigüre edilebilir	Yeniden configure edilebilir ve kendini düzeltebilir.
Merkezileştirme yönetimi	Dağınık kontrol	Merkezli kontrol
Kullanılabilirlik	Yönetimi zordur	Kullanıcı dostu
Standartlaştırma	Standartlaştırma ve birlikte çalışabilirlik	Bulutta birlikte çalışmak için standart elksikliği vardır.
Kullanıcı erişimi	Son kullanıcı için erişim şeffaflığı	Son kullanıcı için erişim şeffaflığı
Ödeme modeli	Esnek değil	Esnek
Qos teminatları	Sınırlı destek, daha çok deneme (çaba)	Sınırlı destek, kullanılabilirlik ve çalışma süresi üzerinde durulur.

Çizelge 2.1. Bulut bilişimi ve Grid paradigmasının karşılaştırması

3. Şifreli Bulut Verileri: Güvenlik

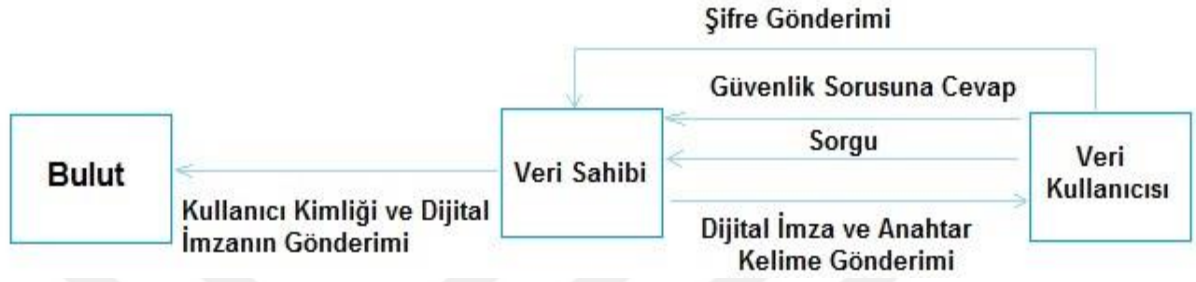
Veri dosyalarının bulut sunucularında depolanması ve kullanıcılar tarafından kullanılması süreci üç taraf arasında gerçekleştirilmektedir: veri sahibi, bulut sunucusu ve kullanıcı (istemci). Veri sahipleri için en önemli konu PT veri dosyalarının sunucuda tutulurken güvenliği ve gizliliğinin sağlanması, veriler yetkili kullanıcılar tarafından kullanılırken bilgi sızıntısı olmamasıdır. Bulut sunucular “dürüst – ama – meraklı” olarak kabul edilir, yani kullanıcılar tarafından oluşturulan sorgulara uygun sonuçlar döndürür, ancak yapılan aramalara ve oluşturulan sorgulara uygun istatistiksel analizler de gerçekleştirilebilir. Bu nedenle veri sahipleri dosyalarını bulut sunucularına aktarmadan önce belirli şifreleme yöntemleri kullanarak şifrelemektedir. Ancak dosyaların şifrenmesi düz metinler üzerinde gerçekleştirilen arama gibi işlemleri imkansız kılmaktadır. Şekil 3.1’de bulut sunucusuna aktarılan veriler üzerinde yapılan arama prosesinin genel yapısı gösterilmektedir.



Şekil 3.1. Şifreli bulut verileri üzerinde arama

Kullanıcıların bulut sunucularını efektif şekilde kullanabilmesi için depolanan verilere hızlı şekilde erişebilmeleri ve gereken dosyaları kullanabilmek için arama yapabilmeleri gerekmektedir. Veri sahibi bulut sunucusu ile paylaşmadan önce PT veri dosyaları için l indisi inşa eder, PT veri dosyaları şifreler. Bulut sunucusunda depolanan dosyaların verimli ve efektif kullanımı için dosyaların şifrenmesinde aranabilir şifreleme algoritmaları kullanılmaktadır. Ayrıca, PT veri dosyaları

üzerinde arama yapılabilmesi için önceden oluşturulan I indisleri de şifrelenir ve C şifreli dosyalarını ve şifrelenmiş I indisi bulut sunucusuna gönderilir. Daha sonra yetkili kullanıcı, veri sahibi tarafından ona verilen anahtar kelimeyi kullanarak sorgu oluşturur ve bu sorgu ile birlikte veri sahibi tarafından oluşturulan ve arama kontrol mekanizmaları tarafından istemciye gönderilen T “geçidini” bulut sunucusuna gönderir.



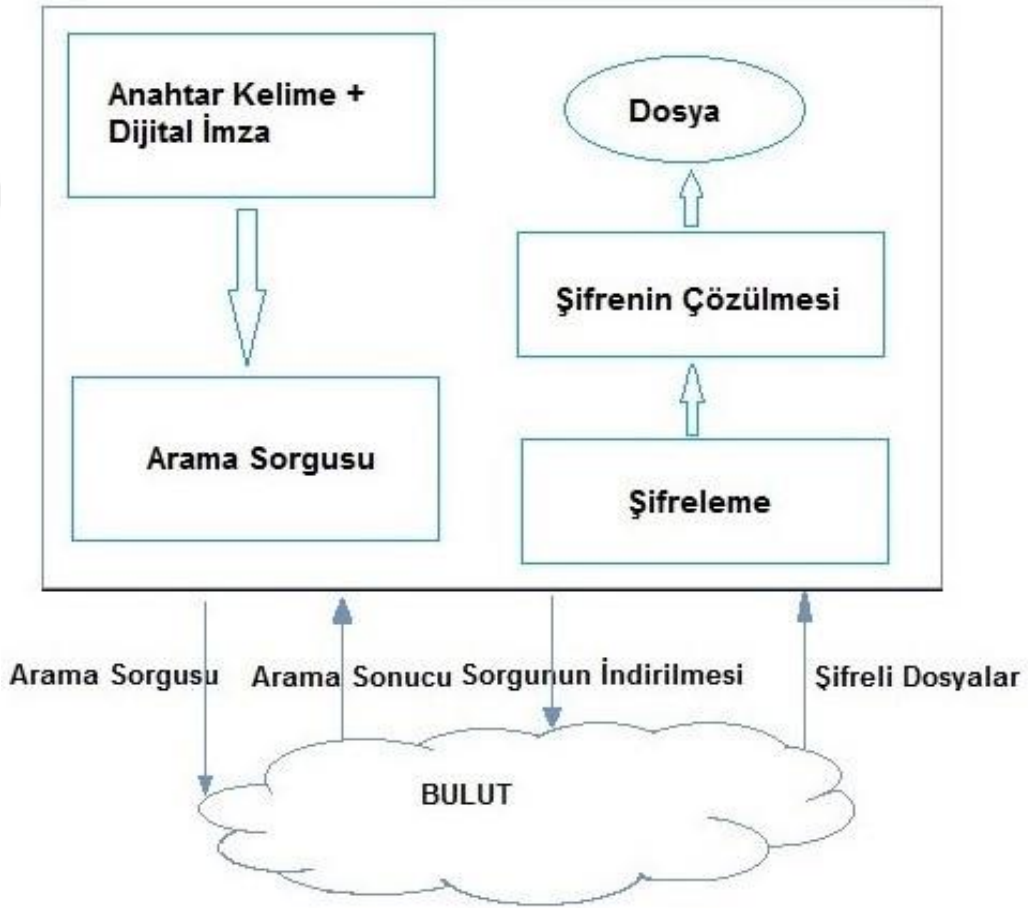
Şekil 3.2. Kimlik Doğrulama

Ayrıca, sıralama yeteneğine sahip arama sistemleri için, döndürülen dosya sayısına sınırlamalar da konabilir. Bulut sunucu T “geçidini” aldıktan sonra buna uygun I indisini arayarak bu indise uygun dosyaları döndürmektedir.

3.1. İlgili Çalışmalar

Bulut sunucularında depolanan şifreli bulut verileri üzerinde anahtar kelimeler ile arama yöntemi ilk kez Song ve ark. [4] tarafından sunulmuştur. Yapılan çalışmalara düz metin şeklinde dış kaynağa aktarılan verilerin güvenlik açısından yarattıkları endişelerden yola çıkarak başlanmış ve güvenlik risklerini azaltmak için dosyaların sunucuda paylaşılmadan önce şifrelenmesi önerilmiştir. Geliştirilen yöntem güvenilmeyen uzak sunucuya aktarılan herhangi bir hassas bilgi sızıntısı olmadan şifrelenmiş veriler üzerinde indisli arama yapılmasını sağlamaktadır. Boneh ve ark. [20] asimetrik şifreleme yöntemlerini kullanarak ilk kez aranabilir şifreleme yöntemini geliştirmiştir. Bu yöntem, ortak anahtara sahip her kullanıcının bulut sunucusunda depolanan veri dosyalarına yazabilmesini sağlamaktadır. Dosyalar üzerinde arama yapma yetkisi ise ancak özel anahtara sahip kullanıcılara verilmiştir. Başka bir asimetrik şifreleme tabanlı algoritma, Di Crescenzo ve ark. [21] tarafından sunulmuştur. Araştırmacılar açık anahtarlı şifreleme ile anahtar kelime arama düzeni önermiştir. Buraya kadar tüm araştırmalar destekledikleri tam anahtar kelime arama bağlamında bulunan

güvenli dizin tabanlı algoritmaların kullanımları ile sınırlanmıştır. Goh [22], ilk kez Bloom filtresine dayalı indisli arama yöntemini geliştirdi. Bu yöntem, üzerinde arama yapılan dosyaların sayısı ile orantılı olarak yapılan arama maliyetini azaltma imkanı sağlamaktadır. Moataz ve ark. [23] çalışmalarında Boolean simetrik aranabilir şifrelemeye dayalı arama yöntemini önermiştir. Geliştirilen arama yöntemi Gram-Schmidt işlemine uygun olarak anahtar kelimelerin ortogonalliğine dayanmaktadır.



Şekil 3.3. Veri erişimi

Örencik ve ark. [24] tarafından sunulan şifreli bulut veriler üzerinde arama yöntemi veri gizliliğini temin ederken çoklu anahtar kelime ile arama yapılmasını sağlamaktadır ve bunun için minhash fonksiyonları kullanılmıştır. Wang ve ark. [25] tarafından şifreli bulut verileri üzerinde güvenli sıralı anahtar kelime arama sorunu araştırılmaktadır. Bu çalışmada, şifreli belgeler için aranabilir indis oluşturulurken her belgenin alaka puanını gömen istatistiksel bir yaklaşım geliştirilmiştir. Cao et al. [8] çoklu anahtar kelime arama sorgusu ve veri belgeler arasında benzerliği belirlemek için "koordinat eşleştirmesi" ilkesini kullanarak ilk

kez çoklu anahtar kelime ile sıralı arama düzenini sunmuştur. Bu çalışma şifreli bulut verileri üzerinde güvenli arama yöntemlerinde çok önemli yere sahip olan bir araştırmadır. Ancak, bu algoritma indis yapısının belge terimlerinin bir ikili vektör şeklinde gösterimini kullanmıştır ve bu nedenle, sıralı arama sonuçlarını döndürürken, anahtar kelimelerin sadece bir kısmının tekrarlandığı belgeler ile kullanılan tüm anahtar kelimelerin tekrarlandığı belgeler arasında ayırım yapmamaktadır.

3.2. Veri Güvenliği

3.2.1 Güvenlik ve Gizlilik Koşulları

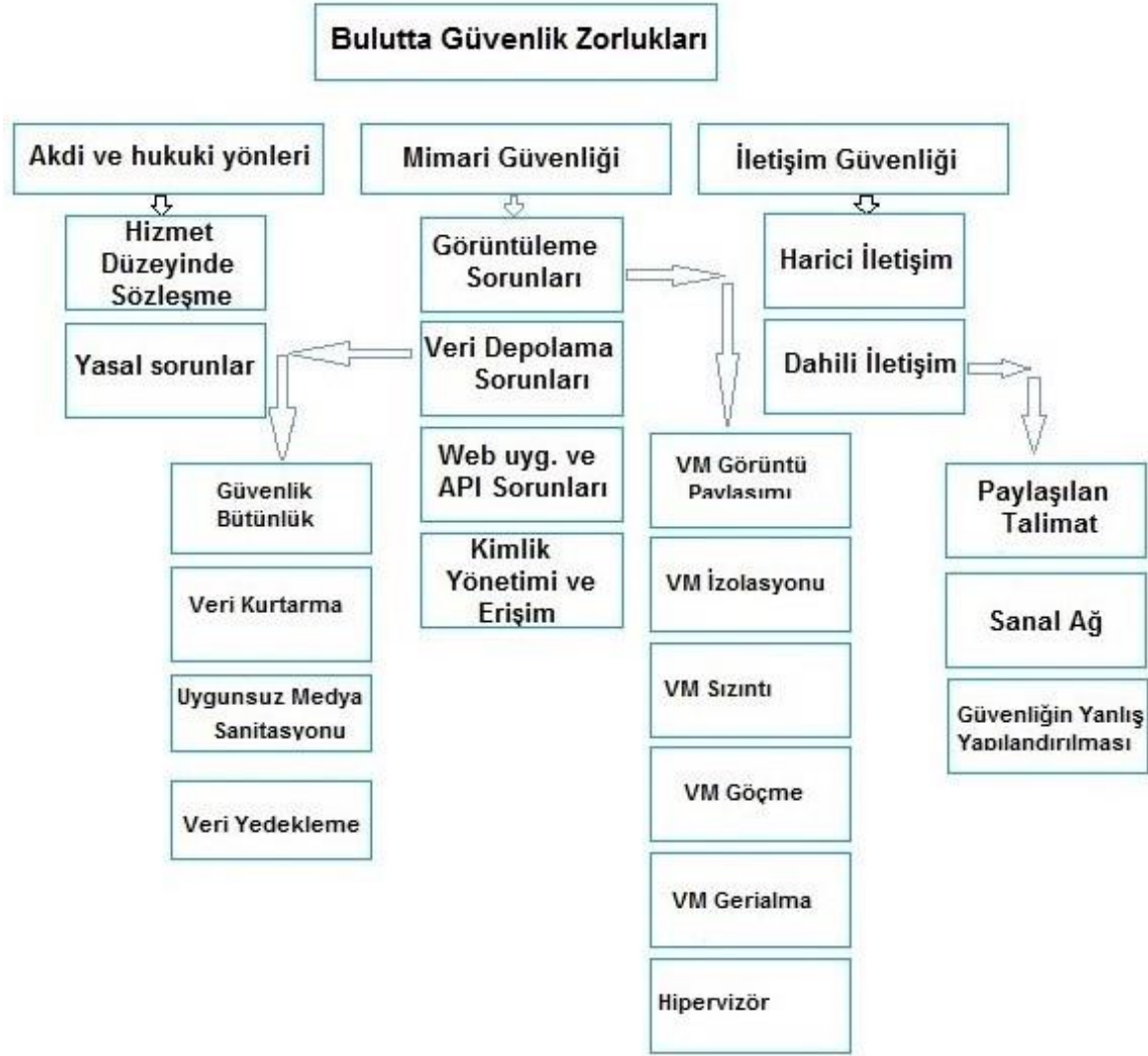
Bulut sunucusu kaynak havuzuna aktardıkları verilerin güvenliği ve gizliliği kullanıcıların endişesine neden olan en önemli konulardır. Şifreli bulut verileri üzerinde yapılan araştırmalarda, bulut sunucu “meraklı ve dürüst” olarak kabul edilir [26, 27]. Daha açıklayıcı şekilde, bulut sunucusu gönderilen dosyalarda ve belirlenen sorgularda “dürüst” davranır ve belirlenmiş protokolu izler. Ancak sunucu işlemler sırasında “meraklı” davranır ve istatistiksel şekilde içeriklere ulaşmayı dener veya ek bilgi almak için analiz ve çıkarım yapar. Ayrıca, kötü niyetli üçüncü taraflar sahip oldukları gizli anahtarları kullanarak yetkileri dışında bilgilere ulaşmaya çalışabilmektedir. Şekil 3.4’de bulutta güvenlik zorlukları verilmiştir.

Bulut sunucusunun güvenlik için şifrelenen anahtar kelime ve şifreli belgeler arasındaki ilişkiyi anlaması durumunda, belgelerin içeriğini öğrenmesi muhtemeldir. Bu nedenle aranabilir indisler bu tür saldırıları önleyecek şekilde inşa edilmelidir.

Anahtar kelimelerin gizliliği geçitlerin kriptografik şekilde oluşturulması ile elde edilebilmektedir. Bulut sunucu belgeler üzerinde bazı istatistiksel analizler yapabilir ve sonuç olarak önceden yapılmış aramalar ve döndürülen sonuçlara dayalı bilgiler elde edebilir. Sunucu veya kötü niyetli üçüncü taraflar elde edilen bilgilere dayanarak yapacağı saldırılara Şekil 3.4’te olduğu gibi örnekler gösterilebilir.

Yetkisiz üçüncü taraflar veya bulut sunucusu aşağıda gösterilen saldırı yöntemlerini kullanarak kullanıcıların (veri sahibi) bulut sunucularında depoladığı veri dosyalarına erişim elde edebilir [28]:

- *Bilinen arkaplan modeli:* Belirtilen bu saldırı şekli güçlü bir saldırı modelidir. Bulut sunucusunun daha fazla bilgiye sahip olması gerekmektedir ve sunucu veri dosyalarıyla ilgili istatistiksel bilgileri ve verilen arama isteklerinin geçit korelasyon ilişkisini içerir. Bu tür saldırılarda, sunucu oluşturulan sorguda kullanılan belirli anahtar kelimeleri belirlemek için, anahtarların veya belgelerin sıklığı ile bilinen geçit bilgilerini kullanabilir.

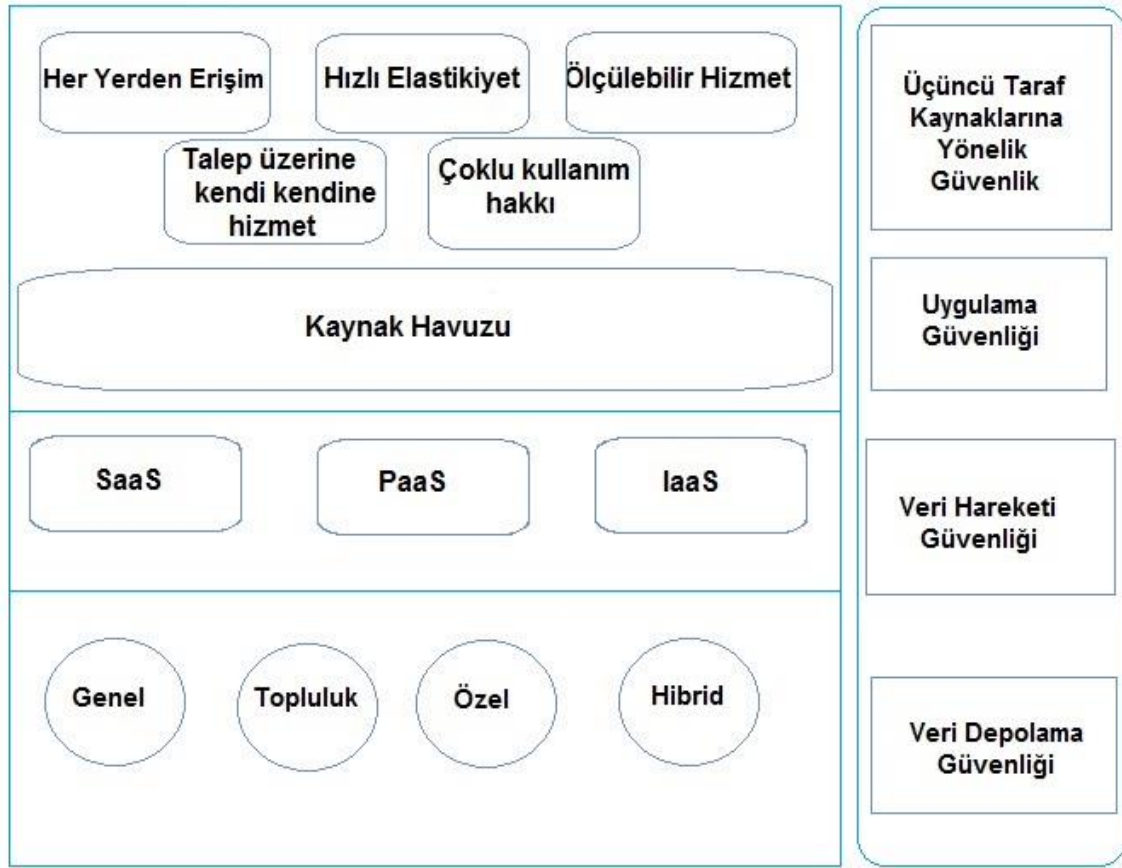


Şekil 3.4. Bulut bilişimde güvenlik zorlukları

- *Bilinen şifreli metin modeli:* Bulut sunucusu veri sahibi tarafından dış kaynağa aktarılan C şifreli metnini ve I aranabilir indisini kullanarak paylaşılan veri dosyalarının içeriğine ulaşabilir.
- *Geçit bağımsızlığı:* Birbiri ile ilişkili geçitlerin kullanılması veri güvenliği ve gizliliğini bozmaktadır ve verilerin istatistiksel saldırılara karşı zayıf olmasını sağlar. Kullanıcıların ihtiyaç duydukları sonuçları elde etmeleri için veri

sahipleri tarafından oluşturularak onlara gönderilen veya kendi gizli anahtarları ile oluşturdukları geçitler kullanılmaktadır.

Geçitlerin oluşturulması için iki farklı yöntem kullanılmaktadır: deterministik ve deterministik olmayan, yani rastlantısal. Deterministik şekilde oluşturulan geçitler farklı anahtar kelimeler ile ilgili farklı arama frekanslarını biriktirir ve bu da sonuçta, anahtar kelime gizliliğini ihlal eder. Rastgele geçitler ise herhangi iki geçit arasındaki ilişkinin belirlenmesini önler. Bu nedenle güvenli arama yöntemleri rastgele oluşturulan geçitler yöntemini tercih etmektedir.



Şekil 3.5. Bulut bilişimi güvenlik gereksinimleri

Bulut bilişimi sağladığı hizmetlere bağlı olarak farklı seviyelerde güvenlik gereksinimleri gerektirmektedir. Bunlardan yukarıda anlatıldığı gibi, üçüncü taraf veya sunuculara karşı güvenlik gereksinimleridir. Diğer bir güvenlik gereksinimi uygulama güvenliğidir ki bu, bulut bilişiminin sağladığı platform ve uygulamaların kullanılması ve kullanıcıların hangi uygulamaları kullandığının güvenliği ve gizliliğinin sağlanmasını kapsamaktadır. En önemli güvenlik gereksinimleri veri hareketleri ve veri depolaması için gereken güvenlik gereksinimleridir. Bu güvenlik

gereksinimleri aşağıdaki bölümlerde geniş şekilde ele alınmıştır. Şekil 3.6'da güvenlik ilkeleri verilmektedir.



Şekil 3.6. Genel güvenlik ilkeleri

3.2.2. Aranabilir Şifreleme

Veri güvenliği gizlilik, kullanılabilirlik ve bütünlük gibi üç önemli hedefi içermektedir. Verilerin güvenliğini ve bilgi gizliliği sağlamak için kimlik ve veri kaynağı doğrulaması gibi veri güvenliği yöntemleri kullanılmaktaydı. Bulut sunucularında depolanan veri gizliliğinin şifreleme yöntemleriyle sağlanması daha iyi sonuçlar vermektedir. İletim ya da depolama esnasında metin, resim, ses, video gibi veri içeriğinin anlamsız, okunmaz veya görünmez hale getirilerek belli bir kurala uygun karıştırılması şifreleme olarak bilinmektedir. Şifrelemenin en önemli görevi saldırılara karşı verilerin güvenliğini sağlamasıdır. Şifreli verilerden orijinal verilerin geri alınma süreci şifre çözme işlemidir. Bulut sunucularında veri dosyalarını depolamadan önce hem simetrik anahtarlı, hem de asimetrik-anahtar algoritmaları [29] kullanılabilir. Şek. 3.6'da bulut bilişiminde genel güvenlik ilkeleri gösterilmektedir.

Aranabilir şifreleme tanımı ilk kez Song ve ark. [4] tarafından önerilmiştir. Önerilen bu yöntemde dosyada olan kelimelerin her biri iki katmanlı bir şifreleme yapısı ile tamamen bağımsız olarak şifrelenmekteydi. Yapılan aramalar kaynakta mevcut olan dosyaların miktarına göre doğru orantılı olacaktır. Aranabilir şifreleme

yöntemleri hem açık anahtarlı, hem de simetrik anahtarlı ortamlarda ele alınmıştır. İlk yapılan çalışmalarda, aranabilir şifreleme sadece tek kullanıcı olarak kullanılmaktaydı. Daha sonra geliştirilen yöntemlerde çok kullanıcıli aranabilir şifreleme modelleri geliştirilmiştir. Veri belgeleri tek bir kişiye ait olduğu halde, veriler üzerinde birden fazla kullanıcı arama yapabilmektedir. Veri sahibi dış kaynağa aktarılan dosyalar üzerinde erişim kontrolüne, ayrıca kullanıcılara yetki verme yeteneğine sahiptir.

Her bir şifreleme yönteminin sağlaması gereken en önemli özellikler tutarlılık ve güvenlidir. Goh ve ark. [22] ve Golle ve ark. [30] tarafından geliştirilen yöntemlerde, tutarlılık koşulu mükemmel tutarlılık koşuluna (BDOP) analog olarak verilmektedir. [22] çalışmasında, yanlış pozitiflik olasılığının sıfırdan farklı olduğu gösterilmektedir. Ancak bu çalışmalardan hiç birinde mükemmel tutarlılık kesin olarak gösterilmemiştir. Şek. 3'te, çeşitli aranabilir şifreleme yöntemleri kullanılarak geliştirilmiş arama yöntemleri oranı verilmektedir.

Şema	Song ve ark. [4]	Goh ve ark. [22]	Cao ve ark. [8]	Moataz ve ark. [20]	Curtmola ve ark. [22]	Orencik ve ark. [24]	Mikhail S. ve Indrajit R. [35]
Eşleştirme	Tam	Tam	Benzerlik	Tam	Tam	Tam	Benzerlik
Rastgele sorgu	Hayır	Hayır	Evet	Evet	Hayır	Hayır	Evet
Güvenlik	CPA	CKA1	CKA1	CKA2	CKA2	CKA2	CKA2
Arama karmaşıklığı	$O(PT)$	$O(\#PT)$	$O(\#PT)$	$O(\#PT)$	$O(\#PT_{k^-})$	$O(\#PT)$	$O(\#PT_{k^-})$

Çizelge 3.1. Aranabilir şifreleme yöntemlerinin karşılaştırılması

Son yıllarda, dış kaynağa aktarılan verilerin güvenliğini sağlamak için geliştirilen metotları genel olarak bir kaç farklı yönde geliştirilmiştir:

- Açık anahtarlı aranabilir şifreleme metodu
- Tek bir veri tabanında toplanan belgelerden bilgi alma
- Özel anahtarlı aranabilir şifreleme metodu

Bu üç yaklaşım için ortak olan özellikler, veri belgelerinin tutulduğu sunucu, sunucuya aktarılan verilerle ilgili bilgi sızdırılmaması ve bu verilere efektif erişimi, arama süreci ve değişiklik yapabilme gerekliliğidir. Çizelge 3.1’de farklı aranabilir şifreleme yöntemleri kullanılarak geliştirilen arama yöntemleri karşılaştırılmıştır. Burada #PT – PT kümesinde olan veri dosyalarının sayısını, PT_k ise belirli k - anahtar kelimesini içeren dosyaların sayısını göstermektedir.

Açık anahtar şifrelemesinin kullanıldığı arama yöntemlerinde, veri sahibi tarafından yapılan ön çalışmalar verinin dosyalarının boyutunu arttırmaktadır. Daha sonraki dönemlerde yapılan çalışmalarda ise ön çalışma, dış kaynağa aktarılan verilerin boyutu ile veri sahibi ve kullanıcı arasında çok düşük bir ilişki vardır. Ayrıca, bu yöntemler erişim örüntüsünü de gizli şekilde tutmaktadır [36, 37]. Aranabilir şifrelemenin en önemli özellikleri aşağıdaki gibi sıralanabilir:

- 1) Belgeler anahtar kelimelerle ilişkili olduğu sürece onlara ulaşma imkanı sağlamaktadır.
- 2) Belgeler üzerinde kullanıcılar tarafından sorgular oluşturulmasına izin vermektedir.
- 3) Sunucuya veri dosyalarının içeriği ile ilgili hiç bir bilgi sızıntısına izin vermemektedir.

3.2.1. Aranabilir Açık Anahtarlı Şifreleme

Açık anahtarlı şifreleme algoritmalarında güvenlik tanımı, genelde, gizlilik olarak tanımlanmaktadır. Metinlerin şifrenmesi sunucu veya üçüncü tarafın içeriklere erişmesini engellerken, istemci tarafından anahtar kelime ile oluşturulan “geçit” hangi belgelerin bu anahtar kelimeyi içerdiği bilgisini öğrenemeye izin vermesi gerekmektedir.

Bulut verilerinin şifrenmesi için aranabilir açık anahtarlı şifreleme kullanıldığı zaman, veri sahipleri ve kullanıcıların farklı taraflar olması mümkündür. Açık anahtarlı şifreleme yöntemleri ile şifrelenen belgeler bulut sunucusuna aktarıldıktan sonra, veri sahibi açık anahtarını paylaşır. Bu anahtara erişimi olan her kullanıcı, veri dosyalarına eklemeler yapabilir de, ancak sadece özel anahtara sahip olan kişiler, sorgular oluşturma ve “geçit” üretme yetkisine sahiptir. Boneh ve ark. [34] tarafından 2004 yılında geliştirilen açık anahtar şifreleme yöntemiyle anahtar kelime arama yöntemi (PEKS) ilk kez bulut sunucusuna paylaşılacak

verilerin aranabilir açık anahtarlı şifreleme yöntemiyle şifrelenmesini önermiştir. Bu yaklaşım, belgenin içeriğini açıklamadan sorguların gerçekleştirilmesine ve hangi belgelerin anahtar kelimeleri içerip içermeyeceğini öğrenmesine izin vermektedir. Açık anahtar şifrelemesinin tutarlılığı desteklemesine ve iyi sonuçlar vermesine rağmen, kullanıcının erişim örüntüsünü ortaya koyması nedeniyle güvenlik ve etkinliği azaltmaktadır. Şek. 3.8’de PEKS algoritması kullanılarak açık anahtar şifrelemesi ile bulut verileri üzerinde arama yönteminin genel yapısı verilmiştir.



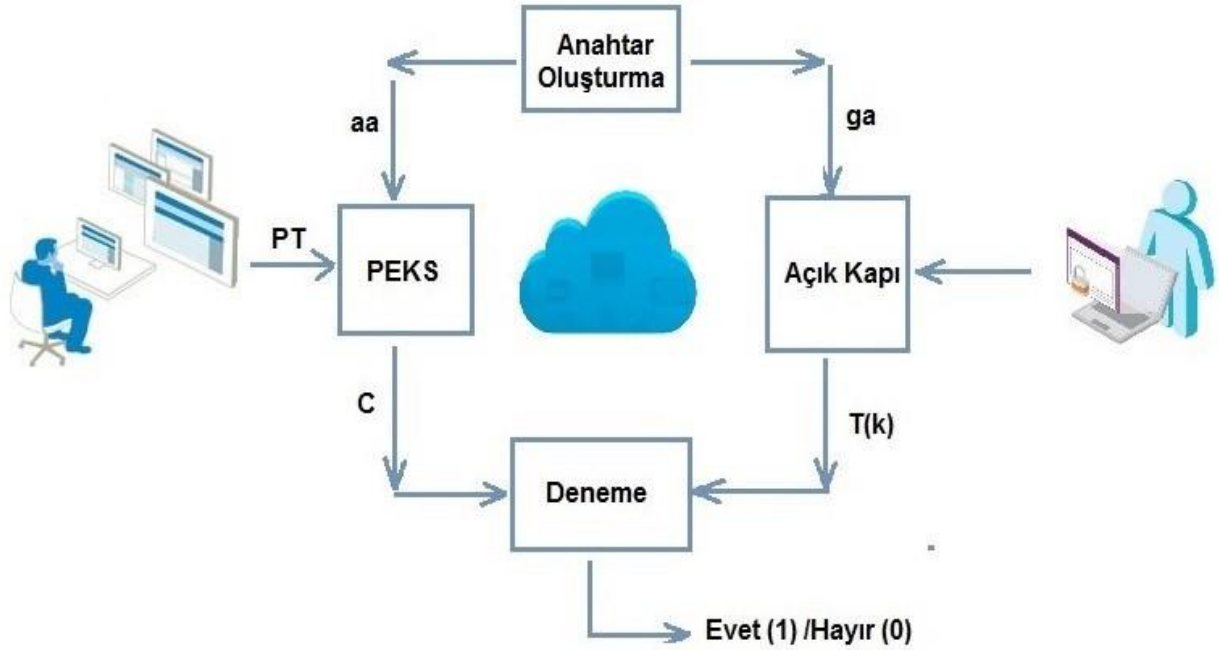
Şekil 3.7. Bulut verileri üzerinde arama yöntemlerinde anarabilir şifreleme yöntemleri

Erişim örüntüsü: D sözlüğü verilmektedir ve $PT \subseteq 2^K$ - D sözlüğü üzerinde veri belgeleri koleksiyonudur. Q sorgu $H = (PT, k)$ geçmişi için erişim örüntüsü $\alpha(H) = (PT(k_1), \dots, PT(k_1))$ gibi ifade edilebilir.

Sorgu geçmişi: Bir D sözlüğü verilmiştir ve $PT \subseteq 2^D$ - PT üzerinde belgeler koleksiyonudur. PT üzerinde Q -sorgu geçmişi PT belge koleksiyonundan ve D anahtar kelimeler vektörü $D = (d_1, \dots, d_q)$ içeren $H = (PT, d)$ üçlüsünden oluşmaktadır.

Bellare, Boldyreva ve O'Neill tarafından sunulan açık anahtarlı verimli aranabilir şifreleme (ESE) [38] yaklaşımı PEKS [34] yaklaşımının aksine, veri sahibinin açık anahtarına erişimi olan tüm kullanıcılara verilere ekleme yapma ve arama sorguları sonuçları için “geçit” üretme yetkisi vermektedir. Bu yaklaşım

geniřletilmiř ortak anahtar ortamına sahiptir. ESE yaklařımı kullanıcılara optimum arama sũresi saęlamasına raęmen, deterministik olması nedeniyle daha zayıf gũvenlik garantisine sahiptir. Bu arama yũntemini kullanan istemci veriler ũzerinde seęici arama yeteneęini kaybetmektedir.



řekil 3.8. Anahtar kelime araması ile aęık anahtar řifreleme

3.2.2. Veritabanlarında ȧzel bilgi eriřimi (PIR)

Veritabanında depolanan veriler tũm kullanıcılara aęık řekilde tutulur. Kullanıcının bir veri dosyasını kullandığı zaman bu konuda gizlilięi saęlamak ięin sunucuda depolanan tũm verileri indirmesi gerekmektedir. Ȑzel Bilgi Eriřimi (PIR) protokolleri ilk kez Kushilevitz ve Ostrovsky [39] tarafından sunulmuřtur ve kullanıcılara kullanıcı eriřim ȧrũntũsũ bilgisini vermeden gereken veri dosyalarını elde etme imkanı sunmaktadır. Bir veritabanının birbirleriyle iliřkisi olmayan birkaç kopyasının olması [20] ve tek bir veritabanı ięin PIR [2] tanımı sunulmuřtur. Bu yaklařım, kullanıcıya eriřim ȧrũntũsũ bilgisini aęıkلامadan verilere eriřim imkanı saęlamaktadır. PIR protokollerinin kullanıldığı sunuculardan veriler kullanırken eriřim ȧrũntũsũnũ gizli řekilde tutabilmek ięin ęalıřtırılan uygulamanın sunucuda mevcut olan her bir belgeye uęraması gerekmektedir. Aksi taktirde, sunucu hangi belgeler ięin sorgular oluřturulduęu bilgisini elde edecektir ve bu da bilgilerin gizlilięi, gũvenlięi ve bũtũnlũęũnũn bozulmasına neden olmaktadır. Yani, PIR

şemaları veritabanı boyutu ile doğrusal olarak çalışır [40]. Her sorgu için sunucunun işlevlerinin kaydedilmesi durumunda, çok sayıda kullanıcı ve sorgu için uyarlanabilir [41, 42]. Bu da veriler üzerinde arama yapıldığı zaman, arama işleminin uzun sürmesi nedeniyle sonucun etkinliğini azaltmaktadır. Literatürde, PIR yaklaşımıyla ilgili yapılmış çalışmalar genellikle çok sayıda güvenlik sınırları nedeniyle pratik uygulamalarının genişletilmesi zordur. Ayrıca, bu yaklaşımlar sorgu izolyasyonu ve şifreli güvenliği desteklemez.

3.2.3. Aranabilir Simetrik Şifreleme

3.2.3.1. Simetrik şifreleme:

Simetrik şifreleme modeli üç polynomial zamanlı $SKE = (Gen, Enc, Dec)$ algoritmasından oluşmaktadır. Gen algoritması mevcut olan f güvenlik parametresini girdi olarak alarak ga gizli anahtarını döndüren polynomial zamanlı algoritmadır. Enc polynomial zamanlı algoritması girdi olarak f gizli anahtarını ve PT düz metnini alır ve çıktı olarak, C şifreli metnini döndürür. Dec algoritması girdi olarak, K anahtarını ve C şifreli metnini alır ve çıktı olarak, PT düz metnini döndürür. Curtmola ve ark. [43] tarafından geliştirilen arama yönteminde seçilmiş şifresiz saldırılara karşı ayırt edilemezlik kavramı (PCPA) [44] ileri sürülmüştür. Bu kavram şifreli metinlerin rastgele ayırt edilemez olduğunu garantilemektedir. Ayırt edilemezlik kavramı aşağıdaki gibi anlatılabilir:

$SKE = (Gen, Enc, Dec)$ - simetrik şifreleme algoritması ve A - saldırgan (sunucu) tarafı verilmiştir. Öncelikle, Gen algoritması f güvenlik parametresini alır ve $ga \leftarrow Gen(1^f)$ - gizli anahtarını döndürür. A saldırganı Enc(p) algoritmasına erişim elde eder. A saldırganı p düz metnini elde eder. Burada C – simetrik şifreleme yönteminin şifreli metin uzayını göstermektedir. $c_0 \leftarrow Enc(m)$, $c_1 (\$) \leftarrow C$, iki şifreli metni oluşturulur. Rastlantısal b biti rastgele seçilir ve c_b A saldırganına (sunucuya) döndürülür. A yeniden şifreli metin uzayına ayrıcalıklı erişim elde eder ve polynomial çoklu sorgularının devamında çıktı olarak b' bitini verir. $b' = b$ şartı ödenirse, sonuç olarak 1, aksi taktirde 0 döndürülür.

Simetrik anahtar şifreleme yöntemi, olasılığın b biti seçilmesi ile sonlanması durumunda tüm polynomial zamanlı saldırılar için CPA- güvenli olarak kabul edilir.

$$Pr [PCPA_{SKE,A}(f) = 1] \leq \frac{1}{2} + negl(f) \quad (1)$$

Ortak simetrik anahtar şifreleme düzenleri (AES vb.) bu kavramın gereklerini yerine getirmektedir. Ayrıca simetrik anahtar şifrelemesi verilere sınırlı sayıda kullanıcının ulaşmasına izin verir veya kullanıcıların sınırlı kaynağa erişimini sağlar.

3.2.3.2. Simetrik Aranabilir Şifreleme

Simetrik aranabilir şifreleme [25] yaklaşımı son dönemlerde şifreli bulut verileri üzerinde çok sayıda anahtar kelime ile arama metodlarında kullanılan en önemli yaklaşımlardan biridir. Bu şifreleme yöntemi, veri sahiplerine verilerini uzak sunucuya güvenli şekilde aktarma ve depolama imkanı sağlarken, aynı zamanda seçici aramalar yapılmasına da izin vermektedir. Veri sahipleri veri belgelerini kendi isteklerine uygun şekilde organize eder ve belirli şifreleme yöntemi kullanarak şifreler. Ayrıca, bu şifreleme yöntemini kullanan veri sahiplerinin daha verimli ve efektif sonuçlara ulaşabilmesi için ek veri yapıları da oluşturulabilmektedir. Simetrik aranabilir şifreleme yöntemlerinin sağladığı en önemli özelliklerden biri de veri sahiplerine belgelerine ek veri yapıları ekleme, yani indisleme imkanı sağlamasıdır. Veri sahipleri şifreli dosyaları ve indis dış kaynağa gönderdikten sonra anahtar kelimeler ile belirli trapdoorlar oluşturabilmektedir.

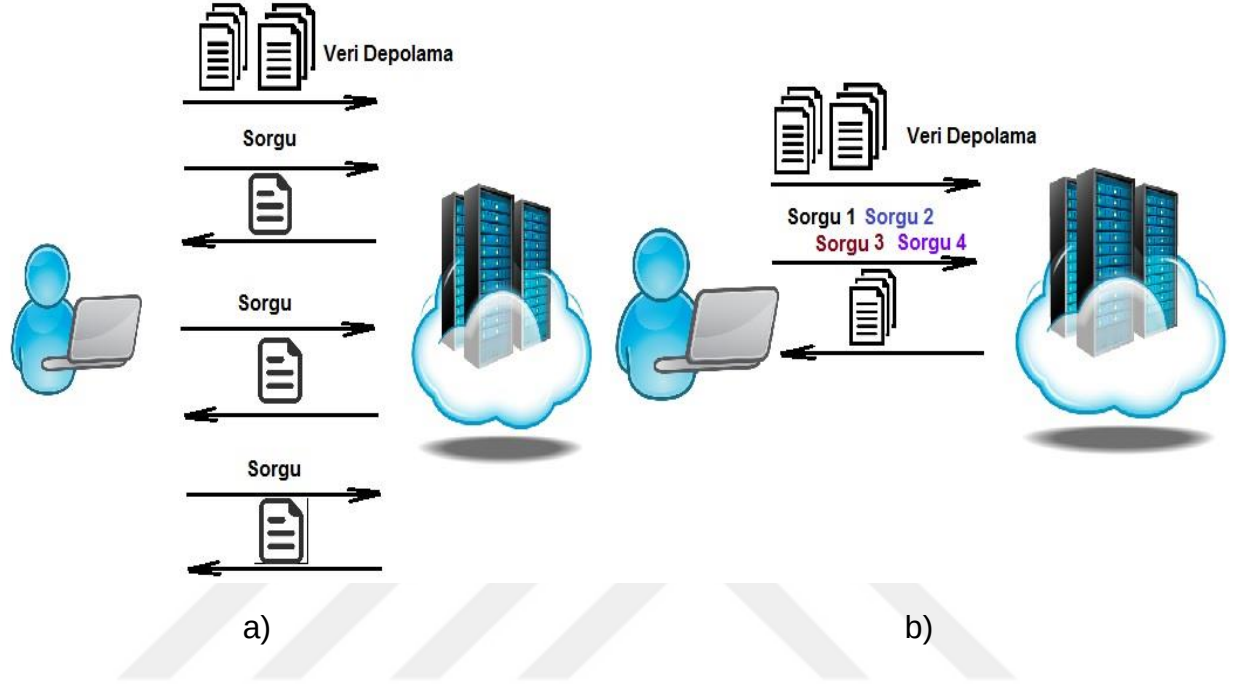
Kullanıcı önceden yapılmış arama sonuçlarını fonksiyon olarak kullanarak anahtar kelimelerini oluşturmayı seçebilir. Simetrik aranabilir şifreleme yöntemleri kullanılırken, anahtar kelimeler üretildiğinde önceki arama sonuçlarının fonksiyon olarak kullanılıp kullanmadığını bilmek önemlidir. Yapılmış olan aramaların sonuçlarına dayanarak, oluşturulan sorguların adaptif veya adaptif olmayan sorgular olup olmadığı belirlenmektedir. Adaptif sorgular kullanıcı tarafından önceden yapılmış sorguların sonuçlarına dayanarak oluşturulan yeni sorguları kapsamaktadır. Adaptif olmayan sorgular ise kullanıcının sorgu oluşturmak için fonksiyonunu kendinin oluşturmasını ve eski sonuçları baz almamasını gerektirir. Sorguların oluşturulmasında bu ayrımı yapılmasının nedeni gereken gizlilik gereksinimlerinin belirlenmesidir. Adaptif olmayan sorgular sadece kendi anahtar kelimelerini toplu şekilde oluşturan kullanıcılar için veri güvenliği sağlar. Adaptif sorgular anahtar kelimelerinde hatta eski arama sonuçlarını fonksiyon olarak

kullanmış kullanıcılar için dahi güvenliği sağlamaktadır. Aranabilir simetrik şifreleme yöntemi adaptif sorguların oluşturulması sürecine uygundur. Şek 5'te adaptif ve adaptif olmayan sorguların genel yapısı gösterilmiştir.

Simetrik aranabilir şifreleme ile birlikte geliştirilen en önemli yöntemlerden biri güvenli indis yöntemidir [22]. İndis yapısı anahtar kelime ile efektif arama yapılmasını destekleyen, bulut sunucusuna aktarılmadan önce belgelere eklenen ek veri yapılarıdır. Bu yaklaşım, veri belgelerini depolarken, anahtar kelime ile verimli şekilde arama yapmayı destekleyen bir yapıdır. Anahtar kelime verildiği zaman, indis bu kelimeyi içeren belgeler için bir işaretçi döndürür. Bu yaklaşım, “geçite” sahip kullanıcılar tarafından anahtar kelime kullanılarak yapılan aramalarda güvenlik sağlar ve bilgi sızıntısına izin vermez. “Geçitler” sadece gizli anahtar kullanılarak oluşturulmaktadır.

Şifreli bulut verileri üzerinde arama gerçekleştirildiği zaman, kullanıcı tarafında anahtar kelimelerle oluşturulan sorgular sunuculara gönderildiğinde, veri dosyasının indisi bu belgenin içerdiği işaretçiyi sunucuya döndürür. İndekslerin güvenli olmasını sağlamak için, sorgu oluşturma ve arama yapma yeteneği sadece “geçite” sahip kullanıcılara verilmektedir. Yetkili kullanıcıların sonuca ulaşmak için sunucuya gönderdikleri “geçitler” ise sadece gizli anahtar kullanılarak ve veri sahibi tarafından oluşturulabilir. Yetkisiz kullanıcıların veri dosyalarının içeriğiyle ilgili hiçbir bilgiye erişememesi sağlanır. Güvenli indis yapısının uygulanma süresi bu şekildedir: veri sahipleri veri belgeleri için indis oluşturur ve şifreler, daha sonra bu indis şifreli belgelerle birlikte bulut sunucusuna aktarılır. Anahtar kelime ile arama yapmak için, kullanıcı sorgu oluşturur ve sahip olduğu “geçiti” ve sorguyu sunucuya gönderir. Sunucu “geçite” uygun indisi bulur, ve daha sonra bu indisten alınan işaretçiye uygun veri dosyalarının döndürülmesini sağlar. Ostrovsky ve Goldreich tarafından sunulan habersiz-RAM yöntemi [45] kullanılarak arama sorguları oluşturulduğu zaman daha iyi güvenlik elde edilmektedir. Bilgi sızıntısını engellemek için anahtar kelimelerin birleşme ve kesişimleri kullanılmıştır. Logaritmik artış problemini çözmek için, araştırmacılar tarafından iki döngülü çözüm önerilmiştir. Ancak bu yöntemin dezavantajı, sonuçların oldukça büyük karekök yükü ile elde edilmesidir. [4], [22], [45] ve [46] çalışmalarında, daha efektif ve verimli çözümler elde etmek için güvenlik gereksinimlerini düşürmeye ihtiyaç duyulmuştur.

Güvenli indis yaklaşımıyla ilgili bu güne dek bir çok tanım önerilmiştir. Bunlara örnek olarak, kullanıcı sorgularının güvenliğini garanti etmese de, verimlilik açısından gelişmiş bir yöntem olan seçilen anahtar kelime saldırılarına karşı ayırt edilemezlik (IND2-CKA) [22], trapdoor ve indis güvenliğini sağlayan simülasyon tabanlı tanımı [46] vb. gösterilebilir.



Şekil 3.9. a) Adaptif sorgular, b) adaptif olmayan sorgular

Aranabilir Simetrik Şifreleme yönteminin genel yapısı aşağıdaki gibi gösterilebilir: K anahtar kelime sözlüğü verilmiştir. Bu sözlük üzerinde index-tabanlı aranabilir şifreleme yöntemi beş polynomial zamanlı SSE = (Gen, Enc, Geçit, Arama, Dec) algoritmasından oluşur:

- 1) $k \leftarrow \text{Gen}(1^f)$ olasılıklı bir anahtar oluşturma algoritmasıdır. Kullanıcı tarafından şemanın kurulumunda kullanılır. f güvenlik parametresini alır ve k gizli anahtarını döndürür.
- 2) $(I, C) \leftarrow \text{Enc}(f, D)$: bir olasılık algoritmasıdır. Kullanıcılar tarafından uzak sunucuya aktarmadan önce belgelerin şifrlenmesi için kullanılır. Girdi olarak g gizli anahtarını ve belgeler koleksiyonunu ($PT=(p_1, \dots, p_n)$) alır ve çıktı olarak I güvenlik indisini ve $C = (c_1, \dots, c_n)$ şifreli belgelerini döndürür. Bazı kaynaklarda $(I, C) \leftarrow \text{Enc}_K(PT)$ şeklinde de ifade edilmektedir.
- 3) $T \leftarrow \text{Geçit}(k, w)$: deterministik algoritmadır. Kullanıcılar tarafından trapdoor oluşturmak için çalıştırılır ve bunun için anahtar kelimeleri kullanır. Girdi

olarak g gizli anahtarını ve k anahtar kelimelerini alır, çıktı olarak ise T geçitini döndürür. Bazı kaynaklarda $T \leftarrow \text{Geçit}_K(k)$ şeklinde de ifade edilmektedir.

- 4) $X \leftarrow \text{Arama}(I, T)$: deterministik algoritmadır. Uzak sunucu tarafından şifreli belgeler arasından k anahtar kelimesini içeren belgeleri bulmak ve döndürmek için kullanılmaktadır. Bu algoritma veri dosyalarında geçen dosyaları almak için I şifreli indis ve T trapdoorunu kullanır ve P belge tanımlayıcıları dizinini döndürür.
- 5) $D_i \leftarrow \text{Dec}(k, c_i)$: deterministik algoritmadır. Kullanıcı tarafından döndürülen belgelerin şifresini çözmek için kullanılır. k gizli anahtarını ve c şifreli metnini alır ve p_i dosyasını döndürür. Bazı kaynaklarda $p_i \leftarrow \text{Dec}(c_i)$ şeklinde ifade edilmektedir.

Tüm $d \in N$ için, $\text{Gen}(1^d)$ tarafından döndürülen tüm k 'lar için, tüm $p_i \in 2PT$ için, $\text{Enc}^k(PT)$ tarafından döndürülen tüm (I, c) için, tüm $k \in K$ için, aşağıdaki şart sağlanırsa, indis tabanlı aranabilir simetrik şifreleme doğru sonuçlanmıştır.

$$1 \leq i \leq n \text{ için, } \text{Arama}(I, \text{Geçit}_K(k)) = K(k) \quad \text{Dec}_K(c_i) = p_i \quad (2)$$

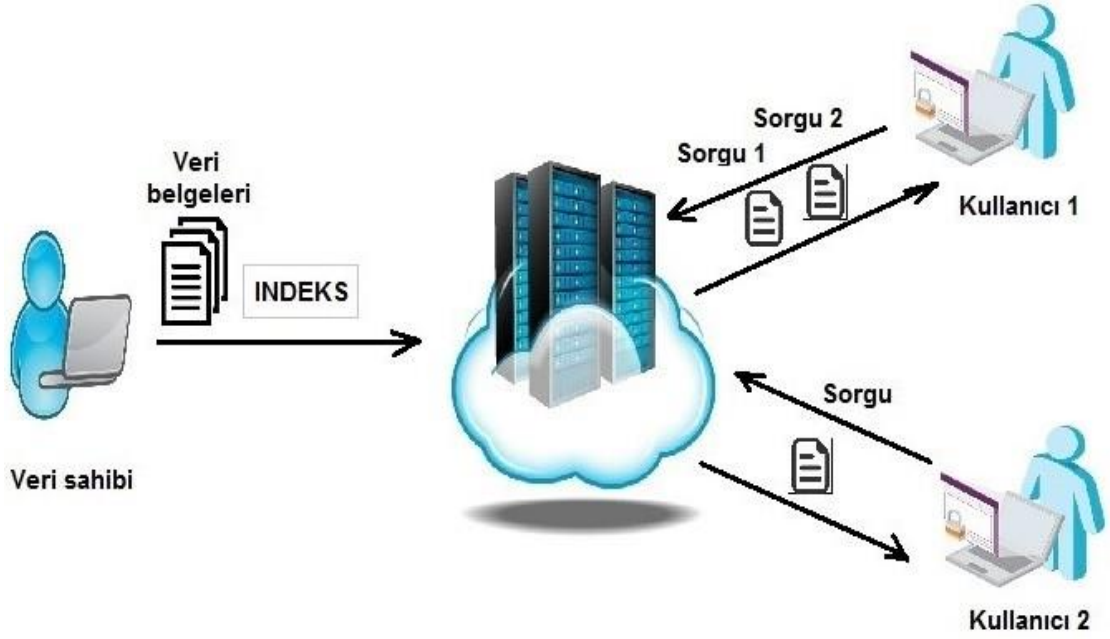
İlişkili döngü karmaşıklığının aradan kaldırılmasının daha düşük güvenli bir başka bir yaklaşımla gerçekleştirilebileceği Song, Wagner ve Perrig [4] tarafından gösterilmiştir. Geliştirilen habersiz RAM aranabilir simetrik şifreleme ile ilişkili döngü karmaşıklığı aradan kaldırılmıştır. Sunulan bu algoritmanın güvenlik gereksinimleri düşük seviyededir. Belirtilen aranabilir şifreleme özel iki katmanlı şifreleme yapısı ile elde edilmiştir. Ancak bu yaklaşımın bazı önemli sınırlamaları vardır: bu aranabilir şifreleme düzeninin güvenli olduğu kanıtlanırsa bile, güvenli bir aranabilir şifreleme şeması olduğu kanıtlanmamıştır. Şifrelenmiş olsa bile, mevcut belgeler istatistiksel saldırılara karşı savunmasızdır ve arama süresi veri belgelerinin boyutu ile doğrusal olarak artmaktadır.

Curtmola ve ark. [43] tarafından iki aranabilir simetrik şifreleme yöntemi önerilmiştir. Birinci yöntem SSE-1 yaklaşımı verimli olmanın yanı sıra, sadece adaptif olmayan saldırılara karşı bulut verilerinin güvenliğini sağlarken, önerilen ikinci SSE – 2 yapısı sunucuda oluşturulan her sorguya karşılık yüksek iletişim boyutu ve daha fazla depolama alanı kullanarak adaptif saldırılara karşı güvenlik sağlamaktadır. Goh [22] seçilmiş anahtar kelime saldırılarına karşı ayırt

edilemezlik (IND2-CKA) yaklaşımı sunmuştur. IND2 –CKA güvenlik tanımında, indise erişim verilmesi durumunda, saldırganın (sunucunun) veri dosyaları ile ilgili bilgilere erişmesi imkansızdır, seçilmiş anahtar kelime saldırısında, yani sunucu tarafından seçilen anahtar kelimeye göre geçit oluşturulması durumunda bile belgelerin içeriğine erişim sağlanamaz. Bundan başka, Chang ve Mitzenmacher [47] tarafından simülasyon tabanlı tanım geliştirilmiştir. Bu yaklaşımlar mevcut simülasyon tabanlı tanımlarla ilgili olarak, daha güvenli yapı elde etmek için daha güçlü geçitler oluşturulmasını ile daha güvenli yapının elde edilmesini önermektedir.

Aranabilir simetrik şifrelemenin kullanıldığı yapılara karşı iki mevcut saldırı modeli belirlenmiştir: adaptif olmayan ve adaptif saldırı modelleri. Adaptif olmayan saldırılar, oluşturulan geçitlerin, önceki aramaları ve döndürülen sonuçları dikkate almadan kendi oluşturduğu şemaya uygun saldırılar yapar, saldırgan (sunucu) önceden kullanıcı tarafından yapılmış arama sonuçları ile ilgilenmez. Önceden önerilmiş habersiz - RAM [36, 37] dışında bütün saldırı şekilleri adaptif olmayan saldırılara aittir. Adaptif saldırılarda ise, saldırgan (sunucu) saldırı fonksiyonu olarak önceden oluşturulmuş geçitleri ve arama sonuçlarını baz alır. Bu saldırı modelinde, veri yapıları üzerinde önceden yapılan arama sonuçlarını, oluşturulan geçitler kullanılarak fonksiyon oluşturulur. Habersiz RAM [36, 37] yaklaşımında ele alınmış olan saldırı modelleri adaptif saldırılardır. Bu modelde, kullanıcılar tüm aramalarını tek seferde yaptıkları taktirde, yapılan aramalar ve döndürülen sonuçların saldırılara karşı tam güvenliği sağlanır. Aksi takdirde, bu saldırılar elde ettikleri her tür bilgiyi adaptif şekilde kendi fonksiyonlarını oluşturmak için kullanarak veri belgelerinin içeriğine erişme imkanı elde ederler. Ayrıca, adaptif ortamlar için oyun tabanlı ve simülasyon tabanlı tanımları geliştirilmiştir.

İlk uygulanan simetrik aranabilir şifreleme yöntemleri tek kullanıcılar tarafından sorguların oluşturularak arama yapılmasına sağlamaktaydı. Tek kullanıcıli aranabilir simetrik şifreleme yöntemlerini çok kullanıcıli aranabilir şifreleme yöntemleri arasındaki farkı yayın şifreleme metodunun (geri alma) kullanılmasıdır. Bu şifreleme yöntemi anonimlik özelliğine sahiptir, yani bulut sunucusu sorguların hangi kullanıcı tarafından oluşturulduğunu bilmemektedir. Şek. 3.10'da çok sayıda kullanıcıli aranabilir simetrik şifreleme yöntemi şeması verilmiştir.



Şekil 3.10. Çok kullanıcılı simetrik aranabilir şifreleme yönteminin genel yapısı

Aranabilir simetrik şifreleme algoritmalarının dezavantajları, sorgu sonuçlarının döndürülmesi, erişim örüntüsünün gizli tutulması ve belgelerin elde edilmesi için oluşturulan geçitlerin deterministik şekilde oluşturulmasından kaynaklanır. Buna örnek olarak [22, 46, 49] makalelerinde sunulan yaklaşımlarda kullanılan deterministik geçitlerin oluşturulması gösterilebilir.

3.2.4. Özellik Korumalı Şifreleme

Özellik koruyarak şifreleme yöntemi her zaman aynı düz metni aynı şifreli metne çeviren deterministik bir şifrelemedir. PPE yönteminin en dikkat edilmesi gereken özelliği düz metinlerin bazı özellikleriyle ilgili bilgi sızmasına izin vererek metnin şifrelemesini gerçekleştirmektedir. Bu şifreleme yönteminde, deterministik şifreleme [48] ile korunan özellik eşitliktir. Diğer bir deyişle,

$$\begin{aligned} c_1 &= E_k(p_1) \\ c_2 &= E_k(p_2) \end{aligned} \tag{3}$$

$c_1 = c_2$ kontrol ederek düz metin eşitliğini belirlenebilmektedir. PPE yönteminin geliştirilmiş bir kaç farklı türü önerilmiştir. Bunlardan düzen korumalı şifreleme, ortogonalite korumalı şifreleme ve düzen açıklayıcı şifreleme yöntemlerinin gösterebiliriz.

PPE yöntemi öncelikle veritabanlarında, daha sonra ise şifreleme yöntemleri arasında yerini almıştır. Şifreleme yöntemi olarak ilk defa 2006 yılında Bellare ve ark. [k34] tarafından sunulmuştur. Bu çalışmada deterministik şifreleme yöntemine çok dikkat ayrılmamış olsa da, şifreleme üzerine yapılan araştırmalarda geniş yer edinmesine neden olmuştur.

E_D deterministik şifreleme ve E_R standart CPA güvenli (rastgele) şifreleme şeması ise sunucuya aktarılan veri dosyalarının bu yöntemle şifrlenmesi aşağıdaki şekildedir: $PT = (p_1, \dots, p_2)$ veri belgeleri kümesindeki her p_i dosyası için dosyadaki her anahtar kelimenin deterministik şifrelemesi hesaplanır. p_i dosyasının her birinin m sayıda $(d_{i,1}, \dots, d_{i,m})$ anahtar kelimesi olduğu varsayalım. Veri kümesi N sayıda

$$r_i = (b_{i,1}, \dots, b_{i,m}, ptr(c_i)) \quad (4)$$

$b_{i,j} = E_{K_2}^{PT}(k_{i,1})$, $c_i = E_{K_1}^R(PT_{i,1})$ ve $ptr(c_i)$ - c_i şifreli metnin işaretçisidir. Veri sahibi şifreli veriyi $(c_1, \dots, c_n)$ rastgele şifrelemeler ile sunucuya gönderir.

d anahtar kelimesi ile arama yapmak için, kullanıcı deterministik şifreleme kullanarak k anahtar kelimeyi şifreler ve sunucuya gönderir. $b_k = E_{K_2}^{PT}(k_{i,1})$, anahtar kelimenin şifreli hali işaretçi olarak kullanıcaktır. Sunucu bu b_k şifreli anahtar kelimeyi sunucuya gönderilen veri kümesinde olan deterministik şifrelerle karşılaştırır. b_k eğer veri tabanında olan herhangi bir işaretçi ile eşleşirse, bunu izleyerek bir diğerine geçer ve şifreli dosyayı döndürür. Diğer bir deyişle, $1 \leq i \leq n$ ve $1 \leq j \leq m$ şartları için sunucu $b_k = b_{i,j}$ ve onlar eşitse, $ptr(c_i)$ işaretçisini izler ve c_i şifreli metnini döndürür.

Bu yöntemin iyi çalışması ve gerekli sonuçları döndürmesine karşın bazı sınırlamaları vardır. Sunucu için arama ve sonuçlar döndürülürken $O(nm)$ olarak belirlenir, başka bir deyişle, dosyaların sayısı ile lineer olarak artar. Veri tabanının boyutu ile lineer olarak artan yöntemler pratik olarak kullanılmak için fazla yavaştır. Ancak gerçek hayatta bu çok fazla problem yaratmamaktadır, çünkü hızlı aramayı desteklemesi gibi deterministik şifreler saklanabilir ki, bu da daha hızlı sonuçların döndürülmesine sebep olacaktır ($O(\log(n))$).

Bu yöntemin dezavantajı kullanıcı tarafından herhangi bir arama yapılmasa bile, şifreli veri kümesi dosya bilgileriyle ilgili bazı bilgiler vermektedir. Başka bir zayıf tarafı ise aynı dosyanın her zaman aynı şifrelenmesidir.

PPE-tabanlı yöntemlerin daha sonra geliştirilen çeşitleri kullanılarak hızlı sunucu aramasını destekleyen arama yöntemleri geliştirilmiştir. Jie Wang ve ark. [7] tarafından 2015 yılında sunulan bulanık arama yönteminde şifreleme yöntemi olarak düzen korumalı şifreleme yöntemi kullanılmıştır.

3.2.4.1. Düzen Açıklayıcı Şifreleme

Düzen açıklayıcı şifreleme yöntemi verimli aralık sorguları, sıralama ve filtreleme özelliklerine izin vermesiyle aranabilir simetrik şifreleme yöntemleri arasında en önemli temel yapılardan biridir. Bu kavram şifreli veriler üzerinde verimli aralık sorguları çözüm olarak önerilmiştir ve bu özelliği ise düzen korumalı şifreleme (OPE) [50] yönteminden temelini almaktadır. Şifreleme düz metinlerin sayısal sıralamasını korumaktaysa, o zaman bu simetrik aranabilir şifreleme düzen korumalı şifrelemedir. Ancak OPE yöntemi bulut sunucularına aktarılan düz metinle ilgili çok sayıda ek bilgileri açığa çıkarması nedeniyle daha güvensizdir. Ayrıca OPE yöntemi şifrelemesi sayılardır ve iki şifre alanı karşılaştırılırken şifreledikleri düz metni düzenlemektedir ki, bu da arzu edilen güvenliği (“en iyi” CPA güvenliği) sağlamamaktadır. ORE yöntemi [67] düzen korumalı şifreleme yönteminin sınırlamaları ve güven problemleri olmadan şifreli veriler üzerinde aralık sorgular yapılmasına bir çözüm sunmaktadır. Bu yöntem fonksiyonlar tarafından açığa çıkarılan bilgiler için simülasyon tabanlı güvenlik kavramı sunmaktadır.

Tanım: Gizli anahtar şifreleme yöntemi girdi olarak şifreli iki metni alarak onların sözlükteki düzenini bildirecek genel bir işlem uygularsa, bu düzen açıklayıcı şifreleme yöntemidir. Bu işlem gizlilik gerektirir ve herkes tarafından değerlendirilebilir.

Düzen açıklayıcı şifrelemenin özellikleri aşağıda özetlenmektedir:

- Kanıtlanabilir güvenilirlik – uygulanan şifreleme işlemleri OPE’den farklı olarak düz metinle ilgili bilgilerin açığa çıkmasına izin vermemektedir.

- Şifre boyutu kısadır: Şifreli metnin boyutunun düz metnin boyutuyla yakın veya aynı olması gerekmektedir.
- Durumsuz ve etkileşimsiz – şifrelemeler bir birinden bağımsız olmalı ve paralel olarak hesaplanabilmelidir.
- Pratik – şema sadece basit gerçekleştirilebilir ve verimli kriptografik ilkeleri üzerinde geliştirilmelidir.

Düzen açıklayıcı şifreleme algoritması (KeyGen, Şifreleme, Karşılaştırma) gibi üç algoritmadan oluşmaktadır. KeyGen algoritması çıktı olarak (k, hesp) sonucunu verir, burada k gizli anahtar (güvenlik parametresine göre oluşturulur), hesp ise girdi olarak şifrelenmiş iki metni alarak karşılaştıran, çıktı olarak '<' veya '≥' sonucunu veren deterministik algoritmadır. E (k, PT) ve D (k, c) algoritmaları standart şifreleme ve şifre çözme algoritmalarıdır. Şifrelemenin standart doğruluğunun yanı sıra, G algoritmasının tüm (k, hesp) çıktıları ve tüm düz metinler için:

$$\begin{aligned} p_1 < p_2 & \quad \Pr[\text{hesp}(E(k, p_1), E(k, p_2)) = '<'] = 1 \\ p_1 \geq p_2 & \quad \Pr[\text{hesp}(E(k, p_1), E(k, p_2)) = '≥'] = 1 \end{aligned} \quad (5)$$

3.2.4.2. Tam Anahtar Kelime Arama

Şifreli bulut verileri üzerinde geliştirilen arama yöntemlerinin iki farkı türü mevcuttur: tam ve bulanık anahtar kelime arama yöntemleri. Tam anahtar kelime arama yöntemleri gereken bilgilere ulaşmak için kullanıcının oluşturulan sorgularda anahtar kelimeleri doğru şekilde kullanılması beklentisindedir. Anahtar kelimelerin yanlış dahil edilmesi durumunda gereken sonuçların döndürülmemesi muhtemeldir. Bu yöntemin aksine, bulanık anahtar kelime arama yöntemleri genellikle, yanlışların düzeltilerek sonuçların üretilmesi için kullanılır. Örneğin, kullanıcının girdiği anahtar kelimeler yazım hataları içermesi durumunda (kullanıcılar tarafından yanlış anahtar kelimelerin girilmesi yaygındır), ama arama motoru (örneğin, Google arama) hala onları tolere edebilme ve kullanıcının ihtiyaç duyduğu sonuçları döndürme yeteneğine sahiptir.

3.2.5. Açık Anahtar Şifreleme Tabanlı Arama

Açık anahtar şifreleme (PKC) [32] tabanlı arama şemalarında, veri sahipleri tarafından indis ve geçit oluşturmak için farklı anahtarlar kullanılmaktadır.

Kullanıcısı, genellikle, veri sahibi ile etkileşimde olmadan, oluşturulmuş anahtar kelimelere göre serbest şekilde bir geçit üretebilmektedir. Böylece, bu arama yöntemleri dinamik ortamda kullanılmaya çok daha uygun olacaktır. PKC-tabanlı arama algoritmaları, genellikle, SKC tabanlı algoritmalar ile karşılaştırıldığında, daha esnek ve etkileyici sonuçların elde edilmesini sağlamaktadır. Bu şema ile aralık sorgusu, alt küme sorguları kolay şekilde yönetilebilir. Ayrıca, algoritmalarda kullanılan benzerlik ve sıralama kriterlerine dayanarak, sınırlı sayıda ve sıralanmış sonuçlar elde edilmektedir. PKC- tabanlı bir ortamda daha efektif sonuçlar için birleşme ve parçalama aramalarının hibriti çoklu anahtar kelime araması gibi kullanılsa da, çoklu anahtar kelime araması gerçekleştirilememektedir. Birleşik anahtar kelime kullanılmasına rağmen, sıralı sonuçlar, genellikle, düşük verimlilikle elde edilmektedir. Ayrıca bu arama teknikleri, sorguların oluşturulması ve sonuçların döndürülmesi için çok fazla zaman harcamaktadır.

Arama işlevlerini geliştirmek, verimliliği arttırmak ve sarf edilen zamanı kısaltmak için çok sayıda PKC-tabanlı arama yöntemleri önerilmiştir. Boneh ve ark. [32] açık anahtar kelime araması tekniğini kullanarak sunucuda saklanan veri dosyaları üzerinde sadece özel anahtara sahip yetkili kullanıcıların arama yapabileceği ilk PKC-tabanlı arama algoritmasını geliştirmiştir. Hwang ve ark. [51] şifreli veri şeması üzerinde birleşik anahtar kelime araması yöntemini sundu. Anahtar kelimelerin gizlilik ihlalinin hafifletmek için indisle birlikte sorgulanan anahtar kelimeleri gruplaştırıldı. Ancak bu yöntemin en önemli dezavantajı, veri sahibinin anahtar kelimelerden oluşturulabilecek tüm kombinasyonları oluşturmadığı takdirde efektif ve esnek olmamasıdır. Golle ve ark. [30] tarafından önerilen arama yönteminde veri belgesinin anahtar kelimesi bu belge için indise dönüştürülmüştür. Önerilen bu yöntemde, sorgu işlemi yapılırken, sunucu indis olarak hangi rastgele anahtar kelimenin kullanılacağını bilmesi gerekmektedir. Bu sistem bazı gizlilik ihlallerine açık olduğu için verimli sonuçlar sağlamamaktadır. Şifreli bulut veri dosyaları üzerinde arama yapmak için geliştirilen ve verimli sonuçlar elde etmek açısından umut verici olan bir başka sistem de yüklem şifreleme (PE) tekniğidir. Boneh ve ark. [34] tarafından önerilen yöntemde, algoritmanın oluşturulması için yüklem şifreleme tekniği kullanılmıştır. Bu algoritma birleştirici, parçalama, alt küme ve aralık sorguları, ve polinom denklemleri gerçekleştirmektedir. Li ve ark. [52] bulut sunucusunda anahtar kelime

arama düzeni inşa etmek için hiyerarşik yüklem şifreleme tekniği kullanmışlardır. Bu tasarımda, arama yetkisi yalnızca yetkili veri kullanıcılarına verilir ve yetkisiz kullanıcılar kesinlikle arama gerçekleştirememektedir. Bununla birlikte, PE tabanlı güvenli arama algoritmalarının en önemli dezavantajı, pratik kullanıma uygun olmamasıdır, diğer bir deyişle, çok fazla hesaplama yoğunluğuna sahiptir.

3.2.6. Simetrik Anahtar Şifreleme Tabanlı Arama

SKC - tabanlı arama yöntemlerinde [43], kullanıcı birden fazla veri sahibi tarafından paylaşılmış veri dosyaları üzerinde arama yapmak istediği zaman, veri sahiplerinin her birinden oluşturulmuş geçitleri alması gerekmektedir. Bu tür sistemlerde iletişim maliyeti bulutta olan verilerin miktarı, kullanıcının arama yaptığı veri yapısı ve veri sahiplerine bağlı olarak belirlenmektedir. Simetrik ortamda, geçitlerin üretilmesiyle ilgili hala bazı önemli sorunlar bulunmaktadır. SKC-tabanlı arama yöntemlerinin en önemli avantajlarından biri genel arama sürecinin çok daha verimli olmasıdır. Bu nedenle, verimli SKC-tabanlı şifreli veri arama mekanizmalarının geliştirilmesine önemli bir ilgi olmuştur. Simetrik arama ortamlarının diğer bir önemli avantajı çoklu anahtar kelime ile aramayı desteklemesidir. Çizelge 3.2'de bazı simetrik anahtar kelime ile arama yöntemlerinin karşılaştırılması gösterilmiştir.

Curtmola ve ark. [43] simetrik ortamda güvenlik garantileri ile tek anahtar kelime ile sorguları destekleyen güvenli simetrik arama şeması tasarlamıştır. Kullanılan indis oluşturma tekniği arama işlemlerinin son derece verimli olmasını sağlamaktadır. Wang ve ark. [25] tarafından önerilen yöntem, sıralama düzenini korumak için kullanılmaktadır. Ayrıca, anahtar kelimenin frekans bilgilerini ve ters indis yapısının birleşmesi ile, doğru ve etkili arama yapılabilmesini sağlamaktadır. Bu algoritmanın dezavantajı, tekli anahtar kelime arama yöntemini desteklemesidir. Kamara ve ark. [53] verimli dosya ekleme ve silme imkanı sağlayan dinamik bir arama yöntemi önermiştir. Bu yöntemlere ek olarak, çok kullanıcı ortamında uygulanmaya uygun bir sıra SKC-based arama yöntemleri de önerilmiştir. Goh ve ark. [22] indisler için bir INDICKA güvenlik modelini formüle etmiştir. Her dosya için güvenli bir indis oluşturmak için sözde rastlantısal fonksiyonları ve Bloom filtresini ön plana çıkarmıştır. Bu algoritma, arama sırasında veri kümesinde olan dosyaların sayısı ile doğru orantılıdır. Yöntemin temel sorunu, bloom filtersinin temel aldığı indis oluşturulması tekniği nedeniyle arama sonuçları kaçınılmaz yanlış pozitifler

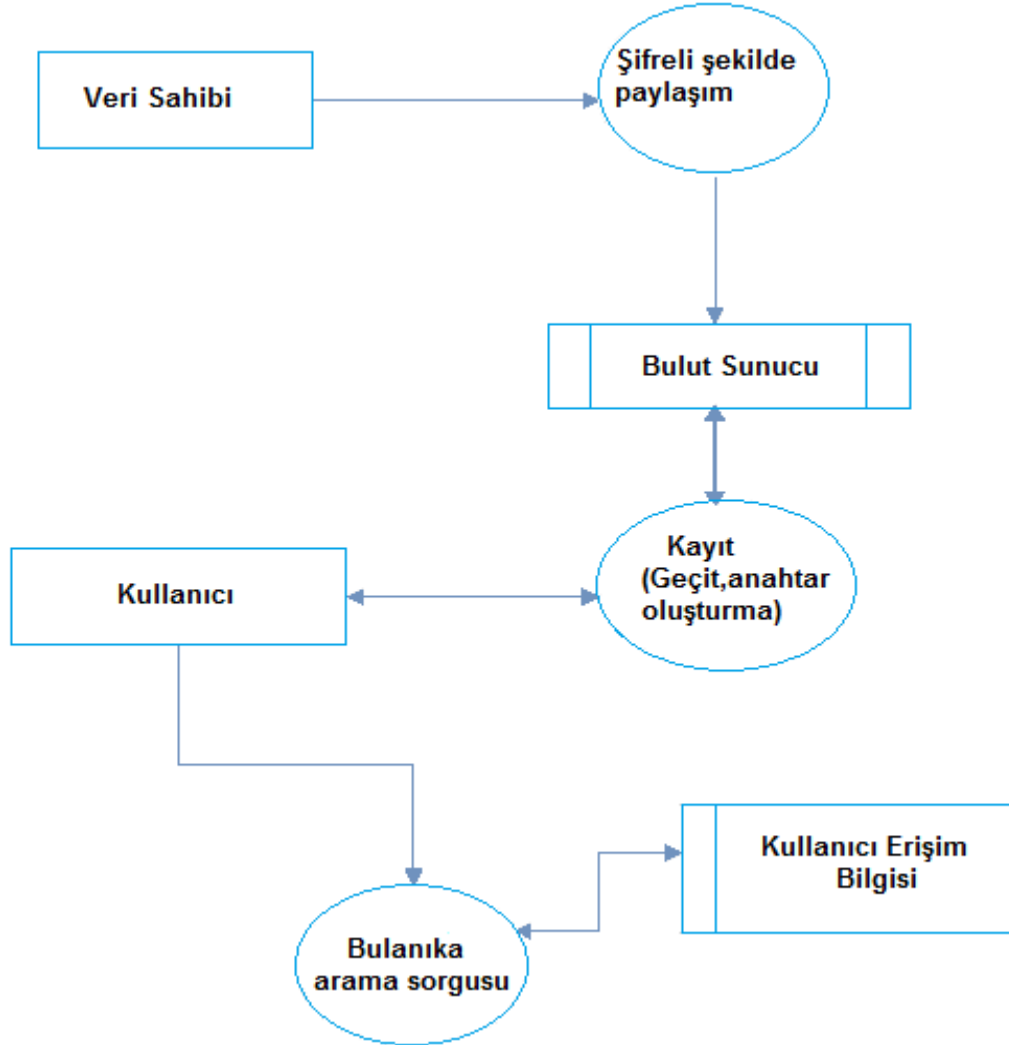
içermesidir. Chang ve ark. [47] IND2-CKA benzer, sözde rastlantısal fonksiyonları ile indis oluşturan güvenli arama modelini sunmuştur. Bu yöntem, [22]'de tanımlanan yönteme benzer olarak her dosya için ayrıca oluşturulan indis yapısına sahiptir. Cao ve ark. [8] güvenli iç çarpım tekniği kullanarak indisin oluşturulduğu çoklu anahtar kelimeli sıralı arama yöntemini destekleyen ilk SKC-tabanlı şifreli veri arama düzenini önermiştir. Sıralama mantığı “koordinat eşleştirmesi” benzerlik ölçüsü ile sağlanmaktadır. Sun ve ark. [35] son teknoloji (state-of-the-art) benzerlik ölçüsünü kullanarak sıralamada daha doğru arama sonucunu sağlayan güvenli çoklu-anahtar kelime arama düzeni sunmuştur. Bu yöntemde, vektör uzay modeli kosinüs ölçüsü kullanılmış ve pratikte arama karmaşıklığını daha verimli yerine getirmek için önerilen ağaç tabanlı indis yapısı kullanılarak arama algoritması tasarlanmıştır.

3.3. Bulanık Anahtar Kelime Arama

Bulanık arama mantığı [52], kullanıcılar tarafından sorgu oluşturulurken anahtar kelimelerde yapılan kelime yanlışlarını ve biçim tutarsızlıklarını destekler, bu hataların ve tutarsızlıkların düzeltilerek aramanın gerçekleştirilmesi ve gereken sonuçların döndürülmesini sağlar. Tam anahtar kelime araması tarafından desteklenen veri gizliliği bulanık arama yöntemleri tarafından da desteklenmektedir. Arama algoritmasının anahtar kelime ve yazım hatası arasındaki farkı daha doğru algılayabilmesi için, şifreli bulut verileri üzerinde arama şemasının da desteklendiği yeni bir benzerlik ölçüsü gerekmektedir. Burada, kullanıcının arama sorgusu anahtar kelimelerle tam eşleştiğinde veya farkın önceden tanımlanmış tolerans aralığında olduğu durumlarda eşleştirme gerçekleştirilir.

Li ve ark. [52] tarafından ilk kez şifreli bulut verileri üzerinde bulanık anahtar kelime arama yöntemini önerilmiştir. Burada, her bir anahtar kelime için, anahtar kelimenin tüm türevlerini içeren bir joker anahtar sözcük kümesini inşa edilmektedir. Joker-tabanlı yöntem, bulanık anahtar kelime dizileri oluşturmak için kullanılır. Ayrıca, kelime benzerliğini değerlendirmek için mesafe düzenlemesi algoritması kullanılmıştır. Bu kümeler tüm bulanık anahtar kelimelerin oluşturulması ihtiyacını ortadan kaldırdığı için, depolama ve sunma değerlerini önemli ölçüde azaltır. Kullanıcının geçit elde etmesinden sonra, kelime benzerliğini ölçmek için düzenleme mesafesi kullanılmıştır. Amaçlanan anahtar kelime bulanık

anahtar kelime kümesine dahilse, bu anahtar kelimenin arama ile eşleştiği kabul edilir ve döndürülür. Mesafe düzenleme algoritması kullanılarak, bulanık anahtar kelime arama tanımı aşağıdaki gibi formüle edilebilir:



Şekil 3.11. Bulanık anahtar kelime arama

$PT = (p_1, p_2, \dots, p_N)$ bulut sunucuda depolanan N sayıda şifrelenmiş veri dosyaları koleksiyonu, d önceden tanımlanmış mesafe düzenlemesi olsun. $K = \{k_1, k_2, \dots, k_p\}$ belirli anahtar kelimeler kümesi ve m ($m \leq d$) düzenleme mesafesi ile (k, m) arama girişi, bulanık anahtar kelime arama yönteminin yürütülmesi kimlik dosyasının dizinini döndürür. Döndürülen veri dosyaları dizini muhtemelen k anahtar kelimesini içerir. Arama gerçekleştirilirken kimlik F olarak ifade edilir: $k = k_i \in K$ olursa, o zaman $F = ID_k$ kimlikleri döndürülür; aksi taktirde, $k_i \in K$, $ed(k, k_i) \leq m$ olursa, $\{F = ID_{k_i}\}$ döndürülür. Bu yukarıdaki tanımda $m \leq d$ varsayımına

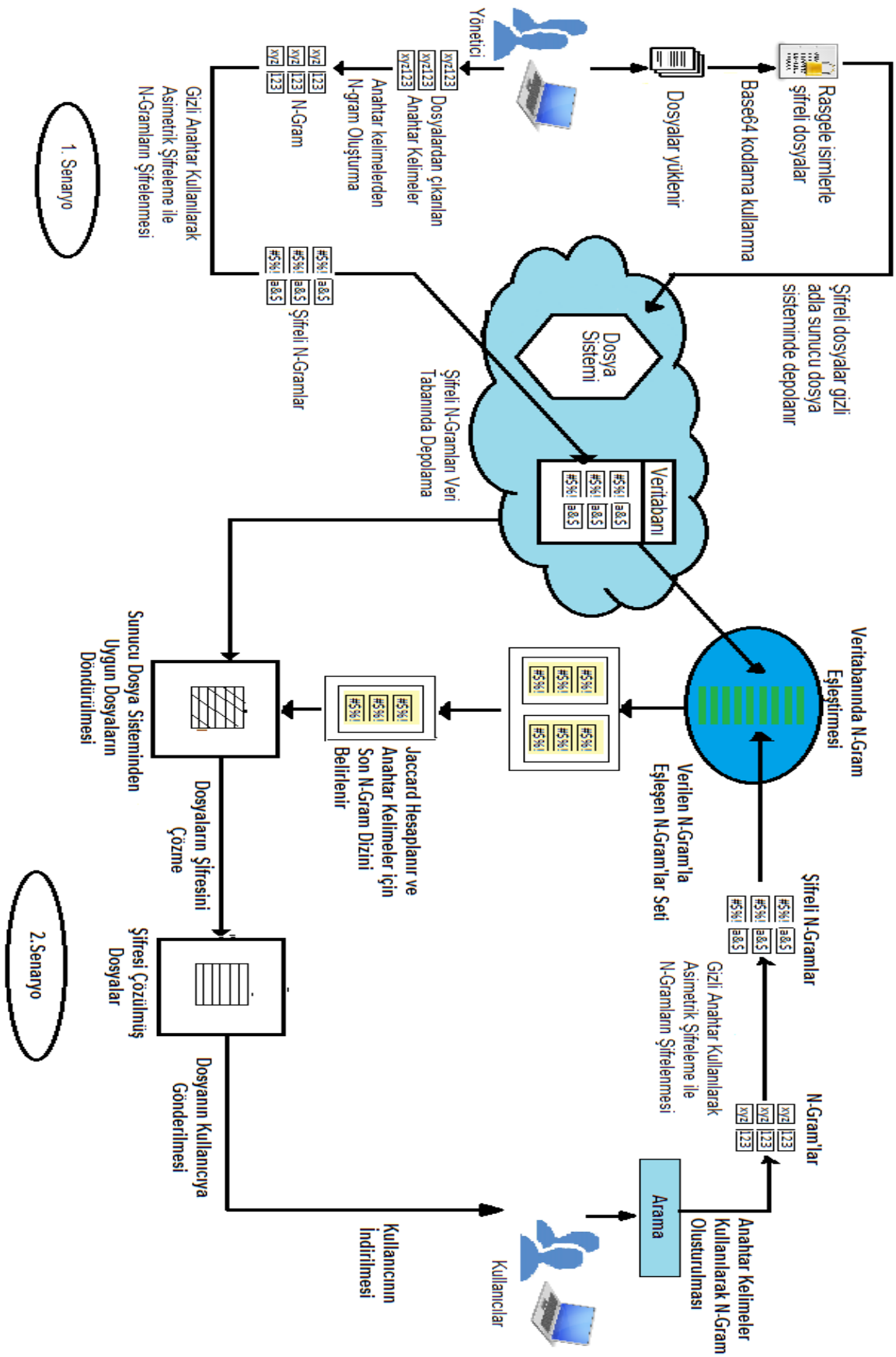
dayanmaktadır. Aslında, d belirli anahtar kelimeler için farklı olabilir ve tam eşleşme başarısız olursa, system $\{F IDk_i\}$ döndürür.

$$\text{tatmin edici } ed(k, k_i) \leq \min\{m, d\} \quad (6)$$

Liu ve ark. [54] daha sonra, boyutu azaltılmış indis kullanılmış bulanık anahtar kelime arama düzeni sunmuştur. Chuah ve ark. [55] bulanık anahtar kelime arama düzeni inşa etmek için bir B-ağaç tabanlı indis yapısı kullanmıştır. Bu yöntemde [56], birkaç anahtar kelime birlikte bir ifade oluşturacak şekilde gruplaştırılarak aramalar yapılabilirdi. Anahtar kelime kombinasyonu indis oluşturma aşamasında veri sahibi tarafından kabul edilmezse, veri kullanıcısı kendi seçtiği anahtar kelimelerin herhangi bir kombinasyonunu sorgulayamamaktadır. Bu yöntemin en önemli dezavantajı esnek olmamasıdır.

Özellikler	Goldreich [k57]	Song [k31]	Goh [k22]	Chang [k47]	Curtmola [k43]
Erişim örüntüsünün gizliliği	Evet	Hayır	Hayır	Hayır	Hayır
Sunucu Hesaplamaları	$O(\log^3 n)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$
Sunucu Depolaması	$O(n \cdot \log n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Döngü sayısı	$\log n$	1	1	1	1
İletişim	$O(\log^3 n)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
Adaptif Saldırıları	Evet	Hayır	Hayır	Hayır	Hayır (SSE-1) Evet (SSE-2)

Çizelge 3.2. Bazı aranabilir simetrik şifreleme ile arama yöntemlerinin karşılaştırılması



Şekil 3.12.Bulanık Anahtar Kelime Arama Şeması

4. ARAMA YÖNTEMLERİ

4.1. Genel Tanım

Bulut sunucusuna aktarılan şifreli veri dosyaları üzerinde yapılan arama işlemleri kullanılan anahtar kelimelere göre üç farklı gruba ayrılmaktadır [34]:

- Tek anahtar kelime ile arama,
- Boolean anahtar kelime araması
- Çoklu anahtar kelime araması

Tek anahtar kelime ile arama yöntemleri, yapılan ilk çalışmalarda geliştirilen, her bir anahtar kelime için farklı bir sorgu oluşturulmasını gerektiren bir arama tekniğidir. Boole anahtar kelime arama yöntemi, birleştirici, parçalanma yöntemleri ve bunların hibridi kullanılarak yapılan arama yöntemleridir. Çoklu anahtar kelime arama yöntemleri son yıllarda geliştirilen, kullanıcıya ihtiyaç duyduğu belgeleri elde etmek için birden fazla anahtar kelime ile tek bir sorgu oluşturarak sonuçların döndürülmesi imkanı sağlayan arama yöntemleridir.

4.2. Tek Anahtar Kelime ile Arama

Uzak sunucuda depolanan şifreli veriler üzerinde arama yapmak için geliştirilen ilk algoritmalar istemcilere sadece tek anahtar kelimeye dayalı sorgular oluşturmaya izin vermekteydi. İstemcinin birden fazla anahtar kelimeyi içeren veri dosyalarına erişim istediği durumlarda, her bir anahtar kelime için bulut sunucusuna ayrı ayrı sorgu göndermesi gerekmektedir. Bu şekilde gerekli dosyaların belirlenerek döndürüleceğinin garantisi yoktur. Ayrıca, kullanıcı arama işlemlerini kolaylaştırmak için sunucuya ek bilgiler gönderebilmektedir. Bu ek bilgiler sonuçların daha efektif ve verimli olmasını sağlamaktadır. Tek anahtar kelime kullanılarak arama yöntemlerinin en önemli dezavantajı, bütün anahtar kelime dizileri ile arama gerçekleştirilmesi durumunda, sunucu hangi anahtar kelimelerin hangi belgelerle eşleştiği bilgisini elde etmesidir. Bundan başka, veriler üzerinde yapılan aramalar bulut sunucusunda depolanan veri dosyalarının miktarına bağlı olarak sonuçlar üretmektedir, diğer bir deyişle, sunucuda çok sayıda belge depolanması durumunda verimli sonuçlar döndürememektedir.

Tek anahtar kelime ile arama yöntemleri sonuç olarak çok sayıda veri belgesi döndürür ve bu algoritmalar sıralama mantığını desteklemediği için, bu da

kullanıcının ihtiyaç duyduğu belgeyi kullanabilmesi için çok sayıda veri dosyasını incelemesini gerektirir. Bu nedenle tek anahtar kelime ile arama işlemi kullanım ve sonuç açısından verimli olarak nitelendirilmemektedir.

Tek anahtar kelime arama yöntemlerinde, genellikle, gizli anahtar kullanılarak oluşturulan geçitler gönderilmediği sürece içeriği gizli şekilde tutacak aranabilir indis kullanılmaktadır. Tek anahtar kelime ile arama çalışması ilk defa Song ve ark. [4] tarafından önerilmiştir. Bu arama tekniğinin güvenlik gereksinimlerinin geliştirilmesi ve daha efektif sonuçların elde edilmesi Goh [22], Chang ve ark. [47], Curtmola ve ark. [43] tarafından çalışmalar sunulmuştur. [22] çalışmasında, aranabilir simetrik şifreleme algoritmalarının güvenlik gereksinimlerinin resmileştirilmesi için yeni bir güvenlik tanımını önerilmiştir. [43] çalışmasında, gizlilik korumalı anahtar kelime arama protokolleri için uyarlanabilir güvenlik tanımları sağlanmış ve bu tanımlara uygun bir yaklaşım önerilmiştir. Verimliliği arttırmak için tüm veri dosyaları için anahtar kelimelere uygun geçit ve ilgili şifreli dosya tanımlayıcıları ile tek bir indis tablosu oluşturulmaktadır. Başka bir tek kelime arama düzeni Wang ve ark. [25] tarafından önerilmiştir. Bu yöntemle her anahtar kelime - dosya çifti için alaka puanı ve şifreli ters indis birlikte tutulması önerilmiş ve sıralama mantığı imkanı sunulmuştur. Son zamanlarda, Kuzu ve ark. [58] tarafından mekana duyarlı özetleme (LSH) fonksiyonları kullanan ve adaptif semantik güvenlik sağlayan gelişmiş bir yöntem sunulmuştur. Bu çalışmanın en önemli avantajlarından biri, benzerlik arama düzenine sahip olmasıdır. Algoritma, sorgularda bazı hataların olması durumunda bile eşleşmeyi sağlamaktadır. Bunu takip eden tek anahtar kelime arama çalışmaları daha kısa sürede daha verimli sonuçlar elde etmek için sıralama mantığı kullanılmasına odaklanılmıştır.

4.3. Boolean Anahtar Kelime ile Arama

Bulut sunucusunda depolanan veri dosyaları, genellikle, geniş ölçülü belgeler koleksiyonudur ve tek anahtar kelime arama yöntemleri verimli sonuçlar elde edilmek için yeterli değildir. Moataz ve ark. [23] bu problemi kısmen ortadan kaldıracak boolean anahtar kelime ile arama düzenini önermiştir. Şema Gram-Schmidt işlemine uygun olarak anahtar kelimelerin dikleştirilmesi metoduna [26] dayanmaktadır. Arama işlemlerini tek anahtar kelime ile arama yöntemleriyle kıyaslandığında, daha verimli hale getirmek ve zenginleştirmek için birleştirici [30] anahtar kelime metot kullanılmıştır. Avantajlarının yanı sıra, bilinear eşleştirmenin

ve gizli paylaşımın artan hesaplama yüküne neden olmaktaydı. Daha sonra, bu yöntem geliştirilerek birleştirme ve parçalama ile aramayı destekleyecek yöntem önerildi [k59].

Şifreli bulut verileri üzerinde geliştirilen arama yöntemleri parçalanma, birleştirme ve olumsuzluk (inkar) Boolean işlemleri kullanılarak geliştirilmektedir [k30]. Birleştirme ile anahtar kelime araması, sadece istemcinin gönderdiği arama sorgusunda kullanılan anahtar kelimelerin olduğu sonuçları, parçalama ile anahtar kelime araması ise, sorguda belirtilen anahtar kelimelerden birini bile içeren tüm belgeleri döndürmektedir. Yükleme şifrelemesinde iç çarpım sorguları iki vektörün ortogonal olup olmadığına esaslanmaktadır, yani, iç çarpım değeri sıfıra eşitlik dışında gizli tutulur. Boolean anahtar kelime arama yöntemlerinin bir çoğu eliptik eğrilerin eşleştirme işlemlerinin masraflı değerlendirilmesi üzerine oluşturulmuştur. Kullanıcı veri kümesinin gereken alt kümesini alacağını bilirse, Boolean sorguları kullanılması kullanıcı tarafından belirtilen hassas arama şartı ile iyi performans göstermektedir. Ancak bu şemalar, pratikte bulutta depolanan veriler üzerinde arama yapıldığı zaman verimli performans elde edilmesini sınırlar.

4.4. Birden Fazla Anahtar Kelime ile Arama

Birden fazla anahtar kelime ile arama yöntemleri son zamanlarda geliştirilen ve diğer arama yöntemlerine göre daha verimli ve efektif sonuçlar elde edilmesini sağlayan yöntemlerdir. İstemci bulutta depolanan belgeler üzerinde arama yapmak için tek bir sorguyu oluşturmak için birden fazla anahtar kelime kullanır. Bu yöntem arama sonuçlarını kısıtlayarak en alakalı belgelerin döndürülmesini sağlar. Orencik ve Savas [60] daha sıkı güvenlik gereksinimlere sahip, sıralama mantığını destekleyen birden fazla anahtar kelime ile arama yöntemini önermişlerdir. Cao ve ark. [8] tarafından önerilen metotta iki ikili matris içine aranabilir veritabanı indisi kodlanmış ve eşleştirme zamanı iç ürün benzerliği kullanılmıştır. Bu yaklaşım, döndürülen sorgu sonuçlarının sayısını azaltırken, kullanıcının gerek duyduğu sonuca en yakın sonuçların döndürülmesini sağlar. Döndürülen belgeler önemlilik sırasına göre anahtar kelimelerin hepsini içeren belgelerden başlayarak sıralanır. Döndürülen belgelerin önemlilik sırasına göre sıralanması ve döndürülen belge sayısına sınırlamalar için istemcinin ihtiyaç duyduğu belge sayısını sorgu ile birlikte sunucuya göndermesi gerekmektedir. Bugün internette arama motorlarının çoğu (örneğin, Google arama) kullanıcılara daha kısa sürede daha verimli

sonular dndrlmesini saėlamak iin tek bir arama isteėinde birden fazla anahtar kelime sorgulaya izin vermektedir. Tek anahtar kelime sorgusu ile karřılařtırıldığında, birden fazla anahtar kelime ile arama yntemlerinin bařlıca avantajı verimli ve daha alakalı arama sonuları elde edilmesine izin vermesidir.

5. İNDİS OLUřTURMA YNTEMLERİ

PT veri kmesi iinde her bir p_i belgesi iin I aranabilir index oluřturulması iin, ncelikle p_i belgelerinden ıkarılmıř anahtar kelimeler dizini veri vektrne eřlenir. Eřleřtirme veya řifreleme ařamasında maliyet verileri D_i szlk boyutuna, yani indis anahtar kelime sayısına gre belirlenen veri vektr boyutları ile doėrudan baėlıdır. Btn dizinin oluřturulma sresi maliyeti veri kmesi iinde belgelerin sayısına eřit olan alt indis sayısına baėlıdır. Her alt indisi oluřturma maliyeti sabit olduėu srece tm indislerin oluřturulmasının zaman maliyeti neredeyse lineerdir. Szlkte anahtar kelime sayısı alt indis oluřturulduėu zamanın maliyetini belirler.

Bir dosyanın dndrlmesi iin sorgu oluřturulurken, veri dosyaları koleksiyonunun řifreli gvenilir indisinin veri sahibi tarafından oluřturulması gerekmektedir. Oluřturulan indis veri belgeleriyle birlikte řifrelenir ve gvenilmeyen bulut sunucusuna gnderilir. řifreli indis tablosundaki her satır, bir belge dizisine karřılık gelmektedir. Her satır iki stn: keyfi atanan benzersiz bir belge kimliėi ve kelime vektrlerinde anahtar kelimenin konumunun řifreli srmleri ile iliřkili kesik simetrik anahtar řifreli anahtar kelimeleri ieren zel bir veri yapısını ieren stnlardan oluřur. Buna ek olarak, bu veri dizesi gvenilir istemci tarafındaki sunucusunda depolanan dosyalar ile kesik řifreli anahtar kelimenin eřleřtirilmesi iin kullanılan bir tamamlayıcı iermektedir. Her iki řifreleme iin kullanılan k řifreleme anahtarları aynıdır ve bu k anahtarı řifre zme iin kullanılabilir. Sadece yetkili kullanıcıların k anahtar deėerine eriřimi vardır. Sunucuya gnderilen řifreli indisler oluřturulan sorgularla birlikte sunucuya gnderilen geitlerle eřleřtikleri zaman, sunucu eřleřen belgelerin iřaretcilerini yetkili kullanıcılara geri dndrr. Bu da gereken dosyalara hızlı eriřim ve en yakın belgelerin dndrlmesini saėlar.

5.1. İndeks Mimarisi

Gnmzde řifreli bulut verilerinin zerinde arama yapmak iin geliřtirilen daha gvenli ve “meraklı” bulut sunucusuna bilgi sızıntısına izin vermeyen yntemler indis mimarisini desteklemektedir. řekil 5.1’de veri dosyaları iin oluřturulan indis

mimarisinin genel akış diagramı verilmiştir. Bulut sunucusuna gönderilen veri dosyalarının daha güvenli şekilde depolanması için oluşturulan indis mimarisi aşağıdaki gibi farklı metotlar kullanılarak oluşturulmaktadır:

- a) Ters indis – diğer bir deyişle, anahtar kelime tabanlı indis mimarisi [7]: bu indis yapısı dosyalarla eşleşen anahtar kelimelerin eşleşme listesini tutar ve tam metin aramasına izin vermektedir. Her anahtar kelime, veri kümesinde bu anahtar kelimeyin dahil olduğu tüm dosyaları içeren bir dosya listesi tarafından takip edilir. Bu tür indisler anahtar kelimeler kullanılarak üretilir. Sıralama yeteneğine sahip arama yöntemlerinde, bu indis yapısı, her dosyaya önceden belirlenmiş fonksiyonla hesaplanan sayısal puan atayarak en alakalı belgeleri belirler. Çizelge 5.1’de dosya kimlikleri ve onlara aktarılan sayısal puanları ifade etmektedir.

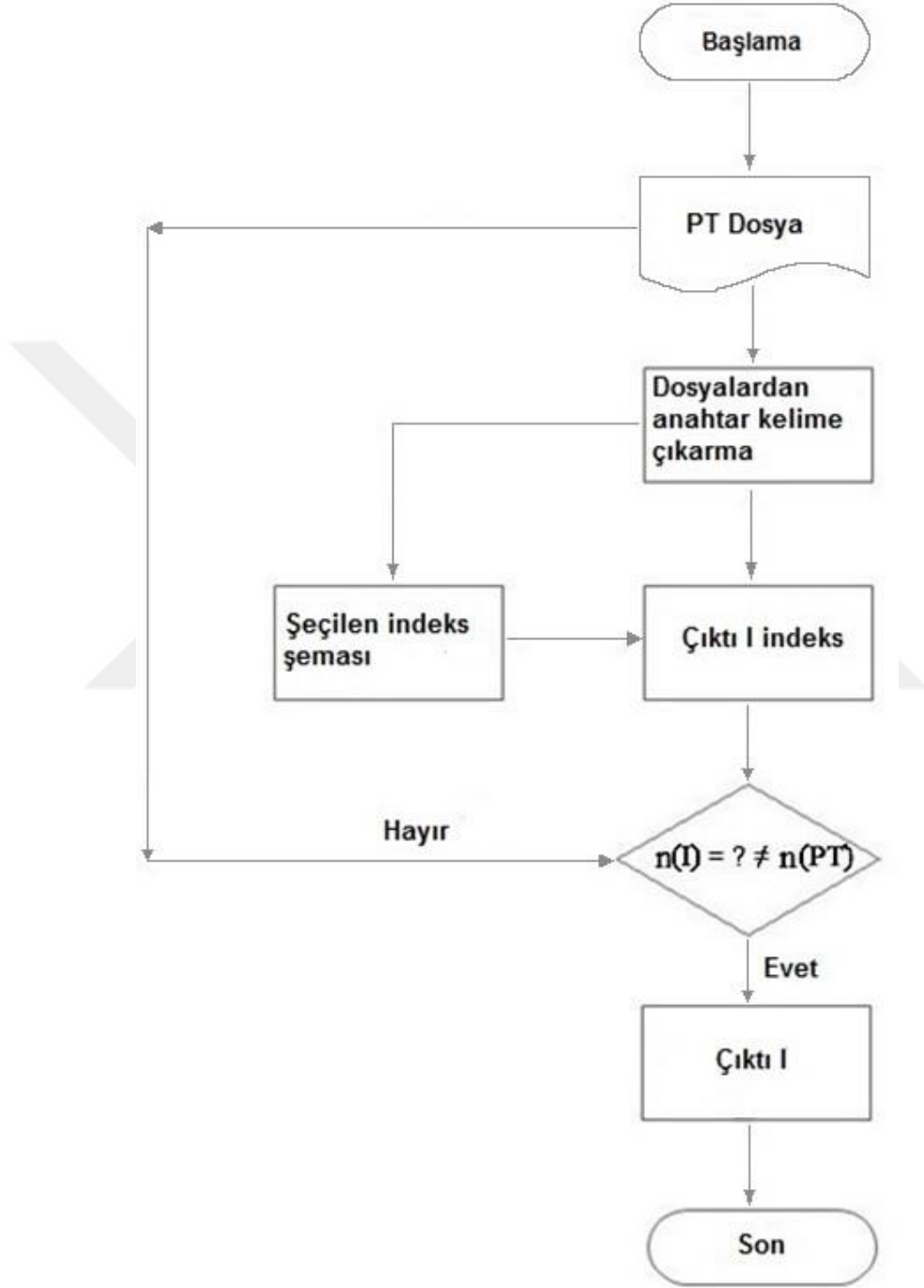
Kelime		k_i		
Dosya kimliği (PT - ID)	p_{i_1}	p_{i_2}	...	p_{i_N}
Uygunluk Puanı	4.51	5.63	...	11.73

Çizelge 5.1. Ters indis mimarisinde dosyaların alaka puanı

Ters indis yapısının avantajı, tam alanda metin araması yerine daha verimli metin araması sağlamasıdır. Veri kümesine yeni bir dosya ilave edildiği zaman, bu dosyadaki anahtar kelimeleri içeren tüm indislerin güncellenmesi gerekirken, yapılan işlemler (hesaplamalar) artar. Bu indis mimarisine dayalı arama işlemlerinin oldukça hızlı şekilde sonuç döndürdüğü gözlemlenmektedir. Bu tür anahtar kelime tabanlı indis yapısı kullanılarak geliştirilen arama yöntemleri sadece tek anahtar kelime ile sorgularının oluşturulmasına imkan sağlar.

- Belge başına indis mimari: birçok güvenli arama düzeni tarafından benimsenen bir başka popüler indis yapısıdır. Veri kümesine dahil olan her dosya için bir indis oluşturulur. Bu da, dosyalar üzerinde yapılan güncellemelerin sadece ona uygun dosyanın indisini etkilemesi anlamına gelir. Indis yapısı veri koleksiyonunda olan her belgeye özeldir ve hedef belgedeki tüm anahtar kelimelere göre oluşturulur.
- Ağaç tabanlı indis mimarisi: bu indis yapısı bazı iyi geliştirilmiş ağaç yapıları kullanılarak oluşturulmaktadır. Bu tür hiyerarşik yapılara örnek olarak B-

ağacı [k56], MDB-ağacı [k31], vs. gösterebiliriz. Birkaç mevcut araştırmada [k35, k60] farklı senaryolarda etkili güvenli arama planları tasarlamak için ağaç tabanlı yapılar kullanılmaktadır. Şekil 5.1’de genel bir ağaç tabanlı indis oluşturma şeması verilmiştir.



Şekil 5.1. İndeks mimarisinin akış şeması

5.2. Bloom Filtresi

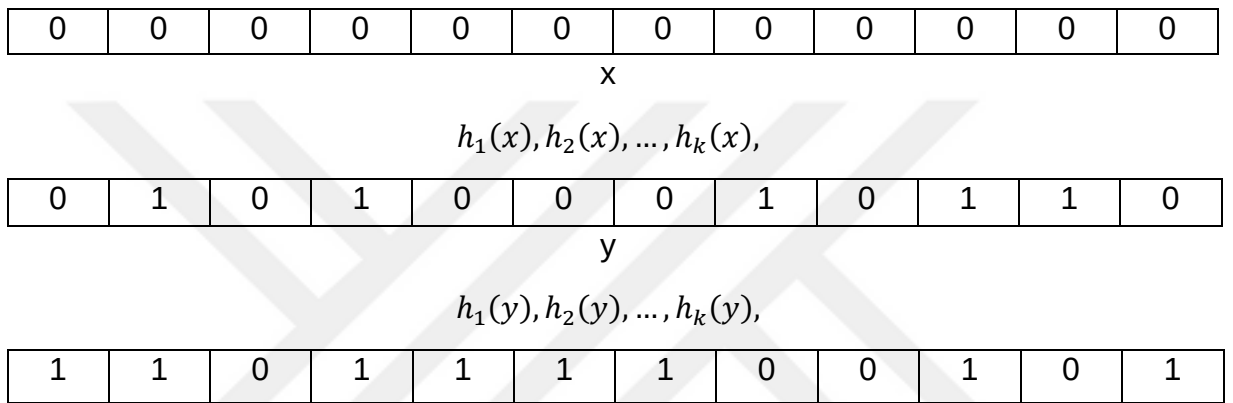
Artmakta olan veri miktarları bu veriler üzerinde gerekli işlemlerin yapılmasını zorlaştırmaktadır. Çok büyük miktarda veriler (Big Data) üzerinde istatistiksel analizler yapılması günümüz bilgi yoğunluğunda büyük önem kazanmaktadır. İstatistiksel analizleri gerçekleştirilmesi için Hadoop, MapReduce gibi maliyetli sistemler kullanılmaktadır. Tüm bunların yanında olasılıksal veri yapıları, düşük maliyetlerle veriler hakkında tahmini bilgi sağlama özellikleri ile öne çıkmaktadır. Olasılıksal veri yapıları veri kümesinde olan farklı eleman sayısı, en sık rastlanan elemanlar ve bu elemanların sıklığı/frekansı, kullanıcı tarafından belirlenen aralıkta olan eleman sayısı ve belirli elemanların bu veri kümesinde bulunup bulunmadığı ile ilgili bilgi sağlamaktadır.



Şekil 5.2. Ağaç tabanlı indis mimarisinin genel akışı

Son dönemlerde geliştirilen en verimli indis oluşturma algoritmalarından biri Bloom filtresidir. Bloom filtresi belge başına indis oluşturulması için kullanılan olasılıksal veri yapısıdır. Şifrelenmiş veriler üzerinde aralık sorgular oluşturulmasını ve bu sorgular için kanıtlanabilir güvence verilmesini desteklemektedir. Bloom filtresi yöntemi Burton Bloom [62] tarafından önerilmiştir ve ağ bağlamında kullanımı Marais ve Bharat [63] tarafından geliştirilmiştir. Şifreli bulut verileri üzerinde uygulanması Goh [22] tarafından üzerinde arama yapılan veri dosyalarının miktarına orantılı olarak iletişim maliyetinin azaltılmasına izin veren bir yöntem olarak geliştirilmiştir. Bu veri yapısı daha düşük maliyetle gönderilen sorguda kullanılan anahtar kelimelerin bulut sunucusunda depolanan veri dosyalarının

içinde yer alıp almadığını sorgulamaya izin verir. Bloom filtresinin tahmini kabul edilmesinin nedeni belirlediği sonuçlar içinde yanlış pozitifler içermesi ihtimalidir. Diğer bir deyişle, bloom filtresinin “evet” olarak döndürdüğü sonuç yanlış pozitif içerebilmesi nedeniyle yanlış olabilir. Ancak belgenin bulunmaması sonucunun döndürülmesi her zaman doğrudur. Bloom filtresinin düzgün yapılandırılması durumunda yanlış pozitif 1% altında tutulabilmektedir. Bu da, bulut sunucularında depolanan veri dosyalarında gereksiz disk erişiminin önüne geçebilmektedir. Bu tür kullanım şekli Google BigTable, Apache Cassandra ve Squid içerisinde kullanılmaktadır. Şekil 5.3’de Bloom Filtresinin genel yapısı gösterilmektedir.



Şekil 5.3. Bloom filtresinin çalışma prensibi

Bloom filtresinin dezavantajı, dinamik olmaması, yani veri dosyaları üzerinde sorgu gerçekleştirildikten sonra güncelleme yapılamaması idi. 2015 yılında Jie Wang ve ark. [7] tarafından bloom filtresi geliştirildi ve sıralama mantığını destekleyen ilk çoklu anahtar kelime ile bulanık arama yönteminde kullanıldı.

5.2.1. Bloom Filtresinin Çalışma Prensibi

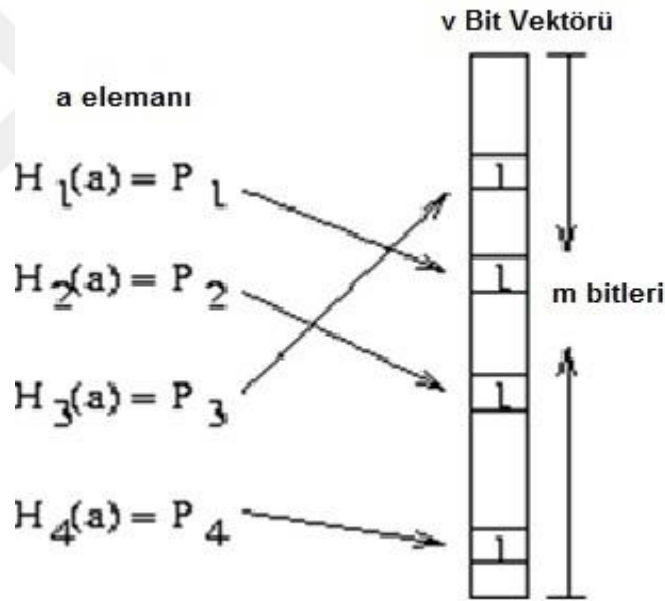
Oluşturulan bloom filtresi boş olursa, elemanları 0 olan m bitten oluşturulan bit dizisidir. Bir s farklı özet fonksiyonu tanımlanmışsa, bu fonksiyon her bir verilen a elemanı m bit dizisinin indisine rastgele olarak yerleştirecektir. Başka bir deyişle, bir a anahtarı (bu araştırma kapsamında, bulut sunucularında şifreli şekilde depolanan belgelerin işaretçileri) eklenirse veya silinirse, $c(h_1(a))$, $c(h_2(a))$, ..., $c(h_k(a))$ sayısı yükseltilebilir veya azaltılabilir. Bloom filtresi bit dizisine yeni bir eleman eklenmek için, bu eleman öncelikle $h_1(a)$, $h_2(a)$, ..., $h_m(a)$ şeklinde özetleme fonksiyonundan geçirilir, s indisi elde edilir, bloom filtresinin bit dizisine indislerde olan değerleri 1 olarak yerleştirir.

Burada not etmek gerekir ki, her hangi bir bit birden fazla defa 1 sayısına set edilebilir. Eğer bir elemanın mevcut olup olmadığı sorgulanmak istenirse, bu eleman s farklı özetleme fonksiyonundan geçirilir ve s indisi elde edilir. Eğer indislerde olan değerlerin en az biri 0 olursa, o zaman bu eleman veri dosyaları kümesinde bulunmamaktadır. Yalnız değerlerin hepsinin 1 olması durumunda bu elemanın kümede bulunduğu söylenebilir, ki bu da yanlış pozitifler nedeniyle kesin olarak söylenememektedir. Bloom filtresinde çeşitli bit dizisi boyutu, özetleme fonksiyonu değerleri doğru şekilde seçildiği takdirde, yanlış pozitif değeri oldukça küçülebilmektedir.

Bloom filtresinin yanlış pozitif içerme olasılığı

$$\left(1 - \frac{1}{m}\right)^{kn} \quad (7)$$

şeklinde hesaplanmaktadır.



Şekil 5.4. Dört özetleme fonksiyonu ile Bloom filtresi.

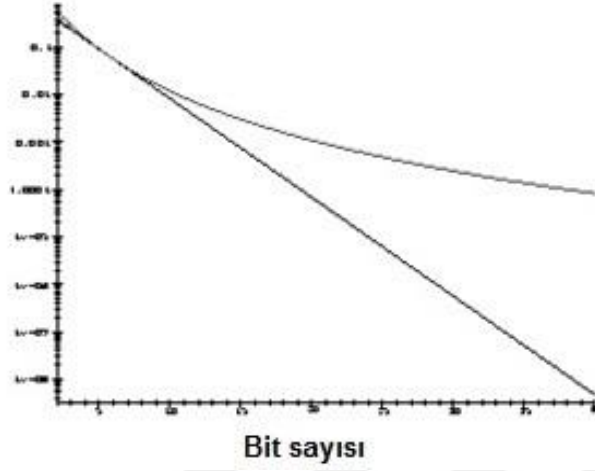
Burada m aralığı, n eleman sayısını, k ise özetleme fonksiyonlarını göstermektedir. Bu durumda, yanlış pozitif olasılığı

$$\left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx \left(1 - e^{-kn/m}\right)^k \quad (8)$$

Burada $k = \ln 2 \cdot \frac{m}{n}$ için sağ taraf minimize edilir ve

$$\left(\frac{1}{2}\right)^k = (0.6185)^{m/n} \quad (9)$$

Burada k değerinin tam değer olması gerekmektedir ve hesaplama yükünü azaltmak için optimum değer seçilmelidir.



Şekil 5.5. Yalrıř pozitif oranı.

Şekil 5.4'de ilk grafik 4 özetleme fonksiyonuna göre yalrıř pozitif oranını, ikinci grafik ise optimum sayıya uygun yalrıř pozitif değeri göstermektedir.

m/n	k	k=1	k=2	k=3	k=4	k=5
2	1.39	0.393	0.4			
3	2.08	0.283	0.237	0.253		
4	2.77	0.221	0.155	0.147	0.160	
5	3.46	0.181	0.109	0.092	0.092	0.101
6	4.16	0.154	0.0804	0.0609	0.0561	0.0578
7	4.85	0.133	0.0618	0.0423	0.0359	0.0347
8	5.55	0.118	0.0489	0.0306	0.024	0.0217
9	6.24	0.105	0.0397	0.0228	0.0166	0.0141
10	6.93	0.0952	0.0329	0.0174	0.0118	0.00943

Çizelge 5.2. Çeřitli m / n ve k kombinasyonları altında yalancı pozitiflik oranı

Not etmemiz gereken önemli detaylardan biri de, yanlış pozitif oranı fazla yüksek olduğu zaman, Bloom filtresinin rejenere edilebileceğidir ki, bu çok nadir rastlanan bir durumdur.

6. BENZERLİK TABANLI SIRALI ARAMA

6.2. Benzerlik Ölçüsü

Bulut sunucusuna aktarılan veri dosyaları üzerinde arama yöntemleri, kullanıcı tarafından oluşturulan sorgular ve belgeler arasında benzerliği tespit edilmesine dayanmaktadır. En önemli benzerlik ölçeklerinden biri Cao ve ark. [8] tarafından sunulan koordinat eşleştirme ilkesidir. Koordinat eşleşmesi, sorgu ile veri dosyasının eşleşmesini belirlemek için sorgu anahtar kelimelerinin sayısını kullanmış, birleştirme ve parçalama arama yöntemlerinin birleşmesi (hibrit) [64] bir ara benzerlik ölçüsüdür. Bu yöntem kullanıcıların çoklu anahtar kelime ile arama yönteminde belirttiği anahtar kelimelerin listesini belirleme ve döndürülen belgelerin en üst sırasına en yüksek alaka puanına sahip olan belgeleri yerleştirmek konusunda esnek bir yöntemdir. Ayrıca aramalar sırasında eşleşen koordinatları desteklemek için MRSA [8] şemalarında benzerlik ölçüsü olarak kullanılmıştır.

Arama gerçekleştirilirken gereken veri dosyalarını bulmak için sunucu oluşturulan sorguda kullanılan anahtar kelimelerin dosyalarda görülmesi frekansını ve belirlenen anahtar kelimelerin sayısını belirler. Veri dosyasında sorguda kullanılan anahtar kelimelerin hepsinin bulunması ve çok sayıda görünmesi belgenin alaka puanının yüksek olduğu anlamına gelir. Arama sonuçları olarak tüm anahtar kelimeleri veya bir kısmını içeren belgeler döndürülebilir. Daha açıklayıcı olması için aşağıdaki örnek incelenebilir:

{birleştirme, parçalanma, hibrit, eşleşme} kelimelerini içeren D sözlüğü oluşturulduğu varsayalım. Dış kaynağa aktarılmış iki X, Y dosyaları mevcuttur. X dosyası için ikili bir $ZX = \{1, 0, 1, 1\}$ indis vektörü olsun. Y dosyası için de aynı şekilde, ikili $ZY = \{1, 0, 1, 0\}$ indis vektörü olsun. Vektörler sözlükte olan anahtar kelimeyi içerirse 1, aksi takdirde 0 değerini alacaktır. Veri kullanıcısının ihtiyaç duyduğu dosyalara ulaşabilmek için $\{1, 0, 0, 1\}$ şeklinde bir sorgu oluşturulur. Bu bilgilerle, X ve Y belgelerinin benzerlik puanları olarak kullanıcının oluşturduğu Q sorgu vektörünün iç ürününü ve DX, DY vektörleri hesaplanabilir.

$$\text{BenzerlikPuanı } X = Q \cdot ZX = (1, 0, 0, 1) \cdot (1, 0, 1, 1) = 2$$

$$\text{BenzerlikPuanı } Y = Q \cdot ZY = (1, 0, 0, 1) \cdot (1, 0, 1, 0) = 1 \quad (10)$$

Sonuçlar incelendiği zaman, X dosyasının benzerlik puanının Y benzerlik puanından daha büyük olduğu gözlemlenmektedir. Benzerlik puanı daha yüksek olan belgeler diğerlerine tercih edilir. Ayrıca, bu bir $X > Y$ sıralamasını verir, yani kullanıcı belirli bir sayıda sonuç almak istediği zaman X belgesi Y belgesine göre sıralamadan daha yüksekte olacaktır. Koordinat eşleştirmesi benzerlik ölçüsünü kullanarak, etkili birden fazla anahtar kelime ile sıralı arama işlevselliği fark edilebilir. Bununla birlikte, bu tür ölçüm başlangıçta düz metin bilgisi almak amacı ile tasarlanmıştır ki, gizliliği ihlal etmeden şifreli veriler üzerinde arama yöntemlerine uygulanması zor bir sorundur.

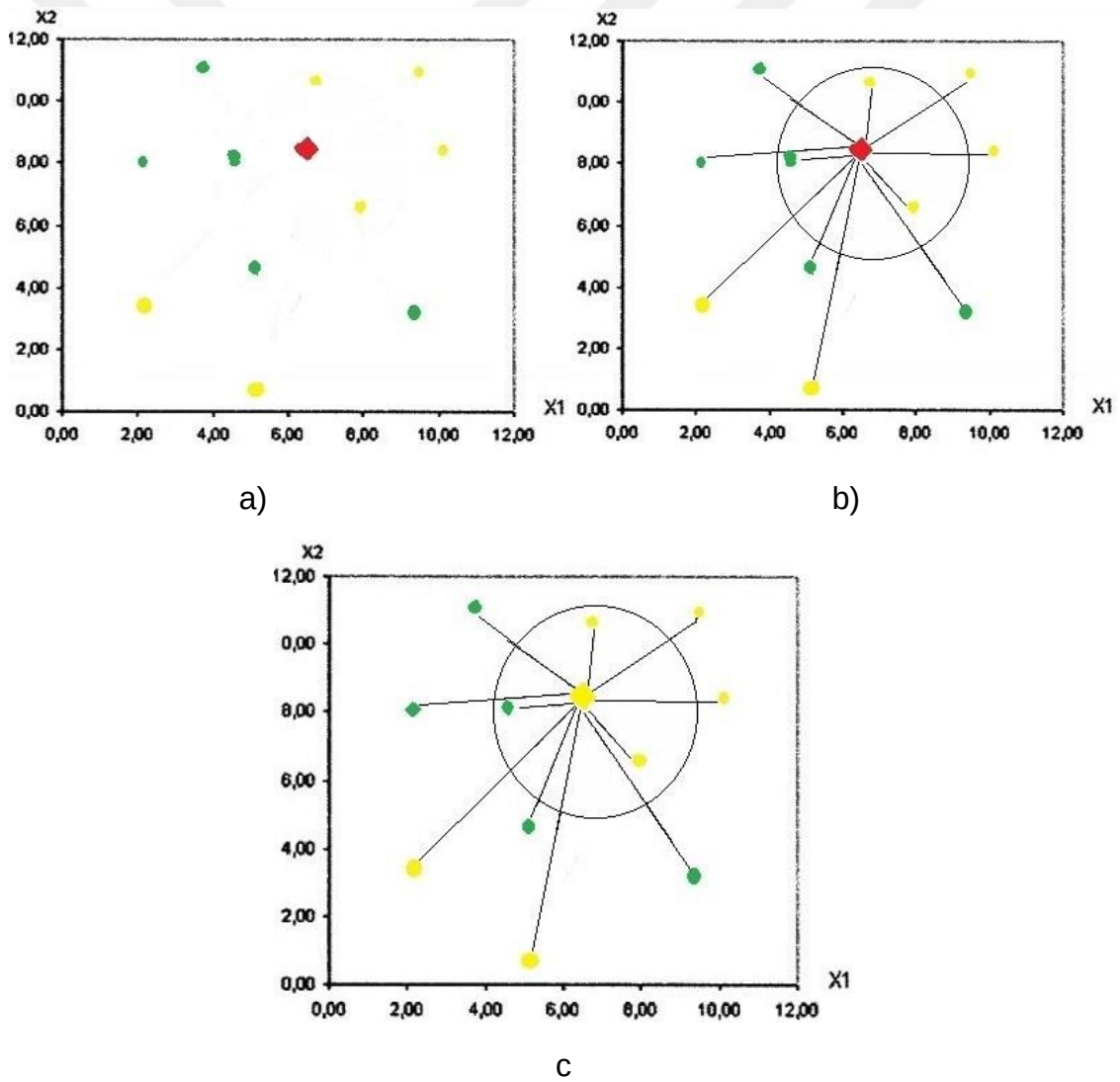
MRSE algoritmasında koordinat eşleşmesi yöntemi için özellikle, iç veri haberleşmesi kullanılmıştır. Daha açıklayıcı şekilde anlatmak gerekirse, bir belgede görülen sorgu anahtar kelime sayısı, arama sorgusu ile bu belgenin benzerliğini değerlendirmek için koordinat eşleştirmesi prensibini kullanır. Veri belge koleksiyonunda olan her bir belge ve istemciler tarafından bulut sunucusuna gönderilen arama sorguları alt indis gibi ikili bir vektör olarak tarif edilir. Veri belgeleri için her bit, sorgulanan anahtar kelimenin uygulanan belgede olup olmadığını gösterir. Arama sorgusunda her bit gelen anahtar kelimenin bu arama isteğinde olup olmadığını gösterir. Böylece, benzerlik ölçüğü tam veri vektörü ile sorgu vektörünün iç ürünü ile ölçülür. Veri vektörlerinin veya sorgu vektörünün bulut sunucusuna doğrudan aktarılması dosyaların indis güvenliği ve gizliliğini, ayrıca arama gizliliğini ihlal edebilmektedir. Bu güvenlik ihlallerini önlemek için, yöntemde güvenli k-en yakın komşu (kNN) tekniği kullanılmıştır. k-NN tekniği en basit eşleştirme metotlarından biridir. Cao ve ark. [8] sorgular sırasında Öklid mesafesi yerine iç ürün benzerliğini ölçmek için kNN tekniğini kullanmıştır. Bu benzerlik mesafesi sürekli tam eşleştirme ve değerlendirmeye uygun olarak azalır. Adım adım geliştirme aşamasının sonunda, d optimum sorgu aralığı E_d ulaşır ve kullanıcının gerek duyduğu belge sayısından daha fazlasının döndürülmesini engeller. K-en yakın komşu tekniği kullanılarak etkin bir sıralama sonucu elde edilir. Bu yöntem, arama sorgusuna üst k- en yakın komşularını tanımlar, gerekenden fazla gerekli dosya adayı üretilmesini engeller. Bu yöntem genel olarak verilerin yerel yapısına duyarlıdır ve Öklid mesafesi kullanılmaktadır.

$$d(i,j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (11)$$

K-en yakın komşu hesaplaması aşağıdaki gibi ifade edilir:

- K parametresi belirlenir
- Belirlenen nokta ile mevcut tüm noktalar arasındaki mesafeler hesaplanır.
- Bu noktalar arasındaki en küçük olan k nokta belirlenir.
- Seçilen noktaların kategorileri belirlenir ve en çok tekrarlanan değer seçilir
- Seçilen kategori aranan değer kategorisi olarak kabuledilir.

Örnek olarak, şekil 6.1'de gösterilen grafiğe incelenebilir. Burada, k = 3 olarak kabul edilmektedir.



Şekil 6.1. k – en yakın komşu: a) noktanın seçilmesi, b) belirlenen noktadan diğer noktalara olan mesafenin belirlenmesi, c) k – en yakın komşunun seçilmesi

Bu yöntemin en önemli özelliklerinden biri belge frekansının gizliliğini koruması ve anahtar kelimelerin yeniden belirlenmesi şansının azaltılması için benzerlik puanlarının rastgele olmasının sağlanmasıdır. Not edilmesi gerekir ki, gizli anahtar gizli tutulduğu sürece, indis ve geçit vektörü altında yatan şifresiz bilgiler sunucu tarafından ele geçirilemez. Ayrıca, arama doğruluğu ve güvenlik arasında esnek bir ödünleşim parametresi olarak, σ standart sapma işlevleri olur. Anahtar kelime gizliliğini korumak için, büyük bir σ sapması nihai benzerlik skorunun yanlış belirlenmesine neden olabileceği sebebiyle ve istatistiksel olarak bilgi elde edilmesinin zor olmasına rağmen, arama sonuçlarının daha kısıtlı döndürüleceği nedeniyle, daha etkin bir arama için küçük bir σ tercih edilmelidir.

Önerilen bir başka benzerlik yöntemi de vektör uzayı modelidir [65]. Vektör uzay modeli [33] düz metinlerde bilgi almak için uygulanan birçok benzerlik önlemleri arasında, birleşik ve ayırıcı arama yöntemini destekleyen ve çok kullanılan benzerlik modelidir. Özellikle, belge sıralaması işlevinde her belge vektörü ve sorgu vektörü arasında açılar, kosinüs değerlerinin sapması karşılaştırılarak gerçekleştirilmektedir. Kosinüs değeri “TF - IDF kuralına” dayanarak doğru sıralamayı sağlar.

Oluşturulan sorgular ve belgeler arasında eşleşme ve ilişkiyi belirlemek için kullanılan en yaygın tekniklerden biri de bu çalışmanın alt yapısını oluşturmak için de kullanılan Terim Frekansı - Ters Belge Frekansı ölçümüdür [4, 19, 65]. Bu ölçüler bilgi getirimi gibi işlemlerde sıralama yöntemi olarak kullanılmaktadır. Bu işlem dosyalarda geçen anahtar kelimeler ve dosyaların sayısını hesaplayarak çeşitli hesaplamaların yapılmasına izin vermektedir. Bu iki bir birinden bağımsız ölçek kullanılarak anahtar kelimelerle oluşturulan sorgularla eşleşen belgeler bulunur. Bu ölçümlerin hesaplanmasında iki çok önemli sayı bulunmaktadır. Bunlardan ilki üzerinde işlem yapılan dosyada olan belirlenmiş anahtar kelime sayısı, bir diğeri ise veri dosyaları kümesinde olan dosya sayısıdır.

Terim Frekansı (TF) belgelerde tekrarlanan anahtar kelimelerin frekansını temsil eden istatistiksel bir ölçüdür. TF değeri, belirli bir anahtar kelimenin tek bir belge içinde görülme sayısını hesaplar. Terim frekansını hesaplanması için öncelikle aranan anahtar kelimenin işlem yapılan dosyada kaç kere geçtiğidir. Gereken diğer bir sayı ise dosyadaki toplam terim sayısıdır. Bazen terim frekansını hesaplamak için toplam terim sayısı yerine belgede en çok geçen terim sayısı da

kullanılır. Ancak bu değerlerin normalleştirme için kullanılması nedeniyle her iki sonuçta geçerli olacaktır. Terim frekansı aşağıdaki gibi hesaplanır:

$$tf_{i,d} = \text{aza} (fr_{i,d} / df_i) \quad (12)$$

Burada *I* anahtar kelime, *d* üzerinde işlem yapılan dosya, *n* ise toplam dosya sayısıdır.

Ters Belge Frekansı (IDF) ise tüm veri dosyaları kümesinde bir anahtar kelimenin önemini gösteren ölçüdür. Belirli bir anahtar kelimeyi içeren belgelerin sayısı bir kümede olan belgelerin sayısına oranının logaritması olarak tanımlanır. Bu ölçümlerin hesaplanmasında iki çok önemli sayı bulunmaktadır. Bunlardan ilki üzerinde işlem yapılan dosyada olan belirlenmiş anahtar kelime sayısı, bir diğeri ise veri dosyaları kümesinde olan dosya sayısıdır. IDF değeri hesaplanırken bazı hususlar dikkat çekmektedir. Öncelikle, fonksiyonda olan logaritmanın tabanının önemi yoktur, çünkü hesaplamalar üssel bir fonksiyonun tersinin hesaplanmasıdır. Ayrıca, *tf* – *idf* değeri dosyaların kıyaslanması için hangi tabana göre hesaplandığının bir önemi olmamaktadır. IDF değeriyle ilgili dikkat edilmesi gereken bir başka önemli nokta da bu değer paydada yerleşmesidir. IDF değerinin 0 olması durumunda sıfıra bölüm belirsizliği oluşacağından genellikle bu değer için 1 ataması yapılmaktadır. Sonuç olarak *tf* – *idf* aşağıdaki gibi hesaplanır:

$$w_{i,d} = tf_{i,d} * \log(n/df_i) \quad (13)$$

TF-IDF değerini ölçmek için son dönemlerde geliştirilen ve bu çalışma kapsamında kullanılan fonksiyon aşağıdaki gibi ifade edilmektedir [35]:

$$\text{Değer} = \log (f_{i,j} + 1) \times \log \left(1 + \frac{N}{\sum_{k=1}^N X(f_{i,j})} \right) \quad (14)$$

Burada, $f_{i,j}$, *I* belgesinde *j* ifadesinin Terim Frekansını gösterir, *N* küme belgelerin toplam sayısıdır ve

$$\sum_{k=1}^n X(f_{i,j}) \quad (15)$$

tüm belgeleri toplamı arasında *j* termi için IDF değerini gösterir.

Tf – *Idf* değerinin hesaplanmasının bir ağırlık değeri olarak farklı türleri vardır. Çizelge 6.1’de bu hesaplama fonksiyonları verilmektedir:

Terim	Frekansı	Dosya	Frekansı	Normalleştirme
n (doğal)	$tf_{t,d}$	n	1	n
l (logaritma)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	$c(\text{kosinüs})$ $\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
A (Artan)	$0.5 + \frac{0.5 * tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N - df_t}{df_t}\}$	u(benzersiz pivotlu) $\frac{1}{u}$
B (boolean)	$\begin{matrix} 1 & tf_{t,d} \\ 0 & \text{diğer durumlarda} \end{matrix}$			b (bayt boyutu) $\frac{1}{CharLength^a} \quad a < 1$
L (log)	$\frac{1 + \log(tf_{t,d})}{1 + \log(ave_{t,d}(tf_{t,d}))}$			

Çizelge 6.1. Tf – idf değerinin hesaplanma fonksiyonları

Tf – idf yönteminde bir anahtar kelimenin dosyada kaç kere geçmesinin yanı sıra, hangi sayıda farklı dosyada geçmesi de önem taşımaktadır. Eğer bir anahtar kelime bir dosyada 50 kere geçmişse, bir başka anahtar kelime ise 10 farklı dosyada beşer kere geçmişse, Tf – idf ikinci değeri birincisinden daha fazla önem verir.

6.3. Sıralama Mantığı

Sıralama fonksiyonu belirli bir arama ile eşleşerek döndürülen dosyaların sorgu ile alaka puanlarının hesaplanması için kullanılır. Kullanıcının arama deneyimini geliştirmek ve daha etkin veri dosyalarının döndürülmesini sağlamak için, pratik şifrelenmiş veri arama düzeni tasarlanırken iki temel özelliğin de dikkate alınması gerekmektedir. Bunlardan ilki tek arahtar kelime yerine çoklu anahtar kelime ile arama yapmak iken, bir diğeri sıralı arama tekniğidir. Sıralama efektif şekilde verilere erişimi sağlamak ve daha kısa sürede daha iyi sonuçlar elde edebilmek için geliştirilen tekniklerinden biridir. Kullanıcıların sorgu gönderirken, döndürülecek belgelerin sayısını belirlemesine izin vermektedir. Bilgi erişim sistemlerinde, sıralama işlevi belirli bir arama isteğine eşleşen dosyaların alaka puanlarını hesaplamak için kullanılır. Arama algoritmasında bu mantık eşleştirme teknikleriyle birlikte ifade edilmektedir. İstemci ne zaman veriler için herhangi istek gönderirse, sorgu ile birlikte döndürülecek belirli bir sayıda belge döndürülür.

Sunucu sonuçların içinden önemlilik sırasının en üstünde olan dosyaların belirli bir miktarını istemciye döndürür.

“Kullandığın kadar öde” bulut paradigmasında sıralı arama işlevi tercih edilmektedir. Bunun nedeni, arama sonuçlarını istemciye aralarındaki farklılaşmamış sonuçları geri döndürmek yerine, sorgulama işleminde sıralama işlevi ile ilişki yaratabilmesi ve en uygun dosya dizisini döndürmesidir. Sıralı arama mantığı, bulut sunucusu ve veri kullanıcıları arasındaki ağ trafiğini önemli ölçüde azaltabilmektedir. Güvenli bir şekilde şifrelenmiş verilerin üzerinde arama yöntemlerinin tasarlanmasında bu özellik geliştirilmiş benzerlik önlemleri ile birleştirilerek uygulanmaktadır. Sonuç olarak, inşa edilen şifrelenmiş veriler üzerinde arama yöntemlerinde sıralama methodu kullanılarak düz metinler üzerinde mevcut işlemlerde olduğu gibi esneklik ve arama sonucu doğruluğu elde edilebilmektedir.

Sıralama mantığı Ning Cao ve ark. [8] tarafından 2011 yılında çoklu anahtar kelime ile arama yöntemi olan MRSE metodunda kullanılmıştır. İstemci tarafından oluşturulan sorgu ve dosya kümesi arasında, anahtar kelimelerin geçtiği dosyaları belirlemek için koordinat eşleşmesi metodu kullanılırken, belirlenen dosyaların sıralaması k-en yakın komşu algoritması kullanılarak gerçekleştirilmektedir.

Son zamanlarda, geliştirilen yöntemlerde eşleştirmenin yanı sıra, sıralama mantığının uygulanması için kullanılan yöntem $TF \times IDF$ fonksiyonudur [52]. Wang ve ark. [25] tarafından geliştirilen arama algoritmasında $TF \times IDF$ sıralama mantığı kullanılmıştır. Yukarıda da belirtildiği gibi $TF \times IDF$ şemasının çok fazla farklı varyasyonu mevcuttur. Bu mantığın mevcut olan farklı varyasyonları arasında, Li ve Wang tarafından geliştirilen arama algoritmasında kullanılan fonksiyon aşağıdaki gibi ifade edilmiştir:

$$Değer(Q, PT_d) = \sum_{t \in Q} \frac{1}{PT_d} * (1 + \ln f_{d,t}) * \ln \left(1 + \frac{N}{f_t}\right) \quad (16)$$

Burada Q aranan anahtar kelimeler sorgusun; $f_{d,t}$ PT dosyası içinde t anahtar kelimesini; f_t , t anahtar kelimesini içeren dosyalarının sayısını; N kümede mevcut olan toplam dosya sayısını; ve $|PT_d|$, PT dosyasının normalizasyon faktörü olarak işleyen indisli anahtar kelimelerin sayısının elde edilen uzunluğunu göstermektedir.

Sunulan en yeni çalışmalarda önerilen bir başka sıralama yöntemi metrik benzerlik olarak nokta ürünü yöntemidir. Mikhail Strizhov ve Indrajit Ray [66] tarafından 2015 yılında sunulan semantik arama algoritmasında ise kullanıcının oluşturduğu sorguların sıralanmış sonuçlarının sağlanması için metrik benzerlik ölçüsü kullanılmıştır. Sunulan yöntemde kullanıcı sorguları ve dosyalar arasında benzerliği belirlemek için yüksek boyutlu vektörler olarak temsil edilen vektör uzayı modeli [65] geliştirilmiştir. Önerilen benzerlik metriği her belge vektörü ve arama sorgusu vektörü arasındaki nokta ürünü olarak aşağıdaki gibi ölçülmüştür:

$$\text{dotprod}(D_i, Q) = D_i \otimes Q. \quad (17)$$

Burada D_i - i'inci belgeyi temsileden bir vektör, Q ise bir arama sorgu vektörüdür.

7. DÜZEN AÇIKLAMA ŞİFRELEME YÖNTEMİNE DAYALI ŞİFRELİ BULUT VERİLERİ ÜZERİNDE ARAMA ALGORİTMASI

7.1. Algoritmanın Genel Yapısı

Bulut sunucusuna aktarılan şifreli veriler üzerinde arama işlemi üç taraf arasında gerçekleştirilmektedir: veri sahibi, bulut sunucusu ve veri kullanıcısı. Veri sahibi rolünde tek bir kullanıcı veya bir kurumun olması mümkün. Veri sahibi veri belgeleri koleksiyonunu güvenli şekilde depolamak için bulut sunucularına göndermeden önce, her dosya için indisler oluşturur, daha sonra bu indisleri ve belgeleri belirli aranabilir şifreleme yöntemi kullanarak şifreler ve bulut sunucusuna gönderir. Bu tez çalışması kapsamında, geliştirilen şifreli veriler üzerinde arama algoritmasında dosyaları şifrelemek için özellik koruma şifreleme yönteminin 2015 yılında Nathan Chenette ve ark. [67] tarafından sözde rastlantısal fonksiyonlar kullanılarak geliştirilen düzen açıklama şifreleme yöntemi kullanılmıştır. Düzen açıklama şifreleme yöntemi sorgular sırasında bilgi sızdırılmasını engeller ve verimli arama yapmaya izin verir.

Şifreli veriler üzerinde arama sorgularını gerçekleştirmek için dosyaları şifrelemeden önce veri sahibi tarafında indis oluşturulur. Bunun için Goh [22] tarafından geliştirilmiş olan Bloom filtresini kullanıldı. Bloom filtresini kullanılması bazı elementlerin kümedeki üyelerden biri olup olmadığını belirlemeye de izin vermektedir. Eleman kümenin üyesi ise 1, olmadığı takdirde 0 olarak işaretlenir. Bloom filtresini Jie Wang 2015 [50] çalışmasında sunduğu şekilde genişlettik ve

bloom filtresinde bazı uygun karma fonksiyonları pozisyona duyarlı karma fonksiyonları ile değiştirdik. Karma fonksiyonları Wang tarafından sunulan yöntemde kullanıldığı gibi p – kararlı LSH fonksiyonları ile değiştirdik.

Günümüzde git gide artan veri kümeleri içinden kullanıcıların en uygun dosyaların sınırlı sayısını elde edebilmeleri için arama yöntemlerine sıralama özelliğini eklemek amacıyla terim frekansı – ters belge frekansı (TF - IDF) yöntemini kullandık. TF – IDF yöntemi kullanıcı tarafından oluşturulan ve sunucuya gönderilen sorguyla eşleşen dosyaları (anahtar kelimeleri içeren belgeleri) belirlemenin yanı sıra, uygun dosyaların benzerlik puanlarını belirler.

Kullanıcıların sorgularını oluşturdukları zaman yanlış anahtar kelimeler dahil etmesi durumlarını da göz önünde bulundurarak, oluşturulan sorgularda aralık sorgu elde etmek için her hangi belgenin işaretleme indisini oluşturmak için ikinci katman indis oluşturmada normal Bloom filtresi kullanıldı.

7.2. Pozisyona Duyarlı Özetleme Fonksiyonu

Bu çalışmada, [21]'de açıklanan p-kararlı LSH özetleme fonksiyonları kullanıldı. Pozisyona duyarlı özetleme fonksiyonu ile sonuca ulaşabilmek için pozisyona duyarlı fonksiyon ailesi kullanılır. Pozisyona Duyarlı Özetleme işlemleri ana bellekte gerçekleştirilebilmektedir, ancak bazı yanlış pozitifler oluşacaktır. Özetleme yöntemleri hızlı yakın komşu (kNN) aramaları geliştirilmesi için çok uygundur. Özetleme fonksiyonu ile karma tablosunda benzer örneklerle birlikte lineer zamanlı arama yapılabilir. Verilerin rastgele tahminlerini elde etmeye izin verir. Güçlü teorik garantiye sahiptir. Pozisyona Duyarlı Özetleme işlemlerinin arkasındaki temel fikir düşük boyutlu ikili (Hamming) uzaya veri yansıtmaktır; karma anahtarı denilen her bir veri noktası b-bit vektörüne eşleştirilir. Her bir özetleme fonksiyonu pozisyona duyarlı özetleme fonksiyonun özelliklerine sahip olmak zorundadır:

$$\Pr[h(x_i) = h(x_j)] = \text{sim}(x_i, x_j) \quad (18)$$

burada, $\text{sim}(x_i, x_j) \in [0, 1]$ benzerlik fonksiyonudur.

LSH işlevleri [21] özetleme fonksiyonları ailesinden aralık dışındakilere nazaran bazı aralıklarda olan yakın öğelerin daha yüksek olasılıkla karmasını oluşturan

özgün karma fonksiyon dizini olarak tarif edilir. Herhangi iki öge göz önüne alındığında:

Temel fikir, objelerin benzer oldukları zaman aynı parçaya (bucket), benzer olmayanların farklı bölgelere aktarılmasını sağlamaktır. Bu fonksiyonları daha iyi anlayabilmek için şöyle örnek verilebilir: x, y ögeleri mevcuttur. Eğer x, y aşağıdaki gibi hesaplanırsa, karma fonksiyon ailesi $H:(r1,r2, p1, p2)$ karşılamaktadır:

$$d(x, y) \leq r1 \text{ olduğunda; } \Pr[h(x) = h(y)] \geq p1$$

$$d(x, y) \geq r2 \text{ olduğunda; } \Pr[h(x) = h(y)] \leq p2 \quad (19)$$

burada, $d(x, y)$ Öklid uzayında x ve y arasında mesafenin hesaplanmış sonucudur. LSH hash aileden seçilen her hash fonksiyonu bu şekilde ifade edilir [7]:

$$h_{\vec{x},y}(\vec{v}) = \left\lfloor \frac{\vec{x} \cdot \vec{v}}{z} \right\rfloor \quad (20)$$

x, v vektörleri düzgün seçilen rastgele değişkenler tarafından oluşturulur ve y sayısı $[0, z]$ 'dan rastgele seçilen bir sayıdır. Bu da alanda düz bir çizgini bölen bir çizgi parçasının uzunluğunu ifade etmektedir.

7.3. Algoritmada Bloom Filtresinin Yeri

Geliştirilen arama yönteminde her belge için indis oluşturulması için veri dosyalarına iki katmanlı Bloom Filtresi uygulanmış ve Bloom Filtresinde mevcut olan hash fonksiyonlarına pozisyona duyarlı özetleme fonksiyonları eklenmiştir. Bunun için Wang tarafından sunulan p- kararlı LSH fonksiyonları kullanılmıştır. Her iki indisin aynı yapıyı kullandığı düşünülürse, karışık indis algoritmalarıyla karşılaştırıldığında bu yöntem çok daha sadedir. Şekil 7.1'de kullanılan indis oluşturma algoritmasının genel akış şeması verilmiştir.

Bloom filtresi başlangıçta her pozisyonda bağımsız özetleme fonksiyonları ile 0 olarak belirlenmiş m bit dizisidir ve her elementin eşleştirmesini yaptıktan sonra 1 olarak belirlenmektedir. Genel olarak, $H = \{h_i | h_i : \{0,1\}^* \rightarrow [1,m], 1 \leq i \leq l\}$ gibi hash fonksiyonlar dizisi verildiği zaman, eğer $a \in S$, m-bit dizisinde $h_i(a)$ -cı 1 olarak, aksi takdirde, 0 olarak ayarlanır. Tüm pozisyonlar 1 ise, a unsurunun S unsurlardan biri olması çok muhtemel olduğu, aksi takdirde S dışında olduğunu sonucuna varabiliriz.

Yanlış pozitif oranı aşağıdaki şekilde hesaplanabilmektedir:

$$\text{Yanlış Pozitif Oranı} = L(1 - e^{-\ln m}) \quad (21)$$

Bu ifadenin sonucu $(\frac{1}{2})L$ gibi belirlenir.

Bloom Filtrende şifreli dosyaların şifreli etiketlerin değerlerinden ziyade sadece şifreli metin aralığı ile ilişkilidir. Bloom Filtresi ile indis oluşturulurken, Puan Tablosu isimli bir tablo oluşturulur. PuanTablosu anahtar kelime işaretçi, onun sıralamasını puanı ve diferansiyel şifreli tablosu da dahil olmak üzere özgül bir veri kümesidir. Bu tablo bulut sunucusuna aktarılması gereken normal metinden çıkarılan her anahtar kelime için $TF \times IDF$ değerlerinin farklı bölümlerini içerir. PuanTablosu aynı anahtar kelimeleri içeren benzer dosyaların sonucunun sıralamasında çok önemli bir rol oynar.

7.4. Düzen Açıklayıcı Şifreleme

Verilen λ güvenlik parametresidir. n-bit pozitif tamsayılar kümesi için düzen açıklayıcı şifreleme yöntemine uygun kaçak fonksiyonu aşağıdaki gibidir:

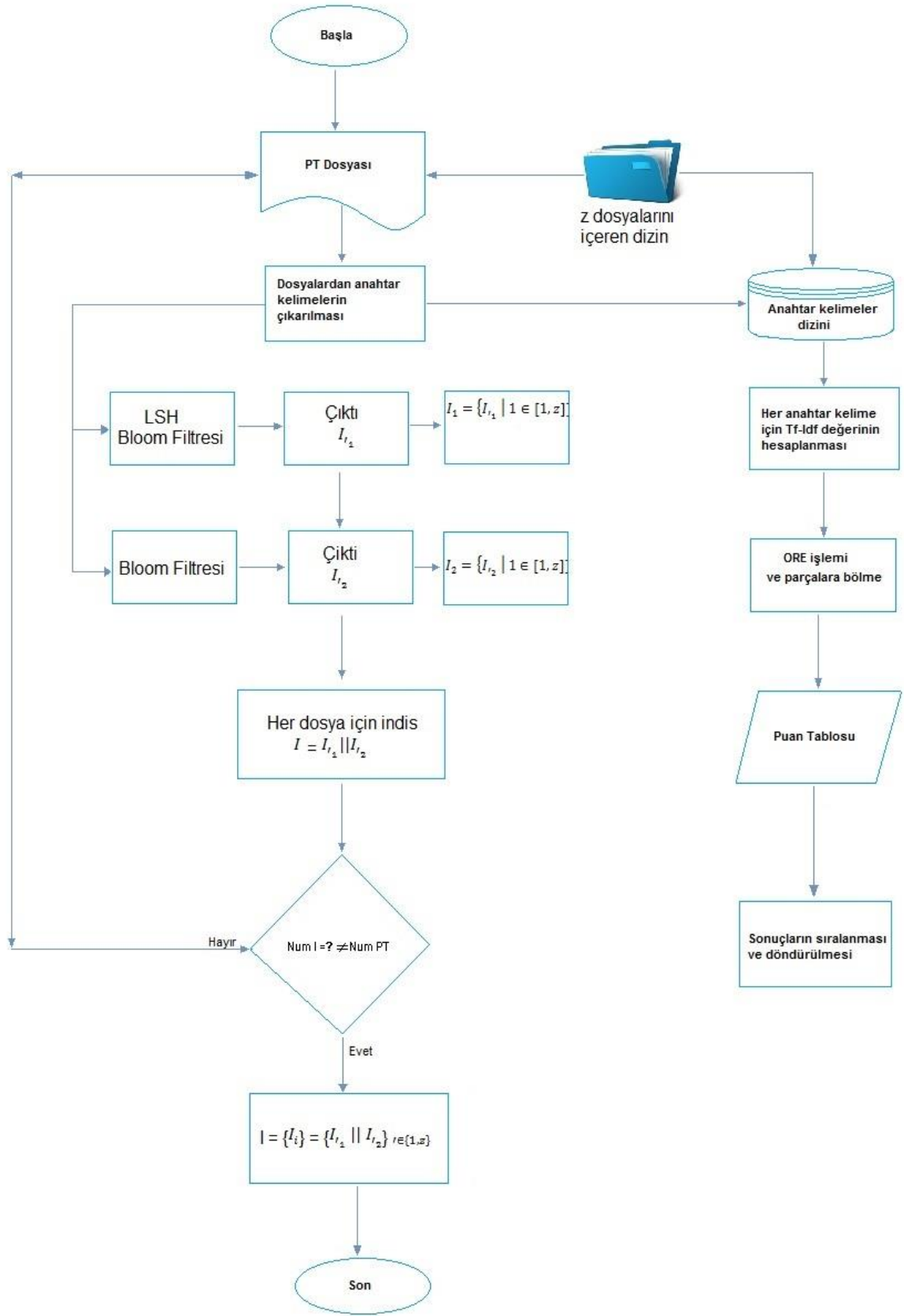
$$\zeta_f(m_1, \dots, m_t) = \{(\text{ind}_{diff}(m_i, m_j), 1(m_i < m_j)) : 1 \leq i < j \leq t\} \quad (22)$$

burada $\text{ind}_{diff}(x, y)$, x ve y farklı olduğu zaman ilk bit için indisi verir. Eğer $x = y$ olursa, $\text{ind}_{diff}(x, y) = n + 1$ (23) olur.

Burada $\lambda \in N$ güvenlik parametresi ve $M > 3$ belirlenir. $F : K \times ([n] \times \{0, 1\}^{n-1}) \rightarrow Z_M$ güvenli sözde rastlantısal fonksiyondur. Düzen açıklayıcı şifreleme yöntemi $\Pi_{ORE} = \{\text{ORE.Kurulum}, \text{ORE.Şifreleme}, \text{ORE.Karşılaştırma}\}$ algoritmalarından oluşur:

- ORE.Kurulum (1^λ): Bu algoritma fonksiyon için rastgele homojen sözde rastlantısal fonksiyonun k anahtarını belirler. Belirlenen gizli anahtar $sk = k$ olur.
- ORE.Şifreleme (sk, m): $b_1, \dots, b_n$ m için ikili gösterimi olsun ve her bir $i \in [n]$ için şifreleme algoritması
$$u_i = F(k, (i, b_1 b_2, \dots, b_{n-1} || 0^{n-1})) + b_i \pmod{M} \quad (24)$$

$$(u_1, u_2, \dots, u_n) \text{ çıktıları olur.}$$



Şekil 7.1. İki katmanlı indis şemasının genel akış diyagramı

- ORE.Karşılaştırma (ct_1, ct_2): bu algoritmanın ayrışması

$$\begin{aligned} ct_1 &= (u_1, u_2, \dots, u_n) \\ ct_2 &= (u'_1 u'_2, \dots, u'_n) \end{aligned} \quad (25)$$

burada $u_1, u_2, \dots, u_n, u'_1 u'_2, \dots, u'_n \in Z_m$. $u_i \neq u'_i$ olduğunda, i en küçük indis olur. Böyle bir indisin olmaması durumunda, çıktı olarak 0 döndürülür, aksi takdirde ise, değer olarak 1 döndürülür ve $u'_i = u_i + 1$ olur. Düzen açıklayıcı şifreleme yöntemi F fonksiyonunun sözde rastlantısal fonksiyon güvenliğine göre kaçak fonksiyonu ile güvenlidir.

7.5. Sıralama Fonksiyonu

Sıralama fonksiyonu anahtar kelime arama isteğine göre farklı dosyalarda kelime ağırlığına dayanarak döndürülen sonuçlarının alaka puanları ölçmek için anahtar fonksiyondur. Şifreli metinler üzerinde sayısal sıralama elde etmenin ana noktası belirli anahtar kelimelerine Bloom Filtre ekleyerek, diğer bir deyişle sorgu öğelerinin bit dizisinde olduğunu belirleyerek gerçek sayı karşılaştırmalarının değiştirilmesine ihtiyaç duymasıdır. Bilgi alma alanında, $TF \times IDF$ üzerinde işlem yapılan dosyada anahtar kelime ağırlığını belirlemek için en geniş kullanılan ilkedir. Genelliği kaybetmeden, bu çalışmada literatürde [k30] detaylı bir şekilde anlatılan aşağıdaki alaka puanı hesaplama denklemini kabul edilmiştir:

$$Değer(k, F_i) = \frac{1}{|F_i|} * (1 + \ln f_{i,k}) * (1 + \frac{N}{N_k}) \quad (26)$$

Burada, d - anahtar kelime, $tf_{i,k}$ - dosyada olan anahtar kelime için TF değeridir. $|F_i|$ - F_i fonksiyonunun uzunluğu, N - bulut sunucusuna aktarılan dosya kümesinde olan veri dosyalarının sayısını belirlemektedir.

7.6. Algoritmanın aşamaları

7.6.1. Birinci Aşama

Her dosyaya göre indislerin oluşturulması için, öncelikle, m bit dizini ve p kararklı LSH fonksiyonlarının alt kümesi kullanıldı. Bulut sunucularının anahtar kelimeler kullanılarak oluşturulan geçitlerin kullanılması ile önemli bilgilere ulaşabildiğini ve yapılan aramalara uygun istatistiksel sonuçlar elde edebileceğinden anahtar kelimeler ve geçitler arasında ulaşılacak açık ilişkilerin olmamasına dikkat edilmesi gerekmektedir. LSH özetleme fonksiyonlarını kullanarak geliştirilmiş

güvenli indisi oluşturmada önce, N - gram vektör dizinine karşılık gelen N parametresinin kullanıldığı anahtar kelimelerin $f: \{0,1\}^{26N} * \{0,1\}^* \rightarrow \{0,1\}^*$ sözde rastlantısal fonksiyonu kullanılmıştır. Bu bilinen arka plan modeli gibi istatistiksel saldırılarda indis ve geçişte terim frekansı gibi önemli içeriklerin belirlenmesi olasılığını azaltır.

- KeyGen $(1^f, 1^V)$: f güvenlik parametresi verilmiştir, $D = (M_1, M_2, k)$ anahtarları oluşturulur. Burada $M_1, M_2 \in R_{f*f}$ ters matrislerdir, $V \in \{0, 1\}^f$ - İndisOluşturma(PT, K, l) işlemi sırasında indis vektörünün 2 parçaya bölünmesi için belirli bir talimat vektörüdür. Diğer bir r güvenlik parametresi için $D' = \{d_i | d_i \xleftarrow{R} [0,1]^s, 1 \leq i \leq l\}$ anahtarlar havuzu oluşturulur.
- İndisOluşturma (D, K, l): Homojen şekilde, H özetleme fonksiyonları ailesi p kararlı LSH ve sözde rastlantısal fonksiyonlarından l bağımsız LSH fonksiyonu seçilir. Bu fonksiyon

$$f_{d_i} = \{d_i, \xleftarrow{R} \{0,1\}^s: \{0,1\}^{26k} * d_i \rightarrow \{0,1\}^*\} \quad (27)$$

H ailesi p kararlı LSH ve sözde rastlantısal fonksiyondan l bağımsız LSH fonksiyonu seçildi. Her bir p_i dosyası için güvenli indis oluşturulması aşağıdaki aşamalarla gerçekleştirildi:

1. $PT = \{p_i | i \in [1, z]\}$ dosyalarından anahtar kelimeler çıkarılır ve $D = \{d_i | i \in [1, v]\}$ anahtar kelimeler dizini oluşturulur. Burada v - dosya dizininde olan maksimum anahtar kelime sayısıdır. Daha sonra, bu d_i anahtar kelimelerin bu anahtar kelimeler uygun V_{k_i} vektörlerine dönüştürülmesi için tablo 9'da verilen algoritma kullanılır (burada $d_i \in [0,1]$)
2. Her bir k gram vektör permütasyonunun konumunu rastgele şekilde kullanmak için f_k sözde raslantısal fonksiyonu kullanılır ($\forall k_i \in K$). Sözde rastlantısal fonksiyon anahtar kelimeler arasında olan ilişkiyi ve benzerliği en aza indirir.
3. Her dosya için p kararlı LSH fonksiyonları sınıfından belirlenen l bağımsız özetleme fonksiyonlarını içeren I_{i_1} m bit Bloom filtresi oluşturulur ve her bir k_i anahtar kelimesi I_{i_1} fonksiyonuna eklenir.

4. $Enc_K(I_{i_1}) - I_{i_1}$ fonksiyonu K gizli anahtar kümesi kullanılarak şifrelenir. I_{i_1} fonksiyonu bölünme kurallarına uygun olarak (I'_{i_1}, I''_{i_1}) şeklinde bölünür. Burada her bir $i_k \in I_{i_1}$ için, $S_k \in 1$ olduğunda $I'_{i_1} = I''_{i_1} = I_{i_1}$ olacaktır. $S_k \neq 1$ olduğunda

$$I'_{i_1} = \frac{1}{2} I_{i_1} + r \text{ ve } I''_{i_1} = \frac{1}{2} I_{i_1} - r \quad (28)$$

burada $r \in \{0,1\}^*$ homojen seçilmiş rastgele sayıdır. Bundan sonra, (I'_{i_1}, I''_{i_1}) ayrılmış indislerini (M_1, M_2) kullanılarak şifrelenir.

$$Enc_K(I_{i_1}) = (M_1^T * I'_{i_1}, M_2^T * I''_{i_1}) \quad (29)$$

Bu ifade sonucunda her dosya için şifreli indis oluşturulmuş olacaktır.

5. Son olarak her bir dosya için benzerlik puanını hesaplamak için önceden ifade ettiğimiz denklemden terim frekansı * ters belge frekansı hesaplanır ($DeğerW_i$). Bu değer düzen açıklayıcı şifreleme yöntemi kullanılarak şifrelenir. $Enc_{ORE}(DeğerW_i)$ her sütunu bir kısmıyla doldurmak için R_i^* olarak m dengeli kısma bölünür. Burada, anahtar kelime dizininden $i \in [1, v]$. Bu değerler $DeğerTablosu$ 'na yerleştirilir.

- $Geçit_Gen(Q, K)$: bu işlem önceden anlattığımız BuildIndex işlemiyle aynı şekilde gerçekleştirilir. Boş m bit Bloom filtresi oluşturulur ve $V_Q Q$ sorgu vektörünün oluşturulması için aynı f_k ve h_i fonksiyonları kullanılır. $V_Q Q \supset \{W_j | j \in [1, x]\}$ sorgu vektöründe x sorguda kullanılan anahtar kelimelerin sayısını ifade eder. Oluşturulan sorgunun Q_v vektörü düzen açıklayıcı şifreleme yöntemi kullanılarak şifrelenir: $Enc_{Geçit}(Q_v)$. Her bir $q_k \in Q_v$ için Q_v fonksiyonu bölünme kurallarına uygun olarak (Q'_v, Q''_v) şeklinde bölünür. Burada $S_k = 0$ olduğunda $Q'_v = Q''_v = Q_v$ olacaktır. $S_k \neq 0$ olduğunda

$$Q'_v = \frac{1}{2} Q_v + r \text{ ve } Q''_v = \frac{1}{2} Q_v - r \quad (30)$$

burada $r \in \{0,1\}^*$ homojen seçilmiş rastgele sayıdır. Bundan sonra, (Q'_v, Q''_v) ayrılmış sorgu vektörleri (M_1, M_2) kullanılarak şifrelenir.

$$Enc_{geçit}(Q_v) = (M_1^{-1} * Q'_v, M_2^{-1} * Q''_v) \quad (31)$$

Bu işlem sonucunda kullanıcının sonuçları elde etmek için bulut sunucusuna gönderdiği şifrelenmiş sorgu elde edilir.

- Arama ($Enc^K(I_{i_1}), Enc^{Geçit}(Q_v)$): W_s sorguda kullanılan anahtar kelimedir ve bu anahtar kelimenin işaretçisi belgeler arasında benzerlik sıralama puanı hesaplamak için talimatçı olarak kullanılmaktadır. İç çarpımı hesaplanır ve her dosya için anahtar kelime sütunları (birinci ve ikinci) seçilir. Hesaplama aşağıdaki gibi yapılmaktadır:

$$\text{Benzerlik Puanı} = (M_1^T * I'_{i_1} * M_1^{-1} * Q'_v + M_2^{-1} * Q''_v * M_2^T * I''_{i_1}) * \text{Değer}_{w_i \in Q} = (I'^T_{i_1} * Q'_v + I''^T_{i_1} * Q''_v) * \text{Değer}_{w_i \in Q} = I_{i_1} * Q_v * \text{Score}_{w_i \in Q} \quad (32)$$

- ŞifreÇözme(C, K): Anahtar kullanılarak sunucu tarafından döndürülen şifreli sonuçların şifresi çözülür ve kullanıcının sorgusuna uygun sonuçlar benzerlik puanının en büyük olanından başlayarak elde edilir.

7.6.2 İkinci Aşama

Daha basit bir sistem kullanarak aralık sorgu oluşturulmasında indis yapısının ikinci katmanı kullanıldı. m - bit dizisine eklenen elemanların düzenlenmesinde şifreli veriler için aralık sorguların oluşturulmasında normal Bloom filtresi kullanılmıştır. Burada indis ve geçit oluşturulması için $f_{K''}(.): \{0,1\}^{K''} * \{0,1\} \rightarrow \{0,1\}^*$ sözde rastlantısal fonksiyon kullanılmıştır. Çok sayıda şifreli veriler üzerinde aralık sorgu ile verimli arama yapmak için aşağıdaki polynomial zamanlı algoritmalar kullanılmıştır.

1. KeyGen(1λ): r güvenlik anahtarı girdi olarak kullanılır ve $K \xleftarrow{R} \{0,1\}^*$ gizli anahtarı seçilir.
2. $S_{PT} \leftarrow Enc(K, ID_{D_i}, S_{PT}, l, m)$ –K gizli anahtarı kullanılır, PT dosyasının tanımlayıcısı ID_{PT_i} , S_{PT} – PT dosyalarının sayısal etiketleri ve Bloom filtresi için m,l parametreleri girdi olarak kullanılır ve PT dosyaları için ST aranabilir şifreleme etiketi oluşturulur.
3. $s, k_k \leftarrow Geçit(K, w', O, T_{type})$: K gizli anahtar, aranan sayısal w' , dosya etiketleri için operatorlar (O), T_{type} aranabilir etiket türleri girdi olarak verilir ve anahtar kelimeler için tr geçiti ve tüm belgeler için ikinci katman indisinde I_2 için k_k konumlandırma karakteri çıktı olarak verilir. s aranabilir

belirtecinin üretmek için aynı işlem prosedürü uygulanır ve bunun için $f(K, O || w')$ sözde rastlantısal permütasyonun çıktısıdır ve q indisin ikinci katmanında Bloom filtresi T_{type} aranabilir etiketin türü ve O operatörleri ile hesaplanır.

4. $ID_{w'} \leftarrow \text{Arama}(\text{SC}, \text{tr}, Q)$: Aralık sorguya uygun sonuçlar için şifreli sayısal etiketlerle arama tamamlanır.
5. $\text{ŞifreÇözme}(C, K)$: Döndürülen şifreli metinler gizli anahtar kullanılarak çözülür ve düz metin elde edilir.

7.7. Sonuçların Analizi ve Karşılaştırma

Şifreli bulut sunucuları üzerinde sunulan bu arama algoritmasının analizlerinin ve denemelerinin yapılması için yerel dizüstü bilgisayar ve yerel sunucu kullanılmıştır. Belirtilen sonuçlar test platformunda yapılan denemelerin sonuçlarına uygun olarak verilmektedir. Bulut ortamını oluşturmak yerine algoritmanın gerçek performansının denenmesi için yerel ortam daha uygun bulunmuştur.

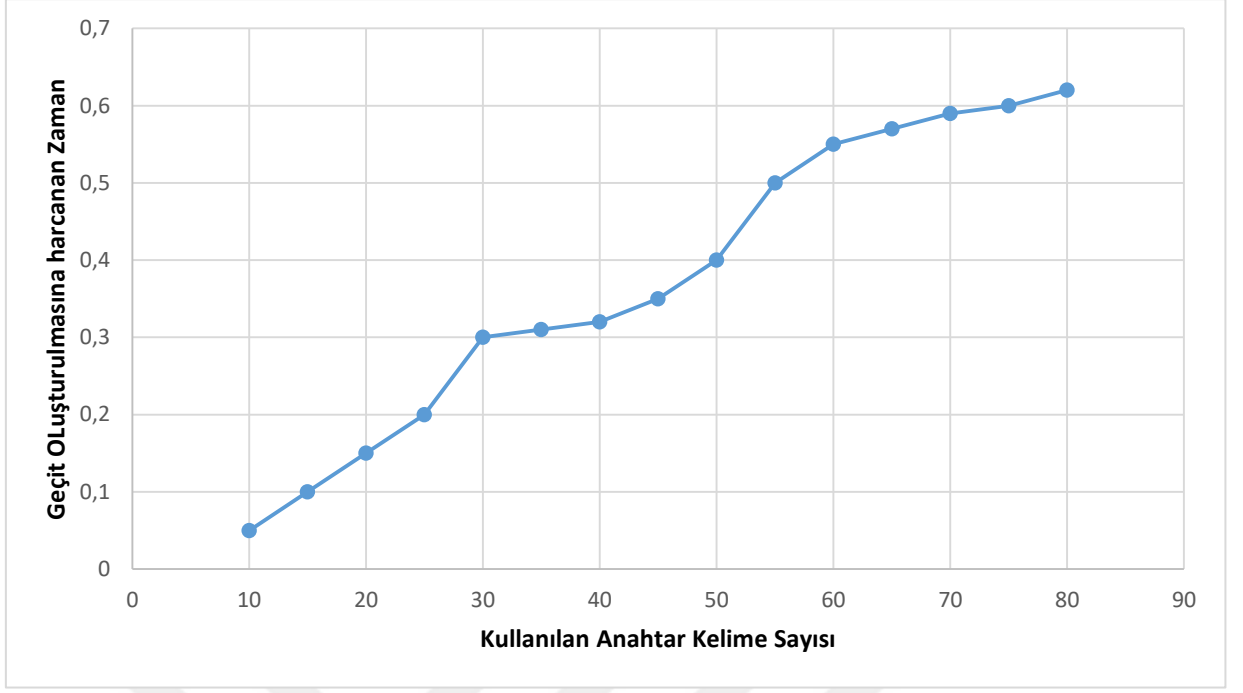
Aşağıda arama algoritması Wang [27] şemasının ve Jie Wang [50] tarafından sunulan yöntem arasındaki ilişki ve farklı tarafları açıklanmaktadır.

- Sunulan yeni yöntemde Jie Wang [50] tarafından teklif edilen bulanık arama yönteminden farklı olarak bulut sunucusuna gönderilmeden önce dosyaların şifrelenmesi için düzen açıklayıcı şifreleme yöntemi kullanılmış ve güvenlik açısından iyi sonuçlar elde edilmesinin yanı sıra, optimum zamanda arama sonuçları elde edilmiştir. ORE ve TF-IDF kullanılarak oluşturulan sistemin arama sonuçlarını daha verimli sürede döndürdüğü ve sıralama mantığını için daha uygun sonuçlara ulaştığı belirlenmiştir.

Düzen açıklayıcı şifreleme yöntemi ve dosyalar arasındaki eşleşme ve benzerliği belirlemek için kullanılmış olan terim frekansı - ters belge frekansı yöntemi sunulan bu algoritmanın Wang tarafından sunulan yöntemden farkını göstermektedir.

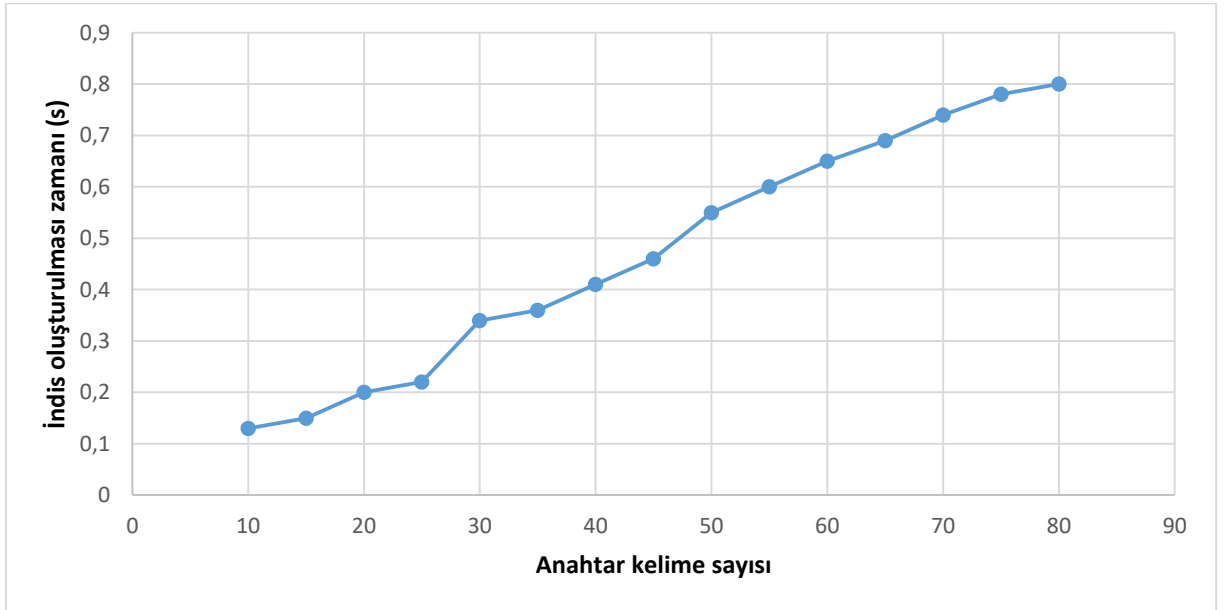
7.7.1. Verimlilik

Şekil 7.2 ve 7.3'de önerilen yöntemde geçit ve indis oluşturulurken harcanan zamanı göstermektedir. Doğal olarak, tek katmanlı indis oluşturma şemalarından

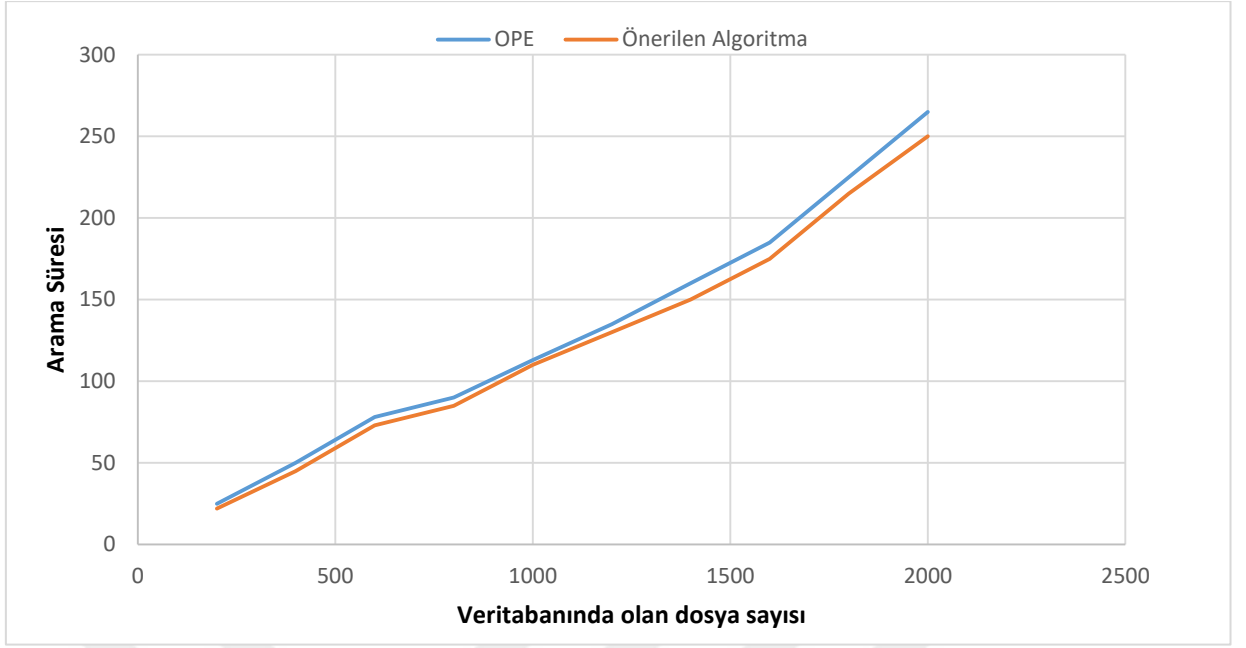


Şekil 7.2. Geçit oluşturulması süresi

farklı olarak, iki katmanlı Bloom filtresi oluşturulması daha fazla zaman gerektirmektedir. Bloom filtresinin birinci katmanının oluşturulmasına gereken zaman geçit oluşturulması için gereken zamanla aynıdır. İndis oluşturulurken ikinci katmanın oluşturulması kullanıcıların veri dosyalarının güvenliğini artırırken sorguların sonuçları döndürmesi için daha fazla süre harcanmasına neden olmaktadır.

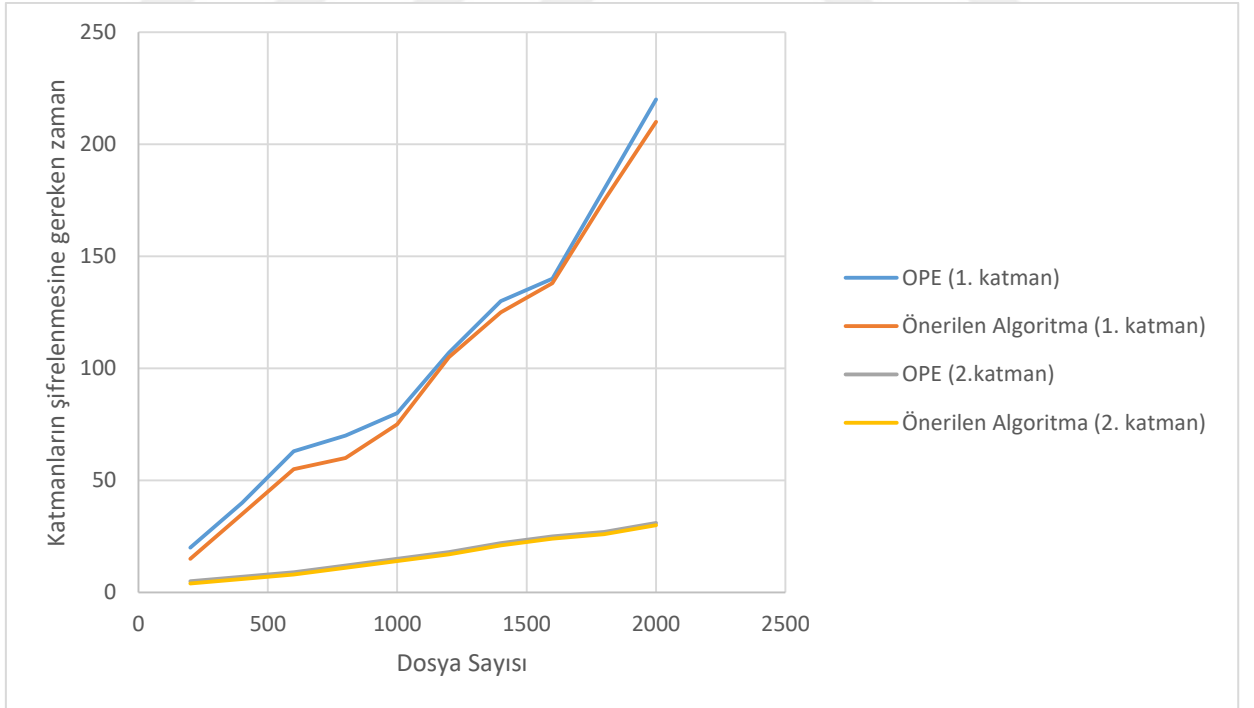


Şekil 7.3. İndis oluşturulması süresi



Şekil 7.4. İndis şifreleme süresi

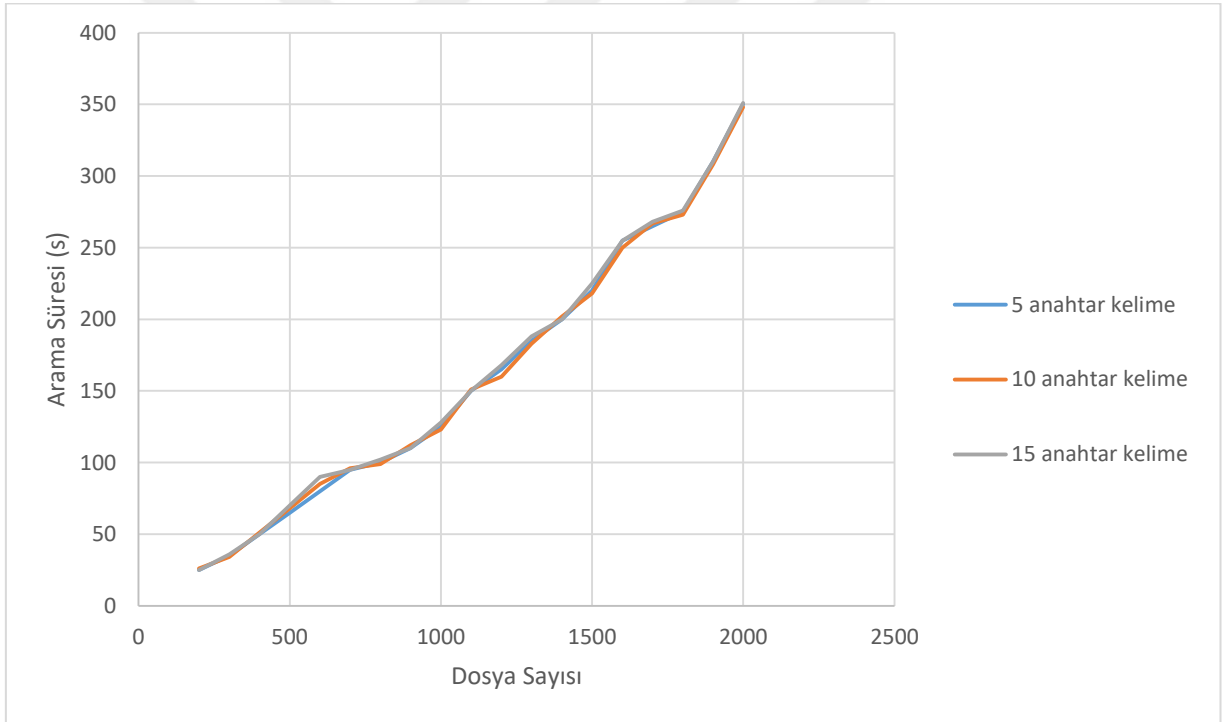
Oluşturulan güvenli indis yapısının şifrlenme zamanı, ayrıca, güvenli indisinin birinci ve ikinci katmanlarının şifrlenme zamanları test edilmiştir. İndis şifreleme yöntemi Jie Wang [50] tarafından sunulan şifreleme yöntemiyle karşılaştırılmıştır. Şifreleme sonuçları aşağıdaki şemalarda verilmiştir.



Şekil 7.5. İndis katmanlarının şifreleme süresinin karşılaştırılması

7.7.2. Farklı şartlarda arama süresi

Şifreli bulut verileri üzerinde arama yapmak için sunulan yöntemin efektif ve verimliliğini belirlemek için farklı sayıda anahtar kelimeler kullanılarak oluşturulan sorgularla aramalar yapılmış ve sonuçları şekil 7.6'da gösterilmiştir. 5, 10 ve 15 anahtar kelime kullanılarak oluşturulan sorgular ile 500, 1000 ve daha fazla dosyadan oluşturulmuş veritabanlarında aramalar yapılmış ve dosyaların ve anahtar kelime sayısına bağlı olarak optimum arama süreleri belirlenmiştir. Arama süresi dosyaların sayısına göre lineer olarak artmaktadır. Ayrıca, aralık sorgu kullanılması sonuçların daha optimum sürede döndürülmesini sağlamaktadır. Bu sonuçlar Jie Wang [50] tarafından geliştirilen yöntemde gösterildiği gibi geliştirilen arama algoritmasının verilerin döndürülmesi işleminde karmaşık aramaları desteklediğini göstermektedir.



Şekil 7.6. Anahtar kelime sayısına bağlı olarak arama süresi

8. SONUÇ

Bu tez çalışması kapsamında, aralık sorgu, çoklu anahtar kelime ve sıralama mantığını destekleyen arama yöntemi sunulmuştur. Bu yöntem kullanıcılar tarafından oluşturulan çoklu anahtar kelime ile arama yöntemini destekleyen sorguların optimum zaman süresinde sonuçları döndürmesini sağlamaktadır.

Önerilen bu arama yöntemi Wang [25] ve Jie Wang [50] tarafından sunulmuş yöntemleri temel alarak geliştirilmiştir. Temel alınan bu güçlü ve verimli yöntemlerin avantajlarının kullanılmasının yanı sıra, güvenlik açısından daha geliştirilmiş bir arama yöntemi geliştirilmesi üzerine çalışılmış ve denemeler yapılmıştır.

Bu tez konusunun günümüzde kitleler arasında geniş yer bulmuş bulut bilişiminin kullanıcıları endişelendiren güvenlik ve gizlilik konularında bir miktar avantaj sağlayacağı umulmaktadır. Ayrıca, kullanıcılara daha kısa süreli arama ve en ilişkili sonuçların döndürülmesini sağlayarak, bulut bilişiminin kurumlar ve büyük şirketlerin yanı sıra, kişiler tarafından da verilerini depolamak için daha fazla tercih edilmesine katkıda bulunacaktır.

Sunulan arama yöntemi kullanıcılara gereken dosyalara ulaşmaları için kolaylık sağlamakla beraber güvenlik ve zaman açısından birçok avantaj sağlamaktadır.

- Yöntemin çoklu anahtar kelime aramasını desteklemesi kullanıcının ihtiyacı duyduğu doğru dosyayı elde etmesini sağlamaktadır.
- Sıralama mantığı kullanıcıya döndürülen çok sayıda dosyayı inceleyerek zaman kaybetmesini engellemektedir. Kullanıcıya sadece ihtiyacı duyduğu sayıda dosya döndürülmesini sağlar.
- Tf-Idf yöntemi kullanılarak dosyalar arasında benzerlik puanının hesaplanması ve DeğerTablosuna yerleştirilmesi, dosyaların ağırlık puanına göre karşılaştırılması kullanıcı sorgularına en uygun sonuçların döndürülmesini sağlayacaktır.
- İki katmanlı Bloom filtresinin kullanılması dosyaların bulut sunucusu ve yetkisiz üçüncü kişiler tarafından ele geçirilmesinin önüne geçilmesi için gereken güvenliğini ve gizliliği gereksinimlerini güçlendirir.
- Bu arama algoritması kullanıcılar tarafından oluşturulan sorgulara uygun sonuçların optimum sürede döndürülmesini sağlar.

- Rastlantısal seçilen güvenlik parametreleri bu yöntemin istatistiksel saldırılara karşı daha güvenli olmasını sağlamaktadır ve tahminle sonuçlara ulaşılmasını engeller.
- Düzen açıklayıcı şifreleme yönteminin kullanılması temel alınmış Jie Wang arama algoritmasında kullanılan şifreleme yönteminden daha sağlam bir güvenlik yöntemi sunmaktadır. İçerdiği özel bilgi sızması fonksiyonu OPE yöntemine göre daha güvenli bir şifreleme sistemi geliştirilmesine imkan verir.
- Üçüncü taraflar veya bulut sunucusu şifreli metinlerin sadece polynomial sayısını görürse, şifreleme yönteminin açıklarını kullanarak şifreli metinlerin altında yatan düz metinlerle ilgili bilgi edinemeyeceği belirlenmiştir.



KAYNAKLAR

- [1] Vaquero, L. M., Rodero-Merino, L., Caceres, J., Lindner, M., A break in the clouds: Towards a cloud definition, *ACM SIGCOMM Computer Communication Review*, 313 – 316, **2009**.
- [2] Kushilevitz, E., Ostrovsky, R., Replication is not needed: Single Database, Computationally Private Information Retrieval, *Foundations of Computer Science*, s.364 – 373, **1997**.
- [3] Catteddu D., Cloud Computing: Benefits, Risks and Recommendations for Information Security, *Communications in Computer and Information Science*, s. 17-17, **2009**.
- [4] Song, D.X., Wagner, D., Perrig, A., Practical Techniques for Searches on Encrypted Data, *Proc. Of S&P*, **2000**.
- [5] Bellovin, S.M., Cheswick, W.R., Privacy-enhanced searches using encrypted Bloom filters, *IACR ePrint Cryptography Archive*, **2004**.
- [6] Oswal E., Fischlin M., "Advances in Cryptology - EUROCRYPT 2015, 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, **2015**.
- [7] Wang, C., Cao, N., Ren, K., Lou, W., Enabling secure and efficient ranked keyword search over outsourced cloud data, *IEEE Transactions on Parallel and Distributed Systems* 23(8), s. 1467–1479, **2012**.
- [8] Cao, N., Wang, C., Li, M., Ren, K., Privacy Preserving Multi-keyword Ranked Search over Encrypted Cloud Data, *INFOCOM, 2011 IEEE*, **2011**.
- [9] Weinhardt, C., Anandasivam A., Blau, B., Borissov, N., Meinel T., Michalk, W., Stöber, J., Cloud Computing – A Classification, Business Models, and Research Directions, *Information Systems Engineering*, **2009**.
- [10] Gajala G., Cloud Computing: A State of Art of the Cloud, *International Journal of Computer Trends and Technology*, **2013**.
- [11] Wong J., Wang X., Li H., Chan G., Li H., A Review of Cloud-Based Bim Technology in the Construction Sector, *Journal of Information Technology in Construction*, **2014**.
- [12] Voorsluys, W., Broberg, J., Buyya, R, Introduction to Cloud Computing", *Cloud Computing: Principles and Paradigms*, s. 1–44, **2011**.
- [13] Kamara, S., Lauter, K., Cryptographic Cloud Storage, *Financial*

Cryptography and Data Security, s. 136-149, **2010**.

- [14] Strachey, C., Time Sharing in Large Fast Computers, the *International Conference on Information processing, UNESCO*, s. 336–341, **2006**.
- [15] Rochwerger, B., Caceres, J., Montero, R.S., Breitgand, D., Elmroth, E., Galis, A., Levy, E., Llorente, I.M., Nagin, K., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W., Galan, F., The RESERVOIR Model and Architecture for Open Federated Cloud Computing, *IBM Journal of Research and Development*, 53/4, **2009**.
- [16] TTNET Bulutu, <http://www.netdisk.com.tr/>, **2011**.
- [17] Rao C.C., Leelarani M., Kumar Y.R., Cloud: Computing Services And Deployment Models, *International Journal Of Engineering And Computer Science*, **2013**.
- [18] Amies, A., Sluiman, H., Tong, Q.G., Liu, G.N., Infrastructure as a Service Cloud Concepts, *Developing and Hosting Applications on the Cloud*, **2012**.
- [19] Miller, R., What's In A Name? Utility vs. Cloud vs Grid, <http://www.datacenterknowledge.com>, **2008**.
- [20] Boneh, D., Crescenzo, D., Ostrovsky, R., Persiano, G., Public key encryption with keyword search, *EUROCRYPTO 2004*, s. 506–522, **2004**.
- [21] Crescenzo, G. D., Saraswat, V., Public key encryption with searchable keywords based on jacobi symbols, *the 8th International Conference on Cryptology*, 4859 Springer, **2007**.
- [22] Goh, E.J., Secure indexes, *Cryptology ePrint Archive*, **2003**.
- [23] Moataz, T., Shikfa, A., Boolean symmetric searchable encryption, *8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, **2013**.
- [24] Orencik, C., Kantarcioglu, M., Savas, E., A practical and secure multi-keyword search method over encrypted cloud data, *the 6th IEE International Conference on Cloud Computing*, **2013**.
- [25] Wang, C., Cao, N., Li, J., Rei, K., Secure Ranked Keyword Search over Encrypted Cloud Data, *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, s. 253 – 262, 2010.
- [26] Yu, S., Wang, C., Ren, K., Lou, W., Achieving secure, scalable, and fine-grained data access control in cloud computing, *IEEE INFOCOM 2010*, **2010**.

- [27] Wang, C., Wang, Q., Ren, K., Lou, W., Privacy-preserving public auditing for data storage security in cloud computing, *INFOCOM*, **2010**.
- [28] Sun, W., Wang, B., Cao, N., Li, M., Lou, W., Hou, T., Li H., Privacy preserving multi-keyword text search in the cloud supporting similarity based ranking, *ASIACCS 2013*, **2013**.
- [29] Waters, B. R., Balfanz, D., Durfee, G., Smetters, D. K., Building an encrypted and searchable audit log, *ISOC NDSS 2004*, **2004**.
- [30] Golle, P., Staddon, J., Waters, B., Secure Conjunctive Keyword Search Over Encrypted Data, *Second International Conference, ACNS 2004*, s.31-45, **2004**.
- [31] Scheuermann, P., Ouksel, M., Multidimensional b-trees for associative searching in database systems. *Information systems* 7(2), s. 123–137, **1982**.
- [32] Boneh, D., Kushilevitz, E., Ostrovsky, R., W. E. S., “Public key encryption that allows pir queries, *CRYPTO*, **2007**.
- [33] Grossman, D., Frieder O., Information retrieval: algorithms and heuristics. Kluwer international series on information retrieval. *Springer*, **2004**.
- [34] Boneh, D., Waters, B., Conjunctive, subset, and range queries on encrypted data. In: *Proceedings of the 4th conference on Theory of cryptography*, s. 535–554. Springer-Verlag (2007)
- [35] Sun, W., Lou, W., Hou, Y.T., Li, H., Privacy Preserving Keyword Search Over Encrypted Data in Cloud Computing, *Secure Cloud Computing*, s. 189-212, **2013**.
- [36] Ostrovsky, R., Efficient computation on oblivious RAMs. *ACM Symposium on Theory of Computing (STOC '90)*, s. 514–523, **1990**.
- [37] Goldreich, O., Ostrovsky, R., Software protection and simulation on oblivious RAMs, *Journal of the ACM*, 43(3), s.431–473, **1996**.
- [38] Bellare, M., Boldyreva, A., O'Neill, A., Deterministic and efficiently searchable encryption, *CRYPTO*, **2007**.
- [39] Kushilevitz, E., Ostrovsky, R., Replication is NOT needed: SINGLE database, computationally-private information retrieval, *IEEE Symposium on Foundations of Computer Science (FOCS '97)*, s. 364–373, **1997**.
- [40] Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M., Private Information Retrieval, *FOCS 95, Journal of ACM*, **1995**.
- [41] Ishai, Y., Kushilevitz, E., Ostrovsky R., Sahai, A., Batch codes and their

- applications, *ACM Symposium on Theory of Computing (STOC '04)*, s. 262–271. ACM, 2004.
- [42] Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A., Cryptography from anonymity, *IEEE Symposium on Foundations of Computer Science (FOCS '06)*, **2006**.
 - [43] Curtmola, R., Garay, J. A., Kamara, S., Ostrovsky, R., Searchable symmetric encryption: improved definitions and efficient constructions, *ACM CCS*, **2006**.
 - [44] Orencik, C., Kantarcioglu, M., Savas, E., “A Practical and Secure Multi-Keyword Search Method over Encrypted Cloud Data”, *2013 IEEE Sixth International Conference on Cloud Computing*, s. 390 – 397, **2013**.
 - [45] Ostrovsky, O., Goldreich, R., Software protection and simulation on oblivious RAMs, *Journal of the ACM (JACM)*, s. 431-473, **1996**.
 - [46] Chang, Y., Mitzenmacher, M., Privacy preserving keyword searches on remote encrypted data, *Applied Cryptography and Network Security (ACNS '05)*, s. 442–455, Springer, **2005**.
 - [47] Chang, Y.C., Mitzenmacher, M., Privacy preserving keyword searches on remote encrypted data, *Proc. of ACNS*, **2005**.
 - [48] Deterministic and Efficiently Searchable Encryption, Preliminary version appears in CRYPTO 2007, s. 535-552, **2006**.
 - [49] Song, D., Wagner, D., Perrig, A., Practical techniques for searching on encrypted data, *IEEE Symposium on Research in Security and Privacy*, s. 44–55. IEEE Computer Society, **2000**.
 - [50] Wang, J., Yu, X., Zhao, M., “Privacy Preserving Ranked Multi Keyword Fuzzy Search on Cloud Encrypted Data Supporting Range Query”, *Arabian Journal for Science and Engineering*, s. 2375-2388, **2015**.
 - [51] Hwang, Y., Lee, P., Public key encryption with conjunctive keyword search and its extension to a multi-user system, *Pairing 2007*, s. 2–22, **2007**.
 - [52] Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., Lou, W., Fuzzy keyword search over encrypted data in cloud computing, *IEEE INFOCOM'10 Mini-Conference*, **2010**.
 - [53] Kamara, S., Lauter, K., Cryptographic cloud storage, *Springer* s.136-149, **2010**.
 - [54] Liu C., Zhu L., Li L., Tan Y., Fuzzy Keyword Search on Encrypted Cloud Storage Data with Small Index, *ICCIS 2011*, s. 269–273, **2011**.

- [55] Chuah, M., Hu, W., Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data, *Distributed Computing Systems Workshops (ICDCSW), 31st International Conference on*, s. 273–281, **2011**.
- [56] Comer, D., Ubiquitous b-tree. *ACM computing surveys* 11(2), s. 121–137, **1979**.
- [57] Goldreich, O., Ostrovsky, R., Software protection and simulation on oblivious RAMs, *Journal of the ACM*, 43(3): s.431–473, **1996**.
- [58] Kuzu, M., Islam, M. S., Kantarcioglu, M., Efficient similarity search over encrypted data, *the 2012 IEEE 28th International Conference on Data Engineering*, s. 1156–1167, **2012**.
- [59] Witten, I. H., Moffat, A., Bell, T. C., Managing gigabytes: compressing and indexing documents and images, 2 edition, **1999**.
- [60] Orencik, C., Savas, E., Efficient and secure ranked multi-keyword search on encrypted cloud data, *the 2012 Joint EDBT/ICDT Workshops. ACM*, s. 186–195, **2012**.
- [61] Lu, Y., Privacy-preserving logarithmic-time search on encrypted data in cloud, *19th Annual Network and Distributed System Security Symposium (NDSS Symposium'12)*, **2012**.
- [62] Burton, B., Space/time trade-offs in hash coding with allowable errors, *Communications of ACM*, s. 13(7):422-426, **1970**.
- [63] Marais, J., Bharat, K., Supporting cooperative and personal surfing with a desktop assistant, *ACM UIST'97*, **1997**.
- [64] Baek, J., Safavi-Naini, R., Susilo, W. Public Key Encryption with Keyword Search Revisited, *Computational Science and Its Applications – ICCSA 2008*, s. 1249-1259, **2008**.
- [65] Barde, C.R., Katkade, P., Shewale, D., Khatale, R., Secured Multiple – Keyword Search over Encrypted Cloud data, *International Journal of Emerging Technology and Advanced Engineering*, s.2250-2459, **2014**.
- [66] Strizhov, M., Ray, I., Multi – keyword Similarity Search Over Encrypted Cloud Data, *ICT Systems Security and Privacy Protection, Information and Communication Technology*, s. 52-65, **2015**.
- [67] Chenette N., Lewi K., Weis S.A., Wu D.J, Practical Order-Revealing Encryption with Limited Leakage, *FSE 2016*, **2016**.

ÖZGEÇMİŞ

Kimlik Bilgileri

Ad, Soyad : Kubra Mammadova
Doğum Yeri : Baku, Azerbaycan
Medeni Hali : Evli
E-posta : kubra.memmedli@gmail.com
Adresi : 25 Mart mahallesi, SSK blokları, 79/11, AOÇ, Yenimahalle

Eğitim

Lisans : Baku Devlet Üniversitesi, Bilgisayar Bilimleri, Baku, 2011
Lise : Nesimi Rzayev Lisesi, Baku, Azerbaycan, 2007

Yabancı Dil ve Düzeyi

İngilizce (çok iyi), Rusça (çok iyi), Azerbeycanca (çok iyi)

İş Deneyimi

2007 -2008 Baku Devlet Üniversitesi, Bilişim Enstitüsü, Hesaplama tecrübesi
Temmuz 2009 Baku Devlet Üniversitesi, Bilgisayar Mühendisliği Bölümü, EHM'de praktikum
Temmuz 2010 Baku Devlet Üniversitesi, Bilişim Enstitüsü, Hesaplama tecrübesi
2013 - 2014 Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, Bilgisayarla Görme Laboratuvarı

Deneyim Alanları

Ağ yönetim sistemleri, Görsel Belirginlik, Pozlandırma Birleştirimi, Bulut Bilişimi

Tezden Üretilmiş Projeler ve Bütçesi

Tezden Üretilmiş Yayınlar

K. Mammadova, M.Aydos, “A Comprehensive Analysis on Privacy Preserving Keyword Search over Encrypted Cloud Data: Recent Trends”, *Turkish Journal of Electrical & Computer Sciences*, 2016 (Gönderildi).

K. Mammadova, M.Aydos, “Efficient Privacy Preserving Multi Keyword Ranked Search Method Over Encrypted Cloud Data”, 2016 (Düzenlenme aşamasında).

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

Hacettepe Üniversitesi Bilgisayar Mühendisliği Lisansüstü Araştırmaları Çalıştayı





HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 3/8/2016

Tez Başlığı / Konusu: Sıfzeli Bulut Verileri Üzerinde Güvenli ve Efektif
Çoklu Anahtar Kelime ile Azama Yöntemi

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve d) Sonuç kısımlarından oluşan toplam 79 sayfalık kısmına ilişkin, 29/07/2016 tarihinde şahsım/tez danışmanım tarafından TURNITIN adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 4 tür.

Uygulanan filtrelemeler:

- 1- Kaynakça hariç
- 2- Alıntılar hariç/dâhil
- 3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orjinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

03.08.2016
Tarih ve İmza

Adı Soyadı: Kubza Mammadova
Öğrenci No: Nu 22 7207
Anabilim Dalı: Bilgisayar Mühendisliği
Programı:
Statüsü: ☒ Y.Lisans ☐ Doktora ☐ Bütünleşik Dr.

DANIŞMAN ONAYI

UYGUNDUR.

Yrd. Doç. Dr. Murat AYDOS
(Unvan, Ad Soyad, İmza)