Dokuz Eylül Üniversitesi-Mühendislik Fakültesi
Fen ve Mühendislik Dergisi
Cilt 20, Sayı 58, Ocak, 2018

Dokuz Eylul University-Faculty of Engineering
Journal of Science and Engineering
Volume 20, Issue 58, January, 2018

DOI: 10.21205/deufmd. 2018205805

Elektronik Sağlık Kayıtlarının Veri Tabanında T-SQL ile Şifrelenmesi ve Başarım Deneyleri

Gökhan DALKILIÇ*[1], Enis KARAARSLAN[2]

[1]Dokuz Eylül Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 35390, İzmir (ORCID: 0000-0002-0130-1716)
[2]Muğla Sıtkı Koçman Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 48000, Muğla (ORCID: 0000-0002-3595-8783)

(Alınış / Received: 14.01.2017, Kabul / Accepted: 19.10.2017,
Online Yayınlanma / Published Online: 20.01.2018)

Anahtar Kelimeler
Elektronik sağlık kayıtları,
Veri tabanı şifreleme,
T-SQL

Özet: Sağlık kayıtlarının bilgisayar ortamında tutulması ve güncel hasta verilerine ulaşılabilmesiyle tıbbi süreçlerin etkinliği artmaktadır. Bu aynı zamanda güvenlik ve mahremiyet sorunlarını da beraberinde getirmektedir. Bu süreçlerde verilerin şifrelenmesi ve şifrelemenin sistemin çalışmasını aksatmadan gerçekleştirilmesi büyük önem taşımaktadır. Bu çalışma kapsamında, CryptDB ve işlemsel yapılandırılmış sorgu dili (T-SQL) tabanlı şifreleme sistemleri kıyaslanmıştır. T-SQL tabanlı şifrelemenin avantajları belirtilmiş, güncel veri seti olarak gözetim, epidemiyoloji ve sonuçlar programı (SEER) kanser verisi değişik boyutlara bölünmüş, şifreli ve şifresiz olarak veri tabanına yüklenmiştir. Şifreli ve şifresiz veri üzerinde sorgular çalıştırılmış, bu işlemler sonucu elde edilen başarım değerleri verilmiştir.

Encrypting Electronic Health Records with T-SQL in Database and Performance Tests

Keywords
Electronic health records,
Database encryption,
T-SQL

Abstract: The effectiveness of the medical processes is increasing as health records are kept in a computerized environment and the current patient data are being accessible. This has also caused security and privacy issues. Encryption of the data and implementing the encryption process without disrupting the operation of the system is of paramount importance. In this study, CryptDB and Transact sequential query language (T-SQL) based encryption systems are compared. The advantages of the T-SQL based encryption is specified, and surveillance, epidemiology, and end results program (SEER) cancer data is used as an actual dataset which is then divided into different sizes and stored in the database as encrypted and unencrypted. After running queries on the encrypted and unencrypted data, the performance values that are results of these operations are given.

*Gökhan Dalkılıç: dalkilic@cs.deu.edu.tr

1. Giriş

Hastane bilgi sistemlerinin etkinliğinin artırılması süreçlerinde elektronik sağlık kayıtlarının bilgisayar ortamında tutulması ile bu verilerin sınıflandırılması ve anlamlandırılması gerçekleşmiştir. Güncel hasta verilerinin farklı kurumlar tarafından ulaşılabilmesi ile zamandan ve maliyetten tasarruf edilebilmektedir.

Hasta verileri işlenmekte ve farklı kurumlar bu verilere erişim sağlamaktadır. Elektronik sağlık kayıtları kurumlar arasında taşınırken şifrelenmekte ama kurumların iç süreçlerinde ve veri tabanında saklanmasında şifreleme teknolojileri genellikle uygulanmamaktadır. Kurumlar, sistemlerinin yavaşlamasından veya bir ürüne bağımlı olmaktan çekinmektedir [1].

Elektronik sağlık kayıtlarında tutulan hasta bilgilerinin bir kısmı, hassas veri dediğimiz kişisel verilerdir. Bu veriler birçok ülkede, kanunlara ve uluslararası hukuka göre korunması gereken değerlerdir. Bu verilerin güvenliğinin ve mahremiyetinin sağlanması ve elektronik sağlık kayıtlarına kimin, hangi şartlarda ulaşabileceğinin denetiminin yapılması gerekmektedir. Bu amaçla, erişimin yetkilendirilmiş kişiler tarafından sağlanması ve kayıtlar veri tabanında depolanırken şifreleme teknolojisinin kullanımı ihtiyacı doğmaktadır. Şifreleme sürecinde yapılacak ek hesaplamalar sistemin işlemci gücünü kullanacağından, bir miktar yavaşlama söz konusu olabilecektir [1].

Tsai ve arkadaşları [2], bulut üzerinde şifreli olarak duran elektronik sağlık kayıtlarına akıllı kart tabanlı eliptik eğri şifreleme sistemi ile erişilebilmesini sağlayan bir sistem önermişlerdir. Ancak, önerilen sistemin performans testi sonuçları verilmemiştir. Her ne kadar eliptik eğri şifreleme sistemi standart asimetrik şifreleme sistemlerine göre daha hızlı çalışsa da simetrik şifreleme sistemlerinden daha yavaştır. Bunun nedeni matematiksel işlemlerin daha fazla olmasından dolayı daha uzun sürede çalışmasındandır.

Fernandez-Aleman ve arkadaşlarının yazdıkları inceleme makalesinde [3], inceledikleri çalışmalarda simetrik ve asimetrik şifreleme tekniklerinin eşit miktarda kullanıldığını ancak simetrik şifrelemenin daha hızlı olduğu ve büyük veriler için simetrik şifrelemenin daha verimli olduğu belirtilmiştir. Ayrıca, asimetrik şifrelemenin ise bazı gizlilik zafiyetleri içerdiğine değinilmiştir. Aynı çalışmada, hasta kayıtlarının şifrelenmesi için geliştirilecek bir sistemin yeni kayıtların eklenmesini kolaylıkla desteklemesi gerektiği sonucuna varılmıştır.

Mohammed ve arkadaşları yaptıkları çalışmada [4], veri madenciliği için elektronik sağlık kayıtlarını içeren veri tabanlarının güvenli bir şekilde yönetimini sağlamak amacıyla bir yapı (framework) önermişlerdir. Bu yapı, hasta kayıtlarına ulaşmak isteyen istemcinin önüne güvenli bir vekil yerleştirip, bu vekil aracılığıyla şifreli sorguların veri tabanına iletilmesi ve sonuçların şifreli olarak alınmasını içermektedir. Ancak, benzer CryptDB tabanlı diğer bir çalışmada [5] olduğu gibi sadece birtakım SQL sorgusu için destek sunulmuştur.

Bu çalışmada; ikinci bölümde şifreleme konusundaki temel kavramlar ve veri tabanlarının şifrelenmesinde kullanılan farklı şifreleme yöntemleri tanıtılmış ve bu yöntemler karşılaştırılmıştır. Üçüncü bölümde, uygulamanın yapıldığı ortam, veri seti ve yöntem ele alınmış, yapılan deney sonuçları tablolar halinde verilerek yorumlanmıştır. Son bölümde, sonuçlar ve gelecek çalışmalar sunulmuştur.

2. Temel Kavramlar

2.1. Şifreleme Yöntemi ve Algoritmanın Seçilmesi

Şifreleme süreçlerinde, çeşitli şifreleme algoritmalarından amaca en uygun şifreleme algoritmasının seçilmesi önem arz etmektedir. Algoritma seçiminde aşağıdaki kriterlerden söz edilebilir:

- Güçlü şifreleme: Şifreleme algoritmasına yapılacak saldırılara karşı algoritmanın dayanıklılığı güçlü şifrelemeyi tanımlar. Bazı algoritmaların ne kadar güçlü oldukları matematiksel yöntemler veya saldırı deneyleri ile ortaya konulmuştur [6, 7].

- Anahtar uzunluğu: Anahtar uzunluğu şifrelemenin başarımını etkilemektedir. Bazı algoritmalarda, farklı anahtar uzunluğunun seçilebilmesi mümkündür ve daha uzun anahtarlar daha güçlü şifreleme anlamına gelmektedir. Uzun anahtar seçilmesinin dezavantajı ise anahtarın üretiminde ve dağıtımındaki zorluktur.

- İşlemci gücü: Şifreleme ve deşifrelemede merkezi işlem biriminin ne oranda kullanıldığını tanımlar. Güçlü şifreleme algoritmaları ve uzun anahtarlar, daha fazla işlemci kaynağı harcamaktadır [8].

Simetrik şifreleme algoritmaları; asimetrik şifreleme algoritmalarına göre daha performanslı çalışacaklarından, özellikle yüksek miktarda veri şifreleneceği zaman tercih edilmesi önerilmektedir. Anahtar dağıtımı gibi düşük miktardaki verilerde asimetrik şifreleme süreçleri etkin olarak kullanılmaktadır. Verinin daha az yer kaplaması için sıkıştırma algoritmaları kullanılabilir. Burada dikkat edilmesi gereken unsur, şifrelenmiş veri sıkıştırılamayacağı için verinin şifrelenmeden önce sıkıştırılmasıdır [8]. Ayrıca, veri tabanındaki bütün alanlar yerine bir kısmının şifrelenmesi (kısmi şifreleme) ya da verilerin anonimleştirilmesi de şifreleme sürecinin performansını olumlu yönde etkilemektedir.

2.2. CryptDB

CryptDB [5] Massachusetts Institute of Technology (MIT)'de geliştirilmekte olan, uygulama ve veri tabanı arasında vekil (proxy) olarak çalışan ve veri tabanını şifreleyerek güvenliğini sağlayan bir sistemdir. Vekil tabanlı çalışan veri tabanı şifreleme sistemlerinin çoğu, CryptDB'nin tasarımına dayanmaktadır. Bu sistemler, rastgele olmayan şifreleme (deterministic encryption DTE) ve sıra koruyarak şifreleme (order preserving encryption OPE) gibi örnekleri olan özellik koruyarak şifreleme (property preserving encryption PPE) şemalarını kullanmaktadır [9].

CryptDB ve benzeri geliştirilmekte olan sistemlerin en büyük sıkıntıları; dokümantasyon eksikliği ve her yazılım geliştirme ortamı için destek ve kütüphanelerinin olmamasıdır. CryptDB'nin kısıtlamaları Patel ve Jiang'ın çalışmasında [10] incelenmiştir. CryptDB gibi sistemlerin en büyük dezavantajı, SQL'in sadece bir kısmını destekleyebilmeleridir. Örneğin "in", "not", "update" işlemleri desteklenmekte ama "like" cümlesi desteklenmemektedir [11].

Güncel bir tez çalışmasında [12] yapılan testlerde, veri tabanındaki veriler arttıkça sistemin çok ciddi şekilde yavaşladığı; CryptDB'nin var olan sürümünün muhtemelen bellek yönetiminde ve iş parçacığı işlemede (thread handling) güvenilir olmadığı sonucuna varılmıştır. CryptDB; güvenilir bir bulut tabanlı uygulama sunucusu ve vekili ile çalıştığını varsaydığından, daha zayıf bir saldırgan modeli sunmaktadır [13]. CryptDB'nin mimarisinden dolayı yapılabilecek saldırılar Naveed ve

arkadaşlarının çalışması [9] ve Akın ve Sunar'ın çalışmasında [14] incelenmiştir. Elektronik sağlık verilerinin veri tabanına CryptDB ile şifrelenmiş olarak yüklendiği Naveed ve arkadaşlarının çalışmasında [9], sorguların çalışması için şifreleme katmanının yeterince soyulması (peel) durumunda çok miktarda hassas verinin ele geçirilebileceği gösterilmiştir. Belirtilen çalışmada, frekans analizi ve sıralaması (frequency analysis and sorting), kombinatoryal optimizasyon gibi saldırılar kullanılmıştır. Saldırılarda sadece şifrelenmiş sütunlar ve genel kullanıma açık dış verilerden yararlanılmıştır. Hasta kayıtlarının OPE ile şifrelenmiş belirli niteliklerinin %80'inden, DTE ile şifrelenmiş belirli niteliklerinin %60'ından fazlasının ele geçirilebileceği gösterilmiştir.

2.3. Transact SQL (T-SQL)

İşlemsel yapılandırılmış sorgu dili (Transact sequential query language T-SQL) şifreleme özelliği Microsoft SQL Server 2005'den itibaren gelen bir eklentidir. SQL Server, Windows CryptoAPI (CAPI)'yi kullanan bütünleşik bir şifreleme güvenlik modeli sunmaktadır. Anahtar yönetimi ise ANSI X9.17'deki metoda benzer katmanlı bir yapıdadır [15].

Microsoft CAPI [16], farklı Microsoft Windows işletim sistemleri bünyesinde bulunan bir uygulama programlama ara yüzüdür. Gelişmiş şifreleme standardı (Advanced encryption standard AES), üç kat veri şifreleme standardı (Triple data encryption standard DES), Rivest şifreleme 5 (Rivest cipher 5 RC5) ve güvenli özet çıkarma algoritması (Secure hashing algorithm 1 SHA-1) gibi çok sayıda şifreleme ve özet çıkarma algoritmasını destekler. MsSQL Server, Microsoft CAPI'yi kullanan şifreleme, şifre çözme, sayısal imza ve doğrulama üzerine birçok fonksiyon ve özelliğe sahiptir. Bu fonksiyonlar ise simetrik ve asimetrik şifreleme ve şifre çözme, sayısal imza ve imza doğrulama, otomatik anahtar taşıma ile simetrik şifreleme, özet (hash) ile şifreleme ve sertifika kopyalama fonksiyonlarıdır.

Microsoft SQL Server'da veri tabanı seviyesinde şifreleme iki farklı şekilde yapılabilir:

- Şeffaf veri şifreleme (transparent data encryption), veri tabanı motoru tarafından otomatik olarak gerçekleştirilen ve veri tabanının bütününe uygulanan bir şifreleme yöntemidir. Veri tabanı şifrelemede kullanılan anahtar sertifika tarafından korunur ve bu sertifika olmadan veri tabanı dosyası başka bir SQL sunucusunda çalışmaz. Veri tabanı şifreleme işlemi sayfa seviyesinde yapılır, diskte bulunan veri tabanı dosyaları şifreli olarak saklanır. Veriler ara bellekte şifrelenmemiş olarak kalır. Veri tabanı ile ilişkilendirilmiş uygulamanın modifiye edilmesine ihtiyaç duyulmaz [8].

- Sütun seviyesinde şifrelemede (column-level encryption) şifreleme işlemi kolon bazlı olarak sadece istenen kolonun şifrelenmesi şeklinde gerçekleştirilir. Veriler şifrelenmiş olarak belleğe yüklenir ve şifreleme/deşifreleme fonksiyonları (EncryptByKey, DecryptByKey) çalıştırılmadan işlenmez. Performans şifrelenecek sütun ve satır adedine göre değişir ve bu şifreleme yöntemi indekslemeyi desteklemez. Karmaşık bir işlem olmasına karşın şeffaf veri şifreleme ve sütun seviyesinde şifreleme birlikte kullanılabilir [8].

2.4. CryptDB ve T-SQL'in Karşılaştırılması

CryptDB ve T-SQL'in temel özelliklerinin karşılaştırılması Tablo 1'de verilmiştir. Tabloda da belirtildiği üzere CryptDB sadece tablonun tümü üzerinden şifrelemeye imkân tanırken, T-SQL ile

tablonun tümü ya da sütun bazlı bir kısmı şifrelenebilir. Ayrıca, T-SQL'in en büyük avantaj sağladığı kısım tüm SQL sorgularının çalıştırılabilmesidir, ancak CryptDB kullanımında SQL sorgularının bir kısmı desteklenmektedir. T-SQL, CryptDB'de olduğu gibi sadece simetrik şifrelemeyi değil, hem simetrik hem de asimetrik şifrelemeyi farklı algoritma destekleri ile sağlamaktadır. T-SQL'in belirtilen avantajlarından dolayı sağlık kayıtlarının şifrelenmesi amacıyla çalışmamızda T-SQL kullanılmıştır.

Tablo 1. CryptDB ve T-SQL karşılaştırılması

	CryptDB	T-SQL
Anahtar yönetimi	Sistemdeki kullanıcı parolalarını anahtarlara bağlama	ANSI X9.17'deki metoda benzer katmanlı bir anahtar dağıtım yönetimi
Şifrelenen veri	Bütün veri tabanı	Tüm veya kısmi (sütun bazında) veriler
Şifreleme yöntemleri	Simetrik şifreleme tabanlı katmanlı şifreleme (onions of encryption) ve bazı süreçlerde homomorfik şifreleme	Simetrik ve asimetrik şifreleme, özet (hash) fonksiyonları
Şifreleme algoritmaları	Her katman (onion) için farklı; 128-bit AES, Blowfish, Paillier kripto sistemi, Song şifreli veri arama algoritması	Desteklenen simetrik şifreleme algoritmaları: DES, Triple DES, Triple_DES_3KEY, RC2, RC4, 128-bit RC4, DESX, 128-bit AES, 192-bit AES, 256-bit AES
SQL desteği	SQL'in sadece bir kısmı	Tam destek
Çalışma ortamı	Linux (Ubuntu'nun belirli bir sürümü üzerine sorunsuz kurulmaktadır)	Microsoft SQL Server 2012 ve sonrası; bulut ortamında Azure SQL veri tabanı ve SQL Data Warehouse
Çalışma ortamı (Sunucu, bulut altyapısı)	Bulut ortamı hedeflenmiş bir veri tabanı yönetim sistemidir, tek bir sunucu üzerinde de çalışabilir [13]	Bulut ortamında çalışabileceği gibi diğer platformlarda da çalışabilir
Uygulama ve veri tabanında değişiklik	Uygulamada ve veri tabanında ufak değişiklikler sonrası şeffaf olarak çalışacağı belirtilmekle birlikte sorunlar yaşanabilmektedir [12]	Uygulamada T-SQL'e ait fonksiyonlar kullanılır

3. Uygulama

Şifreli ve şifresiz veri üzerinde yapılan çözümlemeler, yerel bir bilgisayar üzerinde gerçekleşmiştir. Yerel bilgisayar üzerinde Windows 10 Professional işletim sistemi çalışmakta olup Intel Core 2 Duo işlemci ve 4 GB bellek bulunmaktadır. Microsoft SQL Server Express ve Standart sürümü şeffaf veri şifrelemeyi desteklemediğinden, Microsoft SQL Server Enterprise (64-bit) 12.0.4213.0 sürümü kullanılmıştır.

3.1. Veri Seti

Bu çalışmada, verilerin şifreli saklanmasının ve işlenmesinin sorgulama performansı üzerindeki etkilerini gözlemlemek amacıyla test verisi olarak gözetim, epidemiyoloji ve sonuçlar programı (surveillance, epidemiology, and end results program SEER) kanser verileri kullanılmıştır [17]. 1973 ile 2012 yılları arasında gerçekleşen toplam 4.524.099 kanser vakasını içeren anonim bir veri setidir. İçeriğinde hastaya ait numara, etnik köken, doğum tarihi ve yeri, teşhisin konulduğu ay, yıl ve bu esnadaki

hastanın yaşı ve medeni durumu, hastalığın seyri ile ilgili tümörün büyüklüğü ve yeri, tekrar etme sayısı, ameliyat durumu, hastanın hayatta olup olmadığı gibi bilgilerden oluşan toplam 150 alan bulunmaktadır. Kanser türlerine göre ayrılmış 9 farklı dosyada metin formatında paylaşılan bu veri, çalışmamız sürecinde geliştirilen bir uygulama aracılığıyla MSSQL üzerinde hazırlanan veri tabanına aktarılmıştır.

3.2. Yöntem

CryptDB ve T-SQL yöntemlerinin kıyaslanması sonucunda, aşağıdaki nedenlerden dolayı T-SQL yönteminin kullanılmasına karar verilmiştir:

- CryptDB'nin çözülmesi gereken bazı sorunları olduğu ve bir ürün olarak kullanılmaya hazır olmaması [12],
- T-SQL'in kısmi (sütun bazında) şifreleme yapabilmesi,
- T-SQL'in farklı şifreleme algoritmalarının seçimini mümkün kılması,
- T-SQL'in SQL'i tam olarak desteklemesidir.

Bu çalışmada, sütun seviyesinde şifrelemede simetrik şifreleme kullanılmış ve T-SQL şifreleme fonksiyonlarından ENCRYPTBYKEY yöntemi tercih edilmiştir. Ayrıca, şeffaf veri şifreleme yöntemi de test edilmiştir. Bir veri tabanı sütununun ENCRYPTBYKEY fonksiyonu ile AES algoritması kullanılarak şifrelenmesine dair bir örnek aşağıda verilmiştir. Örnekte, personel tablosunun TCKimlikNo kolonunda bulunan veriler şifrelenerek yeni oluşturulan SifreliTCKimlikNo sütununda saklanmıştır.

```
USE FACTORY2016;
GO

-- Şifreli veriyi saklamak için yeni bir kolon ekleniyor.
ALTER TABLE Personel
   ADD SifreliTCKimlikNo varbinary(128);
GO

-- Veriyi şifrelemede kullanılacak simetrik anahtar açılıyor.
OPEN SYMMETRIC KEY TCNo_Key_01
   DECRYPTION BY CERTIFICATE Personel01;

-- TCKimlikNo kolonundaki veri TCNo_Key_01 anahtarı ile şifreleniyor ve sonuçlar SifreliTCKimlikNo kolonunda saklanıyor.
UPDATE Personel
SET SifreliTCKimlikNo
      = EncryptByKey (Key_GUID('TCNo_Key_01'), TCKimlikNo);
```

Yukarıdaki sorguda kullanılan TCNo_Key_01 isimli simetrik anahtarın oluşturulması için `CREATE SYMMETRIC KEY` fonksiyonu şu şekilde kullanılmıştır:

```
CREATE SYMMETRIC KEY key_name
 [AUTHORIZATION owner_name]
 [FROM PROVIDER provider_name]
 WITH <key_options> [,...n]|
  ENCRYPTION BY <encrypting_mechanism> [,...n]

<key_options> ::=
 KEY_SOURCE='pass_phrase'|
 ALGORITHM=<algorithm>   |
 IDENTITY_VALUE = 'identity_phrase'  |
 PROVIDER_KEY_NAME = 'key_name_in_provider' |
 CREATION_DISPOSITION = {CREATE_NEW|OPEN_EXISTING}

<algorithm> ::=
```

```
 DES|TRIPLE_DES|TRIPLE_DES_
3KEY|RC2|RC4|RC4_128|DESX|
AES_128|AES_192|AES_256

<encrypting_mechanism> ::=
 CERTIFICATE certificate_name
| PASSWORD='password' |
 SYMMETRIC KEY symmetric_
key_name |
 ASYMMETRIC KEY asym_key_name
```

Kullanılan AES algoritması 16 baytlık veriyi girdi olarak kabul edip, 16 baytlık şifreli veri üretmektedir. Ancak, AES bir blok şifreleme metodu olduğu için 16 bayttan az veri girdi olarak verildiğinde veriyi 16 bayta uzatır (padding). Şifrelenmiş veride bulunan ek alanların nelerden oluştuğu Natarajan ve arkadaşları [18] tarafından verilmiştir. Örneğin; "1" değerinin veri tabanında tutulan şifrelenmiş hali şu şekildedir:

```
0027BFE329DD0E479A728B09A0408
0E201000000A70C10882D17F71CDA
D7A76D823F318E67FABB36DAEC657
2C51EEF18CF1DC607
```

Örnek şifrelenmiş veride bulunan alanlar ve büyüklükleri Tablo 2'de verilmiştir. Tablo 2'den de görüldüğü üzere en iyi ihtimalle 16 bayt uzunluğunda girilmiş olan düz metin, 52 bayt uzunluğunda şifreli metine dönüşmektedir. Bu durumda veri %225 oranında artmaktadır. En kötü durum senaryosunda ise, yani girilen içeriğin 1 bayt olması durumunda ise %5100 şeklinde çok yüksek bir oranda artmaktadır.

Farklı veri büyüklüklerinin, şifreli veri üzerinden sorgulama yapmayı nasıl etkilediğini anlamak için veri seti bölünerek 50.000 - 500.000 arası 50şer bin artırılarak 10 farklı veri seti oluşturulmuştur. Bu veri setleri, şifreli ve şifresiz olarak bilgisayarda kurulan SQL veri tabanında saklanmıştır. Şifreleme veri tabanı katmanında gerçekleştirilmiş, şeffaf veri şifreleme ve sütun seviyesinde şifreleme yöntemleri uygulanmıştır.

Tablo 2. AES ile şifrelenmiş veride bulunan alanlar [18]

Açıklama	Büyüklük (bayt)	Örnek İçerik
Simetrik şifrenin evrensel tekil belirteci (Globally unique identifier - GUID)	16	0027BFE329DD0E479A728B09A04080E2
Sürüm numarası	4	01000000
Rastgele üretilen ilk vektör (initial vector IV)	16	A70C10882D17F71CDAD7A76D823F318E
AES ile şifrelenmiş veri	16	67FABB36DAEC6572C51EEF18CF1DC607
Toplam	52	

Şifreli ve şifresiz veri tablolarının oluşturulma süreleri ve boyutları Tablo 3'de belirtilmiştir. Tablo 3'den de görüldüğü üzere, verinin sütun seviyesinde şifreleme yöntemi ile şifreli saklanmasıyla tüm kayıt sayılarında tablo boyutu yaklaşık 11 kat artmıştır. Şeffaf şifreleme yönteminde tablo boyutunda herhangi bir değişiklik olmadığı için Tablo 3'de ayrı bir kolon olarak belirtilmemiştir. Tablo 3'de görülen şifreli ve şifresiz tablo boyutları arasındaki farkın nedeni, Tablo 2'de gösterildiği gibi 16 baytlık bir verinin sistem tarafından 52 bayta arttırılması ve ayrıca 16 bayttan küçük her türlü girdinin 16 bayta tamamlanarak şifrelenmesidir. Şifresiz veriler için tablo oluşturma süreleri ile şifreli verilerin tablolarının oluşturulma süreleri karşılaştırıldığında, veri boyutunun büyümesinden dolayı süreler arasında

büyük farklar oluştuğu Tablo 3'de görülmektedir. Ancak, bu süreler tabloların ilk defa oluşturulma süreleri olduğundan ve veri tabanı oluşturulurken bu işlemin 1 defa yapılacağı düşünüldüğünde sürelerin gözardı edilebilir süreler olduğu görülmektedir.

Tablo 3. Farklı büyüklükteki şifreli ve şifresiz veri tablolarının oluşturulma süreleri (sn) ve boyutları (MB)

Kayıt Sayısı	Şifresiz tablo kayıt süresi (dakika saniye milisaniye)	Şifreli tablo kayıt süresi	Şifresiz tablo boyutu (MB)	Şifreli tablo boyutu (MB)
50.000	9s 369ms	2d 34s	35,602	429,695
100.000	16s 600ms	5d 14s	71,148	859,383
150.000	25s 693ms	8d 36s	106,695	1.289,070
200.000	45s 879ms	10d 25s	142,242	1.718,758
250.000	1d 2s 919ms	12d 47s	177,789	2.148,445
300.000	1d 17s 698ms	15d 34s	213,336	2.578,133
350.000	1d 29s 859ms	17d 47s	248,883	3.007,820
400.000	1d 38s 690ms	20d 23s	284,430	3.437,508
450.000	1d 50s 898ms	22d 56s	319,977	3.867,195
500.000	2d 7s 819ms	25d 06s	355,523	4.296,891

50.000 - 500.000 arası tüm kayıtlar üzerinde çalıştırılan sorguların şifresiz ve şifreli veri tabanları üzerinde çalıştırılacak sürümleri Tablo 4'de detaylı bir şekilde verilmiştir. Verilen ilk 3 sorgu, "where" anahtar kelimesi ile çalışan sorgulardır. Sorgulardan ilki, sabit bir tanımlayıcıya sahip hastayı arama işlemi, ikincisi doğum yeri Türkiye olan hastaların kayıtlarını sayma işlemi, üçüncüsü ise yaşı 20 ile 30 arasında olan hastaların kayıtlarını sayma işlemidir. 4. sorgu ise bir sıralama operatörü sorgusudur ve hastaları doğum tarihlerine göre azalan sırayla sıralayarak, hasta tanımlayıcılarını ve buna karşılık gelen doğum tarihlerini listelemektedir. 5. sorgu bir gruplama operatörü sorgusudur, bu sorgu kanser tümörünün görüldüğü alana göre gruplama yapar. 6. sorgu küme operatörü örneğidir ve görünen farklı boyuttaki tümör büyüklükleri listelemek amacıyla kullanılmıştır. 7. ve 8. sorgular biriktirme (aggregate) operatörü örnekleridir. 7. sorgu teşhisin konulduğu anda hastaların ortalama yaşını hesaplamaktadır. 8. sorgu teşhisin konulmasından sonra hastanın maksimum kaç ay daha hayatta kaldığının sonucunu döndürmektedir.

Tablo 4'de verilen her bir sorgu 50.000-500.000 arası tüm kayıtlar için çalıştırılmış ve her bir sorgunun şifresiz, şeffaf şifreli ve şifreli veriler üzerinde çalışma süreleri Tablo 5'de verilmiştir. İşletim sisteminden kaynaklanan işlemlerden dolayı sürelerin daha sağlıklı hesaplanabilmesi için her bir sorgu 100 defa çalıştırılmış ve ortalama değer hesaplanmıştır. Şifresiz ve şeffaf şifreli tablolar üzerinde uygulanan sorguların çoğunun çalışma süresi milisaniye seviyesinde iken, şifreli tablolarda süreler saniyeler hatta dakikalar seviyesine çıkmaktadır. Bunun ilk nedeni, şifreli tabloların boyutlarının Tablo 3'de verildiği gibi şifresiz tablo boyutlarından büyük olmasıdır. İkinci nedeni ise şifreli tablolarda yapılan sorgu işlemlerinde tablodaki verinin şifresinin çözülmesinin ardından sorgunun çalışmasıdır. Şeffaf şifreli ve şifresiz tabloların yer aldığı veri tabanı dosyaları aynı uzunlukta olduğundan her iki yöntemde de uygulanan sorguların süreleri hemen hemen aynıdır.

Tablo 4. Performans testlerinde kullanılan örnek sorgular

Operatör		No	Şifresiz Sorgu	Şifreli Sorgu
Kısıtlama Operatörü	Where > < =	1	SELECT * FROM INCIDENCE WHERE Patient_ID_number = '07000155'	SELECT CONVERT(NCHAR(8),DECRYPTBYKEY(Patient_ID_number)), CONVERT(varchar(3), DECRYPTBYKEY(Age_at_diagnosis)) FROM INCIDENCE WHERE CONVERT(NCHAR(8),DECRYPTBYKEY(Patient_ID_number)) = '07000155'
	Where LIKE	2	SELECT COUNT(*) FROM INCIDENCE WHERE Birthplace_country LIKE 'TUR'	SELECT COUNT(*) as Count FROM INCIDENCE WHERE CONVERT(NCHAR(3),DECRYPTBYKEY(Birthplace_country)) LIKE 'TUR'
	Where BETWEEN	3	SELECT COUNT(*) FROM INCIDENCE WHERE Age_at_diagnosis BETWEEN 20 AND 30	SELECT COUNT(*) as Count FROM INCIDENCE WHERE CONVERT(varchar(3),DECRYPTBYKEY(Age_at_diagnosis)) BETWEEN 20 AND 30
Sıralama Operatörü	OrderBy	4	SELECT Patient_ID_number, Year_of_Birth FROM INCIDENCE ORDER BY Year_of_Birth desc	SELECT CONVERT(NCHAR(8),DECRYPTBYKEY(Patient_ID_number)), CONVERT(varchar(4),DECRYPTBYKEY(Year_of_Birth)) AS Year_of_Birth FROM INCIDENCE ORDER BY Year_of_Birth desc
Gruplama Operatörü	GroupBy	5	SELECT Primary_Site, COUNT(*) FROM INCIDENCE GROUP BY Primary_Site	SELECT CONVERT(NCHAR(4),DECRYPTBYKEY(Primary_Site)) AS Primary_Site, COUNT(*) as Count FROM INCIDENCE GROUP BY CONVERT(NCHAR(4),DECRYPTBYKEY(Primary_Site))
Küme Operatörleri	Distinct	6	SELECT DISTINCT EOD_Tumor_Size FROM INCIDENCE	SELECT DISTINCT CONVERT(NCHAR(3),DECRYPTBYKEY(EOD_Tumor_Size)) FROM INCIDENCE
Biriktirme (Aggregate) Operatörleri	Sum, Average	7	SELECT AVG(Age_at_diagnosis) FROM INCIDENCE WHERE Age_at_diagnosis != '999'	SELECT AVG(CONVERT(int, CONVERT(varchar(3),DECRYPTBYKEY(Age_at_diagnosis)))) FROM INCIDENCE WHERE CONVERT(varchar(3),DECRYPTBYKEY(Age_at_diagnosis)) != '999'
	Min, Max	8	SELECT MAX(Survival_months) FROM INCIDENCE WHERE Survival_months <> '9999'	SELECT MAX(CONVERT(int, CONVERT(NCHAR(4),DECRYPTBYKEY(Survival_months)))) FROM INCIDENCE WHERE CONVERT(NCHAR(4),DECRYPTBYKEY(Survival_months)) <> '9999'

Tablo 5. Şifreli, şeffaf şifreli ve şifresiz farklı boyuttaki tablolar üzerinde örnek sorguların çalışma süreleri

	Tablolar	SORGULAR (ms)							
		1	2	3	4	5	6	7	8
Şifresiz Tablolar	50.000	18	24	14	317	31	26	19	25
	100.000	27	32	15	452	31	29	19	26
	150.000	33	37	18	640	47	41	30	40
	200.000	42	56	26	972	62	55	40	51
	250.000	53	64	34	196	82	73	49	66
	300.000	64	69	38	1s 475	93	90	57	78
	350.000	74	81	44	1s 746	108	96	67	90
	400.000	84	94	55	2s 16	122	109	78	102
	450.000	95	106	61	2s 263	139	122	93	118
	500.000	110	117	62	2s 543	152	137	103	128
Şeffaf Şifreli Tablolar	50.000	21	26	14	359	35	31	22	28
	100.000	32	34	19	484	49	33	22	30
	150.000	45	40	23	771	59	67	44	44
	200.000	71	66	37	1s 110	67	63	42	61
	250.000	73	67	37	1s 358	97	86	61	69
	300.000	89	76	40	1s 635	101	94	64	83
	350.000	132	92	47	1s 952	115	109	72	96
	400.000	119	104	57	2s 216	129	121	83	114
	450.000	132	118	61	2s 513	147	131	90	129
	500.000	150	129	65	2s 771	164	151	101	134
Şifreli Tablolar	50.000	849	714	1s 143	1s 601	770	724	1s 671	1s 192
	100.000	3s 52	1s 429	2s 292	3s 337	1s 544	1s 450	3s 335	2s 411
	150.000	3s 898	2s 185	3s 595	8s 907	2s 260	2s 217	5s 257	3s 591
	200.000	4s 129	2s 875	4s 742	6s 832	2s 951	2s 897	6s 908	4s 746
	250.000	18s 593	3s 670	6s 103	24s 512	3s 769	3s 701	8s 746	6s 30
	300.000	1d 22s 393	30s 470	30s 417	1d 23s 84	16s 260	15s 705	16s 832	16s 601
	350.000	1d 24s 931	37s 946	37s 612	1d 40s 669	26s 921	26s 37	29s 904	24s 620
	400.000	1d 36s 588	40s 156	40s 182	1d 46s 732	41s 890	41s 706	41s 427	41s 470
	450.000	1d 55s 205	47s 495	47s 168	1d 56s 104	46s 497	46s 945	47s 628	47s 308
	500.000	2d 8s 421	58s 356	1d 8s 27	2d 10s 353	51s 368	51s 625	51s 335	51s 620

Tablo 5'de verilen 4. sorgu, tablonun içindeki tüm kayıtlar için doğum tarihlerini sıraladığından ve tüm hastaların tanımlayıcısını doğum tarihlerine göre sıralı olarak döndürdüğünden, en karmaşık sorgudur. Bu yüzden, hem şifresiz, hem şeffaf şifreli, hem de şifreli veri üzerindeki çalışma süresi uzundur. Tablo 5'deki süreler incelendiğinde 150.000 kayıttan sonra, sürelerde hızlı bir artış olduğu görülmektedir. Bunun nedeni, veri büyüklüğünün artması ile hafızanın dolması, ardından işlemlerin disk üzerinde yapılmasıdır.

4. Sonuçlar ve Gelecek Çalışmalar

Elektronik sağlık kayıtları içeren veri tabanlarındaki içeriğin gizliliğinin ve

mahremiyetinin sağlanması için şifreleme tabanlı sistemlere ihtiyaç duyulmaktadır. Şifreleme süreçlerinin gerçekleştirilmesinde sistemin performansı önemli bir kriter olarak karşımıza çıkmaktadır. Bu çalışmada öncelikle vekil tabanlı bir sistem olan CryptDB ile T-SQL karşılaştırılmıştır. Kolay entegre olması ve sistemin performansını iyileştirmesi nedeniyle CryptDB gibi vekil tabanlı sistemlerin kullanımı öne çıkmakla beraber, bu sistemlerin henüz stabil çalıştığını söylemek mümkün değildir. Bu nedenden dolayı, bu çalışmada T-SQL ile veri tabanının şifrelenmesi uygulamaları üzerine yoğunlaşılmış ve yapılan deneylerin sonuçları tablolar halinde verilmiştir. T-SQL kullanıldığında ve sütun bazlı şifreleme yapıldığında veri tabanında tutulan verilerin büyüklüğünün ortalama 11 kat arttığı ve sorgulama sürelerinin saniyelerden dakikalara yükseldiği gözlenmiştir. Şeffaf şifreleme yapıldığında sorgulama süresi ortalama yüzde 23,78 artmaktadır, şifreleme yapıldığındaysa da ortalama yüzde 25961,28 artmaktadır.

Günümüzün depolama imkânları düşünüldüğünde, verilerin büyüklüğünün artması ciddi bir problem değildir. Daha güçlü işlemci ve mimariye sahip sunucuların kullanılması ile de sorgulama süreçlerinin hızlanması mümkündür. Şeffaf veri şifreleme (TDE) kullanıldığında sorgulamalar çok daha hızlı gerçekleşmektedir. Ancak bu yöntemin dezavantajları, Microsoft SQL Server'ın Enterprise sürümünü gerektirmesi ve süreç içerisinde ara bellekte verilerin şifrelenmemiş olarak durmasıdır.

Veri tabanındaki bütün alanlar yerine bir kısmının şifrelenmesi (kısmi şifreleme) ya da verilerin anonimleştirilmesinin şifreleme sürecinin performansını olumlu yönde etkilediği de gösterilmiştir.

Gelecek çalışmalarda; farklı bilgisayarlar üzerinde testler yapılarak performans kıyaslanması, ayrıca bulut yapısı üzerinden de T-SQL testlerinin yapılması hedeflenmektedir. Şifrelemede kullanılan anahtarların dağıtımı, veri tabanının belli tablolarına birden fazla kullanıcının bağlanmasına izin verebilecek grup anahtarlama sistemleri ileriki çalışmalarda incelenecektir.

Kaynakça

[1] Karaarslan, E., Ergin, A.M., Turğut, N., Kılıç, Ö. 2015. Elektronik Sağlık Kayıtlarının Gizlilik ve Mahremiyeti. XX. Türkiye'de İnternet Konferansı, 1-3 Aralık, İstanbul, 217-222.

[2] Tsai, K.L., Leu, F.Y., Wu, T.H., Chiou, S.S., Liu, Y.W., Liu, H.Y. 2014. A Secure ECC-based Electronic Medical Record System, Journal of Internet Services and Information Security, Cilt. 4, No. 1, s. 47-57.

[3] Fernández-Alemán, J.L., Señor, I.C., Lozoya, P.Á.0., Toval, A. 2013. Security and Privacy in Electronic Health Records: A Systematic Literature Review, Journal of Biomedical Informatics, Cilt. 46, No.3, s. 541-562. DOI:10.1016/ j.jbi.2012.12.003

[4] Mohammed, N., Barouti, S., Alhadidi, D., Chen, R. 2015. Secure and Private Management of Healthcare Databases for Data Mining. IEEE 28th International Symposium on Computer-Based Medical Systems, 22-25 Haziran, São Carlos ve Ribeirão Preto, 191-196. DOI: 10.1109/cbms.2015.54

[5] Popa, R.A., Redfield, C.M.S., Zeldovich, N., Balakrishnan, H. 2011. CryptDB: Protecting Confidentiality with Encrypted Query Processing, 23rd ACM

Symposium on Operating Systems Principles, 23-26 Ekim, Cascais, 85-100. DOI: 10.1145/2043556. 2043566

[6] Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., Ferguson, N. 1998. Twofish: A 128-bit block cipher, NIST AES Proposal. https://www.schneier.com/academic/paperfiles/paper-twofish-paper.pdf (Erişim Tarihi: 27.10.2017).

[7] Daemen, J., Rijmen, V. 1998. AES Proposal: Rijndael, NIST AES Proposal. http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf (Erişim Tarihi: 27.10.2017).

[8] Rankins, R., Bertucci, P., Gallelli, C., Silverstein, A.T. 2015. Microsoft SQL Server 2014 Unleashed, Pearson Education, ABD, 1992s.

[9] Naveed, M., Kamara, S., Wright, C.V. 2015. Inference Attacks on Property-Preserving Encrypted Databases, 22nd ACM SIGSAC Conference on Computer and Communications Security, 12-16 Ekim, Denver, 644-655. DOI: 10.1145/2810103.2813651

[10] Patel, D., Jiang, Y. 2013. Overview of CrytpDB, CPSC 5670 Dönem Ödevi. http://www.utc.edu/center-information-security-assurance/pdfs/course-paper-5670-cryptdb.pdf (Erişim Tarihi: 27.10.2017).

[11] SQL-Like queries in CRYPTDB doesn't work, 2015. http://crypto.stackexchange.com/questions/26423/sql-like-queries-in-cryptdb-doesnt-work (Erişim Tarihi: 08.03.2016).

[12] Skiba, M. 2015. Analysis of Encrypted Databases with CryptDB, Bitirme Tezi, Bochum: Ruhr-University Bochum, Chair for Network and Data Security, s. 40, http://www.nds.rub.de/media/ei/arbeiten/2015/10/26/thesis.pdf (Erişim Tarihi: 27.10.2017).

[13] Dayıoğlu, Z. 2014. Secure Database in Cloud Computing-Cryptdb Revisited, International Journal of Information Security Science, Cilt. 3, No. 1, s. 129-147.

[14] Akın, İ.H., Sunar, B. 2014. On the Difficulty of Securing Web Applications using CryptDB, IEEE Fourth International Conference on Big Data and Cloud Computing, 3-5 Aralık, Sidney, 745-752. DOI: 10.1109/bdcloud.2014.75

[15] Coles, M. 2007. Pro T-SQL 2005 Programmer's Guide, Apress, New York, 560s.

[16] Microsoft CAPI, 2016. http://msdn.microsoft.com/en-us/library/aa380256(VS.85).aspx (Erişim Tarihi: 08.03.2016).

[17] Surveillance, Epidemiology, and End Results (SEER) Program Research Data (1973-2012), National Cancer Institute, DCCPS, Surveillance Research Program, Surveillance Systems Branch, released April 2015, based on the November 2014 submission. https://www.seer.cancer.gov (Erişim Tarihi: 08.03.2016).

[18] Natarajan, J., Bruchez, R., Coles, M., Shaw, S., Cebollero, M. 2015. Pro T-SQL Programmer's Guide 4th Edition, Apress, New York, 744s.