BŞEÜ Fen Bilimleri Dergisi
6(2), 231-242, 2019

BSEU Journal of Science
DOI: 10.35193/bseufbd.592542

e-ISSN: 2458-7575 (http://dergipark.gov.tr/bseufbd)

Araştırma Makalesi - Research Article

Dağıtık Etiketleme Modeli ile Bilgi Akış Denetimi

Çiğdem BAKIR[1*], Veli HAKKOYMAZ[2], Mehmet GÜÇLÜ[3]

Geliş / Received: 16/07/2019 ***Revize / Revised: 11/10/2019*** ***Kabul / Accepted: 10/11/2019***

ÖZ

Bu çalışmada, dağıtık veritabanlarında bilgi akış denetimi ile veri gizliliği ve kullanıcıların veri mahremiyetini sağlamak amaçlanmıştır. Bu çalışmada; ilk kez 1998 yılında Myer tarafından tanıtılmış olan dağıtık etiket modelinin dağıtık veritabanında gerçekleştirilen tüm işlemler (okuma, yazma, güncelleme, silme) için uygulanarak daha pratik ve esnek bir şekilde gerçekleştirilmesi sağlanmıştır. Bu model aktör, nesne ve etiketten oluşur. Literatürde sadece okuma veya sadece yazma işlemleri ayrı etiket kullanılarak gerçekleştirilmiştir. Aynı anda tüm işlemler için veri güvenliğini sağlayacak bir yapı geliştirilmemiştir. Bizim çalışmamızda ise nesne üzerinde gerçekleştirilen tüm işlemler için tek etiket kullanılır. Tek etikete bakılarak hangi işlem ile aktörler arasında nasıl bir yetkilendirme ve erişim denetimi yapılacağı gösterilir. Böylelikle literatüde gerçekleştirilen çalışmaların aksine dağıtık veritabanında gerçekleştirilen tüm işlemler için veri gizliliği, veri bütünlüğü ve veri tutarlılığı sağlanmıştır. Ayrıca her aktör diğerlerinden bağımsız bir şekilde kendi güvenlik ve gizlilik politikasını belirler. Etiket aracılığıyla, güvenli olmayan ulaşım kanallarında, akış kontrolü, sistemde bulunan tüm aktörlerin veri gizliliğini sağlar. Veri nesnesi, güvenli olmayan aktör ve ortamlarda güvenli bir şekilde yayılır ve paylaşılır.

Anahtar Kelimeler- Etiketleme modeli, Veri Gizliliği, Dağıtık Veritabanı, Veri Mahremiyeti, Veri Bütünlüğü

[1]*Sorumlu yazar iletişim: cigdem.bakr@gmail.com.tr (https://orcid.org/0000-0001-8482-2412)
[2]İletişim: veli@ce.yildiz.edu.tr (https://orcid.org/ 0000-0002-3245-4440)
[3]İletişim: mehmetguclu007@gmail.com (https://orcid.org/0000-0002-7507-5694)
Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, TÜRKİYE

BŞEÜ Fen Bilimleri Dergisi
6(2), 231-242, 2019

BSEU Journal of Science
DOI: 10.35193/bseufbd.592542

e-ISSN: 2458-7575 (http://dergipark.gov.tr/bseufbd)

Information Flow Control with Decentralized Labeling Model

ABSTRACT

In this study, it is aimed to provide information flow control in distributed databases, data privacy and users' data secrecy. In this study, by applying the distributed label model, which was first introduced by Myer in 1998, for all transactions performed in the distributed database (read, write, update, delete), it is provided to perform in a more practical and flexible way. This model consists of actor, object and label. In the literature, only reading or only writing transactions were carried out using separate labels. A structure that ensures data security for all transactions at the same time has not been developed. As for that in our study, a single label is used for all transactions performed on the object. By looking at the single label, it is shown that how to perform an authorization and access control between which transaction and actors. Thus, on the contrary for the studies carried out in the literature, the data confidentiality, data integrity and data consistency were ensured for all transactions performed in the distributed database. In addition, each actor sets its own security and privacy policy independently of the others. Through the label, in unsafe transport channels, the flow control ensures data privacy of all actors present in the system. The data object is spread and shared securely in unsafe actors and environments.

Keywords- Label Model, Data Confidentiality, Distributed Databases, Data Privacy, Data Integrity

BŞEÜ Fen Bilimleri Dergisi
6(2), 231-242, 2019

BSEU Journal of Science
DOI: 10.35193/bseufbd.592542

e-ISSN: 2458-7575 (http://dergipark.gov.tr/bseufbd)

I. GİRİŞ

Dağıtık veritabanı sistemlerinde veri güvenliği, verinin yetkisiz kişilerce kullanılması, değiştirilmesi ve yayılmasının önlenmesini gerektirir. Teknolojinin hızla gelişmesiyle birlikte bankacılık, sağlık, e-ticaret ve iletişim gibi birçok alanda veri güvenliği önemli bir sorun haline gelmiştir. Bilgi sızması, bilginin yetkisiz kişilerce ele geçirilmesi, bilginin değiştirilmesi, bilgi gizliliğinin sağlanamaması şeklinde ifade edilen bu sorunların çözülmesi amacıyla bilgi akış denetimi ve erişim denetimi gibi tedbirler kullanılmaktadır[1].

Veri gizliliği (confidentiality), bir verinin sadece yetkisi olan aktörlerce kullanılarak onun üzerinde okuma ve yazma gibi işlemleri yapabilmesini ifade eder. Kişisel verinin korunması (mahremiyet) ile veri gizliliği arasında ortak bazı noktalar da bulunmaktadır. Ancak, mahremiyet (privacy) ve gizlilik aynı kavramlar değildir. Literatürde pek çok makalede bu konu tartışılmıştır . Mahremiyet gizliliğe göre daha karmaşık bir kavram olup odağında insan vardır. Gizlilik haberleşme güvenliğinde kullanılan odağında veri olan şifreleme yöntemlerinin kullanıldığı kriptolojinin bir çalışma alanıdır[2]. Diğer bir ifadeyle, mahremiyetin korunması demek, kişisel veya kurumsal hassas, önemli verinin, onu kötüye kullanabilecek aktörlerin eline geçmesinin engellenmesidir.

Organizasyonların finansal verileri, hastanelerdeki hastaların tanı ve tedavi süreciyle ilgili kişisel bilgiler, bankalardaki müşterilerle ilgili kredi kart bilgileri ya da sanayide ürün tasarım bilgileri hassas ve gizli veri örnekleridir. Bu bilgilerin kaynağında, aktörler arası sirkülasyonunda ve hedefte korunması gerekir.

Örnek olarak, hasta verisi ve vergi formundaki veri, birçok aktör tarafından veri işlemek veya hesaplama yapmak amacıyla erişilmek durumundadır. Erişimden kasıt okuma ve yazma gibi işlemlerdir. Bu erişim sağlanırken, verinin aynı zamanda korunması gibi spesifik bir problemin çözümü önemli bir bilimsel katkıdır. Çalışma bu yönde bir ilk adım olmayı amaçlamıştır.

Konunun daha iyi anlaşılması için Şekil 1'de örnek bir çizim verilmiştir. Bu senaryoya göre, amaç, kayıtların her iki merkezde hızlı ve güvenli bir şekilde paylaşılmasıdır. Bir merkez veriyi güncellemesi durumunda, her iki merkezden bakıldığında birbiriyle eşit hasta kaydının gözlenmesi gerekmektedir. Her tıp merkezinde bir hastanın ad, soyad, TC kimlik numarası, doğum tarihi, doğum yeri, kan grubu, cinsiyet, telefon, adres gibi genel bilgilerinin yanında tanı, tedavi süreci, geçmiş sağlık bulguları, kullandığı ilaçlar, laboratuvar raporları, radyoloji raporları, geçirdiği ameliyatlar, kronik rahatsızlıklar, bulaşıcı hastalıklar, gebelik durumu vb. sağlıkla ilgili kişiye özel gizli kalması gereken bilgileri vardır. Ayrıca hastalarla ilgili genel ve gizli bilgilerin yanı sıra tıp merkezinin kayıt verileri bulunmaktadır. Verilen bu örnek, ortak bir bilgi nesnesi ve buna katkı yapan aktörleri göstermektedir. Başlıca aktörler hasta, doktor ve tıp merkezi personeli olarak görülebilir. Yani, bahsedilen aktörler bu bilgi nesnesine ortaklaşa sahiptir diyebiliriz. Hastalarla ilgili eksik bilgiler, tedavi sürecinin güncellenmesi gibi sorunları çözebilmek için her merkez kendi kayıtlarında değişiklikler yapar. Bir merkez diğer merkezden hasta ile ilgili bilgilere erişmek istediğinde erişmek istediği bilgileri yerel güvenlik politikalarına göre alır ve gönderir. Bilgi akış denetimi ile bu politikaların gerçeklenmesi sağlanır. Hastalarla ilgili kayıtlar, her iki merkezde yetkili olanlara gösterilmeli, diğerleri engellenmelidir.

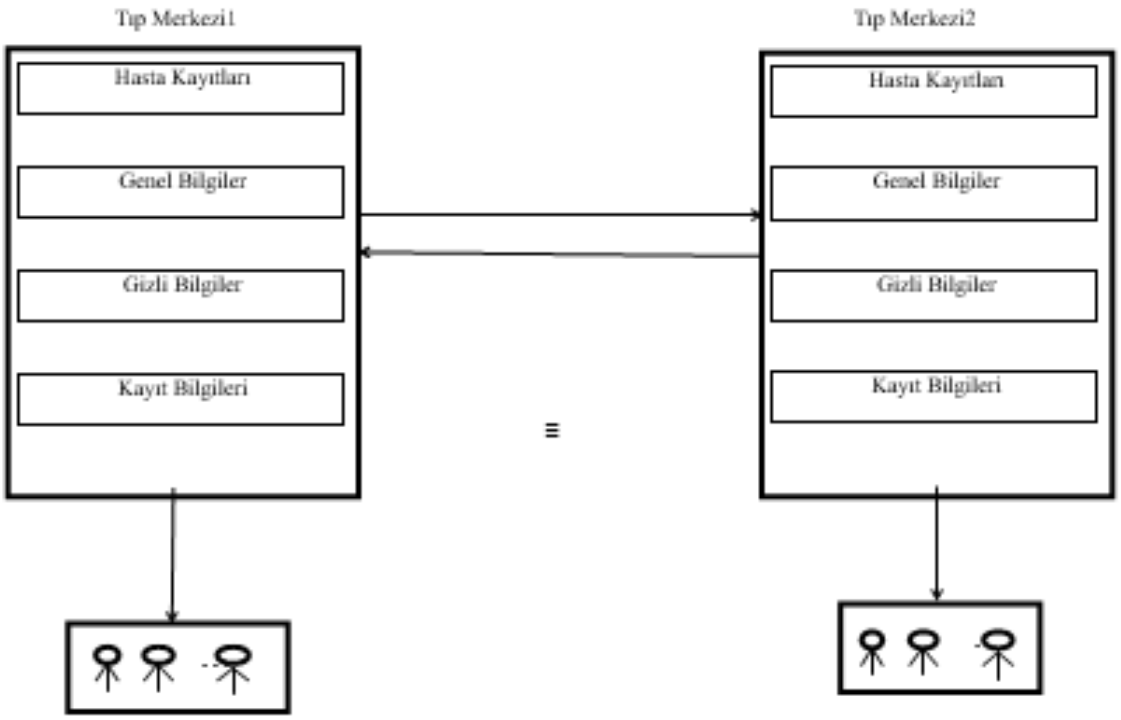
Spesifik bir örnek olarak, tıp merkezlerinde sağlık verilerinin hangi kullanıcılara hangi yetkilerle verileceği ve erişim yetkilerinin belirlenmesi önemlidir. Hem içerdeki personellerin hem de dışarıdaki yetkili/yetkisiz kullanıcıların hastaların sağlık verilerine erişim işlemleri denetime tabidir. Bu denetim, hastanın izin verdiği bilgilere, yetki verdiği aktörler tarafından erişilmesini sağlar. Ancak bu denetimin tam olarak yapılamaması yetkisiz üçüncü kişilerin bu verilere erişimi ve kullanımına ya da verilerin yayılması gibi sorunlara yol açar[3]. Bu çalışmadaki etiketleme modeli ile hasta kayıtları verisinin gizliliği sağlanmış olacaktır. Diğer bir deyişle, her hastanın, kişisel sağlık verisinin güvenlik yönetimini, gerçekleştirmek hedefi de etiketleme modeli ile yakalanacaktır.

Bu çalışmanın amacı, verilerin, dağıtık ortamda farklı kullanıcıların erişmesine izin veren ve aynı zamanda gizliliğin korunmasını sağlayan yöntem geliştirmektir. Veri sızıntısı, bilgi akış denetimi yapılmadığında ortaya çıkan bir durumdur. Bunun, muhtemelen, kişisel verilerin korunması yasasının ihlalinden, ulusal güvenliğin tehlikeye atılmasına değin bir dizi istenmeyen neticeleri olacaktır. Yetkisiz, beklenmeyen ve niyet edilmeden yapılan erişim ve bunun üçüncü kişilere sızdırılması, mahcubiyete, kurum içinde olması elzem olan halkın güven duygusunun kaybına ya da kuruma karşı yasal işlemlerin başlatılmasına neden olacaktır[4].

BŞEÜ Fen Bilimleri Dergisi
6(2), 231-242, 2019

BSEU Journal of Science
DOI: 10.35193/bseufbd.592542

e-ISSN: 2458-7575 (http://dergipark.gov.tr/bseufbd)



Şekil 1. İki tıp merkezi arasındaki bilgi paylaşımı

Bu çalışma ile dağıtık etiketleme modelinin eksik yönleri giderilmiştir. Önceki çalışmalarda okuma etiketi kullanıldığında okuma işlemi, yazma etiketi kullanıldığında yazma işlemi için veri güvenliği sağlanır. Veritabanında gerçekleştirilen tüm işlemler için veri güvenliği sağlanmamıştır[5,6]. Bizim çalışmamızda ise, etiket graf veri yapısı ile modellenerek veritabanında gerçekleştirilen tüm işlemler için tek etiket kullanılarak veri güvenliği gerçekleştirilmiştir. Bu modelin bilimsel katkısı ise, böyle birçok paydaşların kullanımına açık veriler sadece yetkili aktörlerce rahatça kullanılırken, yetkisiz üçüncü aktörlerce kullanılmasına izni vermemesidir. Aynı zamanda ortak olarak kullanılan kaynakların bilgi sızıntısına sebep olmadan kullanılmasını sağlayan yöntemlerin araştırılmasına katkı sağlamaktır. Yani, bu çalışmada, dağıtık veritabanlarında bilgi akış denetimi ile veri gizliliğini muhafaza edecek dağıtık etiket modeli geliştirilmiştir.

Yapılan diğer çalışmalardan farkı, etiketleme modeli ile güvenilir olmayan aktörler ve ortamlarda veri gizliliğini hedeflemesidir. Verilere verilen etiketler aracılığıyla her aktör kendi güvenlik politikasını diğer aktörlerden bağımsız biçimde belirleyebilir ve diğer aktörlerden seçtiklerini yetkilendirir.

Makalenin kalan kısmı şu şekilde düzenlenmiştir: İlgili çalışmalar 2., tanımlar 3., dağıtık etiket modeli 4., sonuçlar ile gelecekte yapılacak çalışmalar son bölümde verilmiştir.

II. İLGİLİ ÇALIŞMALAR

Bir dağıtık hesaplama ortamı, graf veri yapısı ile modellenebilir. Graf, düğümler ve bu düğümleri birbirine bağlayan kenarlar kümesinden oluşan bir veri yapısıdır[7]. Eğer G graf ise, bunun tanımı;

$G = (V,E)$

$V(G) = \{v1, v2, \ldots, vN\}$

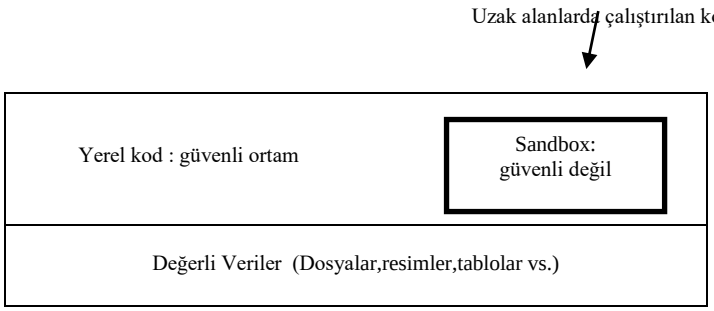
$E(G) = \{e1, e2, \ldots, eM\}$ olmak üzere $E \subseteq VxV$'dir.

BŞEÜ Fen Bilimleri Dergisi
6(2), 231-242, 2019

BSEU Journal of Science
DOI: 10.35193/bseufbd.592542

e-ISSN: 2458-7575 (http://dergipark.gov.tr/bseufbd)

Dağıtık ortam, saklama (storage), çalışan (worker) ve yayın (dissemination) olmak üzere üç çeşit düğüm ile gerçekleştirilir. Kenarlar, bir veri nesnesinin bir düğümden diğerine geçişini gösterir. Saklama düğümü, nesneleri kalıcı bir şekilde saklar. Yayın düğümü kendisinden, nesne istediğinde nesnenin kopyalanmasını sağlar. Çalışan ve yayın düğümünün, nesneleri almaya yetkisi olup olmadığına bakar (gizlilik politikası ile). Çalışan düğüm, programları çalıştırır. Yayın düğüm, ise sık kulanılan nesneleri gruplar halinde saklar[8].



Şekil 2. Java Güvenlik Modeli

Güvenli olmayan kodların yüklenmesine karşı her program bazı güvenlik önlemleri alır. Örneğin, Java uzak sitelerden kod indirilmesine izin verir. Bu, bazı gizli bilgilerin bu sitelere aktarılmasına neden olabilir. Kullanıcılar Java'da applet (kod parçacığı) yükleyerek kendi verilerini hesaplar. Kod parçacığı, Java kodlarının, Internet üzerinden yayınlanıp tarayıcı içerisinde çalışmasını sağlar. Ancak kullanıcı bunu indirdiğinde, çeşitli gizli dosyalara erişebilir ve çeşitli sitelere bu bilgilerin aktarılmasına sebep olabilir. Bu tür kullanıcılardan gelen kod parçacıklarından korunmak gerekir. Java bu risklere karşılık 'sandbox' güvenlik modelini kullanmıştır. Bu güvenlik modeli Şekil 2'de gösterilmiştir. Bu model daha çok yerel kodların çalışmasında güvenilirdir. Yerel uygulamalar, güvenilir olmayan kullanıcılara verilerin paylaşılmasını önler. Ancak çok büyük uygulamalardaki ve uzak alanlarda çalıştırılan güvenli olmayan kod parçacıklarının çalışmasında veri sızıntısına sebep olduğu için, sınırlı alanlarda kullanılan bir güvenlik modelidir[9].

Erişim denetimi, isteğe bağlı erişim denetimi (discretionary access control, DAC), zorunlu erişim denetimi (mandatory access control, MAC) ve rol tabanlı erişim denetimi (rol based access control, RBAC) olmak üzere üç farklı yöntemle gerçekleştirilir[10]. Erişim denetimi, bilgi akış denetimi ile kapsanan problemleri çözmede yetersiz kalır. Çünkü bu teknikler sadece aktörün yetkisini ifade eder[11]. Veri nesnelerinin yetkisiz kişilerin eline geçmesini önleme, yetkisiz kişilere iletilmesini sağlayan ulaşım kanallarının denetim altına alınmasını gerektirir. Gizlilik ihlalini önlemek için, dağıtık ortamda güvensiz düğümlerin olmasına rağmen, veri nesnelerinin güvenli bir şekilde paylaşılması, fonksiyon ve hesaplamaların gerçekleştirilmesi gerekir[12]. Veri gizliliği, alt düzey güvenlik yöntemi olan şifreleme ile de gerçekleştirilmektedir. Şifreleme, verilerin bir anahtar ile farklı bir biçime dönüştürülmesine denir. Verilerin bir göndericiden bir alıcıya ulaşmasından sonra, alıcı şifrelenmiş metni düz metne dönüştürür. Örneğin, RSA (Rivest-Shamir-Adleman) şifreleme yöntemi için public ve private anahtarlar kullanılmaktadır. Ancak, alt düzey güvenlik tekniği olan şifreleme tekniği, dağıtık ortamlarda anahtar dağıtım problemi ortaya çıkarır. Anahtar yani şifrenin kullanıcılara dağıtılması, kendi başına bir güvenlik problemidir[13].

Merkezi yerine tüm aktörlerin güvenlik gereksinimini tanımladığı dağıtık güvenlik modeli önerilmiştir [5,6,14,15]. Bu çalışmada, anahtar olarak etiket kullanılmıştır ve etiketler aracılığıyla veri erişimi denetlenmektedir. Bizim modelde birden fazla kaynaktan sağlanan veriler o sistemdeki bazı aktörlerin ortak verisi olarak kabul edilmiştir. Bu veri ancak veri sahiplerinin ortak onayı alınarak serbest bırakılır. Diğer teknikler güvenli kullanıcılar, güvenli nesneler yani güvenli ortamlarda bilgi akışını sağlarken; etiket modeli tekniği birbirine güvenmeyen aktörler yani güvenilir olmayan ortamlarda da bilgi akışı denetimi yapar. Bu modelde her kullanıcı, kendi güvenlik politikasını diğerlerinden bağımsız olarak belirler.

Erişim yetkilerinin gözden geçirilmesi konusu bilgi güvenliği başlığı altındaki zahmetli çalışmalardandır. Bu konuda yapılan çalışmalar kurumlar içerisinde zaman ve iş gücü maliyeti yüksek

çalışmalardır. Özellikle karmaşık veri tabanı yapıları içerisinde erişim yetkilerinin gözden geçirilmesi her zaman maliyet etkin bir şekilde yapılamamakta, bu nedenle ihmal edilebilmekte veya yeterince nitelikli biçimde gerçekleştirilmemektedir [16,17]. Bizim çalışmamızda erişim denetimi ve yetkilendirme aktörlerin isteğine uygun veri sızıntısına sebep olmadan bilgi akış denetimi gözeterek sağlanır. Aktörler pratik ve esnek bir şekilde kendi güvenlik, gizlilik ve bütünlük politikalarını oluşturabiliyor. Aynı zamanda işleri bittiğinde rahatlıkla bu politikaları değiştirebiliyor ya da tamamen silebiliyor. Bu da çalışmamızda çalışma zamanında ve hem statik hem de dinamik bir şekilde gerçekleştiriliyor. Çoklu nesne ortamlarında, güvenli olmayan birçok aktörün erişim sağladığı ortamlarda rahatlıkla aktörlerin kendi belirlediği güvenlik politikalarıyla verilerini koruması amaçlanmıştır. Diğer çalışmalardan farklı olarak veri gizliliği, veri bütünlüğü ve veri tutarlılığını birlikte sağlamasıdır.

Schultz ve arkadaşları, kullanıcıların veri erişimlerini otomatik olarak takip edilmesini sağlayan bir platform geliştirmişlerdir. Bir kullanıcı sisteme her bir işlem için ayrı ayrı giriş yaptığından dolayı yetki kontrolü yeniden gerçekleştirilir. Kullanıcı her aşamada yetki kontrolü yapmak zorunda kalır. Sadece bir aşamada kontrol yapmadığı takdirde veri gizliliği ihlal edilir. Bu durum yetkinin otomatik olarak takip edilebilmesi ihtiyacını doğurmaktadır[18]. Bizim çalışmamızda ise hem yetki verilmesi hem de yetkinin geri alınması için ayrı bir kontrole ihtiyaç bulunmamaktadır. Çalışmamızda okuma, yazma, güncelleme, silme gibi her bir işlem için ayrı bir yetkilendirme ya da erişim denetimine gerek yoktur. Ayrıca kötü niyetli aktörlerin veriye erişimlerinin takipleri ile bilgi ifşasının önüne geçilmeye çalışılmıştır.

III. TANIMLAR

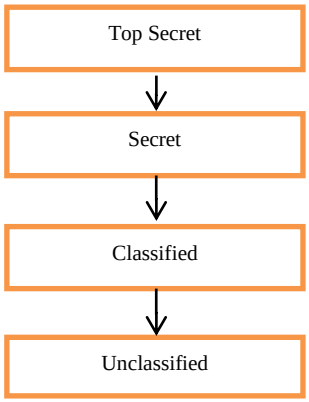
Bu bölümde, çalışmada kullandığımız bazı tanımlar verilecektir[19,20]:

Erişim Denetimi: Bir veriye kimlerin erişeceğini ve bu kişilerin veri üzerinde hangi işlemleri yapabileceğini gösterir.

Veri Akış Denetimi: Bilgilerin yetkisiz kişilerin eline geçmesini önlemek için gizli kanalların kontrol edilmesidir. Verinin yetkisiz kişilere iletilmesine imkan tanıyan ulaşım kanallarını tıkayarak bilgi sızıntısının önüne geçer.

Declassification (downgrading): Verinin güvenlik seviyesinin üst güvenlik seviyesinden daha alt güvenlik seviyesine düşürülmesidir. Veri bir alt güvenlik seviyesine aktarılarak, gizlilik derecesi azaltılmış olur.

Çok katlı güvenlik seviyesi: Kullanıcı ve bilgi nesneleri için farklı piramidal seviyelerin tanımlanması ve bu seviyelere bu aktör ve nesnelerin atanmasıdır. Şekil 3'de 4 tane farklı güvenlik seviyesi gösterilmiştir. Top secret, en üst güvenlik seviyesi iken; unclassified, en alt güvenlik seviyesini ifade eder.



Şekil 3. Güvenlik Seviyeleri

Yeniden etiketleme: Etiketin güvenlik seviyesi azalmamak şartıyla, etikette güvenlik politikalarında değişiklik yapmaktır.

Aktör: Aktör, veri sahipleri ile veriler üzerinde yetki alma ve verme gibi işlemleri gerçekleştiren kullanıcı ya da kullanıcı gruplarını içerir.

Nesne: Aktörlerin verilerini ifade eder.

Etiket: Aktörler tarafından verilen güvenlik politikaları listesinden oluşur. Bir dizi güvenlik politikasından oluşur. Her güvenlik politikası, veri sahiplerinden bir ortağı ve bu ortağın veriyi kimlere kullandıracağını (örn. okuyucu olarak) gösteren bir ifadedir.

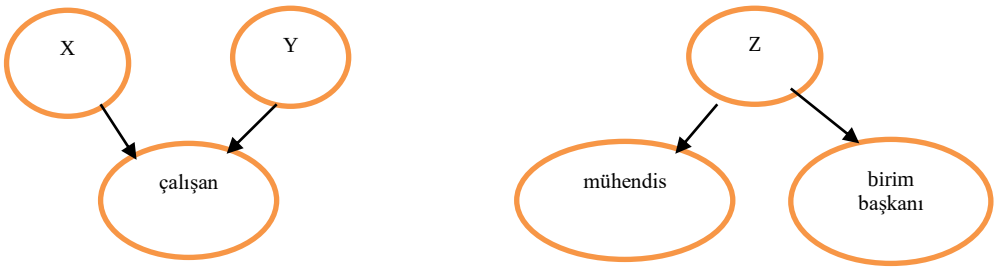
Mahremiyet: Kurum ve kullanıcıların kendi verilerinin kim tarafından, ne zaman ve ne kadarını kullanacağı olgusudur. Kullanıcıların izinleri doğrultusunda veri erişiminin sınırlandırılmasıdır.

IV. DAĞITIK ETİKETLEME MODELİ

Dağıtık etiket modeli farklı aktör, nesne ve etiketlerden oluşur[5].

A. Aktör

Her aktör veri gizliliği için verilerini etiketler. Yani, her bir veri nesnesi ile eşlenik bir etiket tanımlanır. Ayrıca her aktör ayrı ayrı bu güvenlik politikalarını güvenli bir şekilde değiştirme yetkisine sahiptir [14]. Güvenilir olmayan aktörler ve ortamlar için bu model geliştirilmiştir. Her aktör birbirinden bağımsız şekilde kendi politikasını değiştirerek yeniden etiketleme yapar(relabeling). Güvenli yeniden etiketleme (safe relabeling) yapabilmek için tüm aktörlerin güvenlik politikalarını 'emin' bir işlem ile etiketlemesi gerekir[16].



Şekil 4. Aktör hiyerarşisine örnekler

Dağıtık etiket modelinde aktörler sahibi oldukları verileri değiştiren grup ya da rollerden oluşur[16]. Aktörler diğer aktör ya da aktör gruplarına kendi verilerini okumasına izin verir. Bu izin verme işlemi aktör hiyerarşisinde gösterilir. Şekil 4'de örnek bir aktör hiyerarşisi gösterilmiştir. Bu şekilde X ve Y çalışan grubunun temsilcileridir. Çalışan Z ise mühendis ve birim başkanı olmak üzere iki yetki ve görevi vardır. Aktör hiyerarşisindeki izin verme işlemi geçişlidir. Örneğin; X'in Y aktörüne yetki vermesini X→Y şeklinde gösterelim. Eğer X→Y ve Y→Z ise X→Z vardır.

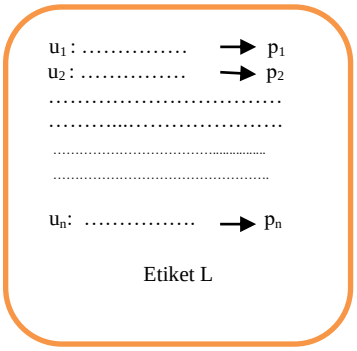
B. Etiket

Şekil 5'de bir etiketin içeriği gösterilmiştir. Burada u_1, u_2,…, u_n sistemde bulunan aktörlerden veri nesnesinin sahiplerini gösterirken; p_1, p_2,…, p_n yani L etiketindeki herbir içerik tanımı ise ilgili aktörün bu ortak veriye ilişkin güvenlik politikasını göstermektedir. Veri nesnesinin sahibi her aktör etikette kendi politikasını belirler. Bundan sonra, herhangi bir aktör bu veri nesnesini etiketiyle beraber diğer aktörlere gönderir.

BŞEÜ Fen Bilimleri Dergisi
6(2), 231-242, 2019

BSEU Journal of Science
DOI: 10.35193/bseufbd.592542

e-ISSN: 2458-7575 (http://dergipark.gov.tr/bseufbd)

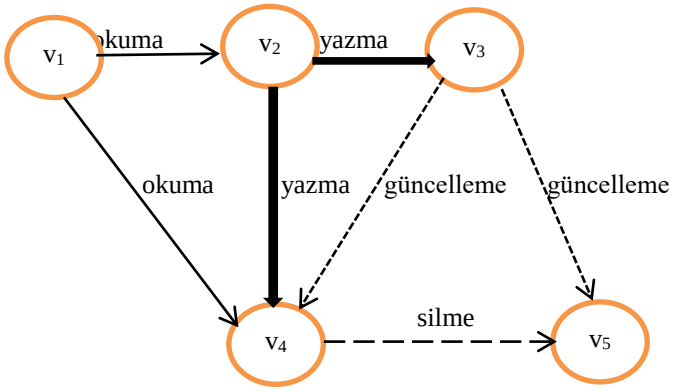


Şekil 5. Bir veri nesnesi için etiket örneği

C. Etiketin Graf ile Gösterilmesi

Literatürde yapılan daha önceki çalışmalarda [5,6,14,15] nesne üzerinde gerçekleştirilen herbir işlem (okuma, yazma) için ayrı bir etiket kullanılmıştır ve sadece okuma ile yazma işlemi yapılmıştır. Bizim tarafımızdan önerilen çalışmada ise nesne üzerinde gerçekleştirilen tüm işlemler (okuma, yazma, güncelleme, silme) tek etiket kullanarak gerçekleştirilir. Böylelikle tek etikete bakılarak hangi işlem ile aktörler arasında nasıl bir yetkilendirme biçimi olduğu gösterilir.

Bir etiket, bir graf ile gösterilebilir. G grafı için belirlenen etiket LG olsun. Dağıtık veritabanında hangi işlemin yapılacağı okun şekline göre belirlenir. Okuma, yazma, güncelleme ve silme işlemlerinin her biri için farklı bir ok kullanılır. Böylelikle tek etiket ile hem daha pratik hem de daha güvenli bir yetkilendirme ve erişim işlemi gerçekleştirilir.



Şekil 6. Etiketi modelleyen bir graf G

L_G={owner:readers:writers:updates:deletes} olmak üzere beş kısımdan oluşur. Buradaki owner (veri sahibi) etiketlenen nesnenin sahibi olan aktörleri, readers(okuyucu) ise veri sahipleri tarafından kendisine okuma işlemi için yetki verilen aktörleri, writers (yazıcı) ise veri sahipleri tarafından kendisine yazma işlemi için yetki verilen aktörleri, updates (güncelleyici) ise veri sahipleri tarafından kendisine güncelleme işlemi için yetki verilen aktörleri, deletes (silici) ise veri sahipleri tarafından kendisine silme işlemi için yetki verilen aktörleri göstermektedir. Şekil 6'daki graf G ile gösterilen etiketi, L_G, yazım biçiminde aşağıdaki şekilde verebiliriz:

BŞEÜ Fen Bilimleri Dergisi
6(2), 231-242, 2019

BSEU Journal of Science
DOI: 10.35193/bseufbd.592542

e-ISSN: 2458-7575 (http://dergipark.gov.tr/bseufbd)

L_G={ v_1:v_2,v_4; v_2:v_3, v_4; v_3:v_4, v_5; v_4:v_5, v_5 }

Bir etiketi oluştururken kullanılan noktalı virgül politikaları birbirinden ayırır. Buna göre, L_G etiketinde {v_1:v_2,v_4}, {v_2:v_3, v_4}, {v_3:v_4, v_5}, { v_4: v_5} ve { v_5: } olmak üzere beş politika vardır. v_1, v_2, v_3 ve v_4 L_G etiketinin ait olduğu veri nesnesinin sahiplerini; v_2, v_3, v_4 ve v_5 ise veri sahipleri tarafından nesne üzerinde çeşitli işlemler (okuma, yazma, güncelleme, silme) için yetki verilen aktörleri gösterir.

İlk politika nesne üzerinde okuma işlemini göstersin:

$v_1 \rightarrow v_1$, $v_1 \rightarrow v_2$, $v_1 \rightarrow v_4$ kenarları ile ifade edilmiştir. Bunun anlamı, v_1 aktörü v_1 ,v_2 ve v_4 aktörlerine verisini okuyabilmesi için izin veriyor.

İkinci politika nesne üzerinde yazma işlemini göstersin:

$v_2 \rightarrow v_2$, $v_2 \rightarrow v_3$, $v_2 \rightarrow v_4$ kenarları ile ifade edilmiştir. Bunun anlamı, v_2 aktörü ise v_2 ,v_3 ve v_4 aktörüne verisini yazması için izin veriyor.

Üçüncü politika nesne üzerinde güncelleme işlemini göstersin:

$v_3 \rightarrow v_3$, $v_3 \rightarrow v_4$, $v_3 \rightarrow v_5$ kenarları ile ifade edilmiştir. Bunun anlamı, v_3 aktörü v_3, v_4 ve v_5 aktörüne verisini güncelleyebilmesi için izin veriyor.

Dördüncü politika nesne üzerinde silme işlemini göstersin:

$v_4 \rightarrow v_4$, $v_4 \rightarrow v_5$ kenarı ile ifade edilmiştir. Bunun anlamı, v_4 aktörü v_4 ve v_5 aktörüne verisini silmesi için izin veriyor.

Son politika;

$v_5 \rightarrow v_5$ kenarı ile ifade edilmiştir. Bunun anlamı, v_5 kendinden başka kimseye verisi üzerinde hiçbir işlem yapma yetkisi vermiyor.

D. Aktörler Arası Veri Transferi

u_j aktörünün u_i aktöründen gelen veriyi alıp üzerinde çeşitli işlemler (okuma, yazma, güncelleme, silme) yapabilmesi için, u_j aktörünün bu verinin etiketi olan L'de bir politikanın veri sahibi ya da yetki verilen tüm listeler içerisinde yer alması gerekir. Bunu aşağıdaki koşul ile ifade ediyoruz.

Veri Üzerinde İşlem Yapma Koşulu:

$i \neq j$ olmak üzere, u_j aktörünün L etiketli veri transferi koşulunu aşağıdaki gibi ifade edebiliriz:

if $\{1 \leq i \leq n; \forall_i\, u_j \in reader_i[L], writer_i[L], updater_i[L], delete_i[L]\}$ or $\{1 \leq i \leq n; \exists_i,\, u_j \in owner_i[L]\}$
{
u_j has permission to read,write, update and delete data w/ label L
}
else
{
u_j has no permission to read,write, update and delete data w/ label L
}

Alınan bir verinin iletilip iletilmeyeceği bu koşul ile denetlenecektir. Eğer bu koşulu sağlamıyorsa, u_j bu veri üzerinde işlem yapamaz. Ancak bir uçtan diğer uca veri nesnesi transferinde aracılık yapmış olur.

BŞEÜ Fen Bilimleri Dergisi
6(2), 231-242, 2019

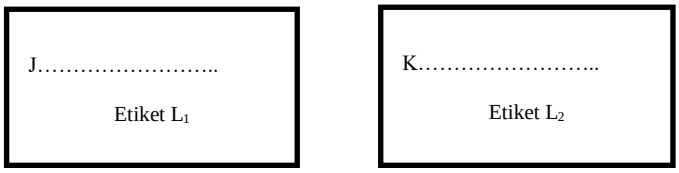
BSEU Journal of Science
DOI: 10.35193/bseufbd.592542

e-ISSN: 2458-7575 (http://dergipark.gov.tr/bseufbd)

E. Kısıtlama ile Yeniden Etiketleme

Her veri nesnesi bir etikete sahiptir. Bir nesneye yeni bir değer atandığı vakit, bu nesnenin aldığı değeri etiketinde de göstermek gerekir. Eski etiketteki tüm politikalara yeni kısıtlamalar koyarak yeni etiket değeri belirlenir. Nesnenin yeni etiket değeri en az eski etiket değeri kadar kısıtlayıcı olmalıdır. Bu, kısıtlama ile yeniden etiketleme kuralıdır.

$L_1 \subseteq L_2$ ifadesinin anlamı L_1 etiketindeki politikaların L_2 etiketindeki politikalara eşit olması gerekir ya da L_2 etiketi L_1 etiketindeki politikalara ek olarak başka politikalar da içerebilir. Kısacası; L_2 etiketi en az L_1 etiketi kadar kısıtlama ya da daha fazla kısıtlama içerir.



Şekil 8. Yeniden etiketleme

Şekil 8'de J, L_1 etiketinin politikası, K ise L_2 etiketinin politikası olmak üzere L_1 'den L_2 'ye yeniden etiketleme ile geçiş yapabilmek için gerekli olan kural eşitlik 1'de verilmiştir[6]:

$$L_1 \subseteq L_2 \rightarrow \text{owner}(K)=\text{owner}(J) \text{ve} \text{readers}(K) \subseteq \text{readers}(J) \quad (1)$$

Kısıtlama ile yeniden etiketlemeye örnekler verelim:

L_1 (eski etiket) $\subseteq$ L_2 (yeni etiket) sağlayan örnekler;

Örnek 1: {X:Y,Z}⊆ {X:Y}. L_1 etiketinin Y ve Z aktörleri okuyucuları iken; L_2 etiketi Z okuyucusunu kaldırarak verisini sadece Y aktörünün görmesine izin vermiştir.(okuyucu kaldırma ile yeniden etiketleme)

Örnek 2: {X:Y}⊆ {X: ,Z:T}. L_2 etiketi Y okuyucusunu kaldırmış ve {Z:T} yeni politikasını ekleyerek kısıtlamayı arttırmıştır.(okuyucu kaldırma ve politika ekleme)

Örnek 3: {X:Y,Z}⊆ {X:Y;X:Z}. L_1 ve L_2 eşit kısıtlama içerir.

Etiketteki veri sahipleri kendi politikalarını silme ya da okuyucu kaldırma gibi çeşitli kısıtlayıcı işlemler ile verinin yayılımını kontrol eder. Bu yeniden etiketlemenin amacı güvenli declassification gerçekleştirmektir. Veri sahibinin eklediği okuyucu, diğer veri sahipleri tarafından da eklenmesi halinde, bu biçimde yeniden etiketlenen veri, bu okuyucu tarafından okunur.

V. SONUÇLAR

Bu çalışmada dağıtık veritabanlarında veri güvenliği problemi, özellikle veri akış denetimi ile ilgili dağıtık etiket modelini kullanılmıştır. Güvenlik ve gizlilik politikalarının uygulanması için aktör, nesne ve etiket olmak üzere üç temel kavram vardır. Her aktör verileri etiketleyerek kendi güvenlik ve gizlilik politikasını belirler. Böylelikle veri gönderiminde veri gizliliği, mahremiye, veri bütünlüğü ve veri tutarlılığı sağlanır.

Bizim çalışmamızda aktörler arasında yetki verme ve alma işlemlerinin her ikisi de yapılır. Çalışmamızda okuma, yazma, güncelleme, silme gibi her bir işlem için ayrı bir yetkilendirme ya da erişim denetimi yapılmaz. Erişim denetimi ve yetkilendirme işlemleri etiketler aracılığıyla yapılır. Önceki çalışmalardan farklı olarak dağıtık veritabanında yapılan tüm işlemler için veri güvenliği sağlanır. Aktörler istedikleri zaman verdiği yetkiyi geri alabilir ya da istediği aktöre yetki verebilir. Dağıtık veritabanlarına güvenlik politikalarının uygulanması sırasında oluşan zorluklar aşılmış olur.

Önceki çalışmalarda nesne üzerinde gerçekleştirilen herbir işlem (okuma, yazma) için ayrı bir etiket kullanılmıştır ve sadece okuma ile yazma işlemi yapılmıştır. Bizim tarafımızdan önerilen çalışmada ise nesne üzerinde gerçekleştirilen tüm işlemler (okuma, yazma, güncelleme, silme) tek etiket kullanarak gerçekleştirilir. Bu da önerdiğimiz modelin esnek olduğunu gösterir. Kötü niyetli aktörlerin veriye erişimlerinin takipleri ile bilgi ifşasının önüne geçilmeye çalışılmıştır.

Günümüzde banka ve şirket gibi birçok kurumda veri güvenliği önemli bir sorundur. Bu çalışma bu sorunun çözümüne yönelik önemli bir yaklaşım getirmiştir. Bu çalışmanın devamında, etiket modelinin çalışmasını gösteren prototip bir uygulama oluşturulacak ve aktörler hiyerarşisini de dikkate alan yeniden etiketleme ile model zenginleştirilecektir. Ayrıca aktör hiyerarşisindeki birtakım eksikliklerin giderilerek çalışmanın genişletilmesi amaçlanmaktadır.

KAYNAKLAR

[1] Lin, J., & Yu, W. & Zhang, N. (2017). A survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy and Applications, *IEEE Internet of Things*, 4(5), 1125-1142.

[2] Clifton, C.W./2014). Privacy Beyond Confidentiality, *In Procedings of the ACM SIGSAC Conference and Communications Security (CCS'14),* 1156-1156.

[3] Al-Jarabi, S.,& Al-Shourbaji, M.S,& Shamshirband, S.(2017). Survey of main challenges (security and privacy) in wireless body area networks for healtcare applications, *Egyptia Informatics Journal*, 18, 113-122.

[4] Gupta, B.B.,& Shingo, Y.,& Agrawal, D.H. (2018). Advances in Security and Privacy of Multimedia Big Data in Mobile and Cloud Computing, *Multimedia Tools and Applications*, 77(7), 9203-9208.

[5] Myers, A. C.,& Liskov, B.(2000). Protecting Privacy using the Decentralized Label Model, *ACM Transactions on Software Engineering and Methodology*, 9(4), 410-442.

[6] Myers, A. C.,& Liskov, B.(1998). Complete, Safe Information Flow with Decentralized Labels, *In Proc. IEEE Symposium on Security and Privacy*.

[7] Esfandiari, H.,& Hajigohayi, M.,& Liaghat, V.,& Monemizadeh, M.(2018). Streaming Algorithms for Estimating the Matching Size in Planar Graphs and Beyond, ACM Transactions on Algorithms, 14(8).

[8] Liu, J.,& Arden, O.,& George, M.,& Myers, A.C.(2017). Fabric:Building Open Distributed Systems Securely by Construction, *Journal of Computer Security*, 25(4-5), 367-426.

[9] Kim, N.Y,& Ryu, J.H.,& Kwon, B.W,& Pan, Y.(2018). CF- CloudOrch:Container fog node-based cloud orchestration for IoT networks, *The Journal of Supercomputing*, 74(12), 7024-7045.

[10] Cai, F.,& Zhu, N.,& He, J.,& Mu, P.,& Li, W.(2018). Survey of Access Control Models and Technologies for Cloud Computing, *Springer Cluster Computing*, 1-12.

[11] Aafaf, Q.,& Hajar, M.,& Anas, A.E.,& Abdellah, A.Q.(2017) Access Control in the Internet of Things: Big challenges and new opportunities, *Elsevier Computer Networks*, 12, 237-262.

[12] Elhoseny, M.,& Gustavo, O.,& Showkat, S.(2018). Secure Medical Data Transmission Model for IoT-Based Healtcare Systems, *IEEE Access Special Section on Information Security Solutions*, 6, 20596-20608.

[13] Alizadeh, M.,& Abolfazli, S.,& Zamari, M.,& Baharun,S.(2016). Authentication in Mobile cloud computing:A survey, *Journal of Network and Computer Applications*, 61, 59-80.

[14] Bakir, Ç.,& Hakkoymaz, V.(2017). Dağıtık Veritabanında Veri Etiketleme ile Bilgi Akış Denetimi, *5.Ulusal Yüksek Başarımlı Hesaplama Konferansı*, Esenler İstanbul.

[15] Myers, A. C.,& Liskov, B.(1997). A Decentralized Model for Information Flow Control, *In Proc. 17th ACM Symp. on Operating System Principles (SOSP),* 129–142.

[16] Cecchetti, E.& Myers, A.C. (2017). Nonmalleable Information Flow Control, *ACM Conference on Computer and Communication Security*.

[17] Arden, O. & Myers. (2016). A.C. A calculus for flow-limited authorization, *IEEE Computer Security Foundations*, 135-149.

[18] Cheng, W.,& Ports,& R.K, Schultz, D.(2012). Abstractions for Usable Flow Control in Aelous, *USENIX ATC'12 Proc. USENIX Conference on Annual Technical Conference*, 1-12, 2012.

[19] Cai, F.,& Zhu, N.,& He, J.,& Mu, P.,& Li, W.(2018). Survey of Access Control Models and Technologies for Cloud Computing, *Springer Cluster Computing*, 1-12.

[20] Servos, D.,& Osborn, S.L.(2017). Current Research and Open Problems in Attribute-Based Access Control, *ACM Computing Surveys*, 49(4).