Pamukkale Univ Muh Bilim Derg, 24(5), 887-897, 2018

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

İlişkisel veritabanlarının bütünlük kontrolü için ayrık kosinüs dönüşümü tabanlı kırılgan sıfır damgalama şeması

A fragile zero watermarking schema to check integrity of relational databases based on discrete cosines transform

Yasin ŞAHİN[1], Güzin ULUTAŞ[2], Mustafa Bilgehan İMAMOĞLU[3]*

[1,2,3]Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Karadeniz Teknik Üniversitesi, Trabzon, Türkiye.
yasinsahin@ktu.edu.tr, gulutas@ktu.edu.tr, bilgehan@ktu.edu.tr

Geliş Tarihi/Received: 08.05.2017, Kabul Tarihi/Accepted: 10.04.2018 doi: 10.5505/pajes.2018.22120
* Yazışılan yazar/Corresponding author Araştırma Makalesi/Research Article

Öz

Günümüzde internet kullanımının hızlı bir şekilde yaygınlaşmasıyla bilgilere erişim daha kolay bir hale gelmiştir. Fakat bu durum veritabanı bütünlüğünün korunuyor olması gibi sorunları da ortaya çıkarmıştır. Son yıllarda veritabanı bütünlüğü önemini arttıran bir konu haline gelmiştir ve bu alandaki çalışmalar gittikçe artmaktadır. İlişkisel veritabanları üzerinde yapılan çalışmaların çoğunda elde edilen damga bilgisi yine orijinal veritabanının içinde tutulmaktadır. Bu durum, veritabanı üzerinde tolere edilebilecek boyutta olsa da bozulmalara neden olmaktadır. Önerilen yöntemde ise elde edilen damga bilgisi veritabanından bağımsız olarak gönderilerek sıfır damgalama şeması uygulanmıştır. Bu yöntemde öncelikle veritabanı gruplara ayrılmaktadır. Ardından her grup bir matris olarak ele alınarak ayrık kosinüs dönüşümü (AKD) katsayıları hesaplanmaktadır. Hesaplanan AKD katsayıların pozitif veya negatif olması durumuna göre 1 veya 0 olarak etiketlenir. Her grup için AKD katsayılarının hesaplanması ve etiketleme işlemlerinin ardından grup damgaları birleştirilerek tüm veritabanı için damga bilgisi elde edilmiş olur. Veritabanı üzerindeki değerlerde meydana gelecek herhangi bir değişiklik, hesaplanan AKD katsayılarını da değiştirecek ve yapılan herhangi bir saldırı grup seviyesinde algılanabilecektir. Yapılan çalışmada kullanılan AKD, ilişkisel veritabanlarının damgalanmasında ilk defa kullanılmıştır. Deneysel çalışmalar sonucunda önerilen yöntemin, veritabanındaki değişimleri benzer çalışmalara göre daha hassas bir şekilde tespit edebildiği gözlemlenmiştir.

Anahtar kelimeler: Sayısal damgalama, veritabanı bütünlük kontrolü, Kırılgan sıfır damgalama, Ayrık kosinüs dönüşümü, Bozulma ve saldırı algılama

Abstract

Nowadays rapid increase of the internet usage makes the data easily accessible. On the other hands, this situation arises another problem like protection of the database integrity. In recent years, database integrity protection becomes more popular and number of the works in this field is increasing in each day. Most works which are related with relational databases hides watermark information, which is generated from the current database, into the same database. This situation causes the data corruptions, which may be tolerable, on the database itself. Proposed method applies zero database watermarking scheme by transferring watermark information as a side information. The method divides the database into groups as the first step and then Discrete Cosine Transform (DCT) is applied by the method on each group which is represented by a matrix. Computed DCT coefficients are labeled as zero or one according to their signs. After computing coefficients and labeling operations for each groups, the watermark information for whole database will be obtained by concatenating groups' watermarks. Computed DCT coefficients will change if any modification has been occurred on the database and any attack could be detected at the group level. DCT is used by the method to obtain watermark information from the database at first time in the literature. Experimental results indicate that the proposed method can detect tampering on the database with finer accuracy compared to similar works.

Keywords: Digital watermarking, Checking integrity of database, Fragile zero watermarking, Discrete cosines transform, Distortion and attack detection

1 Giriş

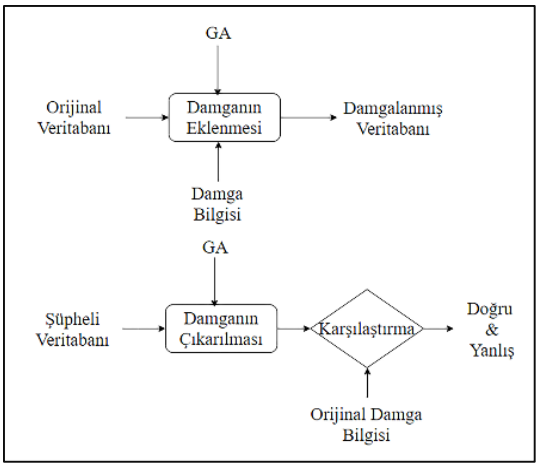
Sayısal damgalama, bir gizli bilgi saklama tekniği olup sayısal varlıkların sahiplik koruması [1], telif haklarının korunması, gizli haberleşme, kopya kontrolü, bozulma ve bütünlük kontrolü gibi alanlarda kullanılabilmektedir.

Sayısal damgalama yine bir veri saklama tekniği olan steganografi ile karıştırılmamalıdır. Steganografide saklanan veri mevcut veriden bağımsız iken, sayısal damgalama yöntemlerinde, saklanacak olan damga bilgisi yine mevcut veri üzerinden üretilmektedir.

Başlangıçta multimedya nesneleri için önerilen damgalama teknikleri üzerindeki çalışmalar, zamanla telif haklarının korunması, bozulma ve bütünlük kontrolü gibi konuları da kapsayacak şekilde genişletilerek ilişkisel veri tabanlarında da kullanılmaya başlanmıştır. Fakat multimedya nesneleri ile ilişkisel veri arasında bulunan yapısal farklılıklar nedeniyle multimedya nesneleri için kullanılan damgalama teknikleri doğrudan ilişkisel verilerin damgalanması için kullanılamamaktadır. Bu farklılıklar, ilişkisel verinin düzensiz olması, güncellemelere açık olması, damgayı saklamak için kullanılabilecek alanın sınırlı olması, kayıtların silme işlemlerine maruz kalabilmesi gibi durumlardır ve bu durumlar ilişkisel verinin damgalanma sürecinde önerilecek olan herhangi bir yöntemin üstesinden gelmesi gereken sorunlar olarak ön plana çıkmaktadır. Veri tabanlarının damgalanması ve sonrasında damganın doğrulanması için temel yapı Şekil 1'de verilmektedir.

Veri kayıtlarının internet üzerinden paylaşımının yaygınlaşmasıyla ilişkisel verinin bütünlüğünün korunması problemi de ortaya çıkmıştır [2],[3]. İnternet ortamına açılan veri, hırsızlıklara veya bozulmalara da açık hale gelmektedir. Damgalama teknikleri sahiplik bilgisinin kanıtlanmasına yardımcı olmasına rağmen veri üzerinde geri dönüşü olmayan değişiklikler de içermekte ve son verinin orijinal veriden farklı olmasına neden olmaktadır. Sayısal damgalama teknikleri, veri

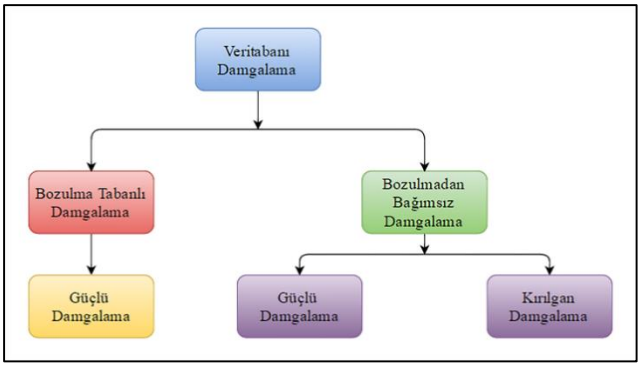
üzerinde değişiklik yapıp yapmamasına göre bozulma tabanlı ve bozulmadan bağımsız olmak üzere iki temel sınıfa ayrılmaktadır. Bozulma tabanlı yöntemler daha çok sahiplik ve telif haklarının korunması için kullanılırken [4]-[7], bozulmadan bağımsız yöntemler daha çok bütünlük kontrolü için kullanılmaktadır [8]-[10]. Ayrıca bozulma algılama ve bütünlük kontrolü için kullanılan damgalama teknikleri özel olarak kırılgan olarak adlandırılmaktadır [10]-[13].



Şekil 1: Veritabanı damgalama temel yapısı.

En iyi bilinen ve ilişkisel veri tabanları üzerinde damgalama işleminin ihtiyacını ortaya koyan ilk çalışmalardan biri olan Agrawal ve diğ. çalışmasından itibaren [2], ilişkisel veri tabanlarının damgalanması alanına ilgi arttı ve bu alanda birçok çalışmalar yapıldı. Guo ve diğ. 2007 yılında yaptıkları çalışma ile problemi tanımladı ve sayısal veri gruplarının bütünlüğünün doğrulanması için kırılgan damgalamanın önemini ortaya koydu [14]. 2008 yılına gelindiğinde ise ilişkisel veri tabanı ile optimizasyon teknikleri arasındaki bağlantı Sheab ve diğ. tarafından ortaya konuldu [15]. Bu metot, damgalama işlemini, damgalama saklama süreci için bir optimizasyon problemi olarak, damganın çıkarılma sürecini de çıkarma hatalarını minimize etmek için eşik değeri tabanlı bir tekniğe dayalı olarak düşünüldü. Khan ve diğ. 2013 yılında ilişkisel veri tabanlarının damgalaması için bir kırılgan sıfır damgalama şeması önerdiler [8]. Bu damgalama şemasında damga bilgisinin elde edilebilmesi için öncelikle rakam, uzunluk ve aralık olarak 3 alt damga üretilmektedir. Ardından üretilen bu 3 alt damga birleştirilerek tüm veri tabanı için tek bir damga elde edilmektedir. 2014 yılında ise Camara ve diğ. önerdikleri kırılgan sıfır damgalama şemasında, gizli anahtar ve birincil anahtar değerleri kullanılarak veri tabanını gruplara ayırmışlardır [9]. Veri tabanının gruplara ayrılmasından sonra her grup bir kare matris olarak değerlendirilerek her kare matris için determinant ve diyagonal minör hesaplaması yapılmaktadır. Yapılan hesaplamalar sonucunda ilgili grup için damga elde edilmektedir. Tüm gruplar için bu işlem tekrarlandıktan sonra her grubun damga bilgisi birleştirilerek tüm veri tabanı için bir damga elde edilmektedir.

İlişkisel veri tabanlarının damgalanması için literatürdeki mevcut çalışmalar belirtildiği gibi temel olarak bozulma tabanlı ve bozulmadan bağımsız olarak iki sınıfa ayrılmaktadır. Bozulma tabanlı yöntemlerin alt sınıfı olarak güçlü damgalama yöntemleri önerilmektedir. Güçlü damgalama yöntemleri genel olarak telif haklarının korunması için kullanılmaktadır. Güçlü damgalama şemasında saklanan damga bilgisi, damgayı silmek isteyecek saldırılara karşı güçlü ve algılanamaz olmalıdır. Bozulmadan bağımsız damgalama ise güçlü damgalama ve kırılgan damgalama olarak iki sınıfa ayrılmaktadır. Bu sınıflardan ilki olan güçlü damgalama teknikleri telif haklarının korunması için kullanılırken kırılgan damgalama teknikleri genel olarak veri tabanının bütünlük kontrolü için kullanılmaktadır. Kırılgan damgalama şemasında, saklanan damga bilgisi, değişikliklere karşı kırılgan olmalıdır. Böylece değişiklikler algılanabilir ve yapılan değişikliklerin yerleri tespit edilebilir. Yapılan bu sınıflandırma Şekil 2'de görülmektedir.



Şekil 2: Veritabanı damgalamanın sınıflandırılması.

Veri tabanı damgalamanın ilk amacı telif haklarının korunması olduğu söylenebilir. Verilerin izinsiz olarak kopyalanması ve dağıtılması da günümüzde veriler üzerinde büyük problemler ortaya çıkarmaktadır. Oluşturulan veri tabanları belirli anlaşmalar ve lisans sözleşmeleri ile satılabilir. Fakat veri dağıtılırken de değiştirilmediğine ve taşınırken bir saldırıya maruz kalmadığına emin olmak gerekmektedir. Bir başka problem ise veri tabanı içeriğinin doğru ve değişmemiş olması yani bütünlüğünü koruyor olmasıdır. Bu tür sorunların üstesinden gelebilmek için Agrawal ve diğ. çalışmasından itibaren birçok veri tabanı damgalama yöntemi literatürde mevcuttur.

Önerilen AKD tabanlı ilişkisel veri tabanı damgalama yöntemi, öncelikle AKD'nin bu alanda kullanılması yönüyle bir yenilik getirmektedir. Yöntem, veri tabanının gruplara ayrılması, her grup için AKD katsayılarının hesaplanması, hesaplanan katsayıların pozitif veya negatif olması durumuna göre 0 veya 1 olarak etiketlenmesi ve bu etiketlerin birleştirilerek ilgili grup için damga bilgisinin elde edilmesi işlem adımlarını gerçekleştirmektedir. Yapılan çalışmada [8] ve [9] çalışmalarıyla karşılaştırmalar yapılarak önerilen yöntemin işlem yükü ve saldırı ölçütleri üzerinden değerlendirmeler yapılmıştır. Önerilen yöntemde veri tabanı gruplara ayrılıp grup seviyesinde saldırı algılama işlemi yaptığı için, tüm veri tabanı üzerinde bir saldırı oranı tespit eden [8] çalışmasına yenilik getirmektedir. [9] çalışması üzerinde yapılan yenilikler ise (i) saldırının tespit aralığının indirgenmesi: [9] çalışmasında sütun sayısı kadar satır alınarak işlem yapılmakta, önerilen yöntemde ise sütun sayısının yarısı kadar satır ile işlem yapılmaktadır. (ii) bir gruba yapılan saldırının başka bir grubu etkilememesi: [9] çalışmasında determinant işlemi yapıldığı için, gruplar sütun sayısı veya sütun sayısının tam katı kadar satırlar içermek zorundadır. Bu koşul sağlanmadığı zaman ilk grubun ilk satırından itibaren eksik satır sayısı kadar ilgili gruba satır ekleme işlemi yapılmaktadır. Bu durum eklenen satıra bir saldırı yapılması durumunda saldırının tespit edileceği grup olarak hem satırın gerçekte olduğu grubu hem de eklenmiş olduğu grupları gösterecektir. Önerilen yöntemde ise gruplara herhangi bir ekleme yapılmadığı için saldırının

yapılmış olduğu grup hem doğru şekilde tespit edilecek hem de bir gruba yapılmış olan saldırı başka bir grubu etkilemeyecektir.

1.1 Problemin tanımı

İlişkisel veri tabanlarının damgalanması süreci işlem adımları fark edilmezlik, dayanıklılık, güvenlik ve körlük gibi sayısal damgalamanın temel özelliklerini de dikkate almalıdır. Bu özelliklerden fark edilmezlik damganın görünmez olması ve damganın saklanma sürecinin verinin kullanılabilirliğini azaltmamasını ifade etmektedir. Dayanıklılık, damganın yok etme saldırılarına karşı güçlü olmasıdır. Güvenlik, damganın saklanması sürecinde, güvenlik amaçlı bir gizli anahtar (GA) kullanılmasını belirtmektedir. Temel özelliklerden körlük ise damga çıkarma sürecinin orijinal veriye veya orijinal veriden elde edilen damga bilgisine ihtiyaç duymaması olarak ifade edilebilir. Bu özelliklerin yanında ilişkisel veri tabanlarının damgalanmasında önemli olan iki temel kısıtlama daha mevcuttur. Bunlardan ilki saklanabilecek damganın kapasitesi, diğeri ise saklanacak damga bilgisinin orijinal verinin kullanılabilirliğini azaltmamasıdır. Verilen bu kısıtlamalar ilişkisel veri tabanlarının damgalanması işleminde özellikle dikkat edilmesi gereken hususlardır. Özellikle saklanacak damga kapasitesi ve veri tabanının bütünlüğünün herhangi bir bozulma ile karşı karşıya kalmaması açısından sıfır damgalama önemli bir konu haline gelmiştir. Çünkü sıfır damgalamada mevcut veriden elde edilen damga bilgisi yine verinin içerine saklanmak yerine ek bir dosya halinde tutulmaktadır. Bu da beraberinde bazı avantajları getirmektedir. Bu avantajlardan ilki veri içerisine herhangi bir bilgi saklanmadığı içi veri bütünlüğünde en ufak bir bozulma bile meydana gelmemektedir. Diğer bir avantaj ise üretilen damga bilgisi veri içerisinde saklanmayacağı için herhangi bir damga boyutu kısıtlaması da mevcut değildir.

Literatürde ilişkisel veri tabanlarının bütünlüğünün kontrol edilmesi için önerilen kırılgan damgalama teknikleri, yaptıkları çalışmalar için bazı saldırı türleri üzerinden test işlemleri gerçekleştirmektedir. Bu saldırı türlerinden bazıları şöyle ifade edilmektedir:

Ekleme Saldırısı: Bu saldırı türünde saldırgan veri tabanına satırlar ekleyerek saldırı gerçekleştirmektedir.

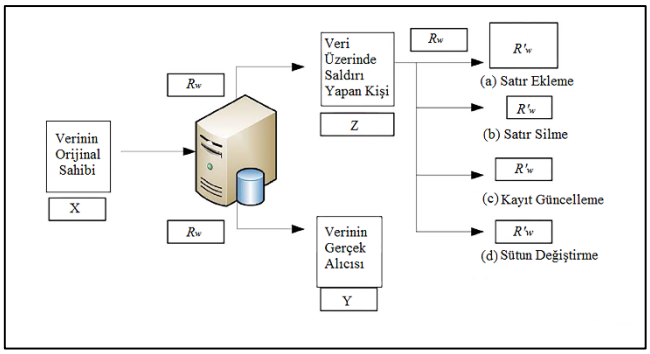
Silme Saldırısı: Bu saldırı türünde ise saldırgan veri tabanından satırlar silmektedir.

Güncelleme Saldırısı: Saldırgan bu saldırı türünde veri tabanı değerlerinden rasgele seçimler yaparak değerler üzerinde değişiklikler yapmaktadır.

Sütun Değiştirme Saldırısı: Veri tabanına saldırı yapmak isteyen kötü niyetli kişiler rasgele sütunların yerlerini değiştirerek de saldırılarını gerçekleştirebilir.

Yapılan çalışmada da önerilen yöntem için yukarıda verilen saldırılara karşı olan direnci test edilmiş ve sonuçlar değerlendirilmiştir. Yapılabilecek bu saldırılar yapısal olarak Şekil 3'teki gibi gösterilebilir.

İlişkisel veri tabanlarının bütünlüğünün korunması için önerilen yöntemlerin birçoğu meydana gelmiş olan bozulma için tüm veri tabanı üzerinden bir bozulma oranı tespit etmektedir. Bu da tek bir satırda dahi bozulma meydana gelmişse dahi tüm veri tabanı için bir bozulma verildiğini göstermektedir. Bunun yanında yapılan saldırıların tespiti için belirli bir aralık veren yöntemler de literatürde mevcuttur.



Şekil 3: Veri tabanına yapılabilecek saldırıların yapısı.

1.2 Ayrık kosinüs dönüşümü (AKD)

Bilginin ifade edildiği ve gösterildiği düzlemden başka bir düzleme aktarılarak, o düzlem üzerinde ifade edilmesi dönüşüm olarak adlandırılmaktadır. Bilgi, zaman, genlik, frekans bilgilerini kullanarak ifade edilir. Ayrık kosinüs dönüşümü, kendisini oluşturan sinyalin kosinüs fonksiyonları şeklinde gösterilerek frekans düzlemine aktarılması işlemidir. Sinyalin içerdiği değişimler bir sinyalin frekans düzlemindeki gösterimi olarak ifade edilir. İmge işlemede, sayısal imge bilgisinde sadece gerçek sayı düzleminden veriler olduğundan dolayı çoğunlukla AKD kullanılır. Bununla birlikte AKD, sinyalin enerjisini daha küçük bir alana sıkıştırarak, sinyalin daha az veriyle ifade edilebilmesini sağlamaktadır [16].

AKD görüntü sıkıştırma işlemlerinde yoğun olarak kullanılmaktadır. Birçok görüntü işleme standardında AKD işlemi 8x8 piksellik parçalara ayrılarak uygulanır. AKD işlemi 8x8 bloklardan daha büyük parçalara ayrılıp uygulandığında sıkıştırmada kayda değer iyileştirme sağlamamaktadır. AKD uygulanması sonucunda elde edilecek katsayılardan sol üst köşedeki katsayı en alçak frekans bileşeni olan DC bileşeni ifade etmektedir. Bu frekans bileşenlerinin elde edilmesi için öncelikle AKD katsayılarının hesaplanması gerekmektedir.

Yapılan çalışmada AKD'nin kullanılmasının amacı her AKD katsayısının hesaplanması aşamasında tüm matris değerleri kullanıldığından herhangi birinde meydana gelmiş değişikliğin üretilen katsayılar üzerinde değişiklikler meydana getirecek olmasıdır. Bu değişiklikler de orijinal veri tabanından üretilmiş olan damga bilgisi ile şüpheli veri tabanından üretilecek olan damga bilgilerinin farklı olmasına neden olacaktır ve böylece yapılmış olan herhangi bir saldırı sonucunda algılama gerçekleştirilmiş olacaktır. (1) formülünün uygulanması sonucunda elde edilecek olan AKD katsayıları pozitif veya negatif olabilmektedir. Bu katsayılar pozitif ise 1 negatif ise 0 olarak etiketlenmektedir. AKD katsayılarının hesaplanması ise (1)'deki formül ile gerçekleştirilmektedir.

$$F(u,v)=\left(\frac{4}{N.M}\right)^{\frac{1}{2}} w(i)w(j) * \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} \cos\left[\frac{\pi.u}{2.N}(2i+1)\right]\cos\left[\frac{\pi.v}{2.M}(2j+1)\right].f(i,j) \quad (1)$$

Formülde görülen N satır sayısını, M ise sütun sayısını, f(i,j) matrisin ilgili hücre değerini ve F(u,v) hücre için elde edilen AKD katsayısını ifade etmektedir. Ayrıca formülde yer alan w(k) fonksiyonu da (2) formülünde olduğu gibi değer almaktadır:

$$w(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{k=0 durumunda} \\ 1 & \text{diğer durumlarda} \end{cases} \quad (2)$$

2 İlgili çalışmalar

Yapılan çalışmada elde edilen sonuçlar, [8] ve [9] çalışmaları ile işlem zamanı, saldırılara karşı dayanıklılık açısından karşılaştırılmıştır.

2.1 Khan ve diğ. kırılgan sıfır damgalama şeması

Yapılan AKD tabanlı çalışmanın karşılaştırıldığı yöntemlerden biri olan [8] çalışmasında, ilişkisel veri tabanları üzerinde bozulmadan bağımsız kırılgan sıfır şeması önerilmiştir. Bu çalışmada rakamların, her hücredeki veri değerlerinin uzunluklarının ve kullanıcı tarafından belirlenmiş aralıklar içerisindeki verilerin frekans değerleri kullanılarak damga bilgisi oluşturulmaktadır. Bu üç karakteristik bilgi için alt damgalar oluşturulmakta ve bu alt damgaların birleştirilmesiyle de ilişkisel veri tabanına ait damga bilgisi elde edilmektedir. [8] çalışmasına ait genel sözde kod ifadesi aşağıda verilmiştir.

```
(1) Wd = rakam_alt_damgası()
(2) Wl = uzunluk_alt_damgası()
(3) Wr = aralık_alt_damgası()
(4) WR = Wd || Wl || Wr
(5) EWR = Şifrele(WR, GA)
(6) WC = EWR || sahip_id || tarih || saat
(7) Sertifikalandırılmış damgayı kaydet
```

Yukarıda verilen sözde kod ifadesinde öncelikle veri tabanına ait olan alt damgalar elde edilmektedir. Bu alt damgalardan ilki rakam alt damgası olup her bir rakamın veri tabanında ne kadar tekrarlandığını belirtmektedir. İkinci alt damga ise uzunluk alt damgasıdır. Bu alt damgada her uzunluk (üç basamaklı veri sayısı, beş basamaklı veri sayısı gibi) değerine sahip verilerin frekans değerleri hesaplanır. Son olarak aralık alt damgası oluşturulmaktadır. Bu alt damgada ise kullanıcı tarafından belirlenmiş olan aralık değerleri (0-100, 101-1000, 1001-10000… gibi) içerisinde kaç adet veri olduğu hesaplanır. Üç alt damga da elde edildikten sonra bu damgalar birleştirilir. Ardından birleştirilen damga bilgisi, kullanıcı tarafından belirlenmiş olan bir gizli anahtar değeri ile şifrelenmekte ve tarih, saat ve veri sahibine ait bir id bilgisi ile sertifikalandırılarak kaydedilmektedir.

2.2 Camara ve diğ. kırılgan sıfır damgalama şeması

Yapılan çalışmada karşılaştırılan yöntemlerden diğeri ise [8] çalışması üzerinde iyileştirmeler yaparak kırılgan sıfır damgalama şeması sunan Camara ve diğ. önermiş oldukları [9] çalışmasıdır. Bu çalışmada öncelikle ilişkisel veri tabanı gruplara ayrılmaktadır. Sütun sayısı kadar satır içeren gruplar kare matris olarak ele alınmakta ve kare matrisler üzerinde determinant ve diyagonal minörler hesaplanarak ilgili gruba ait damga bilgisi elde edilmektedir. Her grup için hesaplanan damga bilgileri birleştirilerek ilişkisel veri tabanına ait damga bilgisi üretilmiş olur. [8] çalışmasında tüm veri tabanı üzerinde bir bozulma oranı verilirken, [9] çalışmasında bozulmanın meydana geldiği grubun tespiti de sağlanmaktadır. [9] çalışmasının bozulmanın yerini algılamadaki bu avantajının yanında determinant ve minör hesaplama gibi uzun işlem zamanı isteyen matematiksel işlemler içerdiği için çok daha fazla CPU zamanı tüketir. Önerilen bu yönteme ait sözde kod ifadesi aşağıda verilmiştir.

```
(1)  Veritabanının gruplara ayrılması
(2)  for i=1 to grup_sayisi
(3)      Wi = grup_damgasi_hesapla(Gi)
(4)      WR = WR || Wi
(5)  End
(6)  EWR = Şifrele(WR, GA)
(7)  WC = EWR || sahip_id || tarih || saat
(8)  Sertifikalandırılmış damgayı kaydet
```

Verilen sözde kod ifadesinde öncelikle veri tabanı gruplara ayrılmaktadır. Ardından her grup için damga bilgisi üretilmekte ve bu damga bilgileri birleştirilerek tüm veri tabanı için bir damga bilgisi elde edilmektedir. Bu işlem sonrasında [8] çalışmasında yapılan işlemler tekrarlanarak üretilen damga bilgisi şifrelenir ve sertifikalandırılarak kaydedilir.

3 Önerilen yöntem

Yapılan AKD tabanlı yöntemde, önceki bölümde belirtilmiş olan [8] ve [9] çalışmaları üzerinde iyileştirmeler yaparak yeni bir kırılgan sıfır damgalama şeması önerilmektedir. Daha önce görüntülerin damgalanmasında kullanılmış olan AKD [17], ilişkisel veri tabanları üzerinde kırılgan damgalama için ilk defa kullanılmaktadır. Önerilen yöntem temel olarak 6 aşamada gerçekleştirilmektedir. Bu aşamalar;

i. Veri tabanının gruplara ayrılması,
ii. Her grubun kendi içerisinde sütun sayısının yarısı kadar satır içeren alt gruplara ayrılması ve her alt grup için AKD katsayılarının hesaplanması,
iii. Hesaplanan katsayıların pozitif veya negatif olması durumuna göre 1 veya 0 olarak etiketlenmesi,
iv. AKD katsayıları kullanılarak elde edilen alt grup etiketleri birleştirilerek ilgili grubun damga bilgisinin elde edilmesi,
v. Tüm gruplar için aynı işlemler yapıldıktan sonra grup damga bilgilerinin birleştirilerek tüm veri tabanı için damga bilgisinin oluşturulması,
vi. Gerektiği zaman üretilmiş olan damga bilgisinin doğrulanmasıdır.

Ayrıca önerilen yöntemin işlem adımları verilirken kullanılacak olan örnek tablo da Tablo 1'deki gibidir.

Tablo 1: Önerilen yöntemin işlem adımları için örnek tablo.

1	2	3	4	5	6	7	8	9	10
26	17	74	24	56	66	76	45	13	5
59	51	98	25	1	51	22	23	14	62
71	48	10	21	6	39	75	42	6	7
15	13	14	26	65	31	53	40	13	42
88	15	18	24	11	2	23	23	22	78
99	45	2	15	1	39	52	23	15	4
86	13	6	30	15	67	23	18	78	97
85	37	7	45	27	40	36	34	91	72

3.1 Veri tabanının gruplara ayrılması

Literatürde mevcut olan kırılgan sıfır damgalama yöntemlerinin bazıları saldırının hangi bölgeye yapıldığının tespiti amacıyla veri tabanını gruplara ayırmaktadır. Önerilen yöntemde de tüm veri tabanı üzerinde bir bozulma oranı vermek yerine saldırının yapılmış olduğu bölgenin algılanabilmesi için veri tabanı öncelikle gruplara ayrılmaktadır. Veri tabanının gruplara ayrılması için GA değeri ile ilgili satırın birincil anahtar (BA) değerinin birleşiminin HASH değeri kullanılarak ilgili satırın hangi gruba dahil edildiği belirlenmektedir. Veri tabanının gruplara ayrılması ile ilgili sözde kod ifadesi de aşağıda verilmiştir

```
Girişler: ilişkisel veritabanı: R, Grup sayısı υ = ⌈ α / γ ⌉, BA
Çıkışlar: her birinin uzunluğu γ olan gruplar
begin
   for i=1 to α do
      h_i^f=hash(GA||r_i·BA||GA) //i. satırın BA
      j=h_i mod υ  // grup indeksi
      r_i satırını G_j grubuna ekle
   end for
   return (G_1,G_2,…, G_υ-1,G_υ)
end
```

Burada öncelikle veri tabanından elde edilecek olan υ grup sayısı belirlenir. Grup sayısı kayıt sayısının özellik sayısına oranı ile hesaplanır. Burada α kayıt sayısını, γ özellik (sütun) sayısını, ⌈ ⌉ ise matematiksel tavan fonksiyonunu belirtmektedir. Veri tabanındaki bütün satırlar incelenerek kullanıcının belirlemiş olduğu GA değeri ve ilgili kaydın BA değerleri metin olarak birleştirilip bu birleşimden bir HASH değeri üretilir. Ardından üretilen hash değerinin grup sayısına göre modülü alınarak ilgili kaydın hangi gruba dahil olduğu tespit edilmiş olur. Bütün kayıtlar için bu işlemler yapıldıktan sonra gruplar elde edilmiş olur.

3.2 Grupların alt gruplara bölünmesi ve AKD katsayılarının hesaplanması

Veri tabanının gruplara ayrılmasının ardından tespit aralığını azaltmak amacıyla her grup kendi içerisinde özellik sayısının yarısı kadar kayıt içeren alt gruplara ayrılır. Örnek verilecek olursa 10 özellik bulunduran bir veri tabanı için ilk gruplara ayırma işleminin ardından elde edilen grup 8 kayıt içeriyor olsun. Bu grup özellik sayısının yarısı olacak şekilde yani 5 kayıt içeren alt gruplara ayrılır. Son alt grupta özellik sayısının yarısı kadar kayıt bulunmadığı için herhangi bir ekleme yapılmadan 3 kayıt olarak kalmaktadır. Verilen örnek (10 özellik 8 kayıt içeren veri tabanı) için elde edilecek alt gruplar, 10 özellik 5 kayıt ve 10 özellik 3 kayıt olmak üzere 2 alt grup olarak belirlenmektedir. Verilen örnekten elde edilen alt gruplar da Tablo 2 ve Tablo 3'teki gibi olmaktadır.

Tablo 2: Örnek tablo için elde edilen ilk alt grup.

26	17	74	24	56	66	76	45	13	5
59	51	98	25	1	51	22	23	14	62
71	48	10	21	6	39	75	42	6	7
15	13	14	26	65	31	53	40	13	42

Tablo 3: Örnek tablo için elde edilen ikinci alt grup.

99	45	2	15	1	39	52	23	15	4
86	13	6	30	15	67	23	18	78	97
85	37	7	45	27	40	36	34	91	72

Alt gruplara ayrılmış bir grup için AKD katsayılarının hesaplanmasını gösteren sözde kod ifadesi aşağıda verilmiştir.

Sözde kod ifadesinde de görüldüğü üzere her alt grup ele alınarak AKD katsayıları hesaplanmaktadır. Hesaplanan AKD katsayılarının sonucunda ilgili grubun alt grupları için elde edilen değerler Tablo 4 ve Tablo 5'te verilmektedir.

Tablo 4: Örnek tablo için birinci alt grubun AKD katsayıları.

247	18	5	38	13	-32	44	1	31	14
29	26	-47	7	-39	-39	-23	28	20	4
4	-10	0.1	-27	7	9	4	10	20	18
-6	-31	-83	17	-4	19	-13	-29	-12	21
-7	9	1	54	-0	16	14	-0.4	-23	-2

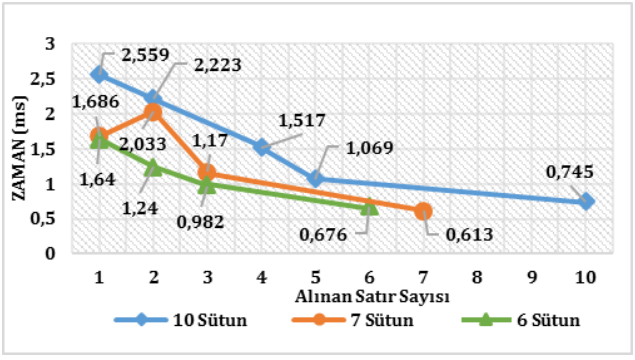
Tablo 5: Örnek tablo için ikinci alt grubun AKD katsayıları.

219	-13	74	47	70	22	19	45	-26	-18
-40	42	-18	43	-7	-17	11	-8	8	-1
-12	35	-12	30	-30	7	0.9	-21	-12	8

3.2.1 Sütun sayısının yarısının seçilmesi

Önerilen AKD tabanlı ilişkisel veri damgalama yönteminde üretilen gruplar için damga oluşturulurken sütun sayısının yarısı kadar satırlar kullanılmakta olduğu yukarıda pek çok yerde vurgulanmıştır. AKD işlemleri gerçekleştirilirken alınan sütun sayısının bu şekilde seçilmiş olması rasgele olmayıp yapılan deneysel sonuçlar sonucunda elde edilmiştir. Ayrıca literatürde sütun sayısı kadar satır alarak saldırının tespit aralığını belirleyen yöntemin mevcut olması da göz önünde bulundurulmuştur [9]. Aşağıda karşılaştırmaları grafikler ile gösterilecek olan bu deneylerde üretilen grup için çeşitli satır sayıları ile AKD uygulanarak zamansal karşılaştırmalar yapılmıştır. Yapılan deneyler sonucunda zamansal olarak en iyi sonucu veren satır sayısı, sütun sayısı değeri ile aynı olmaktadır. Alınacak olan satır sayısı azaldıkça AKD uygulanan matrislerin tükettiği zaman artmaktadır. Fakat yapılan çalışmada saldırı tespit aralığında da iyileştirmeye gidilmek istendiğinden dolayı ilgili grup için AKD uygulanacak olan matrisin satır sayısı, sütun sayısının yarısı olarak belirlenmiştir. Tek sayıda sütun içeren veri tabanları için ise sütun sayısının yarısı saldırı tespit aralığını indirgemek için hesaplanan değerin altı olarak ele alınmıştır. Örnek olarak 7 sütunlu bir veri tabanı için satır sayısının 4 yerine 3 olarak alınması gösterilebilir.

Satır sayısının sütun sayısının yarısı olarak belirlenmesinde, 10, 7 ve 6 sütunlu veri tabanları ele alınmış olup, farklı sayılarda satır içeren alt gruplar için AKD uygulanması sonucunda harcanan işlem zamanları ölçülmüştür. Bu ölçümlere ait değişim bilgileri Şekil 4'te verilmiştir.



Şekil 4: 10, 7 ve 6 sütunlu veri tabanları üzerinde elde edilen grupların farklı sayıda satır içeren alt grupları için AKD katsayılarının hesaplanma işlem zamanı.

3.3 AKD katsayılarının etiketlenmesi

AKD katsayılarının hesaplanmasından sonra her katsayı pozitif veya negatif olma durumlarına göre 0 veya 1 olarak etiketlenmektedir. Etiketleme adımına ait sözde kod ifadesi aşağıda verilmektedir.

Verilen sözde kod ifadesinden de görüldüğü gibi bu adımda yapılan işlem bir önceki adımda üretilmiş olan AKD katsayıları kontrol ederek katsayı pozitif ise 1 negatif ise 0 ile etiketlenmektedir. Etiketleme adımı sonucunda alt gruplar Tablo 6 ve Tablo 7'deki gibi olmaktadır.

```
Giriş: AKD Katsayıları Dj, Alt grup sayısı s
Çıkış: Etiketlenmiş değerler Li
Begin
   for j=1 to s do
      for k=1 to r do   // satır
         for l=1 to c do // sütun
            if Dj(k,l) >= 0 then Lj(k,l) = 1
            else Lj(k,l) = 0
         end for
      end for
   end for
   return Lj
end
```

Tablo 6: Örnek tablo için birinci alt grubun etiket değerleri.

1	1	1	1	1	0	1	1	1	1
1	1	0	1	0	0	0	1	1	1
1	0	1	0	1	1	1	1	1	1
0	0	0	1	0	1	0	0	0	1
0	1	1	1	0	1	1	0	0	0

Tablo 7: Örnek tablo için ikinci alt grubun etiket değerleri.

1	0	1	1	1	1	1	1	0	0
0	1	0	1	0	0	1	0	1	0
0	1	0	1	0	1	1	0	0	1

3.4 Alt grup etiketlerinin birleştirilerek ilgili grubun alt damgasının oluşturulması

AKD katsayılarına göre alt grupların etiketlenmesinin ardından ilgili gruba ait her alt grubun etiket değerleri birleştirilerek grup damgası elde edilir. Etiket değerlerinin birleştirilmesini gösteren sözde kod ifadesi aşağıda verilmektedir.

```
Giriş: Etiketlenmiş değerler Lj, Alt grup sayısı
s
Çıkış: Grup Damgası Wi
Begin
   for j=1 to s do
      for k=1 to r do   // satır
         for l=1 to c do // sütun
            Wi = Wi || Lj(k,l)|| #
         end for
      end for
   end for
   return Wi
end
```

Verilen sözde kod ifadesi bir grup için yapılan işlemleri göstermektedir. Bu işlem tüm gruplar için tekrarlanıp bütün grupların damgaları hesaplanır. Alt gruplar birleştirilirken araya eklenecek olan bir karakter ile saldırılar için bölge tespitinde kolaylık sağlanacaktır. Üretilen damga bilgisinin belirlenen bu karaktere göre parçalanması ile doğrulama aşamasında şüpheli veri tabanı ile orijinal veri tabanı damga bilgileri karşılaştırılırken saldırının yapıldığı bölgenin tespiti de yapılmış olacaktır. Alt grupların birleştirilmesinde kullanılan karakter *#* olarak belirlenmiştir.

3.5 Grup damgalarının birleştirilerek veri tabanı damgasının elde edilmesi

Önerilen yöntemin damga oluşturma sürecinin son işlem adımı ise elde edilen grup damgalarının birleştirilerek tüm veri tabanı için bir damga bilgisinin üretilmesidir. Ayrıca üretilen damga bilgisi kullanıcının belirlemiş olduğu bir gizli anahtar ile şifrelenir ve yine kullanıcının belirlemiş olduğu bir id ve zaman bilgisi ile sertifikalandırılarak kaydedilir. Tüm veri tabanının damga bilgisinin üretilmesine ait sözde kod ifadesi aşağıda verilmektedir.

```
Giriş: Wi, GA, υ
Çıkış: Sertifikalandırılmış damga
begin
//Tüm veritabanı için damga bilgisinin
üretilmesi
WR = W1|| $ ||W2|| $ ||…||Wυ|| $
EWR= Şifrele(WR||GA)
WC = EWR|| K_ID || UTC
Return WC
end
```

Alt grup damgalarının birleştirilip grup damgasının oluşturulmasında olduğu gibi grup damgalarının birleştirilerek tüm veri tabanı için bir damga bilgisinin üretilmesinde de her grup damgasının arasına belirlenmiş olan bir karakter eklenerek grupların da birbirinden ayrılması sağlanmıştır. Böylece saldırının öncelikle hangi grup üzerinde yapıldığı tespit edilecek bunun ardından da hangi alt grupta olduğu algılanabilecektir. Grupların birleştirilmesinde kullanılan karakter de *$* olarak belirlenmiştir.

3.6 Damganın doğrulanması

İlişkisel veri tabanı üzerinden elde edilip saldırı algılanması için saklanan damga bilgisi herhangi bir şüpheli durumda veya istenilen herhangi bir zamanda yine veri tabanından üretilen damga bilgisi ile karşılaştırılabilir. Karşılaştırma sonucunda veri tabanının saldırıya uğrayıp uğramadığı tespit edilebilmektedir. Ayrıca önerilen yöntem ile yalnızca saldırı olup olmadığı değil saldırının yapılmış olduğu bölge için de tespit yapılmaktadır. Şüpheli veri tabanından damga bilgisinin elde edilmesi süreci orijinal veri tabanından damga bilgisinin elde edilmesi ile aynı işlem adımlarını içermektedir. Şüpheli veri tabanından damganın elde edilmesi sonrasında aşağıda sözde kod ifadesinde görüldüğü gibi karşılaştırma işlemi yapılmaktadır.

```
    Girişler: şüpheli veritabanı: R', Grup sayısı
    υ = ⌈ α / γ ⌉, BA, GA
    Çıkışlar: değişen bölgeler B1, B2, …
    begin
(1) WC den EWR elde edilmesi
(2) EWR den WR elde edilmesi
(3) Şüpheli veritabanından WR' elde edilmesi
(4) G' = Split(WR','$')
(5) z = 0
(6) for i=1 to υ do
(7)     d=Length(Split(Gi','#'))
(8)     for j=1 to d do
(9)         if Lj != Lj' then
(10)            Bz = Gi' grubunun Lj' alt grubu
(11)            Z = z + 1
(12)        end if
(13)    end for
(14) end for
(15) return (B1, B2, B3, …, Bz)
(16) end
```

Damga bilgisinin doğrulanması için öncelikle Satır 1 ve Satır 2'de belirtildiği gibi orijinal veri tabanından elde edilmiş ve şifrelenerek sertifikalandırılmış damganın tekrardan elde edilmesi gerekmektedir. Bu işlem için veri sahibinin orijinal veri tabanından damga bilgisini oluştururken belirlemiş olduğu GA değeri kullanılmaktadır. GA değerinin bilinmemesi durumunda doğrulama işlemi de gerçekleştirilemez. Ardından Satır 3'te şüpheli veri tabanından orijinal veri tabanından

damga bilgisinin elde edilmesi için gerçekleştirilen işlem adımları kullanılarak şüpheli damga bilgisi üretilir. Üretilen şüpheli damga bilgisinin Satır 4'de görüldüğü gibi belirlenmiş olan karakter yardımıyla gruplara ayrılır. Satır 5'te kullanılan değişken saldırıya uğrayan alt gruplar için sayaç olarak tanımlanmıştır. Alt grup sayısı her grup için sabit olmadığı ve her gruba ait alt grupların sayısı farklılık gösterebileceğinden dolayı Satır 7'de ilgili grubun alt grup sayısı hesaplanmaktadır. Daha sonra Satır 8'de hesaplanan alt grup sayısına göre bir döngü oluşturulmakta ve Satır 9'da orijinal veri tabanına ait olan alt grup ile şüpheli veri tabanına ait olan alt gruplar tek tek karşılaştırılmakta ve karşılaştırılma sonucunda alt gruplar birbirinden farklı ise saldırıya uğramış alt grup olarak Satır 10'da görüldüğü gibi kaydedilir. Bütün gruplar için yapılan bu işlemlerin ardından ilişkisel veri tabanı üzerinde saldırıya uğramış olan bölgeler tespit edilmektedir. Tespit edilecek saldırı aralığı ise oluşturulan alt gruplar sütun sayısının yarısı kadar satır içerdiğinden dolayı en fazla sütun sayısının yarısı kadar olacaktır.

4 Bulgular

İlişkisel veri tabanlarının bütünlüğünün kontrolü için önerilen AKD tabanlı yöntem, daha önce görüntülerin damgalanmasında kullanılmış olan AKD'nin ilişkisel veri tabanlarının damgalanmasında kullanımını içermektedir. Literatürde mevcut olan çalışmalarda sıfır damgalama işlemi yapan yöntemler olsa da AKD, ilişkisel veri tabanlarının kırılgan şekilde damgalanmasında kullanılmamıştır. Bu açıdan önerilen yöntem yenilik sunmaktadır. Yapılan çalışmada kullanılan veritabanı [8] ve [9] çalışmalarında kullanılan **Covertype Data Set**'dir. http://archive.ics.uci.edu/ml/datasets/covertype adresi üzerinden erişilebilen bu veri tabanı 581.102 satır, 10 tamsayı sütunu, 44 ikili sayı sütunu ve 1 kategorik sütun içermektedir. Önerilen yöntemde, [8] ve [9] çalışmasında olduğu gibi tamsayı alanlar kullanılmış ve testler, ilgili veri kümesinde ilk 6 sütun, 7 sütun ve 10 sütun olarak ayrılıp yapılmıştır. Deneysel sonuçların elde edilmesi aşamasında Intel Core i7-4700HQ 2.4GHz işlemci, 16GB DDR3 RAM özelliklerine sahip platformda, veri tabanı yönetim sistemi olarak SQL Server © 2014, programlama dili olarak da Microsoft Visual Studio Professional © 2015 ortamında C# kullanmıştır.

4.1 AKD tabanlı yöntem ile saldırı algılanması

Bir matris için AKD uygulanarak AKD katsayılarının hesaplanmasında tüm matris değerleri kullanıldığından dolayı herhangi bir değerde meydana gelecek olan değişiklik katsayının da değişimine neden olacaktır.

Önerilen AKD tabanlı ilişkisel veri tabanı damgalama yönteminde üretilen gruplar sütun sayısının yarısı kadar satır içerdiklerinden dolayı saldırının algılanma aralığı en geniş olarak sütun sayısının yarısı kadar olmaktadır. Örnek vermek gerekirse yine 10 sütunlu bir veri tabanı tablosu üzerinde gruplar elde edildikten sonra üretilecek olan alt gruplar için en fazla sütun sayısının yarısı kadar satır alınacağından dolayı 5 satır elde edilecektir. Satır sayısının azalmasının sağladığı avantaj ise saldırı tespit aralığının azaltılmasıdır. Verilen örnek için 10 satırdan birinde veya birkaçında hata var yerine bu aralık 5'e indirilir.

Yukarıdaki paragrafta verilen ifadeyi genel olarak ifade edecek olursak gruplandırma sonucunda N satır, M sütun sayısına sahip bir grup elde edilmiş olsun. Yapılan çalışmada önerilen AKD tabanlı yöntemde grup içerisindeki satırlar M/2 satır, M sütunlu alt gruplara ayrılır. Grup içerisindeki satır sayısı M/2'nin tam katı ise tüm alt gruplar M/2 satır M sütunlu olacaktır. Son alt grubun satır sayısı M/2 değerinden küçük ise ilgili alt grubun satır sayısının M/2 olmasına bakılmaksızın mevcut satır sayısı kadar satır ve M sütunlu bir son alt grup da elde edilir. Tüm grup bu şekilde alt kısımlara ayrıldıktan sonra her alt kısım için AKD katsayılarının hesaplanması işlemi gerçekleştirilmektedir.

Aşağıda 8x8 boyutlarında bir matris için daha önce verilmiş olan saldırılar sonucunda ortaya çıkan değişiklikler Tablo 8, Tablo 9, Tablo 10, Tablo 11 ve Tablo 12'de görülmektedir.

Aşağıda verilmiş olan tablolardan Tablo 8 orijinal verileri içermekte ve orijinal değerlerden üretilen AKD katsayılarını ve etiket değerlerini göstermektedir. Tablo 9'da orijinal veriler üzerinde yapılabilecek saldırılardan sütun değiştirme saldırısını göstermektedir.

Tablo 8: Örnek orijinal matris değerleri.

Veriler								AKD Katsayıları								Etiket Değerleri							
8	10	4	7	3	4	7	10	44.00	3.72	1.96	-4.26	-4.00	-1.02	0.81	-1.32	1	1	1	0	0	0	1	0
9	10	9	8	0	4	8	3	5.81	2.55	3.01	-0.89	-0.90	4.18	-2.90	-3.69	1	1	1	0	0	1	0	0
1	2	8	7	1	8	3	6	-1.54	-1.20	4.00	-2.10	0.79	1.98	2.27	-1.14	0	0	1	0	1	1	1	0
9	10	10	4	8	8	7	2	-0.23	0.35	1.76	0.66	4.18	-4.14	-7.70	-2.05	0	1	1	1	1	0	0	0
6	10	7	7	7	2	7	8	5.50	0.06	5.19	0.74	0.00	0.17	-0.53	3.34	1	1	1	1	1	1	0	1
1	5	0	2	3	5	2	3	2.20	0.34	-4.40	1.90	1.62	-4.49	0.30	0.56	1	1	0	1	1	0	1	1
3	8	8	7	10	4	1	5	-8.29	-7.10	1.27	0.74	-0.06	0.28	4.00	-3.15	0	0	1	1	0	1	1	0
5	1	9	0	0	6	5	7	1.71	-2.65	-1.90	-5.39	3.87	-1.18	0.00	-0.73	1	0	0	0	1	0	1	0

Tablo 9: Sütun değiştirme saldırısı sonucunda etiket değerlerinin değişim tablosu.

Veriler								AKD Katsayıları								Etiket Değerleri							
8	10	7	4	3	4	7	10	44.00	2.90	0.72	-3.28	-0.75	0.44	-2.19	-5.48	1	1	1	0	0	1	0	0
9	10	8	9	0	4	8	3	5.81	3.36	4.22	-1.84	-4.06	2.76	0.02	0.36	1	1	1	0	0	1	1	1
1	2	7	8	1	8	3	6	-1.54	-1.31	3.84	-1.98	1.19	2.16	1.89	-1.66	0	0	1	0	1	1	1	0
9	10	4	10	8	8	7	2	-0.23	1.81	3.95	-1.07	-1.57	-6.73	-2.39	5.32	0	1	1	0	0	0	0	1
6	10	7	7	7	2	7	8	5.50	-0.64	4.14	1.56	2.75	1.41	-3.07	-0.18	1	0	1	1	1	1	0	0
1	5	2	0	3	5	2	3	2.20	0.44	-4.25	1.78	1.23	-4.66	0.66	1.05	1	1	0	1	1	0	1	1
3	8	7	8	10	4	1	5	-8.29	-6.85	1.64	0.44	-1.04	-0.16	4.91	-1.90	0	0	1	1	0	0	1	0
5	1	0	9	0	6	5	7	1.71	-2.14	-1.13	-6.00	1.85	-2.09	1.87	1.87	1	0	0	0	1	0	1	1

Tablo 10: Satır ekleme saldırısı sonucunda etiket değerlerinin değişim tablosu.

Veriler								AKD Katsayıları								Etiket Değerleri							
8	10	4	7	3	4	7	10	45.61	2.47	1.69	-3.48	-4.83	-0.48	0.70	0.36	1	1	1	0	0	0	1	1
9	10	9	8	0	4	8	3	6.59	4.37	2.88	-2.08	0.10	2.56	-2.59	-5.65	1	1	1	0	1	1	0	0
1	2	8	7	1	8	3	6	-0.68	-2.24	3.90	-0.67	-0.97	4.35	1.58	0.56	0	0	1	0	0	1	1	1
9	10	10	4	8	8	7	2	-1.30	1.12	1.75	-1.61	4.47	-4.16	-3.69	-3.64	0	1	1	0	1	0	0	0
6	10	7	7	7	2	7	8	2.68	-0.99	4.12	2.23	1.11	-0.68	-6.00	2.03	1	0	1	1	1	0	0	1
1	5	0	2	3	5	2	3	5.97	1.61	1.58	-0.28	0.67	-1.83	2.33	2.37	1	1	1	0	1	0	1	1
3	8	8	7	10	4	1	5	-2.25	-2.34	-4.57	3.10	1.08	-3.63	-0.36	-0.73	0	0	0	1	1	0	0	0
5	1	9	0	0	6	5	7	-7.18	-6.73	2.54	-1.24	0.30	0.97	4.88	-3.15	0	0	1	0	1	1	1	0
3	2	6	1	6	5	9	3	2.64	-1.83	-2.75	-4.93	3.92	-1.53	-1.11	-0.23	1	0	0	0	1	0	0	0

Tablo 11: Satır silme saldırısı sonucunda etiket değerlerinin değişim tablosu.

Veriler								AKD Katsayıları								Etiket Değerleri							
8	10	4	7	3	4	7	10	42.63	4.67	0.65	-3.83	-3.07	-1.73	-1.57	-2.22	1	1	1	0	0	0	0	0
9	10	9	8	0	4	8	3	4.25	0.99	5.54	-1.47	-2.02	6.17	0.62	-2.72	1	1	1	0	0	1	1	0
1	2	8	7	1	8	3	6	-1.68	-0.18	1.35	-1.50	3.34	-1.81	-2.29	-2.20	0	0	1	0	1	0	0	0
9	10	10	4	8	8	7	2	1.89	-0.48	5.71	1.13	1.98	-148	-5.33	0.34	1	0	1	1	1	0	0	1
6	10	7	7	7	2	7	8	6.58	1.52	-1.31	1.07	1.11	-3.03	0.30	3.13	1	1	0	1	1	0	1	1
1	5	0	2	3	5	2	3	-7.61	-5.86	-0.86	2.14	0.21	-1.52	3.28	-3.10	0	0	0	1	1	0	1	0
3	8	8	7	10	4	1	5	0.04	-4.06	-0.98	-5.27	3.53	-0.58	0.96	-1.19	1	0	0	0	1	0	1	0

Tablo 12: Değiştirme saldırısı sonucunda etiket değerlerinin değişim tablosu.

Veriler								AKD Katsayıları								Etiket Değerleri							
8	10	4	7	3	4	2	10	44.63	1.02	0.64	-1.27	-3.13	-0.75	1.22	-1.16	1	1	1	0	0	0	1	0
9	10	9	8	0	4	8	3	3.02	5.77	2.89	-3.57	-0.19	3.13	1.25	-5.43	1	1	1	0	0	1	1	0
1	2	8	7	1	8	3	6	-1.34	0.64	3.90	-2.80	2.55	-0.70	-1.12	0.18	0	1	1	0	1	0	0	1
9	4	10	4	8	8	7	2	-0.46	1.15	2.70	0.87	2.65	-4.79	-6.72	-5.11	0	1	1	1	1	0	0	0
6	10	7	7	7	9	7	8	5.38	0.68	2.12	1.43	1.63	-1.99	4.13	3.40	1	1	1	1	1	0	1	1
1	5	0	2	3	5	2	3	-0.27	-1.46	-2.60	1.70	2.91	-1.28	-1.87	1.33	0	0	0	1	1	0	0	1
3	8	3	7	10	4	9	5	-8.78	-3.78	-0.12	-1.30	1.25	-2.68	5.85	-2.68	0	0	0	0	1	0	1	0
5	1	9	0	6	6	5	7	3.88	-3.43	-1.13	-3.31	1.29	-2.00	-1.10	-3.35	1	0	0	0	1	0	0	0

Bu saldırı sonucunda AKD katsayıları ve etiket değerlerindeki değişim gösterilmektedir. Tablo 10'da satır ekleme saldırısı sonucunda verilerin değerleri, AKD katsayıları ve etiket değerlerindeki değişim gözlemlenmektedir. Tablo 11'de satır silme saldırısı sonucunda meydana gelen değişimler görülmektedir. Tablo 12'de ise veriler üzerinde rasgele seçilen değerlerin değiştirilmesi sonucunda etiket değerlerindeki değişimler görülmektedir. Verilen tablolarda etiket değerlerinde işaretli hücreler, saldırılar sonucunda elde edilen etiket değerleri ile orijinal verilerden elde edilen etiket değerlerinden farklı olanları belirtmektedir. İlişkisel veri tabanının damga bilgisinin oluşturulması için etiket değerleri kullanıldığından dolayı herhangi bir etiket değerinde ortaya çıkacak bir değişiklik üretilecek olan damga bilgisini de değiştirecektir. Değişen damga bilgisi ile de yapılmış olan saldırı tespit edilecektir.

Önerilen AKD tabanlı yöntem ile [8] ve [9] çalışmaları damga üretimi için işlem zamanı ve saldırılara karşı dayanıklılık açısından karşılaştırılmıştır. Bunun yanında, literatürde farklı saldırı oranlarına karşı sistemlerin ne düzeyde algılama yaptığı da değerlendirilmiştir. Önerilen yöntemde kullanılan AKD de literatürde verilmiş olan oranlar ile test edilmiştir. Test sonucunda elde edilen değerler Tablo 13'teki gibidir.

Tablo 13: Farklı saldırı oranlarına karşı AKD'nin değişim algılama sayıları.

Değişim Miktarı / Değişim Oranı	±5	±10	±20	±30
%10	12	6	23	62
%30	9	51	71	88
%50	17	17	58	115
%70	16	79	76	136
%90	21	63	80	166

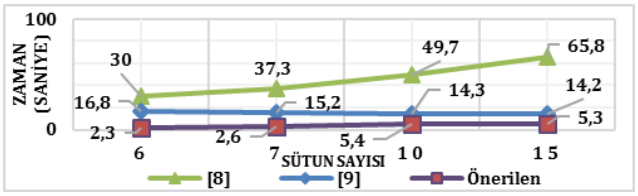
Şekil 5: Farklı saldırı oranlarına karşı AKD'nin değişim algılama sayıları.

Yapılan bu test işleminde 5x10 boyutunda bir veri tabanı tablosu üzerinde farklı oranlarda saldırılar yapılmıştır. Önerilen yöntemde, AKD katsayılarının pozitif veya negatif olması durumuna göre yapılan etiketleme işlemi daha önceki bölümlerde verilmiştir. Saldırı sonucunda değişimin algılanabilmesi için AKD katsayısında pozitif-negatif değişimi meydana gelmelidir. Tablo 13'te 100 iterasyon sonucunda elde edilen toplam etiket değişim sayıları görülmektedir. %30 (50 hücre için 15 tanesinin değiştirilmesi) değişim oranını ele alacak olursak: hücre değerleri üzerinde ±5, ±10, ±20, ±30 gibi değişiklikler yapılmıştır. Her değişiklik için tablo değerleri rasgele oluşturulmuş ve bu işlem 100 defa tekrarlanmıştır. Bu işlemler ile her saldırı için toplam etiketi değişen hücre sayısı belirlenmiştir. Değişen etiket değerleri üretilen damga bilgisini değiştireceğinden dolayı yapılan saldırı tespit edilmiş olur.

Diğer yandan ilişkisel veri tabanları için literatürde önerilen sıfır damgalama şemalarında, damga bilgisinin veri tabanı içerisinde saklanmamasından dolayı veri bütünlüğüne etki etmeyeceği için değerlendirmelerde bir ölçüt olarak ele alınmamıştır. Yapılan çalışmada ise [8] ve [9] çalışması ile önerilen yöntemin, elde edilen damga bilgisi için ihtiyaç duydukları depolama alanı olarak da karşılaştırma yapılmıştır.

4.2 İşlem zamanı olarak karşılaştırma

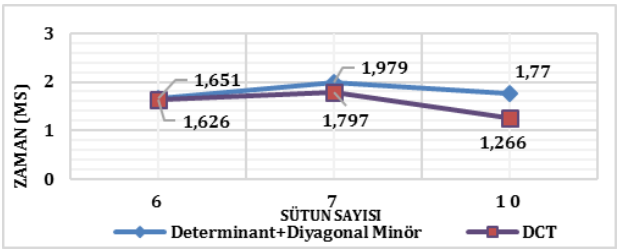
Önerilen AKD tabanlı kırılgan sıfır damgalama şeması öncelikle işlem zamanı olarak [8] ve [9] çalışmaları ile karşılaştırılmıştır. [8] ve [9] çalışmalarında verilen algoritmalara göre ilgili çalışmaların yöntemleri tarafımızca aynı bilgisayar, ortam ve programlama dili kullanılarak kodlanmış ve sonuçlar elde edilmiştir. Bu karşılaştırma sonucunda Şekil 5'teki veriler elde edilmektedir.



Şekil 5: AKD tabanlı yöntem ile [8] ve [9] çalışmasının zamansal karşılaştırılması.

Yapılan bu karşılaştırmada [8] çalışmasının en çok işlem zamanı tüketmesinin sebebi olarak rakam, uzunluk ve aralık alt damgaları için tüm veri tabanının tekrardan incelenmesidir. [9] çalışmasında, determinant ve minör hesaplamaları fazla işlem zamanı tüketmektedir. Örnek vermek gerekirse; sütun sayısının 10 olduğunu düşünürsek 10x10 boyutlu bir matrisin determinantı ve diyagonal minörleri hesaplanacaktır. Bu hesaplamalar da fazla işlem zamanı tüketmekte ve damganın üretimi içim harcanan süreyi arttırmaktadır. Önerilen AKD tabanlı yöntemde ise veri tabanının gruplara ayrılması sonucunda elde edilen gruplar üzerinde yalnızca AKD işlemi yapılacağından dolayı harcanan işlem zamanı daha az olmaktadır. Ayrıca sütun sayısının yarısının seçilmesi de damganın üretilmesi için gereken işlem zamanı üzerinde iyileştirme ortaya çıkarmaktadır. Farklı satır sayıları ile yapılan testlerde 10x10 boyutlarında bir matrisin AKD katsayılarını elde etme zamanının 5x10 boyutlarında iki matrisin AKD katsayılarını elde etmekten daha fazla zaman harcadığı önceki bölümlerde gösterilmiştir.

[9] çalışmasında damganın üretilmesi işlemi için gerçekleştirilen iki temel adım olan determinant hesaplanması ve diyagonal minörlerin elde edilmesi ile önerilen yöntemin temelini oluşturan AKD katsayılarının hesaplanmasının zamansal karşılaştırması Şekil 6'da görülmektedir.



Şekil 6: Determinant ve diyagonal minör hesaplanması ile AKD katsayılarının hesaplanmasının zamansal karşılaştırması.

4.3 Saldırılara karşı dayanıklılık

Literatürde ilişkisel veri tabanları üzerine önerilen damgalama şemalarının çeşitli saldırılara karşı dayanıklılıkları test edilmektedir. Bazı özel saldırılar olmakla beraber en sık kullanılan saldırı türleri önceki bölümlerde verilmiş olan satır ekleme, satır silme, kayıt güncelleme ve sütun değiştirme saldırılarıdır. Önerilen yöntem de bu saldırılar karşısında [8] ve [9] çalışmaları ile dayanıklılık açısından karşılaştırılmıştır. Bu karşılaştırma Tablo 14'te verilmektedir.

Tablo 14: Önerilen yöntemin [8] ve [9] çalışmaları ile saldırılara karşı dayanıklılık açısından karşılaştırılması.

Saldırı Türü	[8]	[9]	Önerilen
Ekleme	Algılanır	Algılanır	Algılanır
Silme	Algılanır	Algılanır	Algılanır
Güncelleme	Algılanır	Algılanır	Algılanır
Sütun Değiştirme	Algılanamaz	Algılanır	Algılanır

Saldırıların algılanmasına rağmen [8] çalışmasında yapılan saldırının yeri hakkında herhangi bir bilgi verilmemektedir. Bunun yerine yalnızca tüm veri tabanı üzerinde yapılan saldırı hakkında bir bozulma oranı vermektedir. Ayrıca verilerde herhangi bir değişiklik yapılmadan sütunların yerleri değiştirilerek yapılan saldırıları algılayamamaktadır.

[9] çalışmasında ise veri tabanı gruplara ayrıldığından yapılan saldırı hakkında yer tespiti de yapılmaktadır. Fakat bu yöntemde her grubun determinantı hesaplanacak olmasından dolayı kare matris oluşturulması gerekmektedir. Bu nedenden dolayı sütun sayısı kadar satır içermeyen gruplar, satır eklemesi yapılarak kare matris formatına getirilmek zorundadır. Yapılan bu işlemde kare matris oluşturmak için eklenecek olan satırlara bir saldırı yapılmışsa eklenen bu satır hem kendi grubunu hem de eklendiği grubu etkileyecektir. Böylece yapılan saldırının hangi grup üzerinde gerçekleştirildiği yanlış tespit edilmiş olabilmektedir. Ayrıca saldırının yapıldığı bölgenin aralığını sütun sayısı kadar satır kullanılmasından dolayı sütun sayısı belirlemektedir. Bu çalışmada önerilen yöntemde yazarlar yöntemlerinin güçlü yanı olarak aynı determinant değerine sahip iki farklı matrisin üretilmesinin zorluğunu göstermektedirler. Fakat bu işlem de büyük boyutlu matrisler için fazla çalışma zamanı gerektirdiğinden dolayı damganın üretilme süresi fazla olmaktadır.

Önerilen AKD tabanlı yöntemde, [9] çalışmasında olduğu gibi ilişkisel veri tabanı gruplara ayrılmaktadır. Elde edilen gruplar sütun sayısının yarısı kadar satır içeren alt gruplara ayrıldığından dolayı saldırının yapıldığı bölgenin aralığı en fazla sütun sayısının yarısı olarak tespit edilmiş olacaktır.

Alt gruplara ayırma işlemini genel olarak ifade etmek gerekirse gruplandırma sonucunda N satır, M sütun sayısına sahip bir grup elde edilmiş olsun. Önerilen AKD tabanlı yöntemde önce grup içerisindeki satırlar M/2 satır, M sütunlu alt gruplara ayrılır. Grup içerisindeki satır sayısı M/2'nin tam katı ise tüm alt gruplar M/2 satır M sütunlu olacaktır. Grubun satır sayısı M/2 değerinden küçük veya M/2'nin tam katı değilse ilgili alt grubun satır sayısının M/2 olmasına bakılmaksızın mevcut satır sayısı kadar satır ve M sütunlu bir son alt grup da elde edilir. Tüm grup bu şekilde alt gruplara ayrılmasından sonra her alt grup için AKD katsayılarının hesaplanması işlemi gerçekleştirilmektedir.

Sütun sayısının yarısının seçilme nedeni önceki bölümde verilmiştir. Diğer yandan saldırıların tespiti açısından AKD

katsayıları hesaplanırken matrisin bütün elemanları kullanılmasından dolayı herhangi bir değerin değişmesi AKD katsayısını da değiştireceği için yapılan herhangi bir saldırı tespit edilebilmektedir.

4.4 Ek depolama alanı

Yapılan çalışma, önceki bölümlerde de belirtildiği gibi kırılgan sıfır damgalama şemasına sahiptir ve veri tabanı içerisinde herhangi bir bilgi saklanmamaktadır. Fakat elde edilen damga bilgisi, veri tabanının içinde tutulmasa da belirli bir ek depolama alanına ihtiyaç duymaktadır. Bu doğrultuda önerilen yöntem, gerek duyulan ek depolama alanı açısından da [8] ve [9] çalışmalarıyla karşılaştırılmıştır. Ek depolama alanı ihtiyacı göz önünde bulundurularak yapılan karşılaştırmada [8] çalışması yaklaşık 50B ve [9] çalışması 5M ek depolama alanına ihtiyaç duymaktadır. Önerilen AKD tabanlı kırılgan sıfır damgalama yöntemi ise bzip2 sıkıştırma uygulanarak 1KB ek depolama alanına ihtiyaç duymaktadır. [8] çalışması çok daha küçük boyutta ek depolama alanına ihtiyaç duyduğu görülmekte fakat gerek saldırıların algılanmasında gerekse saldırının veri tabanının hangi bölgesine yapıldığının tespit edilmesinde önerilen yönteme göre çok daha zayıf olduğu yapılan deneysel çalışmalar ile tespit edilmiştir.

Önerilen AKD tabanlı ilişkisel veri tabanlarının kırılgan sıfır damgalama şeması ile [8] ve [9] çalışmalarının genel karşılaştırılması Tablo 15'te verilmiştir.

5 Sonuçlar

Yapılan çalışmada son yıllarda ilişkisel veri tabanlarının tek bir sisteme bağlı kalmaksızın çoklu şekilde internet üzerinden erişime açılmasıyla ortaya çıkan en büyük problemlerden biri olan bilgiler üzerinde kötü niyetli saldırılarla yapılan değişikliklerin algılanarak veri tabanının bütünlüğünün kontrol edilmesi amaçlanmış ve yapılan saldırının algılanmasının yanında, saldırının hangi bölgeye yapıldığının tespiti de gerçekleştirilmiştir. Önerilen yöntem veri tabanı üzerinde herhangi bir değişiklik meydana getirmeden, bozulmadan bağımsız kırılgan sıfır damgalama sunmaktadır. Veri tabanı bilgileri kullanılarak üretilen ve saklanan damga bilgisi daha sonra veri tabanı üzerinde herhangi bir bozulma şüphesi olduğu zaman veri tabanının bütünlüğünün test edilmesi için kullanılmaktadır. Yöntemin performans testleri yapılırken harcanan işlem zamanının yanında satır ekleme, satır silme, kayıt güncelleme ve sütun değiştirme gibi ilişkisel veri tabanları üzerinde literatürde mevcut olan bazı saldırı türleri açısından da testler yapılmıştır.

Yapılan çalışmada kullanılan AKD, görüntü işleme alanında sıkça kullanılan bir yöntem olmasına rağmen, daha önce ilişkisel veri tabanlarının damgalanmasında kullanılmamıştır. Bu yönüyle önerilen yöntem ilişkisel veri tabanlarının damgalanmasında yeni bir yöntem sunmaktadır. Önerilen yöntemin temelini oluşturan AKD, daha önce görüntülerin damgalanmasında kullanılmış olup ilişkisel veri tabanlarının kırılgan damgalanması için kullanılmamıştır. Bu yönüyle yapılan çalışma veri tabanlarının kırılgan damgalanmasında yeni bir yöntem sunmaktadır. Bu yöntem önerilirken [8] ve [9] çalışmaları referans alınmış ve bu çalışmalar üzerinden iyileştirmeler yapılarak yeni bir kırılgan sıfır damgalama şeması geliştirilmiştir.

Yapılan bu çalışmada, [8] çalışmasının grup düzeyinde saldırı algılanmasındaki eksikliği giderilmekle beraber işlem zamanı olarak daha etkin bir yöntem önerilmiştir. [9] çalışması göz önüne alındığında ise saldırı tespit aralığının indirgenmesi ve saldırı yapılmış herhangi bir satırın başka bir grubu etkilememesi açısından yenilik getirmektedir. Öncelikle veri tabanının gruplandırılmasını yapan bu yöntemde ardından gruplar üzerinde AKD uygulanmaktadır. Damgalama şemasında işlem adımları kısaca şöyle özetlenebilir. Gruplara ayrılan veri tabanı üzerinde her grup kendi içerisinde sütun sayısının yarısı kadar satırlar olacak şekilde alt gruplara ayrılır ve her alt grup için AKD uygulanarak AKD katsayıları hesaplanır.

Tablo 15: AKD tabanlı yöntem ile [8] ve [9] çalışmalarının genel karşılaştırılması.

		[8]	[9]	Önerilen AKD Tabanlı Yöntem
Saldırı olan grubun doğru bulunması	Tümü İçin	Grup seviyesinde algılama yok	Kare matris oluşturmak için yapılan satır eklemeler yanlış grup tespiti ortaya çıkarabilir	Bir gruptan başka bir gruba herhangi bir ekleme yapılmadığı için saldırı yapılan grup doğru tespit edilir
İşlem yükü	Tümü İçin	Rakam, uzunluk ve aralık frekanslarının hesaplanması	Determinant ve diyagonal minör hesaplanması	AKD katsayılarının elde edilmesi
Hata tespit aralığı	Tümü İçin	Gruplara ayırma işlemi olmadığı için Tüm veritabanı için genel bir bozulma oranı	Gruplara ayırıp sütun sayısı kadar satır içeren alt gruplar üzerinde işlemler yapıldığı için Sütun sayısı kadar satır için saldırı tespit edilir	Gruplara ayırıp her grup üzerinde en fazla sütun sayısı yarısı kadar satır içeren alt gruplar üzerinde işlem yapıldığı için En fazla sütun sayısının yarısı kadar satır için saldırı tespit edilir
Saldırı tespiti	Ekleme	Algılanır	Algılanır	Algılanır
	Silme	Algılanır	Algılanır	Algılanır
	Güncelleme	Algılanır	Algılanır	Algılanır
	Sütun değiştirme	Algılanamaz	Algılanır	Algılanır

AKD katsayıları hesaplanarak alt grup damgaları üretilmiş olur ve bu alt grup damgalarının birleştirilmesiyle grup damgası üretilir. Veri tabanından oluşturulan her grup için bu işlemler gerçekleştirildikten sonra tüm grupların damgaları birleştirilerek veri tabanı için tek bir damga üretilmiş olur. Ardından bu damga şifrelenip sertifikalandırılarak veri tabanından bağımsız olarak kaydedilir.

Veri tabanlarının bütünlük kontrolü için şüpheli veri tabanı üzerinden, orijinal veri tabanı üzerinden damga bilgisinin elde edilmesi için gerçekleştirilen işlem adımları gerçekleştirilerek bir damga üretilir. Şüpheli veri tabanından üretilen damga bilgisi, orijinal veri tabanından üretilip ve sertifikalandırılarak saklanmış olan damga ile karşılaştırılarak veri tabanının bütünlüğü hakkında karar verilir. Eğer orijinal damga ile şüpheli veri tabanından elde edilen damga bilgileri eşleşiyorsa veri tabanı bütünlüğü korunuyor demektir. Fakat orijinal damga ile şüpheli veri tabanından üretilen damga bilgisi eşleşmiyorsa yani aynı değillerse veri tabanı üzerinde bir saldırı gerçekleştirildiği ve veri tabanının bütünlüğünün bozulduğu söylenmektedir. Önerilen AKD tabanlı yöntemde ilişkisel veri tabanı üzerinde hem bütünlük kontrolü gerçekleştirilmekte hem de saldırı ile ilgili yer tespiti yapılmaktadır. Bu sayede yapılmış olan saldırının ilişkisel veri tabanının hangi satırları üzerinde gerçekleştirilmiş olduğu belirlenebilmektedir. Saldırı yapılan bölgenin tespit edilmiş olması, değişen satırlar için tüm veri tabanını kontrol etmek yerine yalnızca değişiklik meydana gelmiş olan grupların incelenerek değişiklikleri daha hızlı ve etkin bir şekilde tespit edilmesini sağlamaktadır.

Test işlemleri için literatürde bazı çalışmaların kullanmış oldukları önceki bölümde belirtilmiş **CoverType DataSet** kullanılmıştır. 10 sütunlu olan bu veri tabanı farklı sütun sayılarına ayrılarak önerilen yöntemin performans testleri yapılmıştır. Tablolardaki kayıt değerleri 0-1000 arasında rasgele değerlerden oluşmaktadır. Deneysel sonuçların elde edilmesi aşamasında Intel Core i7-4700HQ 2.4GHz işlemci, 16GB DDR3 RAM özelliklerine sahip platformda, veri tabanı yönetim sistemi olarak SQL Server 2014, programlama dili olarak da C# kullanmıştır.

6 Kaynaklar

[1] Cox I, Miller M, Bloom J. Digital Watermarking. Morgan Kaufmann: San Francisco, California, 2001.

[2] Agrawal R, Kiernan J. "Watermarking Relational Databases". *28th international conference on Very Large Data Bases, VLDB Endowment*, Hong Kong, China, 20-23 August 2002.

[3] Sion R, Atallah M, Prabhakar S. "Rights protection for categorical data". *IEEE Transactions on Knowledge and Data Engineering,* 17(7), 912-926, 2005.

[4] Gupta G, Pieprzyk J. "Reversible and blind database watermarking using difference expansion". *1st International Conference on Forensic Applications and Techniques in Telecommunications, Information and multimedia and workshop*, Adelaide, Australia, 21-23 January 2008.

[5] Farfoura ME, Horng SJ. "A novel blind reversible method for watermarking relational databases". *2010 International Symposium on Parallel and Distributed Processing with Applications (ISPA), IEEE*, Taipei, Taiwan, 6-9 September 2010.

[6] Chang CC, Nguyen TS, Lin CC. "A blind Reversible robust watermarking scheme for relational databases". *The Scientific World Journal*, vol 2013, 1-12, 2013.

[7] Zhang Y, Yang B, Niu XM. "Reversible watermarking for relational database authentication". *Journal of Computers*, 17(2), 59-66, 2006.

[8] Khan A, Husain SA. "A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations". *The Scientific World Journal*, 2013, 1-16, 2013.

[9] Camara L, Li J, Li R, Xie W. "Distortion-Free watermarking approach for relational database integrity checking". *Mathematical Problems in Engineering*, 2014, 1-10, 2014.

[10] Hamadou A, Sun X, Gao L, Shah SA. "A fragile zero-watermarking technique for authentication of relational databases". *International Journal of Digital Content Technology and its Applications*, 5(5), 189-200, 2011.

[11] Li Y, Guo H, Jajodia S. "Tamper detection and localization for categorical data using fragile watermarks". *4th ACM Workshop on Digital Rights Management (DRM '04)*, Washington, DC, USA, 25-29 October 2004.

[12] Kamel I. "A schema for protecting the integrity of databases". *Computers and Security*, 28(7), 698-709, 2009.

[13] Bhattacharya S, Cortesi A. "A distortion free watermark framework for relational databases". *4th International Conference on Software and Data Technologies (ICSOFT '09)*, Sofia, Bulgaria, 26-29 July 2009.

[14] Guo H, Li Y, Jajodia S. "Chaining watermarks for detecting malicious modifications to streaming data". *Information Sciences*, 177(1), 281-298, 2007.

[15] Shehab M, Bertino E, Ghafoor A. "Watermarking Relational databases using optimization based techniques". *IEEE Transactions on Knowledge and Data Engineering*, 20(1), 116-129, 2008.

[16] Atalar M. İmge Dizilerindeki Artıkların İşlenmesi. Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, Türkiye, 2008.

[17] Feng G, Huang X. "An improved DCT based zero - watermarking algorithm for text image". *Anti-Counterfeiting, Security and Identification (ASID), 2012 International Conference on. IEEE*, Taipei, Taiwan, 24-26 August 2012.