KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

İLİŞKİSEL VERİTABANLARININ BÜTÜNLÜĞÜNÜN SAĞLANMASI İÇİN YENİ BİR SIFIR DAMGALAMA ŞEMASI

YÜKSEK LİSANS TEZİ

Bilgisayar Müh. Yasin ŞAHİN

HAZİRAN 2016
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

İLİŞKİSEL VERİTABANLARININ BÜTÜNLÜĞÜNÜN SAĞLANMASI İÇİN YENİ BİR SIFIR DAMGALAMA ŞEMASI

Yasin ŞAHİN

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
" BİLGİSAYAR YÜKSEK MÜHENDİSİ "
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 24 / 05 /2016
Tezin Savunma Tarihi : 10 / 06 /2016

Tez Danışmanı : Yrd. Doç. Dr. Güzin ULUTAŞ

Trabzon 2016

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalında
Yasin ŞAHİN Tarafından Hazırlanan

İLİŞKİSEL VERİTABANLARININ BÜTÜNLÜĞÜNÜN SAĞLANMASI İÇİN YENİ BİR SIFIR DAMGALAMA ŞEMASI

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 17 / 05 / 2016 gün ve 1653 sayılı kararıyla oluşturulan jüri tarafından yapılan sınavda
YÜKSEK LİSANS TEZİ
olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Prof. Dr. Albert LEVİ .....

Üye : Prof. Dr. Vasif V. NABİYEV .....

Üye : Yrd. Doç. Dr. Güzin ULUTAŞ .....

Prof. Dr. Sadettin KORKMAZ
Enstitü Müdürü

ÖNSÖZ

İlişkisel veritabanları üzerinde sahiplik haklarının belirlenmesi ve bütünlük kontrollerinin gerçekleştirilmesinde mevcut damgalama yöntemleri kullanılmaktadır. Ancak bu yöntemlerin bazı eksiklikleri bulunmakta ve iyileştirilmesi gereken problemleri mevcuttur. Bu tez çalışmasında var olan ilişkisel veritabanı damgalama yöntemlerinin bazılarında bulunan problemler üzerinde iyileştirme çalışması yapılmıştır ve görüntü damgalanmasında kullanılan fakat var olan ilişkisel veritabanı damgalama yöntemlerinde kullanılmamış olan ayrık kosinüs dönüşümü (DCT) tabanlı bir şema tasarımı gerçekleştirilmiştir.

Yüksek lisans eğitimim boyunca kendisinden çok şey öğrendiğim, her konuda yol gösterici olan ve yanında çalışmaktan onur duyduğum değerli danışman hocam Sayın Yrd. Doç. Dr. Güzin ULUTAŞ' a, tüm aşamalarda varlığını hissettiğim ve her zaman en büyük destekçim olan eşim Elif ŞAHİN' e ve hoşgörülerinden ötürü sevgili aileme en içten dileklerimle teşekkür ederim.

Yasin ŞAHİN

Trabzon 2016

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum "İlişkisel Veritabanlarının Bütünlüğünün Sağlanması İçin Yeni Bir Sıfır Damgalama Şeması" başlıklı bu çalışmayı baştan sona kadar danışmanım Yrd. Doç. Dr. Güzin ULUTAŞ 'ın sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 10/06/2016

Yasin ŞAHİN

İÇİNDEKİLER

Sayfa No

Yüksek Lisans Tezi

ÖZET

İLİŞKİSEL VERİTABANLARININ BÜTÜNLÜĞÜNÜN SAĞLANMASI İÇİN YENİ BİR SIFIR DAMGALAMA ŞEMASI

Yasin ŞAHİN

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Yrd. Doç. Dr. Güzin ULUTAŞ
2016, 78 Sayfa

Bilgilere erişimin gittikçe daha kolay bir hal aldığı günümüzde, veritabanlarının bütünlüğünün korunması önemini artıran bir konu haline gelmiştir. Bu konunun çözümü için temel olarak bozulma tabanlı ve bozulmadan bağımsız damgalama olmak üzere farklı bakış açıları geliştirilmiştir.

Çalışması kapsamında bozulmadan bağımsız sıfır damgalama şeması olan iki farklı çalışmanın problemleri irdelenmiş ve ilişkisel veritabanları için yeni bir sıfır damgalama şeması önerilmiştir. Önerilen çalışmada kullanılan DCT tabanlı damgalama, görüntülerin damgalanması alanında kullanılmışken ilişkisel veritabanlarının damgalanmasında kullanılmamıştır. Bu yönüyle yapılan tez çalışması yeni bir veritabanı damgalama yöntemi sunmaktadır. Sunulan bu damgalama yöntemindeki amaç verilere yapılan saldırılar sonucu meydana gelen değişiklikler üzerinden veritabanının bütünlüğünün korunup korunmadığının tespit edilmesidir. Yöntem veritabanını gruplara ayırır ve bu gruplar üzerinden sütun sayısının yarısı kadar satırları içeren matrisler elde etmektedir. Elde edilen matrisler üzerinden DCT katsayıları hesaplanarak ilgili grup için bir damga bilgisi üretilmektedir. Bütün gruplar için DCT katsayılarının hesaplanmasının ardından grup damgaları birleştirilir ve veritabanı için bir damga oluşturulur. Deneysel sonuçlar zamansal, tespit aralığı ve damga boyutu olarak önerilen yöntemlerin iyileştirmeler sunduğunu göstermektedir.

Anahtar Kelimeler: Bozulmadan bağımsız damgalama, Kırılgan sıfır damgalama, İlişkisel veritabanı damgalama, Veritabanı bütünlük kontrolü, Ayrık kosinüs dönüşümü, Sayısal damgalama, Bozulma algılama

Master Thesis

SUMMARY

A NEW ZERO WATERMARKING SCHEMA TO ENSURE INTEGRITY OF RELATIONAL DATABASES

Yasin ŞAHİN

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Asst. Prof. Dr. Güzin ULUTAŞ
2016, 78 Pages

Protecting database integrity has become a matter of increasing importance nowadays as accessing information is getting easier. Basically, different perspectives as distortion based and distortion free watermarking has been developed for the solution of this issue.

In this study, two different distortion free zero watermarking schemes are investigated and consequently, a new zero watermarking scheme is proposed for relational databases. While DCT based watermarking proposed in this study, has been used in the field of image watermarking, it has not been used in watermarking of relational databases, which makes the proposed method a novel database watermarking method. The aim of the proposed watermarking method is to identify whether the integrity of the database in question maintained or not by identifying the changes as a result of attacks to the watermarked data. The method divides database into groups and with these groups the method obtains sub-groups which have the row count which is half of attribute count. With the obtained matrix, watermark information is generated for the related group by calculating DCT coefficients. After calculating DCT coefficients for all groups, group watermarks concatenate and a watermark is generated for the database. Experimental results show that methods proposed as temporal, detection range and watermark size present improvements.

Keywords: Distortion free watermarking, Fragile zero watermarking, Relational databases watermarking, Database integrity checking, Discrete cosine transform, Digital watermarking, Distortion detection.

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

SEMBOLLER DİZİNİ

BA	: Birincil Anahtar
GA	: Gizli Anahtar (Secret Key)
HE	: Histogram Genişletme Tekniği (Histogram Expansion)
LSB	: En anlamsız bit (Least significant bit)
MSB	: En anlamlı bit (Most sgnificant bit)
HSM	: Histogram Kaydırma Modülasyonu (Histogram Shifting Modulation)
LP	: Doğrusal Permutasyon (Linear Permutation)
R	: İlişkisel Veritabanı
R'	: Şüpheli Veritabanı İlişkisi
WAR	: Damga Doğruluk Oranı (Watermark Accuracy Rate)
WDR	: Damga Bozulma Oranı (Watermark Distortion Rate)
ω_R	: İlişkisel Veritabanı Damgası
ω_R'	: Şüpheli Veritabanı İlişkisi Damgası
MAC	: Mesaj Doğrulama Kodu (Message Authentication Code)
DEW	: Fark Genişlemeli Damgalama (Difference Expansion Watermarking)
PRSG	: Yalancı rasgele dizi üreteci (Pseudo-Random Sequence Generator)
GADEW	: Genetik Algoritma Tabanlı DEW (Genetic Algorithm based DEW)
PE	: Tahmin Genişlemeli (Prediction Expansion)
PAE	: Polar Açı Genişlemeli (Polar Angle Expansion)
MI	: Karşılıklı Bilgi (Mutual Information)
RRW	: Güçlü ve Geri Dönüştürülebilir (Robust and Reversible Watermarking)
DCT	: Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform)

1. GENEL BİLGİLER

1.1. Giriş

Günümüzde internet kullanımının ve buna paralel olarak bilgi paylaşımının hızlı şekilde artması, bilgilere ait sahiplik hakkının korunması problemini beraberinde getirmiştir. Sayısal veriler üzerindeki sahiplik bilgisini ispatlayabilmek ve doğrulayabilmek amacıyla, literatürde iki farklı teknik önerilmiştir: Sayısal İmzalama ve Sayısal Damgalama.

Sayısal imzalama yöntemi, kişilere ait gizli anahtar (GA) ve genel anahtar değerlerinin olmasını ve hak iddia edilecek olan veri üzerinde özel anahtarın kullanımı ile şifreleme algoritmalarının uygulanmasını gerektirmektedir. Sayısal damgalama yöntemi ise telif haklarının korunması için, özel oluşturulmuş bir verinin (damga olarak adlandırılan) önceden sayısal verinin içerisine özel tasarlanmış algoritmalar yardımı ile saklanmasını gerektirmektedir.

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 kullanılabilir.

Sayısal damgalama, şifreleme teknikleri ile karıştırılmamalıdır. Şifreleme, verinin matematiksel yöntemler ile anlamsızlaştırılması ve sonra veriyi kullanılır hale getirmek için şifrenin çözülmesi olarak tanımlanabilir. Şifreleme işleminde veri tamamen dönüştürülür ve veri üzerinde bütünlük kontrolü veya bozulma algılanması gibi harar verme işlemleri için kullanılmaz. Oysaki damgalama teknikleri veriyi dönüştürmeden alıcılar için kullanılabilir bir şekilde kodlar. Verinin sahibi veriyi damgalar ve sahiplik doğrulamasının gerektiği zaman kodlanmış veriyi çıkartır.

Sayısal damgalama alanında önerilen yöntemlerin bahsi geçen dört özelliğe sahip olması beklenmektedir: (i) Fark edilmezlik: Damga görünmez olmalı ve damga saklama süreci veri kullanılabilirliğini azaltmamalıdır. (ii) Dayanıklılık: Damga yok etme saldırılarına karşı güçlü olmalıdır. (iii) Güvenlik: Damga saklama süreci, güvenlik amaçlı bir GA kullanmalıdır. (iv) Körlük: Damga çıkarma süreci, orijinal verinin ve damga bilgisinin bilinmesini gerektirmemelidir.

Başlangıçta multimedya nesneleri için önerilen damgalama teknikleri üzerindeki çalışmalar, zamanla ilişkisel veritabanlarında telif hakkının korunması, bozulma algılama ve bütünlük kontrol gibi konuları da kapsayacak şekilde genişlemiştir. Mevcut olan damgalama yöntemlerinin çoğu verinin kullanılabilirliğini azaltmadan orijinal veri üzerinde bilinçli hatalar ve bozulmalar gerçekleştirmektedir. Bu bozulmalar ve hatalar verinin bütünlüğünü ve kalitesini etkilemeyecek şekilde veri sahibinin bütünlük kontrolü için veya sahiplik kanıtlanması için verinin içerisine eklediği bilgilerdir ve bu bilgiler ilişkisel veritabanı damgasını oluşturur.

Damgalama işlemi açısından multimedya nesneleri ile ilişkisel veritabanları yapısal özellikleri göz önünde bulundurulduğunda bazı farklılıklar göstermektedir. Bu farklılık Tablo 1'de görülmektedir.

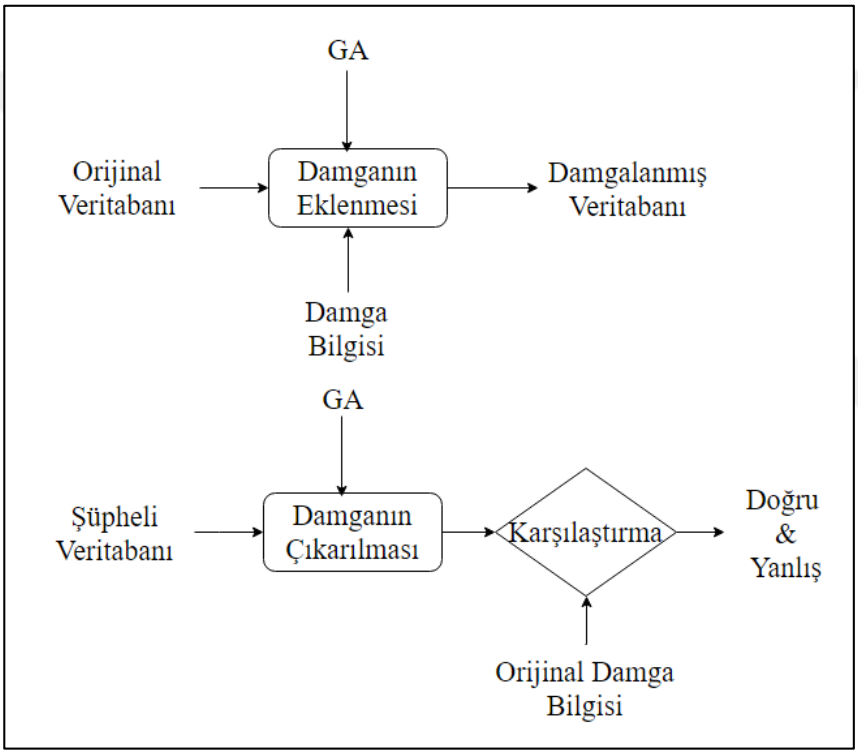
Tablo 1. Multimedya Nesnesi ile Veritabanı İlişkisi Karşılaştırılması

Multimedya Nesnesi	Veritabanı İlişkisi
Damganın saklanması için geniş bir uzay mevcuttur.	Her biri ayrı nesne belirten satırlar içerir. Damga bu ayrık nesnelerin üzerine yayılmalıdır.
Bir nesnenin çeşitli parçaları üzerinde uzaysal / zamansal yer değişimleri yapılmaz.	İlişkide bulunan satırlar bir küme oluşturur ve aralarında belirlenmiş bir sıralama yoktur.
Nesnenin kısımlarının kaldırılması çok kısıtlıdır veya algılanamayacak şekilde keyfi olarak değiştirilemez.	Kötü niyetli kişi kolaylıkla satır veya sütunları kaldırabilir veya diğer bir ilişkiden alınan satır veya sütunlarla temsil edilebilir.

Tablo 1'de verilen farklılıklar nedeniyle multimedya nesneleri için geliştirilmiş olan damgalama teknikleri ilişkisel verilerin damgalanması için doğrudan kullanılamamaktadır. İlişkisel verinin düzensiz olması, güncellemelere açık olması, damgayı saklamak için kullanılabilecek alanın sınırlı olması, kayıtların silme işlemlerine maruz kalabilmesi gibi durumlar, ilişkisel verinin damgalanması sürecinde önerilecek yöntemlerin, üstesinden gelmesini gerektiren sorunlar olarak ön plana çıkmaktadır.

Veri kayıtlarının internet üzerinden paylaşımının yaygınlaşmasıyla ilişkisel veritabanlarının bütünlüğünün korunması problemi ortaya çıkmıştır [2, 3]. Verilerin sahipleri,

kendi verilerini uzak erişime ve kullanıma açmakta ve dolayısıyla erişime açılan bu veriler veri hırsızlıklarına açık hale gelmektedir. Damgalama teknolojisinin sahiplik bilgisinin kanıtlanmasında yardımcı olmasına rağmen, geri dönüşü olmayan veri değişikliklerini de içermektedir ve damgalanmış son verinin orijinal veriden farklı olmasına neden olmaktadır [2-4]. Bu ortaya çıkan farklılıkların tespit edilmesi için daha önce multimedya nesneleri üzerinde kullanılmış olan damgalama yöntemlerine benzer yöntemler kullanılmıştır. Veritabanlarının damgalanması ve sonrasında damganın doğrulanması ile ilgili genel yapı Şekil 1'den görülmektedir.



Şekil 1. Basit Damgalama Tekniği

Sayısal damgalama yöntemleri, orijinal veritabanı üzerinde değişiklikler yapıp yapmamasına göre bozulma tabanlı ve bozulmadan bağımsız (sıfır damgalama) olarak iki sınıfa ayrılabilir. Bozulma tabanlı yöntemler sahiplik haklarının korunması için kullanılan damganın saklanması sürecinde veriyi değiştirir [4-7]. Bozulma tabanlı damgalama teknikleri, mevcut veri üzerinde az bozulma meydana getirirler ve kötü niyetli saldırılara karşı güçlüdürler. Diğer taraftan, sıfır damgalama yöntemlerinde damganın saklanma sürecinde yöntem orijinal veri üzerinde değişiklik yapmaz ve orijinal veri etkilenmemiş olarak kalır [8-

10]. Yapılan tez çalışmasında önerilen yöntem de bir sıfır damlama şemasıdır. Veritabanı üzerinde herhangi bir değişiklik yapmadan damganın oluşturulup kaydedilmesini sağlamaktadır.

Sıfır damgalama tekniklerinin yanında bazı araştırmacılar veritabanı ilişkilerinin bütünlüğünün kanıtlanması için kırılgan sıfır damgalama yöntemi de sunmuşturlar [10-13]. Bu yöntemlerin ana özelliği, orijinal veritabanından üretilen damgayı veritabanı içerisinde saklamak yerine, harici bir yerde saklayıp, veritabanı içeriğinde herhangi bir şüpheli değişiklikler ya da bozulmalar meydana geldiğinde kullanmaktır. Böylece meydana gelen değişiklikler saklanan damgayı etkileyemeyeceği için damga bilgisi de korunmuş olmaktadır.

Bütünlük kontrolü ve bozulma algılama için kullanılan damgalama teknikleri kırılgandır [8]. Bu tekniklerde saldırı girişimini algılamak amacıyla saklanan damga, saldırganlar tarafından yapılan bozma girişimleri nedeniyle oluşan veri değişimlerinin bir sonucu olarak kolaylıkla bozulabilir. Bu nedenle kırılgan teknikler sahiplik veya telif hakkının korunması için uygun değildir. Bu koşullarda zararlı olmayan güncellemelerin yanında kötü niyetli saldırılara karşı da esnek olan güçlü bir damgalama gerekmektedir [14]. Bir damgalama yöntemi ister kırılgan olsun ister güçlü, damgayı yok edebilecek kasıtlı veya kasıtsız saldırılara maruz kalabilir. İlerleyen bölümlerde ayrıntıları verilecek olan bu saldırılara örnek vermek gerekirse, veritabanına yeni satırların eklenmesi, veritabanından satırların silinmesi veya mevcut veriler üzerinde değişiklikler yapılması gibi saldırılardır. Zarar verme amacı olmadan yapılan güncellemeler işaretlenmiş satırlar üzerinde değişiklik yapabilirler veya satırları tamamen silebilirler. Bit veya yuvarlama saldırıları işaretlenmiş veriler üzerinde bit pozisyonlarını değiştirerek bilinçli olarak damgayı tahrip etmeye çalışır. Hile saldırısında, saldırgan veritabanının damgalanmış kopyalarına erişerek sahiplik iddiası için kendi damgasını veritabanına ekleyebilir. Alt küme ters çevirme saldırılarında, saldırgan satırların ya da sütunların yerlerini değiştirerek damgayı yok etmeyi amaçlar.

En iyi bilinen ve ilişkisel veritabanı üzerinde damgalama işleminin ihtiyacını ortaya koyan ilk çalışmalardan biri olan Agrawal vd. çalışmasından itibaren [2], ilişkisel veritabanlarının damgalanması alanına ilgi arttı ve bu alanda birçok çalışma yapıldı. Guo vd. 2007 yılında önerdikleri çalışma ile problemi tanımladı ve sayısal grup verilerinin bütünlüğünün doğrulanması için kırılgan damgalamanın önemini ortaya koydu [15]. 2008 yılına gelindiğinde ise ilişkisel veritabanı ile optimizasyon teknikleri arasındaki bağlantı Shehab vd. tarafından ortaya konuldu [16]. Bu metot damgalama işlemini, damganın saklanma süreci için zoraki optimizasyon problemi olarak, damganın çıkarılma sürecini de

çıkarma hatalarını minimize etmek için eşik değeri tabanlı bir tekniğe dayalı olarak düşünüldü. 2012 yılında yapılan çalışmada Farfoura vd. tarafından bozulmuş veriden orijinal verinin elde edilmesini sağlayan geri dönüştürülebilir bir yöntem önerildi [17]. Bu metot birincil anahtar (BA) değerine bağlı olup lineer dönüşüm saldırılarına karşı esnek değildir ve bundan dolayı işaretlenmiş satırlar silinirse seçilen satırlardan damga tekrar elde edilemeyebilir.

Yapılan tez çalışmasında sıfır damgalama yöntemlerini kullanan iki çalışma üzerinde durulmuştur [8, 9]. Bu çalışmalardan [9] çalışması [8] çalışmasını referans alarak bazı iyileştirmeler yapmıştır fakat yapılan iyileştirmeler beraberinde aşırı işlem yükü ve büyük boyutlu damga ortaya çıkmasına neden olmuştur. Ayrıca verilen yöntem işlem zamanı olarak da uzun zaman almaktadır. Görülen bu eksikliklerin giderilmesi için bazı iyileştirmeler sunulmakla birlikte ilişkisel veritabanları üzerinde bütünlük kontrolü için yeni bir sıfır damgalama şeması önerilmiştir. Yapılan çalışmada iki yöntem için de sonraki bölümlerde anlatılacağı gibi ayrı ayrı iyileştirmeler gerçekleştirmektedir. [8] çalışmasının sütun değiştirme saldırılarına karşı olan eksikliği giderilmiş ve damganın üretim zamanı olarak da iyileştirme sağlanmıştır. [9] çalışması için ise damga boyutu ve damganın üretilme zamanı açısından iyileştirmeler gerçekleştirilerek yeni bir sıfır damgalama şeması önerilmiştir.

1.2. İlişkisel Veritabanı

İlişkisel veritabanı Edgar Codd tarafından 1970 yılında ortaya atılmıştır. [18]. Önerildiği tarihten itibaren özellikle ticari uygulamalar için yoğun şekilde kullanılan bir model haline gelmiştir. Günümüzde Oracle, IBM DB2, Microsoft SQL Server gibi birçok ilişkisel veritabanı yönetim sistemleri mevcuttur.

İlişkisel veritabanında veriler tablolar veya ilişkiler halinde organize edilir. Tablolar satırlar ve sütunlardan oluşmaktadır. Tablolar arasında oluşturulan ilişkiler çok daha büyük boyutta verilerin saklanmasına ve bu verilere kolaylıkla erişilmesine olanak sağlar. İlişkisel veritabanları üzerinde ekle, silme, kayıt güncelleme gibi temel işlemleri gerçekleştirmek üzere SQL isimli bir dil geliştirilmiştir.

1.2.1. Veritabanının Tasarım Amacı, Tasarım Süreci ve Bütünlüğü

İyi tasarlanmış bir veritabanı:

- Veri fazlalıklarını elemine eder: Aynı veri parçaları bir defadan fazla saklanmaz. Verilerin tekrarlanması sadece kayıt alanı israfına neden olmaz aynı zamanda da veri tutarsızlığına neden olur.
- Veri bütünlüğünü ve doğruluğunu sağlamalıdır.

Veritabanları genellikle belirli bir uygulamaya göre düzenlenmiştir. Tasarımın işlem adımları şu şekilde düzenlenebilir:

1) Gereksinim analizi: Gereksinimlerin toplanması ve veritabanının kullanım amacının belirlenmesi
2) Tablo tasarımı ve birincil anahtar (BA) belirlenmesi: Veritabanında saklanacak verilerin toplanması ardından bu verilerin anlamlı şekilde tablolara ayrılması ve BA olarak isimlendirilen, her satırın tekilliğini belirleyen sütun seçilir. İlişkisel modelde bir tablo tekrarlı satırlar içermez çünkü bu durum tutarsızlığa neden olur. Tekillikten emin olmak için her tablo veya tablolar kümesi, tabloların her kaydını ayırt etmek üzere BA adı verilen bir sütun belirler. Çoğu ilişkisel veritabanı yönetim sistemleri hızlı arama ve verilere hızlı erişim için BA kullanır. BA aynı zamanda diğer tablolara referans vermek için de kullanılır. BA değeri ilgili tablo için tekil olmalıdır ve her zaman bir değeri olmalıdır. BA değeri değiştirilememelidir. Değiştirilmesi durumunda diğer tablolara olan referansların tümünü değiştirmek zorunda kalınabilir. BA için genellikle tamsayı değerler kullanılır.
3) Tablolar arası ilişkilerin oluşturulması: Bağımsız ve ilişkisiz tablolardan oluşan bir veritabanı küçük amaçlar için kullanılır. İlişkisel bir veritabanının güçlülüğü tablolar arasında tanımlanmış olan ilişkilerden meydana gelir. Bir ilişkisel veritabanı tasarımının en önemli yönü tablolar arasındaki ilişkilerin tespit edilmesidir.
4) Tasarımın normalizasyonu: Veritabanının yapısal olarak doğru olduğu ve en uygun Şekilde olduğunun kontrolü normalizasyon olarak adlandırılır.

İyi tasarlanmış bir veritabanı için ön önemli konulardan bir tanesi bütünlüğünün korunmasıdır. Çünkü veritabanının bütünlüğünün bilinçli yapılmış olan değişiklikler haricinde bir değişikliğe uğramış olması bütünlüğün bozulmuş olduğu anlamına gelmektedir.

Veri bütünlüğü, koruma ve tüm yaşam döngüsü boyunca verilerin doğruluğunu ve tutarlılığının güvencesi anlamına gelir. Kötü niyetli bir işlem veya bilinçli olarak yapılmayan değişiklikler veri bütünlüğü hatasını meydana getirmektedir.

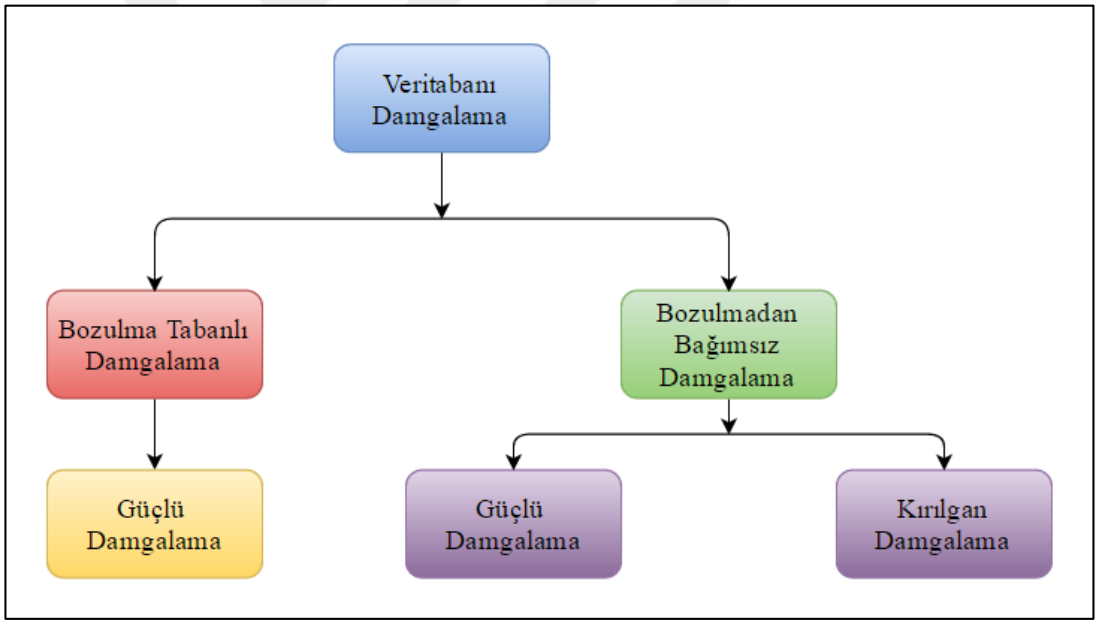
Tasarımın bütünlüğünün belirlenmesi için aşağıdaki kuralların kontrol edilmesi gerekmektedir:

1) Varlık bütünlüğü kuralı: BA boş değer olamaz. Aksi takdirde satırların benzersizliği belirlenemez. Çoğu veritabanı yönetim sistemleri bu kuralı uygular ve kontrol eder.
2) Referans bütünlük kuralı: Her yabancı anahtar değeri referans edildiği tablodaki BA anahtar değeri ile eşleşmek zorundadır. Yabancı anahtar olan satırı eklerken referans tablosunda ilişkili BA değeri mevcut olmalıdır. Referans tablosundaki anahtar değerleri değişirse (güncelleme, silme vb.) yabancı anahtar değerleri bu duruma göre düzenlenmelidir.
3) İş mantığı bütünlük kuralı: Yukarıdaki iki genel bütünlük kuralının yanında iş mantığı ile ilgili bütünlük de olmalıdır. Örneğin posta kodu 5 rakamdan oluşmalıdır, sipariş miktarı stoktaki miktara eşit veya ondan az olmalıdır gibi.

1.3. Veritabanı Damgalama

Yazılım, görüntü, video, ses ve metin gibi sayısal bilgilerin gizliliği ve korunması gibi veritabanlarında da aynı şekilde verilerin korunması ve bütünlüğün kontrol edilebilmesi bu bilgilerin sahipleri tarafından hep bir soru işareti olmuştur. Son yıllarda yüksek bir hızla gelişen internet kullanımı ile veritabanlarının tek bir yere bağlı kalmaksızın geniş bir kullanıcı erişimine açılmasına da neden olmuştur. Veritabanlarına erişen kullanıcı sayıları artması yapılabilecek saldırı olasılıklarını da arttırmaktadır. Burada en önemli konu veritabanının bütünlüğünün korunması yani veritabanı üzerinde kötü niyetli herhangi bir değişiklik olmaması ve sahiplik kontrollerinin yapılmasıdır. Bu iki konu için de birçok farklı yöntem [2] çalışmasından itibaren önerilmiştir. Daha önceleri yazılım, görüntü video gibi nesneler üzerinde kullanılan sayısal damgalama işlemleri temel alınarak ilişkisel veritabanları için de sayısal damgalama kullanılmaya başlanmıştır. Çalışmalar iki temel yapıya dayanmaktadır. Bu yapılardan biri veritabanından elde edilen damga bilginin yine veritabanı içinde saklanmasını öneren bozulma tabanlı yöntemler ve yine veritabanından elde edilen damga bilgisini veritabanı dışında saklayan bozulmadan bağımsız yöntemlerdir.

İki farklı yöntemin de amacı veritabanının, yapılacak olan kötü niyetli değişikliklere karşı korunmasıdır. Bu yöntemler kendi içlerinde alt sınıflara ayrılmaktadırlar. 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 veritabanının bütünlük kontrolü için kullanılmaktadır. Kırılgan damgalama şemasında, saklanan damga bilgisi değişikliklere karşı kırılgan olmalıdır böylece değişikliler algılanabilir ve yapılan değişikliklerin yerleri tespit edilebilir. Yapılan bu sınıflandırma Şekil 2'den de görülmektedir.



Şekil 2. Veritabanı damgalamanın sınıflandırılması

Bilgilerin korunması veya bütünlüğün kontrol edilebilmesi genelde verinin içerisine sayısal bir damga eklemeye dayalı olarak gerçekleştirilmiştir. Bu eklemeler damgalanacak veritabanı üzerinde küçük hatalar meydana getirmektedir. Bu bilinçli hatalar işaretler olarak adlandırılır ve bütün işaretler birlikte damgayı meydana getirir. İşaretler verilerin kullanılabilirliği üzerinde önemli bir etkiye sahip olmamalıdır ve kötü niyetli bir kullanıcı

veriyi daha az kullanılabilir hale getirmeden yok edemeyecek şekilde bu işaretler yerleştirilmelidir. Yani saldırgan yerleştirilen işaretlere veritabanının bütünlüğünü bozmadan erişememelidir. Aksi takdirde işaretleri değiştirip kendi damga bilgisini veritabanına ekleyebilir. Böylece kendi sahiplik iddiasını yapabilir.

1.3.1. Veritabanı Damgalama Uygulamaları

Damgalama veriler üzerinde farklı amaçlar için kullanılır. Daha önce de değinilen ilişkisel veritabanı damgalamanın amaçlarını biraz daha açık şekilde aşağıdaki gibi ifade edebiliriz.

Veritabanı damgalamanın ilk amacı telif hakkının korunması olduğu söylenebilir. En önemli damgalama uygulamalarından bir tanesidir. Eklenmiş olan bilgi diğer telif hakkı iddialarından korunmak için kullanılır. Bu uygulamada verinin asıl sahibi daha sonra bu veri üzerinde oluşabilecek bir sahiplik yani telif hakkı ile ilgili ortaya çıkabilecek herhangi bir sorunda eklemiş olduğu damgayı kullanarak kendi telif hakkını koruma altına almış olur.

Verilerin izinsiz olarak kopyalanması ve dağıtılması günümüzde veriler üzerinde büyük problemler meydana getirmektedir. Bu problem için damgalama kopya korunması amacıyla da kullanılmaktadır. Damgalama işlemi ile verinin sahibi her kopya için kendi damgasını oluşturup verinin içerisinde saklar ve bunların dışında bir kopya oluşturulması durumunda dağıtıcı tarafından kopya yayın belirlenebilecektir.

Veritabanı içeriğinin doğru ve değişmemiş olmasının kontrolü için de damgalama kullanılabilmektedir. Kötü niyetli kullanıcıların ve sistemlere saldırıların yoğun olduğu günümüzde verilerin bütünlüğünün korunması önemli bir konu haline gelmiştir. Verilerin internet ortamına açılması ile bu saldırılar veya veriler üzerindeki kötü niyetli değişiklikler sürekli artmaktadır. Bu saldırıların veya değişikliklerin algılanması için de damgalama yöntemleri kullanılabilir. Sayısal damgalamanın steganografiden farkında belirtildiği gibi, damga üretilirken mevcut veri kullanıldığından dolayı, veride yapılacak bir değişiklik o veri kullanılarak elde edilecek olan damgayı da etkileyecektir. Böylece veri üzerinde değişiklik yapılıp yapılmadığı tespit edilebilecektir.

Oluşturulan veritabanları belirlenen anlaşmalar ve lisans sözleşmeleri ile satılabilir. Fakat veri dağıtılırken de yine değiştirilmediğine ve taşınırken bir saldırıya uğramadığına emin olmak gerekmektedir. Ayrıca veri dağıtılırken yasal alıcıları dışında başkalarının da eline geçebilir. Bu durumda verilere istenmeyen kişiler tarafından ulaşılmış olur. Fakat

damgalama ile birlikte verinin bu istenmeyen kişiler tarafından elde edilmesi sonrasında verinin aslında o kişiye ait olmadığı kanıtlanabilir. Eklenen damga verinin yasal alıcıları hakkında bilgi vermek için kullanılır. Böylece orijinal verinin kopyalarının taşınması ve takip edilmesi açısından mümkün hale gelir.

1.4. Geri Dönüştürülebilir Veritabanı Damgalama

Günümüzde birçok bilgi veritabanları şeklinde bulunmaktadır. Bu yüzden veritabanı her nereye taşınırsa taşınsın ya da nereden erişilirse erişilsin sahiplik haklarının korunması çok önemli bir konudur. Damgalama teknolojisi veri üzerindeki sahteciliği belirleyerek sahiplik kanıtlanmasında yardımcı olmasıyla birlikte, geri dönüştürülemeyen ve damgalanan verinin orijinal veriden farklı olmasına neden olan bir olumsuz yönü de vardır. Sonuç olarak bozulmuş bir veride veri analizi ve karar verme olanaksız hale gelebilir. İlişkisel veritabanları üzerinde geri dönüştürülebilir damgalama yeni ve gelişmekte olan bir alandır. Veritabanının bütünlüğünün yanında sahiplik haklarının korunmasında da iyi bir çözüm sağlamaktadır. İlişkisel veri özel bir formata sahiptir ve ses, video, yazılım ve görüntü gibi sayısal verilerden farklıdır. Sahiplik koruması ve ilişkisel verinin geri dönüşümü için mevcut tekniklerin, damganın saklanacağı satırları (kayıt) içeren veritabanına, veritabanı ilişkisindeki kayıtların sıralanmasına ve ekleme, silme ve değiştirme gibi veri işlemlerinin veritabanı üzerinde olabileceğine dayalı olarak bazı kısıtlamaların olduğunu ortaya koyması gerekmektedir. Buna ek olarak veritabanında tutulan kayıtların veri tipine, uygun bant genişliğine veya damga saklama kapasitesine bağlı olarak değişkenlik gösterebilir ve bu değişkenlikler tekniği daha güçlü yapmak için değerlendirilebilir. Bununla birlikte damganın saklanması aşamasında bant genişliği tüketilmesi orijinal verinin kalitesinin korunması için çok geniş olmamalı. Veri değişiklikleri, damganın eklenmesinden sonra verinin bütünlüğünü ve verinin kalitesini bozmadığı sürece kabul edilebilirdir. Verinin sahibi de, veri kalitesinden ödün vermeden ne kadar değişimlere tolere olduğunu belirleyebilir.

Geri dönüştürülemeyen damgalama, sahiplik haklarının kanıtlanmasına yardımcı olur fakat damganın saklanması süreci genellikle veri üzerinde büyük ölçüde değişiklik yapar [2, 3]. Bu teknikler damganın saklanması ve saklanan damganın çıkarılması olarak iki kısım içermektedir. Birinci kısımda ya bit düzeyinde ya da veritabanının içerdiği kayıtlar üzerinde bir saklama algoritması kullanılabilir. Saklanan damganın çıkarılması sürecinde ise, veri sahibi orijinal damga bilgisi ile şüpheli veritabanına saklanmış olan damgayı sahiplik

haklarını kanıtlamak için doğrular. Diğer taraftan geri dönüştürülebilir damgalama, orijinal veriyi korur ve sahiplik koruması için damga bilgisi sunar [4-6, 19]. Geri dönüştürülebilir teknikler kodlama ve kodun çözülmesi aşamaları geri dönüştürülemeyen yöntemler ile aynıdır. Ek olarak geri dönüştürülebilir tekniklerde üçüncü bir veri kurtarma aşaması vardır.

Veritabanı damgalama daha önce de bahsedildiği gibi bozulma tabanlı ve bozulmadan bağımsız olmak üzere iki şekilde yapılmaktadır.

1.4.1. Geri Dönüştürülebilir Damgalama Uygulamaları

Geri dönüştürülebilir damgalama, damganın eklenmesi nedeniyle ortaya çıkan bozulmaların kontrolü için kullanılır ve sahiplik koruması ile veri kurtarma sağlar. Geri dönüştürülebilir damgalamanın uygulamaları şu şekildedir:

Veritabanları üzerinde en çok dikkat edilmesi gereken konulardan biri bütünlüğün korunmasıdır. Geri dönüştürülebilir damgalama teknikleri, orijinal veritabanı ilişkilerine zarar verecek kötü niyetli değişiklikleri gözlemleyerek meydana gelmiş olan bozulmaları algılama için kullanılır [10, 11, 13]. Veri sahibinin veri bütünlüğünü koruması gerektiği için internet üzerinden açık olan uygulamalarda verinin sonradan değiştirilip değiştirilmediği bilgisi gereklidir.

İlişkisel veritabanlarının damgalama uygulamalarında üretilen damga verinin içine kaydedilebileceği gibi orijinal veriden bağımsız olarak da saklanabilir. Literatürde bu tür uygulamalar sıfır damgalama olarak adlandırılmaktadır. [8, 20-23] çalışmalarında herhangi bir bozulmaya meydana getirmeden veri bütünlüğünü sağlamak için geri dönüştürülebilir damgalama teknikleri kullanılmıştır. Bu teknikler, sayısal hakların korurken orijinal veri üzerinde değişiklikleri önlemek için güçlü veya kırılgan olabilirler. Bu tekniklerin kırılganlığı, kötü niyetli saldırılara karşı savunmasız yapar ve bundan dolayı sahiplik koruması için uygun değildirler.

Damgalama farklı işlemler için kullanılabilmektedir. Bunlardan bir tanesi de deneme sürümü olarak yayınlanmış veritabanları yapılan damgalama uygulamasıdır. Veritabanı uygulamalarının deneme sürüm kontrolünün güvenliği için de geri dönüştürülebilir damgalama teknikleri kullanılabilir [4, 5]. Veritabanı uygulamasının orijinal tam sürümü daha sonra kullanıcının veritabanının lisansını satın alıp almadığına bağlı olarak kodlanmış veritabanından elde edilebilir. Örneğin bunun için kolaylık sağlayan veritabanları IBM DB2 [24] ve Oracle [25].

Damga bilgisinin verinin içerisinde saklanacağı damgalama uygulamalarında yüksek damga kapasitesi amaçlanmaktadır. Geri dönüştürülebilir damgalama teknikleri veri üzerinde bozulmaları en aza indirerek ve damga kapasitesini arttırarak maksimum damga güçlülüğünü sağlamak için mekanizma sunar [26, 27]. Yüksek damga kapasitesi ile birlikte mevcut veri üzerinde daha az bozulma meydana gelir. Yüksek kapasite damgalama özellikle askeri ve sağlık gibi kritik veritabanlarının olduğu sistemler için gereklidir.

Son olarak damgalanan veriden geri dönüştürülebilir damgalama uygulamalarında orijinal verinin tekrardan geri kazanılması gerekmektedir. Veri madenciliği ve diğer veri çıkarma işlemleri için geri dönüştürülebilir damgalama teknikleri herhangi bir değişim olmayan yüksek kalitede veri kümeleri sağlar [4, 6]. Elektronik medikal kayıt sistemleri gibi bazı ilişkisel veritabanı uygulamalarında veri madenciliği karar vermek için hasta bilgisinin çıkarılmasında kullanılır. Medikal verinin boyutları arttıkça kullanıcıların ve sağlık çalışanlarının bu sistemlere erişimleri daha çok oluyor. Bu da dolandırıcılık vakalarını ve veri gizliliğini endişe verici oranda etkilemektedir. Böyle paylaşılan ortamlarda sahiplik ve sayısal haklar, kötü niyetli kullanıcılar tarafından verilerin illegal kullanımından korumak için gereklidir.

1.4.2. Geri Dönüştürülebilir Damgalama Şartları

Geri dönüştürülebilir damgalama sahiplik korumasını ve veri kurtarmayı sağlar. İlişkisel veritabanları için kullanılan geri dönüştürülebilir damgalama teknikleri için bazı önemli şartları şöyle sıralayabiliriz:

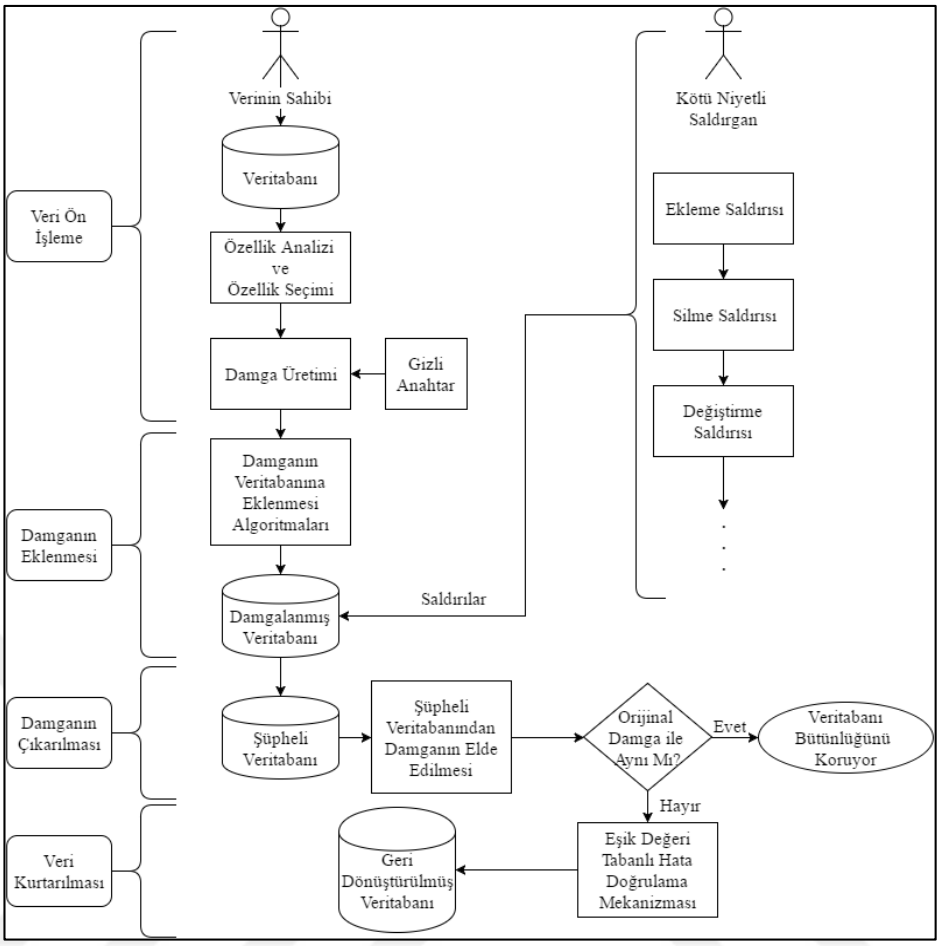
- Veri kurtarma: Damga ekleme ve çıkarma teknikleri, orijinal verinin damgalanmış veriden kurtarılabilecek olmasını sağlamalıdır.
- Veri bozulması minimum: Damganın eklenirken, orijinal veri üzerinde değişiklik yapan yöntemlerde yapılan değişiklikler ihmal edilebilir olmalıdır ve veri analizi veya madenciliği süreçleri bundan etkilenmemelidir.
- Körlük: Damga, işaretlenmemiş orijinal veri ve eklenmiş olan damgaya ihtiyaç duymadan algılanabilir olmalıdır. İlerleyen zamanlarda veritabanının dağıtılmış olan kopyaları gelişebilir ve büyüyebilir. Bu nedenle damganın algılanması için verinin orijinal kopyasının saklanması zordur.

- Algılanamaz: Eklenmiş damga geleneksel yöntemler kullanılarak fark edilemez veya algılanamaz olmalıdır.
- Güçlü: Damga, kötü niyetli saldırılara karşı dirençli olacak şekilde zekice eklenmelidir. Veritabanına yapılacak saldırılar saklanmış olan damga bilgisini bozabilir ve saldırgan bunun yerine kendi damgasını ekleyebilir.
- Damga kapasitesi: Damgalama tekniği uygun damga kapasitesini (veri kalitesinden ödün vermeden damga ekleme için uygun bant genişliği) kullanmalıdır.
- Kullanılabilirlik: İşaretleme işlemi sürecinde yapılan değişiklikler verinin kullanılabilirliğinde bir azaltma meydana getirmemelidir.
- Gizli parametrelerin güvenliği: Damgalama tekniğinin güvenliği temel olarak yalnızca veri sahibine özel olan gizli parametrelere bağımlıdır.
- Artımlı güncellenebilirlik: Her satırdaki damga ekleme diğer satırlardan bağımsız olmalıdır.
- Yanlış pozitif: İşaretlenmiş veriden yanlışlıkla geçersiz bir damganın tahmin edilme olasılığı ihmal edilebilir olmalıdır.
- Yanlış negatif: Damgalanmış veriden doğru damganın tahmininin başarısız olma olasılığı ihmal edilebilir olmalıdır.

1.4.3. İlişkisel Veritabanları Üzerinde Geri Dönüştürülebilir Damgalamanın Genel Çerçevesi

Geri dönüştürülebilir damgalama teknikleri dört aşamada genelleştirilebilir: (i) ön işleme; (ii) damga saklama; (iii) damga çıkarma; (iv) verinin bütünlüğü kontrolü. Genel çerçevenin kavramsallaştırılması tüm prosedürleri adım adım anlamaya olanak sağlar ve aynı zamanda ilişkisel veritabanları üzerinde sayısal ve sayısal olmayan damgalama teknikleri için daha fazla araştırmada referans olarak kullanılabilir. Belirtilen dört aşama ayrıca modüllerden oluşur ve ilişkisel veri için bozulma tabanlı ve bozulmadan bağımsız teknikleri göz önünde bulundurur. Aşağıda bu dört aşama ve bu aşamaların alt modüllerinin ayrıntıları verilmiştir.

İlişkisel veritabanlarında geri dönüştürülebilir damgalamanın genel akış diyagramı ise Şekil 2'de olduğu gibidir.



Şekil 3. İlişkisel veritabanları için geri dönüştürülebilir damgalama genel yapısı

1.4.3.1. Veri Ön İşleme

Verinin ön işleme sürecinde istenen hedefleri gerçekleştirmek için iki alt modül çalıştırılır:

1) Özellik analizi ve uygun özelliğin seçimi: Bazı tekniklerde istatistiksel ölçümler, veri çıkarma sürecinde önemlerine göre ilişkisel veritabanının özelliklerinin sıralanması için kullanılır. Özelliklerin sıralanması ortak bilgi, entropi ve bilgi kazancı ölçülerek gerçekleştirilir.
2) Damga üretimi: İlişkisel veritabanı için damganın üretilmesi sürecinde Genetik Algoritma, DCT vb. çok sayıda yöntem kullanılmaktadır. Bu yöntemlere değinecek olursak şöyle ifade edebiliriz:
   - Evrimsel algoritmalar yoluyla damga üretimi: Bu modül orijinal veriye sonradan eklenecek olan optimum damgayı üretir. Geri dönüştürülebilir damgalama bir

optimizasyon problemi olarak değerlendirilir. Bu sayede damga minimum bozulmalara neden olurken orijinal verinin kurtarılmasına da olanak sağlar. Genetik algoritmalar gibi evrimsel teknikler bu tür durumlarda kullanılabilecek uygun algoritmalardır [28]. Verimli bir şekilde mümkün çözüm uzayının aranmasıyla optimum çözüm gerektiren problemler için geliştirilir. İdeal çözümün aranmasında mevcut veriye bozulmalar eklemek için en uygun değeri belirleyen bir veri yapısı oluşturulur. Optimum değer veri kalitesini çok fazla etkilemez ve veride çok küçük bozulmalara neden olur.

- Yalancı rasgele dizi üreteci aracılığıyla damga üretimi (PRSG): Bir özellik içerisine damga metninin eklenmesinin amacı yalnızca veri kalitesini korumak değil bunun yanında geri dönüştürülebilir damgalamadan emin olmak. Verinin sahibi PRSG'nin girişi olan birincil damga değerini belirler [29].
- HASH teknikleri yardımıyla damga üretimi: MD5 ve SHA-1 gibi tek yönlü hash algoritmaları damga metninin oluşturulması için kullanılabilir. Bazı tekniklerde damga bilgisinin saklanacağı kayıtlar seçilirken BA özelliğinin hash değerinin modu alınarak hesaplanır ve damga bilgisi olarak kullanılır. MD5 ve SHA-1 elde edilen kayıt bilgisinin tersi olmadığından emin olmak için blok şifreleme bileşenlerinden oluşturulmuştur. Böylelikle hash fonksiyonu tabanlı damga metinleri saldırgan tarafından dönüştürülemez ve eklenen damganın algılanması daha zor hale getirecektir. Bu teknik veri kalitesi üzerinde sıfır etkiye sahiptir çünkü hash tabanlı damgalama gerçekte veri içerisine eklenmez böylece herhangi bir bozulma meydana getirmez.

1.4.3.2. Damganın Yerleştirilmesi

Damganın saklanması sürecinde elde edilmiş olan damga, ilerleyen bölümlerde verilecek yöntemler ile belirlenen özellik veya özelliklere saklanır. Bu özellikler satırlar sütunlar veya veriler içerisinde hangi bilgiler olduğu olabilir. Saklanacak olan bilgisinin güvenliği için kullanıcı tarafından belirlenmiş bir GA kullanılmaktadır. Bu GA bilgisi ne kadar güçlü olursa veritabanı damgasının güvenliği de o derece güçlü olacaktır. Aksi takdirde kolay tahmin edilebilecek bir GA, kötü niyetli bir kullanıcı tarafından damganın elde edilebilmesini de kolaylaştıracaktır. Damgayı tespit edebilen bir saldırıda, saldırıyı yapan kişi orijinal damga bilgisinin yerine kendi damga bilgisini ekleyebilecek ve kendi

sahiplik iddiasını yapabilecektir. Veritabanının damgalanmasında bazı parametreler kullanılmaktadır. Bu parametreler GA, veritabanında gruplama işlemi yapılmışsa grup sayısı, gruplardaki satır sayıları gibi parametreler kullanılmaktadır. Bu parametreler damganın saklanması ve sonrasındaki damganın çıkarılması aşamasında kullanılmak üzere hesaplanır.

1.4.3.3. Damganın Çıkarılması

İlişkisel veritabanına eklenmiş olan damga bilgisi bütünlük kontrolü, sahiplik kanıtlanması gibi işlemler için şüpheli veritabanından tekrar elde edilebilmesi gerekmektedir. Bunun için orijinal damganın elde edilmesinde uygulanan işlem adımları tekrardan uygulanır ve şüpheli veritabanından bir damga üretilir. Üretilen bu damga, damga bilgisinin doğrulanmasında, orijinal damga ile karşılaştırılır. Sahipliğin kanıtlanması için algılanan damga ile veritabanı içerisine eklenmiş olan orijinal damga aynı olmalıdır.

1.4.3.4. Verinin Geri Dönüştürülmesi

Orijinal veriden elde edilmiş olan damga ile şüpheli veritabanı üzerinden elde edilen damgaların karşılaştırılması ile veritabanının bütünlüğü hakkında karar verilmektedir. Eğer damgaların karşılaştırılması sonucunda bu damgalar birbirinin aynısı ise veritabanı bütünlüğünü korumaktadır denilmektedir. Aksi takdirde yani damgalar birbirinden farklı ise veritabanı bütünlüğünü kaybetmiştir denilmektedir.

1.4.4. Bozulma Tabanlı Geri Dönüştürülebilir Damgalama

Bozulma tabanlı teknikler damgayı ilişkisel veritabanına eklerken orijinal veri üzerinde değişiklikler yapmaktadır. Amaç genelde sahiplik kontrolünün sağlanmasıdır. Damga, sayısal olmayan verilerin karakterlerinde, tamsayı veya sayısal değerlerin bir kısmında veya kategorik veriler içerisinde saklanabilir. Damgalama tekniklerinin bu kategorisi, damganın eklenmesi sürecinde bazı farklılıklar içermektedirler. Literatürde önerilmiş olan bazı bozulma tabanlı damgalama teknikler için ilerleyen paragraflarda kısa özetler verilmektedir. İncelenecek olan bozulma tabanlı geri dönüştürülebilir damgalama

tekniklerinin hepsinde verinin formatı sayısaldır. Satır veya özelliklerin seçilmesinde fark genişlemeli damgalama (DEW) [4], fark genişlemeli damgalamada genetik algoritma ile bir amaç fonksiyonu tanımlayıp genetik algoritma tabanlı bir teknik (GADEW) [26], tahmin ve hash tabanlı tahmin hatası genişlemeli damgalama (PE-1) [5], veritabanı bilgisi ile tahmin hatası genişlemeli damgalama tekniği (PE-2) [6], sınıflandırma ve polar açı genişleme tabanlı (PAE-1) teknik [30], polar açı genişleme ve PRSG tabanlı (PAE-2) [31] ve karşılıklı bilgi (MI) ile güçlü ve geri dönüştürülebilir damgalama (RRW) tekniği [32] gibi damgalama teknikleri kullanılır. Damga kapasitesi eklenecek damga için ve veri bozulmalarını tolere etmek için kullanılan uygun bant genişliğidir. DEW, PE-1, PE-2, PAE-2 gibi bazı tekniklerde damga kapasitesi düşüktür ve mevcut veri üzerinde daha çok bozulma meydana getirir. GADEW, PAE-1 ve RRW gibi diğer bazı tekniklerde ise damgalama kapasitesi yüksektir ve az bozulma meydana getirir. Bu verilen teknikler için detaylı bilgiler aşağıdadır.

Fark genişlemeli damgalama teknikleri sayısal özellikler üzerinde bozulma meydana getiren aritmetik işlemler yaparlar [4, 33]. Bu aritmetik işlemler genellikle ilişkisel veritabanı özelliklerinin LSB değerlerini kapsar. [4] çalışmasında [2] çalışmasını genişleterek tam sayı değerler üzerinde fark genişlemesi kullanıldı ve geri dönüştürülebilir bir damgalama şeması önerildi. Bu şema sahiplik belirlenmesi ve belirli ikincil saldırılara karşı direnç sağladı. Damganın eklenmesi için her kayıttan iki özellik seçilir. Bozulmayı tolere edebilecek damga eklemeleri için mümkün olan LSB sayısının üst sınırı belirlenir. Bu sayı satırın BA değeri ile gizli damgalama anahtarının birleştirilmiş haline hash fonksiyonunun uygulanması ile elde edilir. Damga hash fonksiyonu ile elde edilen bit numarasına eklenir. $lsb(H(GA||s.BA))$ damganın ekleneceği biti belirler. Burada $s.BA$ değeri s. satırın BA değerini ifade eder. Bu yöntemde damga kapasitesi düşük olmasına rağmen saldırılara karşı doğrulama direnci %89-%98 arasındadır.

Genetik algoritma tabanlı DEW teknikleri (GADEW) ilişkisel veritabanlarında güçlü ve geri dönüştürülebilir bir çalışma olarak kullanıldı [26]. DEW ile seçilen satırdan tolere edilebilecek bozulma miktarına bağlı olarak sadece iki özellik ele alınır ve böylece düşük damga kapasitesi ve yüksek bozulma ortaya çıkmaktadır [4, 33]. Damga kapasitesini arttırmak ve bozulmayı azaltmak amacıyla genetik algoritma, damga eklemek için en uygun yerin seçimi amacıyla çok sayıda özelliği araştıracak bir optimizasyon olarak kullanıldı. Kayıt ve özellik bozulmalarının ölçülmesi amacıyla genetik algoritma için bir uygunluk fonksiyonu tanımlanır. Bu aynı zamanda saldırganın damgalanmış özellikleri doğru şekilde

tahmin edebilmesini ve belirlemesini engeller. Bu yöntemde damga ekleme süreci bir ön işleme modülü içermektedir. Ön işlem ile elde edilen değerler DEW şemasına iletilir.

Farfoura vd. çalışmalarında (PE-1) ise sahiplik koruması için geri dönüştürülebilir bir damgalama önermişlerdir [5]. Bu yöntemde işaretlenecek satır veya sütunlar tek yönlü kriptografik bir hash fonksiyonu ile belirlenmektedir. Damganın eklenme aşamasında ikili bir görüntüden elde edilen bit dizisi sayısal özelliklerin kısmi bölümlerine damga olarak eklenir.

[6] çalışmasında bir güçlü ve geri dönüştürülebilir bir veritabanı damgalama tekniği önerilmiştir. Bu yöntem [5] çalışmasını geliştirmeyi amaçlayarak sahiplik bilgisinin kanıtlanmasını amaçlamıştır. İlk olarak ilişkisel veritabanı bilgilerine dayalı bir damga üretilmektedir. Damganın saklanacağı kayıtlar tek yönlü bir MAC fonksiyonu ile belirlenir. Bu yöntemde damga bilgisi, veritabanı adı, versiyon numarası, satır sayısı, sütun sayısı ve sahip bilgilerini içermektedir.

[30] çalışmasında Li vd. sayısal veriler için PAE-1 ve kümelemeye dayalı geri dönüştürülebilir ve güçlü bir damgalama şeması önerdiler. Bu kümeleme işlemi veritabanını gruplara ayırmak olarak söylenebilir. Damganın eklenmesi için kullanılacak ikili damganın uzunluğuna bağlı olarak belirli grup sayısına göre kümelere ayrılır. Kümeleme metodu yüksek verimlilik ve yüksek kapasite sağlar. Ayrıca önerilen bu kümeleme yani veritabanının gruplara ayrılması işlemi, önerilen yöntemin de temelini oluşturur.

PAE yöntemini referans alan PAE-2 yöntemi de geri dönüştürülebilir veritabanı damgalama için önerilmiştir [31]. Veritabanına eklene damga için seçilen özellikler sözde rasgele sayı üreteci ile oluşturulur. Bu teknik alt küme değişikliklerine, alt küme eklemelerine karşı dayanıklıdır.

Diğer bir yöntem olan [32] yöntemi ilişkisel veri için veri dönüşümünü ve veri kalitesini garanti eden zeki bir geri dönüştürülebilir damgalama tekniği ve güçlü bir geri dönüşüm şeması önermektedir. Bahsi geçen yöntem, damga bilgisinin yerleştirilmesi aşamasında en iyi ve uygun aday özelliklerini seçme esnasında, karşılıklı bilgilerden faydalanmakta ve GADEW'in en uygun aday seçme kriterini iyileştirmeye çalışmaktadır.

Yukarıda belirtilen geri dönüştürülebilir bozulma tabanlı damgalama tekniklerinin çeşitli açılardan karşılaştırılması Tablo 2'de görülmektedir.

Tablo 2. İlişkisel veritabanları için bozulma tabanlı geri dönüştürülebilir damgalama tekniklerinin sınıflandırılması [34]

Teknik	Satır/ Özellik Seçimi	Damga Üretim	Anahtar Üretimi	Damga Boyutu	Yanlış Tahmin	Bozulma Toleransı
DEW	Fark genişlemesine dayalı iki özellik seçimi	Ortalama ve iki özelliğin farkına dayalı	GA ve BA'nın Hash değeri	Düşük	Yüksek	LSB üst sınırına bağlı
GADEW	Fonksiyon tanımlayarak genetik algoritma tabanlı	MAC tabanlı	GA ve BA'nın Hash değeri	Yüksek	Düşük	Tanımlanmış alt ve üst sınır
PE-1	Tahmin ve Hash tabanlı	mxn boyutlu Bitmap görüntü	Kriptografik hash fonksiyonu	Düşük	Yüksek	Limit tanımlanmamış
PE-2	MAC tabanlı	Veritabanı bilgisine dayalı	Hash	Yüksek	Düşük	Histogram çiftine dayalı
PAE-1	Sınıflandırma ve polar açı genişlemeli	İkili görüntüye dayalı	GA ve BA'nın Hash değeri	Yüksek	Düşük	Limitsiz
PAE-2	PRSG ve polar açı genişlemeli	Şifrelenmiş damga	Uygulanabilir değil	Düşük	Yüksek	Limitsiz
RRW	Karşılıklı bilgiye dayalı	Genetik algoritma fonksiyonu tabanlı	Genetik algoritma tabanlı tanımlanmış	Yüksek	Düşük	Genetik A. tabanlı optimum değer.

1.4.5. Bozulmadan Bağımsız Geri Dönüştürülebilir Damgalama

Bozulmadan bağımsız geri dönüştürülebilir damgalama teknikleri veri üzerinde değişiklik yapmazlar ve belirli bir özelliğe bağlı değildir. Bu sınıf altındaki teknikler genel olarak kırılgan tekniklerdir. Asıl amaç veritabanı üzerinde meydana gelmiş bozulmaların algılanmasıdır. Bu damgalama şeklinin arkasındaki ana fikir veritabanına herhangi bir bilgi saklamak yerine daha sonra veritabanının bozulup bozulmadığını kontrol edebilecek bir damganın veritabanından çıkarılmasıdır. Bu yöntemin avantajları şöyledir:

- Bu algoritma sayısal veya sayısal olmayan veritabanı olmasına bakılmaksızın tüm veritabanlarına uygulanabilir.
- Bu algoritma ekleme, silme ve değiştirme saldırılarına karşı etkin bir şekilde kullanılabilir.
- Bir veritabanı üzerinde bozulma olup olmadığının kontrolü için de bu algoritma kullanılabilir.

Bozulmadan bağımsız damgalama teknikleri veri üzerinde değişiklik yapmaz ve elde edilecek olan damga bilgisini orijinal veri içerisinde saklamayacağı için verinin herhangi bir özelliğine bağlı değildir. Bu teknikler genelde kırılgan damgalama teknikleri altında yer alır ve aynı zamanda damganın eklenmesi süreci mevcut veri üzerinde değişiklik yapmadığı için veritabanı içerisindeki bilginin bütünlüğünün bozulmadığını garanti eder. Bu tekniklerin amacı veritabanında herhangi bir bozulma meydana getirmeden bütünlük kontrolü ve bozulma algılaması sunmaktır.

Bozulmadan bağımsız geri dönüştürülebilir damgalama teknikleri kırılgan ve güçlü geri dönüştürülebilir damgalama olarak ikiye ayrılmaktadır. Bu tekniklerin veri kurtarma, bozulma oranı ve saldırılara karşı dayanıklılıklarını karşılaştıran Tablo 3 aşağıda gösterilmektedir.

Tablo 3. Geri dönüştürülebilir damgalama tekniklerinin karşılaştırılması

Teknik	Veri Kurtarma	Bozulma Oranı	Saldırılara Karşı Dayanıklılık
Bozulma tabanlı geri dönüştürülebilir damgalama teknikleri	Düşük	Düşük	Yüksek güçlü
Bozulma bağımsız kırılgan damgalama teknikleri	Tam	Sıfır	Sıfır güçlü
Bozulma bağımsız güçlü damgalama teknikleri	Tam	Sıfır	Düşük güçlü

1.4.5.1. Geri Dönüştürülebilir Kırılgan Damgalama Teknikleri

Bu bölümde, bozulmadan bağımsız geri dönüştürülebilir kırılgan damgalama teknikleri alanında literatürde ön plana çıkan çalışmaların incelenmesi gerçekleştirilmektedir. Bahsi geçen yöntemlerde damga kapasitesinin yüksek olması hedeflenirken, veritabanı üzerinde sıfır bozulma gerçekleştirilmektedir. Sıfır bozulmanın nedeni üretilen damga bilgisi ayrı bir dosya halinde saklanmasıdır.

İlişkisel veritabanları üzerindeki ilk geri dönüştürülebilir damgalama tekniğinde histogram genişletme yöntemi (HE) kullanılmıştır [7]. Zhang ve arkadaşları bu çalışmalarında veri üzerindeki kalıcı bozulmaların tolere edilemediği veritabanlarının doğrulanması üzerinde bozulmadan bağımsız bir öneri yapmışlardır. Damganın eklenmesi, ilk olarak sıfır olmayan bütün d_{ji} rakamlarının D_{ji} hücresinden çıkarılmasıyla yapılır.

Çıkarma işleminden sonra, $H(d_{ji})$ histogramı d_{ji}=1,2,3,4,…,9 şeklinde ifade edilen 1'den başlayıp 9'a kadar giden histogramlar şeklinde ifade edilir. HE ilk basamaklarda hatalara neden olur. 1'den 8'e mutlak genlikle birlikte tepe değeri olarak bir *P* belirteci bulmak için, histogramlardaki genlik değerleri 1 birim sağa kaydırılır. Bu işlem *P* belirtecini boşaltır ve damga *P* üzerinde modüle edilir. Bu teknikte yalnızca ilk basamak değeri d_{ji} değiştirilir. Sonra toplam ekstra bilgi $d_{ji} = 9$ gibi orijinal rakamlardan ayrıştırmak için kaydedilir. Damga bitleri ve yük bitleri HE kullanılarak hataların içerisine eklenir. Sonraki adım olarak işaretlenmiş özelliği elde etmek için ters Haar dalgacık dönüşümü hesaplanır. Damganın çıkarılma süreci de ekleme sürecine benzer şekilde çalışır. Algoritma gösteriyor ki eğer damga meydana gelen değişikliklerden etkilenmemişse çıkarılabilir ve orijinal veritabanı elde edilebilir. Ancak damga bilgisinde bir bozulma meydana gelmişse doğrulama işlemi gerçekleştirilemez.

Coatrieux ve arkadaşları veritabanının kategorik özellikleri için histogram kaydırma modülasyonu (HSM) tekniği tabanlı kayıpsız damgalama algoritması önerdiler [19]. Bu algoritma verimli biçimde medikal veritabanı içeriğinin bütünlüğünü korumaktadır. Gerçek damga ekleme işlemi veritabanı kayıtlarının düzenlenmesinden öncedir. Kayıtlar, kaydın BA ile gizli damgalama anahtarının birleşiminin (1) de verilen hash fonksiyonu kullanılarak belirli sayıda kısma ayrılır.

$$N_u = \text{hash}(<r_u \cdot P, K_w>) \bmod N \tag{1}$$

Burada *N* grup sayısı, r_u grup içerisindeki kayıtları ifade etmekte ve *P* ve *K* değerleri de BA ve GA temsil eder. Sonra damga düzenlenmiş kayıtlar (artan ya da azalan sırada) içerisine sıralı şekilde kaydedilir. Ekleme işlemi kategorik özellik verisi içerisine bit eklenmesini gerektirir. Eklenilecek bitler, veritabanının bütünlüğüne emin olmak için veritabanının kendi bir dijital imzası olarak önerilmiştir. Bu işlem kayıpsız ekleme modülasyonu olan HSM tekniği ile yapılmıştır. Bu teknik aynı zamanda verimli şekilde veritabanının bütünlüğü için geri dönüştürülebilir olmasını sağlar. Bu teknik veritabanının nasıl düzenlendiğinden bağımsızdır. Damganın çıkarılması sırasında veritabanı ekleme sürecinde olduğu gibi benzer şekilde tekrar düzenlenir ve sonra çıkarılan damga (sayısal imza) ve damgalanmamış verinin sayısal imzası tekrar hesaplanır ve çıkarılmış olan ile karşılaştırılır. Herhangi bir değişiklik benzeşmezlik halinde tespit edilebilir. Bu yöntemle

medikal veritabanı koruması için tek bir SHA1 hash fonksiyonu ile yeterli yüksek damga kapasitesi elde edilebileceği gösterilmiştir.

Li ve arkadaşları, hash değerleri GA ve BA parametreleyerek veritabanının kayıtlarını gruplamayı önermişlerdir [12]. Bu teknik doğrusal permutasyona (LP-1) bağlıdır. Bir grup seviyesi hash değeri hesaplanır ve bu gruptan çıkarılan damganın uzunluğu kayıt çiftlerine eşittir. Kayıtların sıralaması hash değerlerine ve eklenmiş damga bitine bağlıdır. [12] çalışmasında veri üzerindeki bozulmaları algılamak ve bozulmaların yerini bulmak için gruplama tabanlı bir kırılgan damgalama tekniği önerilmiştir. Bu çalışmadan hedeflenen problem verinin bütünlüğüne emin olmak için kırılgan damgalamada sayısal imzanın kullanım sınırını belirlemektir. Veride yapılan değişiklikleri kimliklendirme, sınıflandırma ve lokalize bu tekniklerin amacı değildir. Gruplar içerisindeki sayısal değerlere eklenen damga her grup için iki tür damga veri içerisine eklenir: (i) özellik damgası, (ii) kayıt damgası. BA ve GA kayıt ve özellik değerleri için hash değerinin hesaplanmasında kullanılır. Damganın algılanması için tekrardan GA ve grup sayısı belirlenir ve sırasıyla özellik ve kayıt damgalarını doğrulamak için iki vektör tanımlanır. Önerilen teknik, damgalanmış veride özellik değerlerinin değiştirilmesi, silinmesi, eklenmesi gibi değişiklerin lokalize edilmesini ve sınıflandırılmasını iddia eder.

[14] çalışmasında veritabanı üzerinde herhangi bir bozulmayı algılayan LP tabanlı (LP-2) damgalama tekniği önerilmiştir. Bu çalışmanın arkasındaki kilit nokta, verinin kategorik formatının herhangi bir bozulmayı tolere edemeyeceğidir. Bu nedenle, bu teknikle eklenen damga bunu dikkate alır ve veriyi bozulmadan saklar. Veriye W damgasını eklemek için A özelliklerine dayalı olarak gruplara ayrılır. Sonra anahtar tabanlı hash değeri her grubun her satırı için hesaplanır. Yapılan çalışmada asıl hedeflenen bozulmanın algılanması ve sahiplik korumasıdır.

[35] çalışmasında bozulmaya karşı korumak için fark genişlemesi ve destek vektör dönüşümü (SVR) tekniklerini kullanarak veritabanları için geri dönüştürülebilir bir damgalama tekniği önerdiler. Eklenecek damganın hangi özellik değerlerine ekleneceği SVR ile damganın eklenme süreci ise DEW ile gerçekleştirilmektedir. Önerilen bu çalışma veritabanı doğrulama ve bütünlük kontrolünde oldukça etkin bir yöntemdir.

1.4.5.2. Geri Dönüştürülebilir Güçlü Damgalama Teknikleri

Bu bölümde literatürde ön plana çıkan geri dönüştürülebilir güçlü damgama teknikleri incelenmektedir. Güçlü ve geri dönüştürülebilir damgalama tekniklerinde damga kapasitesi oldukça yüksek, sıfır bozulma ve düşük yanlış tahmin oranına sahiptir. Bu tekniklerin çoğunda veri formatı sayısal olmayan veriler üzerinde işlem yapılmaktadır. Satır veya sütunların seçimi hash tabanlıdır, damga üretimi görüntü tabanlıdır ve anahtar değeri BA ve GA yardımıyla üretilmektedir.

[20] çalışmasında verilen yöntem ile Zhang vd. yeni bir güçlü bir damgalama şeması önermişlerdir. Önerilen yöntem veritabanındaki sayısal ve metin değerleri için kullanılır. Bu yöntem ikili bir görüntünün, sayısal özelliklerin veya metin içerisindeki keyfi kelimelerin içerisine eklemeye dayanmaktadır. Ekleme işleminden önce önerilen şema her satırın BA değeri ve belirlenen bir GA'dan üretilen hash değeri ile veritabanı gruplara ayrılır. Veritabanının gruplara ayrılmasından sonra damga ve ikili görüntü 0-1 dizisi şeklinde kayıtların içine eklenir.

Franco vd. [22] çalışmasında güçlü ve geri dönüştürülebilir kayıpsız bir damgalama şeması önerilmiştir. Bu şemada veritabanının sayısal özelliklerinin modülasyonu kullanılmıştır. Yöntemde bütünlük kontrolü ve telif hakkının korunması garanti edilmektedir.

[23] çalışmasında Jian vd. ilişkisel veritabanları için sınıflandırma tabanlı güvenli ve güçlü bir kopya koruması yöntemi önerdiler. Damga bilgisinin oluşturulması için bir sıfır damgalama anahtarı üretilmiştir. Bu sıfır damgalama anahtarının üretilmesi için şu işlem adımları gerçekleştirilmektedir. Öncelikle (2) formülü kullanılarak bir kimlik kodu hesaplanmıştır.

$$Id=H(GA \,||\, r.P \,||\, A_i) \tag{2}$$

Burada GA kullanıcının gizli anahtar değeri, r.P ilgili satırın BA değeri A_i ifadesi ise aday sütun numarasını ifade etmektedir. Bu şekilde elde edilen hash değeri karakteristik çıkarma sürecinde r satırının konumunu belirler. Daha sonra idc=F(id, r.Ai) ile seçilen sütunun alt kümesi üzerinde sınıflandırma işlemi uygulanır. Böylece hangi satır ve sütunda bulunan değerlerin damga üretimi için kullanılacağı belirlenmiş olur.

1.4.6. Khan vd. Kırılgan Sıfır Damgalama Şeması

[8] çalışmasında Khan ve arkadaşları veritabanı ilişkisi üzerinde kötü niyetli değişiklikleri algılayan ve kategorize eden bir kırılgan sıfır damgalama şeması önermiştir. Diğerlerinin aksine bu yöntem veritabanında herhangi bir bozulma meydana getirmez. Aynı zamanda bu teknik veritabanında ne tür bir değişiklik yapıldığını tanımlar. Önerilen çalışmada rakamların, veri değerlerinin uzunluklarının ve veri değeri aralıklarının yerel özelliklerini kullanır. Bu üç karakteristik için alt damgalar oluşturulur ve bu alt damgaların birleştirilmesiyle veritabanının damgası elde edilir. Khan vd. önerdiği şema kırılgan damgalama sisteminin şu önemli özelliklerini içermektedir:

- Kırılganlık: Önerilen şema kırılgan olarak tasarlanmıştır. Yani kötü niyetli veri değişiklikleri meydana gelmesi durumunda eklenen damga algılanamaz.
- Fark edilemezlik: Sıfır damgalama yöntemine dayalı olan bu çalışmada mevcut veri üzerinde herhangi bir değişiklik yapılmadığı için eklenen damga fark edilemez veya görünmezdir.
- Anahtar Tabanlı Sistem: Damganın üretilmesi ve doğrulanması aşamasında GA tabanlı bir sistem kullanılmaktadır. Bu aynı zamanda kötü niyetli değişikliklerin algılanması ve tanımlanmasında gereklidir.
- Körlük: Kötü niyetli değişikliklerin algılanması ve tanımlanması için orijinal veritabanı ilişkisine ihtiyaç yoktur. Damga orijinal veritabanından elde edilip saklandığı için tekrardan damga oluşturulmayacaktır. Yalnızca üretilmiş ilk damga şüpheli veritabanından elde edilecek damga ile karşılaştırılacaktır. Bunun için de tekrardan orijinal veritabanına gerek duyulmamaktadır.

Çalışmada önerilen şemanın özelliklerini gördükten sonra Khan vd. yaptıkları çalışmanın damga üretim algoritmasının yalancı kod ifadesi şu şekilde verilmektedir.

```
(1)  Wd = rakam_alt_damgası()          // rakam alt damgasın
(2)  Wl = uzunluk_alt_damgası()        // uzunluk alt damgası
(3)  Wr = aralık_alt_damgası()         // aralık alt damgası
(4)  WR = Wd || Wl || Wr               // veritabanı damgası
(5)  EWR = Şifrele(WR, GA)             // damganın GA ile şifrelenmesi
(6)  WC = EWR ||sahip_id||tarih||saat// sertifikalandırma
(7)  Sertifikalandırılmış damgayı kaydet
```

Yukarıdaki yalancı kod ifadesini incelenecek olursa Satır 1-3'de veritabanı damgasının alt damgaları olan rakam, uzunluk ve aralık alt damgaları oluşturulur. Bu alt

damgalar oluşturulduktan sonra satır 4'de birleştirilerek veritabanı damgası elde edilir. Satır 5'de, elde edilen veritabanı damgası geri dönüştürülebilir bir şifreleme algoritması ile GA kullanılarak şifrelenir. Şifreleme işleminin ardından veritabanı sahibinin belirlediği bir id, tarih ve saat etiketleriyle oluşturulan şifrelenmiş damga sertifikalandırılır. Bu işlemlerin ardından elde edilen sertifikalandırılmış veritabanı damgası, veritabanından ayrı bir yerde saklanır.

1.4.6.1. Khan vd. Veritabanı İçin Damgasının Oluşturulması

Khan vd. çalışmalarında veritabanının oluşturulmasında 3 temel adım üzerinden hareket etmişlerdir:

İlk adım rakam alt damgasının oluşturulmasıdır. Rakam alt damgasında ilişkisel veritabanında bulunan tüm değerler incelenerek tüm rakamlar için frekans değerleri hesaplanmaktadır. Yani [0-9] aralığındaki bütün değerlerin veritabanında kaçar tane oldukları bulunur ve bulunan bu değer o rakam için frekans değeri olur. Bu alt damganın oluşturulmasını gösteren yalancı kod ifadesi şu şekildedir [8].

```
(1)  for each tuple r_i € R Do
(2)    for each attribute A_j € R Do
(3)      length = Len(r_i · A_j)  // i. satır, j. sütundaki değerin rakam
         sayısı
(4)      for i from 0 to length-1
(5)        d_i = Mid$( r_i · A_j, i, 1)  // i. satır, j. sütundaki değerin
           rakamları
(6)        digit_frequency[d_i]++ // belirlenen rakamın toplam sayısı
(7)        total_digit_count++ // tüm rakam sayısı
(8)      end for
(9)    end for
(10) end for
(11) for each i € digit Do
       // her rakam için bir yüzde değeri hesaplanması
(12)   rfd_i = (digit_frequency[i] / total_digit_count) * 100
(13)   ω_d = ω_d || rfd_i
(14) end for
(15) ω_d = ω_d || total_digit_count
(16) return ω_d
```

Yukarıda verilen rakam alt damgasının oluşturulması ile ilgili yalancı kod ifadesinin açıklaması şu şekildedir: Satır 1 ve Satır 2'de tüm veritabanı hücre değerleri inceleneceği döngüler mevcut. Ardından Satır 3'de ilgili hücre değerinde bulunan sayısal ifadenin uzunluğu alınır. Alınan bu uzunluk ile Satır 4'deki gibi döngü oluşturulur ve Satır 5'de bu

değerin rakamları elde edilir. Elde edilen rakam değerinin frekansı Satır 6' da olduğu gibi arttırılır. Satır 7'de ise ilgili rakamın tüm rakamlar içindeki frekans oranını hesaplamak için ilişkisel veritabanında bulunan tüm rakamların değerlerini hesaplayacak bir değişken kullanılır. Bu işlemler tüm ilişkisel veritabanında tekrarlandıktan sonra her rakam için bir frekans değeri belirlenmiş olur. Ardından Satır 12'de her rakam için bir frekans oranı hesaplanır ve Satır 13'de ilgili rakam için hesaplanan bu frekans oranları birleştirilerek rakam alt damgası üretilmiş olur. Son olarak da üretilen rakam alt damgasına toplam rakam sayısı Satır 14'deki gibi eklenir.

İlişkisel veritabanı damgasının oluşturulması aşamasında hesaplanan ikinci alt damga ise uzunluk alt damgasıdır. Bu alt damgada veritabanında bulunan değerlerin uzunluklarına göre sınıflandırma yapılır. Tüm değerler kontrol edilerek her uzunluğa ait kaç tane değer olduğu hesaplanır. Örneğin 1 uzunluklu kaç değerin veya 2 uzunluklu kaç değerin olduğunun bulunması gibi. Belirtilen uzunluk alt damgasına ait yalancı kod ifadesi ise aşağıda verilmiştir.

```
(1)  for each tuple r_i € R Do
(2)    for each attribute A_j € R Do
(3)      length = Len(r_i · A_j)  // i. satır, j. sütundaki değerin uzunluk
         değeri
(4)      length_frequency[length]++  //  belirlenen  uzunluğun  toplam
         sayısı
(5)      total_length_count++ // tüm uzunluk sayısı
(6)    end for
(7)  end for
(8)  for each i € length Do// her uzunluk değeri için bir yüzde değeri
     hesaplanması
(9)    rfl_i = (length_frequency[i] / total_length_count) * 100
(10)   ω_l = ω_l || rfl_i
(11) end for
(12) ω_l = ω_l || total_length_count
(13) return ω_l
```

Uzunluk alt damgası söylendiği gibi tüm veritabanı damgasının elde edilmesi için gerekli olan ikinci alt damgadır. Yukarıda verilen yalancı kod ifadesinin açıklaması şu şekildedir: Rakam alt damgasında olduğu gibi uzunluk alt damgasının hesaplanmasında da yine tüm değerler ele alınacağı için Satır 1 ve Satır 2 döngüleri oluşturmaktadır. Ardından Satır 3'de ilgili hücrede bulunan değerin uzunluğu belirlenir ve Satır 4'de bu uzunluk değerinin frekansı 1 arttırılır. Satır 5'de rakam alt damgasının hesaplanmasında olduğu gibi hangi uzunluk değerinin tüm uzunluk değerleri içinde hangi orana sahip olduğunun hesaplanabilmesi için kullanılan bir değişken 1 arttırılır. Bu şekilde tüm veritabanı değerleri kontrol edildikten sonra belirlenen her uzunluk değerinin tüm uzunluk değerleri içerisindeki

yüzdesi hesaplanır. Satır 9'da ilgili uzunluk değerinin yüzdesi hesaplanırken Satır 10'da hesaplanan yüzde ile uzunluk damgası belirlenir. Tüm uzunluk değerlerinin yüzdesi hesaplandıktan sonra toplam uzunluk sayısı Satır 12'de olduğu gibi uzunluk alt damgasına eklenir ve böylece uzunluk alt damgasının hesaplanması gerçekleştirilmiş olur.

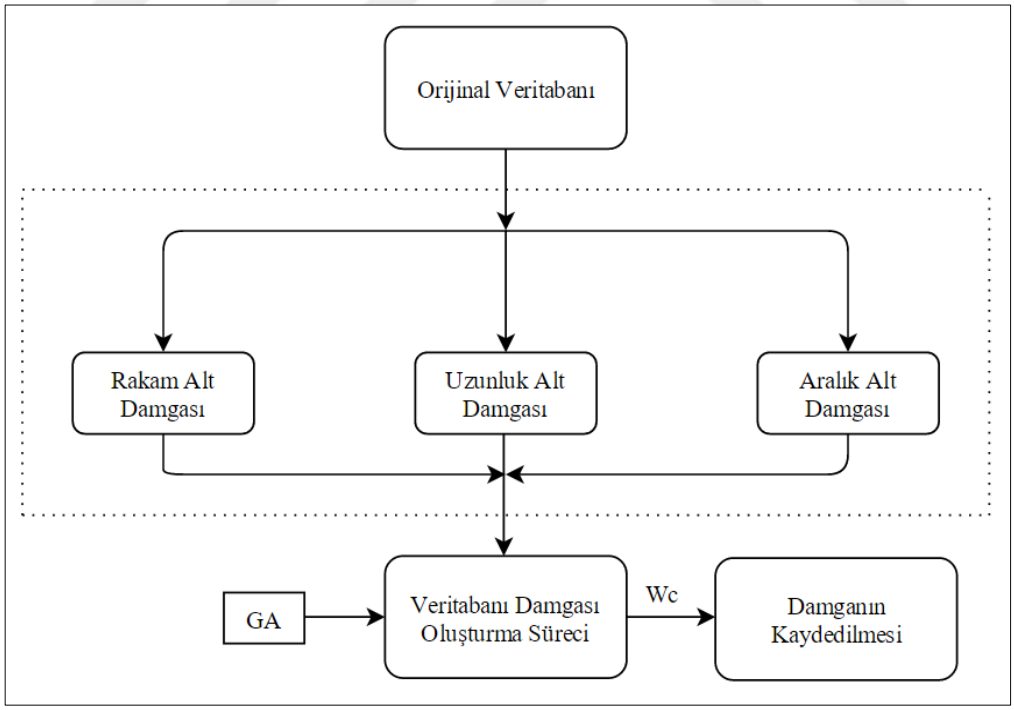
Veritabanı damgasının oluşturulması için gerekli olan son alt damga aralık alt damgasıdır. Aralık alt damgasında kullanıcı tarafından belirlenmiş olan aralık değerleri için uzunluk alt damgasının hesaplanmasına benzer bir yapı kullanılır. Belirlenmiş olan aralık ilerisine kaç tane değerin düştüğü ile o aralık değeri için bir frekans değeri belirlenir. Örnek vermek gerekirse belirlediğimiz aralıklar [0-100], [101,1000] ve [1001-10000] olsun. Tüm veritabanı değerleri incelenerek bu belirlenmiş olan aralık değerlerinin içerisine kaç tane değerin düştüğü hesaplanır. İlişkisel veritabanı damgasının oluşturulması için gereken son alt damga olan aralık alt damgasının oluşturulmasına ait yalancı kod ifadesi de aşağıdaki gibidir.

```
(1)  range = {0-100, 101-1000, 1001-10000, 10001-100000, 100001-1000000}
(2)  // alt damganın oluşturulması için belirlenen aralıklar
(3)  for each tuple r_i € R Do
(4)    for each attribute A_j € R Do
(5)      x = r_i · A_j  // i. satır, j. sütundaki sayısal değer
(6)      select case x
(7)        x in range 0: range_frequency[0]++
(8)        x in range 1: range_frequency[1]++
(9)        x in range 2: range_frequency[2]++
(10)       x in range 3: range_frequency[3]++
(11)       x in range 4: range_frequency[4]++
(12)     end select
(13)     total_range_count++
(14)   end for
(15) end for
(16) for each i € range Do // her aralık değeri için bir yüzde değeri
     hesaplanması
(17)   rfr_i = (range_frequency[i] / total_range_count) * 100
(18)   ω_r = ω_r || rfr_i
(19) end for
(20) ω_r = ω_r || total_range_count
(21) return ω_r
```

Rakam ve uzunluk alanlarında olduğu gibi aralık alt damgasının yalancı kod ifadesinin açıklaması şu şekildedir. İlk olarak belirlediğimiz aralıkları Satır 1'de olduğu gibi tanımlıyoruz. Ardından veritabanındaki tüm değerleri ele alınacağı için diğer alt damgaların hesaplanmasında olduğu gibi Satır 3 ve Satır 4'de döngüler mevcuttur. Satır 5'de ilgili hücredeki değer alınır. Alınan değerin Satır 6-Satır 12 arasında hangi aralığa düştüğü belirlenir. Satır 13'de diğer alt damgalarda olduğu gibi aralık alt damgasında da yine her

aralığın tüm aralıklar içerisinde yüzde kaç oranına sahip olduğunun hesaplanması için bir değişken belirlenir ve bu değişken her döngüye girildiğinde 1 arttırılır. Aralık değerlerinin frekansları hesaplandıktan sonra Satır 17'de her aralık değeri için tüm aralıklar içerisindeki yüzdelik oranı hesaplanır ve Satır 18'de bu oranlar birleştirilerek damga elde edilir. Son olarak da toplam aralık sayısı oranlarla oluşturulan damgaya eklenerek rakam alt damgası elde edilmiş olur.

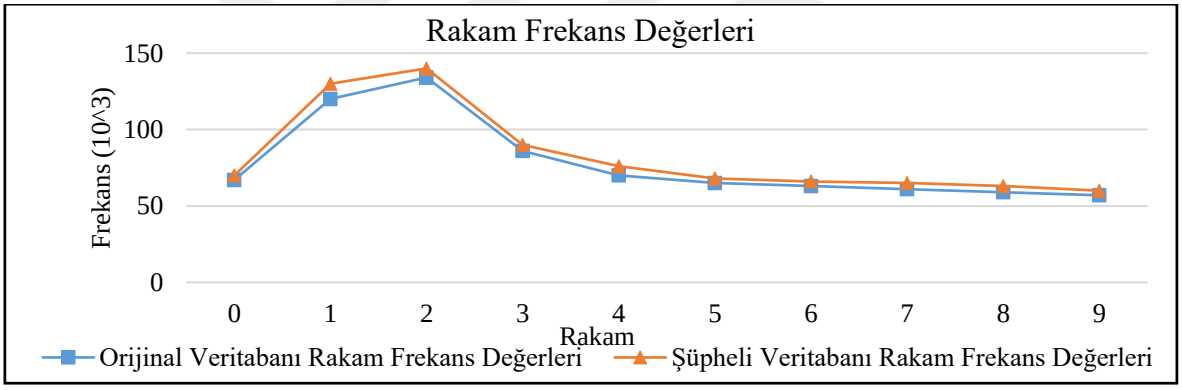
Rakam, uzunluk ve aralık alt damgalarının elde edilmesinin ardından oluşturulan damga, damga üretim algoritmasında da görüldüğü gibi geri dönüştürülebilir bir şifreleme algoritması ile şifrelenir ve sertifikalandırılarak kaydedilir. Damgalama işleminin tamamlanmasının ardından herhangi bir zamanda veritabanı bütünlüğünün kontrolü kaydedilen damga kullanılarak orijinal veritabanına ihtiyaç duyulmadan bulunabilir. Orijinal veritabanı üzerinden rakam, uzunluk ve aralık alt damgaları oluşturularak veritabanı damga bilgisinin üretilmesini gösteren akış diyagramı Şekil 4'de verilmiştir. Şekil 4'de görülen Wc değeri alt damgaların birleştirilmesi sonrasında damganın, zaman ve kullanıcı bilgisi ile sertifikalandırılmış ifadesini göstermektedir.



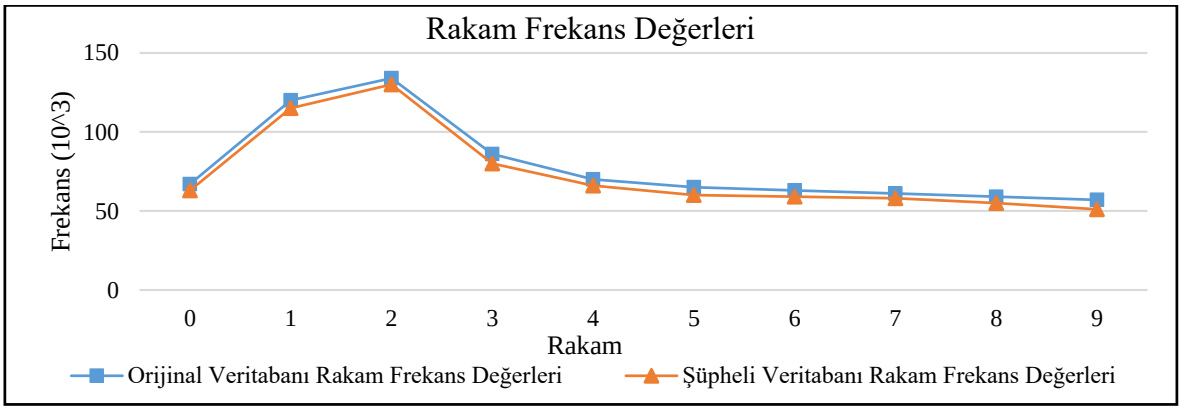
Şekil 4. Khan vd. damga üretim şeması

1.4.6.2. Ekleme, Silme ve Güncelleme Saldırıları Karşısında Alt Damgaların Frekans Değişimleri

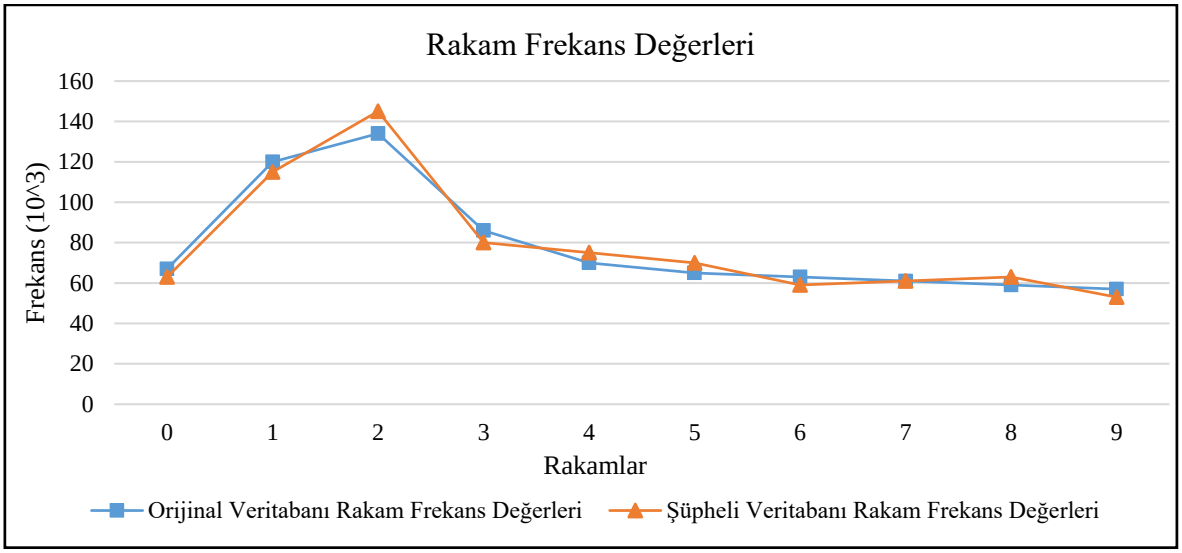
Khan ve arkadaşları çalışmalarında ilişkisel veritabanının damgasını oluşturmak için üretmiş olduğu rakam, uzunluk ve aralık alt damgaları ekleme, silme ve güncelleme saldırıları karşısında frekans değerleri değişeceğinden orijinal veritabanından elde edilen alt damgalar ile farklı olacaktır. Şekil 5, Şekil 6 ve Şekil 7'de ekleme, silme ve güncelleme saldırıları karşısında rakam frekansındaki değişimlerini gösteren grafikler görülmektedir. Bu şekiller [8] çalışmasında verilen algoritmalar kullanılarak kodlanması yapılmıştır. Ekleme saldırısında frekans değerleri artmakta, silme saldırında azalmaktadır. Güncelleme saldırılarında ise bazı rakamların frekans değerleri artış gösterirken bazı rakamların frekans değerleri azalma göstermektedir. Elde edilen grafikler çalışmada verilen algoritmanın kodlanmasıyla tarafımızdan gerçekleştirilmiştir.



Şekil 5. Khan vd. ekleme saldırısı sonucu rakam frekansındaki değişim



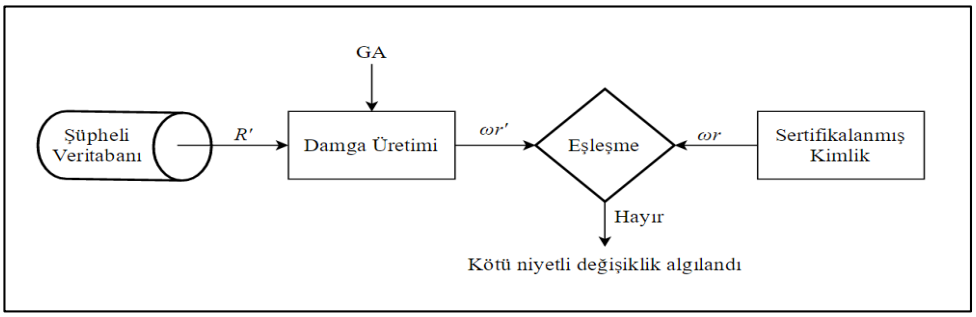
Şekil 6. Khan vd. silme saldırısı sonucu rakam frekansındaki değişim



Şekil 7. Khan vd. güncelleme saldırısı sonucu rakam frekansındaki değişim

1.4.6.3. Bozulma Oranının Tespiti

Khan vd. önerdikleri yöntemde öncelikle ilişkisel veritabanı üzerinde hangi oranda bir bozulma olmuştur bunun kontrolü yapılmaktadır. Burada damga doğruluk oranı (WAR) ve damga bozulma oranı (WDR) olarak ifade edilmektedir. Bu kontrolde orijinal veriden elde edilen damga ile şüpheli veriden elde edilen damga karakter bazında karşılaştırılmaktadır. Karşılaştırma sonucunda aynı olan ve olmayan karakterler belirlenerek veri üzerinde saldırı sonucu oluşan bozulma oranını vermektedir. Bunun yanında rakam, uzunluk ve aralık damgalarının tekrar kontrol edilmesiyle yapılan saldırı ile ilgili bir karakteristik belirlemektedir. Şüpheli bir saldırının algılanması Şekil 8'de olduğu gibidir.



Şekil 8. Khan vd. şüpheli değişikliklerin algılanması diyagramı [8].

Bozulma oranının hesaplanmasına ait yalancı kod ifadesi ise aşağıdaki gibidir.

```
(1)  Önceden verilen algoritmalar kullanılarak şipgeli veritabanından
     ω_R′ üret.
(2)  ω_c üzerinden şifrelenmiş damgayı Eω_R elde et
(3)  ω_R = Decrypt(Eω_R, GA)
(4)  for i = 1 to lenght(ω_R)
(5)     if ω_R[i] = ω_R′[i] then
(6)        match_count = match_count + 1 // eşleşme sayısı
(7)     end if
(8)  total_count = total_count + 1 // toplam karakter sayısı
(9)  end for
(10) WAR = match_count / total_count * 100
(11) WDR = 1 – WAR
(12) if WDR ≠ 0 then
(13)    veritabanı ilişkisi R′ WDR oranında bozulmuştur.
(14) end if
```

Verilen yalancı kod ifadesinde önce orijinal veritabanından elde edilip sertifikalandırılarak saklanmış olan damga üzerinden geriye doğru işlemler gerçekleştirilir. İlk yapılan işlem Satır 1'de şüpheli veritabanından daha önce verilmiş olan algoritmalar kullanılarak şüpheli veritabanı damgasının elde edilmesi Satır 2'de orijinal veritabanından elde edilmiş ve saklanmış olan sertifikalandırılmış veritabanı damgasından sertifika bilgileri çıkartılarak şifrelenmiş veritabanının damgasının elde edilmesidir. Satır 3'de ise şifrelenmiş veritabanı damgasının şifresi çözülerek veritabanının damgası elde edilir ve orijinal veritabanı damgası da elde edilmiş olur. Bunun ardından şüpheli veritabanından üretilmiş damga ile orijinal veritabanından elde edilip saklanan damgalar Satır 4-Satır9' da olduğu gibi karakter bazında karşılaştırılır. Satır 10'da tüm veritabanı için bir doğruluk oranı belirlenir. Belirlenen bu doğruluk oranı kullanılarak Satır 11'de veritabanı için bir bozulma oranı hesaplanır. Satır 12'de bozulma oranı kontrol edilir ve eğer bu oran 1'den farklı ise veritabanı ilişkisi şu oranda bozulmuştur şeklinde Satır 13'de olduğu gibi mesaj verilir.

Khan vd. tarafından yapılan çalışmanın saldırılara karşı tespit oranları Tablo 4'de verilmiştir. Tabloda verilmiş olan WAR damganın doğruluk oranını belirtirken WDR ifadesi ise veritabanının bozulma oranını ifade etmektedir.

Tablo 4. Khan vd. ekleme, silme ve güncelleme saldırılarına karşı bozulma oranı tespiti [8].

Ekleme, Silme, Güncelleme Oranı	WAR			WDR			Bozulma Algılama
	Ekleme	Silme	Güncelleme	Ekleme	Silme	Güncelleme	
%10	18.14	24.32	20.42	81.86	75.68	79.58	Yüksek
%30	18.56	17.88	19.89	81.44	82.12	80.11	Yüksek
%50	20.41	20.95	19.89	79.59	79.05	80.11	Yüksek
%70	16.67	13.08	18.94	83.33	86.92	81.06	Yüksek
%90	16.3	14.14	14.13	83.68	85.86	85.87	Yüksek

1.4.7. Camara vd. Kırılgan Sıfır Damgalama Şeması

Camara vd. çalışmalarında Khan vd. referans alarak iyileştirme yapmaya çalışmıştır [9]. Önerilen teknikte veritabanı gruplara ayrılarak kare matrisler düzenlenmiştir. Bu kare matrisler üzerinden determinant değerleri alınarak damga oluşturulmaya çalışılmıştır. Böylece Khan vd. önerdikleri yöntemde tüm veritabanı üzerinden bozulma oranı belirlenirken Camara vd. grup seviyesinde saldırı tespiti yapılması amaçlanmıştır. Bu yöntem 4 ana adımdan oluşmaktadır:

- Veri kümesinin gruplara ayrılması: İlişkisel veritabanı *R*, α sayıda satır ve γ sayıda sütun içermektedir. Bu ilişkisel veritabanının υ sayıda gruba ayrılması şu şekilde olmaktadır: Öncelikle $\upsilon = \lceil \alpha / \gamma \rceil$ ile elde edilecek grup sayısı belirlenir. Bunun ardından her satın hangi gruba dâhil edeceğini belirlemek için GA ve ilgili satırın BA değeri mesaj doğrulama kod fonksiyonu (MAC) (3) denkleminde görüldüğü gibidir:

$$j = \text{hash}(GA \,||\, \alpha_i \cdot BA \,||\, GA) \bmod \upsilon \quad (3)$$

Bu denklem ile tüm satırlar ilgili gruplara dâhil edilmiş olur. Eğer $\alpha \bmod \gamma \neq 0$ oluyorsa, kare matris elde etmek amacıyla gruptaki diğer herhangi bir satırla aynı olmayacak şekilde ilk grubun ilk satırından itibaren kontrol edilerek satır eklemesi gerçekleştirilir. Eklenen bu satırlar damga üretim işleminden sonra silinmelidirler. Veritabanının gruplara ayrılması ile ilgili yalancı kod ifadesi aşağıdaki gibi verilmektedir:

```
Girişler: ilişkisel veritabanı: R, Grup sayısı
υ  = ⌈ α / γ ⌉, BA
Çıkışlar: herbirinin uzunluğu γ olan gruplar
begin
   for i=1 to α do
      h_i^r=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
```

- Grup damgalarının hesaplanması: Gruplar belirlendikten sonra her grup için damga hesaplama işlemi gerçekleştirilir. Her grup sütun sayısı kadar satır içermektedir ve her grup bir kare matris oluşturmaktadır. Bütün kare matrisler yani gruplar için determinant ve diyagonal minörler hesaplanır. İlgili her grup için hesaplanan determinant ve minör değerleri birleştirilerek o grubun damgasını oluşturur. Matrisin determinantının hesaplanması (4) denkleminde olduğu gibidir:

$$|A| = \sum_{i=1}^{n} (-1)^{i+1} a_{i1} \det(A_i) \qquad (4)$$

Grup damgasının hesaplanması ile ilgili yalancı kod ifadesi ise şu şekildedir:

```
Giriş: Grup G_j
Çıkış: Grup damgası W_j
Begin
//birincil anahtara göre artan sırada grubun satırları
//sıralanır
   for j=1 to υ do
      D_j = Det(G_j)
      M_i^j (0≤i≤γ) = Min(A_j)
      W_j = D_j || M_i^j (0≤i≤γ)  //j. grup damgasını hesapla
   end for
   return W_j
end
```

Burada Det(G_j) j. grubun determinantını, Min(A_j) ise j. diyagonalın minörünü göstermektedir.

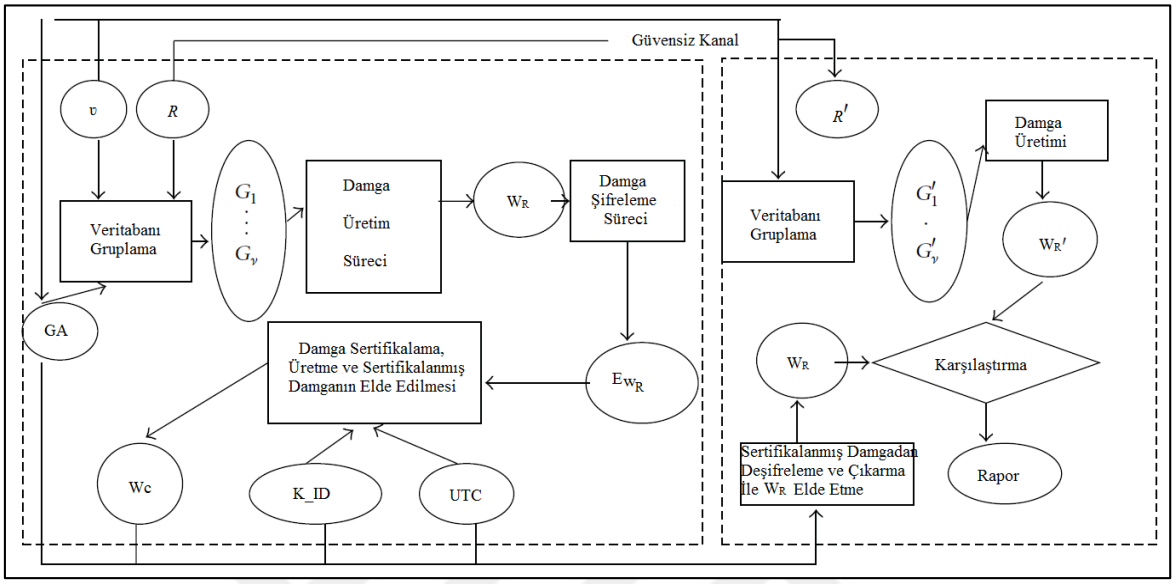
- Damganın hesaplanması ve kaydedilmesi: Grupların damgaları oluşturulduktan sonra ilk işlem olarak elde edilen damgalar birleştirilir. Daha sonra birleştirilen damgalar geri dönüştürülebilir bir şifreleme algoritması ile şifrelenir ve ardından zaman damgası ve kullanıcı bilgisi ile sertifikalandırılır. Böylece veritabanına ait damga elde edilmiş ve kaydedilmiş olur. Damganın hesaplanmasını gösteren yalancı kod aşağıda verilmiştir:

```
Giriş: R, GA, υ
Çıkış: Sertifikalandırılmış damga
begin
//Veritabanının gruplara ayrılması, Grup damgalarının
elde edilmesi
W_R = W_1||W_2||W_3||…||W_υ
E_WR= Şifrele(W_R||GA)
W_C = E_WR|| K_ID || UTC
Return W_C
end
```

- Bütünlük Kontrolü: Veritabanı için damganın elde edilmesinden sonra şüpheli veritabanından aynı yöntemlerle elde edilen damga karşılaştırılarak veritabanının bütünlüğü kontrol edilir. Bütünlük kontrolünde değişimler grup bazında kontrol edildiği için herhangi bir grupta bozulma meydana gelmişse bu belirlenip satır aralığı olarak bozulma tespit edilmiş olacaktır. Bütünlük kontrolü ve bozulma meydana gelmiş olan grubun tespiti ile ilgili yalancı kod ifadesi şu şekildedir:

```
Giriş: Şüpheli veritabanı R', GA, υ, W_C
Çıkış: Doğrulama raporu
begin
  W_C'den E_WC'yi elde et
  W_R = Decrypt(E_WC || GA)
  //Veritabanının gruplara ayrılması
  For j=1 to υ' do
    W'_j = D'_j || M^j_i (0≤i≤Y)    // j. Şüpheli grup damgası
    if W_j = W'_j  // grup damgalarının karşılaştırılması
      return bozulmamış grup
    else
      return bozulmuş grup
    end if
  end for
end
```

Camara vd. önerdikleri çalışmanın temel adımlarını içeren şema yapısı Şekil 9'de görülmektedir.



Şekil 9. Camara vd. kırılgan veritabanı damgalama şeması [9].

1.4.8. Khan vd. ile Camara vd. Çalışmalarının Karşılaştırılması

Bahsedilen iki yöntemden ilk önerilen Khan vd. çalışmasıdır. Bu çalışmayı referans alarak Camara vd. bazı eksiklikleri gidermeye çalışmıştır. Öncelikle gruplama yapılarak saldırı, hangi satır aralıklarında olduğu ile ilgili geliştirme yapılmıştır. Khan vd. daha önce de bahsedildiği gibi bozulmayı tüm veritabanı üzerinden bir oran ile belirtiyordu. Ayrıca [9] çalışmasında sütun sayısı kadar satırlara ayrılarak el edilen matrisler sayesinde [8] çalışmasının sütun değiştirme saldırılarına karşı olan dayanıksızlığını da ortadan kaldırmıştır. Bunun güçlü özelliklerinin yanında [9] çalışması gruplama işlemi, determinant alma ve diyagonal minörlerin hesaplanması gibi uzun işlem zamanı isteyen matematiksel işlemler içerdiği için daha fazla CPU zamanı tüketir. Ayrıca her grup için elde edilen damgaların, hangi grup üzerinde saldırı olduğunun tespit edilmesi için saklanması gerekmektedir. Bu yüzden saklanan damga boyutu da [8] çalışmasına göre oldukça büyüktür. Örnek verecek olursak; çalışmada 581012 satır ve BA hariç 10 tamsayı sütununa sahip veritabanı kullanılmıştır ve bu veritabanından (581012+8) / 10 adet grup elde

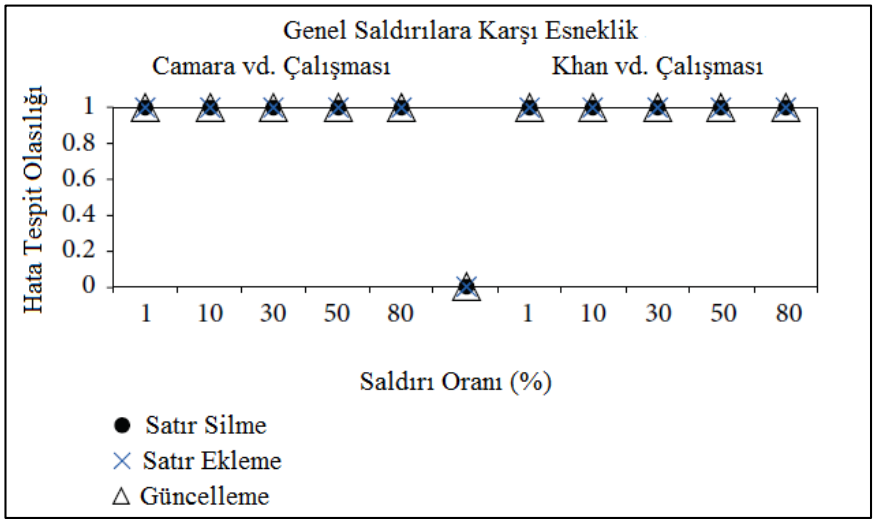
edilmiştir. Her grup için hesaplanan determinant ve diyagonal minörler hesaplanarak metin olarak birleştirilmiştir. Bu da yaklaşık 5MB'lık boyuta sahip bir veri olmaktadır.

[9] çalışması özellikle damganın tespit edilebilmesi konusunda güçlü yönünü elde edilen matrislerin determinantı ile aynı sonucu verebilecek bir matrisin oluşturulma zorluğunda görmektedir. İki çalışmanın genel hatlarıyla karşılaştırılması Tablo 5'deki gibidir.

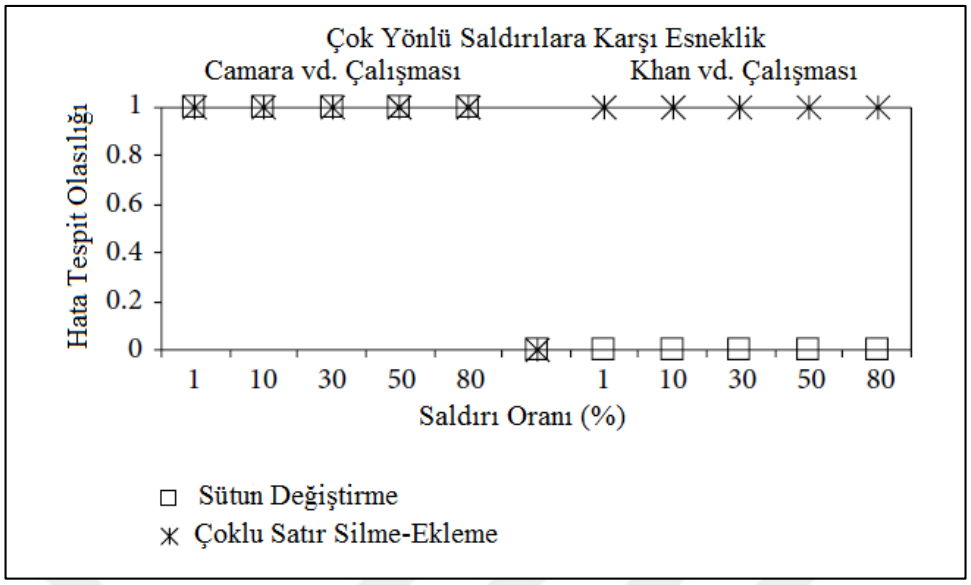
Tablo 5. Camara vd. ile Khan vd. karşılaştırılması [9].

	Camara vd.	**Khan vd.**
Ekleme, Silme, Güncelleme	Grup seviyesinde algılama ve yer bulma	Algılama ve karakterize etme
Veritabanı bölümleme	Bölümleme tabanlı	Bölümleme yok
Satır ekleme-Silme	Bu tür saldırılara karşı esnek	Bu tür saldırılara karşı esnek
Sütun yer değiştirme	Bu tür saldırılara karşı esnek	Bu tür saldırılara karşı esnek değil
Güvenlik	GA tabanlı	GA tabanlı

Genel saldırılar ve çok yönlü saldırılara karşı karşılaştırmaları da Şekil 10 ve Şekil 11'deki gibidir.



Şekil 10.Camara vd. ile Khan vd. genel saldırılara karşı esneklikleri [9].



Şekil 11. Camara vd. ile Khan vd. çok yönlü saldırılara karşı esneklikleri [9].

İlgili yöntemleri açıklayıp karşılaştırdıktan sonra bir veritabanı sistemine yapılabilecek olan saldırılar bir sonraki bölümde açıklanmıştır.

1.5. Ayrık Kosinüs Dönüşümü

Bilginin ifade edildiği ve gösterildiği düzlemden başka bir düzleme aktarılarak, o düzlem üzerinde ifade edilmesi dönüşüm olarak adlandırılır. 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 DCT kullanılır. Bununla birlikte DCT, sinyalin enerjisini daha küçük bir alana sıkıştırarak, sinyalin daha az veriyle ifade edilebilmesini sağlamaktadır [39].

DCT görüntü sıkıştırma işlemlerinde yoğun olarak kullanılmaktadır. Birçok görüntü işleme standardında DCT işlemi 8x8 piksellik parçalara ayrılarak uygulanır. DCT işlemi 8x8 bloklardan daha büyük parçalara ayrılıp uygulandığında sıkıştırmada kayda değer iyileştirme sağlamamaktadır. DCT uygulanması sonucunda elde edilecek katsayılardan sol üst köşedeki katsayı en alçak frekans bileşeni olan DC bileşeni ifade etmektedir. DCT katsayılarının hesaplanması (5)'deki formül ile gerçekleştirilmektedir.

$$F(u,v)=\left(\frac{2}{N}\right)^{\frac{1}{2}}\left(\frac{2}{M}\right)^{\frac{1}{2}}\omega(i)\omega(j)\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\cos\left[\frac{\pi.u}{2.N}(2i+1)\right]\cos\left[\frac{\pi.v}{2.M}(2i+1)\right].f(i,j) \quad (5)$$

Formülde görülen N satır sayısını, M ise sütun sayısını ifade etmektedir. Ayrıca formülde yer alan ω(i) fonksiyonu da (6) formülünde olduğu gibi değer almaktadır:

$$\omega(k)=\begin{cases}\frac{1}{\sqrt{2}} & k=0\\ 1 & \text{diğer}\end{cases} \quad (6)$$

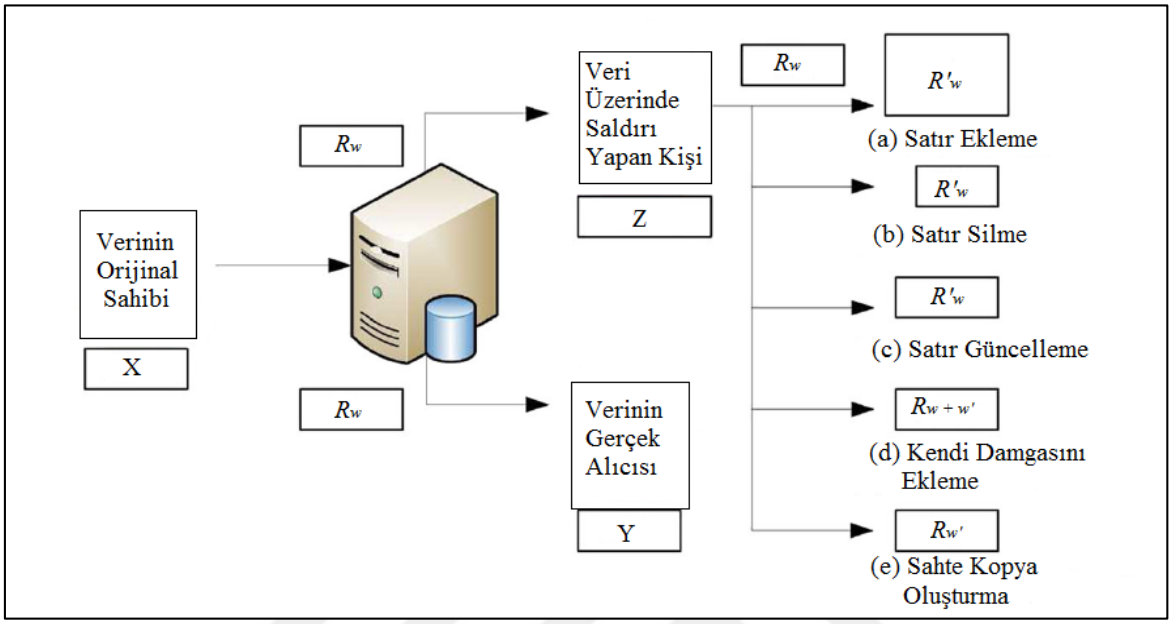
Tez çalışmasında DCT'nin kullanılmasının amacı her DCT katsayısının hesaplanması aşamasında tüm matris değerleri kullanıldığından herhangi birinde meydana gelmiş değişikliğin üretilen katsayıları değiştirecek olmasıdır. Bu değişiklikler de orijinal veritabanından üretilmiş olan damga bilgisi ile şüpheli veritabanından üretilecek olan damga bilgilerinin farklı olmasına neden olacaktır ve böylece yapılmış olan herhangi bir saldırı sonucunda algılama gerçekleştirilmiş olacaktır. (5) formülünün uygulanması sonucunda elde edilecek olan DCT katsayıları pozitif veya negatif olabilmektedir. Bu katsayılar pozitif ise 1 negatif ise 0 olarak etiketlenmektedir.

1.6. Saldırı Türleri

Saldırı türlerini açıklamadan önce yapılabilecek saldırı ile ilgili bir senaryo oluşturalım. Öncelikle şunu belirtelim bir geri dönüştürülebilir damgalama tekniği güçlü ise aşağıdaki saldırılara karlı dayanıklı olmalıdır. Oluşturacak olduğumuz senaryoda verinin sahibi X, alıcı Y ve saldırgan Z olsun.

X, hırsızlık ve yanlış sahiplik iddiası için kendi verisini korumak zorundadır. Y ise veri kalitesinden ödün vermeden veriyi talep eder. Z ise hile veya kimliklendirme yaparak veriyi çalmaya çalışır. Böylece kendi sahiplik iddiasını yapar veya X'in veri üzerindeki sahiplik iddiasını kaldırmaya çalışır. Z damgalanmış veriden edinimler yaparak, damga üzerinde bozulma veya tamamen ortadan kaldırma saldırıları yapabilir. Bu saldırılar kötü niyetli saldırılar kadar ekleme, silme, güncelleme, sıralama veritabanı üzerinde yapılan kasıtsız doğal işlemlerden birisi de olabilir. Diğer taraftan Z de bilinçli olarak seçili satır veya sütunlar üzerinde ekleme, silme, düzenleme sıralama, kendi damgasını ekleme veya sahte bir kopya oluşturma işlemi yapabilir. Veritabanı üzerinde yapılabilecek birçok saldırı

mevcuttur bunlardan bazıları aşağıda verilmiştir. Yapılabilecek bazı saldırılar Şekil 12'de görülmektedir.



Şekil 12. Veritabanı üzerinde yapılabilecek saldırı diyagramı

1) Rasgele Saldırı: Bu saldırı türünde, saldırgan rasgele belirlenmiş satırlar üzerinde belirli hücrelerdeki değerleri rasgele olarak değiştirir. Bu saldırıda ayrıca sıfıra dönüştürme veya bit değiştirme meydana gelebilir. Sıfıra dönüştürmede rasgele seçilen bitin değeri sıfıra dönüştürülür, bit değiştirmede ise bitin değeri tersine çevrilir.
2) Yuvarlama Saldırısı: Saldırgan sayısal özellik içeren damgayı en yakın tamsayı değerine yuvarlayarak kaybetmeye çalışır. Bu saldırı türünde, veri kalitesi büyük ölçüde bozulur. Bunun anlamı da veri tamamen kullanışsız hale gelir. Damgalamada ilgili doğru bit pozisyonu tahmin edilene kadar bu saldırı başarılı olmayabilir.
3) Bit Saldırısı: Saldırgan bu saldırı türünde bazı bitleri değiştirerek damgayı yok etmeye çalışır [3]. Eğer saldırgan tüm bitleri değiştirebilirse kolaylıkla damgayı yok edebilir. Fakat bu saldırının veriyi tamamen kullanışsız yapabilecek bir sakıncası mevcuttur.

4) Alt Küme Saldırısı: Saldırgan damgalanmış ilişkinin satır veya sütun alt kümesini alarak damganın kaybolduğunu veya yok edildiğini umar. Eğer saldırgan çok sayıda satır ya da sütun alırsa verinin kalitesi bozulur.
5) Karışık ve Ekleme Saldırısı Karışık ve eşleşme saldırısında saldırgan benzer bilgileri içeren çoklu ilişkilerden ayrık olarak alarak kendi ilişkisini oluşturabilir. Bu durumda saldırgan tam veritabanını oluşturması gerekmektedir ve kolaylıkla verinin sahibinin kim olduğu ve kimin saldırgan olduğu belirlenir.
6) Tersinirlik Saldırısı: Saldırgan eğer başarılı şekilde hayali bir damga üretebilirse sahiplik iddiası için tersinirlik saldırısı yapabilir [37].
7) Hile Saldırısı: Eğer saldırgan verinin bir veya daha fazla kopyasına erişirse hile yaparak damga verisini belirleyebilir veya kaldırabilir [38].
8) Ekleme Saldırısı: Saldırgan veri kümesine yeni satırlar ekler. Yeni satırların eklenmesi damga üzerinde ek gürültü olarak ortaya çıkar. Bununla birlikte fazladan satır eklenmesi damgalanmış veritabanı üzerinde birçok hatayı meydana getirir. Eğer bu hataların sayısı fazla artarsa veritabanı kullanılamaz hale gelebilir. Ek olarak damgalanmış veriye yeni satırlar eklenmesi senkronizasyon hatası da yaratabilir.
9) Değiştirme Saldırısı: Bu saldırıda saldırgan satırlardaki veri değerlerinde değişiklik yapar. Burada saldırgan veri değiştirmenin damgayı bozabilme ihtimali ile karşı karşıya kalacaktır. Saldırgan veri kümesini değiştirmek için erişim hakkına sahip değildir ve böylece kolaylıkla kullanılabilirlik kısıtlamalarını ihlal eder ve veriyi kullanılamaz hale getirir.
10) Silme saldırısında saldırgan bilinçli olarak işaretlenmiş veri kümesinden satırlar siler. Satırların rasgele silinirse ortalama olarak her bölümden silinen satır sayısının tüm bölümlere oranı kadar satır kaybolmuş anlamına gelir. Damganın eklenmesi veri bölümlerinin başlangıcını ve bitişinin belirlemek için işaretçi satırlar kullanır [40] Eğer satırlar veri kümesi üzerinde büyük bir hata meydana getiriyorsa başarılı silmedir.
11) Mozaik Saldırı: Mozaik saldırıda veri kümesi çok sayıda küçük bölümlere ayrılır [41-43]. Bu saldırı daha çok görüntü, video ve diğer multimedya veriler üzerinde yapılmaktadır. Çok küçük bölümler oluşturduktan sonra bazı küçük boyutlu veriler veri kümesinden silinir.

2. YAPILAN ÇALIŞMALAR

Günümüzde internet kullanımın artması bilgilerin daha geniş kullanıcılara ulaşabilmesini sağlamıştır. Fakat daha geniş kullanıcı kitlesine ulaşmak beraberinde bazı problemleri de getirmektedir. Bu sorunlar önceki bölümde de bahsedildiği gibi bilginin sahipliği, telif hakları, bütünlük kontrolleri gibi sorunlardır. Özellikle görüntü olmak üzere damgalama çeşitli alanlarda kullanılmıştır. Veritabanları da bu alanlardan birini oluşturmaktadır. Sahiplik koruması, bütünlük doğrulanması gibi veritabanı için önem arz eden konular veritabanlarının kullanımının artması ve internet erişimleriyle dışa açık hale gelmesi damgalama işlemlerinin bu alanda kullanılmasının önünü açmıştır. Özellikle yüksek önem arz eden veritabanları için verinin güvenliği ve bütünlüğü büyük öneme sahiptir. Örneğin bir medikal veritabanına yapılacak kötü niyetli saldırılar ile ortaya çıkacak değişiklikler kritik sonuçları beraberinde getirebilir. Saldırı sonucu herhangi bir hasta bilgisinde oluşabilecek bir değişiklik hasta ile ilgili verilecek tıbbi kararları ciddi şekilde etkileyebilir. Başka bir örnek vermek gerekirse; internet üzerinden ticaret yapan bir sisteminin veritabanında yapılacak kötü niyetli değişikliklerle ürünlerin fiyatları üzerinde oynanıp maddi zararlara neden olunabilir. Veritabanının bütünlüğünün korunması her iki örnekte de görüldüğü gibi burada önem arz etmektedir.

Bilinen ilk çalışmanın 2002 yılında [2] olduğu düşünülürse veritabanı damgalamanın uzun geçmişe sahip olmadığını söyleyebiliriz. Son yıllarda bu alandaki çalışmalar hızla artmaktadır ve birçok alternatif yöntem sunulmaktadır. Önerilen yöntemler temel olarak bozulma tabanlı ve bozulmadan bağımsız olmak şekilde ikiye ayrılıyor. Bozulma tabanlı yöntemlerde mevcut veriden elde edilen damga yine veri içerisinde tutulmaktadır. Fakat ilk bölümde vurgu yapıldığı gibi veri içerisinde tutulan damga, verinin kalitesi açısından sorunlar teşkil edebilmekte ve saklanacak olan damga bilgisinin kapasitesini etkilemektedir. Çünkü üretilen damganın veritabanı içerisine eklenmesi sürecinde eklerken verinin kalitesi ve veritabanının bütünlüğü bozulmamalıdır. Bu da bazı kısıtlamaları getirmektedir. Bozulmadan bağımsız yöntemlerde ise mevcut veri üzerinde herhangi bir değişiklik yapılmamaktadır. Bu sayede damga bilgisinin kapasitesi ile ilgili sınırlandırmalar da ortadan kalkmaktadır. Ayrıca damga bilgisi mevcut veri içerisine saklanmadığı için veri kalitesinde herhangi bir kayba da neden olmamaktadır. Böylece bozulma tabanlı tekniklerin getirdiği bazı kısıtlamalar da aşılması sağlanmıştır. Yapılan çalışmada iki farklı yöntem üzerinde

durulmuş ve bu yöntemlerin zayıf yanları üzerinde iyileştirme yapılmaya çalışılarak daha etkin yeni damgalama yöntemleri önerilmiştir.

Yapılan çalışmalar için ayrıntılı anlatıma geçmeden önce genel olarak içerikleri aşağıdaki şekilde verilebilir.

İlk önerilen çalışmada [8] çalışması üzerinde durularak bu çalışmada mevcut olan bazı eksikliklerin giderilmesine çalışılmıştır. [8]'de önerilmiş olan yöntem çalışma daha önce de bahsedildiği gibi kırılgan sıfır damgalama şeması sunmaktadır. Önerilen şemada ilk önce 3 alt damga üretilir. Ardından elde edilen bu alt damgaların birleştirilmesi sayesinde ilişkisel veritabanı için genel bir damga elde edilmektedir. İlişkisel veritabanı üzerinde bütünlük kontrolü yapılmak istendiğinde orijinal veritabanı için yapılan işlem adımları şüpheli veritabanı için de tekrardan yapılır. Bu işlemler yukarıda verildiği gibi 3 alt damganın elde edilmesi ve ardından bu 3 alt damganın birleştirilerek ilişkisel veritabanı için bir damganın elde edilmesi işlemleridir. Üretilen damga daha önce orijinal veritabanından elde edilip saklanmış olan damga ile karşılaştırılır ve tüm veritabanı için bir bozulma oranı tespit edilir.

Çalışma kapsamında 3 alt damga oluşturulması yerine sütun saldırılarını da tespit edebilmek amacıyla veritabanının hücre değerleri üzerinden elde edilen ikililerin frekans değerleri oluşturulmuştur. Elde edilen her ikili değere o hücrenin sütun numarası da eklenerek sütun değiştirme saldırılarına karşı dayanıklılık sağlanmaya çalışıldı. Böylece ilişkisel veritabanı için 3 histogram elde edip bu elde edilen histogramların birleştirilmesiyle bir genel veritabanı damgası elde etmek yerine tek histogram kullanılarak hem saklanacak damga bilgisinin boyutu küçültülmüş hem de sütun değiştirme saldırılarına karşı dayanıklılık sağlanmıştır. Ayrıca [8]'de önerilen yöntemde olduğu gibi tüm veritabanı üzerinde meydana gelen bozulma oranı da tespit edilebilmektedir. Elde edilen deneysel sonuçlar da önerilen çalışmanın [8] çalışmasına kıyasla daha az işlem zamanı tükettiği de gösterilmiştir. İlerleyen bölümlerde deneysel sonuçlar ve ayrıntılar kapsamlı şekilde verilecektir.

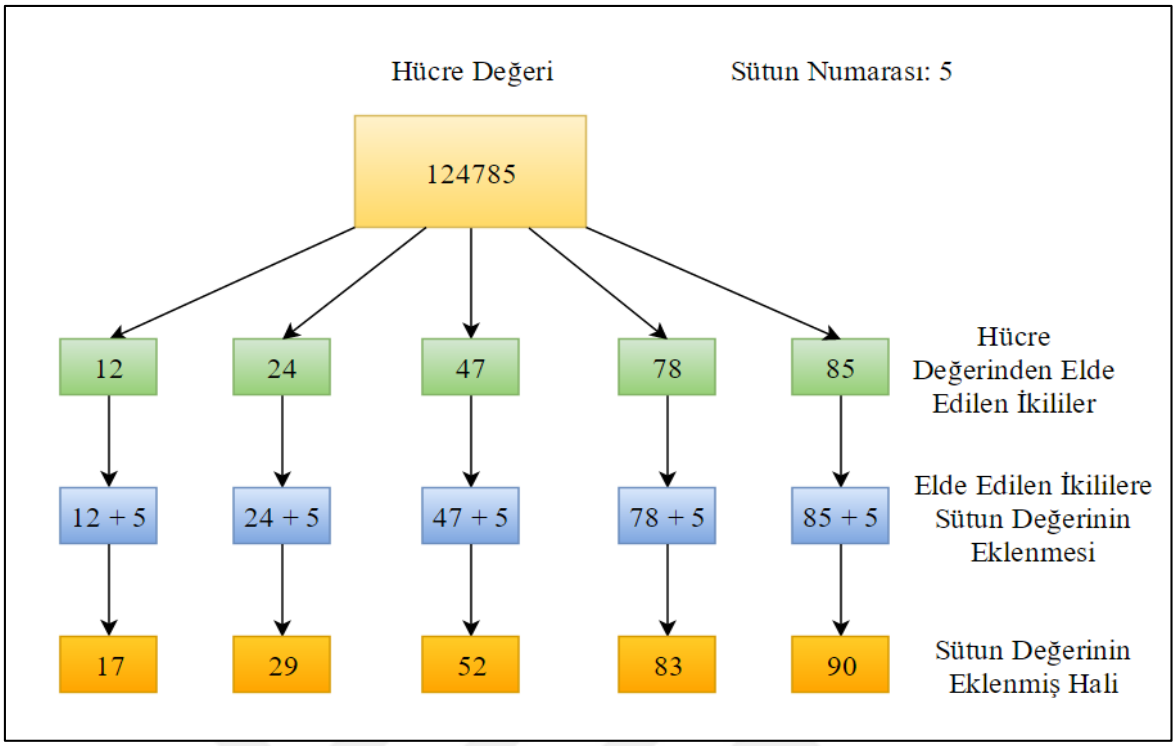
Tez kapsamında gerçekleştirilen diğer bir çalışmada [9] ele alınmış ve var olan problemleri ortadan kaldırmıştır. [10]'da verilen, yöntem sütun saldırılarına karşı direnç sağlamak amacıyla daha önce de ayrıntısı anlatıldığı gibi matris üzerinde determinant hesaplama ve diyagonal minörlerin hesaplanması gibi yoğun matematiksel işlemler kullanıldığı için fazla işlem zamanı gerektirmektedir. Bunun yanında saklanan damga boyutu da oldukça büyük olmaktadır. Fakat [8] çalışmasının sütun değiştirme saldırılarına karşı olan dayanıklılık problemini gidermektedir. Çalışma ilk adımda sütun sayısı kadar satır içeren grupların elde edilmesi ile başlar. Böylece her grup için bir matris elde ediliyor. Bu

matrisler üzerinde önce determinant bulma işlemi gerçekleştirilir. Ardından ilgili matris için diyagonal minörler bulunarak, determinant ve minör değerleri birleştirilir. Her grup için yapılan bu işlem tekrarlandıktan sonra bütün grupların damgaları birleştirilerek tüm ilişkisel veritabanı için bir damga elde edilir.

Yapılan çalışmada [9] çalışmasının, yukarıda verilmiş olan işlem zamanı ve damga boyutu açısından ortaya çıkarmış olduğu problemler üzerinde iyileştirilmeye çalışılmıştır. Matris üzerinde yapılan işlemler fazla işlem zamanı tükettiği için bunun yerine damga üretiminde DCT kullanılmıştır. DCT tabanlı özellik çıkarımı daha önceleri görüntü üzerinde kullanılmış olan bir yöntem olmakla beraber ilişkisel veritabanı üzerinde kullanılmamıştır. Önerilen çalışma veritabanını gruplandırmaktadır. Fakat bu gruplandırmada sütun sayısı kadar satır almak yerine determinant ya da minör işlemleri yapılmayacağından dolayı sütun sayısının yarısı kadar satır alınmıştır. Elde edilen gruplar üzerinde DCT uygulanmış ve DCT katsayıları elde edilmiştir. Böylece işlem zamanı azalmakla birlikte saklanan damga verisinin boyutu da azalmış olduğu deneysel çalışmalarda gösterilmiştir. Tez kapsamında önerilen her iki yönteme ilişkin detaylar ilerleyen bölümlerde verilmektedir.

2.1. İkililerin Histogramına Dayalı İlişkisel Veritabanı Damgalaması

Kırılgan sıfır damgalama yöntemi öneren [8] çalışmasından önceki bölümlerde ayrıntılı olarak bahsedilmektedir. Yapılan tez çalışmasında bu yöntemin dayanıksız olduğu yönler ve tükettiği işlem zamanı üzerinde durulmuştur. Önerilen yöntem yine bir kırılgan sıfır damgalama yöntemidir ve veritabanı üzerinde herhangi bir bozulma meydana getirmemektedir. Elde edilen damga veritabanından bağımsız olarak saklanmaktadır. Daha sonra şüpheli veritabanından elde edilen damga ile karşılaştırılıp veritabanının bütünlüğünün bozulup bozulmadığı hakkında karar verme işlemi gerçekleştirmektedir. Önerilen yöntem [8]'deki gibi 3 alt damga oluşturmak yerine tüm veritabanından tek damga elde etmektedir. Bu damganın elde edilmesi her hücrede bulunan değerlerin ikili olarak gruplandırılmasına dayanır. Bu gruplandırma işlemi Şekil 13'de gösterildiği gibidir.



Şekil 13. Hücre değerlerinin ikili olarak ayrılması

Görüldüğü gibi Şekil 13'den veritabanının hücrelerindeki sayısal değerlerin ikililere ayrılması ve bu ikililere sütun saldırılarını tespit edebilmek için sütun numaralarının eklenmesi işlemi ayrıntılı olarak örnek üzerinde gösterilmiştir. Örnekte ilgilenilen hücredeki sayısal değer 124785'dir. Bu değer algoritma önerildiği gibi ilk adım olan ikililere ayrılmalıdır. İkililere ayırma için sayısal ifadenin en büyük basamağından başlanarak her basamak değeri ve sonrasında gelen basamak değeri alınır ve ikililer elde edilir. Şekil 12'de de verildiği gibi 124785 değeri öncelikle 12, 24, 47, 78 ve 85 ikililerine ayrılır. Ardından algoritmada önerilen sütun değişim saldırılarına karşı direncin sağlanması amacıyla sayının bulunduğu sütun değeri tüm elde edilen ikililere eklenir. Sütun numarasının eklenmesi ardından elde edilen ikililer 17, 28, 52, 83 ve 90 ikilileri olmaktadır.

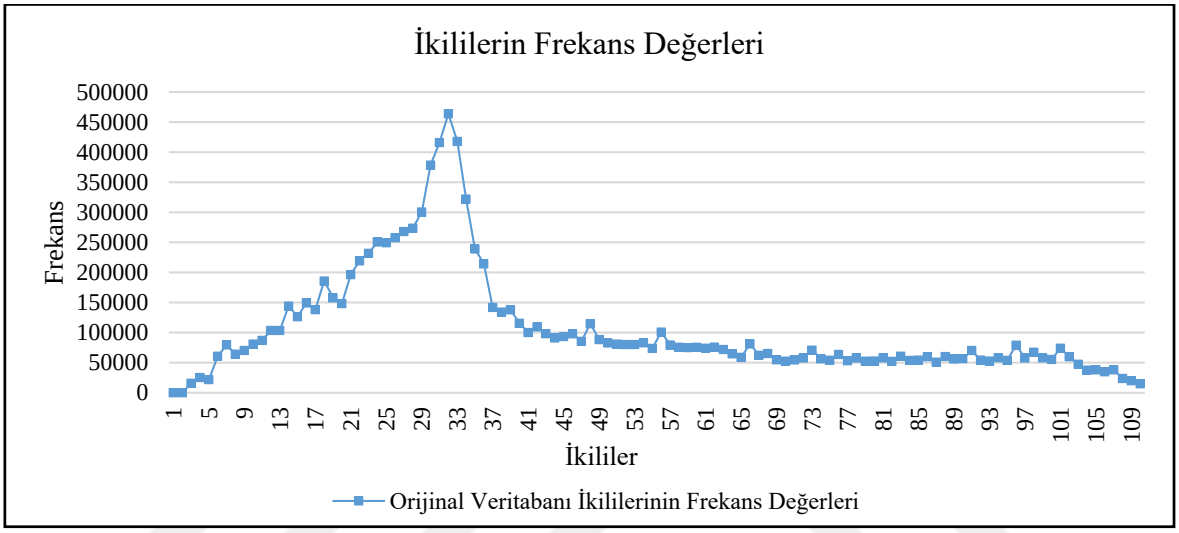
İkililerin hesaplanması, bu ikililere ait frekans değerlerinin hesaplanması ve son olarak her ikiliye ait frekans değeri ile elde edilen oranlar kullanılarak ilişkisel veritabanına ait damganın elde edilmesi ile ilgili yalancı kod ifadesi ve verilen yalancı kod ifadesinin açıklaması aşağıda verilmiştir.

```
Girişler: R ilişkisel veritabanı, GA Gizli Anahtar.
Çıkış: ω_R, R ilişkisel veritabanına ait damga
(1)  For each satir s_i ∈ R do
(2)    For each sutun A_j ∈ R do
(3)      uzunluk = Len(s_i · A_j)
(4)      If uzunluk=1 then
(5)        p = s_i · A_j + j
(6)        ikililer[p]++
(7)        toplam_ikililer++;
(8)      Else
(9)        For k=0 to uzunluk-1
(10)         İkili = (s_i · A_j)[k] || (s_i · A_j)[k+1]
(11)         p=ikili + j
(12)         ikililer[p]++;
(13)         toplam_ikililer++;
(14)       End for
(15)     End if
(16)   End for
(17) End for
(18) For each i ∈ ikililer do
(19)   i_frekans_i = (ikililer[i] / toplam_ikililer)*100
(20)   ω_R = ω_R || i_frekans_i
(21) End for
(22) ω_R = ω_R || toplam_ikililer
(23) Eω_R = Şifrele(ω_R || GA)
(24) ω_C = Eω_R || sahip_id || tarih || saat
(25) Return ω_C
```

Önerilen algoritmada elde edilen ikililer [1 - (99+sütun sayısı)] aralığında değişmektedir. Üretilen bu frekans değerleri histogram oluşturulmasında kullanılmakta ve oluşturulan histogram da veritabanı bütünlüğünün kontrol edilmesi için kullanılmaktadır.

Yukarıdaki verilen yalancı kod ifadesini adım adım inceleyelim: Satır 1 ve Satır 2'de veritabanının tüm kayıtları inceleneceğinden döngüler mevcuttur. Satır 3'de s_i. satır, A_j. sütununda bulunan kaydın uzunluğu alınır. Ardından alınan bu değerin Satır 4'de uzunluğu kontrol edilir ve bu uzunluk değeri 1 ise bu değer tek basamaklı bir değerdir ve Satır 5'de olduğu gibi doğrudan sütun numarası eklenir. Satır 6 ve Satır 7'de sütun numarası eklenerek elde edilen değere ait frekans değeri ve toplam ikililerin değeri 1 arttırılır. Eğer s_i. satır, A_j. sütununda bulunan kaydın uzunluğu 1'den büyük ise bu değer önerilen yönteme göre ikililere ayrılması gerekmektedir. Bunun için Satır 9'daki gibi kaydın uzunluğunun 1 eksiğine kadar bir döngü kurulur. Bu döngüde ilk olarak Satır 10 ve Satır 11'de Şekil 15'deki işlem gerçekleştirilir. Yine elde edilen sütun numarası eklenmiş değer için Satır 12'de frekans değeri 1 arttırılırken, toplam ikililerin sayısı da 1 arttırılır. Elde edilen ikili değerler için Satır 19'da her ikilinin tüm ikililer içindeki yüzdelik oranı hesaplanır ve yüzdelik ifadeler birleştirilerek damga elde edilir. Satır 22'de bu ikililere toplam ikili sayısı eklenir

ve Satır 23'de oluşturulan damga GA ile şifrelenir. Son işlem olarak da şifrelenmiş olan damga bilgisi kullanıcının kimlik bilgisi ve tarih bilgileriyle sertifikalandırılır. Yapılan uygulamada orijinal veritabanı üzerinden elde edilen ikililer için oluşturulan frekans grafiği Şekil 14'deki gibidir. 10 sütunlu bir veritabanında bulunan değerlerden sütun numarası eklenmiş şekilde elde edilebilecek en küçük ikili değer 1 (0+1. sütun numarası) ve en büyük ikili değer ise 109 olacağından ikililer için frekans aralığı [1-109] olmaktadır.



Şekil 14. Orijinal veritabanından elde edilen ikililerin frekans değerleri

Tez çalışmasında önerilen yöntemde Şekil 14'deki grafikte görülen frekans değerlerinin elde edilmesini işlemlerini basit bir veritabanı üzerinde incelenecektir:

Tablo 6. Örnek veritabanı tablosu

Sütun1	**Sütun2**	**Sütun3**
1361	29	573
177	2558	61
59	512	1150

Tablo 6'da verilen veritabanından öncelikle Şekil 13'de gösterildiği gibi ikililer elde edilir. Örneğin Sütun 1'deki ilk satır değeri olan 1361 değeri 13, 36, 61 olarak ikililere ayrılarak bu ikililere 1361 sayısının bulunduğu sütun numarası olan 1 eklenerek 1361 değeri

için sütun numarası eklenmiş ikililer elde edilmektedir. Bu işlem tablodaki tüm değerler için tekrarlandıktan sonra Tablo 7'deki ikililer elde edilir.

Tablo 7. Örnek veritabanından elde edilen sütun numaraları eklenmiş ikililer

Sütun1	**Sütun2**	**Sütun3**
14, 37, 62	31	60, 76
18, 78	27, 57, 60	64
60	53, 14	14, 18, 53

İkililer elde edilirken aynı zamanda ikililer için frekans değerleri hesaplanmaktadır. Frekans değerleri hesaplanırken sütun numarası eklenmiş ikililerin her birinden kaç üretildiği kullanılır. Tablo 7'da görülen ikililer incelenirse Sütun1'in ilk satırında, Sütun2'nin ve Sütun 3'ün üçüncü satırlarında olmak üzere 3 tane 14 değeri vardır. Yine aynı şekilde Sütun1'in ikinci satırında ve Sütun3'ün üçüncü satırında olmak üzere 2 tane 18 değeri vardır. Bu şekilde her değerin frekans ifadesi çıkarılmaktadır. Tüm ikililer için yapılan işleminin sonucunda örnek veritabanı üzerinden elde edilen ikililerin frekans değerleri Tablo 8'deki gibi olmaktadır:

Tablo 8. Örnek veritabanından elde edilen ikililerin frekans değerleri

İkili Değer	**Frekans**
14	3
18	2
27	1
31	1
37	1
53	2
60	3
62	1
64	1
76	1
78	1

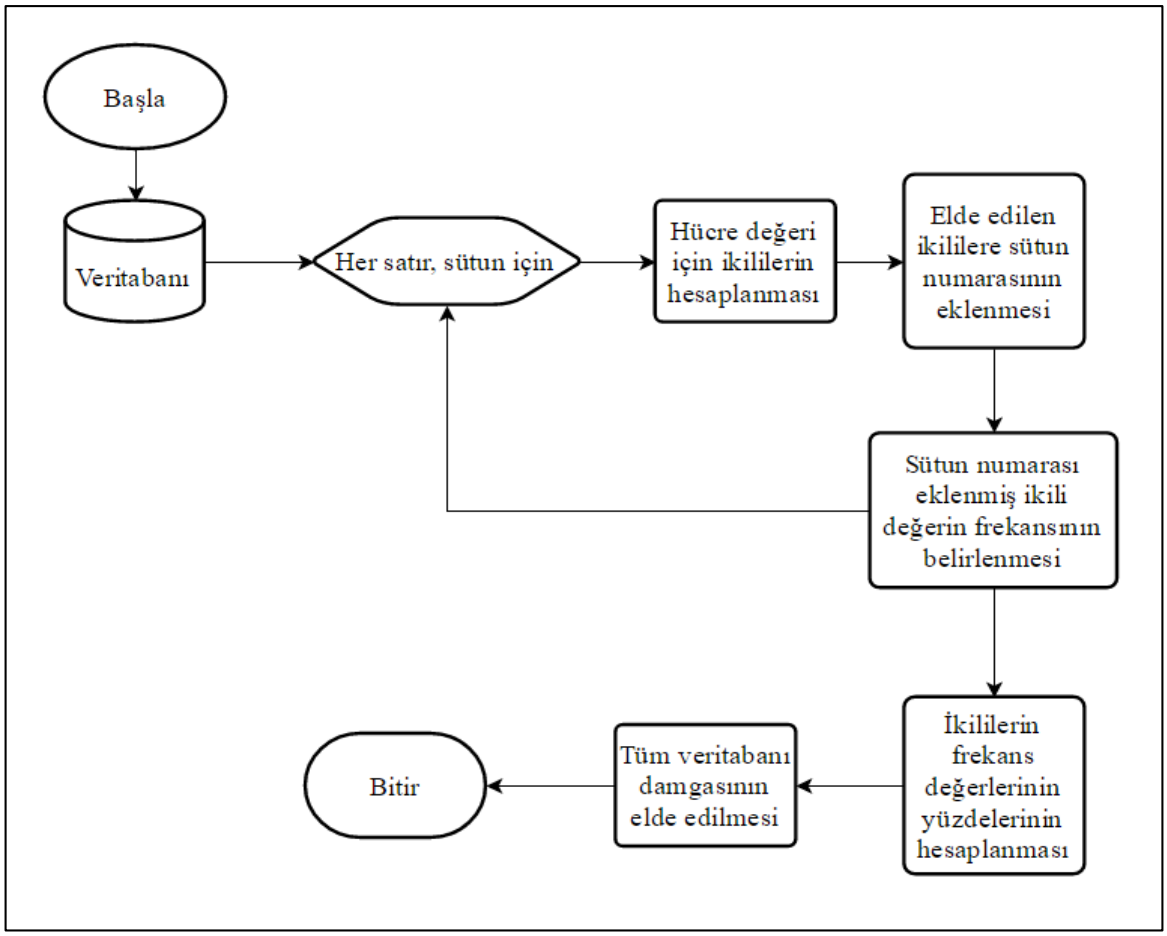
İkililerin frekans değerlerinin bulunmasının ardından her ikili frekans değeri için tüm frekans değerleri içerisindeki yüzdelik dilimi hesaplanmaktadır. Bu hesaplama işleminde ilgili ikili için hesaplanmış olan değerin frekans değerinin veritabanındaki tüm frekans değerlerinin toplamına bölünmesi sonucunun 100 ile çarpılması ile gerçekleştirilmektedir. Tablo 10'daki ikililer için hesaplanmış frekans değerlerinden örnek verecek olursak 14 ikili değeri için hesaplanacak olan yüzdelik oran 3/(3+2+1+1+1+2+3+1+1+1+1)*100 işlemi sonucunda 17,64 çıkacaktır. Her değer için bu şekilde hesaplanan yüzdelik oran sonucunda örnek veritabanı tablosu için yüzdelik oranlar Tablo 9'daki gibi olmaktadır.

Tablo 9. İkililerin frekans değerlerinin yüzdelik oranları

İkili Değer	**Yüzdelik Oran**
14	3/17*100 = 17,647%
18	2/17*100 = 11,764%
27	1/17*100 = 5,882%
31	1/17*100 = 5,882%
37	1/17*100 = 5,882%
53	2/17*100 = 11,764%
60	3/17*100 = 17,647%
62	1/17*100 = 5,882%
64	1/17*100 = 5,882%
76	1/17*100 = 5,882%
78	1/17*100 = 5,882%

İkililerin frekans değerleri üzerinden her ikili için yüzdelik oranların belirlenmesi sonucunda bu oranlar metin olarak birleştirilmektedir. Birleşme sonucunda 17,64711,7645,8825,8825,88211,764,17,6475,8825,8825,8825,882 şekilden bir damga üretilmektedir. Bu damga bilgisine toplam ikililerin sayısı eklenerek veritabanı damga bilgisi üretilmiş olur ve son hali ile üretilen damga 17,64711,7645,8825,8825,88211,764,17,6475,8825,8825,8825,88217 şeklinde olmaktadır. Ardından damga bilgisini saklamak için üretilen damga önce kullanıcının belirlemiş olduğu bir GA değeri ile şifrelenir, ardından sahip bilgisi, tarih ve saat bilgileriyle damgalanarak kaydedilir. Yalancı kod ve örnekle yukarıda anlatılmış olan ikililere dayalı ilişkisel veritabanı damgalama yönteminin akış diyagramı da Şekil 15'de olduğu gibidir. Akış diyagramının vermiş olduğu algoritma öncelikle veritabanının her satır ve sütununda bulunan değerlerinin incelenmesini gerektirir. Her satır ve sütunda bulunan değerler ikililere

ayrılarak tüm veritabanından ikililer elde edilir. Elde edilen bu ikililerin frekans değerleri hesaplanarak ilgili frekans değerinin tüm frekans değerleri içerisindeki yüzdelik oranı bulunmaktadır. Yüzdelik oranlarının birleştirilmesiyle birlikte veritabanının damga bilgisi üretilmektedir. Son işlem adımı olarak da veritabanı damga bilgisi şifrelenerek sertifikalandırılır ve kaydedilir.



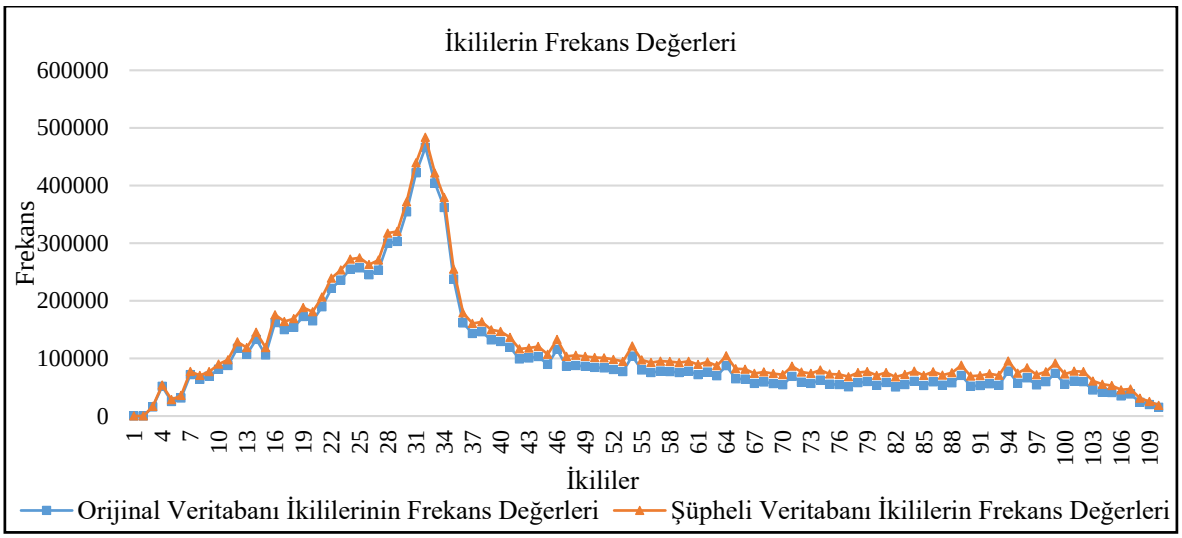
Şekil 15. İkililerin histogramına dayalı veritabanı damgalama akış diyagramı

2.1.1. İkililerin Frekansına Dayalı Yöntemin Saldırılara Karşı Direnci

İlişkisel veritabanı üzerinde yapılabilecek olan saldırılar ayrıntılı olarak önceki bölümlerde anlatılmıştır. Tez çalışmasında [8] yöntemi referans alınarak yapılan uygulamada saldırıların tamamı yerine en sık kullanılan saldırı türleri incelenmiştir. Bu saldırılar satır ekleme, satır silme ve güncelleme saldırılarıdır. Bu temel saldırıların yanında

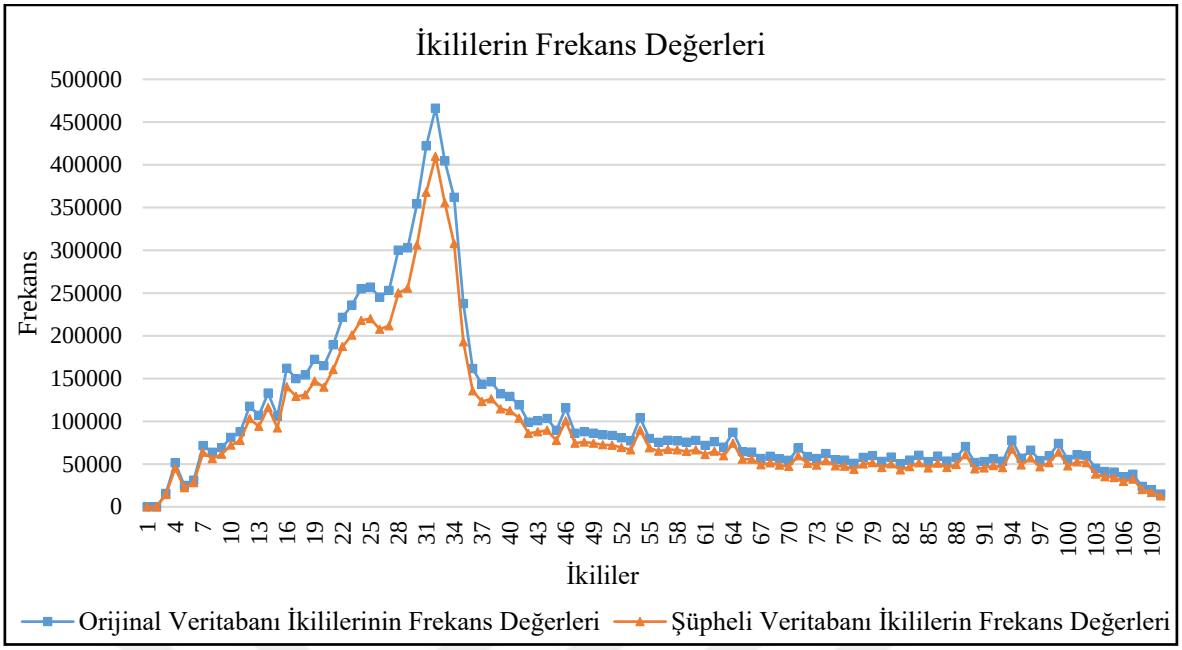
referans alınan çalışmanın dayanıksız olduğu sütun değiştirme saldırıları üzerinde de durulmuştur.

- Ekleme Saldırısı: Bu saldırıda veritabanına değerleri rasgele oluşturulan satırlar eklenmektedir. Yapılan uygulamada eklenecek olan rasgele satırlar belirli bir sayıda veya veritabanının satır sayısının belirli bir oranında gerçekleştirilmektedir. Bu saldırılar veritabanından elde edilecek olan ikililerin frekans değerlerini değiştirmektedir. Değişim ekleme yapıldığı için pozitif eğilimlidir. Frekans değişikliklerine ait grafiği Şekil 16'da görülmektedir.



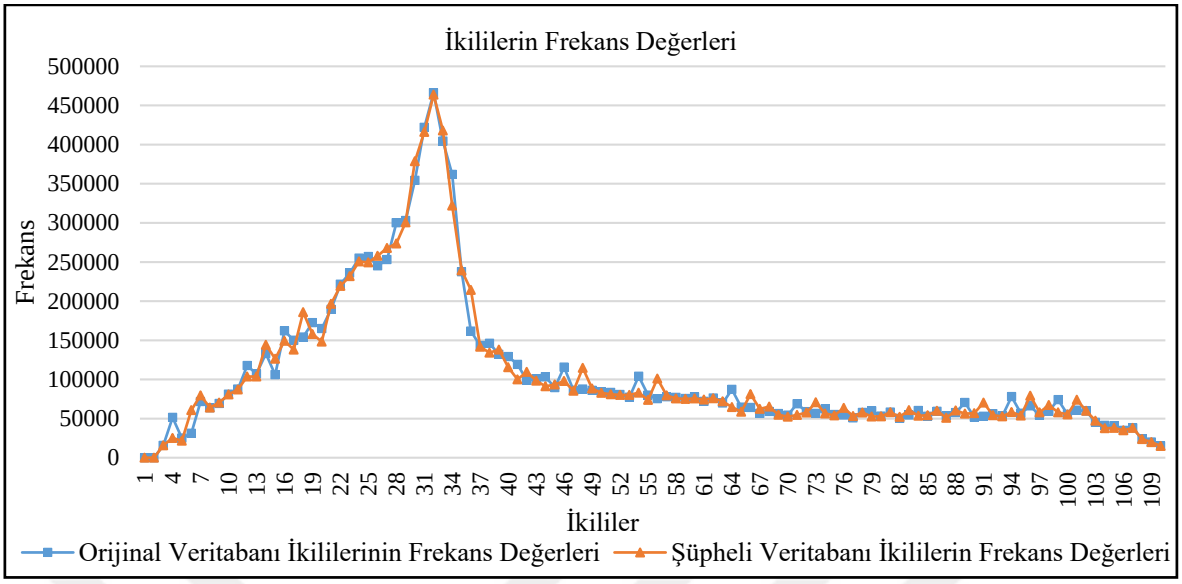
Şekil 16. Ekleme saldırısı sonucunda elde edilen frekans değerleri

- Silme saldırısı: Bu saldırı türünde veritabanından rasgele sütunların silinmektedir. Silinen bu sütunlar veritabanının bütünlüğünü etkilemekte ve orijinal veritabanından elde edilen ikililerin frekans değerleri üzerinde negatif bir eğilim meydana getirmektedir. Silinen satırlar herhangi özel şekilde değil tamamen rasgele seçilmektedir. Silme saldırısı sonucunda elde edilen ikililerin frekans değişim grafiği de Şekil 17'de verilmektedir.



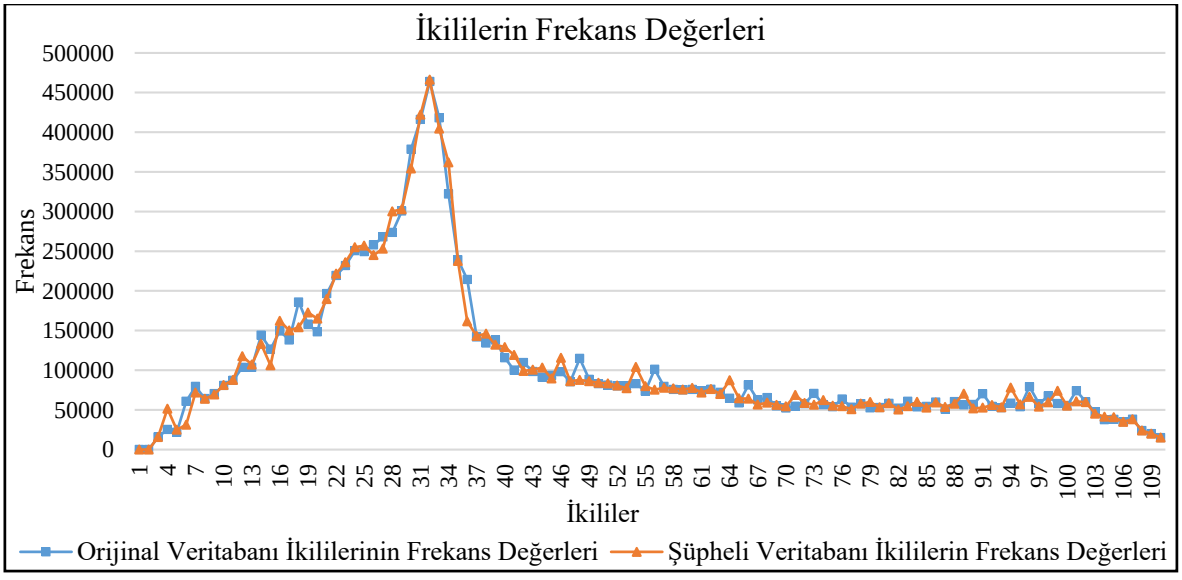
Şekil 17. Silme saldırıları sonucunda elde edilen frekans değerleri

- Güncelleme saldırısı: Ekleme ve silme saldırılarında olduğu gibi güncelleme saldırısında da veritabanındaki değerler üzerinde değişiklik yapılmaktadır. Bu değişiklik rasgele seçilen satır ve sütunlarda bulunan değerlerin yerine yine rasgele üretilen değerler yazılması ile yapılmaktadır. Bu değişimler sonrasında orijinal veritabanından elde edilen ikililerin frekans değerlerinin bazılarında pozitif eğilim görülürken bazılarında ise negatif yönde eğilim görülmektedir. Güncelleme saldırısı sonucunda ikililerin frekans değerlerinde meydana gelen değişiklikleri gösteren grafik Şekil 18'de verilmektedir.



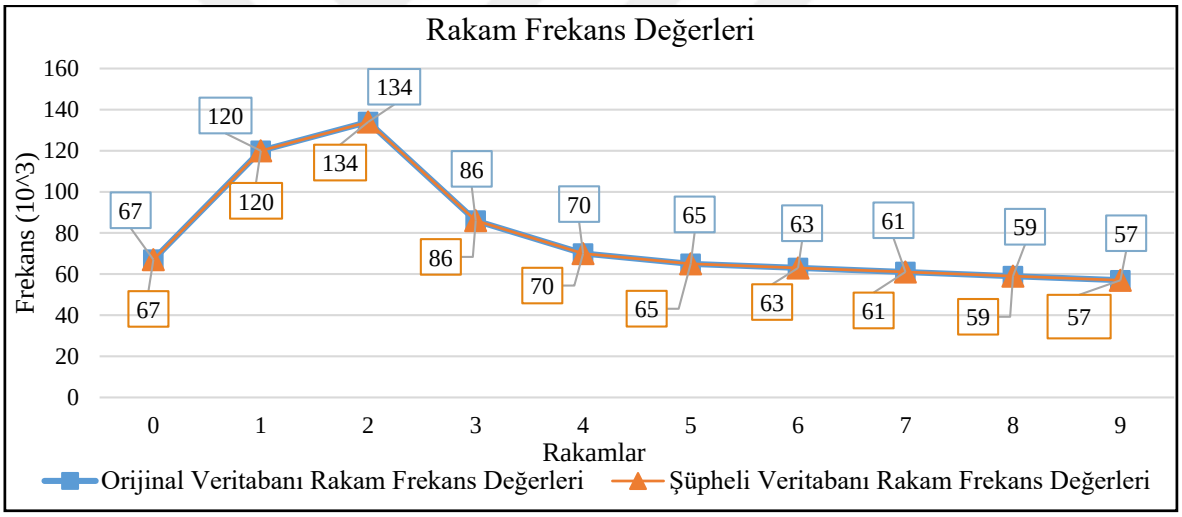
Şekil 18. Güncelleme saldırıları sonucunda elde edilen frekans değerleri

- Sütun değiştirme saldırısı: Son olarak ele alınan sütun değiştirme saldırısında verilerde herhangi bir değişiklik yapılmadan sütunları yerlerinin değiştirilmesi durumunda dahi saldırının algılanması gerçekleştirilmektedir. Referans alınan [8] çalışmasında ise verilerde bir değişiklik yapılmadığından dolayı bu saldırı türünü algılayamaz. Sütun değiştirme saldırıları sonucunda ikililerin frekans değişimlerini gösteren grafik Şekil 19'da verilmektedir.



Şekil 19. Sütun değiştirme saldırıları sonucunda elde edilen frekans değerleri

Önerilen yöntemin 10 sütunlu bir veritabanı için yapılan sütun değiştirme saldırısı sonucunda ikililerin frekans değerlerindeki değişimler Şekil 19'da görülmektedir. Yapılan saldırıda veritabanının 3. ve 5. sütunları yer değiştirilmiştir. Yer değiştirme sonucunda veritabanındaki mevcut veriler üzerinde herhangi bir değişiklik olmayacaktır. Fakat üretilecek olan ikililere eklenecek olan sütun numarası değişiklik göstereceğinden sütun numarası eklenmiş ikililerin frekans değerleri değişecek ve böylece orijinal veritabanından elde edilen ikililerin frekans değerleri ile şüpheli veritabanından elde edilecek ikililerin frekans değerleri farklılık gösterecektir. Aynı saldırı türünde [8] çalışması verilerde değişiklik olmadığından dolayı veritabanının damga bilgisini oluşturmak için üretmiş olduğu rakam, uzunluk ve aralık frekans değerlerinde herhangi bir değişiklik olmayacaktır. Şekil 20'de [8] çalışmasının sütun değiştirme saldırısı sonucunda rakam alt frekans grafiğindeki değişim durumu görülmektedir.

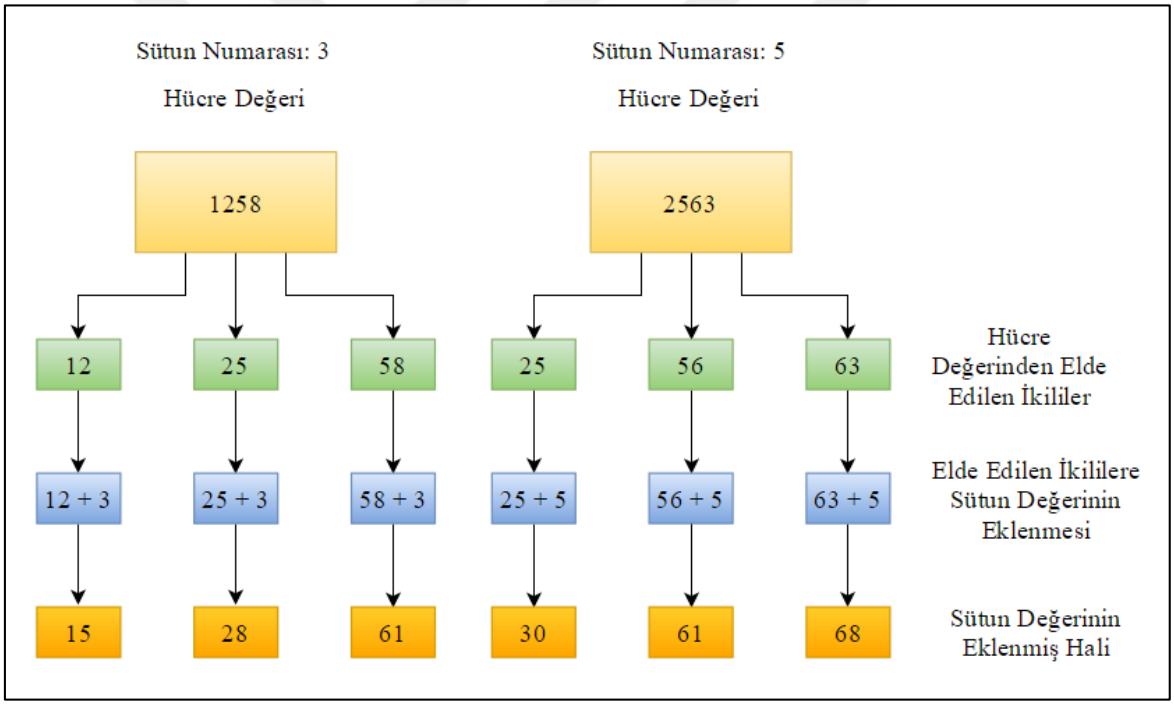


Şekil 20. [8] çalışmasında önerilen yöntemde sütun değiştirme saldırısı sonucunda rakam frekans durumu

2.1.2. İkililerin Elde Edilmesiyle Sütun Değişimi Saldırılarına Karşı Direnç Sağlama

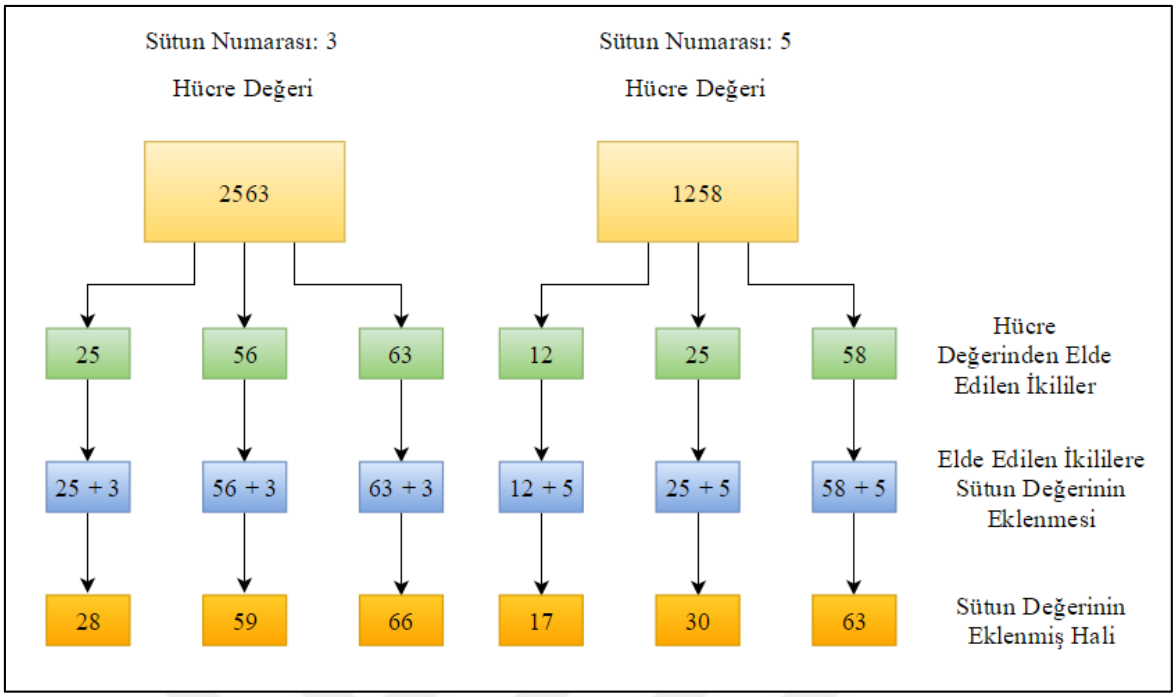
Tez çalışmasında önerilen yöntem ile [8] çalışması ekleme, silme ve güncelleme saldırılarında aynı direnci sağlamaktadır. Bunun yanında ilişkisel veritabanı üzerinde bozulma oranının tespit edilmesinde de aynı sonucu vermektedir. Önerilen yöntemin getirdiği avantaj sütun değiştirme saldırılarında ortaya çıkmaktadır. Önceki bölümlerde

verilmiş olan ikililerin frekans değerlerinin elde edilmesi algoritmasında gösterildiği gibi veritabanı değerlerinden üretilen ikililere eklenen sütun numarası sayesinde bu direnç sağlanmaktadır. Örnek vermek gerekirse 3. sütunda bulunan değer 1258, 5. sütunda bulunan değer de 2563 olsun. Bu değerden üretilecek ikililer 3. sütun için 12, 25, 58 ve 5. sütun için 25, 56, 63'dür. Bu ikililere sütun numarasını da eklersek 3. sütundaki ikililer 15, 28, 61 olarak, 5. sütundaki değerler de 30, 61, 68 olarak değişir. Sonrasında veritabanı üzerinde sütun değiştirme saldırısı yapıldığını düşünelim. 3. sütun ile 5. sütun herhangi bir veri değişikliği yapılmadan yer değiştirmiş olsun. Yeni damga oluşturulurken 3. sütun için üretilen ikililer (25+3), (56 + 3), (63+3) ve 5. sütun için üretilen ikililer (12+5), (25+5), (58+5) olacaktır ve sonuçta ikililerin frekans değerleri değişecektir. Şekil 21'de orijinal veritabanından ikililerin oluşturulması gösterilmektedir.



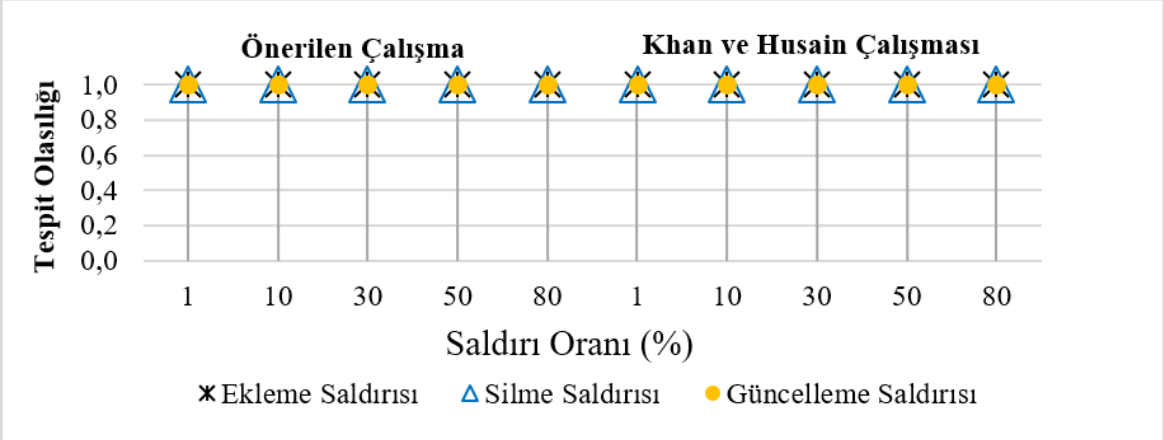
Şekil 21. Orijinal veritabanı için ikililerin oluşturulması

İlişkisel veritabanı üzerinde yapılacak olan ve yukarıda anlatılan sütun değiştirme saldırı sonunda elde edilecek olan ikililer Şekil 22'deki gibi olacaktır. Bunun sonucunda değişecek olan frekans değerleri de üretilecek olan damgayı da değiştirecektir.



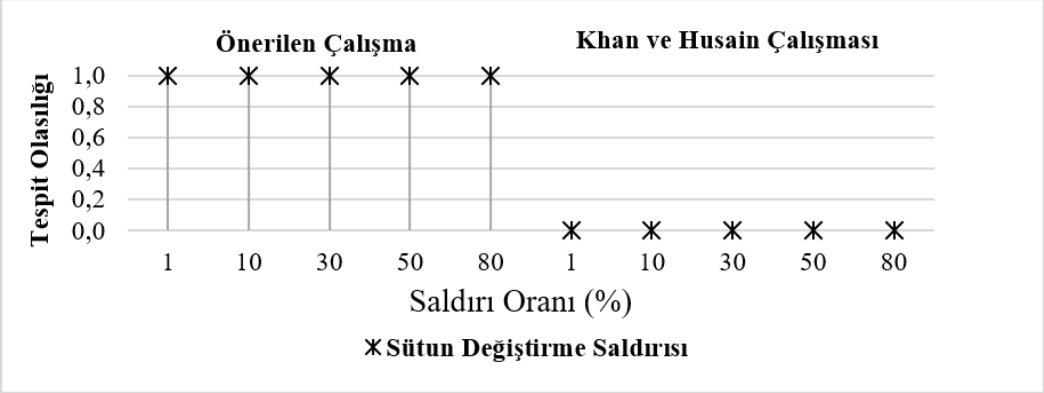
Şekil 22. Sütun değiştirme saldırısı sonucunda ikililerin değişimleri

Sütun değiştirme saldırısının ikililerin frekans değerlerini nasıl etkilediğini inceledikten sonra [8] çalışması ile tez çalışmasında önerilen yöntemin ekleme, silme ve güncelleme saldırılarına karşı olan dirençleri Şekil 23'de verilmiştir.



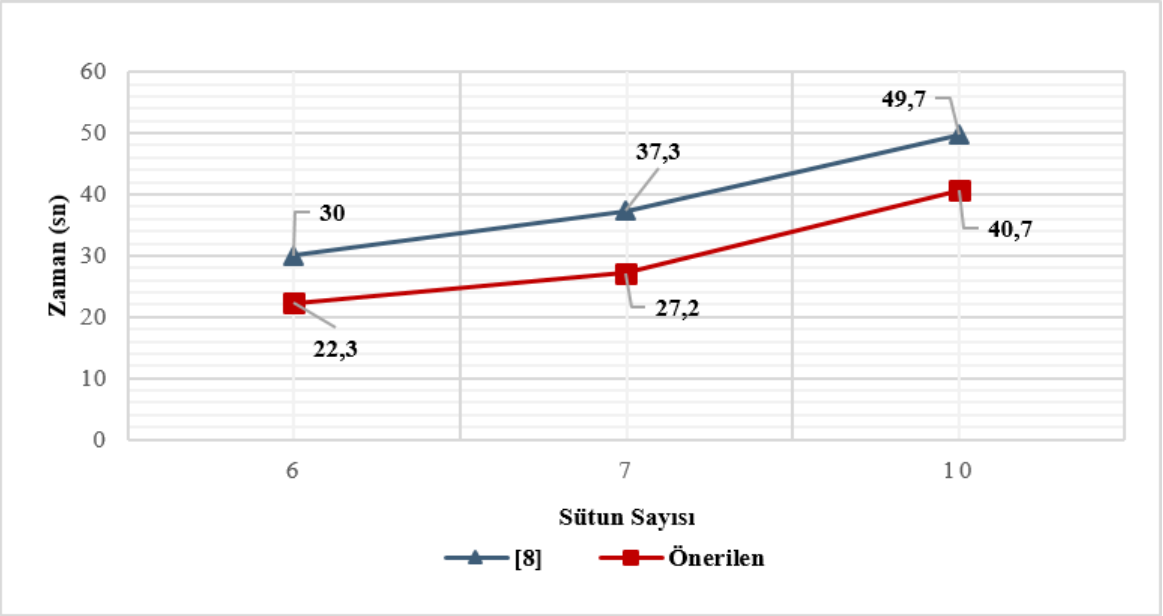
Şekil 23. Tez çalışmasında önerilen yöntem ile Khan vd. önerdikleri yöntemin ekleme, silme ve güncelleme saldırıları için karşılaştırılması

Çalışmada referans alınan yöntemin sütun saldırılarına karşı olan zayıf yönü giderilmeye çalışılmıştır. İkililere eklenen sütun numaraları bu zafiyeti ortadan kaldırmış ve sütun değiştirme saldırılarına karşı %100 başarı sağlanmıştır. Bu tür saldırılara karşı önerilen yöntem ile [8] çalışmasının karşılaştırılması Şekil 24'de görülmektedir.



Şekil 24. Tez çalışmasında önerilen yöntem ile [8] çalışmasında önerilen yöntemin sütun değiştirme saldırıları için karşılaştırılması

Önerilen yöntem, sütun değiştirme saldırılarına karşı direnç sağlamanın dışında işlem zamanı olarak da daha iyileştirme sunmaktadır. Veritabanında mevcut olan sütun sayıları değiştirilerek karşılaştırmalar yapılmıştır. 6, 7 ve 10 sütun içeren veritabanı tabloları üzerinde iki çalışmanın zamansal karşılaştırılması yapılmıştır. Bu karşılaştırma veritabanı damga bilgisinin oluşturulması için geçen zamanı göstermektedir. [8] çalışmasında önerilmiş olan yöntemde ilişkisel veritabanına ait damga bilgisinin oluşturulması sürecinde ilk olarak rakam, uzunluk ve aralık olmak üzere 3 alt damga bilgisi oluşturulmaktadır. Bu işlem de veritabanında bulunan bütün değerlerin 3 defa taranması anlamına gelmektedir. İkililerin oluşturulmasında ise 3 alt damga oluşturmak yerine ikillerden oluşan tek bir alt damga oluşturulduğundan işlem zamanında daha az maliyet getirmektedir. Şekil 25'den de görüldüğü gibi önerilen yöntem harcanan zaman açısından daha iyi bir sonuç sunmaktadır.



Şekil 25. Tez çalışmasında önerilen yöntem ile [8] çalışmasında önerilen yöntemin damga oluşturulması açısından zamansal karşılaştırılması

[8] çalışmasında önerilen ilişkisel veritabanları için damgalama yöntemi ile tez çalışmasında önerilen ikililerin histogramına dayalı ilişkisel veritabanı damgalama yöntemlerinin genel karşılaştırması Tablo 10'daki gibidir.

Tablo 10. Khan vd. önerdikleri yöntem ile ikililerin histogramına dayalı yöntemin genel karşılaştırması

		[8]	İkililerin Histogramı
Damga üretim zamanı (sn)	6x6	30	22,3
	7x7	37,3	27,2
	10x10	49,7	40,7
Saldırı algılama	Tümü İçin	Tüm veritabanı için genel bozulma oranı	Tüm veritabanı için genel bir bozulma oranı
İşlem yükü / zamanı	Tümü İçin	Rakam, uzunluk ve aralık frekanslarının ayrı ayrı hesaplanması için tüm veritabanının 3 defa taranması	İkililerin frekans değerlerinin hesaplanması için veritabanı yalnızca 1 defa taranır
Saldırı tespiti	Ekleme	Algılanır	Algılanır
	Silme	Algılanır	Algılanır
	Güncelleme	Algılanır	Algılanır
	Sütun Değiştirme	Algılanamaz	Algılanır

2.2. Ayrık Kosinüs Dönüşümü (DCT) Tabanlı İlişkisel Veritabanı Damgalama

Yapılan tez kapsamında önceki bölümde gördüğümüz ikillerin frekans değerlerine dayalı veritabanı damgalamanın yanında daha önce görüntülerin damgalanmasında kullanılan fakat ilişkisel veritabanlarının damgalanması konusunda kullanılmayan DCT damga üretiminde kullanılmıştır [44]. Önerilen bu yöntem önceki bölümlerde ayrıntısı verilmiş olan [9] çalışmasını iyileştirmeyi hedeflemektedir. Bahsedildiği gibi referans alınan yöntem veritabanı satırlarını gruplandırarak her grup için sütun sayısı kadar satır içerecek şekilde bir kare matris elde etmektedir. Daha sonra bu kare matrisin önce determinantı, sonra da diyagonal minörleri hesaplanmaktadır. Her grup için elde edilen determinant ve minör değerleri birleştirilerek ilişkisel veritabanı için bir damga oluşturulmaktadır. Bu yöntemde determinant hesaplanması ve diyagonal minörlerin hesaplanması işlemleri ilişkisel veritabanından damganın elde edilmesi işlem yükünü ve işlem zamanını yüksek oranda arttırmaktadır. Örnek vermek gerekirse sütun sayısı 10 olan bir tablo için 10x10 boyutlarında kare matrisler oluşturulacaktır. Bu büyüklükteki bir kare matrisin önce determinant değerinin sonrasında da diyagonal minör değerlerinin hesaplanması çok fazla işlem maliyeti getirmektedir. Ayrıca elde edilen damganın boyutu da oldukça büyük olmaktadır. Bu dezavantajların önüne geçebilmek için DCT tabanlı bir damgalama yöntemi önerilmiştir. Burada sütun sayısı kadar satır ile kare matrisler oluşturup determinant ve diyagonal minörlerin hesaplanmasının yerine sütun sayısının yarısı kadar satırlar ile gruplar oluşturarak bu gruplar için DCT katsayıları elde edilmiştir.

Önerilen DCT tabanlı yöntemde işlem adımları şöyle olmaktadır:

1) Veritabanı gruplama: İlişkisel veritabanının gruplama işlemi referans alınan [9] çalışmasına benzer şekilde gerçekleştirilmektedir. Fakat tez çalışmasında önerilen bu yöntemde kare matris oluşturmak için gruplara herhangi bir satır ekleme işlemi yapılmamaktadır. Veritabanının gruplara ayrılma işleminin yalancı kod ifadesi aşağıda verilmiştir.

```
Girişler: ilişkisel veritabanı: R, Grup sayısı υ  = ⌈ α / γ ⌉, BA
Çıkışlar: herbirinin uzunluğu γ olan gruplar
begin
   for i=1 to α do
     h_i^r=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
```

Yapılan gruplama işlemi sonucunda elde edilen örnek bir grup da Tablo 11'de görülmektedir.

Tablo 11. DCT tabanlı yöntem için veritabanı gruplanması ile elde edilen örnek tablo

Sütun1	Sütun2	Sütun3	Sütun4	Sütun5	Sütun6	Sütun7	Sütun8	Sütun9	Sütun10
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

2) DCT katsayılarının elde edilmesi: Oluşturulan gruplar için DCT katsayılarını elde edebilmek için (5)'de verilmiş olan formül uygulanmaktadır.

Grupların oluşturulmasından sonra her grup için DCT katsayılarının hesaplanmasını gösteren yalancı kod da aşağıdaki gibidir.

```
Giriş: Grup Gj
Çıkış: DCT Katsayıları Dj
Begin
//birincil anahtara göre artan sırada grubun satırları
//sıralanır
  for j=1 to υ do
    for k=1 to r do   // grubun satır sayısı
      for l=1 to c do // grubun sütun sayısı
        dctDegerikl=DCT(υkl) // υ. grup için DCT uygulaması
      end for
    end for
    if dctDegeri > 0 then
      Dj = Dj || 1
    else
      Dj = Dj || 0
  end for
  return Dj
end
```

Burada grup içerisindeki satırların tümü alınarak DCT uygulamak yerine her grup içerisinde grubun satır sayısı sütun sayısının yarısı olacak şekilde matrisler oluşturuluyor.

Örneğin 1. Adımda elde edilen bir grupta 10 sütun 8 satırlı bir grup elde edilmiş ise bu grup 5x10 ve 3x10 boyutlarında iki matris olarak ele alınır ve ede edilen matrislerin DCT katsayıları hesaplanır. Tablo 11'de birinci işlem adımı olan veritabanı gruplandırma ile elde edilmiş grubun iki ayrı matrise dönüştürülmüş hali Tablo 12 ve Tablo 13'de verilmiştir.

Tablo 12. DCT tabanlı yöntem için oluşturulan grubun ilk kısmı

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

Tablo 13. DCT tabanlı yöntem için oluşturulan grubun ikinci kısmı

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

Birinci adımda oluşturulmuş olan 8 satır 10 sütunlu grup Tablo 12 ve Tablo 13'de de görüldüğü gibi 5x10 ve 3x10 boyutlarında iki matris haline getirilmiştir.

Oluşturulmuş olan birinci grubun birinci ve ikinci kısımları için DCT katsayılarının hesaplanmış hali Tablo 14 ve Tablo 15'da gösterilmektedir.

Tablo 14. DCT tabanlı yöntem için oluşturulan grubun birinci bölümünün DCT katsayıları

247,3	18,9	5,3	38,1	13,0	-32,6	44,5	1,5	31,6	14,0
29,6	26,1	-47,8	7,0	-39,2	-39,9	-23,9	28,0	20,0	4,7
4,8	-10,8	0,1	-27,5	7,4	9,4	4,5	10,5	20,5	18,4
-6,3	-31,6	-83,2	17,6	-4,5	19,4	-13,2	-29,2	-12,7	21,9
-7,5	9,9	1,1	54,6	-0,9	16,0	14,9	-0,4	-23,9	-2,9

Tablo 15. DCT tabanlı yöntem için oluşturulan grubun ikinci bölümünün DCT katsayıları

219,4	-13,3	74,2	47,1	70,4	22,2	19,3	45,9	-26,6	-18,0
-40,0	42,7	-18,1	43,8	-7,6	-17,6	11,2	-8,35	8,0	-1,5
-12,5	35,6	-12,3	30,4	-30,0	7,6	0,9	-21,2	-12,9	8,2

3) Tablo 14 ve Tablo 15'de görüldüğü gibi gruplar için hesaplanan DCT katsayıları float sayılardır ve negatif veya pozitif olabilmektedir. Önerilen yöntemde negatif sayılar için 0, pozitif sayılar için ise 1 olarak ele alınmıştır. Bu değerler birleştirilerek grubun damgası elde edilir. Grup damgasının elde edilmesini gösteren yalancı kod ifadesi de aşağıdaki gibidir.

```
Girişler: Gruba ait DCT Katsayıları Dj
Çıkışlar: Grup Damgası Wj
begin
   for k=1 to r do    // grubun satır sayısı
      for l=1 to c do  // grubun sütun sayısı
         Wj = Wj || Dkl // j. grup için damganın elde edilmesi
      end for
   end for
   return Wj
end
```

İlgili grubun damgasının oluşturulmasında DCT katsayılarının negatif veya pozitif olmasına göre hücre değeri 0 veya 1 değerine dönüştürülmektedir. Bu dönüşüm sonucunda Tablo 14 ve Tablo 15'de verilen DCT katsayıları Tablo 16 ve Tablo 17'daki gibi olmaktadır.

Tablo 16. Birinci kısmı için DCT katsayıları üzerinden ikili sayıya dönüşüm sonucu oluşan değerler

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 17. İkinci kısım için DCT katsayıları üzerinden ikili sayıya dönüşüm sonucu oluşan değerler

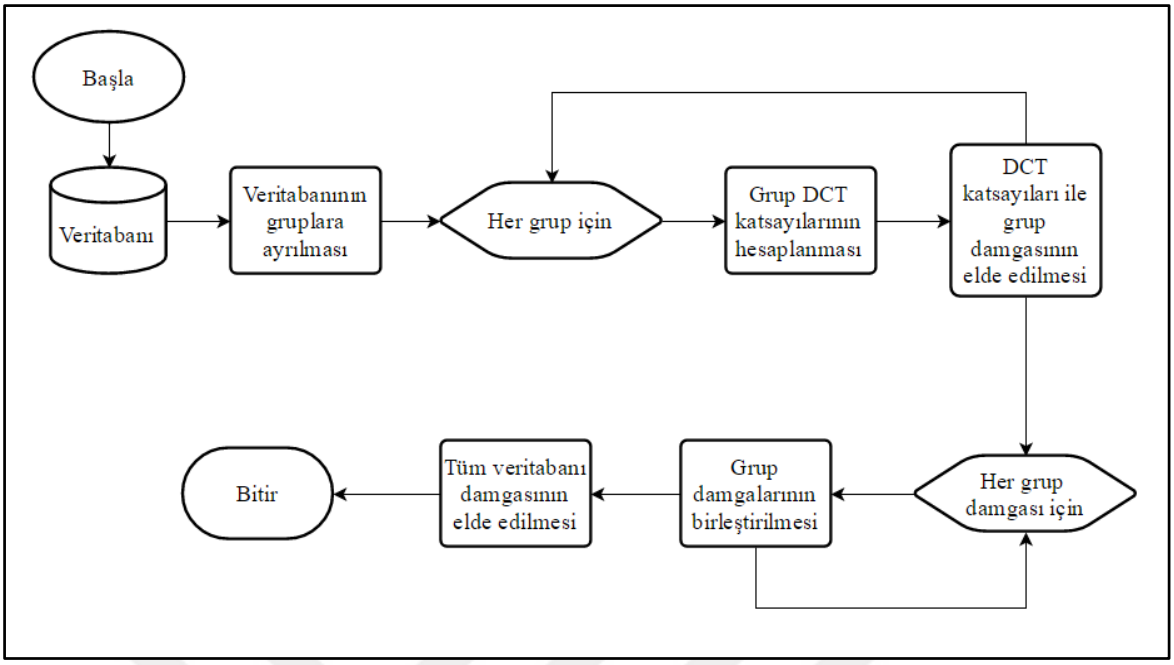
1	0	1	1	1	1	1	1	0	0
0	1	0	1	0	0	1	0	1	0
0	1	0	1	0	1	1	0	0	1

Birinci kısım ve ikinci kısım için üretilen ikili sayılar birleştirilerek ilgili grubun damgası oluşturulmaktadır.

4) Son olarak tüm gruplar için yapılan bu işlemler tamamlandıktan sonra bütün grupların DCT katsayıları ile elde edilmiş olan damgaları birleştirilir ve tüm veritabanı için bir damga oluşturulmuş olur. Veritabanının tümü için bir damganın elde edilmesi ve bu damganın sertifikalandırılarak saklanmasını gösteren yalancı kod aşağıdaki gibidir.

```
Giriş: R, GA, υ
Çıkış: Sertifikalandırılmış damga
begin
//Veritabanının gruplara ayrılması, Grup damgalarının elde
edilmesi
W_R = W1||W2||W3||…||Wυ
E_WR= Şifrele(W_R||GA)
W_C = E_WR|| K_ID || UTC
Return W_C
end
```

DCT tabanlı ilişkisel veritabanı damgalamanın işlem adımları yukarıdaki gibi olup bu işlemlere ait akış diyagramı da Şekil 26'deki gibidir.



Şekil 26. DCT tabanlı ilişkisel veritabanı damgalama yöntemi için damganın oluşturulması için akış diyagramı

DCT tabanlı veritabanı damgalama yönteminde veritabanı damgasının oluşturulma adımlarını gördükten sonra önerilen bu yöntemin saldırılara karşı nasıl dayanıklılık sağladığını incelemek amacıyla yapılan test sonuçları bir sonraki bölümde verilmiştir.

2.2.1. DCT Tabanlı Damgalama ile Saldırı Algılanması

[9] çalışmasında önerilen yöntemin güçlü yanını aynı determinant değerini veren bir matrisin oluşturulabilmesinin çok zor olması olarak ifade edilmektedir. Sütun sayısını 10 olduğunu düşünürsek 10x10 boyutlu bir matrisin değerlerini değiştirerek yine aynı determinant sonucunu verebilecek bir matris elde etmek veya boyutlar bu kadar büyük olmasa da aynı determinant sonucunu verecek iki farklı matris oluşturmak oldukça zordur. Bu sebepten dolayı da önerilen yöntem güçlü olmasını böyle açıklamaktadır. Fakat yöntem her ne kadar güçlü olsa da işlem zamanı ve işlem yükü açısından dezavantajları mevcuttur.

İşlem zamanı ve işlem yükü dezavantajlarının yanında saldırı yapılmış grubun tespitinde [9] çalışması her grup için sütun sayısı kadar satır ile matris oluşturulduğundan saldırı tespitindeki satır aralığı sütun sayısı kadar olmaktadır. Yani 10 sütunlu bir veritabanı tablosu için üretilen gruplar 10 veya 10'un katı olacak şekilde satır içerdiklerinden saldırının

o grup için seçilen 10 satırdan birine veya birkaçına yapıldığı tespit edilir. Önerilen DCT tabanlı ilişkisel veritabanı damgalama yönteminde ise üretilen gruplar sütun sayısının yarısı kadar satır içerdiklerinden dolayı saldırının algılanma aralığı sütun sayısının yarısına düşmektedir. Örnek vermek gerekirse yine 10 sütunlu bir veritabanı tablosu üzerinde gruplar yapılırken 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.

Genel olarak ifade etmek gerekirse gruplandırma sonucunda N satır, M sütun sayısına sahip bir grup elde edilmiş olsun. Tez çalışmasında önerilen DCT tabanlı yönteme 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 DCT katsayılarının hesaplanması işlemi gerçekleştirilmektedir.

Veritabanına yapılacak olan daha önceki bölümlerde verilmiş olan saldırı şekillerinde önerilen DCT tabanlı ilişkisel veritabanı damgalama yöntemi ile [9]'daki yöntem %100 başarılı olmaktadır. Uygulanan saldırılar olarak [8] yönteminde yapıldığı gibi satır ekleme, satır silme, satır güncelleme ve sütun değiştirme saldırılarıdır. Yapılan karşılaştırmalara ilişkin sonuçlar Tablo 18'de görülmektedir.

Tablo 18. DCT tabanlı damgalama yöntemi ile [9] yönteminin saldırılara karşı dirençlerinin karşılaştırılması

Ekleme, Silme, Güncelleme Oranı	[9]			Önerilen Çalışma		
	Ekleme	Silme	Güncelleme	Ekleme	Silme	Güncelleme
%10	Var	Var	Var	Var	Var	Var
%30	Var	Var	Var	Var	Var	Var
%50	Var	Var	Var	Var	Var	Var
%70	Var	Var	Var	Var	Var	Var
%90	Var	Var	Var	Var	Var	Var

Tablo 18'de de görüldüğü gibi önerilen DCT tabanlı yöntem ile [9] yöntemlerinin ekleme, silme, güncelleme veya satır değiştirme saldırıları açısından sağladıkları direnç aynıdır. [9] çalışmasında yöntemin güçlülüğü olarak aynı determinant sonucunu verecek farklı iki matrisin elde edilme zorluğu gösterilmişti. Yapılacak olan herhangi bir saldırıda tekrar aynı determinant sonucuna ulaşılamayacağından saldırıların tespiti yapılmaktadır. Önerilen yöntemde de yine aynı şekilde elde edilen gruplar üzerinde DCT katsayıları hesaplandığından yapılacak olan satır ekleme, satır silme, satır güncelleme veya sütun değiştirme gibi saldırılar üretilecek olan DCT katsayılarını da değiştirecektir. Bu değişim sonucunda saldırı yapıldığı ve veritabanının bütünlüğünün bozulduğu tespit edilecektir.

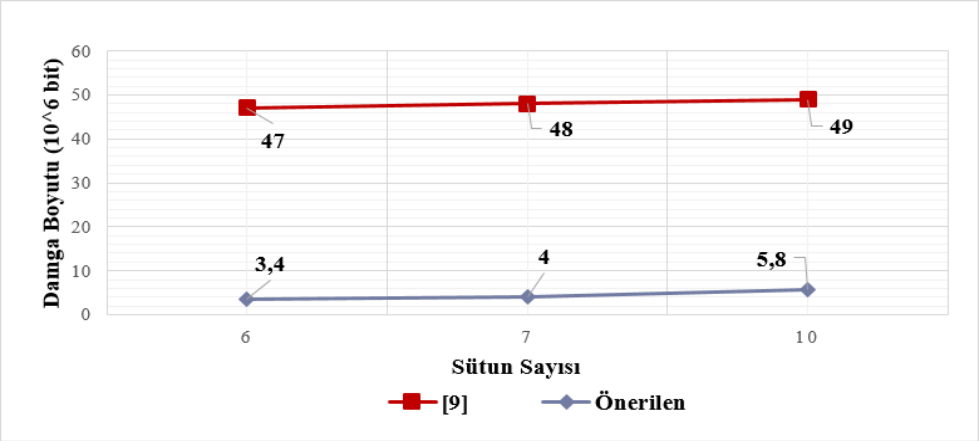
Saldırıların algılanması açısından bir başka avantaj ise şöyle ifade edilebilir. [9] çalışmasında kare matrisler elde etmek için gruplara aynı satırı içermeyecek şekilde ilk grubun ilk satırından başlanarak sütun sayısının tam katı olana kadar satır eklemeleri yapılmaktadır. Fakat bu yöntem bir sorunu da beraberinde getirmektedir. Eğer yapılan saldırı sütun sayısının tam katı olması amacıyla diğer bir gruba eklenen satır üzerinde yapılmışsa, saldırı algılama sürecinde hem satırın bulunduğu gruba hem de satırın eklenmiş olduğu gruba saldırı gerçekleşmiş gibi algılanır. Bu da saldırının algılanmasında aslında saldırı olmamış bir grubu da saldırı yapılmış gibi gösterir. DCT tabanlı önerilen yöntemde ise oluşturulan gruplar üzerinde herhangi bir satır ekleme gerçekleştirilmediği için saldırı hangi grup üzerine yapılmışsa sadece o grup belirlenir. Bir grupta bulunan satıra gerçekleştirilen saldırı başka bir gruptaki algılama işlemini etkilemez. Her grup kendi içerisinde bağımsız olarak ele alınır.

Önerilen yöntemin avantajı ise işlem zamanı ve işlem yükü göz önüne alındığı zaman ortaya çıkmaktadır.

2.2.2. DCT Tabanlı Damgalama Yönteminin Katkıları

Önerilen DCT tabanlı yöntemi ilk olarak elde edilen damga boyutu dikkate alınarak değerlendirilmiştir. Bu değerlendirme sonucunda DCT tabanlı yöntemin saklanacak olan damga boyutunu önemli ölçüde iyileştirdiği görülmektedir. Yapılan çalışmada [9] ve önerilen yöntem sonucunda oluşturulan veritabanı damga bilgisinin boyut olarak karşılaştırılması yapılmıştır. Yapılan karşılaştırma sonucunda [9] çalışmasında önerilen determinant ve diyagonal minörlerin hesaplanmasıyla elde edilen damga bilgisinin tez kapsamında yapılan DCT tabanlı yöntemin elde ettiği damga bilgisinden ortalama 10 kat

daha büyük boyutta olduğu gözlemlenmiştir. İki çalışmadan elde edilen damgaların boyutları 6, 7 ve 10 sütun sayılı farklı veritabanları üzerinde hesaplamalar yapılarak karşılaştırılmıştır. Bu karşılaştırmaya ait veriler Şekil 27'de görülmektedir.



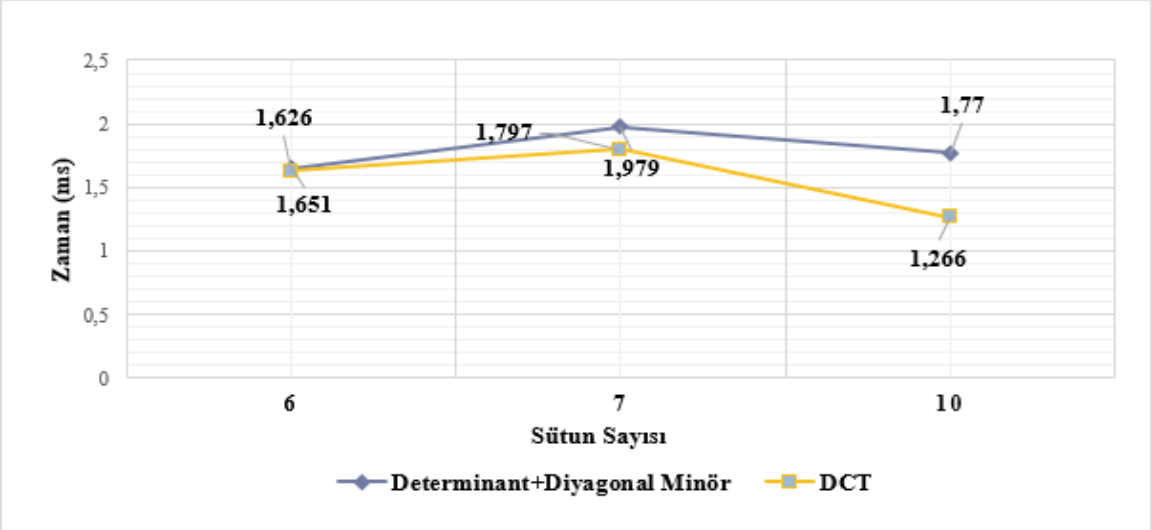
Şekil 27. DCT tabanlı yöntem ile [9] çalışmasının elde edilen damga boyutu olarak karşılaştırılması

Elde edilen damganın boyutlarının karşılaştırılmasının ardından DCT kullanılması [9] çalışması ile işlem zamanı ve işlem yükü olarak da karşılaştırılmıştır. Bu karşılaştırma sonucunda da DCT tabanlı ilişkisel veritabanı damgalama yönteminin saldırıların algılanmasında önceki bölümde bahsedildiği gibi aynı başarıyı vermekle beraber gerek damganın oluşturulması gerekse saldırıların algılanması aşamalarında işlem zamanının ve işlem yükünün azaltılmasında büyük avantaj sağlamaktadır. Önerilen DCT tabanlı yöntem özellikle bu noktada öne çıkmaktadır ve ilişkisel veritabanlarının damgalanmasında kullanılmadığı için de yeni bir bakış açısı getirmektedir.

İşlem zamanı ve işlem yükü açısından da elde edilen damga boyutlarının karşılaştırmasında olduğu gibi 6, 7, 10 sütunlu veritabanı tablolarında karşılaştırmalar yapılmıştır. Bu karşılaştırma için grafiği görmeden önce örnek ile yapılacak olan işlemlerden bahsedelim. Örneğin 10 sütunlu bir veritabanı tablosu üzerinde damgalama işlemi yapılacak olsun. Veritabanının gruplandırılması yapılmış ve elde edilen grupta da 20 satır olsun. Burada öncelikle [9] çalışması açısından yapılacak işlemlere bakalım. Determinant ve diyagonal minörlerin hesaplanması yapılacak ve sütun sayısı 10 olduğu için 10x10 boyutlarında 2 adet matris elde edilecektir. Bu matrislerin ilk olarak önceden de algoritması ve yalancı kodu verildiği gibi determinantı ardından da diyagonal minörleri hesaplanacaktır.

Bu grup için 2 adet determinant ve diyagonal minör hesaplanıp birleştirilecek ve gruba ait damga elde edilmiş olacaktır. Tez kapsamında önerilen DCT tabanlı yöntem ile grup damgasının üretilmesi işleminde ise sütun sayısı 10 olduğu için sütun sayısının yarısı kadar satır yani 5 satır ile matris oluşturulacaktır ve buradan 5x10 boyutlarında 4 adet matris karşımıza çıkacaktır. Fakat burada her matris için sadece DCT katsayılarının hesaplanması işlemi yapılacaktır.

[9] çalışmasında ilişkisel veritabanından damganın elde edilmesi işleminin temeli olan determinant ve diyagonal minör hesaplanması ile önerilen DCT tabanlı yöntemin ilişkisel veritabanından damganın elde edilmesi işleminin temelini oluşturan DCT katsayılarının hesaplanmasının harcamış olduğu zamanlar açısından karşılaştırılması Şekil 28'de verilmiştir.

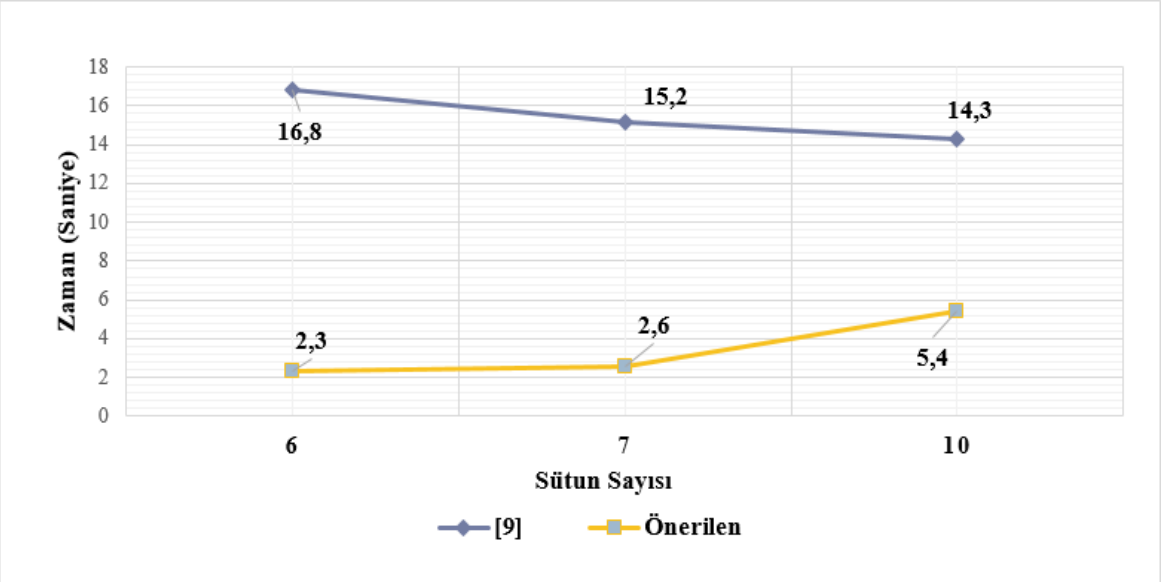


Şekil 28. Determinant ve minör uygulanması ile DCT uygulanmasının zamansal karşılaştırılması

Şekil 28'de görülen bu karşılaştırma işlemi, [9] yönteminin damga bilgisi üretmek için kullanmış olduğu determinant ve diyagonal minörlerin hesaplanması ile önerilen çalışmada kullanılan DCT katsayılarının hesaplanması işlemleri veritabanından bağımsız olarak değerleri rasgele oluşturulmuş matrisler üzerinden hesaplanması ile yapılmıştır. Yapılan bu karşılaştırmada 10x10 boyutlarında bir matris üzerinden örnek vermek gerekirse [9] çalışması gereğince rasgele üretilmiş olan 10x10 boyutlarında matris üzerinde önce determinant hesaplaması ardından da diyagonal minörlerinin hesaplanması işlemleri

gerçekleştirilecektir. Bu iki işlemin tüketmiş oldukları zamanların toplamı alınmıştır. Diğer taraftan önerilen yöntemde determinant ve diyagonal minörlerin hesaplanması yerine sütun sayısının yarısı kadar olacak şekilde 10x10 boyutlarındaki matris 5x10 boyutlarında iki ayrı matrise ayrılarak DCT katsayılarının hesaplanması yapılmaktadır. İki yöntem için yapılan bu işlemlerin sonuçları hesaplanarak Şekil 28'deli değerler elde edilmiştir. Şekilden de görüldüğü gibi sütun sayısı arttıkça DCT daha da avantajlı hale gelmektedir.

Determinant ve minör hesaplama ile DCT hesaplama arasındaki zamansal farkı gördükten sonra yukarı verilen örnek göz önünde bulundurularak yapılan karşılaştırmalar sonucunda önerilen DCT tabanlı yöntem ile [9] çalışmasında önerilen yöntemin karşılaştırmaları Şekil 29'da yapılmıştır. Burada da yine karşılaştırma yapılırken 6, 7 ve 10 sütunlu veritabanları üzerinden işlemler gerçekleştirilmiştir.



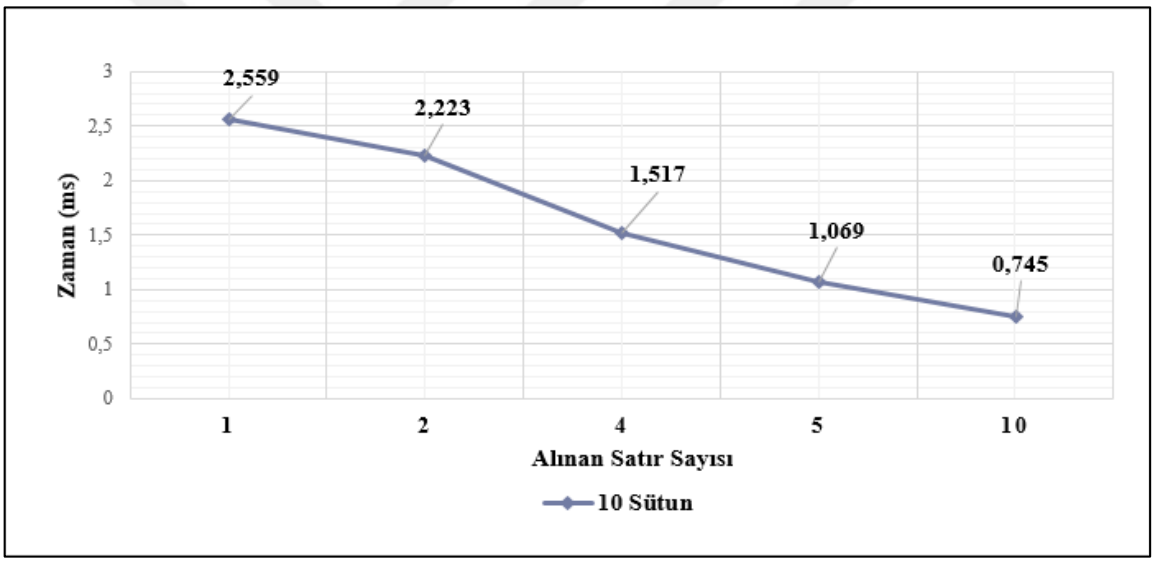
Şekil 29. DCT tabanlı yöntem ile [9] yönteminin veritabanı damga bilgisinin oluşturulması açısından zamansal karşılaştırılması

Şekil 29'dan de görüldüğü gibi DCT tabanlı ilişkisel veritabanı damgalama yöntemi zamansal açıdan önemli bir iyileştirme sunduğu görülmektedir.

Tez kapsamında önerilen DCT tabanlı ilişkisel veri damgalama yönteminde üretilen gruplar için damga oluşturulurken sütun sayısının yarısı kadar satırlar kullanılmakta olduğu yukarıda pek çok yerde vurgulanmıştır. Bu seçim saldırıların algılanmasında ve saldırının yapılmış olduğu yerin belirlenmesinde [9] çalışmasına göre daha küçük aralık bir aralık

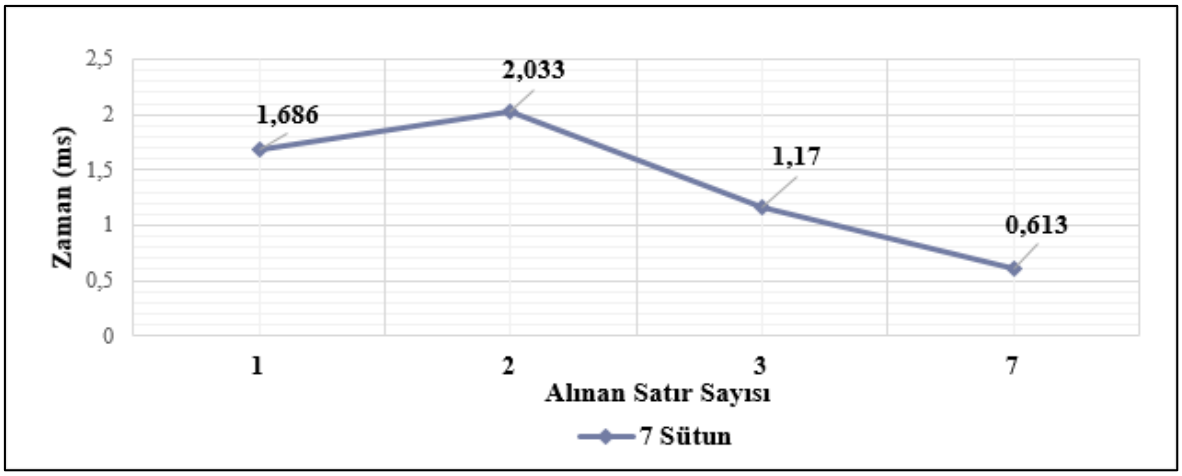
sunmaktadır. DCT işlemleri gerçekleştirilirken alınan sütun sayısının yarısı değeri rasgele seçilmiş bir değer olmayıp yapılan deneysel sonuçlar sonucunda elde edilmiştir. Aşağıda karşılaştırmaları grafikler ile gösterilecek olan bu deneylerde üretilen grup için çeşitli satır sayıları ile DCT uygulanarak 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 DCT uygulanan matrislerin tükettiği zaman artmaktadır. Fakat yapılan çalışmada saldırı tespit aralığında da iyileştirmeye gidilmek istendiğinden dolayı DCT uygulanacak olan matrisin satır sayısı, sütun sayısının yarısı olarak belirlenmiştir.

10 sütunlu bir tablo için 1, 2, 4, 5 veya 10 satır seçildiği takdirde ortaya çıkan zamansal değişimler Şekil 30'da görülmektedir.



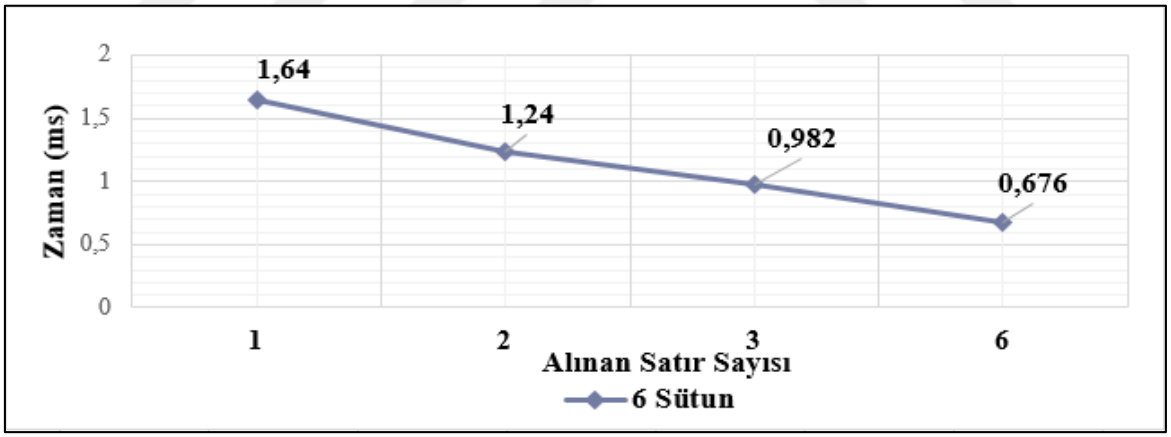
Şekil 30. DCT uygulanması için 10 sütunlu bir veritabanının grup büyüklüğüne bağlı zamansal karşılaştırması

Aynı karşılaştırma sütun sayısının 7 olduğu durum için de yapılmıştır. Yapılan karşılaştırma sonucunda ortaya çıkan zamansal değerler Şekil 31'de verilmiştir.



Şekil 31. DCT uygulanması için 7 sütunlu bir veritabanının grup büyüklüğüne bağlı zamansal karşılaştırması

Sütun sayısının yarının seçilmesi için son yapılan deneysel çalışmada 6 sütun üzerinden zamansal karşılaştırmalar yapılmıştır. Yapılan karşılaştırmalar sonucunda elde edilen zamansal değerler Şekil 32'de verilmektedir.



Şekil 32. DCT uygulanması için 6 sütunlu bir veritabanının grup büyüklüğüne bağlı zamansal karşılaştırması

Özet olarak vermek gerekirse [9] çalışması ve DCT tabanlı önerilen çalışma Tablo 19'daki gibidir.

Tablo 19. DCT tabanlı yöntem ile [9] yöteminin genel karşılaştırılması

		Camara vd.	**DCT Tabanlı**
Damga boyutu	6x6	~47x10^6 bit	~3.4x10^6 bit
	7x7	~48x10^6 bit	~4x10^6 bit
	10x10	~49x10^6 bit	~5.8x10^6 bit
Saldırı yapılan grubun doğru tespiti	Tümü İçin	Kare matris oluşturmayan gruplara ekleme yapıldığı için, eksik olan gruba eklenen satıra saldırı olmuşsa hem eklendiği grup hem de satırın bulunduğu kendi grubuna saldırı olmuş gibi algılanır.	Herhangi bir ekleme yapılmadığı için bir gruba olmuş olan saldırı başka grupları etkilemez.
İşlem yükü / zamanı	Tümü İçin	Matris üzerinde determinant ve diyagonal minör hesaplamanın getirdiği işlem yükü fazla.	İşlem yükü sadece DCT uygulanması
Hata tespit aralığı	Tümü İçin	Sütun sayısı kadar boyutlu bir kare matris üzerinden damgalar hesaplandığından hata satır aralığımız minimum sütun sayısı kadar olacaktır.	Matrisler oluşturulurken sütun sayısının yarısı kadar satır alacağımızdan hata aralığı sütun sayısının yarısına düşüyor.
Güçlülük	Tümü İçin	Güçlülük olarak aynı determinant ve değerlerine sahip iki ayrı matris oluşturulmasının çok zor olması	DCT katsayıları hesaplanırken matrisin tüm elemanları kullanıldığı için herhangi birindeki değişiklik katsayı da değiştirir.
Saldırı tespiti	Ekleme Saldırısı	Ekleme saldırısı grup sayısını değiştirecek kadar ise orijinal veritabanı ile şüpheli veritabanının grup sayılarının karşılaştırılması ile kolaylıkla tespit edilir.	Aynı şekilde grup sayısını değiştirecek kadar satır eklemesi yapılmışsa grup sayılarının karşılaştırılması ile saldırı kolaylıkla tespit edilebilir.
	Silme Saldırısı	Silme saldırısı da ekleme saldırısında olduğu gibi grup sayılarının karşılaştırılması ile kolaylıkla tespit edilebilir.	Silme saldırısı da ekleme saldırısında olduğu gibi grup sayılarının karşılaştırılması ile kolaylıkla tespit edilebilir.
	Güncelleme Saldırısı	Veritabanındaki değerlerin değiştirilmesi durumunda değiştirilen veri kendi grubu içerisinde kolaylıkla tespit edilebilir. Bunun yanında eğer değişime uğramış satır başka bir gruba tamamlama için eklenmişse o zaman saldırı olmamış grup da saldırı olmuş gibi algılanır.	Diğer yöntemde olduğu gibi veritabanı değerlerinde değişiklik olduğu zaman grup DCT katsayıları farklı çıkacağından kolaylıkla tespit edilebilir. Bunun yanında diğer yöntemdeki gibi tamamlama işlemi yapılmadığı için değişiklik olan satır sadece kendi grubunu etkiler, diğer gruplara herhangi bir etki etmez.

3. SONUÇLAR

Yapılan tez çalışmasında ilişkisel veritabanlarının damgalanması ve bütünlük kontrolü için iki farklı yöntem önerilmektedir. Bu yöntemler veritabanı üzerinde herhangi bir değişiklik yapmayarak bozulmadan bağımsız bir sıfır damgalama sunmaktadır. Veritabanı bilgileri kullanılarak üretilen ve saklanan damga daha sonra veritabanı üzerinde herhangi bir bozulma şüphesi olduğu zaman veritabanının bütünlüğünün test edilmesi için kullanılmaktadır. Önerilen yöntemler damga boyutu, işlem zamanı gibi durumların yanında satır ekleme, satır silme, satır güncelleme ve sütun değiştirme gibi ilişkisel veritabanları üzerinde yapılabilecek olan temel bazı saldırılar açısından da test edilmiştir. Bu testler, önerilen yöntemlerin referans alınan yöntemlere kıyasla önemli bir iyileştirme sunduğunu göstermektedir. Çalışmada önerilen yöntemlerden ilki veritabanı tablolarından elde edilen ikililerin histogramına dayalı bütünlük kontrolü sağlamaktadır. Bu yöntemde referans alınan [8] çalışması iyileştirilmiştir. İkililerin oluşturulması, ikililerin frekans değerlerinin hesaplanması ve hesaplanan frekans değerlerinin birleştirilerek damganın oluşturulması olmak üzere üç işlem adımından oluşan yöntem referans çalışmaya ek olarak sütun değiştirme saldırılarının algılanmasını da sağlamaktadır. Bu saldırının algılanması için veritabanındaki değerlerin değiştirilmesine gerek yoktur. Sütunların yerlerinin değiştirilmesi bile saldırının algılanmasına olanak sağlamaktadır. Önerilen yöntem veritabanındaki verileri her hücrede bulunan değeri tek tek ele alarak ikililere ayırır. Ardından ikililere bulunduğu sütun numarasını ekler ve frekansını bu şekilde belirler. Eklenen sütun numaraları sayesinde sütun değiştirme saldırılarına karşı dayanıklılık da sağlanmış olmakta ve bunun yanında işlem zamanı ve üretilen damga boyutunda iyileştirmeler gerçekleştirilmektedir.

Tez kapsamında önerilen yöntemlerden ikincisi ise DCT tabanlı bir ilişkisel veritabanı damgalama yöntemidir. DCT frekans dönüşümü yönteminin ilişkisel veritabanları üzerinde kullanılmamış olması nedeniyle konuya bir yenilik katmaktadır. Bu yöntem önerilirken [9] çalışması referans alınmıştır ve bu çalışma üzerinden yeni bir yöntem geliştirilerek iyileştirmeler gerçekleştirilmiştir. Öncelikle veritabanının gruplandırılmasını yapan bu yöntemde ardından gruplar üzerinde DCT uygulanmaktadır. Damgalama şemasında işlem adımları kısaca şöyle özetlenebilir. Gruplara ayrılan veritabanı üzerinde her grup kendi içerisinde sütun sayısının yarısı kadar satırlar olacak şekilde alt gruplara ayrılır ve her alt grup için DCT uygulanarak DCT katsayıları hesaplanır. DCT katsayıları hesaplanarak alt

grup damgaları üretilmiş olur ve bu alt grup damgalarının birleştirilmesiyle grup damgası üretilir. Veritabanından oluşturulan her grup için bu işlemler gerçekleştirildikten sonra tüm grupların damgaları birleştirilerek veritabanı için tek bir damga üretilmiş olur. Ardından bu damga şifrelenip sertifikalandırılarak veritabanından bağımsız olarak kaydedilir.

Veritabanlarının bütünlük kontrolü için şüpheli veritabanı üzerinde aynı işlemler gerçekleştirilerek bir damga üretilir. Daha sonra bu üretilen damga başta üretilmiş ve saklanmış olan damga ile karşılaştırılarak veritabanının bütünlüğü hakkında karar verilir. Eğer damgalar eşleşiyorsa veritabanı bütünlüğünü koruyor demektir. Fakat orijinal damga ile şüpheli veritabanından üretilen damgalar eşleşmiyorsa yani aynı değillerse o zaman veritabanı üzerinde bir saldırı gerçekleştirildiği ve veritabanının bütünlüğünün bozulduğu söylenebilir.

Çalışma kapsamında önerilen DCT tabanlı yöntem, [9] yöntemi ile yukarıda verilmiş olan kıyaslamalar haricinde saldırının yapıldığı satır aralıklarının tespit edilmesi açısından da kıyaslanabilir. Örneğin 10 sütunlu bir grup için referans alınan çalışmada 10 satır ele alındığı için saldırının tespit aralığı 10 satır olmaktadır. Fakat DCT yönteminde sütun sayısı kadar satır yerine sütun sayısının yarısı kadar satır alınmakta ve saldırının tespit aralığı 5 olarak belirlenmektedir. Yani saldırı gerçekleşmiş satırların tespitinde de 2 kat avantaj sağlamaktadır.

Önerilen çalışmada, test işlemleri, referans alınan çalışmalarda kullanılan 581.012 satır ve birincil anahtar dışında 10 tam sayı (integer) sütundan oluşan veritabanı üzerinde yapılmıştır. Ayrıca test işlemlerini genişletip farklı sütun sayıları ile de karşılaştırmalar gerçekleştirilmiştir. Tablolardaki kayıt değerleri 0-1000 arasında rasgele değerlerden oluşturulmuştur. Deneysel sonuçların elde edilmesi aşamasında Intel Core i7-4700HQ 2.4GHz işlemci, 16GB DDR3 RAM özelliklerine sahip platformda, veritabanı yönetim sistemi olarak SQL Server 2014 kullanmıştır. Referans alınan iki çalışma için yapılan karşılaştırmalar ve verilen deneysel sonuçların elde edilmesi aşamalarında yöntemlerin kodlanması verilen çalışmalarda verilmiş olan algoritmalar yardımıyla tarafımızdan gerçekleştirilmiş ve tüm testler aynı sistem üzerinde yapılmıştır.

Tez çalışmasında önerilen ikililerin histogramına dayalı ve DCT tabanlı ilişkisel veritabanları için sıfır damgalama şemaları ile [8] ve [9] çalışmalarında önerilen yöntemlerin genel karşılaştırılmaları Tablo 20'de görülmektedir.

Tablo 20. Tez çalışmasında önerilen yöntemler ile [8] ve [9] çalışmasının genel karşılaştırması

		[8]	[9]	**Önerilen İkililerin Histogramı Tabanlı Yöntem**	**Önerilen DCT Tabanlı Yöntem**
Saldırı olan grubun doğru bulunması	Tümü İçin	Grup seviyesinde algılama yok	Kare matris oluşturmak için yapılan eklemeler yanlış grup tespiti ortaya çıkarabilir	Grup seviyesinde algılama yok	Herhangi bir ekleme yapılmadığı için saldırı yapılan grup doğru tespit
İşlem Yükü	Tümü İçin	Rakam, uzunluk ve aralık frekanslarının hesaplanması	Determinant ve diyagonal minör hesaplanması	İkililerin frekanslarının hesaplanması	DCT katsayılarının elde edilmesi
Hata Aralığı	Tümü İçin	Genel bir bozulma oranı	Sütun sayısı kadar satır için	Genel bir bozulma oranı	Sütun sayısının yarısı kadar satır için
Saldırı Tespiti	Ekleme	Algılanır	Algılanır	Algılanır	Algılanır
	Silme	Algılanır	Algılanır	Algılanır	Algılanır
	Güncelleme	Algılanır	Algılanır	Algılanır	Algılanır
	Sütun değiştirme	Algılanamaz	Algılanır	Algılanır	Algılanır

4. ÖNERİLER

Çalışma kapsamında yapılan testler ve bu testler sonucundaki karşılaştırmalar için önceki bölümde de belirtildiği gibi referans alınan çalışmalarda kullanılan veritabanı temel alınarak ve veritabanının sütun sayıları değiştirilerek 3 farklı şekilde karşılaştırmalar yapılmıştır. Veritabanı genişletilerek daha fazla sütun içeren veritabanlarında da performans testleri yapılarak karşılaştırılabilir.

Önerilen yöntemde kullanılan veritabanı sayısal değerler içermektedir. Aynı yöntem sayısal olmayan veriler için de uygulanıp sonuçları analiz edilebilir. Böylece aynı yöntemin sayısal ve sayısal olmayan veriler üzerindeki performansı da karşılaştırılabilir.

5. KAYNAKLAR

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

2. Agrawal, R. ve Kiernan, J., Watermarking Relational Databases, Proceedings of the 28th international conference on Very Large Data Bases, VLDB Endowment, Hong Kong, 155–166, 2002.

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

4. Gupta, G. ve Pieprzyk, J., Reversible and blind database watermarking using difference expansion, Proceedings of the 1st international conference on Forensicapplications and techniques in telecommunications, information, and multimedia and workshop, Adelaide, Australia, 24, 2008.

5. Farfoura, ME. ve Horng, S-J., A novel blind reversible method for watermarking relational databases, 2010 International Symposium on Parallel and Distributed Processing with Applications (ISPA), IEEE, Taipei, Taiwan, (2010) 563–569.

6. Chang, C-C,, Nguyen, T-S, ve Lin, C-C., A blind Reversible robust watermarking scheme for relational databases. The Scientific World Journal, 2013.

7. Zhang, Y., Yang, B. ve Niu, X-M., Reversible Watermarking for relational database authentication. Journal of Computers, 17,2 (2006) 59–66.

8. Khan, A. ve Husain, SA., A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations. The Scientific World Journal (2013) 16.

9. Camara. L., Li, J., Li, R. ve Xie W, Distortion-Free Watermarking Approach for Relational Database Integrity Checking, Mathematical Problems in Engineering, (2014) 1–10.

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

11. Li, Y., Guo, H., ve Jajodia, S., Tamper detection and localization for categorical data using fragile watermarks, in Proceedings of the 4th ACM Workshop on Digital Rights Management (DRM ’04), 73–82, ACM Press, Washington, DC, USA, 2004.

12. Kamel, I., A schema for protecting the integrity of databases, Computers and Security, 28, 7 (2009) 698–709.

13. Bhattacharya, S. ve Cortesi, A., A distortion free watermark framework for relational databases, in Proceedings of the 4th International Conference on Software and Data Technologies (ICSOFT ’09), 229–234, INSTICC Press, Sofia, Bulgaria, 2009.

14. Kamran, M., Suhail, S. ve Farooq, M., A robust, distortion minimizing technique for watermarking Relational databases using once-for-all usability constraints. IEEE Transactions on Knowledge and Data Engineering, 25,12 (2013) 2694–2707.

15. Guo, H., Li, Y. ve Jajodia, S., Chaining watermarks for detecting malicious modifications to streaming data, Information Sciences, 177, 1 (2007) 281–298.

16. Shehab, M., Bertino, E. ve Ghafoor, A., Watermarking Relational databases using optimization based techniques, IEEE Transactions on Knowledge and Data Engineering, 20, 1 (2008) 116–129.

17. Farfoura, M. E., Horng, S. J., Lai, J. L., Run, R. S., Chen, R. J. ve Khan, M. K., A blind reversible method for Watermarking relational databases based on a time-stamping protocol, Expert Systems with Applications, 39, 3 (2012) 3185–3196.

18. Codd, E.F. A relational model of data for large shared data banks. Communications of the ACM, 13, 6 (1970) 377-387.

19. Coatrieux, G., Chazard, E., Beuscart, R. ve Roux, C., Lossless watermarking of categorical attributes for verifying medical data base integrity, IEEE Annual International Conference of the Engineering in Medicine and Biology Society, EMBC, IEEE, Boston, (2011), 8195–8198.

20. Zhang, L., Gao, W., Jiang, N., Zhang, L. ve Zhang, Y., Relational databases watermarking for textual and numerical data, 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), IEEE, Jilin, China, 1633–1636, 2011.

21. Khanduja, V. ve Verma, O., Identification and proof of ownership by watermarking relational databases. International Journal of Information and Electronics Engineering, 2,2 (2012) 274–277.

22. Franco Contreras, J., Coatrieux, G., Chazard, E., Cuppens, F., Cuppens Boulahia, N. ve Roux, C., Robust lossless watermarking based on circular interpretation of bijective transformations for the protection of medical databases, 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, San Diego, CA, 5875–5878, 2012.

23. Jian, Y., Hongjun, Z., Wenning, H., Gang, C. ve Bin, L., A zero-watermarking algorithm for relational database copyright protection, 2012 IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS), IEEE, Beijing, China, 28–31, 2012.

24. http://www-01.ibm.com/software/data/db2/linux-unix-windows/security Db2 security. 22 Ağustos 2014.

25. http://www.oracle.com/us/support/assurance/overview/index.html Oracle software security assurance. 22 Ağustos 2014.

26. Jawad, K. ve Khan, A., Genetic algorithm and difference expansion based reversible watermarking for Relational databases. Journal of Systems and Software, 86**,**11 (2013) 2742–2753.

27. Kamran, M. ve Farooq, M., An information-preserving watermarking scheme for right protection of EMR systems. IEEE Transactions on Knowledge and Data Engineering; 24**,**11 (2012) 1950–1962.

28. Mitchell, M., An Introduction to Genetic Algorithms. MIT press: Cambridge, Massachusetts. London, England, (1996).

29. Rukhin, A., Soto, J., Nechvatal, J., Smid, M. ve Barker E. A statistical test suite for random and pseudorandom number generators for cryptographic applications. DTIC Document, Technical Report, 2001.

30. Li Z, Liu J. ve Tao W., Robust and reversible Relational database watermarking algorithm based on clustering and polar angle expansion. http://onlinepresent.org/proceedings/vol12_2012/1.pdf 20 Mart 2016

31. Tao, W-C., Li, Z-Y. ve Li, H-F., Reversible and blind database watermark algorithm based on polar angle expansion. Computer Engineering, 22 (2010) 58.

32. Iftikhar, S., Kamran, M. ve Anwar, Z., RRW—A Robust and Reversible Watermarking Technique for Relational Data. Knowledge and Data Engineering, IEEE Transactions on, 27,4 (2014) 1132-1145.

33. Alattar, AM., Reversible watermark using difference expansion of triplets, Proceedings of the 2003 International Conference on Image Processing, 2003. ICIP 2003, IEEE, Barcelona, Spain 2003 Bildiriler Kitabı I:501.

34. Iftikhar, S., Kamran, M. ve Anwar, Z., A survey on reversible watermarking techniques for relational databases. Security and Communication Networks, 8,15, (2015) 2580-2603.

35. Chang, J-N. ve Wu, H-C., Reversible fragile database watermarking technology using difference expansion based on SVR prediction, 2012 International Symposium on Computer, Consumer and Control (IS3C), IEEE, Taichung, Taiwan, 690–693, 2012.

36. Watson, B.A., Image Compression Using the Discrete Cosine Transform, Mathematica Journal, 4,1 (1994), 81-8.

37. Deshpande, A., ve Gadge, J., New Watermarking Technique for Relational Databases" In Emerging Trends in Engineering and Technology (ICETET), 2nd International Conference on, 664-669, 2009.

38. Saxena, V. ve Gupta, J.P., Collision Attack Resilient Watermarking scheme for Colored Images Using DCT, IAENG International journal of Computer Sciences, (2007).

39. M. Atalar, İmge Dizilerindeki Artıkların İşlenmesi, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2008.

40. Odeh, A. ve Al-Haj, A., Watermarking Relational Database System, First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT), 2008.

41. Petitcolas, FA., Watermarking schemes evaluation. IEEE Signal Processing Magazine; 17**,**5 (2000) 58–64.

42. Fabien, A.P., Petitcolas Ross, J., Anderson, and Markus G. Kuhn, Fabien A.P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn, "Attacks on Copyright Marking System." Notes in Computer Science, Portland, Oregon, USA, April, 1998.

43. Jonathan K., Su., Hartung, Frank. ve Girod, B., Spread Spectrum Watermarking: Malicious Attacks and Counterattacks. Telecommunication Laboratory, University of Erlangen Nuremberg, Germany, 1999.

44. Gui, Feng. ve Huang, Xihui., "An improved DCT based zero-watermarking algorithm for text image." Anti-Counterfeiting, Security and Identification (ASID), 2012 International Conference on. IEEE, 2012.

ÖZGEÇMİŞ

Yasin ŞAHİN; 1985 yılında Arsin/Trabzon'da doğdu. İlk ve orta öğretimini Arsin ilçesinde tamamladı. 2004 yılında Sürmene Hasan Sadri Yetmişbir Anadolu Lisesi'nden mezun oldu. 2006 yılında ÖSYS ile yerleştirildiği Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü'nden 2011 yılında mezun oldu. 2011-2012 eğitim-öğretim yılının bahar döneminde Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans programına başlamıştır. Eylül 2013 tarihinden itibaren Gümüşhane Üniversitesi Torul Meslek Yüksekokulu Bilgisayar Teknolojileri Bölümü'nde öğretim görevlisi olarak çalışmaktadır. Yabancı dil olarak İngilizce bilmektedir.