T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İKİ ADIMLI ŞİFRELEME İLE DİNAMİK VERİ KAYIT SİSTEMİ

YÜKSEK LİSANS TEZİ

Zehra Nur KÖSE

Bilgisayar Mühendisliği Anabilim Dalı

TEMMUZ 2023

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İKİ ADIMLI ŞİFRELEME İLE DİNAMİK VERİ KAYIT SİSTEMİ

YÜKSEK LİSANS TEZİ

Zehra Nur KÖSE

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Muhammed Fatih ADAK

Ortak Danışman: Dr. Öğr. Üyesi Mustafa AKPINAR

TEMMUZ 2023

Zehra Nur KÖSE tarafından hazırlanan "İki Adımlı Şifreleme ile Dinamik Veri Kayıt Sistemi" adlı tez çalışması 06.07.2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Bilim Dalı'nda Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Jüri Başkanı :	**Dr. Öğr. Üyesi** Sakarya Üniversitesi	.....
Jüri Üyesi :	**Dr. Öğr. Üyesi** BAE Higher Colleges of Technology	.....
Jüri Üyesi :	**Dr. Öğr. Üyesi** Ankara Sosyal Bilimler Üniversitesi	.....
Jüri Üyesi :	**Dr. Öğr. Üyesi** Sakarya Üniversitesi	.....
Jüri Üyesi :	**Dr. Öğretim Üyesi** Kırıkkale Üniversitesi	.....

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum "İki Adımlı Şifreleme ile Dinamik Veri Kayıt Sistemi" başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

15/06/2023

Zehra Nur KÖSE

TEŞEKKÜR

Yüksek lisans eğitimim boyunca bilgi ve deneyimlerinden yararlandığım, araştırmaların tüm aşamalarında beni yönlendiren danışman hocalarım Dr. Öğr. Üyesi Muhammed Fatih ADAK ve Dr. Öğr. Üyesi Mustafa AKPINAR'a teşekkürlerimi sunarım.

Ayrıca bu süreçte yanımda olan, her türlü desteği sağlayan, öncelikle annem ve babam olmak üzere tüm aileme ve yakınlarıma teşekkürlerimi sunarım.

Bu tez, Sakarya Üniversitesi Bilimsel Araştırma Projeleri Komisyonu (BAPK) tarafından 2022-7-24-149 proje numarası ile desteklenmiştir.

Zehra Nur KÖSE

İÇİNDEKİLER

Sayfa

KISALTMALAR

3DES	:	Üçlü Veri Şifreleme Standardı (Triple Data Encryption Standard)
AES	:	Gelişmiş Şifreleme Standardı (Advanced Encrytion Standard)
API	:	Uygulama Programlama Arayüzü (Application Programming Interface)
ASIC	:	Uygulamaya Özel Tümleşik Devre (Application Specific Integrated Circuit)
ASP	:	Aktif Sunucu Sayfası (Active Server Page)
CBC	:	Şifre Blok Zincirlemesi (Cipher Block Chaining)
CFB	:	Şifre Geri Beslemeli Model (Cipher FeedBack)
char	:	Karakter (Character)
CPU	:	Merkezi İşlemci Birimi (Central Process Unit)
DAPI	:	Veri Koruma Uygulama Programlama Arayüzü (Data Protection API)
DES	:	Veri Şifreleme Standardı (Data Encryption Standard)
DMK	:	Veritabanı Ana Anahtarı (Database Master Key)
ECB	:	Elektronik Kod Defteri (Electronic Code Book)
GB	:	Gigabayt
GHz	:	Gigahertz
GUID	:	Genel Benzersiz Tanımlayıcı (Globally Unique Identifier)
ID	:	Kimlik Numarası (Identification Number)
IDEA	:	Uluslararası Veri Şifreleme Algoritması (International Data Encryption Algorithm)
Int	:	Tamsayı (Integer)
IT	:	Bilişim Teknolojisi (Information Technology)
IV	:	Başlatma Vektörü (Initialization Vector)
KVKK	:	Kişisel Verilerin Korunması Kanunu
MB	:	Megabayt

MS SQL	:	Microsoft Yapılandırılmış Sorgu Dili Sunucusu (Microsoft SQL Server)
nvarchar	:	Ulusal Değişken Karakter (National Variable Character)
OECD	:	İktisadi İş birliği ve Kalkınma Teşkilatı (Organisation for Economic Co-Operation and Development)
OFB	:	Çıktı Geri Beslemeli Model (Outpout FeedBack)
PCBC	:	Yayılımlı Şifre Blok Zincirlemesi (Propagating Cipher Block Chaining)
RAM	:	Rastgele Erişim Belleği (Random Access Memory)
RC2	:	Rivest Şifresi 2 (Rivest Cipher 2)
RC4	:	Rivest Şifresi 4 (Rivest Cipher 4)
RC6	:	Rivest Şifresi 6 (Rivest Cipher 6)
RSA	:	Rivest, Shamir, Adleman
SMK	:	Servis Ana Anahtarı (Service Master Key)
sn	:	Saniye
SQL	:	Yapılandırılmış Sorgu Dili (Structured Query Language)
SSD	:	Katı Hal Sürücü (SSD)
T-SQL	:	İşlem Yapılandırılmış Sorgu Dili (Transact-SQL)
varbinary	:	Değişken İkili Sayı (Variable Binary)
varchar	:	Değişken Karakter (Variable Character)
vb.	:	ve benzeri
XOR	:	eXclusive OR

TABLO LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

İKİ ADIMLI ŞİFRELEME İLE DİNAMİK VERİ KAYIT SİSTEMİ

ÖZET

Günümüz şartlarında gerek devlet kurumlarında gerekse özel kuruluşlarda her gün binlerce kişiye ilişkin çeşitli bilgilere erişilmektedir. Erişilen bu bilgiler, dijital dünyada yaşanan gelişmelerin etkisiyle kolaylıkla aktarılmakta ve işlenmektedir. Kişisel Verilerin Korunması Kanunu (KVKK) ve bilgi güvenliği kapsamında, verilerin iyi bir şekilde korunması, belirli, açık ve meşru amaçlar içerisinde sadece yetkili kişiler tarafından işlenmesi önemlidir. Bu kapsamda, kişisel ve hassas verileri işleyen tüm kurum ve kuruluşlar belirli önlemleri almalıdır. Alınabilecek önlemlerden biri de verileri şifrelemektir. Sistemin güvenlik ve performans gereksinimleri göz önünde tutularak kullanılacak olan şifreleme teknikleri belirlenmelidir. Şifreleme işlemi güvenliği artırırken beraberinde sistemin performansını düşürmektedir. Bu yüzden şifreleme yöntemleri tercih edilirken hem en güvenli hem de en hızlı olan teknik tercih edilmelidir.

Bu çalışmada şifreleme algoritmaları, çalışma mantığı ve aralarındaki farklar incelenmiştir. Aynı zamanda farklı seviyelerde art arda iki kez şifreleme işleminin gerçekleştiği güvenli, dinamik bir veri kayıt sistemi geliştirilmiştir. Geliştirmenin kullanıcı ile etkileşimi sağlanmış olup yapısı, mimarisi, kullanılan algoritmalar ve elde edilen sonuçlar hakkında bilgi verilmiştir.

Uygulama varsayılan olarak iki kez art arda şifreleme yapılacak şekilde ayarlanmıştır. İlk şifreleme işleminde simetrik şifreleme yöntemlerinden olan Gelişmiş Şifreleme Standardı (AES) algoritması tercih edilmiş olup algoritmanın arkayüzde (backend) gerçekleşmesi sağlanmıştır. İkinci şifreleme yönteminde ise asimetrik şifreleme yöntemlerinden olan RSA (Rivest-Shamir-Adleman) algoritması tercih edilmiş ve veritabanı seviyesinde kolon düzeyinde şifreleme işlemi yapılmıştır. Bu işlemler sonucunda açık veriye ulaşmak için önce veritabanı seviyesinde gerçekleşen RSA şifreleme işleminin çözülmesi, ardından farklı seviyede AES algoritmasıyla gerçekleşen şifreli metnin çözülmesi gerekecektir. Bu durumların hepsi güvenliği artırıp, yetkisiz kişilerin açık verilere erişmesini zorlaştıracaktır.

Geliştirilen uygulama vasıtasıyla AES algoritması üzerinde Elektronik Kod Defteri (ECB), Şifre Blok Zincirlemesi (CBC) ve Şifre Geri Beslemeli (CFB) blok modelleri arasındaki performans farkı incelenip, sonucunda en hızlı şifre çözme süresine sahip olanlar sırasıyla ECB, CFB ve CBC şeklinde olmuştur. Bu blok modelleri arasında en güvensiz olan ECB, en yavaş olan ise CBC ‘dir. Bu nedenle üç blok modeli arasında en iyi tercih CFB modeli olacaktır. Tek adımda AES ile şifrelenen ve tek adımda veritabanı seviyesinde kolon düzeyinde RSA algoritması ile şifrelenen verilerin performansı incelendiğinde ise RSA’nın asimetrik şifreleme tekniğini kullanması nedeniyle performansı AES’e oranla daha çok düşürdüğü gözlenmiştir.

DYNAMIC DATA RECORDING SYSTEM WITH TWO STEP ENCRYPTION

SUMMARY

In today's conditions, it has been ensured that almost every transaction made can be reflected in the digital environment. While the digital world is expanding rapidly, the security problems that arise with it are also increasing. Especially the increase in digital users, the fact that some of these users are young children and the fact that a large user group is unaware of the threats and dangers that may occur, and continues to use the digital environment unconsciously causes security problems to increase. Digital transactions should be used with the right technology, for the right purpose and consciously, unauthorized persons should not be able to access this information, and information security should be ensured in this context.

Information security not only prevents unauthorized access to information but also aims to destroy information, change its content and prevent possible damage to information. Another advantage of information security is the protection of privacy.

Data that is important to any person and that is not easily accessible by anyone is defined as sensitive data. Information that enables the physical, cultural, economic or social identity of a natural person to be determined directly or indirectly is defined as personal data. In today's conditions, it is important that sensitive and personal data do not fall into the hands of malicious people or third parties and that the data is delivered to the right recipient without changing the content.

Within the scope of the Personal Data Protection Law (KVKK), the sharing of personal data with third parties without the consent of the person should be prevented. The main purpose of the KVKK is to protect the rights of the person, especially the private life, regardless of religion, language, race or gender, and to prevent any harm that may come. In this context, it is important to protect the data well and process it only by authorized persons for specific, clear and legitimate purposes. Data should be consistent and limited with the purpose for which it is collected and processed, and should be kept for as long as necessary for the purpose for which it is processed. All institutions and organizations that process personal and sensitive data are obliged to implement certain measures. One of these measures is to encrypt data. The encryption process can take place at different levels and with different techniques.

The encryption techniques to be used should be determined by considering the security and performance requirements of the system. It is not possible to talk about one hundred percent reliability in the digital world but it should be aimed to prevent security vulnerabilities as much as possible, depending on the requirements of the designed system.

While the encryption process increases security, it also reduces the performance of the system. Therefore, when choosing encryption methods, the most secure and fastest

technique should be preferred. Some of these techniques are symmetric encryption and asymmetric encryption methods. Symmetric ciphers are generally faster than asymmetric ciphers, but they also have more security vulnerabilities than asymmetric ciphers. Symmetric ciphers can use either block cipher or stream cipher. In blocking cipher technique, Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB) etc. block models can be used.

In block and stream ciphers, if the same plaintexts are encrypted with the same key, the same ciphertext appears every time. In order to prevent this situation, the initialization vector (IV) is used in both encryption techniques. Although the initialization vector is used in many models of the block cipher method, it is not used in the ECB model. Therefore, it has more security vulnerabilities than other models such as ECB, CBC and CFB. If the same plaintext is encrypted with the same key in ECB, attackers can identify from similar or same traffic flow.

Since only one secret key is used in symmetric encryption technique and secure key distribution is difficult, it is more correct to use asymmetric encryption technique in some systems. Especially as the network grows, the number of keys increases, which may cause capacity problems. In the asymmetric encryption technique, these problems are solved since the public key is accessible to everyone, on the other asymmetric passwords also work slower than symmetric passwords.

The encryption process can take place at different levels and at different layers. Encryption can be done at the operating system layer, server layer or database layer. Encryption at the database layer can be done at the column level, file level or database level. When choosing encryption method, how secure the system should be and the reflection of encryption processes on performance should be considered. If encryption is done at the database layer, the decryption process must also take place at the database layer.

Even if unauthorized persons have access to the data as a result of encryption, they will not be able to see the open text directly. In order to be able to read the data, it will also need to take care of the decryption process. This will also increase security.

In this study, encryption algorithms, working logic and differences between them are examined. At the same time, a secure, dynamic data recording system has been developed in which encryption is performed twice in succession at different levels. The performance differences between the developed application and the algorithms were analyzed. The interaction of the development with the user was provided and information was given about its structure, architecture, algorithms used and the results obtained.

Depending on the user's request, encryption can be done once or twice in succession. It can be determined at what level and with which technique the encryption process will be performed.

The application is set to encrypt twice in succession by default. In the first encryption process, the Advanced Encryption Standard (AES) algorithm, which is one of the symmetric encryption methods, was preferred and the algorithm was realized on the backend. In the second encryption method, the RSA (Rivest-Shamir-Adleman) algorithm, which is one of the asymmetric encryption methods, was preferred and column-level encryption was performed at the database layer. As a result of this

operation, it will not be possible to directly read/write on the database. When the attacker wants to read any encrypted data kept in the database, the attacker will not be able to access the clear text. He/she will first need to decode the RSA encryption process done at the database layer, then he/she will need to decrypt the ciphertext performed with the AES algorithm at different level. All of these situations will increase security and make it harder for unauthorized persons to access open data.

By means of the developed application, the performance difference between the Electronic Code Book (ECB), Cipher Block Chaining (CBC) and Cipher Feedback (CFB) block models on the AES algorithm was examined, and as a result, the ones with the fastest decryption time were ECB, CFB and CBC, respectively. Among these block models, ECB is the most insecure and CBC is the slowest. Therefore, the CFB model is the best choice among the three block models.

When the performance of the data encrypted with AES in one step and encrypted with RSA at the database layer at the column level in one step is examined, it is observed that RSA reduces the performance more than AES because it uses asymmetric encryption technique.

At the same time, when the performance of the data encrypted in two steps at different levels and with different techniques, and the data that went through a single encryption process in one step, was examined, it was seen that the used algorithm technique rather than the repetition of encryption had more effect on the performance. For example, when the performance of data encrypted with RSA in one step and data encrypted first with AES and then with RSA is examined, it is seen that the decryption times are very close to each other. From here, it is seen that since the AES algorithm is a symmetric method, it does not affect the performance much, even though it be done twice encryption.

1. GİRİŞ

Teknolojinin gelişimiyle birlikte birçok veri dijital ortamda depolanmakta ve işlenmektedir. Günümüz dijital dünyasında bilgiyi paylaşmak ve bilgiye ulaşmak basit hâle gelmiş fakat doğru bilgiye güvenilir referanstan nasıl ulaşılacağı ve bilginin güvenliği sorunu ortaya çıkmıştır. Bu bağlamda, dijital sistemler bilgi güvenliği dikkate alınarak tasarlanmalıdır [1].

Bilgi güvenliği, bilgiye yetkisiz kişiler tarafından erişilmesi, bilginin değiştirilmesi veya silinmesi durumlarının önlenmesidir. Bilgiler depolandığı sistem içerisinde yetkisiz kişiler tarafından erişime açık olmamalı aynı şekilde yetkisiz kişiler tarafından içeriğinin değiştirilmesine veya silinmesine izin verilmemelidir [1].

Güvenli haberleşmenin oluşabilmesi için kimlik doğrulama (authentication), bütünlük (integrity) ve gizlilik (confidentiality) durumları sağlanmalıdır [2].

Kimlik doğrulama, mesajın doğru kişiden geldiğinden ve sahtekarlık içermediğinden emin olunmasıdır. Gönderici ve alıcı arasında doğrulama yapılmasını sağlar. Bütünlük, verinin bir başkası tarafından ele geçirilip değiştirilmediğinden emin olunmasıdır. Verinin göndericiden çıktığı halde alıcıya ulaştığını doğrular. Gizlilik ise verilere yetkisiz kişilerin ulaşmasının engellenmesidir. Veriler işlenirken, depolanırken, transfer edilirken yetkisiz erişimlerden korunmasıdır [2].

Kimlik doğrulama, bütünlük ve gizlilik kriptografinin bileşenleridir. Kriptografi, şifreleme bilimin temel alt dallarından biridir [2]. Gönderici ve alıcı arasında gerçekleşen haberleşmenin güvenliğini sağlar. Matematiksel problemlere dayanır ve açık metinlerin şifrelenmesiyle ilgilenir [1].

Herhangi bir kişi için önemli olan ve kolayca herkes tarafından erişilmesi istenmeyen veriler hassas veri olarak tanımlanır. Gerçek bir kişinin doğrudan veya dolaylı olarak fiziksel, kültürel, ekonomik veya sosyal kimliğinin belirlenebilir olmasını sağlayan bilgiler ise kişisel veri olarak tanımlanır. Hassas ve kişisel verilerin günümüz şartlarında kötü niyetli kişilerin veya üçüncü şahısların eline geçmemesi, verilerin doğru alıcıya içeriği değiştirilmeden ulaştırılması önem arz etmektedir [3].

Kişisel Verilerin Korunması Kanunu (KVKK) kapsamında, kişisel verilerin kişinin izni olmadan üçüncü şahıslarla paylaşılması engellenmelidir. KVKK'daki asıl amaç kişinin özel hayatı başta olmak üzere din, dil, ırk, cinsiyet ayrımı yapılmaksızın haklarının korunması ve kendisine gelebilecek olan zararın engellenmesidir. Bu kapsamda verilerin iyi bir şekilde korunması, belirli, açık ve meşru amaçlar içerisinde sadece yetkili kişiler tarafından işlenmesi önemlidir. Veriler toplandığı ve işlendiği amaç ile bağlantılı, tutarlı ve sınırlı olmalı, işlendiği hedef için gerekli olduğu süre kadar tutulmalıdır. Kişisel ve hassas verileri işleyen tüm kurum ve kuruluşlar, herhangi bir güvenlik ihlali olduğunda şahıslara bilgi vermek ve belirli önlemleri uygulamakla yükümlüdür. Bu önlemlerden biri verileri şifrelemektir [4].

Şifrelemeyle ilgili yapılan çalışmalar ve geliştirilen teknikler, verilerin güvenli bir şekilde iletilmesi ve depolanması konusunda katkı sağlamıştır. Bu tekniklerden bazıları simetrik şifreleme ve asimetrik şifreleme yöntemleridir. Simetrik şifreler genel olarak asimetrik şifrelere oranla daha hızlıdır fakat güvenlik zafiyeti daha fazladır. Simetrik şifreler kendi içerisinde blok veya akış şifreleme modellerinden birini kullanabilir [5].

Blok şifreleme yönteminde birçok farklı teknik kullanılmaktadır. Bu tekniklerden biri de Şifre Blok Zincirlemesi (CBC) yöntemidir. CBC, verilerin güvenliğini ve gizliliğini artıran, yaygın olarak kullanılan bir şifreleme modelidir. Genellikle Amerika Birleşik Devletleri (ABD) hükümeti tarafından hassas bilgilerin güvenliğini sağlamak için simetrik şifreleme algoritması olan Gelişmiş Şifreleme Standardı (AES) ile birlikte kullanılır [6]. Benzer çalışmalar incelendiğinde, Ors ve arkadaşları, bir ASIC (Uygulamaya Özel Tümleşik Devre) AES uygulamasına yönelik güç analizi saldırısı önermiştir [7]. Alwan ve Kashmar, tamamen bağlı bir sinir ağına dayalı olarak blok şifreleri analiz etmiş, sonucunda ise yüzde seksen bir doğruluk elde etmiştir [8]. Chowdhary ve arkadaşları, AES odaklı olarak görüntü şifreleme ve şifre çözme için hibrit teknikler üzerinde çalışmış, elde edilen performansın yüzde yüze yakın olduğunu belirtmiştir [9]. Goel ve arkadaşları, AES'in şifreleme ve şifre çözme sürecini tartışmış ve AES'ye karşı hiçbir pratik kriptanalitik saldırının keşfedilmediğinden bahsetmiştir [10]. Huang ve arkadaşları, güvenli video izleme için AES-CBC ile yazılım tanımlı bir ultrasonik iletişim sistemi önermiştir [11]. Amudha

ve arkadaşları, tıbbi görüntü şifreleme için CBC modunda AES ile hibrit bir Baker haritası kullanmıştır [12]. Bu ilgili çalışmalar, blok şifreleme modelleri ve AES şifreleme tekniklerinin geliştirilmesine ve iyileştirilmesine katkıda bulunmuştur. Bu çalışmalardan AES algoritması ve CBC gibi blok şifreleme modellerinin iyi sonuçlar verdiği görülmektedir. CBC modelini AES ile birleştirmek şifrelemenin güvenliğini daha da artırır. CBC modelinin zincirleme mekanizması, düz metindeki kalıpların şifreli metinde gözlemlenmesini engelleyen bir rastgelelik unsuru sunmaktadır [13].

Blok ve akış şifreleme tekniklerinde aynı açık metinler aynı anahtar ile şifrelenirse her seferinde aynı şifreli metin ortaya çıkmaktadır. Bu durumu engellemek için her iki şifreleme tekniğinde de başlatma vektörü (IV) kullanılmaktadır [5]. Blok şifreleme yönteminin birçok modelinde başlatma vektörü kullanılmasına rağmen Elektronik Kod Defteri (ECB) modelinde kullanılmadığı için ECB, CBC ve Şifre Geri Beslemeli (CFB) gibi diğer modellere oranla daha fazla güvenlik zafiyeti barındırır. Çünkü aynı açık metnin aynı anahtarla şifrelenmesi sonucunda saldırganlar benzer veya aynı trafik akışından belirleme yapabilir [14].

Simetrik şifreleme tekniğinde başlatma vektörü kullanılarak bazı güvenlik zafiyetlerinin önüne geçilmiştir fakat sadece bir tane gizli anahtar kullanılması ve güvenli anahtar dağıtımının zorluğundan dolayı bazı sistemlerde asimetrik şifreleme tekniğinin kullanılması daha doğrudur. Özellikle ağ büyüdükçe anahtar sayısı arttığı için kapasite sorununa sebep olabilmektedir. Asimetrik şifreleme tekniğinde açık anahtar herkesin erişimine açık olduğu için bu sorunlar çözülmektedir fakat asimetrik şifreler de simetrik şifrelere oranla daha yavaş çalışmaktadır [15].

Şifreleme işlemi farklı seviyelerde ve farklı katmanlarda gerçekleşebilmektedir. İşletim sistemi katmanında, sunucu katmanında veya veritabanı katmanında şifreleme yapılabilir. Veritabanı katmanında yapılan şifreleme işlemleri kolon seviyesinde, dosya seviyesinde veya veritabanı seviyesinde gerçekleşebilir [16]. Yapılan bir başka çalışmada ise Bouchti ve arkadaşları, veritabanındaki veri güvenliğini artırmak için şifreleme sınıflarına dayalı yeni bir veritabanı şifreleme modeli önermiştir [17].

Aynı şifreleme tekniğiyle tasarlanan algoritmalarının performans ve güvenlik unsurları farklılık gösterebilmektedir. Şifreleme tercihi yapılırken sistemin ne kadar

güvenli olması gerektiği ve şifreleme işlemlerinin performansa olan yansıması göz önünde bulundurulmalıdır. Eğer veritabanı katmanında bir şifreleme yapılmış ise şifre çözme işlemi de veritabanı katmanında gerçekleşmelidir [16].

Şifreleme sonucunda verilere yetkisiz kişiler erişmiş olsa bile doğrudan metinlerin açık halini göremeyecektir. Verileri okuyabilmek için ayrıca şifre çözme işlemiyle ilgilenmesi gerekecektir. Bu işlemlerle kriptanaliz bilimi ilgilenmektedir.

Kriptanaliz şifreli metinleri açık metne dönüştürmeyi amaçlar. Kriptanaliz yapan kişi gizli anahtarı bilmeden şifreli metni çözmeye çalışır. Şifreli metinleri çözmek için bazı metotlar kullanılmaktadır. Bu metotlardan en çok tercih edilenlerden biri kaba kuvvet saldırısıdır [2].

Kaba kuvvet saldırısında, saldırgan mümkün olan tüm anahtarları deneyerek şifreli metni deneme yanılma yöntemiyle çözmeye çalışır. Tüm şifreleme algoritmaları kaba kuvvet saldırısına maruz kalabilir. Modern şifreleme algoritmalarının asıl amacı şifreyi kırılamaz yapmaktan ziyade şifreyi uzun zamanda kırılacak şekilde geliştirmektir [2].

Şifreleme algoritmaları saldırılara karşı dayanıklı olmalıdır. Algoritmalardaki güvenlik zafiyetleri iyileştirilirken bu durumun performansa karşı tolere edilemeyecek şekilde olumsuz etkisi olmamalıdır. Performans ve güvenlik açısından aradaki denge iyi korunmalıdır. Kullanılan şifreleme algoritması, kullanılan anahtarın uzunluğu vb. durumlar şifre çözme işlemini etkilemektedir. Eğer güçlü bir şifreleme algoritması kullanılmış ise saldırganın şifreyi çözme işlemi de uzayacaktır.

Sonuç olarak hassas ve kişisel verilerin güvenliği önemli bir konudur. Bu verilerin korunması için hem şahısların hem de kurum ve kuruluşların önlem alması gerekmektedir. Alınabilecek önlemlerden biri de verileri şifrelemektir. Sistemin ihtiyaçlarına göre kullanılacak olan şifreleme algoritması ve bu şifreleme işlemlerinin hangi katmada, hangi seviyede gerçekleşeceği belirlenmelidir.

Bu çalışmada şifreleme algoritmaları, özellikleri ve aralarındaki farklar incelenmiştir. Blok şifreleme yöntemini kullanan algoritmalar için farklı blok modellerindeki performans kıyaslaması yapılıp sonuçlar analiz edilmiştir. Aynı zamanda farklı yöntemlerle ve farklı seviyelerde art arda gerçekleşen iki şifreleme işleminin yapıldığı

dinamik bir veri kayıt sistemi geliştirilmiş ve geliştirilen uygulama hakkında bilgi verilmiştir.

Çalışmanın ilk bölümünde temel kavramlar ve metotlar hakkında bilgi verilmiştir. Ardından materyal ve yöntem bölümünde blok şifreleme modelleri, Veri Şifreleme Standardı (DES), Üçlü Veri Şifreleme Standardı (3DES), AES, RSA (Rivest, Shamir, Adleman) algoritması ve SQL Server (Yapılandırılmış Sorgu Dili Sunucusu) kolon seviyesinde şifreleme yöntemi hakkında bilgi verilmiştir. Bir sonraki bölüm olan modellemede farklı seviyelerde farklı şifreleme algoritmalarının kullanıldığı, güvenli, iki adımlı şifrelemeden oluşan dinamik bir veri kayıt sistemi geliştirilip, geliştirilen uygulama hakkında bilgi verilmiştir. Ardından araştırma bulguları olan bölümde, uygulama içerisinde kullanılan algoritma ve şifreleme tekniklerinin performans kıyaslaması yapılıp sonuçlar analiz edilmiştir. Çalışmanın son kısmında ise elde edilen sonuçlar hakkında özet bilgi verilmiştir.

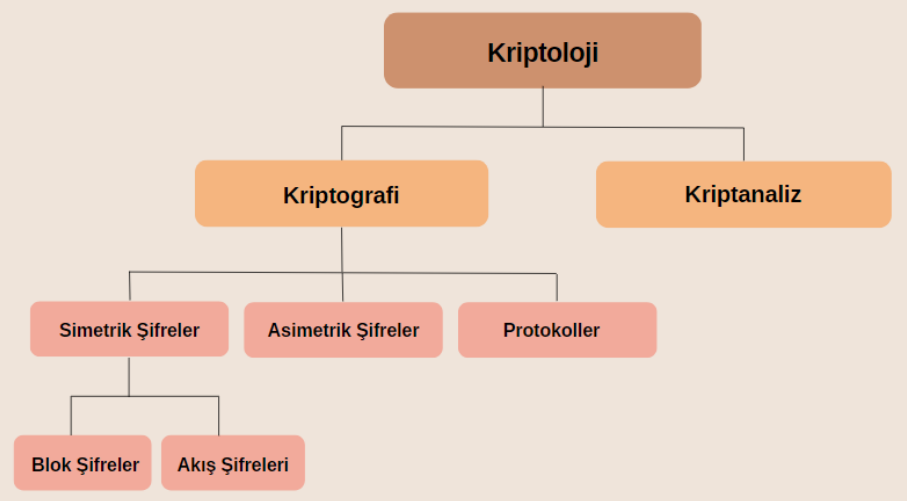
2. TEMEL KAVRAMLAR VE METOTLAR

2.1. Kriptoloji

Kriptoloji, matematiksel ve karmaşık işlemlere dayanan bir şifreleme bilimidir. Şifreleme, açık verilerin yetkisiz kişiler tarafından okunamayacak şekilde karıştırılmasıdır. Kriptoloji bilimi temelde kriptografi ve kriptanaliz olmak üzere iki alt dala ayrılmaktadır [18].

Kriptografi, verilerin paylaşılacak kişiler dışında kalan kişilerce anlaşılmaması için açık veriler üzerinde şifreleme işlemi yapan daldır. Simetrik şifreler, asimetrik şifreler ve protokoller olmak üzere 3 ana gruba ayrılır. Kimlik doğrulama, gizlilik, bütünlük gibi amaçlara ulaşmayı ilke edinir [4].

Kriptanaliz ise şifrelenen verilerin çözülerek orijinal metnin elde edilmesini amaçlayan daldır. Şifreyi çözmek için sistemin analizini yaparak kuvvetli ve zayıf yönlerini ortaya çıkarır. Analiz yapacak kişi şifreleme algoritmasında kullanılan gizli anahtarı elde ederse şifreyi çözme işlemi de kolaylaşır [19]. Şekil 2.1'de kriptolojinin dalları verilmiştir.



Şekil 2.1. Kriptolojinin dalları.

Kriptolojiyle ilgili bazı terimler:

- Plaintext/Cleartext (Açık/Düz Metin): Şifresiz metin.
- Ciphertext: Şifreli metin.
- Encryption: Açık metni şifreleme.
- Decryption (Deşifreleme): Şifreli metni açık metne çevirme.
- Kriptoanalist: Anahtarı bilmeden şifreli metni, açık metne çevirmek için analiz eden kişi.
- Key (Anahtar): Açık metni şifrelemek veya şifreli metni açık metne dönüştürmek için şifreleme algoritmaları tarafından kullanılan bir bit dizisidir [20].

2.2. Simetrik Şifreleme

Simetrik şifreleme yöntemi, açık metni şifreleme ve şifreli metni çözmek için tek bir anahtar kullanmaktadır. Veri transferi sırasında gönderici açık metni şifrelemek için bir anahtar üretir ve metni şifreler. Üretilen bu anahtar sadece gönderici ve alıcı arasında şifreleme ve şifre çözme işlemini gerçekleştirmek için kullanılan ortak ve gizli bir anahtardır (private key). Bu yüzden şifreleme tekniği tek anahtarlı kriptografi, paylaşılan gizli anahtar kriptografisi vb. şekilde de isimlendirilmektedir. Haberleşme içinde olan gönderici ve alıcı anahtar için anlaşır ve bu işlemi gizli tutar. Gönderilecek mesajla birlikte oluşturulan gizli anahtar da metne eklenir ve önceden anlaşılmış olan alıcıya aynı anda, birlikte gönderilir [19].

Alıcıya şifreli metin gönderilirken, şifreli metinle birlikte gizli anahtarın da güvenli bir şekilde gönderilmesi gerekmektedir [15]. Çünkü şifreleme ve şifre çözme işleminde aynı anahtar kullanıldığı için anahtarın üçüncü bir şahsın eline geçmesi halinde şifreli mesaj çok kolay bir şekilde çözülebilir. Şekil 2.2'de simetrik şifreleme yöntemine göre açık metnin gizli anahtar ile şifrelenerek şifreli metnin oluşturulduğu, ardından aynı gizli anahtar ile şifreli metnin çözülerek açık metnin orijinal haline ulaşıldığı diyagram verilmiştir [21].

Şekil 2.2. Simetrik şifreleme ve şifre çözme diyagramı.

Simetrik şifreleme algoritmalarında şifreleme ve şifre çözme işlemleri hızlı bir şekilde gerçekleşir. Bu durum simetrik şifrelerin en büyük avantajlarındandır ve algoritmaların donanımla birlikte kullanılmasını kolaylaştırır [22]. Hızlı bir algoritma olduğu için günümüzde oldukça yaygın kullanılmaktadır [15].

Simetrik şifreleme tekniğinin en büyük dezavantajlarından biri ise güvenli anahtar dağıtımının zor olmasıdır. Hem şifreleme hem de şifre çözme işleminde tek ve ortak bir anahtar kullanıldığı için anahtarın ortaya çıkması şifreli metnin çözülmesini kolaylaştırmakta ve anahtarın güvenli bir şekilde iletilmesi sorununu ortaya çıkarmaktadır. Bu nedenle anahtarın güvenli bir şekilde dağıtılması çok önemlidir [19]. Şifreleme tekniğinin diğer dezavantajı ise "bütünlük" ve "kimlik doğrulama" hizmetlerinin güvenli bir şekilde gerçekleştirilmesinin zor olmasıdır [22].

Simetrik şifreleme tekniğini kullanan bir sistemde anahtar sayısı formül 2.1'de verilen denklem ile hesaplanır. Burada n, kullanıcı sayısını ifade etmektedir.

$$n * (n - 1)/2 \tag{2.1}$$

Bu durum kullanıcı sayısının fazla olduğu büyük ağlarda çok fazla anahtar üretilmesi anlamına gelir ve beraberinde kapasite sorununa sebebiyet verebilir. DES, 3DES, AES, RC2, RC4, RC6, IDEA ve Blowfish algoritmaları simetrik şifreleme yöntemini kullanan algoritmalardandır [23].

2.2.1. Blok şifreleme

Blok şifrelemede açık metinler bitişik bloklara bölünür. Bölünen bloklar ayrı ayrı şifrelenerek şifreli metin bloklarına dönüştürme ve bu şifreli blokları şifreli metin çıkışı olarak gruplama işlemi yapılır [24].

Kullanılan algoritmaya göre anahtar uzunluğu 56 bit, 128 bit, 192 bit, 256 bit olabilir. Anahtar kaba kuvvet saldırısına dayanıklı olmalıdır bu yüzden tercih edilen anahtarın uzunluğunun büyük olması önemlidir [24].

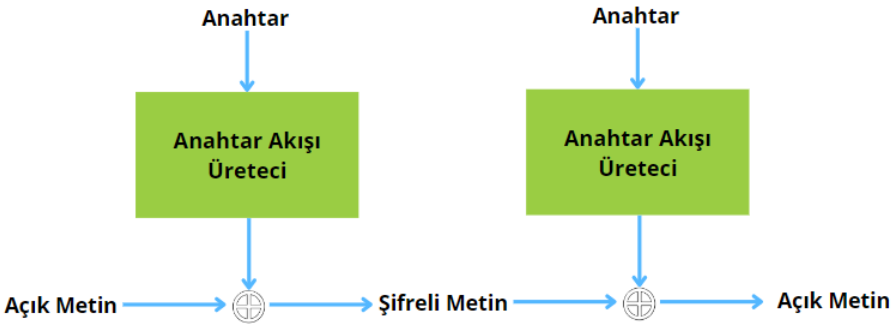
Blok boyutu, ne kadar verinin aynı anda şifreleneceğini temsil eder. Blok boyutu 64 veya 128 bit gibi farklı uzunluklarda olabilir. Blok boyutu arttıkça şifreleme işleminin uygulanması karmaşık hale gelir. Blok şifreleme algoritmalarının ECB, CBC, CFB gibi birçok farklı çalışma modeli mevcuttur. Şifreleme işlemi tercih edilen çalışma modeline göre işlenmektedir. Aynı blok girdisi aynı anahtar ile şifrelendiğinde, sonucunda her zaman aynı şifreli veri bloğu oluşur. Bu durumu engellemek için birçok blok modelinde başlatma vektörü (IV, Initialization Vector) adı verilen bir vektör kullanılır. DES, 3DES, AES, RC2, Blowfish, IDEA, Camellia algoritmaları blok şifreleme yöntemini kullanan algoritmalardandır [5].

2.2.2. Akış şifreleme

Akış şifreleme tekniğinde şifreleme ve şifre çözme işlemi için mesajın bir baytı alınarak işlem yapılır. Blok şifreleme yöntemi ile akış şifreleme yöntemi arasındaki en büyük fark, blok şifrelemenin bir seferde bir metin bloğunu şifrelemesi ve çözmesidir. Akış şifresi ise bir seferde metnin bir baytını alarak şifreleme ve şifre çözme işlemini gerçekleştirir [25].

Akış şifreleri başka bir deyişle belli bir uzunlukta belirli bir periyottan sonra kendini tekrar eden anahtar üreteçleri olarak tanımlanmaktadır. Bu yöntemde, öncelikle üreteç ve gizli anahtar yardımıyla bir anahtar dizisi oluşturulmaktadır. Ardından girdi mesajın ve dizinin her biti ayrı ayrı XOR'lanmaktadır (mod 2 toplama). Şifre çözme işlemi de aynı şekilde XOR işlemine tabi tutularak gerçekleşmektedir. Hem şifreleme hem de deşifreleme işleminde mod 2 toplaması yapılmasının sebebi mod 2 işleminde toplama ve çıkarmanın aynı işlemler olmasıdır. Bu şifreleme tekniklerinin güvenliği, oluşturulan anahtar dizisinin rastgeleliğine bağlıdır. Bu yüzden dizi tamamen rastgele bir veri olarak kullanılmalıdır. Kısacası anahtar dizisi üreteci, akış şifreleme tekniğinde önemli bir unsurdur. Anahtar üreteçlerinde oluşturulan dizinin kendisini tekrarlamaması gerekmektedir ayrıca sonraki anahtar bitleri de öncekiler yardımıyla elde edilememelidir [24]. Şekil 2.3'de akış şifreleme algoritmasına göre oluşturulan

anahtar dizilerinin, şifreleme ve şifre çözme işlemlerinde XOR işlemine tabi tutulduğu yapı verilmiştir.



Şekil 2.3. Akış şifreleme algoritmasının çalışma yapısı.

Aynı açık veri aynı anahtar ile şifrelendiğinde ortaya her zaman aynı şifreli veri çıkacaktır. Bu durumu engellemek için tıpkı blok şifreleme yönteminde olduğu gibi başlatma vektörü kullanılmaktadır [5]. Tablo 2.1'de blok şifreleme ve akış şifreleme teknikleri arasındaki temel benzerlikler ve farklar verilmiştir [26].

Tablo 2.1. Blok şifreleme ve akış şifreleme arasındaki farklar.

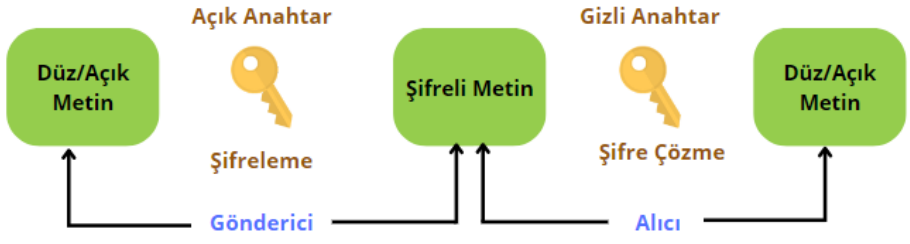
Karşılaştırma Unsuru	Blok Şifreleme	Akış Şifreleme
Çalışma Prensibi	Blok bazında işlem yapar.	Bayt bazında işlem yapar.
Bit Sayısı	64 bit ve üzeri	8 bit
Çözülme Zorluğu	Şifreli veriyi çözmek zordur.	Şifreli veriyi çözmek daha kolaydır.
Başlatma Vektörü (IV)	Bazı blok modellerinde başlatma vektörü kullanılır.	Başlatma vektörü kullanılır.

2.3. Asimetrik Şifreleme

Simetrik şifreleme yöntemindeki anahtar dağıtım problemini çözmek için asimetrik şifreleme yöntemi geliştirilmiştir. Bu yöntemde simetrik şifrelemenin aksine şifreleme ve şifre çözme işlemlerinde ayrı ayrı anahtarlar kullanılır [15].

Şifreleme herkes tarafından bilinen açık anahtar (public key) ile yapılır, şifre çözme işlemi ise gizli anahtar ile yapılır. Gizli anahtarın sadece bir sahibi vardır. Gizli anahtara sahip olan kişi, gizli anahtarla şifrelenmiş verilerin şifresini çözebilir. Açık anahtar herkese açık olduğu için simetrik şifrelemedeki anahtar dağıtım problemi bu yöntemde ortadan kalkmıştır [15]. Bu şifreleme yöntemi genelde şifreleme ve dijital imza işlemleri için kullanılmaktadır [27].

Asimetrik şifreler eski posta kutularına benzetilebilir. Posta kutusuna herkes mektup atabilir fakat kutuyu sadece anahtarın sahibi açabilir. Asimetrik şifreleme de bu mantığa benzer bir şekilde çalışmaktadır [15]. Şekil 2.4'de asimetrik şifrelemeye göre açık anahtarla açık metinlerin şifrelenmesi, gizli anahtarla ise şifreli metinlerin çözülmesi süreci verilmiştir.



Şekil 2.4. Asimetrik şifreleme.

Asimetrik şifrelerde, tercih edilen algoritma ve anahtar uzunluğuna bağlı olarak işlemlerin hızı değişir. Bu algoritmalar çok yoğun matematiksel hesaplamalar gerektirir. Bu nedenle simetrik şifrelere göre daha yavaştır fakat güvenli anahtar dağıtımı problemini ortadan kaldırması, kriptanaliz direncinin yüksek olması nedeniyle daha güvenilirdir. Bu durum da simetrik şifrelere göre tercih edilme durumunu artırmaktadır [28]. RSA, El-Gamal ve Eliptik Eğri algoritmaları asimetrik şifreleme tekniğini kullanan algoritmalardandır [27]. Tablo 2.2'de simetrik şifreleme yöntemi ile asimetrik şifreleme yöntemi arasındaki temel farklar verilmiştir [29].

Tablo 2.2. Simetrik şifreleme ve asimetik şifreleme arasındaki farklar.

Kıyaslama Unsuru	Simetrik Şifreleme	Asimetrik Şifreleme
Anahtar Dağıtımı	Güvenli anahtar dağıtımı problemi mevcuttur.	Açık anahtar, güvenli anahtar dağıtımı problemini çözmüştür.
Anahtar Sayısı	Bir tane gizli anahtar kullanılmaktadır.	Gizli ve açık anahtar olmak üzere iki anahtar kullanılmaktadır.
Performans	Hızlıdır.	Simetrik şifrelere göre daha yavaştır.
Bilgi Güvenliği	Bütünlük ve kimlik doğrulamanın güvenli bir şekilde gerçekleştirilmesi zordur.	Bütünlük, kimlik doğrulama ve gizlilik güvenli bir şekilde gerçekleşir.
Kapasite	Büyük ağlarda kapasite sorunu oluşabilmektedir.	Açık anahtar kapasite sorununu çözmektedir.

2.4. Microsoft SQL Server Şifreleme Yöntemleri

Yazılım seviyesinde veri güvenliği sağlanmak istenildiğinde genel olarak 3 veri şifreleme katmanı ortaya çıkmaktadır. Bu katmanlar; işletim sistemi, sunucu ve veritabanı katmanlarıdır [16].

İşletim sistemi seviyesinde şifrelemeye Microsoft örnek verilebilir. Microsoft, Windows 2000 işletim sistemiyle birlikte Data Protection API (DAPI) isimli kütüphanesiyle verilerin şifrelenmesi konusunda hizmet vermeye başlamıştır. DAPI ile kullanıcı bilgilerinden üretilen simetrik bir anahtarla veriler şifrelenebilir ve çözülebilir olmuştur [16].

Sql Server üzerinde sunucu ve veritabanı seviyesinde şifreleme işlemleri yapılabilmektedir. Sql Server üzerinde şifreleme işlemi ilk olarak 2005 versiyonu ile ortaya çıkmıştır. Sonraki versiyonlarda bu alanda çeşitli yenilikler ve gelişmeler yaşanmıştır [16].

SQL Server şifreleme yöntemlerinde genel olarak kolon seviyesinde şifreleme (column level encryption), dosya seviyesinde şifreleme (file level encryption), yedekleri şifreleme (backup encryption) ve iş yükü istemcide olan şifreleme (always encryption) yöntemleri kullanılmaktadır [16].

Eğer bir tablonun sadece belirli kolonları hassas veya kişisel veriden oluşuyorsa kolon seviyesinde şifreleme yapılabilir. Bu yöntemde sadece şifreli kolonlar üzerinde doğrudan okuma veya yazma işlemleri yapılamaz. Böylece şifreleme ve şifre çözme yükü ihtiyaç doğrultusunda kullanılmış olur [16].

Eğer veritabanının ve yedeklerinin tamamı şifrelenmek isteniyorsa veritabanı seviyesinde (transparent data encryption) şifreleme yapılabilir. Bu şifreleme yönteminde veriler sadece kendi sunucumuzda bellekte şifresiz olarak görünür. Veritabanı yedekler de dahil olmak üzere başka sunucuda kullanılamaz. Veriler sayfalara (page) yazılır ve çağrıldığında şifresi çözülerek belleğe çıkarılır. Bu nedenle bu şifreleme yönteminde şifreleme yükü fazla olur ve sayfa düzeni bozulur [16].

Eğer sadece veritabanının yedeklerinin şifrelenmesi isteniyor ise yedekleri şifreleme yöntemi ile şifreleme gerçekleştirilebilir. Bu şifreleme yöntemi genelde farklı konumlara alınan yedeklerin korunması için tercih edilir. Bu yöntemde yedekleme yapılırken (backup) ve yedekten dönerken (restore) iş yükü ortaya çıkar [16].

Eğer tüm veritabanının şifrelenmesi isteniyorsa ve iş yükünün istemciye yüklenilmesi isteniyorsa “always encrytion” yöntemi tercih edilebilir. Bu yöntem SQL Server 2016 versiyonu birlikte ortaya çıkmıştır [16].

2.4.1. Sunucu seviyesinde şifreleme

SQL Server servisi kurulurken Windows Veri Koruma Uygulama Programlama Arayüzü (Windows DPAPI) kullanılarak Servis Ana Anahtarı (Service Master Key - SMK) oluşturulur. SMK “master” veritabanında “Security” dizini altında ‘Symmetric Key’ klasörünün içerisinde yer almaktadır.

Her veritabanının kendine ait bir SMK’si mevcuttur ve sunucu seviyesinde şifrelenirler. SQL Server SMK’yi otomatik olarak üretmektedir. “Sys.symmetric_keys” view’i ile anahtar hakkındaki bilgilere erişilebilir [16]. Şekil

2.5'de veritabanı üzerinde çalıştırılması halinde SMK hakkında ne zaman oluşturulduğu ve kullanılan algoritma ile ilgili bilgileri görüntüleyen script verilmiştir.

```
USE master;

SELECT *
FROM sys.symmetric_keys
WHERE name = '##MS_ServiceMasterKey##';
```

Şekil 2.5. SMK hakkındaki bilgileri görüntüleyen script.

SMK kullanılan veritabanından tamamen kaldırılamaz yalnız yeni bir anahtar oluşturulması sağlanabilir. Bu işlem "*ALTER SERVICE MASTER KEY REGENERATE"* komutu ile gerçekleştirilebilir [30]. SMK'nin yedeği ise Şekil 2.6'da belirtilen İşlem Yapılandırılmış Sorgu Dili (T-SQL) komutları ile alınabilir.

```
BACKUP SERVICE MASTER KEY
TO FILE = 'Dizin\Servis_ana_anahtari'
ENCRYPTION BY PASSWORD = 'sifre';
```

Şekil 2.6. SMK'nin yedeğinin alınması.

Gerekli durumlarda yedekten geri dönülebilir. Bu işlemin T-SQL komutları şekil 2.7'de verilmiştir.

```
RESTORE SERVICE MASTER KEY
FROM FILE = 'Dizin\Servis_ana_anahtari'
DECRYPTION BY PASSWORD = 'sifre';
```

Şekil 2.7. Yedekten geri dönme.

SMK, veritabanında üretilen Veritabanı Ana Anahtarını (Database Master Key - DMK) şifrelemektedir. Bu nedenle yedeğinin alınması önemlidir aksi durumlarda yaşanılan sorunlardan sonra şifrelenen veritabanlarına bir daha erişilemeyebilir [16].

2.4.2. Veritabanı seviyesinde sifreleme

Veritabanı Ana Anahtarı (DMK – Database Master Key), manuel olarak oluşturulan veritabanına özel olan bir anahtardır. Her veritabanı bir tane DMK'ya sahiptir. DMK'lar SMK'lar tarafından koruma altındadır ve simetrik anahtarlı şifrelerdir. Bu anahtar AES 256 algoritması ve kullanıcı tarafından belirtilen şifreyle korunur [30].

Verilerin şifrelenmesi için kullanılan sertifika ve asimetrik anahtarlar DMK ile şifrelenip korunur. Eğer sertifika ve asimetrik anahtarlar oluşturulurken verilerin nasıl şifreleneceği belirtilmez ise otomatik olarak DMK kullanılır [30]. Şekil 2.8'de DMK'yi oluşturmak için kullanılan T-SQL konutu verilmiştir.

```
CREATE MASTER KEY
ENCRYPTION BY PASSWORD ='sifre'
```

Şekil 2.8. DMK'nin oluşturulması.

Şekil 2.9'da DMK hakkında detaylı bilgiye erişmek için kullanılan T-SQL komutları verilmiştir.

```
SELECT *
FROM sys.symmetric_keys
WHERE name='##MS_DatabaseMasterKey##'
```

Şekil 2.9. DMK hakkında detaylı bilgiye erişme.

3. MATERYAL VE YÖNTEM

3.1. İlklendirme / Başlatma Vektörü (IV)

Başlatma vektörü, blok şifreleme yönteminde birçok blok modelinde kullanılmaktadır. Aynı açık metnin aynı anahtar ile tekrar şifrelenmesi durumunda anahtar değişimine ihtiyaç duyulmadan farklı şifreli metnin oluşturulmasını sağlar. Adından da anlaşıldığı üzere sadece ilk blokta kullanılmaktadır [31].

Başlatma vektörü genellikle gizli olmak zorunda değildir yalnız bir başlatma vektörünün aynı anahtar ile tekrar kullanılmaması tercih edilmelidir [31].

Saldırgan başlatma vektörünü sonraki açık metni seçmeden önce bilirse, aynı anahtarla daha önceden şifrelenmiş olan açık metinleri analiz edebilir. Bu yüzden özellikle aynı anahtarı kullanan şifreli veriler için her seferinde rastgele farklı bir başlatma vektörünün kullanılması önerilir [31].

3.2. Blok Şifreleme Modelleri / Modları

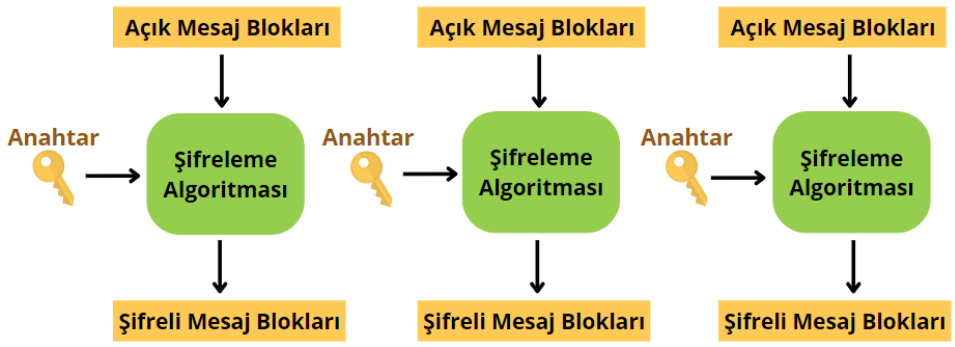
Blok şifreleme işlemi bir simetrik şifreleme yöntemidir. Bu yüzden şifreleme ve şifre çözme işlemlerinde aynı gizli anahtar kullanılmaktadır. Blok şifrelemenin arkasındaki asıl amaç şifreleme veya şifre çözme işlemleri yapılırken metinleri bloklara bölerek işlem yapmaktır. Kullanılan algoritmanın özelliğine göre blok boyutları değişkenlik göstermektedir. Elektronik Kod Defteri (ECB), Şifre Blok Zincirlemesi (CBC), Çıktı Geri Beslemeli Model (OFB), Yayılımlı Şifre Bloğu Zincirlemesi (PCBC), Şifre Geri Beslemeli Model (CFB) gibi birçok farklı blok modeli bulunmaktadır [5].

3.2.1. Elektronik kod defteri modeli (ECB)

ECB, her açık metin bloğunun bağımsız olarak şifrelendiği ve benzer açık metin blokları için benzer şifreli metin bloklarıyla sonuçlanan en temel blok şifreleme modelidir [32].

Şifreleme işleminde açık metin blokları anahtar ile şifrelenerek şifreli metin bloklarını oluşturur. Bu işlem tüm bloklar bitene kadar devam eder. Örneğin AES 256

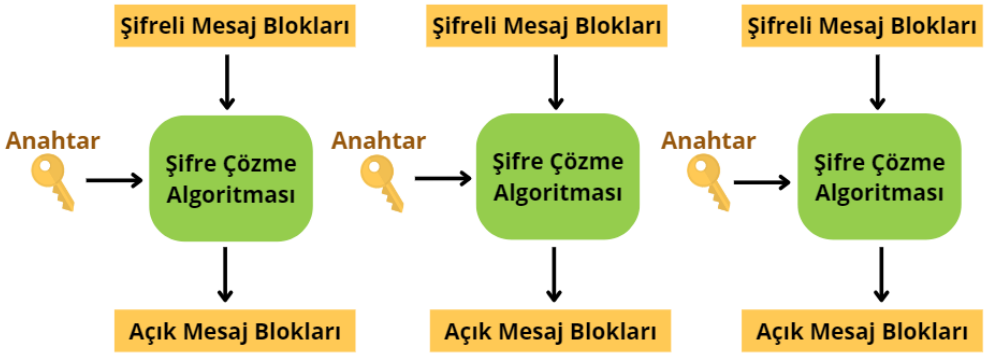
algoritması için açık metinler 256 bit uzunluklara bölünerek işleme alınır. Şekil 3.1'de ECB modeline göre gizli anahtarla şifrelenen bloklar verilmiştir.



Şekil 3.1. ECB modeli şifreleme işlemi.

Aynı blok benzer anahtarlarla şifrelenirse onların şifreli halleri de benzer olur. Bu yüzden saldırganlar benzer veya aynı trafik akışından belirleme yapabilir. Bu algoritma belirli saldırılara karşı savunmasız olduğu için pratikte önerilmez [31].

Şifre çözme işleminde şifreli metin blokları anahtar ile çözülerek açık metin blokları elde edilir. Bu işlem tüm bloklar bitene kadar devam eder [31]. Şekil 3.2'de ECB modeline göre şifrelenmiş olan blokların gizli anahtar ile çözülme süreci verilmiştir.



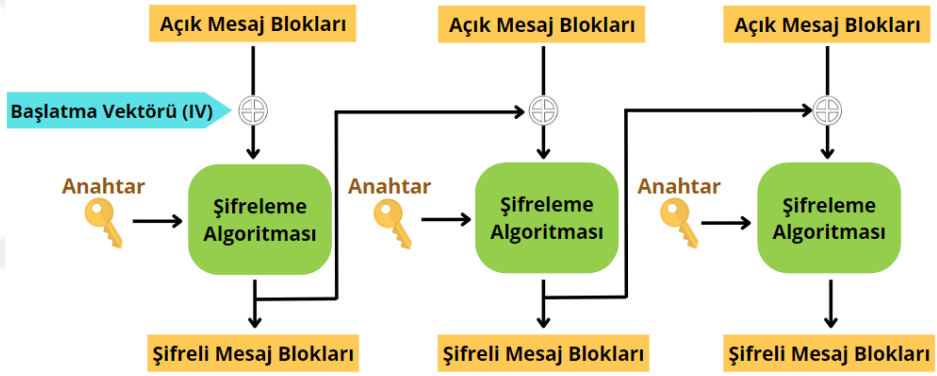
Şekil 3.2. ECB modeli şifre çözme işlemi.

3.2.2. Şifre blok zincirlemesi modeli (CBC)

CBC modeli, ECB modeline göre daha güvenlidir. Burada her açık metin bloğu, şifrelemeden önce bir önceki şifreli metin bloğu ile XOR'lanır. Bu model, bir başlatma vektörünün rastgele seçilmesini ve gizli tutulmasını gerektirir. CBC modeli pratikte yaygın olarak kullanılmaktadır [33].

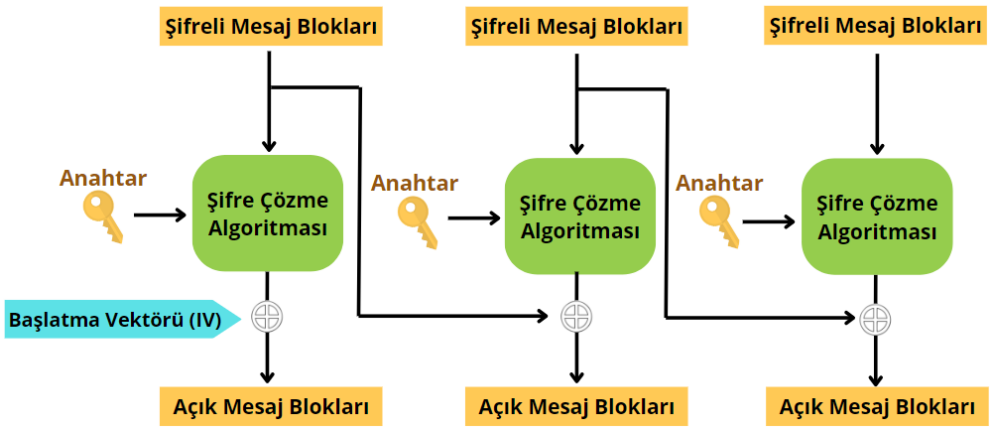
CBC'de ECB modundan farklı olarak her bloğun şifrelenmesi önceki bloğa bağlıdır. Böylece aynı blokların farklı şifreli metinlere dönüşmesi sağlanarak ECB'deki güvenlik zafiyeti çözülmüştür [31].

CBC modeline göre şifrelemede, ilk blokta açık metin blokları başlatma vektörü ile XOR işlemine tabi tutulur. XOR işleminden elde edilen sonuç ile anahtar kullanılarak şifreleme işlemi yapılır ve sonucunda şifreli metin blokları elde edilir. İkinci blok grubuna geçildiğinde ise birinci bloktan elde edilen şifreli metin ikinci bloğun açık metni ile XOR'lanır. XOR işleminden elde edilen sonuç ile anahtar tekrar şifrelenir ve sonucunda şifreli metin blokları oluşur. Bu işlem tüm bloklar bitene kadar devam eder. Şekil 3.3'de CBC modeline göre açık mesaj bloklarıyla şifreli mesaj bloklarının XOR işlemine tabi tutularak, gizli anahtar vasıtasıyla gerçekleşen şifreleme işlemi verilmiştir.



Şekil 3.3. CBC modeli şifreleme işlemi.

Şifre çözmede ilk blok için şifreli metin blokları anahtar ile çözülerek elde edilen deşifreli veri ile başlatma vektörü XOR işlemine tabi tutulur. Bunun sonucunda açık metin blokları oluşur. Ardından diğer bloğa geçildiğinde yine şifreli metin blokları anahtar ile çözülür, ardından elde edilen sonuç ile bir önceki bloğun şifreli metin blokları XOR işlemine tabi tutulur. Bunun sonucunda açık metin blokları elde edilir. Bu işlem tüm bloklar tamamlanana kadar devam eder. CBC modeline göre gerçekleşen bu şifre çözme süreci Şekil 3.4'de verilmiştir.

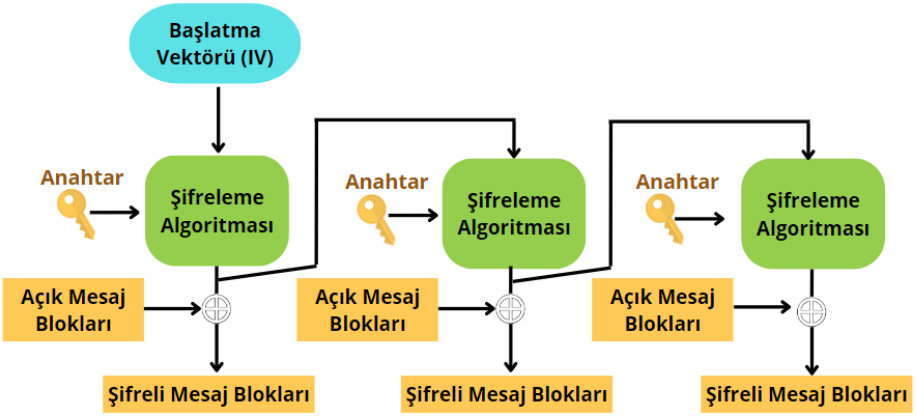


Şekil 3.4. CBC modeli şifre çözme işlemi.

3.2.3. Çıktı geri beslemeli model (OFB)

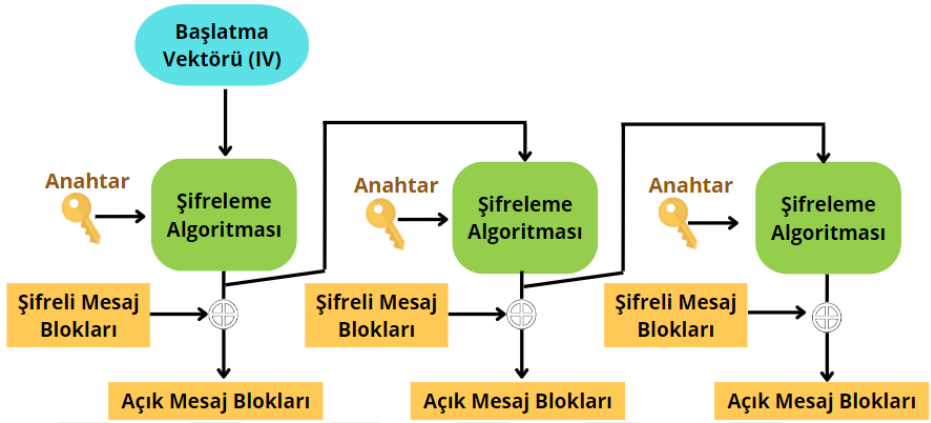
OFB modelinde CBC modelinden farklı olarak açık metinler bölünmektedir. Şifreleme işlemi doğrudan açık metin üzerinden gerçekleşmez. Şifreleme, başlatma vektörü ve anahtar üzerinde gerçekleşmektedir. OFB şifreleme modelinde yine ECB modelinin aksine bloklar arasında bir ilişki söz konusudur, her bloğun şifrelenmesi önceki bloğa bağlıdır [34].

OFB modeline göre şifreleme işleminde ilk blok için başlatma vektörü ve anahtar şifrelenir. Bu işlemin sonucunda çıkan şifreli veri ile açık metin blokları XOR işlemine tabi tutulur ve sonucunda şifreli metin blokları oluşur. İlk bloktan elde edilen şifreli veri ile anahtar tekrar şifrelenir. Elde edilen sonuç ile açık metin XOR işlemine tabi tutulur ve sonucunda tekrardan şifreli metin blokları oluşur. Bu işlem tüm bloklar tamamlanana kadar devam eder. Şekil 3.5'de OFB modeline göre gizli anahtar vasıtasıyla elde edilen şifreli verilerle açık mesaj bloklarının XOR işlemine tabi tutulmasıyla gerçekleşen şifreleme işlemi verilmiştir.



Şekil 3.5. OFB modeli şifreleme işlemi.

Şifre çözme işleminde ise başlatma vektörü ve anahtar şifrelenir. Elde edilen şifreli veri ile şifreli metin blokları XOR'lanır ve sonucunda açık metin blokları elde edilir. İkinci blok için birinci bloktan elde edilen açık metin ile anahtar şifrelenir. Elde edilen sonuç ile şifreli metin blokları XOR'lanır ve açık metin blokları elde edilir. Bu işlem tüm bloklar bitene kadar devam eder. Burada en önemli fark şifre çözme işleminde yine şifreleme algoritmasının kullanılmasıdır [34]. Şekil 3.6'da OFB modeline göre gizli anahtarla elde edilen şifreli verilerin şifreli mesaj bloklarıyla XOR işlemine tabi tutularak açık mesaj bloklarının elde edilmesi süreci verilmiştir.



Şekil 3.6. OFB modeli şifre çözme işlemi.

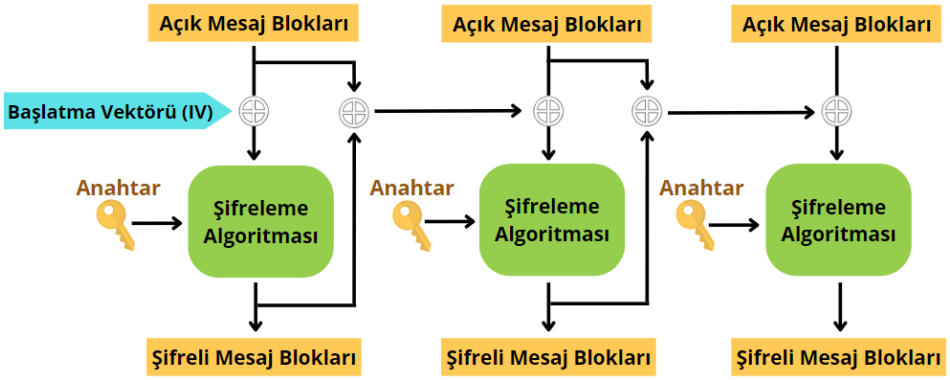
3.2.4. Yayılımlı şifre blok zincirlemesi (PCBC)

PCBC yönteminde, yine ECB'den farklı olarak şifrelenen her blok kendinden önceki bloğa bağlıdır. Bu sayede saldırganın bloklara bağımsız saldırı yapmasının önüne geçilmesi hedeflenmiştir [35].

CBC modelinden tek farkı şifrelenen bloğa sadece bir önceki bloktaki şifreli mesajın değil aynı zamanda açık mesajın da dahil edilmesidir. Açık mesaj blokları, şifreli mesaj bloklarıyla XOR'lanarak tekrardan bir sonraki açık mesajla birlikte XOR değeri hesaplanmaktadır. Şifreli mesaj, ayrı ayrı bloklardan oluşturulan şifreli mesaj bloklarının birleşimidir [35].

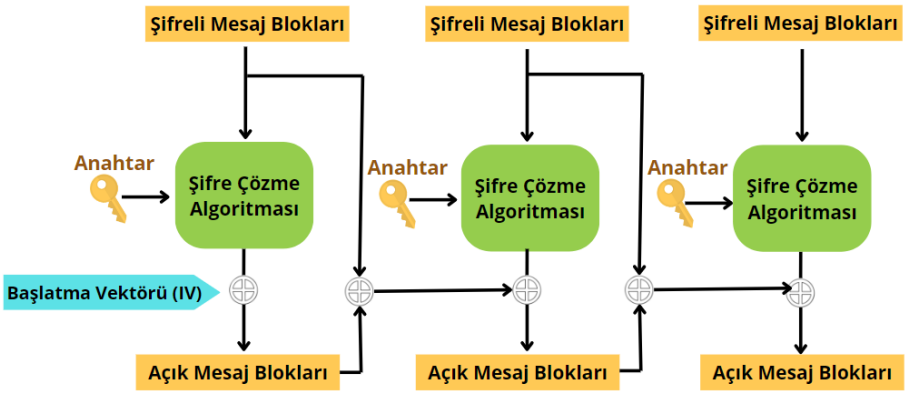
PCBC modeline göre şifreleme işlemi için açık mesaj blokları başlatma vektörü ile XOR işlemine tabi tutulur. Elde edilen XOR işlemi sonucu anahtar ile şifrelenerek şifreli mesaj blokları elde edilir. Aynı zamanda açık mesaj blokları ile elde edilen şifreli mesaj blokları tekrardan XOR işlemine tabi tutulmaktadır. Sonuçta iki kere XOR işlemi yapılan mesaj, anahtar ile şifreleme algoritmasına girmekte ve bunun

sonucunda şifreli mesaj oluşmaktadır [35]. PCBC modeline göre gerçekleşen bu şifreleme işlemi Şekil 3.7'de verilmiştir.



Şekil 3.7. PCBC modeli şifreleme işlemi.

PCBC modeline göre şifre çözme işlemi şifrelemenin tam tersidir. Önce şifreli mesaj blokları gizli anahtar ile çözülür ve elde edilen deşifreli veri ile başlatma vektörü XOR'lanır. Bu işlem sonucunda açık mesaj blokları oluşur. Ardından elde edilen açık mesaj bloklarıyla şifreli mesaj bloğu da XOR işlemine tabi tutulur. Bu XOR işleminin sonucunda ortaya çıkan veri bir sonraki bloğun deşifrelenmiş verisiyle XOR işlemine tabi tutularak açık mesaj bloklarını elde etme süreci devam eder [35]. PCBC modeline göre gerçekleşen bu şifre çözme işlemi Şekil 3.8'de verilmiştir.

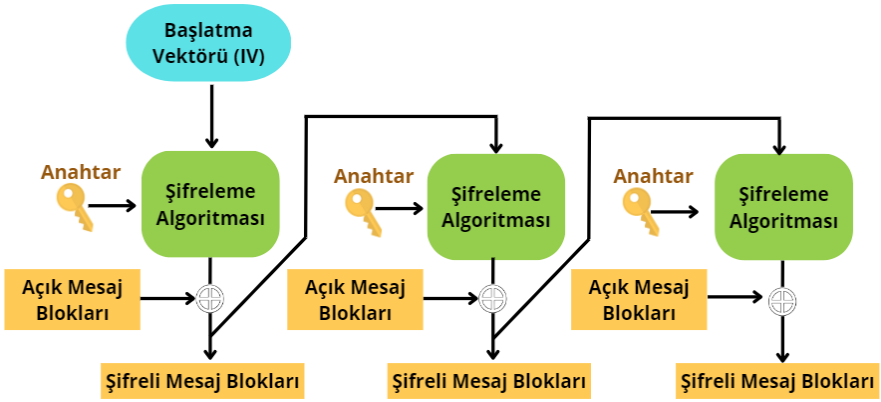


Şekil 3.8. PCBC şifre çözme işlemi.

3.3.5. Şifre geri beslemeli model (CFB)

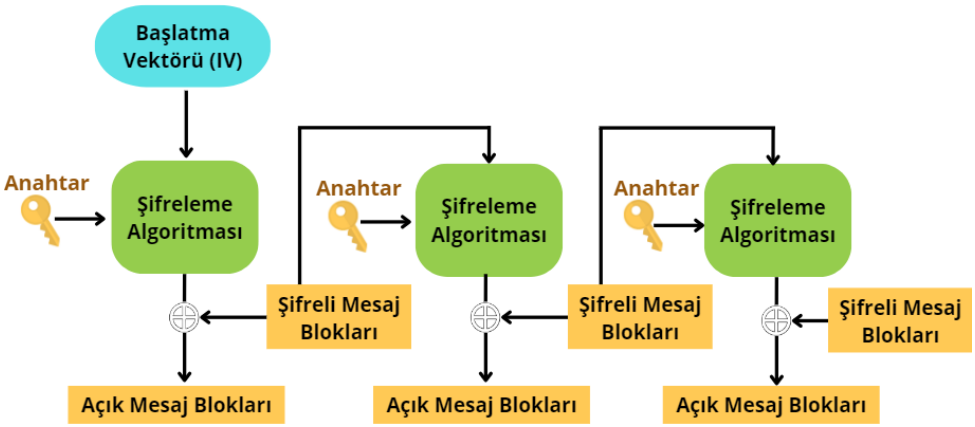
CFB modeli, CBC modeline benzer ancak düz metni önceki şifreli metin bloğuyla XOR'lamak yerine, son şifreli metin bloğunun şifreleme çıktısıyla XOR'lar. Bu mod ayrıca bir IV gerektirir ve pratikte yaygın olarak kullanılır [36].

CFB modeline göre ilk blokta başlatma vektörü anahtar ile şifreleme işlemine tabi tutulur. Bu işlem sonucunda elde edilen şifreli veri ile açık mesaj blokları XOR'lanır ve sonucunda şifreli metin blokları elde edilir. İkinci bloğa geçildiğinde ise birinci bloktan elde edilen şifreli mesaj blokları tekrardan anahtar ile şifreleme işlemine tabi tutulur. Bu işlem sonucunda şifreli bir veri oluşur. Oluşan şifreli veri ile açık metin blokları tekrardan XOR işlemine tabi tutulur ve şifreli mesaj blokları oluşur. Bu işlem aynı şekilde diğer bloklar için de devam eder [37]. CFB modeline göre gerçekleşen bu şifreleme işlemi Şekil 3.9'da verilmiştir.



Şekil 3.9. CFB modeli şifreleme işlemi.

Şifre çözme işlemi için şifreli bloklar şifre çözme algoritmasından çıkan açık mesaj ile XOR işlemine tabi tutulmakta ve sonuçta blokların orijinal hali elde edilmektedir [37]. Bu modele göre şifre çözme işleminde ilk blok için başlatma vektörü, anahtar ile şifrelenir. Ardından elde edilen şifreli veri ile şifreli mesaj blokları XOR işlemine tabi tutulur ve sonucunda açık metin blokları oluşur. İkinci blok için birinci bloktan elde edilen şifreli mesaj blokları ile anahtar şifrelenir. Elde edilen şifreli veri ile şifreli mesaj blokları XOR işlemine tutulur ve sonucunda açık metin blokları oluşur. Bu işlem tüm bloklar bitene devam eder. CFB modelinde de tıpkı OFB modelindeki gibi şifre çözme işleminde şifreleme işlemi yapılmaktadır [31]. CFB modeline göre gerçekleşen bu şifre çözme işlemi Şekil 3.10'da verilmiştir.



Şekil 3.10. CFB modeli şifre çözme işlemi.

3.3. Veri Şifreleme Standardı (DES)

DES, 1970 yılında IBM tarafından geliştirilen, simetrik şifreleme yöntemini kullanan bir blok şifreleme algoritmasıdır. Açık metni parçalara bölerek (blok), her parçayı bağımsız olarak şifreler. Bu parçaların uzunluğu 64 bittir. Hem aynı algoritma hem de aynı gizli anahtar, şifreleme ve şifre çözme işleminde kullanılır. [24].

56 bitlik anahtarı DES'in en büyük zayıflarından biridir. Günümüz şartlarında yapılan anahtar ataklarına karşı zayıf kalmıştır. DES algoritmasının başka bir zayıflığı da yavaş olmasıdır [38].

Yayılma (confusion) ve nüfuz etme (diffusion) özellikleri DES'in en önemli özelliklerindendir. DES'de her bloğun her biti diğer bitlere ve anahtarın her bitine bağımlıdır. Bunun birinci sebebi anahtar üzerindeki bilinmezliği (uncertanity) arttırmaktır. İkinci sebebi ise açık metindeki veya anahtardaki bir bitin değişmesinin bütün şifreli metnin değişmesine sebep olmasıdır [38].

3.4. Gelişmiş Şifreleme Standardı (AES)

DES'in anahtar boyutu 56 bit, blok boyutu ise 64 bittir. DES günümüz şartlarında hem güvenlik hem de performans açısından zayıf kalmıştır. Bu nedenle zaman içerisinde DES'in arka arkaya çalıştırılması anlamına gelen 3DES algoritması ortaya çıkarılmıştır. 3DES ile anahtar uzunluğu 112 bite kadar çıkarılmış fakat blok uzunluğu 64 bitte sabit kalmıştır. Aynı zamanda bu algoritmanın yavaş olmasından kaynaklı olarak AES algoritması geliştirilmiştir [24].

AES algoritması farklı blok boyutlarını ve farklı anahtar büyüklüklerini destekleyen simetrik bir blok şifreleme algoritmasıdır [24]. Şifreleme gerçekleşirken 32 bitin

katları tercih edilir, bu kapsamda en az 128 en fazla ise 256 bit uzunluğunda anahtar kullanılabilir. Bu algoritmaya göre 128 bit uzunluğundaki bloklar üzerinde şifreleme yapılırken aynı amanda 128 bit, 192 bit veya 256 bit boyutundaki anahtarlardan biri tercih edilebilir. AES, anahtar uzunluğundan dolayı DES'den daha güçlüdür ayrıca 3DES'den daha hızlıdır [39].

AES algoritması, genelde tekrarlı bir yapı olmakla birlikte anahtar üretim bloğu ve tur dönüşüm bloğundan oluşmaktadır. Kullanılan anahtar uzunluğuna göre tekrarlı yapının sayısı belirlenmektedir. Bu yapıya göre farklı anahtar uzunluklarında gerçekleşen tekrarlama miktarları Tablo 3.1'de verilmiştir [39].

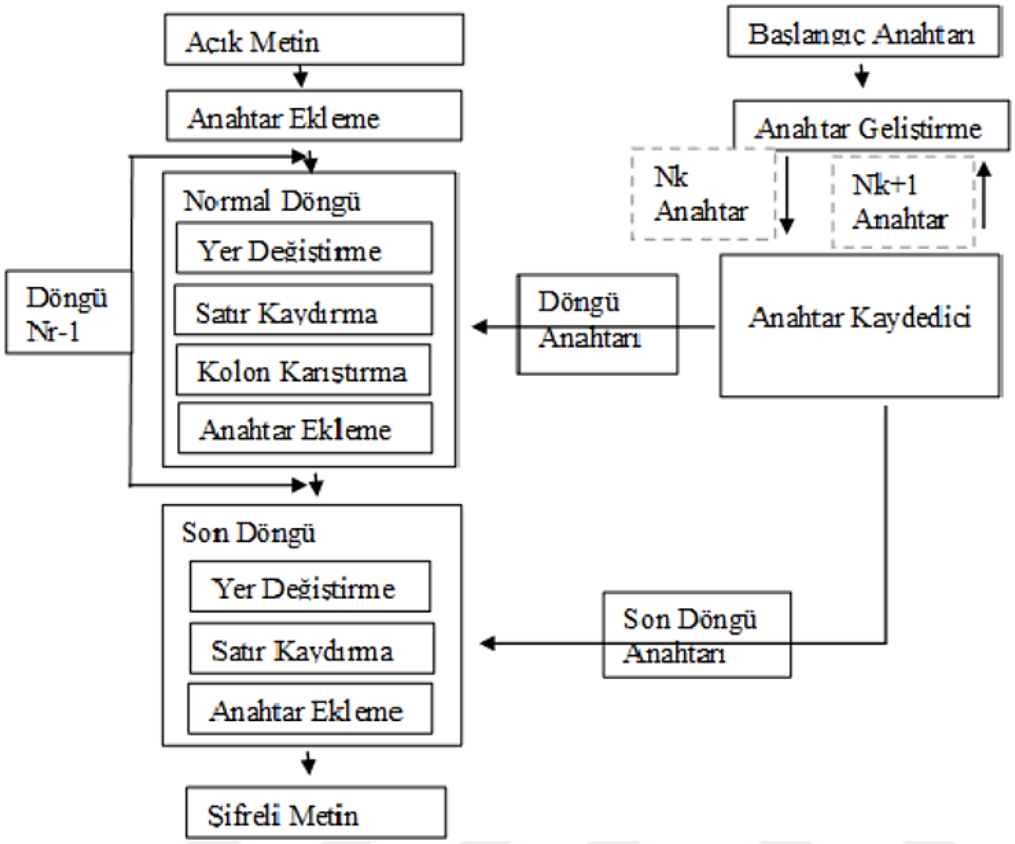
Tablo 3.1. AES algoritmasının anahtar uzunlukları ve döngü sayıları.

Metin Uzunluğu	Anahtar Uzunluğu			
	128	192	256	
128	10	12	14	Döngü sayıları
192	12	12	14	
256	14	14	14	

Algoritmanın her turu 4 bileşenden oluşmaktadır. İlk olarak 128 bit (16 bayt) boyutundaki blok 4x4 bayt'lık matrise dönüştürülür. Ardından her turda sırasıyla şu işlemler geçekleşir; bayt'ların yer değiştirmesi (sub-bytes dönüşümü), satırların ötelenmesi (shift-rows dönüşümü), sütunların karıştırılması (mix-columns dönüşümü) ve döngüye anahtar eklenmesi (AddRoundKey dönüşümü) [39].

Bayt'ların yer değiştirmesi olan ilk adımda 16 bayt (4x4 bayt matris) 8 bit giriş ve 8 bit çıkışı olan S kutusuna sokularak değerleri güncellenir. Bu değerler Galois alanında tersi alındıktan sonra lineer bir hale dönüştürülmüş olur [39].

Satırların ötelenmesi işlemi olan ikinci adımda ise satırlar kaydırılır. Bu durumda ilk satır değişmez diğer satırlar ötelenir, taşan bölümler ise ötelemenin başına eklenir. Sütunların karıştırılması olan üçüncü adımda ise ilgili sütun için o sütundaki değerler karıştırılır. Bu işlem gerçekleşirken Galois alanında iki sayının çarpımı kullanılır. Döngüye anahtar eklenmesi olan son adımda ise döngüye ait olan anahtar ile XOR işlemi gerçekleştirilir [39]. AES'in tür dönüşüm blokları ve bu turların bileşenleri Şekil 3.11'de verilmiştir.



Şekil 3.11. AES şifreleme algoritmasının genel yapısı [40].

Donanımsal ürünlerde ve yazılım platformunda iyi performans göstermesi, anahtar ayarlamasında hızlı olması, düşük bellek ve ekipman sınırlarının olduğu alanlarda kullanımının kolay olması AES'in en önemli avantajlarındandır. Aynı zamanda blok ve anahtar uzunluklarında farklı seçeneklerin kullanılabilmesi son derece esnek bir yapıya sahip olmasına neden olmuştur. Bu nedenle DES ve 3DES'e göre kullanımı daha çok tercih edilmiş ve günümüz yazılım geliştirme alanlarında kullanılan en popüler şifreleme algoritmalarından biri olmuştur [39].

3.5. RSA Algoritması

Ron Rivest, Adi Shamir, Leonard Adleman tarafından 1977'de geliştirilmiştir. Algoritma adını geliştiricilerinin soyadlarının ilk harflerinden almaktadır. Asimetrik şifreleme yöntemini kullanmaktadır. Bu algoritmanın güvenliği tam sayılarda çarpanlara ayırma probleminin kolay olmaması problemine dayanır. Aralarında asal iki büyük sayının seçilmesi, bu sayılardan elde edilen yüksek mod değerinin

kullanılması ve çarpanlarına ayırma işlemi temeline dayanmaktadır [41]. E-imza, kimlik doğrulama ve güvenli anahtar paylaşımı işlemlerinde kullanılmaktadır [27].

Asimetrik şifreleme yöntemine göre çalıştığından dolayı alıcıya şifreli mesaj gönderilebilmesi için alıcının açık anahtarına ihtiyaç vardır. Açık anahtar gizli değildir, herkes tarafından bilinir. Gönderici, açık anahtarı ile mesajı şifreler ve alıcıya gönderir. Alıcının mesajı okuyabilmesi için gizli anahtar olması gerekir. Önceden üretilmiş olan bu gizli anahtar ile şifreli mesajı çözer ve açık metne ulaşır [27].

Anahtar oluşturma algoritması:

- İki tane farklı, rastgele ve yaklaşık aynı uzunlukta olan p ve q asal sayıları seçilir. Daha sonra denklem 3.1'deki formüle göre p ve q'nun çarpımından oluşan n değeri ve (p-1)*(q-1) çarpımlarından oluşan φ değeri hesaplanır (φ: Euler phi fonksiyonu/Totient fonksiyonu).

$$n = p * q \; ve \; \varphi = (\mathrm{p} - 1) * (\mathrm{q} - 1) \tag{3.1}$$

- 1'den büyük φ 'den küçük yani 1<e< φ olan ve φ ile aralarında asal olan yani gcd(e, φ)=1 olacak şekilde rastgele bir e sayısı seçilir (gcd: en büyük ortak bölen).
- 1<d< φ olacak şekilde denklem 3.2'deki formüle göre d sayısı elde edilir. Buradan elde edilen d, gizli anahtardır.

$$\mathrm{e} * \mathrm{d} \equiv 1 \bmod (\varphi) \tag{3.2}$$

- Tüm işlemlerin sonucunda açık anahtar (n,e) çifti ve gizli anahtar d hesaplanmış olur. Hesaplanan n ve e değeri mesajı şifreleyecek olan kişiye iletilir [38].

Şifreleme algoritması:

- Gönderici, öncelikle alıcının açık anahtarı olan (n,e) çiftini alır.
- Şifrelenecek olan m mesajı, [0,n-1] aralığında tam sayıya dönüştürülür.
- Denklem 3.3'de verilen formüle göre şifrelenecek m mesajının e'ninci kuvveti alınıp mod n'ye göre karşılığı hesaplanarak c değeri hesaplanır (c: şifreli mesaj, m: açık mesaj, e: anahtar oluşturma algoritmasından elde edilen rastgele bir değer, n:

anahtar oluşturma algoritmasından elde edilen iki tane farklı, rastgele ve aynı uzunlukta olan sayıların çarpımı)

$$c \equiv m^e (mod\ n) \tag{3.3}$$

- Gönderici oluşturulan c şifreli mesajını alıcıya gönderir [38].

Şifre çözme algoritması:

- d gizli anahtarı kullanılarak c şifreli mesajından açık mesajı bulabilmek için alıcı denklem 3.4'de verilen işlemi uygular (m: açık mesaj, c: şifreli mesaj, d: gizli anahtar, n: anahtar oluşturma algoritmasından elde edilen değer) [38].

$$m \equiv c^d (mod\ n) \tag{3.4}$$

RSA algoritmasının güvenliğini artırmak için şifreleme yönteminde kullanılan asal sayıların değeri artırılır. Dolayısıyla sayının değeri arttıkça şifreleme ve şifre çözme işlemlerinin süresi uzamaktadır. Bu yüzden çok büyük mesajları şifreleme işleminde uygun görülmez. Şifreyi çözme süresi büyük şifre çözme üssüne sahip olduğundan dolayı şifreleme işleminden daha uzun sürer [27].

Şifreleme ve şifre çözme sürelerini düşürmek için zamanla birçok farklı algoritma geliştirilmiştir. Bu algoritmalardan birisi Montgomery tarafından geliştirilen Montgomery algoritmasıdır. Montgomery algoritmasında kullanılan anahtar boyutunun büyük olması şifreleme ve şifre çözme işleminin hızını artırmıştır. RSA'da 256, 512, 1024 ve 2048 bit uzunluklarında anahtar kullanılabilmektedir [27].

3.6. SQL Server Kolon Seviyesinde Şifreleme

Veritabanında tablolarda tutulan verilerin tamamının değil, bir veya birkaç kolonun şifrelenmesi işlemine ihtiyaç duyulması halinde kolon düzeyinde şifreleme yöntemi tercih edilebilir. Şifrelenen veriler hem bellek hem de diskte tutulur. Bu sayede veriler üzerinde okuma yapılmak istenildiğinde şifrelenmiş olarak görünür, yazma yapılmak istenildiğinde ise veriler açık metin olarak eklenemez. Veriler üzerinde her okuma/yazma yapılmak istenildiğinde tercih edilen şifreleme algoritması için kullanılan nesnelere ihtiyaç duyulur. Şifreli veriler sadece bu nesnelere özel metotlarla çözülebilir [42].

Bu yöntemle veriler, verileri doğrudan okuyan herkesin gözünden saklanmış olur. Belli bir alanı şifrelemek için kullanılır bu yüzden işlemciye (CPU) çok fazla yük binmemiş olur [16].

Kolon seviyesinde şifreleme yönteminde simetrik anahtar (symmetric key), asimetrik anahtar (asymmetric key), T-SQL fonksiyonları (T-SQL functions) veya sertifikalar (certificates) kullanılabilir [16].

3.6.1. T-SQL fonksiyonları

Şifreleme işleminde 128 bitten oluşan 3DES algoritması kullanılır. Bu yöntemde verileri şifrelemek için "EncryptByPassPhrase" metodu, şifreli verileri çözmek için ise "DecryptByPassPhrase" metodu kullanılır. Metotlar en az 2, en fazla ise 4 parametre ile çalışmaktadır [16].

Metotlarda kullanılan parametrelerin kısaca tanımı:

@passphrase: Şifreleme işleminde kullanılan herhangi bir şifredir. Char, nvarchar, binary, varbinary veya nchar veri tipleri kullanılabilir. Bu şifreden arka planda bir simetrik anahtar elde edilir.

@cleartext: Şifrelenecek olan açık metindir. @passphrase parametresinde kullanılan veri tipleri burada aynı şekilde kullanılabilir.

@chippertext: Varbinary tipinde elde edilmiş olan şifreli veridir. Veri tipi varbinary olduğu için en fazla 8000 byte olabilir.

@add_authenticator: Tamsayı (int) tipinde kullanılır. Açık metinle birlikte ek bir bilgi tutulmak istenildiğinde 1 değeri atanır.

@authenticator: @add_authenticator parametresinin 1 olarak atanması durumunda açık metinle birlikte gizlenecek olan veriye tekabül eder [16].

3.6.2. Simetrik anahtarlar

Verilerin şifrelenmesi ve çözülmesinde aynı ortak anahtar kullanılır. SQL Server 2012 üzerinde en hızlı çalışan şifreleme yöntemlerinden biridir. Simetrik anahtar sisteminin şifre, sertifika, asimetrik anahtar veya başka bir simetrik anahtar kullanılarak korunması gerekmektedir. Simetrik anahtar yönteminde DES, TRIPLE_DES, TRIPLE_DES_3KEY, RC2, RC4, AES_128, AES_192, AES_256 algoritmaları kullanılabilir. Bu algoritmalar arasında en güçlüsü AES_256'dır [42]. Şekil 3.12'de AES 256 algoritmasıyla oluşturulan simetrik anahtarın T-SQL komutları verilmiştir.

```
CREATE SYMMETRIC KEY Symmetrickey
WITH ALGORITHM=AES_256
ENCRYPTION BY PASSWORD='simetrik_key_sifresi';
```

Şekil 3.12. AES 256 algoritmasıyla simetrik anahtar oluşturma.

Simetrik anahtarı açmak ve şifreli veriler üzerinde okuma yapabilmek için kullanılması gereken T-SQL komutları Şekil 3.13'de verilmiştir.

```
OPEN SYMMETRIC KEY Symmetrickey
DECRYPTION BY PASSWORD='simetrik_key_sifresi';
```

Şekil 3.13. Simetrik anahtar ile şifrelenen verileri açma.

Açılan simetrik anahtarı kapatmak için kullanılan T-SQL komutu Şekil 3.14'de verilmiştir. Bu komut çalıştırılarak anahtar kapatılmasa bile oturum (session) sonlandığında anahtar da kapanır.

```
CLOSE SYMMETRIC KEY Symmetrickey
```

Şekil 3.14. Simetrik anahtarı kapatma.

Açıkta kalan anahtarları görüntüleyen T-SQL komutları Şekil 3.15'de verilmiştir.

```
SELECT * FROM sys.openkeys;
```

Şekil 3.15. Açıkta kalan anahtarları görüntüleme.

Verilerin şifrelenmesi için "EncrypByKey", deşifrelenmesi için ise "DecryptByKey" metodu kullanılır. EncrypByKey metodu en az 2, DecryptByKey metodu ise en az 1 parametre almaktadır [16].

Metotlarda kullanılan parametrelerin kısaca tanımı:

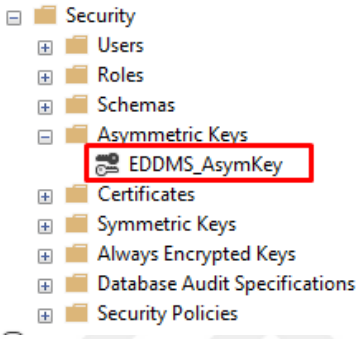
@Key_guid: Simetrik anahtarın Genel Benzersiz Tanımlayıcı (GUID) çıktısıdır. "KEY_GUID" metodu ile elde edilir.

@Clear_text: Şifrelenecek olan açık metindir. Nvarchar, char, varchar, binary, varbinary veya nchar tiplerinde tutulabilir.

@Ciphertext: Simetrik anahtar ile şifrelenmiş olan verilerdir [16].

3.6.3. Asimetrik anahtarlar

Asimetrik anahtarlarda (asymetric keys) gizli ve açık olmak üzere iki anahtar kullanılmaktadır. Şifreleme işlemi için açık anahtar, şifreyi çözme işlemi için ise gizli anahtar kullanılır. Asimetrik anahtarlar DMK tarafından şifrelenir bu yüzden önce DMK oluşturulmalıdır. Eğer DMK oluşturulmaz ise yine de asimetrik anahtar oluşturulur fakat bu asimetrik anahtarın korunması gerekir [42]. Oluşturulan asimetrik anahtar ilgili veritabanında "Security" dizini altında "Asymmetric Keys" klasörü içerisinde görüntülenebilir. Bu yöntemle RSA 512, RSA 1024 ve RSA 2048 olmak üzere 3 çeşit şifreleme algoritması kullanılabilir [16].



Şekil 3.16. Asimetrik anahtarın yer aldığı dizin.

Şekil 3.17'de RSA 2048 algoritmasıyla oluşturulan asimetrik anahtarın komutları verilmiştir. Bu komutta şifre ataması yapılmadığı için gizli anahtar DMK tarafından şifrelenir [16].

```
CREATE ASYMMETRIC KEY AsimetrikAnahtar
WITH ALGORITHM=RSA_2048
```

Şekil 3.17. RSA 2048 ile asimetrik anahtar oluşturma.

Şekil 3.18'de şifre kullanılarak RSA 2048 algoritmasıyla oluşturulan asimetrik anahtarın komutları verilmiştir. Bu komutun en sonunda yer alan "ENCRYPTION BY PASSWORD" ifadesi şifreleme yapıldığı anlamına gelir. Eğer bu atama yapılmazsa DMK tarafından şifreleme yapılır. Atanan şifre SQL Server'ın kurulu olduğu cihazdaki Windows Şifre Politikası (Windows Password Policy) gereksinimlerine uygun ve en fazla 128 karakterde olmalıdır [16].

```
CREATE ASYMMETRIC KEY AsimetrikAnahtar
WITH ALGORITHM=RSA_2048
ENCRYPTION BY PASSWORD='sifre'
```

Şekil 3.18. Şifre kullanarak RSA 2048 ile asimetrik anahtar oluşturma.

Oluşturulmuş olan asimetrik anahtarların görüntülenmesini sağlayan T-SQL komutu Şekil 3.19'da verilmiştir.

```
SELECT * FROM sys.asymmetric_keys
```

Şekil 3.19. Asimetrik anahtarları görüntüleme.

Şifreleme işlemi için "EncrypByAsymKey", deşifreleme işlemi için ise "DecryptByAsymKey" metodu kullanılmaktadır. EncrypByAsymKey metodu 2, DecryptByAsymKey metodu ise 3 parametre alır. Şifreleme işleminde ilk parametre asimetrik anahtarın kimlik numarası (id - identification number) değeri, ikinci parametre ise şifrelenecek açık veridir. Şifre çözme işleminde ise ilk parametre asimetrik anahtarın id değeri, ikinci parametre deşifrelenecek olan veri, son parametre ise şifre olarak kullanılan ifadedir [16].

Metotlarda kullanılan parametrelerin kısaca tanımı:

@ Key_ID: Int tipinde "ASYMKEY_ID" metoduyla elde edilen asimetrik anahtarın id değeridir.

@Clear_text: Şifrelenecek olan açık metindir. Nvarchar, char, varchar, binary, varbinary veya nchar tiplerinde tutulabilir.

@Ciphertext: Asimetrik anahtar ile şifrelenmiş olan verilerdir.

@Asym_Key_Password: Asimetrik anahtar oluşturulurken kullanılan şifredir [16].

Şekil 3.20'de daha önce oluşturulmuş olan "AsimetrikAnahtar" asimetrik anahtarıyla gerçekleşen şifreleme işlemi verilmiştir.

```
SET @acikVeri = 'Açık metin'
SET @sifreliVeri = ENCRYPTBYASYMKEY(
                        ASYMKEY_ID('AsimetrikAnahtar'),
                        @acikVeri
                    )
```

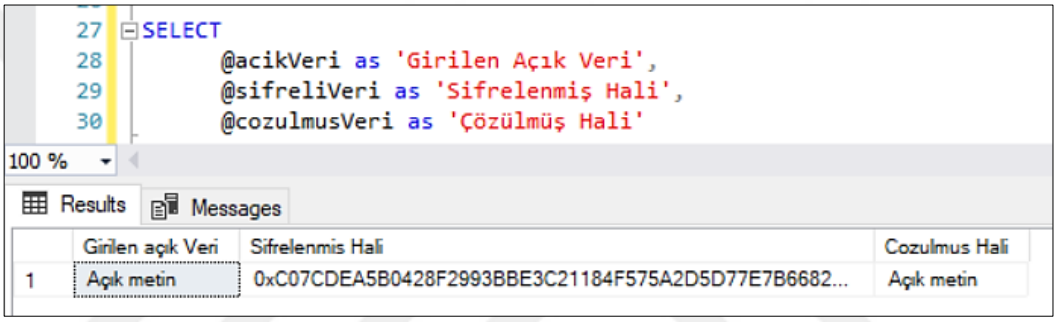
Şekil 3.20. Asimetrik anahtar ile şifreleme yapma.

Şekil 3.21'de "AsimetrikAnahtar" asimetrik anahtarı ve "sifre" şifresiyle şifreli verileri çözme işlemi verilmiştir.

```
SET @cozulmusVeri = DECRYPTBYASYMKEY(
                        ASYMKEY_ID('AsimetrikAnahtar'),
                        @sifreliVeri,
                        N'sifre'
                    )
```

Şekil 3.21. Asimetrik anahtar ile şifrelenen verileri çözme.

Şekil 3.22'de, Şekil 3.20 ve Şekil 3.21'de yapılan şifreleme ve şifre çözme işlemleri sonucunda elde edilen verilerin değerleri verilmiştir.



Şekil 3.22. Asimetrik anahtar ile şifrelenen ve şifresi çözülen verilerin sonuçları.

Asimetrik anahtarlarda simetrik anahtarlardaki gibi "open" ve "close" komutları kullanılmaz. Yani anahtarı açma veya kapama işlemi yapılmaz. Asimetrik anahtarlar simetrik anahtarlara göre daha yavaştır çünkü daha fazla sistem kaynağı kullanırlar. [42].

3.6.4. Sertifikalar

Sertifika, cihazların veya insanların tanınması için açık anahtar ile ilişkilendirilecek olan açık anahtarı içeren dijital olarak imzalanmış güvenlik nesnesidir. Veritabanı yansıtmada (Database mirroring), veri şifrelemede ya da paket gibi nesnelerin imzalanmasında kullanılmaktadır [16]. Oluşturulan sertifikalar veritabanında "Security" dizininde "Certicates" klasörü altında yer almaktadır.

4. MODELLEME

Dinamik, iki adımlı, farklı seviyelerde şifreleme ve şifre çözme işlemini gerçekleştiren bir veri kayıt sistemi geliştirilmiştir. Programlamada C# dili, Asp .Net Web Forms teknolojisi kullanılmış veritabanı olarak ise MS SQL tercih edilmiştir.

Config dosyasında tanımlanan parametreler sayesinde, uygulama içerisinde tek veya çift adımlı şifreleme yapılıp yapılmayacağı, eğer tek şifreleme işlemi yapılacak ise hangi katmanda ve seviyede şifreleme yapılacağı, blok şifreleme işlemi yapılacaksa hangi blok modeline göre şifreleme yapılacağı vb. işlemler tanımlanmaktadır. Bu sayede kullanıcının ihtiyacı doğrultusunda, istenilen adım kadar istenilen seviyede şifreleme gerçekleşecektir. Uygulama varsayılan olarak iki kez şifreleme yapılacak ve blok şifreleme algoritmasında CFB modeli kullanılacak şekilde ayarlanmıştır.

İlk şifreleme yönteminde simetrik blok şifreleme modeli olan AES 256 algoritması tercih edilip şifreleme ve şifre çözme işleminin arkayüzde (backend) gerçekleşmesi sağlanmıştır. İkinci şifreleme yönteminde ise asimetrik şifreleme yöntemi olan RSA 2048 algoritması tercih edilip veritabanı seviyesinde kolon düzeyinde şifreleme işlemi yapılması sağlanmıştır.

AES algoritmasında kullanılacak olan anahtar bilgisi, kullanıcı tarafından arayüzden belirlenmektedir ve kullanıcının oluşturduğu her kayıt kümesi için farklı anahtar kullanılabilmektedir.

Veriler, kullanıcının belirlediği anahtar ile bir sefer AES algoritmasıyla şifrelendikten sonra, şifreli veriler aynı zamanda ikinci kez veritabanına kolon düzeyinde şifreli olarak işlenecektir. Bu durumda veritabanında okuma yapılmak istenildiğinde şifreli olan verilerin ikinci farklı bir algoritmayla şifrelenmiş hali görünecektir.

Verilerin açık halini okuyabilmek için önce veritabanı seviyesinde RSA algoritmasına göre şifre çözme işlemi yapılması, ardından arkayüzde kullanılan AES algoritmasına göre gizli anahtarın tespit edilip şifre çözme işleminin yapılması gerekecektir. AES algoritmasının CFB modeline göre gerçekleşmesi ve başlatma vektörünün kullanılmasından dolayı birbirinin aynısı olan veriler aynı anahtar ile şifrelense bile

her seferinde farklı şifreli metin oluşacaktır. Bu durum saldırganın açık metinlere ulaşmasını zorlaştıracak, güvenliği artıracaktır.

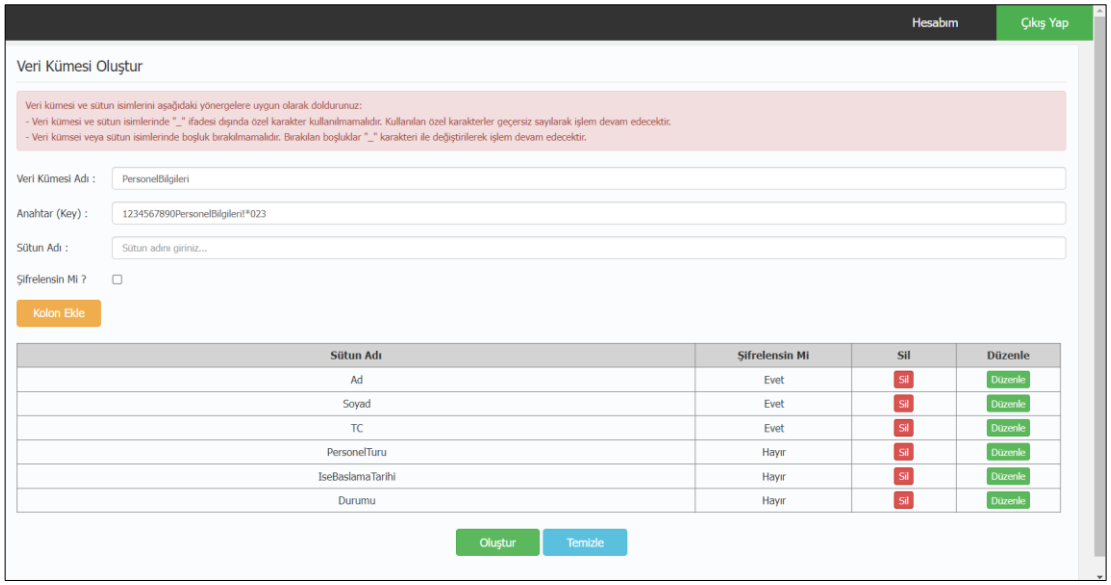
4.1. Uygulama Arayüzü ve Çalışma Mimarisi

Uygulama temel olarak 4 ana bölümden oluşmaktadır. Bunlar; yeni veri kümesi oluşturma, oluşturulmuş olan veri kümeleri üzerinde şema ve veri bazında işlemlerin yapılması ve kullanıcı hesabına ilişkin işlemlerin yapıldığı hesap bölümleridir.

Kullanıcı veri kümesini oluştururken hangi verilerin şifreli veya şifresiz tutulacağını seçebilmektedir. Tüm verilerin doğrudan şifrelenerek tutulması tercih edilmemiştir çünkü kullanıcının oluşturduğu her veri hassas veya kişisel bir veri olmayabilir. Ayrıca her veriyi şifreleyerek tutmak performansı düşüreceği için şifrelenecek verilerin tercihi kullanıcıya bırakılarak, bu sayede sadece gerekli olan verilerin şifrelenmesi sağlanmıştır.

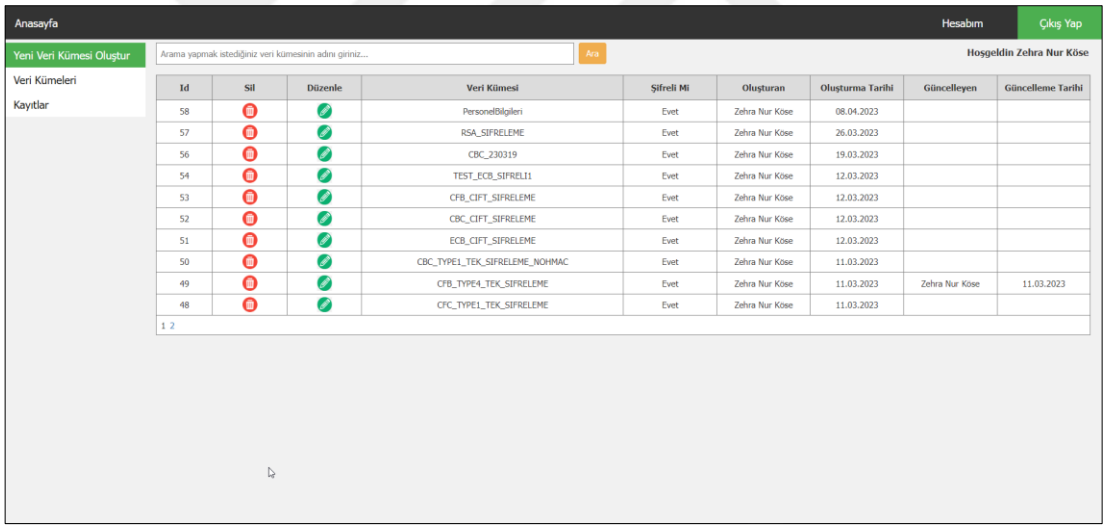
Eğer kullanıcının oluşturacağı veri kümesinde şifrelenecek sütunlar olacaksa ilk şifreleme işlemi olan AES algoritması için bir gizli anahtar kullanılacaktır. Bu anahtarı kullanıcı arayüzden girerek kendisi oluşturmaktadır. Kullanılan algoritma AES 256 algoritması olduğu için 256 bit uzunluğunda anahtar kullanılacaktır bu nedenle kullanıcıların arayüzden 32 karakter uzunluğunda bir anahtar bilgisi girmesi gerekmektedir. Girilen 32 karakter anahtar bilgisi daha sonra kullanıcının oluşturmuş olduğu veri kümesinde verileri şifrelemek veya şifre çözmek için kullanılacaktır.

Şekil 4.1’de veri kümesinde şifrelenecek alanların seçildiği ve AES şifreleme yönteminde kullanılacak olan simetrik anahtarın kullanıcı tarafından belirlendiği veri kümesi oluşturma ekranı verilmiştir.

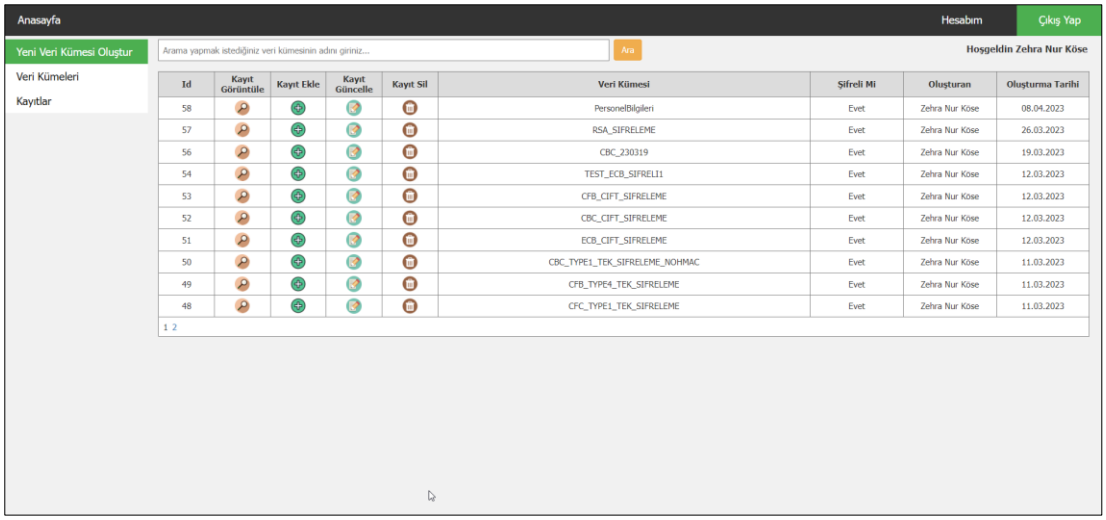


Şekil 4.1. Veri kümesi oluşturma ekranı.

Şekil 4.2'de veri kümeleri üzerinde şema işlemlerinin yapıldığı ekran, Şekil 4.3'de ise veri kümeleri üzerinde veri bazında işlemlerin yapıldığı ana ekran verilmiştir.

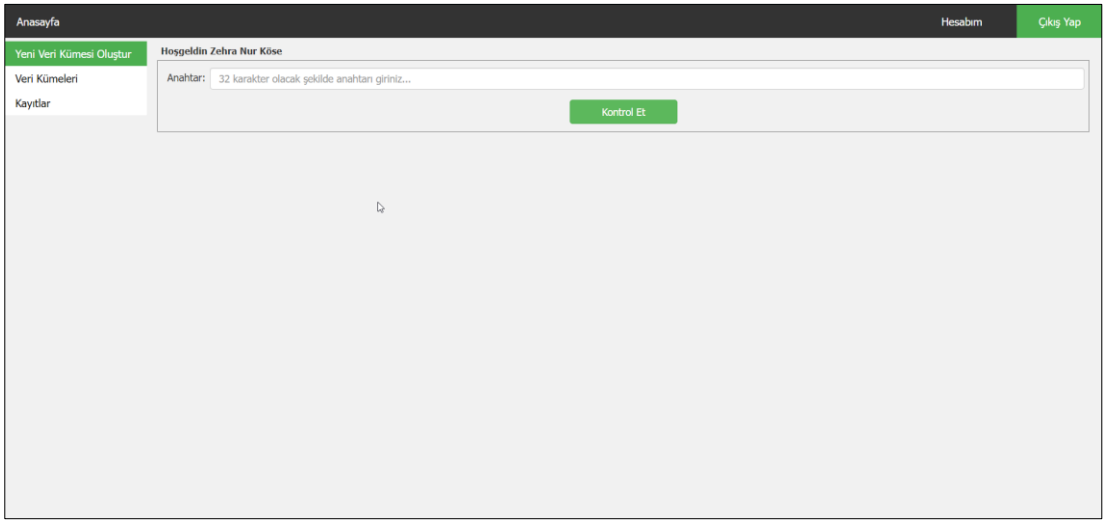


Şekil 4.2. Veri kümelerin listelendiği ve üzerinde şema işlemlerinin yapıldığı ekran.



Şekil 4.3. Veri kümelerinin listelendiği ve üzerinde veri bazında okuma/yazma işlemlerinin yapıldığı ana ekran.

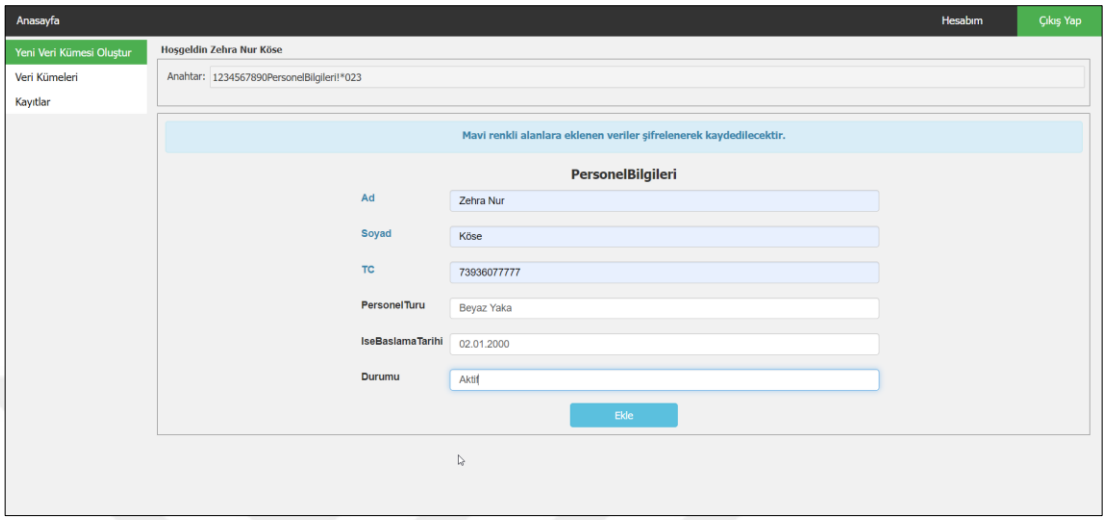
Eğer oluşturulan veri kümesi içerisinde şifrelenecek sütunlar var ise veri kümesine kayıt eklenirken doğrudan yazma işlemi yapılmasına izin verilmeyecektir. Önce kullanıcının ilgili veri kümesi için önceden belirlemiş olduğu anahtar bilgisinin girilmesi beklenmektedir. Şekil 4.4'de şifreli verilerin çözülmesi için daha önceden kullanıcı tarafından oluşturulmuş olan simetrik anahtar bilgisinin istendiği ekran verilmiştir.



Şekil 4.4. Şifreleri veriler üzerinde işlem yapabilmek için simetrik anahtar bilgisinin alınması.

Anahtar bilgisinin doğruluğu teyit edilirse yeni kayıt oluşturma ekranı açılacaktır. Kullanıcın önceden oluşturmuş olduğu veri kümesindeki tüm sütun bilgileri görüntülenecek ve kullanıcı verileri ekleyerek yeni kaydı oluşturabilecektir. Burada

kullanıcının daha önceden şifreli olması gerektiğini belirlediği alanlar kayıt eklenirken veritabanına şifreli olarak kaydedilecektir. Diğer alanlar ise herhangi bir şifreleme işlemi yapılmadan açık metin olarak eklenecektir. Şekil 4.5'de veri kümesine kayıt ekleme ekranı verilmiştir.

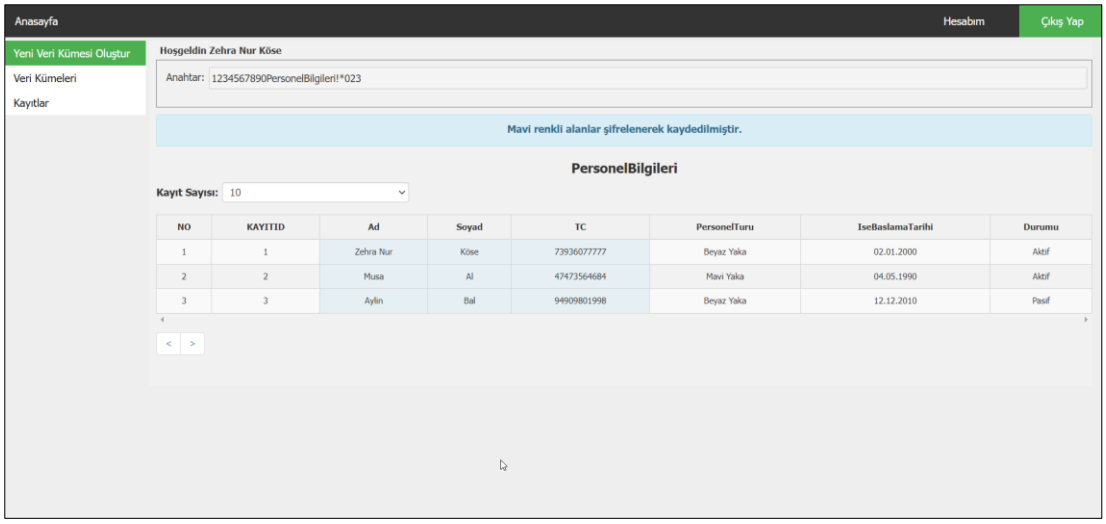


Şekil 4.5. Yeni kayıt ekleme ekranı.

İlk şifreleme işlemi kullanıcının belirlemiş olduğu simetrik anahtar ile AES algoritmasıyla gerçekleşecektir. Ardından bu şifreli veri tekrardan ikinci kez RSA algoritmasıyla veritabanı seviyesinde kolon düzeyinde şifrelenecektir.

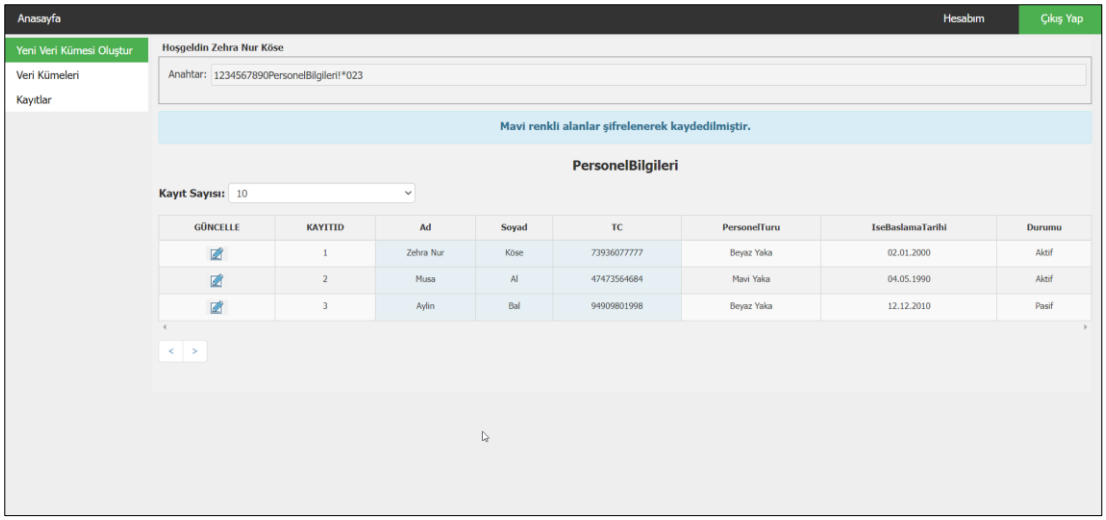
Kullanıcı eklenen kayıtları görüntülemek istediğinde, eğer ilgili veri kümesi içerisinde şifreli veriler var ise doğrudan okuma yapılmasına izin verilmeyecektir. Önce ilgili veri kümesi için kullanıcı tarafından daha önceden oluşturulmuş olan anahtar bilgisi istenilecek ve anahtarın doğru olması durumunda veriler listelenecektir.

Listeleme ekranında verilerin açık metin olarak görüntülenmesi için önce RSA 2048 ile veritabanı seviyesinde kolon düzeyinde şifre çözme işlemi yapılıp, ardından arkayüzde (backend) AES 256'ya göre kullanıcının belirlemiş olduğu anahtar ile şifre çözme işlemi yapılmaktadır. Şekil 4.6'da şifreli verilerin çözülerek şifresiz verilerin ise herhangi bir çözme işleminden geçirilmeden listelendiği ekran verilmiştir.

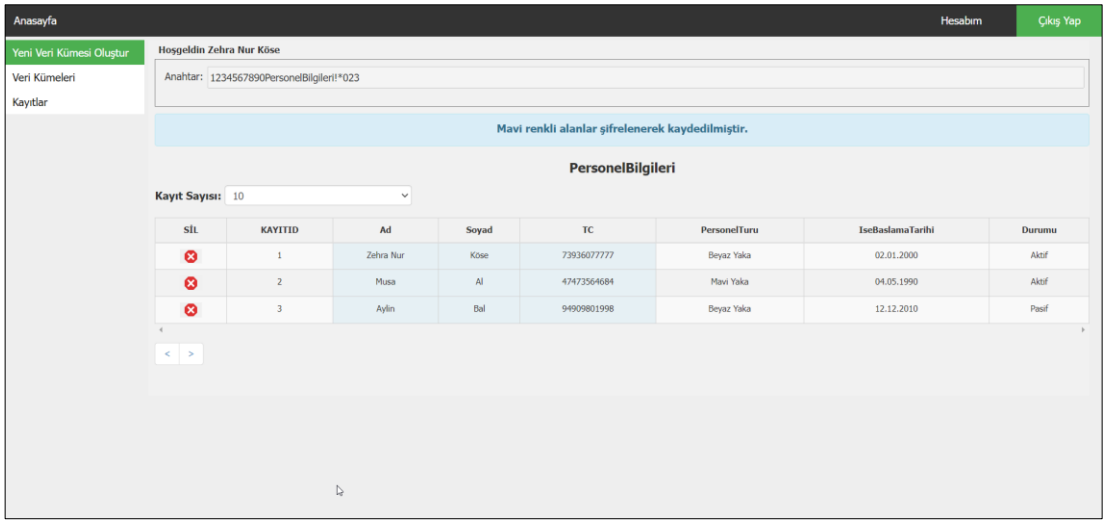


Şekil 4.6. İlgili veri kümesinde şifreli verilerin çözülerek listelendiği ekran.

Güncelleme ve silme işlemlerinde içerisinde şifreli veri bulunan herhangi bir veri kümesinde işlem yapılıyor ise doğrudan güncelleme veya silmeye izin verilmeyecektir. Kullanıcı anahtar bilgisini girip doğruluğu teyit edildikten sonra işlem yapılmasına izin verilecektir. Şekil 4.7'de kayıt güncelleme, Şekil 4.8'de ise kayıt silme işlemlerinin yer aldığı ekranlar verilmiştir. Eğer işlem yapılan veri kümesinde şifreli veriler var ise bu ekranlarda önce simetrik anahtar istenecek ardından anahtarın doğru olması sonucunda işlem yapılmasına izin verilecektir.



Şekil 4.7. Veri kümesinde kayıt güncelleme.



Şekil 4.8. Veri kümesinden kayıt silme.

İçerisinde şifreli verilerin yer aldığı veri kümeleri oluşturulurken kullanıcıdan anahtar bilgisi alındığı için her veri kümesine her kullanıcının erişmesi engellenmiş olmaktadır. Sadece doğru anahtarı bilen kullanıcılar ilgili kayıtlar üzerinde okuma/yazma işlemi yapabilmektedir. Uygulama içerisinde bu şekilde bir yetkilendirme mimarisi yürütülürken aynı zamanda profil bazlı yetkilendirme işlemi de yapılmaktadır.

Her kullanıcının sisteme giriş yapabilmesi için en az bir profile sahip olması gerekmektedir. Bu profillerin yeni kayıt ekleme, güncelleme, silme, görüntüleme ve listeleme işlemlerinden hangilerine yetkisinin olup olmadığı belirlenmektedir. Böylece kullanıcı sahip olduğu profillere göre uygulama içerisinde yetkisi doğrultusunda işlem yapabilmektedir. Şekil 4.9'da uygulamada kullanılmak üzere tanımlı olan profiller verilmiştir.

	PROFILKODU	PROFILADI	ACIKLAMA
1	admin	Sistem Yöneticisi	Her işeleme yetkisi var.
2	gyk	Genişletilmiş Yetkili Kullanıcı	Listeleme, görüntüleme ve yeni kayıt ekleme yetkisi var. Silme ve güncelleme yetkisi yok.
3	nyk	Normal Yetkili Kullanıcı	Sadece listeleme ve yeni kayıt ekleme yetkisi var.
4	tyk	Tam Yetkili Kullanıcı	Silme dışında tüm işlemlere yetkisi var.

Şekil 4.9. Uygulamada kullanılmak üzere tanımlanan profiller.

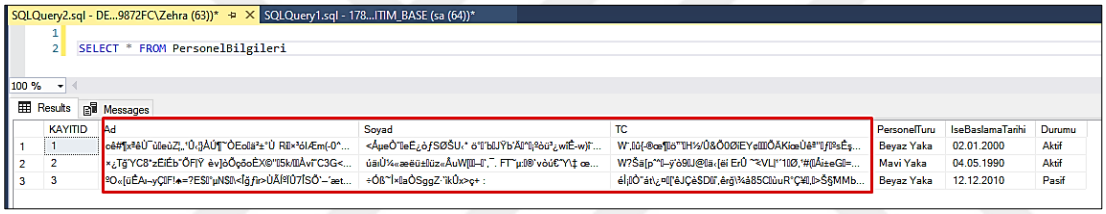
Şekil 4.10'da uygulamada tanımlanmış olan profillerin yetkileri verilmiştir. "1" olan alanlar ilgili işleme yetkisi var, "0" olan alanlar ise yetkisi yok demektir. Örneğin admin profilinin listeleme, güncelleme, yeni kayıt oluşturma, görüntüleme ve silme olmak üzere tüm işlemlere yetkisi bulunmaktadır.

	PROFILKODU	LIST	EDIT	NEW	VIEW	DELETE
1	admin	1	1	1	1	1
2	nyk	1	1	0	0	0
3	gyk	1	1	0	1	0
4	tyk	1	1	1	1	0

Şekil 4.10. Tanımlanan profillerin yetkileri.

4.2. Şifreli Verileri Çözme

Veritabanı seviyesinde kolon düzeyinde asimetrik anahtar kullanılarak RSA 2048 şifreleme işlemi yapıldığı için veritabanına erişim yetkisi olan kişiler dahil doğrudan okuma işlemi yapamayacaktır. Veriler SQL Server'da görüntülenmek istenildiğinde doğrudan açık metin olarak listelenmeyecek, verilerin şifreli halleri görünecektir. Şekil 4.11'de veritabanı üzerinde okuma yapılmak istenildiğinde elde edilen şifreli veriler verilmiştir.

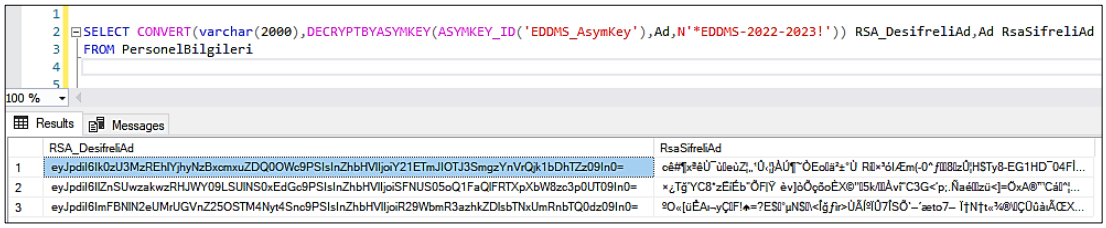


Şekil 4.11. Asimetrik anahtar ile şifrelenen verileri veritabanı üzerinde görüntüleme.

Verilerin açık halini görebilmek için SQL Server'da oluşturulmuş olan şifre ve asimetrik anahtarın bilgisi girilerek şifre çözme işlemi gerçekleşmelidir.

Şekil 4.12'de "DECRYPTBYASYMKEY" metodu ile şifreli verilerin çözülmüş hali verilmiştir. Bu şekle göre verilerin RSA algoritmasıyla çözülmüş hali de şifrelidir çünkü şifreleme iki adımdan oluşmaktadır ve ilk şifreleme işlemi arakayüzde ikinci şifreleme işlemi ise veritabanı katmanında gerçekleşmiştir. Burada SQL Server'da yapılan şifre çözme işlemi sadece veritabanı katmanındaki verileri çözmektedir. Yani SQL Server'da yapılan şifre çözme işlemi sonucunda elde edilen veriler, açık metinlerin AES algoritmasıyla şifrelenmiş halidir. Şifreli verileri tamamen çözebilmek için ikinci kez AES 256 algoritmasıyla şifre çözme işlemi yapılmalıdır. RSA şifreleme yöntemi veritabanı seviyesinde gerçekleştiği için SQL Server üzerinde şifre çözme işlemi yapılabilmektedir fakat AES şifreleme işlemi arkayüzde gerçekleştiği için SQL Server üzerinde doğrudan şifre çözme işlemi yapılamaz. Bir kez RSA ile şifre çözme

işleminden geçen verilerin AES algoritması ile kullanılan anahtar tespit edilip tekrardan şifre çözme işlemine tabi tutulması gerekmektedir.



Şekil 4.12. İki adımda şifrelenen verilerin asimetrik anahtarla çözülmesi.

5. ARAŞTIRMA BULGULARI

5.1. Şifreleme Algoritmalarının Performans Analizi

Açık metinler ile tek şifreleme işleminden geçen metinler ve çift şifreleme işleminden (önce AES sonra da RSA algoritmasıyla şifreleme) geçen metinler arasında performans analizi yapılmıştır. Performans analizi şifre çözme hızına (zaman karmaşıklığı) göre değerlendirilmiştir. AES algoritması C# dilinde gerçekleştirilmiş ve "System.Security.Cryptography" kütüphanesi kullanılmıştır. RSA algoritması SQL Server 2019'da asimetrik anahtar kullanılarak veritabanı seviyesinde kolon düzeyinde şifreleme yapılarak gerçekleştirilmiştir.

AES algoritmasının blok uzunluğu 128 bit, anahtar uzunluğu ise 256 bit olarak tercih edilmiştir. AES algoritmasıyla ECB, CBC ve CFB blok modellerine göre şifreleme ve şifre çözme işlemleri yapılıp, aralarındaki performans farkı incelenmiştir. RSA algoritmasında ise 2048 bit uzunluğunda anahtar kullanılmıştır.

Şifreleme işleminde aynı şema ve veri yapısına sahip tablolar kullanılmıştır. Tabloların içerisindeki kayıtlar SQL Server üzerinde manuel olarak oluşturulmuştur. Oluşturulan her tablonun veri sayısı, kolon sayısı ve içeriğindeki veriler aynıdır. Böylelikle şifreleme işlemi veya şifre çözme işlemi yapılırken birbirinin aynısı olan açık veriler, aynı anahtar ve aynı algoritma kullanılarak şifrelenmiş, sadece şifreleme modeli farklılık göstermiştir. Performans/yük testi 3 adet örnek tablo üzerinde gerçekleştirilmiş olup, bu tablolardaki kayıtların tamamının veritabanından "Select" sorgusu ile çekilmesi sonucunda, şifreli verilerin çözülüp arayüze yansımasına kadar geçen zamana göre değerlendirme yapılmıştır. Bu veri kümelerinde 10000, 80000 ve 160000 adet kayıt mevcuttur.

Testlerin yapıldığı makinenin özellikleri:

İşlemci: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.10 GHz

RAM: 16 GB

Disk: 480 GB SSD

İşletim Sistemi: Windows 10 Pro

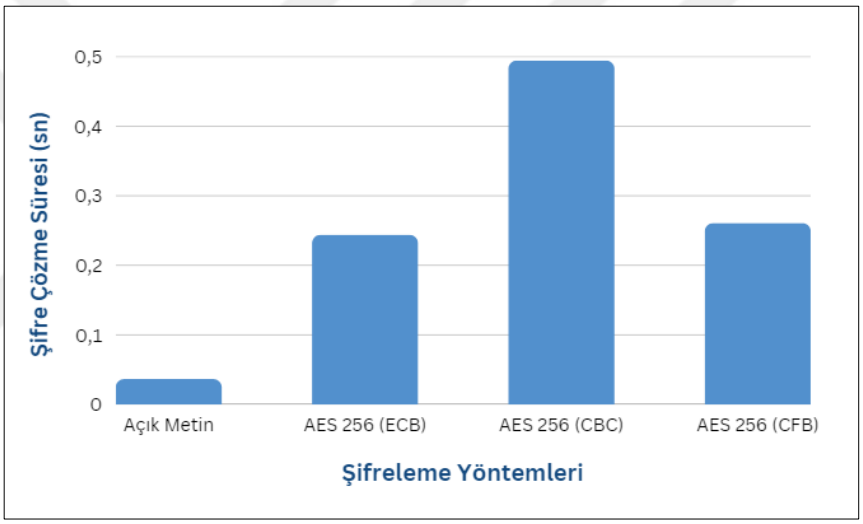
Algoritmaların örnek veri kümeleri üzerinde şifre çözmek için harcadığı zaman saniye (sn) cinsinden Tablo 5.1'de verilmiştir.

Tablo 5.1. Algoritmaların şifre çözme süreleri.

Şifreleme Tekrarı	AES 256 ECB Modu	AES 256 CBC Modu	AES 256 CFB Modu	RSA 2048	Kayıt Sayısı	Toplam Boyut (MB)	Çalışma Süresi (sn)
Açık Metin	-	-	-	-	10000	1.95	0.036
Açık Metin	-	-	-	-	80000	4.63	0.261
Açık Metin	-	-	-	-	160000	9.13	0.697
Tek Adımlı Şifreleme	+	-	-	-	10000	3.20	0.243
Tek Adımlı Şifreleme	+	-	-	-	80000	13.88	2.203
Tek Adımlı Şifreleme	+	-	-	-	160000	27.38	7.401
Tek Adımlı Şifreleme	-	+	-	-	10000	9.63	0.494
Tek Adımlı Şifreleme	-	+	-	-	80000	48.13	4.539
Tek Adımlı Şifreleme	-	+	-	-	160000	96.20	13.44
Tek Adımlı Şifreleme	-	-	+	-	10000	6.07	0.260
Tek Adımlı Şifreleme	-	-	+	-	80000	29.26	2.610
Tek Adımlı Şifreleme	-	-	+	-	160000	58.38	7.723
Tek Adımlı Şifreleme	-	-	-	+	10000	24.32	181.57
Tek Adımlı Şifreleme	-	-	-	+	80000	125.26	1488.22
Tek Adımlı Şifreleme	-	-	-	+	160000	250.26	4616.10
Çift Adımlı Şifreleme	+	-	-	+	10000	24.38	177.74
Çift Adımlı Şifreleme	+	-	-	+	80000	160.07	1451.61
Çift Adımlı Şifreleme	+	-	-	+	160000	250.26	4625.23
Çift Adımlı Şifreleme	-	+	-	+	10000	24.57	184.94
Çift Adımlı Şifreleme	-	+	-	+	80000	160.20	1499.31
Çift Adımlı Şifreleme	-	+	-	+	160000	250.32	4730.73
Çift Adımlı Şifreleme	-	-	+	+	10000	24.45	181.53
Çift Adımlı Şifreleme	-	-	+	+	80000	160.07	1485.67
Çift Adımlı Şifreleme	-	-	+	+	160000	250.26	4630.20

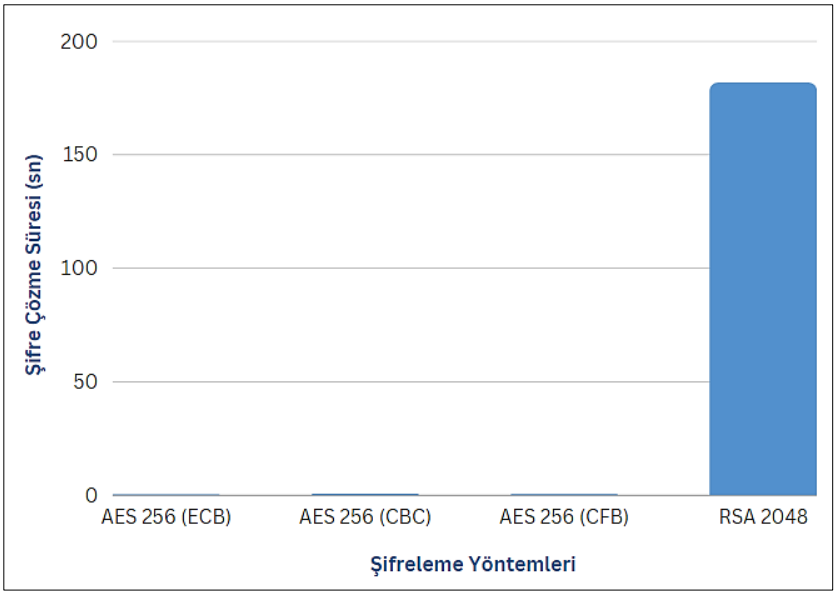
Tek adımda AES 256 algoritmasıyla şifrelenmiş olan verilerin şifre çözme hızı kıyaslandığında; ECB ve CFB blok modellerinin şifre çözme süresinin birbirine yakın olduğu, CBC modelinin ise diğer iki modele göre yaklaşık 2 kat daha uzun sürede çalıştığı gözlemlenmiştir. En kısa sürede çalışan blok modeli sırasıyla ECB, CFB ve CBC şeklinde olmuştur.

10000 adet kayıt üzerinde şifresiz veriler ile tek adımda AES 256 algoritmasıyla şifrelenen veriler kıyaslandığında, ECB ve CFB modellerinin şifresiz metinlere göre yaklaşık 7 kat daha uzun sürede çalıştığı, CBC modelinin ise şifresiz verilere göre yaklaşık 14 kat daha uzun sürede çalıştığı gözlemlenmiştir. Şekil 4.13'de 10000 adet kayıtta AES algoritmasının farklı blok modellerindeki şifre çözme süresi ve şifresiz verilerin çalışma süresi verilmiştir.



Şekil 5.1. Tek adımda AES algoritmasıyla şifrelenen verilerin şifre çözme süresi.

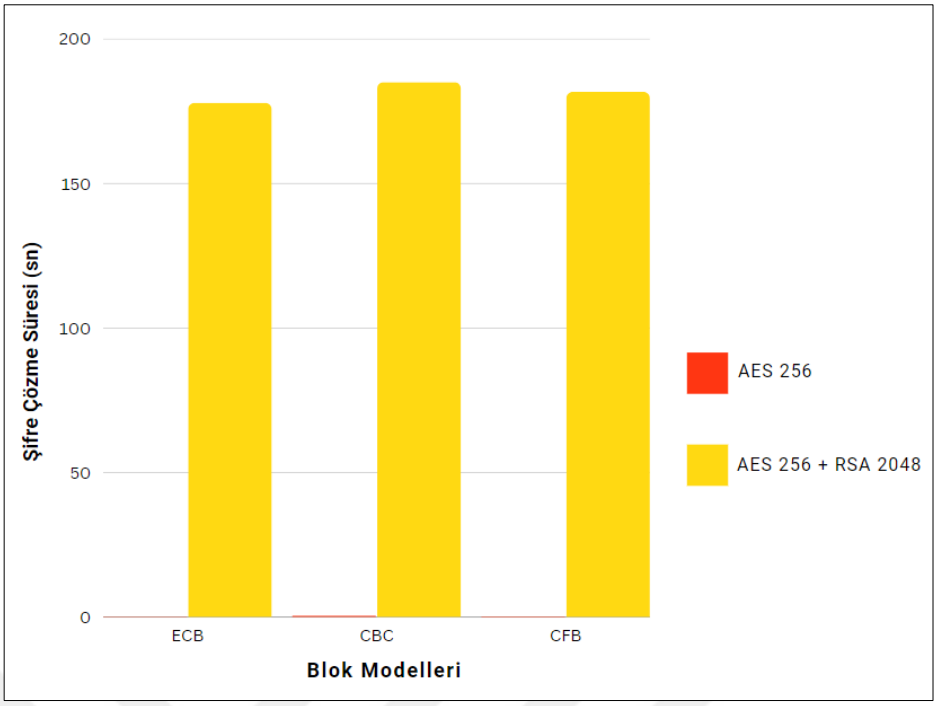
Tek adımlı şifrelemede tüm algoritmalar kıyaslandığında şifre çözme süreleri hızlıdan yavaşa doğru AES-256 ECB, AES-256 CFB, AES-256 CBC modeli ve RSA-2048 şeklinde olduğu gözlemlenmiştir. RSA-2048 algoritması 10000 adet kayıt üzerinde AES-256 ECB ve AES-256 CFB algoritmasına göre yaklaşık 726 kat, AES-256 CBC modeline göre ise yaklaşık 368 kat daha yavaş çalışmaktadır. RSA algoritması asimetrik şifreleme tekniğini kullandığı için AES algoritmasına göre daha yavaş çalışması beklenen bir durumdur. Şekil 4.14'de tek adımda şifreleme yapılan tüm algoritmaların (AES-ECB, AES-CBC, AES-CFB ve RSA) 10000 kayıtta gerçekleşen performans sonuçları verilmiştir. Bu sayede asimetrik şifreleme yönteminin simetrik şifrelere oranla daha yavaş çalıştığı kanıtlanmıştır.



Şekil 5.2. Tek adımda gerçekleşen şifreleme algoritmalarının performans sonucu.

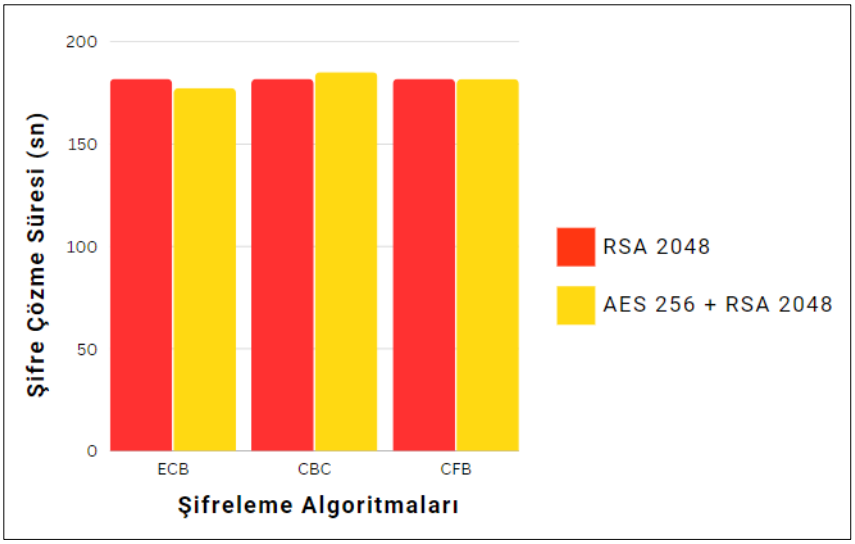
Önce AES 256 ardından veritabanı seviyesinde kolon düzeyinde RSA 2048 ile şifrelenmiş olan verilerin (2 adımlı şifreleme) şifre çözme hızı incelendiğinde, sonuçların tek adımlı şifrelemeden farklı olarak birbirine daha yakın olduğu gözlenmiştir. Çalışma süresi hızlıdan yavaşa doğru sıralandığında su şekilde olduğu görülmüştür: AES ECB+RSA, AES CFB+RSA, AES CBC+RSA.

10000 adet kayıt üzerinde çift şifreleme işlemi yapıldığında ECB modelinde performansın tek şifreleme işlemine göre yaklaşık 731 kat, CBC modelinde yaklaşık 374 kat, CFB modelinde ise yaklaşık olarak 698 kat düştüğü gözlenmiştir. Ayrıca 80000 kayıt üzerinde performans analizi yapıldığında performansın ECB modelinde yaklaşık 659 kat, CBC modelinde yaklaşık 330 kat, CFB modelinde ise yaklaşık 569 kat düştüğü görülmüştür. Şekil 4.15'de 10000 adet kayıtta çift şifreleme işleminden geçen verilerle tek adımda AES 256 ile şifrelenen verilerin şifre çözme süresi verilmiştir. Bu şekil incelendiğinde çift şifreleme işleminde performansı asıl düşüren unsurun RSA algoritması olduğu görülmüştür.



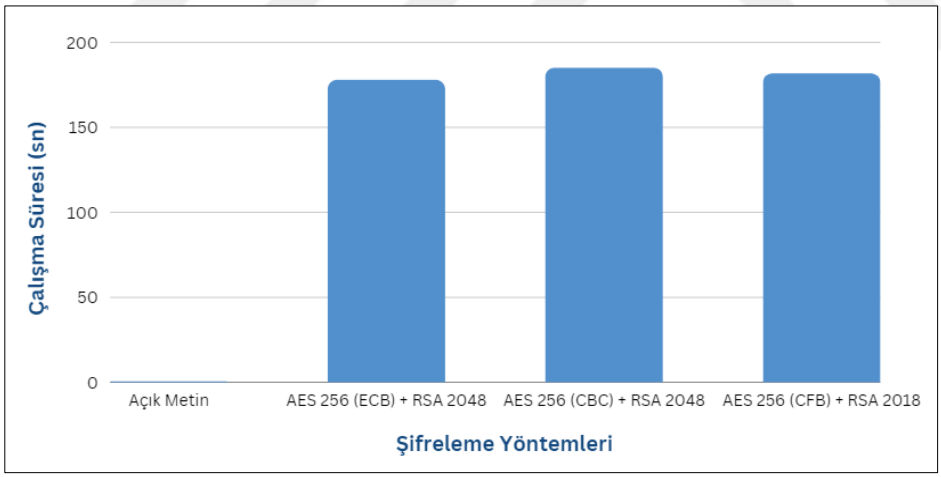
Şekil 5.3. Tek şifreleme ve çift şifreleme işleminden geçen verilerin şifre çözme süresi.

İki adımda gerçekleşen şifre çözme işlemi (AES 256 ardından RSA 2048) ve tek adımda veritabanı seviyesinde RSA 2048 algoritmasına göre gerçekleşen şifre çözme süresi kıyaslandığında çalışma sürelerinin birbirine yakın olduğu görülmektedir. Bu durumda çift şifreleme işleminde performansı düşüren asıl unsurun iki kez şifreleme yapılması değil ikinci şifreleme adımının asimetrik şifreleme yöntemini kullanmasından kaynaklı olduğu görülmektedir. Şekil 4.16'da 10000 kayıtta çift şifreleme işleminden geçen verilerle tek adımda RSA 2048 ile şifrelenen verilerin şifre çözme süresi verilmektedir.



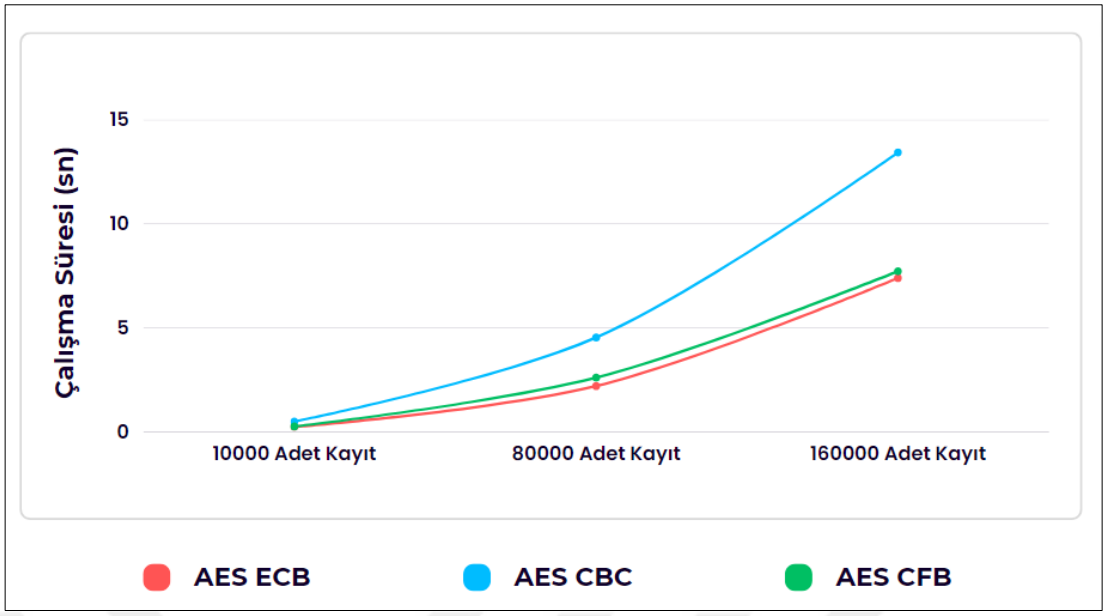
Şekil 5.4. Tek adımda gerçekleşen şifreleme (RSA 2048) ve çift adımda gerçekleşen şifreleme işlemlerinin performans sonucu.

Çift şifreleme işleminin çalışma süresi açık metinlerin çalışma süresi ile kıyaslandığında performansın 10000 kayıtta ECB modunda 4937, CBC modunda 5111, CFB modunda ise 5042 kat düştüğü gözlenmiştir. Şekil 4.17'de 10000 kayıtta çift şifreleme işleminden geçen verilerin şifre çözme süresi verilmektedir.



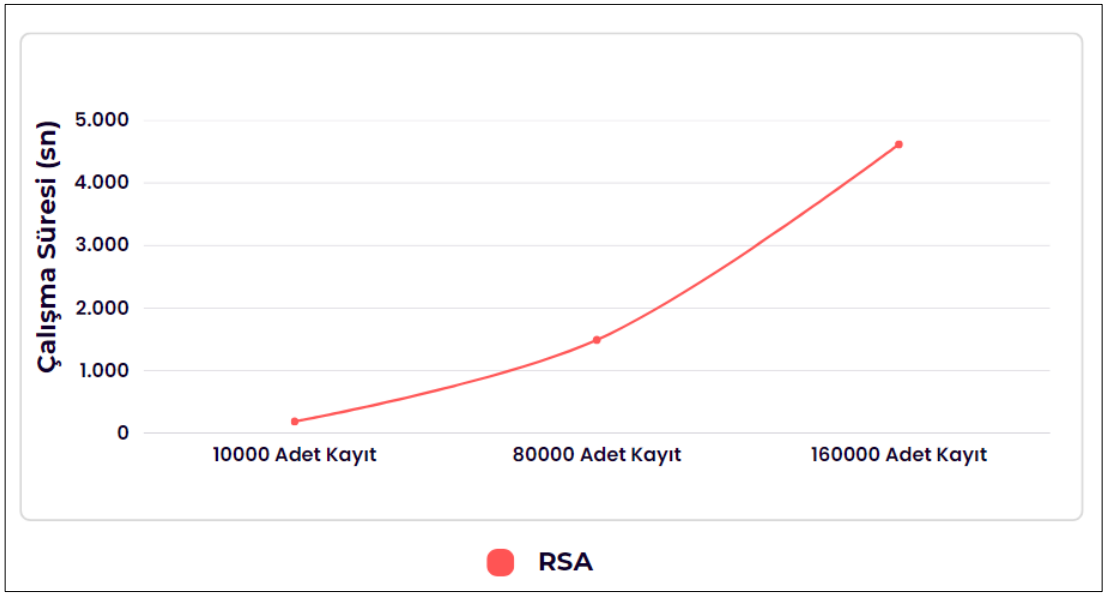
Şekil 5.5. Çift şifreleme işleminden geçen verilerin şifre çözme süreleri.

Tek adımda AES 256 algoritmasıyla şifrelenen veriler incelendiğinde ECB, CBC ve CFB blok modelinde 80000 adet kaydın 10000 adet kayda göre yaklaşık 9 kat daha yavaş, 160000 adet kaydın 80000 adet kayda göre ise yaklaşık 3 kat daha yavaş çalıştığı gözlenmektedir. Şekil 5.6'da AES algoritmasına göre tek adımda ECB, CBC ve CFB blok modellerinde şifrelenen verilerin farklı sayıdaki veri kümelerinde şifre çözme süresi verilmiştir.



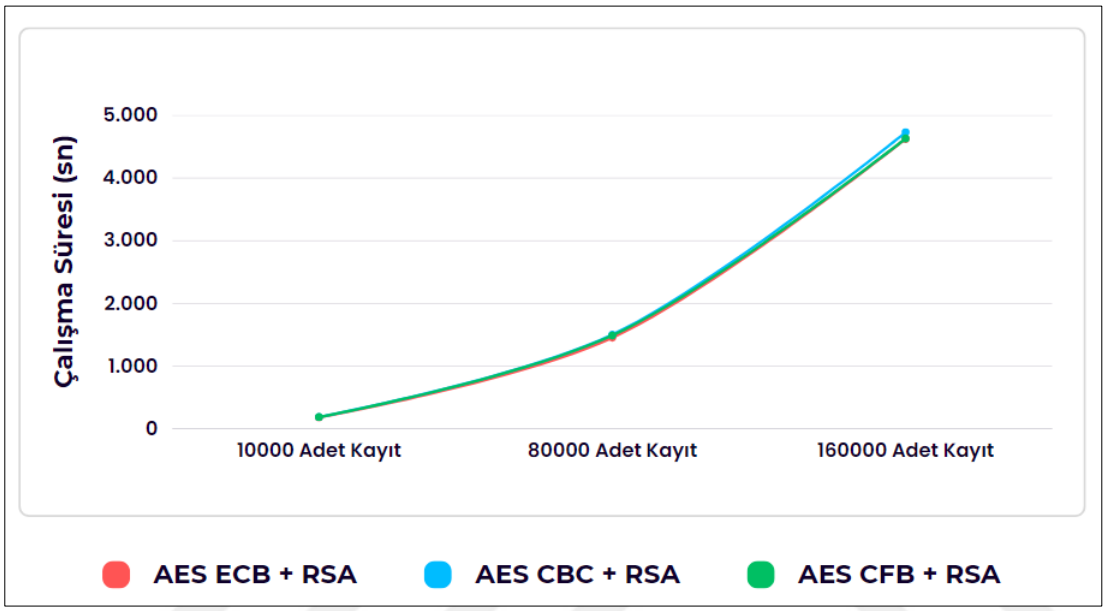
Şekil 5.6. Farklı veri kümelerinde AES 256 ile tek adımda şifrelenen verilerin performans sonucu.

Tek adımda RSA 2048 ile farklı veri kümelerinde gerçekleşen şifre çözme süreleri kıyaslandığında 80000 adet verinin çalışma süresinin 10000 adet veriye göre yaklaşık 8 kat, 160000 adet verinin ise 80000 adet veriye yaklaşık 3 kat daha yavaş sürede çalıştığı gözlenmektedir. Şekil 5.7'de RSA algoritmasıyla veritabanı katmanında kolon seviyesinde şifrelenen verilerin farklı veri kümelerindeki şifre çözme süresi verilmiştir.



Şekil 5.7. Farklı veri kümelerinde RSA 2048 ile tek adımda şifrelenen verilerin performans sonucu.

İki adımda art arda şifrelenen verilerin farklı veri kümelerindeki şifre çözme süresi incelendiğinde tüm şifreleme yöntemlerinde 80000 adet verinin 10000 adet veriye göre yaklaşık 8 kat yavaş, 160000 adet verinin 80000 adet veriye göre ise yaklaşık 3 kat yavaş çalıştığı gözlenmiştir. Şekil 5.8'de iki kez art arda gerçekleşen şifreleme (önce AES 256 daha sonra RSA 2048 algoritmasına göre şifreleme) işlemi sonucunda farklı veri kümelerindeki verilerin şifre çözme süresi verilmiştir.



Şekil 5.8. Farklı veri kümelerinde iki adımda şifrelenen verilerin performans sonucu.

5.2. Performans Sonuçları

Şifreleme işlemlerinin gerçekleştiği sistemlerde ekstra şifreli verileri çözme işlemi gerektiği için sisteme ek yük binmektedir. Bu durum beraberinde şifreli verilerin açık metinlere oranla daha yavaş çalışmasına yani performans düşüklüğüne sebep olmaktadır.

AES 256 algoritmasıyla şifrelenen veriler incelendiğinde bu şifreleme tekniğinin simetrik şifreleme yöntemi olması nedeniyle performansı çok fazla düşürmediği görülmektedir.

AES'in ECB, CBC ve CFB blok modelleri arasındaki performans farkı incelendiğinde CBC modelinin en yavaş çalışan model olduğu, ECB ve CFB modellerinin ise birbirine yakın sürelerde çalıştığı görülmektedir. ECB modelinde başlatma vektörü kullanılmadığı için, bloklar arasında bir ilişki olmadığı için ve en temel blok şifreleme modeli olduğu için diğer blok modellerine oranla güvenlik zafiyeti daha fazladır. Bu

durumda ECB modeli, CBC modeline göre daha hızlı ama daha güvensizdir. CFB modelinde ise başlatma vektörü sayesinde aynı açık metnin aynı anahtar ile şifrelenmesi sonucunda her seferinde farklı şifreli metnin elde edilmesi, ECB'nin aksine blokların arasında bir ilişkinin olması, ayrıca XOR işlemine tabi tutulması ECB'den daha güvenli bir model olduğunu kanıtlamaktadır. CFB modelinde de başlatma vektörünün kullanılması, blokların arasında ilişki olması ayrıca XOR işlemlerinin gerçekleşmesi nedeniyle CFB'nin ECB modeline göre daha güvenli bir şekilde çalıştığı kanıtlanmaktadır. CBC modeli, CFB modeline göre yaklaşık 2 kat daha yavaş çalışmaktadır. Bu nedenle 3 blok modeli arasında hem performans hem de güvenlik açısından CFB modelinin kullanılmasının daha uygun olduğu gözlenmiştir.

Art arda farklı seviyelerde farklı şifreleme tekniği kullanılarak 2 kez şifreleme yapılması sonucunda performansın büyük oranda düştüğü görülmüştür. İkinci şifreleme işleminde kullanılan algoritmanın (RSA 2048) asimetrik şifreleme tekniği olmasının bunda büyük payı vardır. Çünkü asimetrik şifreler simetrik şifrelere göre daha güvenli olmalarına rağmen daha yavaştır.

Bir kez RSA 2048 algoritmasıyla veritabanı seviyesinde gerçekleşen şifreleme işleminin performansı incelendiğinde ise şifre çözme süresinin iki adımlı şifrelemedeki süreye çok yakın olduğu görülmüştür. Bu durumda iki kez art arda şifreleme yapıldığında performansı düşüren en büyük unsurun şifreleme tekrarından ziyade ikinci şifreleme tekniğinde asimetrik şifreleme yöntemi kullanılmasından kaynaklı olduğu görülmektedir.

İki kez art arda gerçekleşen şifreleme işlemleriyle bir kez RSA 2048 ile gerçekleşen şifreleme işlemlerinin çalışma süreleri birbirine çok yakındır. Bunun sebebi ilk şifreleme tekniğinde simetrik şifreleme yöntemi olan AES algoritmasının kullanılmış olmasıdır. AES algoritması birkaç saniye civarlarında çalışmaktadır, RSA algoritması ise AES'e oranla çok daha yavaş çalışmaktadır. Bu nedenle AES algoritmasının şifre çözme süresine çok fazla etki etmediği görülmektedir. Bu durumda eğer asimetrik şifreleme tekniği kullanılarak bir şifreleme gerçekleşiyorsa, yanına ikinci kez simetrik şifreleme ile çalışan bir algoritma eklendiğinde, simetrik algoritmanın güvenliğe olumlu etkisi olurken aynı zamanda performansa da olumsuz bir etkisi olmayacaktır. Performansı asıl düşüren unsur asimetrik şifreleme olacaktır.

6. TARTIŞMA VE SONUÇ

Dijital dünyanın genişlemesi sonucunda kişisel ve hassas verilerin korunmasının önemi artmaktadır. Verileri korumak için en çok tercih edilen yöntemlerden biri verileri şifrelemektir. Burada önemli olan konu şudur; verilerin ne kadarının şifrelenmesi istenmektedir ve ne kadar güçlü şifreleme işlemi yapılması istenmektedir? Bu sorulara verilecek cevaplar kullanılacak şifreleme tekniklerini değiştirecektir.

Eğer hızlı bir sistem tasarlanıyorsa simetrik şifreleme yöntemlerinden biri tercih edilebilir. Daha güvenli bir sistem tasarlanıyorsa asimetrik şifreleme yöntemlerinden biri tercih edilebilir fakat bu yöntem simetrik yöntemlere göre daha yavaş çalışmaktadır.

Eğer veritabanı seviyesinde şifreleme yapılacaksa, veritabanının tamamının veya yedeklerinin şifrelenmeye ihtiyacının olup olmadığı tartışılmalıdır. Tüm veritabanının şifrelenmesine ihtiyaç yoksa kolon bazında şifreleme tekniği kullanılmalıdır. Aksi halde gereksiz yere tüm veriler şifrelenir, bu durum iş yükünün artmasına beraberinde performansın düşmesine sebep olur.

Veritabanı katmanında gerçekleşen şifreleme işlemlerinde veriler sadece saldırganın gözünden korunmakla kalmaz aynı zamanda veritabanına erişim yetkisi olan, sistemin bakımı yapan bilişim teknolojisi (IT) personeli dahil olmak üzere doğrudan veriler üzerinde okuma/yazma işlemi yapamaz. Bu durum da güvenliği artırır. Veriler üzerinde okuma/yazma yapabilmek için kullanılan şifreleme tekniğine uygun olan nesneler ve doğru anahtar bilgisi girilerek şifre çözme işlemi gerçekleşebilir.

Bu tez çalışmasında şifreleme teknikleri incelenmiş, farklı seviyelerde birbirinden farklı şifreleme tekniklerinin kullanıldığı ve farklı şifreleme tekrarlarının yapıldığı bir uygulama geliştirilerek elde edilen performans sonuçlarının somut bir şekilde analiz edilmesi sağlanmıştır.

Simetrik şifreleme algoritmalarının en çok tercih edilen ve en güçlü olanlarından biri AES algoritmasıdır. Bu algoritmanın aynı zamanda blok şifreleme yöntemini

kullanması ile farklı blok modelleri arasındaki performans farkı kıyaslanmıştır. AES simetrik şifreleme yöntemini kullandığı için ve simetrik şifreleme tekniğinin asimetrik şifreleme tekniğine göre daha hızlı olması nedeniyle şifre çözme işlemlerinde performansın çok fazla düşmediği görülmüştür.

AES'in ECB, CBC ve CFB blok modelleri arasında performans farkı incelendiğinde en hızlısının ECB, sonra sırasıyla CFB ve CBC şeklinde olduğu görülmüştür. Aralarında en fazla güvenlik zafiyeti barındıran model ECB, en yavaş olan ise CBC'dir. Bu nedenle bu üç blok modeli arasında şifreleme tercihi yapılması halinde en doğru seçim CFB modeli olacaktır.

RSA asimetrik şifreleme yöntemini kullanan bir algoritmadır. Dolayısıyla AES algoritmasına oranla daha yavaş çalıştığı görülmektedir. RSA şifreleme işlemi veritabanı seviyesinde kolon düzeyinde gerçekleştiği için şifre çözme işleminin de mutlaka veritabanı üzerinde yapılması gerekmektedir. Her iki şifreleme tekniğinin hem farklı seviyelerde hem de farklı yöntemlerle gerçekleşmesinden dolayı daha güvenilir bir yapı oluşmuştur.

İki kez art arda önce AES ardından RSA ile şifreleme yapıldığında performansı asıl düşüren unsurun şifreleme tekrarının fazlalığından ziyade ikinci şifreleme tekniğinde asimetrik yöntemin tercih edilmesinden kaynaklı olduğu görülmüştür. Simetrik şifreleme yöntemini kullanan AES, asimetrik şifreleme yöntemini kullanan RSA'ya göre çok daha hızlı çalışmaktadır.

Eğer hali hazırda asimetrik şifreleme tekniği kullanılıyorsa güvenliği artırmak için ekstra simetrik şifreleme tekniğinin eklenmesinde performans açısından çok fazla bir düşüş yaşanmayacaktır. Bu durum aksine güvenliği artıracaktır. Fakat yüksek hız istenilen ve güvenliğin çok önem arz etmediği durumlarda sadece simetrik şifreleme tekniği kullanılabilir. Bu durumda sadece simetrik şifreleme yapıldığı için performansın çok daha hızlı olduğu görülecektir.

Bu tez çalışmasında performans sonuçları, bir istemci tarafından gerçekleşen işlemlere göre test edilmiştir. Gelecekte yapılacak çalışmalarda, bir istemci tarafından gerçekleşen şifre çözme süresi ile birden fazla istemci tarafından gerçekleşen şifre çözme süreleri arasında kıyaslama yapılarak, performans sonuçları analiz edilebilir.

Gelecekteki çalışmalarda iki adımlı şifreleme modeli kullanılarak, ilk şifreleme yönteminde simetrik şifrelerden biri, ikinci şifreleme yönteminde yine simetrik

şifrelerden biri tercih edilebilir. Bu durumda her iki şifreleme işlemi de simetrik yöntem olmasına rağmen farklı katmanlarda farklı anahtarlar kullanılarak şifreleme gerçekleşeceği için güvenlik artacak, üstelik performans RSA'ya göre daha iyi bir durumda olacaktır. Ayrıca SQL Server, simetrik anahtar kullanılarak gerçekleşen şifrelemelerde asimetrik anahtarın aksine birçok farklı algoritmayı desteklemektedir.

Gelecekte yapılacak çalışmalarda tek adımda gerçekleşen şifreleme algoritmaları ayrı ayrı arkayüzde (backend) ve veritabanı katmanında çalıştırılarak, hangi seviyede şifreleme yapıldığında performansın daha iyi durumda olduğu gözlemlenebilir.

Ayrıca gelecekte önyüz(frontend)-arkayüz (backend)-veritabanı katmanlarında ya da arkayüz-veritabanı-sunucu katmanlarında şifreleme işlemi gerçekleştirilebilir. Bunun sonucunda art arda üç kez çalışan şifreleme sonucunun güvenliğe ve performansa olan etkisi analiz edilebilir.

KAYNAKLAR

[1] Avaroğlu, E. Bilgi Güvenliğinin Temel Yapı Taşı: Kriptoloji. Düşünce Dünyasında Türkiz, 53-59. https://www.tasav.org/media/k2/attachments/43_BASKI.pdf

[2] Cisco CCNA Cybersecurity Operations v1.1. Chapter 9: Cryptography and the Public Key Infrastructure. Erişim adresi: https://netacad.fit.vutbr.cz/wp-content/uploads/ccnp/cops/CyOps1.1_Chp09.pdf

[3] Çelik, M. (2021). Kurumlar Arasında Web Servis Aracılığı ile Sunulan Kişisel Verilerin Yazılım Geliştiricilere Karşı Korunması [Yüksek Lisans Tezi]. Gazi Üniversitesi

[4] Hakan, K. (2019). İstemci Taraflı Şifreleme ve Dağıtık Depolama ile Kişisel Veri Güvenliğinin Arttırılmasına Yönelik Bulut Tabanlı Uygulamanın Geliştirilmesi [Yüksek Lisans Tezi]. Trakya Üniversitesi

[5] Gül, Ç. (2022). Spn Yapıdaki Simetrik Şifreleme Algoritmalarının Tasarımı, Güvenlik ve Performans Analizleri [Yüksek Lisan Tezi]. Marmara Üniversitesi

[6] Hameed, M. E., Ibrahim, M. M., Abd Manap, N., & Mohammed, A. A. (2020). A lossless compression and encryption mechanism for remote monitoring of ECG data using Huffman coding and CBC-AES. Future generation computer systems, 111, 829-840. https://doi.org/10.1016/j.future.2019.10.010

[7] Ors, S. B., Gurkaynak, F., Oswald, E., & Preneel, B. (2004, April). Power-analysis attack on an ASIC AES implementation. In International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. (Vol. 2, pp. 546-552). IEEE. https://doi.org/10.1109/ITCC.2004.1286711

[8] Alwan, A. H., & Kashmar, A. H. (2023). Block Ciphers Analysis Based on a Fully Connected Neural Network. Ibn AL-Haitham Journal For Pure and Applied Sciences, 36(1), 415-427. https://doi.org/10.30526/36.1.3058

[9] Chowdhary, C. L., Patel, P. V., Kathrotia, K. J., Attique, M., Perumal, K., & Ijaz, M. F. (2020). Analytical study of hybrid techniques for image encryption and decryption. Sensors, 20(18), 5162. https://doi.org/10.3390/s20185162

[10] Wadehra, S., Goel, S., & Sengar, N. (2018). AES Algorithm: Encryption and Decryption. International Journal of Trend in Scientific Research and Development, 2(3), 1075-1077. https://doi.org/10.31142/ijtsrd11221

[11] Huang, X., Saniie, J., Arnold, D., Fang, T., Heifetz, A., & Bakhtiari, S. (2022). Software-Defined Ultrasonic Communication System With OFDM for Secure Video Monitoring. IEEE Access, 10, 47309-47321. https://doi.org/10.1109/ACCESS.2022.3168706

[12] Amudha, K., Nelson Kennedy Babu, C., & Balu, S. (2016). Hybrid baker map with AES in cipher block chaining mode in medical images. Int. Res. J. Pharm., 8(4), 67-73. https://doi.org/10.7897/2230-8407.080451

[13] Fazackerley, S., McAvoy, S. M., & Lawrence, R. (2012, March). GPU accelerated AES-CBC for database applications. In Proceedings of the 27th Annual ACM Symposium on Applied Computing (pp. 873-878). https://doi.org/10.1145/2245276.2245446

[14] Bilgisayar Kavramları (2023, 3 Ocak). Elektonik Kod Defteri Şekli (Electronic Code Book Mode). https://bilgisayarkavramlari.com/2009/03/06/elektronik-kod-defteri-sekli-electronic-code-book-mode/

[15] Yerlikaya, T., Buluş, E., & Buluş, N. (2006). Asimetrik Şifreleme Algoritmalarında Anahtar Değişim Sistemleri. Akademik Bilişim, 9-11. https://ab.org.tr/ab06/bildiri/102.pdf

[16] Kise, A. (2017). Veri Şifreleme Microsoft SQL Server. https://1drv.ms/b/s!Ap5-WHzOJRrTi-5xYhoiCWwIRlyucQ?e=Usiqeh

[17] El Bouchti, K., Ziti, S., Omary, F., & Kharmoum, N. (2019). A new database encryption model based on encryption classes. Journal of Computer Science, 15(6), 844-854. https://doi.org/10.3844/jcssp.2019.844.854

[18] Küçükbacak, R. (2022). Securıty Analysıs of Advanced Encryptıon Standard (AES) for Image Encryptıon [Yüksek Lisans Tezi]. Yeditepe University

[19] Aslanyürek, C. (2018). Şifreleme Algoritmalarının Hızını Etkileyen Faktörler [Yüksek Lisans Tezi]. Trakya Üniversitesi

[20] Gençoğlu, H. (2017). Hibrit Şifreleme Algoritması [Doktora Tezi]. Trakya Üniversitesi

[21] Beşkirli, A., Özdemir, D., & Beşkirli, M. (2019). Şifreleme Yöntemleri ve RSA Algoritması Üzerine Bir İnceleme. Avrupa Bilim ve Teknoloji Dergisi, 284-291. https://doi.org/10.31590/ejosat.638090

[22] Kaldirim, Ü. (2023, 18 Mart). Simetrik – Blok Şifreleme Algoritmaları ve Yapısı. https://medium.com/ltunes/si%CC%87metri%CC%87k-blok-%C5%9Fi%CC%87freleme-algori%CC%87tmalari-ve-yapisi-c76fbe80ed4

[23] Sofuoğlu, B. (2023, 3 Mart). Simetrik ve Asimetrik Şifreleme Algoritmaları. https://www.gokyuzuavcisi.com/simetrik-ve-asimetrik-sifreleme-algoritmalari/

[24] Ciğer, İ. (2012). Data Şifreleme Algoritmaları ve Performans Analizi [Yüksek Lisans Tezi]. İstanbul Üniversitesi

[25] Cyberparts. (2023, 2 Nisan). Blok Şifreleme ve Akış Şifreleme Arasındaki Fark. https://tr.cyberparts.pl/difference-between-block-cipher-and-stream-cipher-312#menu-3

[26] Berry, P. (2023, 30 Mart). Blok Şifreleme ve Akış Şifreleme Arasındaki Fark. https://tr1.surveillancepackages.com/difference-between-block-cipher-and-stream-cipher-4b2a2

[27] Yerlikaya, T., & Aslanyürek, C. (2019). RSA Algoritmasının Şifreleme Hızını Arttıran Algoritmalar ve Performansları. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(3), 853-862. https://doi.org/10.24012/dumf.559789

[28] Yüksel, T., & Özgün, B. (2021). RSA ve RC4 Algoritmalarının Performans Karşılaştırması. AURUM Journal of Engineering Systems and Architecture, 5(1), 29-40. https://dergipark.org.tr/en/download/article-file/1520045

[29] Gökhan Aryız (2023, 15 Nisan). Simetrik ve Asimetrik Şifreleme Algoritmaları Arasındaki Farklar. https://www.gokhan4.dev/simetrik-ve-asimetrik-sifreleme-algoritmalari-arasindaki-farklar/

[30] Erdağı, E. (2017). Veritabanı Güvenlik Riskleri, Şifreleme Algoritmaları ve Enjeksiyon Modelleri [Yüksek Lisans Tezi]. İstanbul Teknik Üniversitesi

[31] Vikipedi (2023, 20 Şubat). Blok Şifreleme Çalışma Kipleri. https://tr.wikipedia.org/wiki/Blok_şifre_çalışma_kipleri

[32] Alslman, Y. S., Ahmad, A., & AbuHour, Y. (2023). Enhanced and authenticated cipher block chaining mode. Bulletin of Electrical Engineering and Informatics, 12(4), 2357-2362. https://doi.org/10.11591/eei.v12i4.5113

[33] Saidi, R., Bentahar, T., Cherrid, N., Bentahar, A., & Mayache, H. (2020). Evaluation and Analysis of Interferograms from an InSAR Radar Encrypted by an AES-Based Cryptosystem with The Five Encryption Modes. International Journal on Electrical Engineering and Informatics, 12(4), 912-932. https://doi.org/10.15676/ijeei.2020.12.4.13

[34] Security Siber Güvenlik & Linux Blog (2023, 2 Ocak). Crypto 101 – Şifreleme Operasyonu Modları. https://www.mehmetince.net/crypto-101-5-sifreleme-operasyonu-modlari-ecb-cbc-ofb/

[35] Bilgisayar Kavramları (2023, 5 Ocak). Yayılımlı Şifre Blok Zincirlemesi (Propagating Cipher Block Chaining, PCBC). https://bilgisayarkavramlari.com/2009/03/07/yayilimli-sifre-blok-zincirlemesi-propagating-cipher-block-chaining-pcbc/

[36] Kane, L. E., Chen, J. J., Thomas, R., Liu, V., & Mckague, M. (2020). Security and performance in IoT: A balancing act. IEEE access, 8, 121969-121986. https://doi.org/10.1109/ACCESS.2020.3007536

[37] Bilgisayar Kavramları (2023, 5 Ocak). Şifre Geri Beslemeli (Cipher Feedback, CFB). https://bilgisayarkavramlari.com/2009/03/08/sifre-geri-beslemeli-cipher-feedback-cfb/

[38] Altan, K., Kaşkaoğlu, K., Kındap, N., Özakın, Ç.,Saygı, Z., Yıldırım, E., & Yıldırım, M., Yıldız, S. (2004). Kriptografi Bölümü, Kriptlojiye Giriş Ders Notları. Erişim adresi https://iam.metu.edu.tr/system/files/iamData/LectureNotes

[39] Yılmaz, M. (2017). Simetrik Şifreleme Algoritmalarının Kullanımı için Akıllı Bir Seçim Sistemi Geliştirilmesi [Yüksek Lisans Tezi]. Muğla Sıtkı Koçman Üniversitesi

[40] Şahin, F. (2015). Modern blok şifreleme algoritmaları. İstanbul Aydın Üniversitesi Dergisi, 7(26), 23-40. https://dergipark.org.tr/tr/download/article-file/319383

[41] Çavuşoğlu, Ü. (2016). Kaos Tabanlı Hibrit Simetrik ve Asimetrik Şifreleme Algoritmaları Tasarımı ve Uygulaması [Doktora Tezi]. Sakarya Üniversitesi

[42] Filizbay, Y. (2023, 4 Nisan). SQL Server Güvenlik Kavramı – Bölüm 3. https://yavuzfilizlibay.com/sql-server-guvenlik-kavrami-bolum-3/

ÖZGEÇMİŞ

Ad-Soyad : Zehra Nur KÖSE

ÖĞRENİM DURUMU:

- **Lisans :** 2020, Düzce Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yüksek lisans :** Devam ediyor, Sakarya Üniversitesi, Bilgisayar ve Bilişim Mühendisliği Anabilim Dalı, Bilgisayar ve Bilişim Mühendisliği Programı

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2020 yılında Akademedya Yazılım'da yazılım geliştirici olarak çalışmaya başladı, halen görevine devam etmektedir.