



**T.C.  
DÜZCE ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**DNA TABANLI KRİPTOLOJİ UYGULAMASI**

**FURKAN TALO**

**YÜKSEK LİSANS TEZİ  
ELEKTRİK ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ  
ANABİLİM DALI**

**DANIŞMAN  
DR. ÖĞR. ÜYESİ ESRA ŞATIR**

**DÜZCE, 2021**

**T.C.**  
**DÜZCE ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**DNA TABANLI KRİPTOLOJİ UYGULAMASI**

Furkan TALO tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Dr. Öğr. Üyesi Esra ŞATIR

Düzce Üniversitesi

**Jüri Üyeleri**

Dr. Öğr. Üyesi Esra ŞATIR

Düzce Üniversitesi

Prof. Dr. Ahmet Bedri ÖZER

Fırat Üniversitesi

Dr. Öğr. Üyesi İrem DÜZDAR ARGUN

Düzce Üniversitesi

Tez Savunma Tarihi: 06/07/2021

## BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

06 Temmuz 2021

Furkan TALO



## TEŞEKKÜR

Yüksek lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Dr. Öğr. Üyesi Esra ŞATIR'a en içten dileklerimle teşekkür ederim. Tez çalışmam boyunca değerli katkılarını esirgemeyen ve desteklerini sunan Ercan ATAGÜN'e teşekkür ederim.

Çalışmamda manevi desteklerini hep yanımda hissettiğim sevgili Eda ÇETİN'e en derin duygularıyla teşekkür ediyorum.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

**06 Temmuz 2021**

**Furkan TALO**

# İÇİNDEKİLER

## Sayfa No

|   |     |
|---|-----|
| ŞEKİL LİSTESİ.....  | i   |
| ÇİZELGE LİSTESİ.....  | iii |
| KISALTMALAR.....  | iv  |
| ÖZET .....  | v   |
| ABSTRACT .....  | vi  |
| 1. GİRİŞ.....   | 1   |
| 1.1. ŞİFRELEME.....   | 2   |
| 1.2. ALGORİTMALAR VE ANAHTARLAR.....  | 6   |
| 1.2.1. Simetrik Algoritmalar .....  | 8   |
| 1.2.2. Asimetrik Algoritmalar .....   | 9   |
| 1.2.3. Asimetrik ve Simetrik Algoritmaların Avantajları ve Dezavantajları ... | 10  |
| 2. DNA YAPISI.....  | 12  |
| 2.1. DNA STEGANOĞRAFİSİ.....  | 14  |
| 2.2. DNA KRİPTOĞRAFİSİ .....  | 15  |
| 2.2.1. LİTERATÜR ÖZETİ .....  | 16  |
| 2.3. DNA HESAPLAMASI.....   | 22  |
| 3. BÖLÜM ASİMETRİK ALGORİTMALAR .....   | 25  |
| 3.1. RSA ALGORİTMASI .....  | 25  |
| 3.2. DSA ALGORİTMASI .....  | 27  |
| 4. BÖLÜM SİMETRİK ALGORİTMALAR .....  | 28  |
| 4.1. DES ALGORİTMASI .....  | 28  |
| 4.1.1. DES Algoritmasının Yapısı .....  | 28  |
| 4.2. AES ALGORİTMASI .....  | 33  |
| 4.3. BLOWFISH ALGORİTMASI .....   | 36  |
| 4.4. TWOFISH ALGORİTMASI .....  | 39  |
| 4.5. IDEA ALGORİTMASI .....   | 41  |
| 4.6. TEA ALGORİTMASI .....  | 43  |
| 4.7. HASH ALGORİTMALARI.....  | 44  |
| 5. DNA TABANLI KRİPTOLOJİ UYGULAMASI .....                                    | 46  |
| 5.1. ÖNERİLEN DNA TABANLI MODEL.....  | 46  |
| 5.2. DES ALGORİTMASI İÇİN ÖNERİLEN DNA MODELİ.....                            | 47  |
| 5.3. BLOWFISH ALGORİTMASI İÇİN ÖNERİLEN DNA MODELİ.....                       | 51  |
| 5.4. DES ALGORİTMASININ DNA İLE PERFORMANS ANALİZİ .....                      | 55  |
| 5.4.1. Performans Test Sonuçları.....   | 55  |
| 5.4.1.1. DES Algoritmasının Şifreleme ve Şifre Çözme Sonuçları .....          | 56  |
| 5.5. BLOWFISH ALGORİTMASININ DNA İLE PERFORMANS ANALİZİ... ..                 | 66  |
| 5.5.1. Performans Test Sonuçları.....   | 66  |
| 5.5.1.1. Blowfish Algoritmasının Şifreleme ve Şifre Çözme Sonuçları .....     | 67  |

|   |           |
|---|-----------|
| <b>5.6. BLOWFISH VE DES ALGORİTMALARININ PERFORMANS ANALİZİ</b> | <b>76</b> |
| 5.6.1. Bilgisayar-1 Performans Analizi.....                     | 77        |
| 5.6.2. Bilgisayar-2 Performans Analizi.....                     | 82        |
| <b>6. SONUÇLAR VE ÖNERİLER.....</b>                             | <b>87</b> |
| <b>7. KAYNAKLAR .....</b>                                       | <b>89</b> |
| <b>ÖZGEÇMİŞ.....</b>  | <b>93</b> |



## ŞEKİL LİSTESİ

|   | <u>Sayfa No</u> |
|---|-----------------|
| Şekil 1.1. Düz metnin şifrlenmesi ve çözülmesi.....             | 2               |
| Şekil 1.2. Bilginin iletilmesi. ....                            | 3               |
| Şekil 1.3. Farklı anahtarlar ile şifreleme ve şifre çözme. .... | 7               |
| Şekil 1.4. Simetrik anahtar ile şifreleme.....                  | 9               |
| Şekil 1.5. Asimetrik anahtar ile şifreleme.....                 | 10              |
| Şekil 2.1. DNA baz çiftlerinin kimyasal yapısı. ....            | 13              |
| Şekil 2.2. DNA dijital kodlaması.....                           | 14              |
| Şekil 2.3. DNA dijital anahtar kodlaması. ....                  | 15              |
| Şekil 2.4. DNA merkezi oluşumu.....                             | 17              |
| Şekil 2.5. DNA indeksleme. ....                                 | 19              |
| Şekil 3.1. RSA algoritması. ....                                | 26              |
| Şekil 4.1. IP (Giriş permütasyonu). ....                        | 29              |
| Şekil 4.2. DES algoritması.....                                 | 30              |
| Şekil 4.3. DES genişletme permütasyonu (EP). ....               | 30              |
| Şekil 4.4. S-Box örneği.....                                    | 31              |
| Şekil 4.5. S permütasyonu. ....                                 | 31              |
| Şekil 4.6. PC-1.....  | 32              |
| Şekil 4.7. Kaydırma permütasyonu.....                           | 32              |
| Şekil 4.8. PC-2.....  | 32              |
| Şekil 4.9. F fonksiyonu. ....                                   | 33              |
| Şekil 4.10. AES tur sayısına göre anahtar sistemi. ....         | 34              |
| Şekil 4.11. AES durum dizisi.....                               | 35              |
| Şekil 4.12. AES akış diyagramı. ....                            | 35              |
| Şekil 4.13. Blowfish akış diyagramı. ....                       | 37              |
| Şekil 4.14. Blowfish S-Kutuları.....                            | 38              |
| Şekil 4.15. Blowfish F fonksiyonu. ....                         | 38              |
| Şekil 4.16. Twofish yapısı. ....                                | 40              |
| Şekil 4.17. IDEA yapısı. ....                                   | 42              |
| Şekil 4.18. TEA yapısı.....                                     | 44              |
| Şekil 4.19. Hash fonksiyonu. ....                               | 45              |
| Şekil 4.20. Hash fonksiyonu çıktıları.....                      | 45              |
| Şekil 5.1 DES algoritması için tasarlanan akış şeması. ....     | 47              |
| Şekil 5.2 DNA bağları. ....                                     | 48              |
| Şekil 5.3. Watson Crick's DNA bağları. ....                     | 48              |
| Şekil 5.4. Şifrelenecek veri. ....                              | 49              |
| Şekil 5.5. Şifreleme işlemi. ....                               | 50              |
| Şekil 5.6. Şifre çözme işlemi. ....                             | 50              |
| Şekil 5.7 Blowfish için önerilen akış şeması. ....              | 52              |
| Şekil 5.8. Şifrelenecek veri. ....                              | 53              |
| Şekil 5.9. Düz metnin şifrlenmesi.....                          | 53              |
| Şekil 5.10. Şifre çözümü.....                                   | 54              |
| Şekil 5.11. 128 Bit işlem zaman grafiği.....                    | 57              |
| Şekil 5.12. 256 Bit işlem zaman grafiği.....                    | 58              |
| Şekil 5.13. 512 Bit işlem zaman grafiği.....                    | 59              |
| Şekil 5.14. 1024 Bit işlem zaman grafiği.....                   | 61              |
| Şekil 5.15. 2048 Bit işlem zaman grafiği.....                   | 62              |

|  |    |
|--|----|
| Şekil 5.16. 4096 Bit işlem zaman grafiği.....                    | 63 |
| Şekil 5.17. 8192 Bit işlem zaman grafiği.....                    | 65 |
| Şekil 5.18. 128 Bit işlem zaman grafiği.....                     | 67 |
| Şekil 5.19. 256 Bit işlem zaman grafiği.....                     | 69 |
| Şekil 5.20. 512 Bit işlem zaman grafiği.....                     | 70 |
| Şekil 5.21. 1024 Bit işlem zaman grafiği.....                    | 71 |
| Şekil 5.22. 2048 Bit işlem zaman grafiği.....                    | 73 |
| Şekil 5.23. 4096 Bit işlem zaman grafiği.....                    | 74 |
| Şekil 5.24. 8192 Bit işlem zaman grafiği.....                    | 75 |
| Şekil 5.25. DES ve Blowfish işlem zaman grafiği. ....            | 77 |
| Şekil 5.26. DES ve Blowfish tabanlı DNA işlem zaman grafiği..... | 79 |
| Şekil 5.27. DES ve Blowfish işlem zaman grafiği. ....            | 82 |
| Şekil 5.28. DES ve Blowfish DNA işlem zaman grafiği.....         | 84 |



## ÇİZELGE LİSTESİ

|   | <b><u>Sayfa No</u></b> |
|---|------------------------|
| Çizelge 2.1. DNA kriptolojisi (Literatür çalışmaları).....                                      | 20                     |
| Çizelge 2.4. DNA hesaplaması (Literatür çalışmaları).....                                       | 23                     |
| Çizelge 5.1. Bilgisayar donanımları. ....   | 55                     |
| Çizelge 5.2. 128 Bit performans değerleri. ....   | 56                     |
| Çizelge 5.3. 256 Bit performans değerleri. ....   | 58                     |
| Çizelge 5.4. 512 Bit performans değerleri. ....   | 59                     |
| Çizelge 5.5. 1024 Bit performans değerleri. ....  | 60                     |
| Çizelge 5.6. 2048 Bit performans değerleri. ....  | 62                     |
| Çizelge 5.7. 4096 Bit performans değerleri. ....  | 63                     |
| Çizelge 5.8. 8192 Bit performans değerleri. ....  | 64                     |
| Çizelge 5.9. Bilgisayar donanımları. ....   | 66                     |
| Çizelge 5.10. 128 Bit performans değerleri. ....  | 67                     |
| Çizelge 5.11. 256 Bit performans değerleri. ....  | 68                     |
| Çizelge 5.12. 512 Bit performans değerleri. ....  | 70                     |
| Çizelge 5.13. 1024 Bit performans değerleri. ....   | 71                     |
| Çizelge 5.14. 2048 Bit performans değerleri. ....   | 72                     |
| Çizelge 5.15. 4096 Bit performans değerleri. ....   | 74                     |
| Çizelge 5.16. 8192 Bit performans değerleri. ....   | 75                     |
| Çizelge 5.17. DES ve Blowfish şifreleme algoritmalarının karşılaştırılması. ....                | 77                     |
| Çizelge 5.18. DES ve Blowfish tabanlı DNA şifreleme algoritmalarının<br>karşılaştırılması. .... | 78                     |
| Çizelge 5.19. DNA tabanlı şifreleme sonuçları. ....   | 80                     |
| Çizelge 5.20. DNA tabanlı şifreleme sonuçları. ....   | 81                     |
| Çizelge 5.21. DES ve Blowfish şifreleme algoritmalarının karşılaştırılması. ....                | 82                     |
| Çizelge 5.22. DES ve Blowfish tabanlı DNA şifreleme algoritmalarının<br>karşılaştırılması. .... | 83                     |
| Çizelge 5.23. DNA tabanlı şifreleme sonuçları. ....   | 85                     |
| Çizelge 5.24. DNA tabanlı şifreleme sonuçları. ....   | 86                     |

## KISALTMALAR

|         |  |
|---------|--|
| AES     | Gelişmiş Şifreleme Standartı             |
| ASCII   | American Standard Code for Information   |
| CBC     | Cipher Block Chaining                    |
| CFB     | Cipher FeedBack Mode                     |
| CPU     | Merkezi İşlem Birimi                     |
| DES     | Veri Şifreleme Standartı                 |
| DNA     | Deoksiribo Nükleik Asit                  |
| DSA     | Dijital İşaret Algoritması               |
| EFT     | Elektronik Fon Transferi                 |
| EP      | Genişletme Permutasyonu                  |
| EFTPOSS | Elektronik Fon Transfer Satış Cihazı     |
| FIPS    | Federal Bilgi İşleme Standartları        |
| FP      | Final Permutation                        |
| HMAC    | Keyed-Hashing for Message Authentication |
| IBM     | Uluslararası İş Makinleri                |
| IDEA    | Uluslararası Veri Şifreleme Algoritması  |
| IP      | Initial Permutation                      |
| ISO     | Uluslararası Standartlık Örgütü          |
| MAC     | Mesaj Doğrulama Kodu                     |
| MB      | Mega Byte                                |
| MD5     | Mesaj Özet 5                             |
| MDC     | Modification Detection Code              |
| MOSS    | Object Security Services                 |
| mRNA    | Mesajcı Ribonükleik Asit                 |
| NP      | Deterministik Olmayan Polinom            |
| NIST    | Ulusal Standart ve Teknoloji Enstitüsü   |
| NSA     | Ulusal Güvenlik Teşkilatı                |
| NW      | Needleman-Wunsch                         |
| OBEB    | Ortak Bölenlerin En Büyüğü               |
| ODN     | Optik Dağıtım Ağı                        |
| OTP     | Tek Kullanımlık Ped                      |
| PC-1    | Birimci Sıkıştırma Permutasyonu          |
| PC-2    | İkinci Sıkıştırma Permutasyonu           |
| PCR     | Polimeraz Zincir Reaksiyonu              |
| PEM     | Privacy Enhanced Mail                    |
| PGP     | Pretty Good Privacy                      |
| PKA     | Negatif Baz Logaritmasıdır               |
| PKB     | Baz ayrışma Sabitinin Logaritması        |
| RAM     | Rastgele Erişimli Hafıza                 |
| RC2     | Ron's Code 2                             |
| RSA     | Rivest-Shamir-Adleman                    |
| SHA     | Güvenli Hash Algoritması                 |
| ssDNA   | Tek İplikli Deoksiribonükleik Asit       |
| SSL     | Güvenli Yuva Katmanı                     |
| TEA     | Ufak Şifreleme Algoritması               |

# ÖZET

## DNA TABANLI KRİPTOLOJİ UYGULAMASI

Furkan TALO

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Elektrik Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Esra ŞATIR

Temmuz 2021, 92 sayfa

Gelişen teknoloji ile birlikte bilgi güvenliği önemli bir alan olmuştur. Bilgi, saklama ya da bir başkasının anlamayacağı şekilde içeriğinin değiştirilmesi ile korunmaya çalışılmaktadır. Bütün bu şifreleme işlemlerini yapmak için günümüzde çeşitli şifreleme teknikleri kullanılmaktadır, bu şifreleme teknikleri simetrik ve asimetrik olarak sınıflandırılabilir. Bu tez çalışmasında asimetrik ve simetrik algoritmalar hakkında genel bilgiler verildi, hedef çözüm olarak yeni bir teknik olan DNA tabanlı şifreleme kullanıldı. Bu teknik insan biyolojik DNA yapısını örnek alarak DNA bazlarını bilgisayar diline benzetme temeline dayanan bir modellemedir. Bu çalışmada incelenmesi yapılan simetrik ve asimetrik algoritmalarından simetrik algoritma sınıfına giren Blowfish ve DES algoritmaları, DNA'nın yapısına göre yeniden modellenerek güncellendi. Bu hedef doğrultusunda elde edilen bulgular, süre karmaşıklığı, bellek karmaşıklığı ve işlemci karmaşıklığı açısından incelendi. Performans analizi yapılırken bilgisayar donanımları performansla etki edebileceği için farklı donanım ve yazılım özellikleri içeren bilgisayarlar üzerinde sonuçlar alındı. Bu sonuçlar alınırken "Python" dili kullanıldı ve sonuçlar "PyCharm 2020.2.4" ortamında elde edildi.

**Anahtar sözcükler:** Kriptoloji, DNA şifreleme, Blowfish, DES.

# **ABSTRACT**

## **DNA-BASED CRYPTOLOGY APPLICATION**

Furkan TALO

Düzce University

Graduate School of Natural and Applied Sciences, Department of Electrical Electronics  
and Computers

Master's Thesis

Supervisor: Assist. Prof. Dr. Esra ŞATIR

July 2021, 92 pages

Information security has become an important area with the developing technology. It is tried to be protected by storing the data or changing its content in a way that someone else cannot understand. Various encryption techniques are used today to perform all these encryption processes. We can classify these encryption techniques as symmetric and asymmetric. In this study, asymmetric and symmetric algorithms are generally examined. DNA modeling is a new technique in the field of coding, and this technique is based on simulating the human biological DNA structure and simulating DNA bases into computer language. Among the symmetric and asymmetric algorithms examined in this study, Blowfish and DES algorithms have been updated by remodeling with DNA, also time, memory and processor complexity were evaluated after the update. While performing the performance analysis, results were obtained on computers with different hardware and software features, as computer hardware may affect performance. While getting these results, Python language was used. The results were obtained in “PyCharm 2020.2.4” platform.

**Keywords:** Cryptology, DNA encryption, Blowfish, DES.

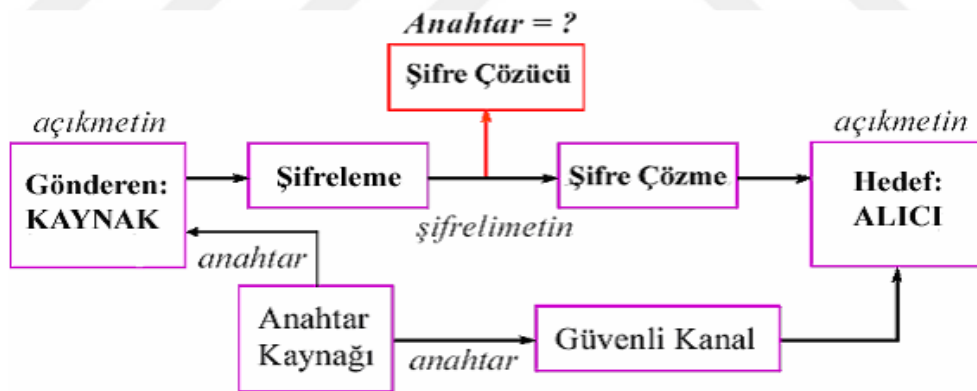
# 1. GİRİŞ

Son yıllarda teknoloji ve bilişim dünyasında devrim niteliğinde gelişmeler ile birlikte güvenlik açığının taşıdığı önem artmaktadır. Gün geçtikçe artan internet ve ağ teknolojisi ile ağdaki bilgi akışı nedeniyle kullanıcılar için güvenlik tehditleri de artmaktadır. Dolayısıyla modern bilgi işlem sistemleri için bilgi güvenliği gerekli hale gelmektedir. Modern sistemler güvenlik tehditlerine karşı verilerini kriptografi uygulamalarıyla korumaktadır. Kriptografide veri ya da düz metin şifrelenmesi/çözülmesi anahtar yardımıyla yapılır [1]. Şuan kullanılan bu kriptografi yöntemleri fazla matematiksel işlemler ve iki tip özel anahtar sistemi ile çalışır. Her ne kadar güvenli olarak görünse bile günümüzde her geçen gün algoritmaların güvenlik açıkları ortaya çıkmaktadır. Bu açıkları kapatmak için çalışmalar yapılmakta ve yeni alanlar araştırılmaktadır. İlerleyen teknoloji ile kriptoloji alanında DNA kriptografisi denilen yeni bir şifreleme tekniği daha ortaya çıkmıştır. Bu tez çalışmasının ana amacı düz metni şifrelemek ve DNA dijital formunda saklamaktır. DNA tabanlı hesaplamalar diğer algoritmalara göre çok daha fazla zaman alır. Herhangi bir şifreleme algoritmasının görevi, verileri çok uzun bir süre boyunca güvence altına almaktır. Bu teknikte, DNA bazları rasgele sırada düzenlenir ve düz metin bitleri bu bazlar kullanılarak başarılı bir şekilde saklanabilir ve böylece veriler çok uzun süreler için korunabilir. DNA tabanlı tekniklerin bu olumlu yönleri göz önüne alınarak günümüzdeki şifreleme algoritmalarına uygulanabilir olması ve DNA tabanlı şifreleme sonucunda daha performanslı ve güvenilir hale getirebileceği önceki çalışmalardaki sonuçlar ışığında görülmüş ve bu çalışmada da DNA tabanlı bir şifreleme metodu gerçekleştirilmiştir.

Bu çalışmada, şifrelemede kullanılan algoritmalar ve genel yapıları hakkında özet bilgiler verildi. Bu bilgiler eşliğinde şifreleme için yeni bir teknik olan DNA tabanlı şifreleme hakkında bilgiler verilip ve uygulaması yapıldı. Bunun yanı sıra literatürdeki DNA ile ilgili çalışmalardan bahsedilerek ve simetrik şifreleme algoritmalarından olan DES ve Blowfish algoritması üstünde DNA tabanlı şifreleme yapılarak avantaj ve dezavantajları incelendi. Bu tez çalışmasında farklı boyutlardaki düz metinler şifrelendi. Bu inceleme yapılırken donanım ve işletim sistemi farklı olan iki ayrı bilgisayarda

Bu tez çalışmasının birinci bölümünde, genel olarak kriptoloji kavramı ve kriptoloji içinde kullanılan algoritmaların yapılarından bahsedildi. İkinci bölümde, biyolojik DNA yapısı ve bilgisayarlara bu yapının uyarlanmasından bahsedildi. Üçüncü bölümde, asimetrik algoritmalarından iki tanesi ön bilgi olması amacıyla açıklandı. Dördüncü bölümde simetrik algoritmalar hakkında bilgiler verildi. DNA tabanlı şifreleme yaptığımız DES ve Blowfish algoritmaları detaylı olarak anlatıldı. Beşinci bölümde tasarımı yapılan DNA tabanlı şifreleme modeli anlatılarak sonuçları çizelge ve grafiklerle açıklandı. Altıncı bölümde şifreleme sonucu çıkan sonuçlar yorumlanarak gerçekleştirilen çalışmaya dair iyileştirici öneriler yapıldı.

Bir mesaj düz metinden oluşmakta ve bazen açık metin olarak da bilinmektedir. Bir mesajın ya da dokümanın içeriğini gizlemek üzere yapılan gizleme süreci şifreleme olarak açıklanmaktadır. Şifrelenmiş bir ileti şifreli mesajı oluşturur. Şifreli metni düz metine yani orijinal haline dönüştürme işlemi şifre çözmedir [1].



Şifrelemenin amacı, iletinin istenmeyen kişiler ya da üçüncü şahıslar tarafından okunmasını önlemektir. Şifre çözme prosesi şifrelemenin aksi şeklinde yapılır, buna paralel şifreli metnin düz metine dönüşüm süreci denilebilir. Çok sayıda kullanıcı barındıran ortamlarda ya da güvenlik endişesi bulunan ortamlarda, kişiler arasında güvenli bir yol olmasa bile şifreleme kullanılarak bilgi akışı sağlanabilir. Günümüzde internet ortamı mükemmel bir örnek teşkil eder. Örneğin, Furkan, Hamza'ya sadece onun okuması için özel bir mesaj göndermek istemektedir. Furkan, okunabilir haldeki

mesajı bir şifreleme anahtarı ile şifreler. Gizlenmiş olan mesaj Hamza'ya iletilir. Hamza da bu anahtarla gönderici tarafından gönderilen mesajı elde edip mesajı okur. Üçüncü şahıs olan Mirza ise gizli anahtarla ya da şifreli veriyi anahtarsız açarak mesajın içeriğine erişmek ister. Güvenli bir şifreleme ortamında anahtar olmadan açık mesajı okumak imkansızdır [1].

Standart şifreleme işlemlerinde veriyi şifrelemek için tek bir anahtar kullanılmaktadır. Şifreleyecek ve şifreyi çözecek kişide aynı anahtarı kullanır. Gönderen şahıs bu anahtarla mesajı gizler, alıcı kişide aynı anahtarla gizlenmiş mesajı orijinal haline çevirir. Bu metot “gizli anahtar şifrelemesi veya simetrik şifreleme sistemi” olarak tanıtılmıştır. Burada en dikkat edilmesi gereken durum, gönderici ve alıcı arasında bir anahtar üzerine mutabık kalınmasıdır. Bununla beraber gönderici ve alıcı aynı ortamda değillerse birbirlerine anahtar konusunda bilgi vermek için telefon veya farklı bir iletişim yolunu seçmek zorundadırlar. Anahtarı resmi ya da gayri resmi bir şekilde elde eden bir kişi o anahtar ile şifrelenmiş tüm metinleri okuyabilmektedir. Anahtarlar üzerinde yapılan her türlü işlem, üretim, iletim ve saklanması gibi konuların tümü anahtar yönetimi olarak isimlendirilmektedir. Bu, tüm şifreleme sistemlerinin dikkate alması gereken durumların başında gelir. Şifreleme genel yapısı Şekil 1.1’de gösterilmiştir.

Gelişen teknoloji ile birlikte günümüzde artık kâğıt ve yazılı belgelerin yerini dijital ortamlardaki dokümanlar almaya başlamıştır. Kişiler ve kuruluşların, özel ve resmi haberleşmelerini dijital iletişim kanalları üzerinden yapabilmeleri, dışarıdan erişime açık olan kanallar yoluyla olur. Bu yapılan işlemlerin tümü kanalın güvenilirliği ve güvenliği ile alakalıdır. Açık kanallardan gönderilen mesajlar Şekil 1.2’de gösterilmiştir. Bu durum istenmeyen şahıslar tarafından okunma, bilgiye erişim ve değiştirilme riskini taşır.



Şekil 1.2. Bilginin iletilmesi.

Şifrelenmemiş mesaj, farklı formatlardan oluşabilir; metin dosyası, resim, ses ve görüntü dosyası olabilir yani sabit bir dosya değildir. Bilgisayar için mesaj tek düze ikili bir veri grubudur denilebilir. Açık metin birine ulaştırılmak üzere ya da daha sonra kullanmak için muhafaza edilebilir. Bahsedilen bu iki olayda da mesaj şifrelenecektir.

Şifreli metin de açık metin gibi ikili bir değere sahiptir sadece boyutsal olarak bazen açık metinden daha uzun ya da aynı olabilir. Şifreli metni üretme süreci açık metin üzerinden yapılır. Aynı şekilde şifreyi çözmek için de şifreli metin üzerinde yapılan birtakım işlemler sonucu özgün metin, bir başka ifade ile orijinal metin ortaya çıkar.

Şifreleme, açık bir metni ya da herkes tarafından okunabilir bilgiyi başkalarının anlayamayacağı bir karmaşıklığa kavuşturan bir bilim dalıdır denilebilir, bir başka ifadeyle bilginin güvenliğini sağlar. Güvenilirlik, veri bütünlüğü, kimlik doğrulama gibi bilgi güvenliği başlıklarını içinde barındıran ve matematiksel teoremler kullanılarak yapılan incelemeler olarak da nitelendirilebilir. Şifreleme kelimesi, yunanca gizli anlamına gelen “kryptos” kelimesinden esinlenerek ortaya çıkarılmıştır. Bu süreçte, bilgi hedeflenen alıcı dışında istenmeyen şahıs ya da şahısların erişemeyeceği veya değiştiremeyeceği bir şekilde kodlanır. Şifreleme ve şifre çözme işlemi, verinin okunabilir ve kodlanmış formatlara dönüştürülmesini sağlayan matematiksel işlemler dizisi veya mantıksal bir algoritma ve bir anahtar üçlüsünden oluşmaktadır. Anahtar, şifreli mesajı oluşturabilmek için kullanılan mantıksal bir dizidir denilebilir. Şifreli mesaj, ulaştırma sırasında istenmeyen birinin eline geçse bile anahtara sahip olmayan bir kişi tarafından çözülmesi ya da okunması imkansızdır. Modern şifreleme sistemlerinin üzerinde durduğu dört ana başlık aşağıda sıralanmıştır.

**Gizlilik:** Bilgi sadece istenilen kişiler tarafından çözülür diğer durumlarda çözülemez.

**Bütünlük:** Bilgi, depolanması ya da iletilme esnasında, farkına varılmadan bozulmamalıdır. Bilgi ya da veri bütünlüğü, bilginin gönderim yolu üzerinde değişikliğe uğramadığından emin olmak için gerekmektedir.

**Reddedilemezlik:** Bilgiyi oluşturan ya da gönderen kişi, daha sonra bilgiyi kendisinin oluşturduğunu veya gönderdiğini inkâr edememektedir. Reddedilemezlik işlemi, bir mesajı gönderen kişinin o mesajı size gönderdiğini inkâr edememesini sağlayan alındı onaylarını sağlamaktadır.

**Kimlik belirleme:** Gönderen ve alıcı kişiler, birbirlerinin kimliklerini doğrulayabilirler. Bir iletinin alıcısı bu iletinin kaynağını araştırmak isteyebilir. Davetsiz bir misafir



başkasının kimliğine bürünme şansına erişmemelidir. Kimlik belirleme işlemi, bilginin doğru kaynaktan alındığını doğrulamak için kullanılır.

Bilginin güvenliği, başkası tarafından izinsiz dinlenme, bilginin mutasyona uğraması, kimlik sahteciliği ve bununla ilişkili tehditlerin ortadan kaldırılması ile sağlanabilir ve bu gaye ile kullanılan temel teknik şifre analizidir.

Gelişen ve bilgiyi kullanan ülkelere bakıldığında, teknolojinin desteği ile milyonlarca insanın gözetiminin devletler tarafından yapıldığı görülebilir. Şifre analizi, sayısal dünyadaki kişilere bu özelliği sağlayan başlıca araçtır. Şifre analizi, günümüz dünyasında sadece resmî kurumlarda ya da askeriyede kullanılan bir protokol olmaktan çıkmıştır.

Şifre analizini öğrenmek ve onun modern toplumlara sağladığı pozitif yönlerden yararlanmak hem kişisel gizliliğimiz hem de dijital dünyadaki güvenliğimiz için kaçınılmazdır. Şifrelemenin bazı çok kullanılan uygulamaları donanım gerekliliklerini de istemektedir. Bununla beraber geçmiş yıllarda yazılımsal şifreleme tekniklerinde bir artış görülmüştür. Devletlerin askeriye gibi önemli kurumları donanımsal tekniklere dikkat ederler.

Donanım tekniklerinin sınıflandırılmasındaki 3 ana nokta şunlardır;

- Güvenlik
- Hız
- Kurulum kolaylığı

Özel tasarım koşullarında hazırlanmış donanımlar bütün yazılım tekniklerinden daha verimli ve hız açısından daha etkilidir.

Yazılımsal şifreleme teknikleri bilgisayar üzerinde fiziksel olarak güvenlik sağlamazlar. Bundan dolayı fark edilmeden algoritma değiştirilebilir. Bunun yanı sıra donanım teknikleri bahsedilen tarzdaki sorunlara karşı daha iyi emniyet sağlamaktadır. Erişilmesi güç kasaların, saldırganlardan uzak tutulması donanım kullanmanın farklı bir yoluna örnek olabilir. Ayrıca elektronik çipler, algoritma anahtarlarını ele geçirme çabalarına karşı emniyetli bir ortam oluşturulacak şekilde uyarlanabilir.

3 tür standart şifreleme donanımı bulunmaktadır;

- Tek başına çalışan şifreleme donanımları

- İletişim bağlantıları için kandırılan şifreleme kutuları
- Kişisel bilgisayarlara takılan çeşitli kartlar

Kendi başına çalışan şifreleme modülleri, tipik anahtar ve şifre gibi işlemleri kontrol etme mekanizmasına sahiptir. Banka ve benzeri kuruluşlar tarafından kullanılmaktadır. Özel şifreleme kutuları, iki nokta arasındaki bilgi aktarımını emniyetli ve etkin bir hale yükseltmek için işleme dahil olmaktadır.

Kişisel bilgisayarlarda genellikle kullanılan kart kodlayıcılar, standart durumlarda sabit diske ulaştırılan her şeyi kodlarlar ve disket sürücü gibi farklı depolama bölümlerine ulaştırılan bilgileri de şifrelemek için adapte olurlar. Dijital dosyalar genellikle yazılımsal tekniklerle şifrelenir. Bu teknikler anahtar ile şifreleme mantığı üzerine çalışırlar. Bu yüzden şifreleme anahtarları, erişimi kolay olan disk gibi oluşumlarda saklanmamalıdır [1].

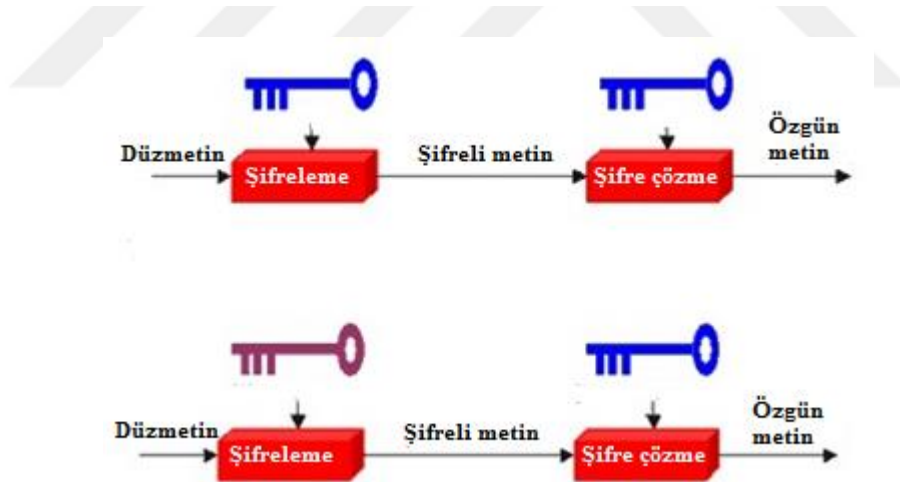
## **1.2. ALGORİTMALAR VE ANAHTARLAR**

Algoritmalar ve anahtarlar şifrelemede ve şifre çözme işlemlerinde kullanılan bir dizi sayısal ve mantıksal süreçlerdir. Eski sistem şifreleme metotları algoritmanın gizliliğine göre şekil alırlar. Ancak günümüzde geleneksel metotlar arz talep konusunda yetersiz kaldıkları için anahtar tabanlı şifreleme sistemi çok daha yaygın kullanılmaya başlanmıştır. Bunun en büyük etkenlerinden biri de çok kullanıcı bir ortamdan bir kişinin çıkması durumunda ya da her giriş çıkışta, ortamdaki diğer kullanıcıların yeni bir algoritmaya geçmelerinin bir zorunluluk olmasıdır. Ortamdan birisi gizli anahtarı istenmeyen şahıslara kaptırdığında, diğer herkesin algoritmalarını değiştirmeleri gerekmektedir. Her kullanıcı grubunun şahsi bir algoritmasının olması gerekir. Böyle bir grup, hazır şifre çözme anahtarının yazılım veya donanım mahsullerini kullanamaz çünkü istenmeyen bir şahıs aynı ürünü kullanıp algoritmayı öğrenebilmektedir. Modern algoritmalarda kullanılan şifreleme ve çözme metoduna cipher denir. Cipherlar kriptografik kodlama sistemleri olarak da adlandırılabilir.

Gelişmiş şifreleme algoritmalarında şifre çözme süreçlerinde anahtarlara ihtiyaç duyulur, şifrelenen veri ile anahtar arasında bir uyum varsa şifre çözülür. Anahtar şifreleme metoduna dayalı iki tip şifreleme algoritması bulunur. Bunlar; simetrik ve asimetrik algoritmalarıdır.

Simetrik algoritmalar anahtarın gizli tutulduğu algoritmalar olarak da bilinir. Bu tip şifreleme algoritmalarında şifre çözme ve şifreleme aynı anahtar yardımıyla yapılır veya çözücü anahtar basit bir şekilde şifreleme anahtarından ortaya çıkar. Asimetrik algoritmalarda ise iki tip anahtar kullanılır. Diğer bir tanımıyla açık anahtarlı algoritmalar, bu algoritmalarda şifreleme ve şifre çözme işlemi için iki anahtar kullanılır [1].

Bütün bu zoraki durumlar olsa bile, kısıtlı algoritmaların aşırı güvenlik gerektirmeyen uygulamalarda çok fazla kullanıldığı görülmektedir. Kullanıcılar donanımlarında bulunan güvenlik zafiyetlerinin ya farkına varmamış veya bunları önemsemiyor olabilir. Günümüz şifre analizi, bir anahtar yardımıyla bu sorunu ortadan kaldırmayı amaçlamıştır. Bu anahtar sayısız değer ve ihtimal barındırıp çok yönlü olabilir. Anahtarın alabileceği olası değerler kümesi “anahtar uzayı” olarak bilinir. Şifreleme ve şifre çözme adımlarında anahtar uzayı kullanılır. Bununla beraber bazı algoritmik sistemlerde şifreleme ve şifre çözme anahtarının farklı olabilme durumundan da bahsedilmektedir. Yani, anahtarlar arasında denklik ve eşitlik söz konusu olmayabilir. Şekil 1.3’de farklı anahtar boyutlarıyla şifreleme ve şifre çözme işlemi gösterilmiştir.



Şekil 1.3. Farklı anahtarlar ile şifreleme ve şifre çözme.

Bu algoritmalarındaki tüm gizlilik süreci, anahtar veya anahtarlar bağıdır ve hiçbirinde algoritmanın detaylarından söz edilmemiştir. Bu durumda kısacası algoritmanın açık olmasının önemi yoktur, algoritma analiz edilebilir. İstenmeyen kişilerin algoritmayı bilmesi güvenlik tehdidi değildir, gözlemci sizin özel anahtarınızı ele geçiremediği sürece verilerinizin doğru çözümünü yapamayacaktır. Şifre sistemi, algoritmalarından

olası bütün düz metinlerden, şifreli metinlerden ve anahtarlardan meydana gelmektedir. Algoritmalar anahtar çeşitleri bakımından sınıflandırdığında iki gruba ayrılır.

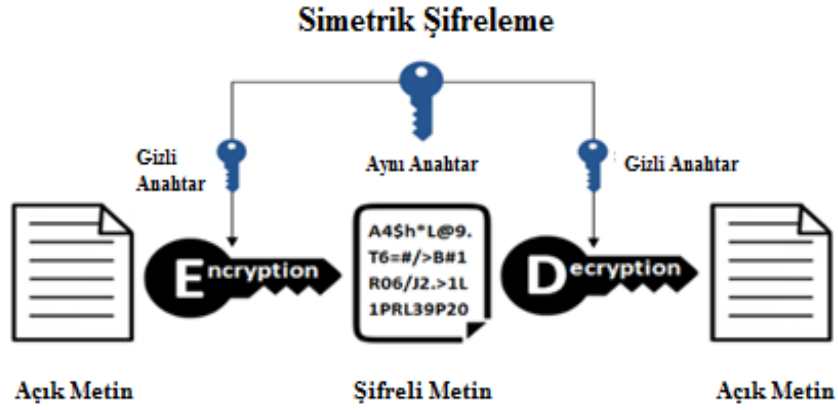
### **1.2.1. Simetrik Algoritmalar**

Gizli anahtarlı şifreleme, simetrik şifreleme veya bir anahtarlı şifreleme olarak da bilinir. Metnin şifrlenmesinde ve şifreyi çözme esnasında aynı anahtarın kullanıldığı standartlaşmış bir metottür. Bazen geleneksel olarak bilinen gizli anahtarlı algoritmalar, şifreleme anahtarının şifre çözme anahtarından hesaplanabildiği veya tam tersi olduğu algoritmalarlardır. Çoğu simetrik algoritma sisteminde aynı anahtar ile şifreleme ve şifre çözme yapılır. Gizli anahtar metodunu kullanan bu algoritmalar, mesajı gönderen ve alan kişinin ortak bir anahtar üstünde karar almasını hedefler. Simetrik bir algoritmanın erişilebilirliği anahtarla ilişkilidir. Anahtarın gizliliğini kaybetmesi, herkesin mesaj ya da mesajları şifreleyebileceği veya mesaja erişebileceği anlamına gelir. Kısacası eğer mesajların güvenliğini sağlanmak isteniyorsa, anahtar da gizli tutulmalıdır [3], [5].

Gizli anahtarlı şifrelemede temel amaç, gönderici ve alıcı kişilerin, anahtarın istenmeyen bir kişinin elde etmesine mâni olarak, ortak bir anahtar üstünde ortak bir karar almalarıdır. Bu durumda iki tarafında anahtarın analiz edilmeyeceği konusunda emin olarak, iletişim kurmasını sağlayacak bir yöntem oluşturulmalıdır. Bununla beraber açık anahtarlı sistemler, gizli anahtar kullanan sistemlere göre yavaştır. Asimetrik algoritmalarda iki farklı anahtar kullanılır, birinci anahtar şifreleme yapılırken, ikinci anahtar ise şifreyi çözmek için kullanılır. Bu iki anahtar arasında herhangi bir bağlantı bulunmamaktadır [4].

Simetrik algoritmalar iki kategoriye ayrılabilir. Bazıları düz metin üzerinde bir seferde tek bir bit (veya bazen bayt) çalıştırır; bunlara akış algoritmaları veya akış şifreleri denir. Diğerleri düz metin üzerinde bit grupları halinde çalışır. Bit gruplarına blok adı verilir ve bu algoritmalara blok sistemler veya blok şifreleri denir. Modern bilgisayar algoritmaları için örnek verirse; bir blok boyutu 64 bittir, analizi engelleyecek kadar büyük ve uygulanabilir olacak kadar küçüktür [5].

Bununla beraber simetrik sistemler tek bir anahtar üzerine işlem yapar ikinci bir anahtara gerek duymaz. Bundan dolayı şifreleme ve şifre çözme aynı anahtar üzerinden yapılır. Simetrik anahtar şifrelemesi Şekil 1.4'de gösterilmektedir.



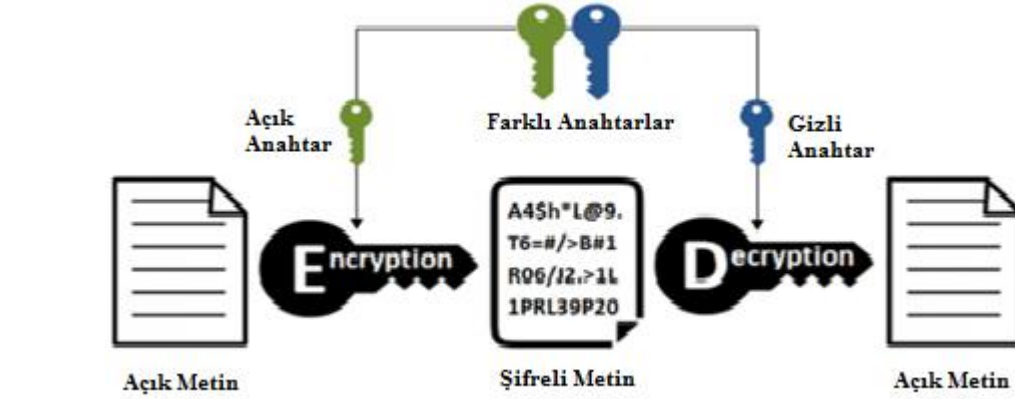
Şekil 1.4. Simetrik anahtar ile şifreleme.

### 1.2.2. Asimetrik Algoritmalar

Anahtarın gizli tutulduğu şifreleme sistemlerinde anahtar yönetimi konusunda aksaklıklar oluşabilmektedir. Bu sorunu çözebilmek amacıyla, *Whitfield Diffie* ve *Martin Hellman*, 1976 itibarıyla açık anahtarlı ilk sistemleri ortaya atmıştır. Anahtarın açık tutulduğu sistemlerde temelde iki tip uygulama vardır, bunlardan bahsetmek gerekirse, şifreleme ve dijital imza uygulamaları denilebilir. Uygulamaların kullanıldığı bu sistemlerde açık ve gizli anahtar olmak üzere iki tip anahtar bulunur. Açık anahtar isminden de anlaşılacağı üzere erişimi açıktır, gizli anahtar ise istenilen kişinin erişebileceği şekilde saklanır. Böylelikle kullanıcılar arasında aynı anahtar bilgisini saklama durumu ortadan kalkar. Sistemin işleyişi göz önüne alındığında sistemin aksamadan devam edebilmesi için açık anahtarların kullanımı yeterli olur ve bununla beraber gizli anahtarın gönderilmesine ya da bir başkasına iletilmesine gerek kalmaz. Bu sistemde kaygı duyulacak tek nokta açık anahtarın doğru ve güvenilir bir şekilde uygulanabilmesidir. Şifreleme yapabilmek için sadece açık anahtar kullanmak yeterli olabilir, ancak gönderilen şifreli mesajı sadece kullanılan açık anahtarın tamamlayıcısı olan gizli anahtar çözebilmektedir. Bundan başka, açık anahtarlı şifreleme sadece gizliliği sağlamak amacı ile değil, kimlik denetimi diğer bir deyişle sayısal imza ve daha birçok teknik ile kullanılabilir [1].

Asimetrik anahtar sistemini kullanan tüm uygulamalarda aleni anahtar ve saklı anahtar arasında bir ilişki mutlaka bulunur. Bunun getirdiği olumsuz etkilerden biri de bu tip sistemlere açık anahtar üzerinden bir saldırı olanağı sunmasıdır. Bu duruma karşı güvenlik sağlayabilmek için de açık ve kapalı anahtar arasındaki ilişkiyi olabildiğince karmaşık bir hale getirmek gerekir [14]. Ama genellikle bu şifrelemeleri yaparken

aradaki ilişkiyi şu ana kadar çözülmemiş matematiksel işlemler ile yaparak anahtarın gizliliği korunmaya çalışılır. Şekil 1.5’de açık anahtarlı şifreleme yapısı gösterilmektedir.



Şekil 1.5. Asimetrik anahtar ile şifreleme.

### 1.2.3. Asimetrik ve Simetrik Algoritmaların Avantajları ve Dezavantajları

Açık anahtarlı şifrelemenin en bariz üstünlüğü, gizli anahtarın hiçbir şekilde karşı tarafa gönderilmiyor olması, anahtarın bir başkasının eline geçme ihtimalini ortadan kaldırmaktadır ve daha fazla güvenlik sağlamaktadır. Simetrik anahtarlı sistemlerde ise bu durumun tersi ortaya çıkmaktadır. Anahtarın tek olması, şifreleme ve şifre çözmede aynı anahtarın kullanılmasından dolayı anahtarın karşı tarafa ulaştırılması gerekmektedir. Bu da ekstra bir güvenlik problemi olarak anahtarın istenmeyen kişilerin eline geçme ihtimalini ortaya çıkarır.

Açık anahtarlı sistemlerin önemli üstünlüklerinden birisi reddedilemez sayısal imzalar üretebilmesidir. Simetrik anahtarlı şifreleme yapan sistemlerde bazen kullanıcılar bilgi iletimi yaparken üçüncü bir kişiye ihtiyaç duyarlar. Bunun sonucunda taraflar arasında gizli anahtarın kötüye kullanımı konusunda anlaşmazlıklar ortaya çıkabilir. Fakat açık anahtarlı sistemlerde kişiler kendi anahtarlarının güvenliğini kendileri koruduğu için böyle bir durum söz konusu olamaz. Bu özelliğe “reddedilemezlik” denilir.

Açık anahtarlı sistemler yapıları gereği gizli anahtara sahip olan sistemlere göre genellikle çalışma hızı olarak yavaş kalır. Çoğu gizli anahtarlı sistemler, açık anahtarlı sistemlere göre daha hızlıdır. Her iki sistemi ortak bir paydaya alıp beraber kullanmak ise en etkili sonuç olacaktır.

Simetrik şifrelemede, kullanıcı açık anahtarına yapılacak herhangi bir başarılı saldırı karşısında, kullanıcının anahtarının yerine muadili koyulup, bu kullanıcıya ait mesajlara erişilebilir, daha sonra gelen mesajlar değiştirilebilir ve kullanıcının mesajları simetrik anahtarıyla şifrlenerek iletilebilir. Bundan dolayı bazen açık anahtarlı sistemleri kullanmak gereksizdir. Bunun yerine simetrik (gizli) anahtarlı sistemler kullanılmalıdır. Örneğin mesaj ya da bilgi paylaşımı yapacak kişiler herhangi bir sanal ortam ya da bir kurye kullanmadan birebir görüşerek ortak bir anahtar belirleyebilir. Sahip ve bunları düzenleyebilen bir kişinin bulunduğu ortamlarda açık anahtarlı şifreleme güvenliğini kaybetmektedir. Ancak kullanıcı sayısı arttıkça güvenlik için kullanımı idealdir.

Asimetrik şifrelemenin önem arz etmediği durumlardan biri ise tek kullanıcı sistemlerdir. Örneğin kişisel dosyalarınıza şifre koyarak güvenliğini sağlamak istediğinizde, herhangi bir gizli anahtar algoritmasıyla kendi kişisel şifrenizi anahtar olarak kullanarak şifreleme yapmak daha anlamlı olur. Genel olarak halka açık anahtarlı sistemler, kullanıcı sayısının fazla olduğu genel ortamlar için uygundur. Açık anahtarlı sistemler gizli anahtarlı sistemlerin bir alternatifi değil, birbirlerini güvenlik açısından tamamlayacak tamamlayıcılardır. Örneğin, “gizli anahtarları açık ağlar üzerinden taşımak için açık anahtarlı şifreleme kullanılır.”

Bir şifreleme algoritmasının performansı;

- Saldırlara karşı dayanım süresi
- Toplam şifreleme ve deşifreleme süresi
- Şifreleme ve çözüm sürecindeki kullanılan bilgisayar alanı
- Bu algoritmaya bağlı şifreleme uygulamalarının uyumu
- Bu uygulamaların yayılmasında kolaylık ya da algoritmaların standartlaşması
- Algoritmanın uyumu sistemin gerektirdiği şartlara göre belli olur [6].

## 2. DNA YAPISI

21. yüzyıla her yerden bilgi akan dijital çağ denilebilir. İnternet, iletişim ve veri işlemlerinde bir devrim niteliğindedir. Ağ güvenliği ile ilgili konular yakın geçmişten bu yana yapılan araştırmalardan büyük ilgi görmüştür. Ancak veri güvenliği ve gizlilik söz konusu olduğunda, tehlikeli etkiler kaçınılmazdır. Her birey verilerinin yetkisiz erişimini engelleyerek, verilerinin güvenliğini ve bilgi gizliliğinin korunmasını ister. Gizli verilerle ilgilenen tüm alan ya da kuruluşlar için aynı ölçütler geçerlidir. Bilgi güvenliği, herhangi bir alanda birey seviyesinden ulusal seviyeye kadar, bilgi gizliliği ve güvenliği etrafında döndüğü için son derece önemlidir.

Bilgi güvenliğinin başlıca yönlerini sıralarsak cihaz güvenliği, veri güvenliği ve içerik güvenliği denilebilir. Veri güvenliğinde, veri internet üzerinden aktarıldığında büyük bir endişe kaynağı olduğu söylenebilir. İnternetin büyümesi de saldırganların ve bilgisayar korsanlarının büyümesine yol açmıştır. Böylece güvenlik büyük bir endişe haline gelmiştir. Bu güvenlik sorunlarına uygun düzeyde çözüm, güçlü kriptografik teknikler kullanılarak sağlanabilir. Güvenlik alanında DNA hesaplama kavramı, kriptografi teknikleri ile birleşen ve çok katmanlı güvenli bir algoritma oluşturan biyomoleküler kavramlardan faydalanabileceğinden iyi bir teknik olarak görülebilir.

DNA hesaplama, DNA'nın biyolojik moleküler yapısını temel alan ve Adleman'ın yaratıcı yaklaşımının bir sonucu olarak ortaya çıkmıştır. Moleküler biyoloji kavramlarıyla bilgisayarları uyarlayan disiplinler arası bir alandır. Bu, biyomoleküler kavramların ve kriptografik tekniklerin bir entegrasyonu olmuştur. Güvenlik için yeni bir bilim dalının başlangıcı denilebilir, daha sonra bu alanda DNA kavramlarına dayanan birçok teknik ortaya çıkmıştır [7].

DNA kriptografisi, DNA hesaplama alanındaki araştırmalara dayanan yeni doğan bir kriptografik alandır. Bir gram DNA  $10^6$  TB'lık mükemmel bir depolama kapasitesine sahiptir, bu da birkaç gram DNA'nın dünyadaki tüm verileri saklama kapasitesine sahip olabileceğini gösterir. Depolama kapasitesi dışında DNA çeşitli güvenlik özelliklerini destekler, böylece DNA'da depolanan bilgiler kriptografi teknikleri kullanılarak korunabilir.

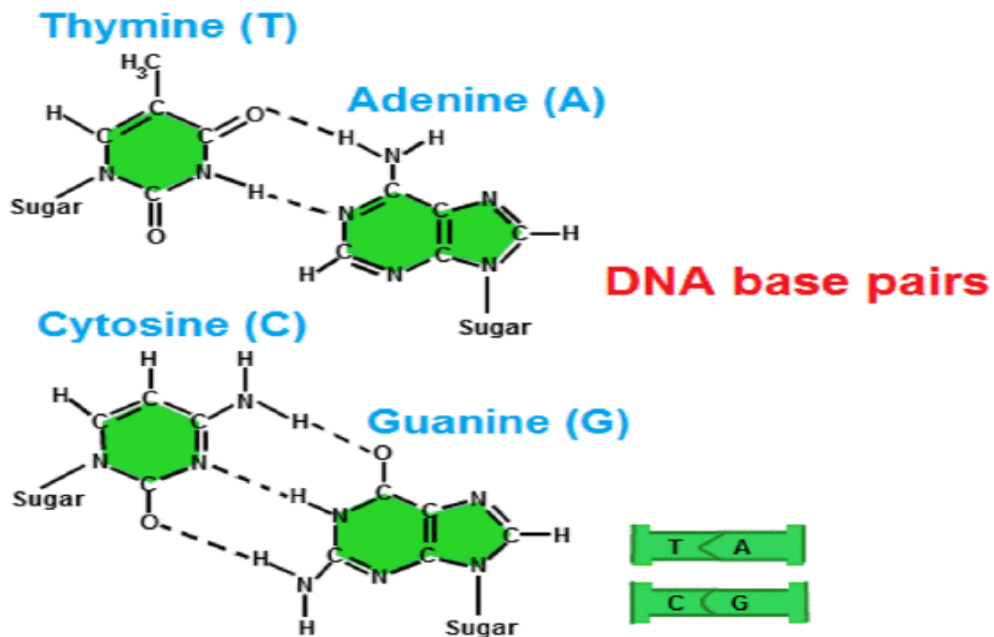


DNA, hücredeki potansiyel olarak anlamlı bilginin nihai deposudur. Hücre yaşamın temel birimidir ve DNA tüm canlı hücrelerin genetik mavi baskısıdır. DNA ilk olarak 1869'da İsviçreli doktor Friedrich Miescher tarafından tanımlanmış ve izole edilmiştir. DNA, tüm yaşam tarzlarının bir mikrop plazmasıdır. Nükleotidlerden oluşan bir çeşit biyolojik makro moleküldür. DNA uzun bir polimerdir. DNA'nın monomer birimleri nükleotidlerdir ve polimer bir "polinükleotid" olarak bilinir. Her nükleotid üç bileşenden oluşur.

- Azotlu Baz
- Beş karbon şeker
- Bir fosfat grubu

DNA'da bulunan ve sadece azotlu bazda farklılık gösteren dört farklı nükleotid türü vardır. Dört nükleotide dört baz için "Steno" olarak bir harfli kısaltmalar verilir. DNA, çift sarmallı bir nükleotid dizisi olarak bilinir. Dört baz, A, G, C, T'den birini içerir ve her nükleotid sırasıyla Adenin için A, Guanin için G, Sitozin için C ve Timin için T anlamına gelir. Adenin ve Guanin purinler olarak bilinir, Timin ve Sitozin ise pirimidinler olarak bilinir. A ve T bazları çift bağ ile eşleştirilir, C ve G ise üçlü bağ ile bağlanır. G ve T gibi başka bir bağ türü de mümkündür, ancak oluşan bağın mukavemeti doğal DNA dizilerindeki kadar güçlü olmadığı görülmüştür [21].

Şekil 2.1'de DNA baz çiftlerinin kimyasal yapısı gösterilmiştir.



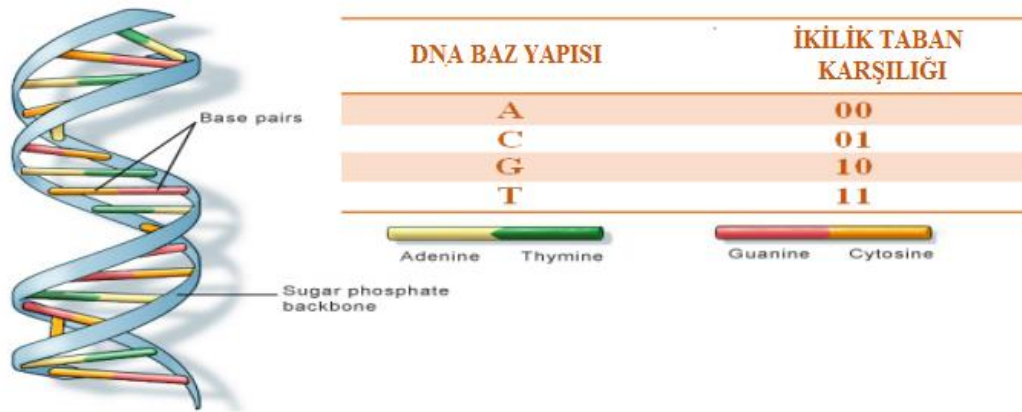
Şekil 2.1. DNA baz çiftlerinin kimyasal yapısı.

## 2.1. DNA STEGANOĞRAFİSİ

Steganografi sanatı yaklaşık 2500 yıldır kullanılır. Bilgiyi, gizli ve güvenli hale getirmek için bir yardımcı teknoloji olarak nitelendirilebilir, etkisi her zaman devam ettiği için varlığını korur. Steganografi görüntüyü, yalnızca gönderenin ve hedeflenen alıcının gönderdiği mesajı algılayabileceği şekilde değiştirir. Görünmezdir ve bu nedenle tespiti kolay değildir [7].

DNA Steganografisi, bir şifreleme tekniği olmamasına rağmen kriptografi tekniği olarak kabul edilir. DNA Steganografisi tekniğinde Steganografi ve DNA'nın özelliklerinden yararlanılır. DNA Steganografisi ilk olarak Crater Bancroft tarafından, İkinci Dünya Savaşı'na ait ünlü ve gizli bir bilgiye uygulanmıştır ve bilgi başarıyla tekrar geri alınmıştır. Bahsedilen yöntem DNA kodlu bir mesajın bir genomik DNA örneği içinde gizlenmesini ve ardından DNA örneğinin bir mikro noktaya gizlenmesini içermektedir. Bu kimlik doğrulama yöntemi genomik steganografi kullanımının başlangıcını işaret etmiştir. Şekil 2.2'de A, C, T, G bazlarının 00,01,10,11 olarak kodlaması gösterilmiştir.

Bu kodlama düz metin mesajını, DNA zincirine dönüştüren bir tekniktir. DNA gibi görülse bile bazı bilgileri gizler. Düz metindeki karakterler eşdeğer ikili değerlere dönüştürülür ve bu ikili değerlerin her biri iki bite bölünür. DNA'nın nükleotid bazları bu iki bit ile değiştirilebilir [8].



Şekil 2.2. DNA dijital kodlaması.

Günümüzde biyoloji ve kriptografi alanı birleşmeye başlamıştır. DNA uygulaması, DNA ve bir kerelik pedlerle (OTP-One Time Pad) birlikte DNA şifreleme sistemlerine uygulanabilir. Ayrıca doğru kullanılırsa sistemi kırmanın neredeyse imkânsız olduğu söylenebilir. Bir defalık pedin boyutu şifreleme sistemine bağlıdır ayrıca OTP şifreleme

şemaları için çeşitli prosedürler kullanılır [2].

OTP anahtarlarının kullanımı ve boyutu ile verilerin modern yöntemlerden daha yüksek gizliliği sağladığı görülmüştür. Ayrıca DNA şifrelemesinde anahtarın çok büyük miktarda veri için üretilebileceğine inanılırken, modern yöntemlerde anahtarın yalnızca daha küçük bir veri uzunluğu için üretildiğine inanılmaktadır. DNA yönteminin daha kısa sürede daha geniş bir veri yelpazesi için gizlilik sağladığı açıktır [8].

| Text | Code | Text | Code | Text | Code  | Text  | Code |
|------|------|------|------|------|-------|-------|------|
| A    | ATAT | a    | ACGC | 1    | GATA  | /     | CATC |
| B    | ATAG | b    | TATG | 2    | GATG  | <     | CAGA |
| C    | ATAC | c    | TAGA | 3    | GATC  | >     | CAGT |
| D    | ATGA | d    | TAGT | 4    | GAGA  | ?     | CAGC |
| E    | ATGT | e    | TAGC | 5    | GAGT  | :     | CACA |
| F    | ATGC | f    | TACA | 6    | GAGC  | "     | CACT |
| G    | ATCA | g    | TACT | 7    | GACA  | [     | CTAT |
| H    | ATCT | h    | TACG | 8    | GA CT | ]     | CTAG |
| I    | ATCG | i    | TGAT | 9    | GACG  | {     | CTAC |
| J    | AGAT | j    | TGAG | 0    | GTAT  | }     | CTGA |
| K    | AGAG | k    | TGAC | ~    | GTAG  | x     | CTGT |
| L    | AGAC | l    | TGTA | !    | GTAC  | °C    | CTGC |
| M    | AGTA | m    | TGTG | @    | GTGA  | α     | CTCA |
| N    | AGTG | n    | TGTC | #    | GTGT  | β     | CTCT |
| O    | AGTC | o    | TGCA | \$   | GTGC  | γ     | CTCG |
| P    | AGCA | p    | TGCT | %    | GTCA  | δ     | CGAT |
| Q    | AGCT | q    | TGCG | ^    | GTCT  | ε     | CGAG |
| R    | AGCG | r    | TCAT | &    | GTCG  | θ     | CGAC |
| S    | ACAT | s    | TCAG | *    | GCAT  | λ     | CGTA |
| T    | ACAG | t    | TCAC | (    | GCAG  | μ     | CGTG |
| U    | ACAC | u    | TCTA | )    | GCAC  | Δ     | CGTC |
| V    | ACTA | v    | TCTG | -    | GCTA  | Space | CGCG |
| W    | ACTG | w    | TCTC | +    | GCTG  | Enter | CCTA |
| X    | ACTC | x    | TCGA | W    | GCTC  | '     | GCGC |
| Y    | ACGA | y    | TCGT | _    | GCGA  | ,     | CATA |
| Z    | ACGT | z    | TCGC | ;    | GCGT  | .     | CATG |

Şekil 2.3. DNA dijital anahtar kodlaması.

Yukarıda Şekil 2.3’de ikili DNA bazlarının örnek bir dijital anahtar kodlaması gösterilmiştir.

## 2.2. DNA KRİPTOGRAFİSİ

DNA kriptografisi, güçlü kriptografik teknikler üretmek için çeşitli araştırmaların yapıldığı günümüzde ortaya çıkan yeni bir alan denilebilir. DNA kriptografisi, DNA hesaplama veya geleneksel kriptografik yaklaşımla gerçekleştirilebilir. DNA

hesaplamasına dayanan DNA kriptografisi hem simetrik hem de asimetrik anahtar kriptografisi ile gerçekleştirilebilir. DNA mikro dizisi, DNA fragmentasyonu, DNA hibridizasyonu, merkezi dogma vb. gibi yaygın tekniklerden bazılarını içeren moleküler teoriyi kullanır. DNA moleküllerinde bulunan geniş paralellik ve olağanüstü bilgi yoğunluğu, şifreleme, kimlik doğrulama ve imza gibi kriptografik amaçlar için uygulanabilir.

Geleneksel kriptografik yaklaşıma dayalı DNA kriptografisi anahtar oluşturma, şifreleme ve şifre çözme sürecinden oluşur. Ancak DNA şifrelemede geleneksel tekniklerden farklılıklar vardır. Anahtar dizilerini (ATCGCCAG) zaman aralığı gibi gören bir DNA biçimi kullanıldığında, anahtar oluşturma ve şifreleme işlemi sırasında düz metinden dönüştürerek üretilen şifreli metin DNA biçimindedir. Şifre çözme işlemi sırasında DNA formundan düz metine çevrilir [24].

DNA kriptografisi hem simetrik hem de asimetrik anahtar şifrelemesine dayanmaktadır. Ancak asimetrik anahtar yerine simetrik anahtarla gerçekleştirilmesi daha kolaydır. Ayrıca biyo-hesaplama karmaşıklığı istenmeyen kişiler tarafından erişimini daha zorlaştırır. DNA moleküllerindeki muazzam yoğunluk ve tekdüzelik ile ilgili kriptografik çalışmalar da hala tartışılmaktadır.

### **2.2.1. LİTERATÜR ÖZETİ**

2003 yılında Jie Chen, moleküler teoriye dayalı DNA şifreleme yaklaşımını tek seferlik pedi ve 2 boyutlu görüntünün şifreleme / şifre çözme işlemini gerçekleştirmiştir [25].

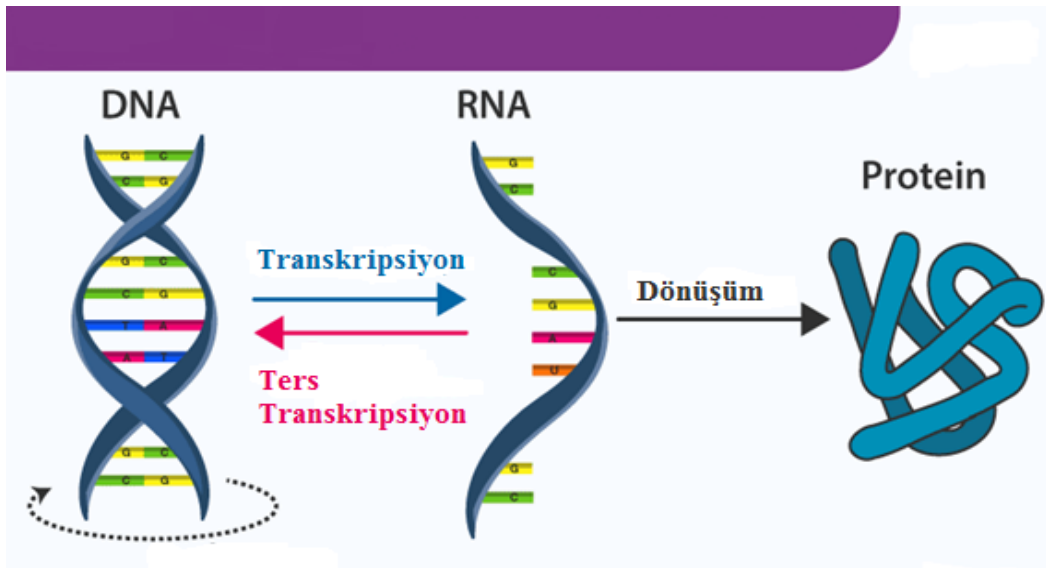
2004'te Ashish Gehani ve ekibi, Vernam ve Shannon'a göre moleküler yaklaşımı ve mükemmel gizliliğe sahip tek seferlik ped kavramını kullanarak DNA kriptografisinin temelini atmıştır. Tek kullanımlık pedin mucidi DNA çipine ve tek seferlik pede dayanan bir şifreleme ve şifre çözme yöntemi önermişler, kullanılan bu teknik ile istenmeyen kişiler tarafından şifrelenmiş mesajın tahmin edilmesini çok zorlaştırmışlardır [26].

2005 yılında Kazuo Tanaka ve ark. Açık Anahtar'a (tek yönlü) dayalı DNA şifreleme yaklaşımını önermiştir. Bu yaklaşımda, PKA (Negatif Baz Logaritması) için katı destek karışımı ve PKB (Baz ayırma Sabitinin Logaritması) için ODN (Optik Dağıtım Ağı) karışımı kullanarak genel anahtarların oluşumunu açıklamışlardır. Anahtarlar oluşturulduktan sonra mesaj, DNA sentezleyicisi ile tekrar sentezlenir. Daha sonra kodlanan mesaj dizisi, başka bir genel anahtar ile bağlanan anahtarlardan birinin

yardımla bir DNA dizisinde kodlanır. Ardından önceki işlemin sonucu immobilizasyon işlemine dahil edilir ardından kodlanmış DNA dizisinin kodunu çözmek için amplifikasyonun gizli dizi yardımla yapıldığı, PCR (Polimeraz zincir reaksiyonu) amplifikasyonuna iletilir [28].

2006 yılında Sherif T. Amin ve ekibi anahtar dizilerinin genetik veri tabanından elde edildiği ve her iki uçta da (gönderici ve alıcı) aynı kaldığı, simetrik anahtara dayalı DNA şifreleme yaklaşımını önermiştir [29].

2008 yılında Anıl Verma ve arkadaşları “Adhoc” ağlarının güvenliğini sağlamak için sözde DNA kriptografi yaklaşımını kullanan “Mobil Adhoc” ağlarına (MANET’ler) güvenli yönlendirme için yeni bir paradigma önermiştir. Adhoc ağı sabit bir altyapısı olmayan, her düğümünü bir ana bilgisayar ve yönlendirici görevi gördüğü bir ağı yapısıdır. Mobil Adhoc güvenlik saldırılarına karşı savunmasız olan merkezi bir otoritenin bulunmadığı kablosuz bir ağıdır. Kullandıkları sözde DNA kriptografi yaklaşımı moleküler biyolojinin merkezi dogmasına dayanmaktadır. Bu kavrama mesajların DNA’da nasıl saklandığı daha sonra mRNA’ya (transkripsiyon) ve ardından şifreli metnimiz olan proteinlere aktarıldığı kavram denilebilir. Şekil 2.4’de bu durum görselleştirilmiştir. Ayrıca şifreli metin güvenli kanal üzerinden istenen alıcıya gönderilir ve her iki uçta tek kullanımlık pedli simetrik anahtar kullanılır [31].



Şekil 2.4. DNA merkezi oluşumu.

2008’de Guangzhao Cui ve çalışma ark. iletişim sırasında güvenlik önlemi sağlamak için DNA sentezi, DNA dijital kodlama ve PCR amplifikasyonunu kullanan açık

anahtar şifreleme tekniğini önermiştir. Bu şifreleme tekniğinin yüksek gizlilik gücü vardır [32].

2010 yılında, Lai Xuejia ve ark. DNA çipinin problemlerle üretildiği DNA mikrodizi / çip teknolojisine dayanan bir DNA genel anahtar şifreleme sistemi önermiştir. Şifreleme işlemi için bir “prob” seti ve şifre çözme işlemi için başka bir set kullanılmıştır [34].

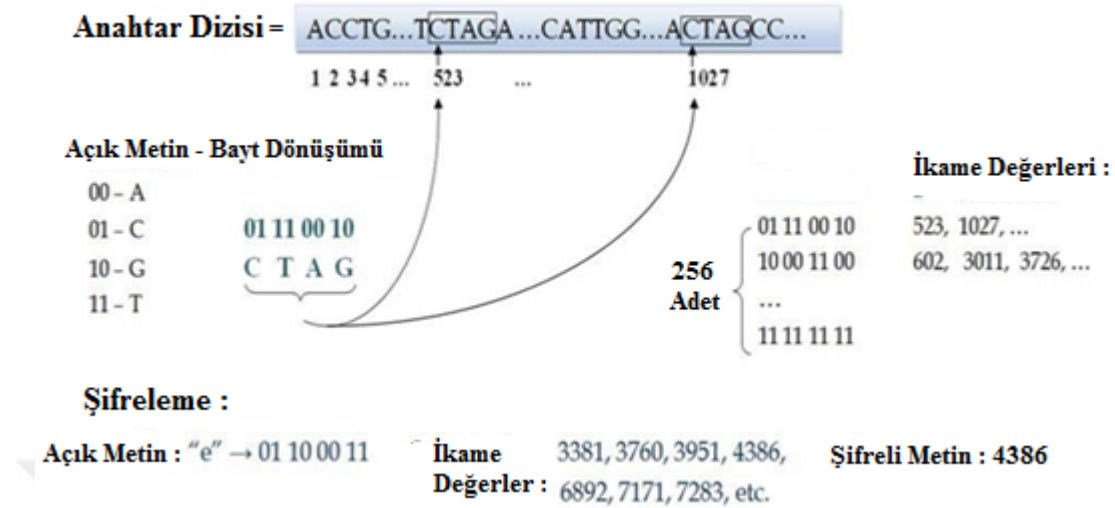
2011’de Deepak Kumar ve Shailendra Singh DNA dizilerine dayalı yeni bir gizli veri yazma tekniği önermişlerdir. Bu algoritmayı, düz metin olarak “HELLO” kelimesinin basit bir örneğini kullanarak açıklamışlardır. Üçyüzelli bitlik bir “ssDNA (Tek İplikli Deoksiribonükleik Asit)” ile tek seferlik ped anahtarı oluşturmuşlardır. Düz metinden 70 kat daha uzun ve simetrik anahtar şifreleme kullanarak düz metin üzerinde şifreleme ve şifre çözme gerçekleştirmişlerdir. Bunun sonucunda tam anahtarı bulmak için  $4^{310}$  farklı “ssDNA” dizisi arasında arama yapılması gerekir ki bu da nerdeyse imkansızdır [35].

2012’de Sabari Pramanik ve Sanjit Kumar Setua DNA moleküler yapısı ve hibridizasyon tekniğini kullanan yeni bir paralel DNA kriptografi tekniği önermişlerdir. Bu teknikte zaman gereksinimini kesinlikle en aza indirmişlerdir. Mesajın gönderen ve alıcı arasında nasıl güvenli bir şekilde değiştiğini bir örnekle açıklamışlar [2].

2012 yılında Yunpeng Zhang ve ark. DNA molekülünün küçük parçalarının birleştirilmesi mantığına dayanan bir DNA kriptografisi önermişlerdir. Algoritmalarında göndericinin düz metni ikili diziye ve ardından uzun DNA zincirine nasıl dönüştürdüğünü açıklamışlardır. Ardından daha da küçük DNA zincirlerine bölündüğünü açıkça belirtmişlerdir. Kısa zincir implantasyonunun anahtarı parçalarda yer alır ve alıcıya şifreli metin olarak iletilir. Ardından alıcı şifresini çözer ve düz metni elde etmek için parçaları yeniden birleştirmeye başlar [36].

2013 yılında Olga Tornea ve Monica E. Borda DNA indekslemesine dayanan DNA tabanlı Şekil 2.5’de gösterilen bir şifreleme tekniği önermişlerdir. Rastgele DNA dizisini genetik veri tabanından alır. Alıcıya güvenli bir iletişim kanalıyla gönderilen tek seferlik ped anahtarı kullanılır. Şifreleme mekanizmaları düz metni ASCII (American Standard Code for Information) koduna dönüştürür ve ardından onu DNA dizisine (A, C, G ve T) dönüştürülen ikili biçime çevrilerek oluşturulur. Yeni oluşturulan DNA dizisi, anahtar dizide aranır ve dizin numaralarını yazar. Elde edilen tamsayı sayı dizisi,

alıcı tarafından yalnızca anahtar ve dizin işaretçisi kullanılarak şifresi çözülüp açık metin elde edilir [37].



Şekil 2.5. DNA indeksleme.

DNA kriptografisi DNA hibridizasyonu, DNA sentezi, DNA mikroarray / çip teknolojisi, merkezi dogma, PCR amplifikasyonu ve tek seferlik ped temeline dayanır. DNA'yı, içindeki moleküler hesaplama nedeniyle geleneksel kriptografik tekniklerden benzersiz ve güvenli kılar. Ayrıca DNA simetrik anahtar şifrelemesine tek seferlik pedin eklenmesi, onu daha güçlü ve güvenli hale getirir ayrıca kaba kuvvet saldırılarına karşı korur. Ayrıca geleneksel depolama cihazlarına kıyasla yüksek gizlilik gücü ve doğasında bulunan büyük depolama yoğunluğu sunar. IBM tarafından enerji verimli DNA bilgisayar çipinin ortaya çıkmasıyla, günümüzde bilgi işlem ve bilgi güvenliği alanında araştırmacılar tarafından parlak buluşların önü açılmaktadır. DNA'nın farklı alanlarda araştırmacıları büyüleyen birçok olumlu yönü olmasına rağmen, biyo-moleküler laboratuvarların az oluşu, çevre etkisi ve kuantum saldırıları gibi bazı yönler DNA gelişimi için hala bir sorun oluşturmaktadır [37].

2014 yılında Shruti Kalsi ve ark. DNA kriptolojisini ve derin öğrenme mantığını birleştirmişlerdir. Yapılan bu çalışmayı, NW algoritmasının anahtar üretim sürecinde kullanmışlardır. DNA kriptolojisinin derin öğrenme ve yapay sinir ağlarında kullanılabileceğini göstermişlerdir ve yeni bir araştırma alanı ortaya çıkarmışlardır [42].

2016 yılında Bonny B. Raj ve ark. DNA kriptografisi alanında yeni bir simetrik algoritma önermişlerdir. Bu algoritma çalışma süresi olarak geleneksel algoritmalara göre daha hızlıdır. Yapılan yeni algoritmanın kablosuz ağlarda kullanımı konusunda önerilerde bulunmuşlardır [43].

2017 yılında S Bismi Beegom ve Sangeetha Jose DNA kriptolojisine dayanan bir anahtar üretim tasarımı yapmışlardır. Bu tasarımın geleneksel anahtar üretim süreçlerine göre daha güvenli ve hızlı olduğunu öne sürmüşlerdir [44].

2018 yılında Sreeja Cherillath Sukumaran ve Misbahuddin Mohammed bulut sistemlerdeki güvenlik problemlerine çözüm odaklı bir algoritma önermişlerdir. Bu algoritma güvenliği arttırmak amacıyla DNA indeksleme ve steganografi tekniklerini ile oluşturulmuştur [45].

2019 yılında Kazi Md. Rokibul Alam ve ark. DNA kriptografisi ile dinamik mekanizmaları güvenli hale getirecek bir algoritma önermişlerdir. Bu algoritma üç aşamalı bir şifreleme sistemi ile çalışmaktadır [46].

2020 yılında Prema T. Akkasaligar ve Sumangala Biradar tıbbi görüntülerin güvenliğini korumak için DNA tabanlı bir görüntü şifreleme algoritması önermişlerdir. Bu algorithmada DNA ve hiperkotik harita teknikleri kullanılmıştır ve güvenlik açısından daha emniyetli bir algoritma oluşturulmuştur [47].

Genel olarak DNA kriptolojisi ile ilgili yapılmış çalışmaları göstermek ve detaylarını açıklamak amacıyla aşağıda Çizelge 2.1’de DNA ile ilgili işler yıl, çalışma ismi, kullandığı anahtar yapısı ve içinde barındırdığı teknoloji açısından gösterilmiştir.

Çizelge 2.1. DNA kriptolojisi (Literatür çalışmaları).

| Yıl  | Çalışmanın İsmi                              | Anahtar Tipi      | Kullanılan Teknoloji                       |
|------|--|-------------------|--|
| 2003 | A DNA-based, Bimolecular Cryptography Design | Simetrik Anahtar  | Moleküler, OTP                             |
| 2004 | DNA-Based Cryptography                       | Simetrik Anahtar  | Moleküler, DNA Çip, OTP.                   |
| 2005 | Public-key system using DNA                  | Asimetrik Anahtar | Moleküler, DNA sentezi, PCR amplifikasyon. |
| 2006 | YAEA DNA Encryption                          | Simetrik Anahtar  | OTP  |



Çizelge 2.2. (devam) DNA kriptolojisi (Literatür çalışmaları).

|      |  |                   |   |
|------|--|-------------------|---|
| 2008 | DNA<br>Cryptography:<br>secure routing<br>in MANETs  | Simetrik Anahtar  | Merkezi dogma,<br>Moleküler biyoloji, OTP                     |
| 2008 | Encryption<br>Scheme Using<br>DNA  | Asimetrik Anahtar | DNA sentezi, DNA dijital<br>kodlaması, PCR<br>ampfilikasyonu. |
| 2010 | Asymmetric<br>Encryption and<br>Signature with<br>DNA  | Asimetrik Anahtar | DNA çip teknolojisi,<br>Hibritleştirme.                       |
| 2011 | Secret Data<br>Writing Using<br>DNA<br>Sequences   | Simetrik Anahtar  | OTP.  |
| 2012 | DNA<br>Cryptography  | Simetrik Anahtar  | Hybridization, One time<br>pad.                               |
| 2012 | DNA<br>Cryptography<br>Based on<br>Fragment<br>Assembly  | Simetrik Anahtar  | DNA Fragment assembly.  |
| 2013 | Security and<br>Complexity of<br>DNA Based Cipher  | Simetrik Anahtar  | OTP, DNA İndeksleme.  |
| 2014 | DNA Cryptography<br>and Deep Learning<br>using Genetic<br>Algorithm with<br>NW Algorithm for<br>Key Generation | Simetrik Anahtar  | Derin Öğrenme   |

Çizelge 2.3. (devam) DNA kriptolojisi (Literatür çalışmaları).

|      |   |                           |   |
|------|---|---------------------------|---|
| 2016 | Secure Data Transfer through DNA Cryptography using Symmetric Algorithm | Simetrik Anahtar          | Moleküler biyoloji, DNA dijital kodlaması |
| 2017 | An enhanced cryptographic model based on DNA approach                   | Simetrik Anahtar          | Moleküler DNA, OTP                        |
| 2018 | DNA Cryptography for Secure Data Storage in Cloud                       | Anahtar Kullanılmamıştır. | DNA Steganografisi and indexing           |
| 2019 | A technique for DNA cryptography based on dynamic mechanisms            | Asimetrik Anahtar         | Dinamik Frekans Tablosu                   |
| 2020 | Selective medical image encryption using DNA cryptography               | Simetrik Anahtar          | Çift Hiperkaotik Harita Teknikleri        |

### 2.3. DNA HESAPLAMASI

1994 yılında Adleman, “Hamilton yolu” problemi olan moleküler hesaplama kullanarak, kombinatoriyal problemlere çözümler sunarak DNA hesaplamanın temelini atmıştır. Yedi köşe içeren grafik örneğini bir algoritma kullanarak moleküler forma kodlayarak çözmüştür. Ardından bazı standart enzimler yardımıyla hesaplama işlemleri yapılmıştır ve bu durum kaba kuvvet yöntemi ile çözülmüştür [38].

1995’de Lipton yedi köşe içeren grafiği örnek almıştır. İki bitlik sayılar kodlamak için bir test tüpündeki DNA moleküllerini kullanarak “NP-complete (Deterministik olmayan polinom)” olarak adlandırılan başka bir problemi çözerek Adleman’ın çalışmasını genişletmiştir [39].

1996’da Dan Boneh ve ark. DES (Data Encryption Standard) olarak bilinen kriptografik amaçlar için kullanılan simetrik anahtar algoritmalarından birini kırmak için Adleman

ve Lipton tarafından kullanılan DNA hesaplama yaklaşımlarını uygulamıştır. Ekstraksiyon, DNA ile polimerizasyon, ve PCR ile amplifikasyon gibi bir test tüpündeki DNA zincirleri üzerinde biyolojik işlemler gerçekleştirmişlerdir. Ardından ikili dizilerin kodlamasına sahip DNA zincirleri üzerinde işlemler gerçekleştirmişlerdir. Daha sonra şifreli metinden hangi anahtarın kolayca tahmin edilebileceği hesaplamışlardır. DES devresini, arama tablosunu ve XOR kapılarını daha fazla değerlendirebileceği için çalışmada  $DES^{-1}$  çözümü oluşturulmuştur. DES saldırısı ve moleküler DNA uygulanarak DES'i yalnızca dört ayda kırmışlardır [40].

1997'de Qi Ouyang ve ark. başka bir "NP-complete" problem olan maksimal grup problemine çözüm üretmek için DNA moleküler teorisinin yaklaşımlarını uygulamıştır. Bunun sonucunda, DNA'nın etkinliğini gösterip, ardından zor problemleri ve bunun doğasında bulunan geniş paralellliği çözmek için operasyonlar hızlı hale getirilmiştir [41].

Çizelge 2.4. DNA hesaplaması (Literatür çalışmaları).

| Yıl  | Çalışmanın İsmi  | Çözülen Problem  | Kullanılan Teknoloji                         |
|------|--|------------------|--|
| 1994 | Molecular Computation of Solutions to Combinatorial Problems | Hamiltonian Path | DNA Moleküler Teorisi.                       |
| 1995 | DNA Solution of Hard Computational Problems                  | SAT              | DNA Moleküler Teorisi.                       |
| 1996 | Breaking DES Using a Molecular Computer                      | DES (56 bit)     | Moleküler Bilgisayar Tabanlı DNA Uygulaması. |
| 1997 | DNA Solution of the Maximal Clique Problem                   | Clique Problem   | DNA Moleküler Teorisi.                       |

Genel olarak DNA hesaplaması ile ilgili yapılmış çalışmaları gösterilmeye çalışıldı. Detaylarını açıklamak amacıyla yukarıda Çizelge 2.2’de DNA hesaplama ile ilgili işler yıl, çalışma ismi, çözüm sağladığı problem ve içinde barındırdığı teknoloji açısından gösterilmiştir.

Literatürdeki diğer çalışmalar incelendiğinde çeşitli problemlere çözüm üretmek adına DNA ile ilgili birçok çalışma yapılmıştır ve bu çalışmalar henüz gelişmekte olan DNA teknolojisine yön vermek açısından büyük önem taşımaktadır. Bu çalışmada daha önce literatürde yapılmamış olan bir modelleme tekniği önerilmiştir. Bu teknik hedef çözüm olarak DES ve Blowfish simetrik algoritmalarında DNA kullanılarak performans analizini yapıp, algoritmaların DNA modellemesi sonucundaki performans değerlerini incelemek amacıyla yapılmıştır.



### 3. BÖLÜM ASİMETRİK ALGORİTMALAR

#### 3.1. RSA ALGORİTMASI

R. Rivest, A. Shamir ve L. Adleman tarafından 1977 yılında RSA (Rivest-Shamir-Adleman) asimetrik şifreleme algoritması ortaya çıkmıştır. Bunun paralelinde diğer asimetrik şifreleme algoritmalarına (genel anahtar şifrelemesi) uyum sağlayacak tarz da iyileştirmeler yapılmıştır. RSA'nın kullanımından kısaca bahsetmek gerekirse açık anahtarlı sistemler ve dijital imza gibi uygulamalarda kullanılmaktadır [11], [12].

“PEM, S/MIME, MOSS ve PGP” gibi gizli haberleşme protokolleri temel olarak RSA kullanır ve bazı “SSL” gibi kullanım yerleri vardır.

RSA sistemi asimetrik algoritmalar sınıfında temel diyebileceğimiz bir algoritmadır. Bu şifreleme metodunun göze ilk etapta çarpan yönü ise mesajın alıcısının ve göndericisinin hiçbir şekilde daha önce bir bağlantı kurmamış olmasıdır ve kendi aralarındaki iletişimin güvenli bir durumda olmasıdır [4].

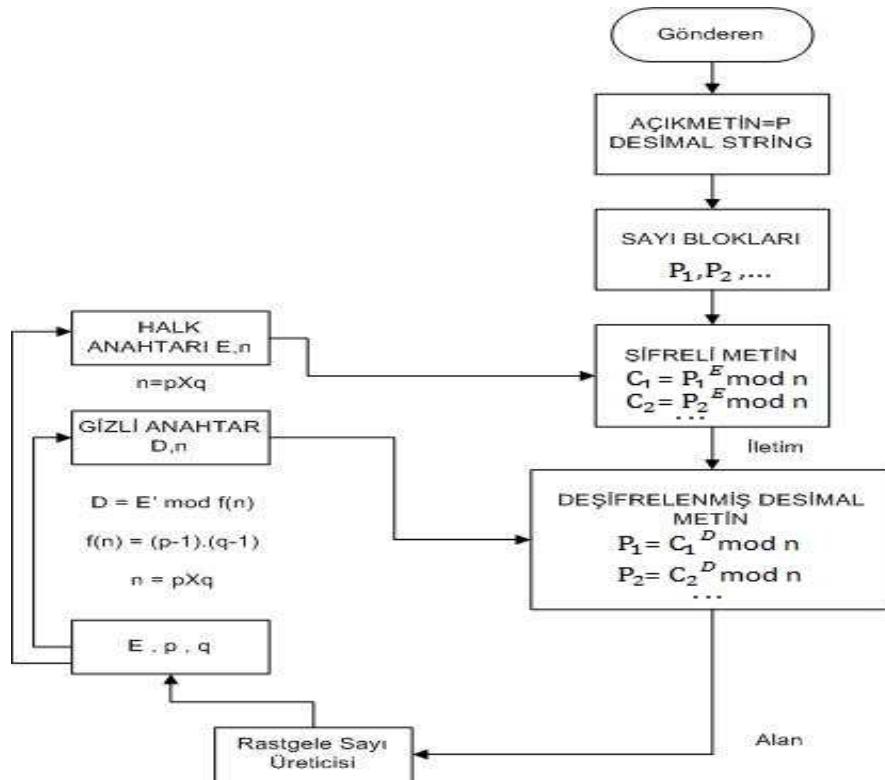
İçinde iki anahtar bulunduran bu sistemde, ilk bakışta gayet basit matematiksel ve mantıksal işlemler vardır. Bu anahtarlardan biri halka açık (public) herkes tarafından erişilen bir anahtardır diğer bir anahtar ise gizlidir. Şifreleme işlemi şahısların ortaya sunduğu açık anahtarlar tarafından şifrelenir. Kısacası size şifreli bir mesaj gönderilecekse, kendi açık anahtarınızdan faydalanılırken şifrelenen mesajı çözecek anahtar ise gizli anahtardır. Gizli anahtar çözüm yapacak kişide bulunmalıdır. Bunun getirdiği artı durum, iki anahtar kullanılır ve gizli anahtardan bağımsız bir şekilde güçlü bir şifreli mesaj elde edilebilir. Diğer bir artısı da kişiler ya da şahıslar birbiri ile herhangi bir iletişim halinde olmadan aralarında şifreli bir iletişim yapabilirler. Örneğin kredi kartı sistemlerini kullanarak internet alışverişinde bulunurken hiç bilmediğimiz bir internet sitesinin açık anahtarını alıp, kredi kartı numaramızla şifreleyip karşı tarafa iletiriz. Şifreli kart bilgisini, kartın sahibi dahil hiç kimse analiz edemez, yalnızca internet sitesinde bulunan saklanmış anahtarın sahibi, saklı anahtarla gelen şifrelenmiş içeriği çözebilir. Bu sayede kart sahibi de kartının numarasının kimse tarafından erişilemeyeceğinden emin olur [6].

İnternet ortamında, çok güvenli iletişim ortamları yer almaz. Buna karşılık olarak asimetrik şifreleme tekniğiyle internette güvenilir bir ortam oluşturmaya çalışılmıştır.

RSA şifreleme algoritmasının çalışması Şekil 3.1’de gösterilmiştir. Şekilde görüldüğü üzere asal olacak şekilde 2 adet asal sayı ( $p$  ve  $q$ ) üretilir. Bulunan asal sayıların çarpılmasıyla “ $n = p \cdot q$ ” işleminden “ $n$ ” gibi bir sayı ortaya çıkar. Bundan sonra  $n$  sayısından küçük ve  $(p-1) \cdot (q-1)$  sayısı ile aralarında asal olacak şekilde bir “ $e$ ” sayısı seçilir.

Daha sonra “ $E \cdot D = 1$ ” sayısının “ $(p-1) \cdot (q-1)$ ” çarpımına tam olarak bölünmesini sağlayan bir “ $D$ ” sayısı bulunur.  $E$  ve  $D$  değişkenleri sırasıyla, açık ve gizli anahtar olarak adlandırılırlar. Açık anahtarı  $(n, E)$  çifti, gizli anahtarı ise  $(n, D)$  çifti oluşturur,  $p$  ve  $q$  sayıları güvenlik açısından silinmeli ya da gizli anahtar gibi oda saklanmalıdır.

Gizli anahtar olan  $D$  sayısının açık anahtarı temsil eden sayılardan elde edilmesi imkânsız denilebilir. Bununla birlikte bir kişi  $n$  sayısını çarpanlarına ayırarak  $p$  ve  $q$  sayılarını ortaya çıkarırsa, gizli anahtarı da kolaylıkla çözebilir. Bunun sonucu olarak RSA sisteminin güvenliği algoritmanın çarpanlara ayırma proses sürecindeki işlem yüküne bağlıdır. Kısacası çarpanlarına ayrılmanın yolu bulunursa, RSA algoritmasının güvenilirliği de ortadan kalkmış olur ve algoritma çözülür [6], [10].



Şekil 3.1. RSA algoritması.

### 3.2. DSA ALGORİTMASI

Dijital İmza Algoritması (DSA)’i anlayabilmek için ilk olarak dijital imzanın tanımını inceleyelim. Sanal ortamda elektronik veriye eklenen veya mantıksal bir ilişki içinde olan elektronik ortamda bulunan imza olarak nitelenebilir, kimlik doğrulama nedeniyle kullanılan elektronik bir veridir. İmzamız şahsi olarak bize aitse, elektronik imza da herhangi bir elektronik ortama eklenen imzalı verinin bize ait olup olmadığını doğrulamak için kullanılır. Elektronik imza, kullanan kişinin kimlik teyidini sağlar. İmzalanan verilerin özelliğinin başka bir kişi tarafından modifikasyona maruz kalıp kalmasını (bütünlüğünün ihlal edilip edilmediğini) tespit eder.

Dijital imza son birkaç yılda birçok alanda yaygın olarak kullanılmakta olup işin özünde kullanım gayesi üç ana başlıktan oluşur [9].

Authentication (Doğruluk): Gönderenin kimliğinin doğrulanması, sizden gelen dijital imzalı herhangi belgenin gerçekten size ait olup olmadığının doğrulanmasıdır.

Data Integrity (Veri bütünlüğü): Verinin imzalandıktan sonra alıcıya ulaşana kadar herhangi bir şekilde modifikasyona uğratılmadığının garanti edilmesi olarak tanımlanabilir. Gönderilen mesajda yapılacak en ufak bir oynama dahi hash’lerin tutmamasına neden olacağından veri bütünlüğünün doğrulanması bu yolla sonuca vardırılabilir [13].

Non-Repudiation (İnkâr edilemezlik): İnkâr edilemezlik önemli bir unsurdur. İmzanın size ait olduğunun teyit edilmesi durumunda bunu ben göndermedim gibi bir argüman geçersiz olur.

Dijital İmza Algoritması, tarafından bir imza standardıdır. Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından Ağustos 1991’de ortaya çıkmıştır [15].

## 4. BÖLÜM SİMETRİK ALGORİTMALAR

### 4.1. DES ALGORİTMASI

DES (Data Encryption Standart) kullanıcılar tarafından günümüzdeki teknolojik platformlarda en çok kullanılan modern blok metotlu şifrelemedir. Protokol 64 bitlik veri bloklarını 56 bitlik anahtar kullanarak şifreleme ve şifre çözme için tasarlanmıştır. DES ayrıca 64 bit veri blokunu ve bununla beraber Feistel sistemini kullanan bir şifreleme protokolüdür. Amerikan Ulusal Standartlar Bürosu'nun çok çeşitli sistemlere uyumlu olacak, aşırı gizlilik gerektirmeyen kamu kuruluş verilerini veya diğer sektörleri kapsayan hassas ticari bilgilerin güvenliğini sağlayacak herkese açık bir algoritma geliştirme amacının ortaya koyduğu bir algoritmadır. DES algoritması 1976 yılında kabul görmüş ve 1977 yılında "Data Encryption Standard FIPS PUB 43" numarasıyla yayınlanmıştır. DES ile ilgili son güncellemelerin olduğu standart 1999 yılında tekrar ortaya çıkan "FIPS PUB 43-3" tür. 1986 yılı itibariyle de DES algoritması ISO tarafından uluslararası standart olarak kabul görmüştür. DES'in en büyük kullanıcı kitlelerinden biri bunu "EFT" ve "EFTPOS" işlemlerinde kullanan bankacılık alanıdır [17].

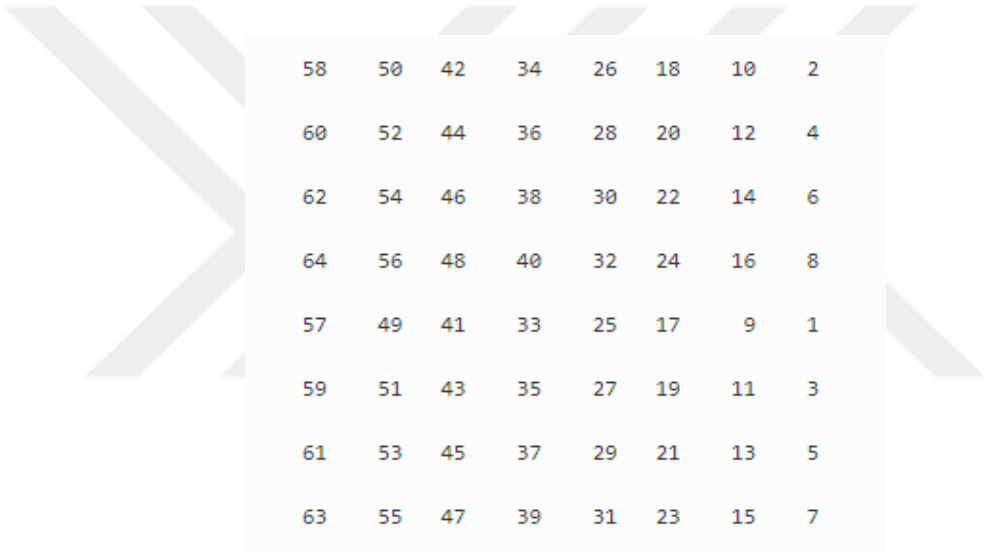
#### 4.1.1. DES Algoritmasının Yapısı

DES algoritması, 64 bit boyutundaki açık metni, 56 bit anahtar ile işleme sokarak, 64 bit şifreli metne dönüştüren blok mantığı kullanan şifreleme sistemidir. DES tanımında, bit sıralaması soldan sağa 1'den 64'e kadar yapılmıştır. Başka bir ifadeyle, bir numaralı bit birinci sekizlinin (byte) en yüksek anlamlı bitine ve 64 numaralı bit sekizinci sekizlinin en düşük anlamlı bitine karşılık gelecek şekilde oluşturulmuştur. IBM tarafından önem arz eden birtakım verilerin güvenliğini etkin bir şekilde sağlamak için güncellenmiştir ve 1976 yılı ve sonrasında da popülerliğini korumaktadır. Şekil 4.2'de algoritma yapısı gösterilmiştir [18].

Protokolün ilk bölümünde şifrelenecek metnin bit sıralamasının yeniden sıralandığı giriş permutasyonu (Initial Permutation: IP) (Şekil 4.1), sonunda da şifreli metnin bit sıralamasının değiştirildiği ve giriş permutasyonunun terslenmiş şeklindeki olan çıkış permutasyonu (Final Permutation, FP) vardır. Bu permutasyonlara diferansiyel kriptoloji



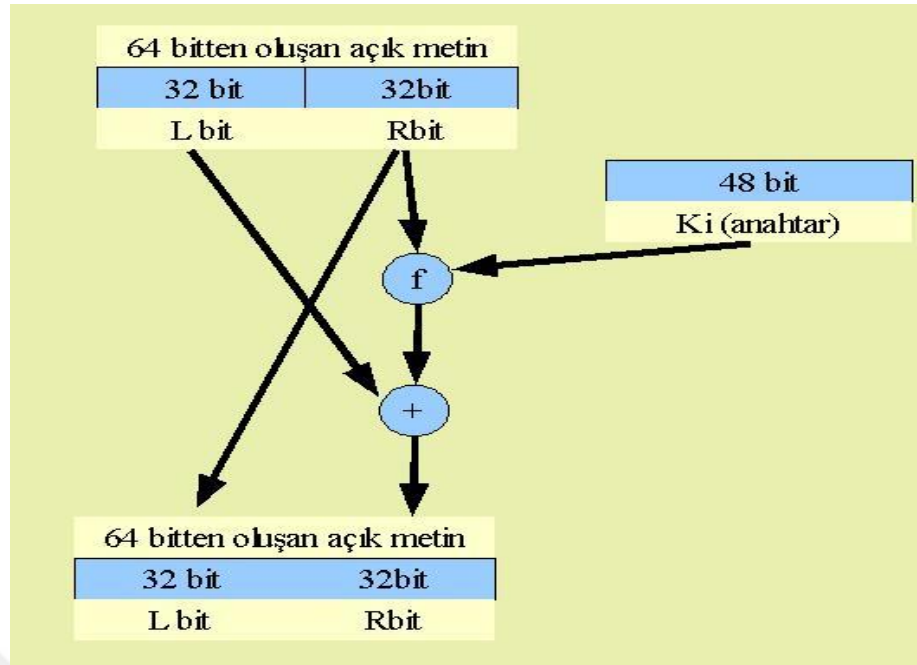
analizde dikkat edilmez. İki permutasyon arasında tekrarlı dönüşümlerden (round) meydana gelen, algoritmanın ana gövdesi vardır. Algoritmanın ana gövdesi, veriyi tam ortadan 32 bitlik eşit iki parçaya ayırıp, bunları sağ ve sol kısım diye böler. Algoritmanın temel işlemi “çevrim” olarak söyleyebiliriz. Her çevrimde, sağ ve sol diye ayrılan veriler ile anahtar düzenleme permutasyonundan elde edilen 48 bit anahtarlar kullanılarak güncel sağ ve sol diye ayrılan veriler ortaya çıkar. Çeşitli sayılarda çevrime sahip olan versiyonlar vardır. Bununla birlikte DES kendi içinde 16 çevrimden oluşacak şekilde şifreleme yapar. Her çevrimde verinin sağ yarısı ve her çevrim için farklı bir sıralamaya sahip olan 48 bit anahtar kullanılarak F fonksiyonuna sokularak çalışır. Verinin sol yarısı bu F fonksiyonunun çıkışı ile “XOR” işlemine tabii tutulur. İki çevrim arasında verinin iki yarısı çaprazlanmış olur [18].



|    |    |    |    |    |    |    |   |
|----|----|----|----|----|----|----|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9  | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

Şekil 4.1. IP (Giriş permütasyonu).

F fonksiyonu, verinin 32 bitlik sağ yarısını Şekil 4.3’de gösterilen E permutasyonunu (Expansion) baz alarak 48 bit olacak şekilde günceller. Elde edilen çıktı 48 bit anahtar ile XOR işlemine sokulur. Buradan ortaya çıkan bitler, her biri kendi özel tablolarını kullanan, 6 bit girişi 4 bite indirgeyen S1, S2, ....., S8 S-kutularına gönderilir. Örnek olması amacıyla S1 kutusu aşağıda tanıtılmıştır. Bu S kutularının çıktıları sırasıyla yazılır ve Şekil 4.5’de gösterilen P permutasyonu sokularak çıkış bitleri tekrar sırası güncellenir. Bu işlemlerin sonucunda F fonksiyonu son bulur.



Şekil 4.2. DES algoritması.

DES'in S-kutuları, 6 bitten 4 bite indirgeyen bir çeşit tablo olarak nitelendirilebilir. Her bir S kutusu 64 giriş değerini (6-bit), 16 çıkış değerine (4-bit) karşı indirger (Şekil 4.4). DES'in standart tanımlamasında S-kutuları, 0,1,2, .....,15 sayı değerlerinin karışık olarak sıralanıp dört ayrı permutasyon şeklinde sıralanmasıyla oluşur. Altı giriş bitinin ortadaki 4 tanesi sıralanacak değeri, ilk ve son bit (bit 1 ve bit 6) 4 ayrı permutasyondan hangisinin geleceğini belirtir. Örneğin; S1 kutusunda 100001 giriş değeri 4 çıkış değerine denk gelecek şekilde çıktı oluşturur. Örneğin; 0000 gibi bir değer çıktı olur [18].

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 32 | 1  | 2  | 3  | 4  | 5  |
| 4  | 5  | 6  | 7  | 8  | 9  |
| 8  | 9  | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 |
| 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 28 | 29 | 30 | 31 | 32 | 1  |

Şekil 4.3. DES genişletme permütasyonu (EP).

| S1 |    |    |   |    |    |    |    |    |    |    |    |    |    |   |    |
|----|----|----|---|----|----|----|----|----|----|----|----|----|----|---|----|
| 14 | 4  | 13 | 1 | 2  | 15 | 11 | 8  | 3  | 10 | 6  | 12 | 5  | 9  | 0 | 7  |
| 0  | 15 | 7  | 4 | 14 | 2  | 13 | 1  | 10 | 6  | 12 | 11 | 9  | 5  | 3 | 8  |
| 4  | 1  | 14 | 8 | 13 | 6  | 2  | 11 | 15 | 12 | 9  | 7  | 3  | 10 | 5 | 0  |
| 15 | 12 | 8  | 2 | 4  | 9  | 1  | 7  | 5  | 11 | 3  | 14 | 10 | 0  | 6 | 13 |

Şekil 4.4. S-Box örneği.

| P  |    |    |    |
|----|----|----|----|
| 16 | 7  | 20 | 21 |
| 29 | 12 | 28 | 17 |
| 1  | 15 | 23 | 26 |
| 5  | 18 | 31 | 10 |
| 2  | 8  | 24 | 14 |
| 32 | 27 | 3  | 9  |
| 19 | 13 | 30 | 6  |
| 22 | 11 | 4  | 25 |

Şekil 4.5. S permütasyonu.

Anahtar düzenleme permutasyonu, 56 bit anahtardan 16 tane 48 bit büyüklüğünde olacak olan K1, K2, ..., K16 şeklinde birbirinden farklı anahtar değerlerini ortaya çıkarır. Bu K1, K2, ....., K16 anahtarlar DES'in çevrimlerinde kullanılan F fonksiyonu girişi olarak işleme girerler. Başlangıçtaki anahtar bitleri, bit numarası sekiz ve sekizin katları olacak şekilde (8, 16, ...) eşlik biti l'den 64'e kadar isimlendirilir. Ortaya çıkan anahtar bitleri Şekil 4.6'da görülen PC-1 permutasyona sokularak yeniden güncellenir ve 28 bitlik C ve D olacak şekilde iki eş parçaya bölünür. C bölümünün bitleri anahtarın 57, 49, ..., 36. bitleri, D bölümünün bitleri de anahtarın 63, 55, ..., 4. bitleridir. Her bir çevrimde C ve D kaydedicileri Şekil 4.7'de gösterildiği gibi bir ya da iki bit sola kaydırılır. Elde edilen C ve D bölümleri bit sırasına göre birleştirilip l'den 56'ya kadar isimlendirilir ve Şekil 4.8'de gösterilen PC-2 permutasyonuna sokularak 48 bit anahtar elde edilir. Daha sonra anahtar ile genişletilmiş metin F fonksiyonuna sokulur. Şekil 4.9'da F fonksiyonu şeması tanımlanmıştır [18].

PC-1

|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| 57 | 49 | 41 | 33 | 25 | 17 | 9  |
| 1  | 58 | 50 | 42 | 34 | 26 | 18 |
| 10 | 2  | 59 | 51 | 43 | 35 | 27 |
| 19 | 11 | 3  | 60 | 52 | 44 | 36 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 |
| 7  | 62 | 54 | 46 | 38 | 30 | 22 |
| 14 | 6  | 61 | 53 | 45 | 37 | 29 |
| 21 | 13 | 5  | 28 | 20 | 12 | 4  |

Şekil 4.6. PC-1.

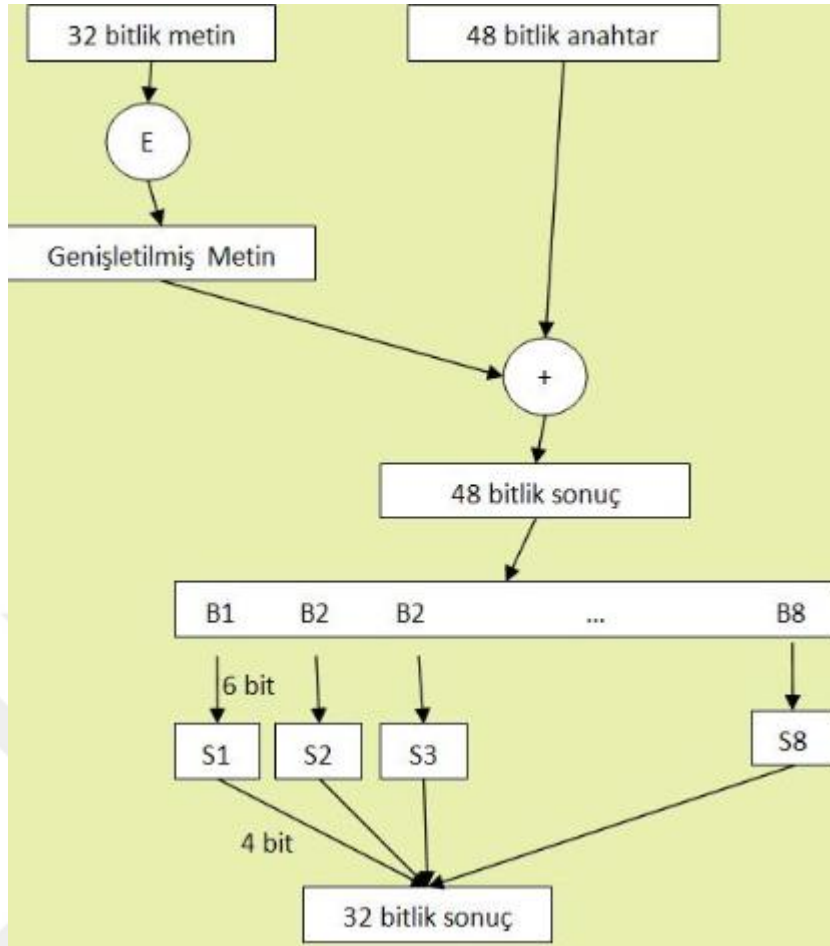
|                    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |
|--------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| Çevrim Numarası :  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Kaydırma Miktarı : | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2  | 2  | 2  | 2  | 2  | 2  | 1  |

Şekil 4.7. Kaydırma permütasyonu.

PC-2

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 14 | 17 | 11 | 24 | 1  | 5  |
| 3  | 28 | 15 | 6  | 21 | 10 |
| 23 | 19 | 12 | 4  | 26 | 8  |
| 16 | 7  | 27 | 20 | 13 | 2  |
| 41 | 52 | 31 | 37 | 47 | 55 |
| 30 | 40 | 51 | 45 | 33 | 48 |
| 44 | 49 | 39 | 56 | 34 | 53 |
| 46 | 42 | 50 | 36 | 29 | 32 |

Şekil 4.8. PC-2.



Şekil 4.9. F fonksiyonu.

## 4.2. AES ALGORİTMASI

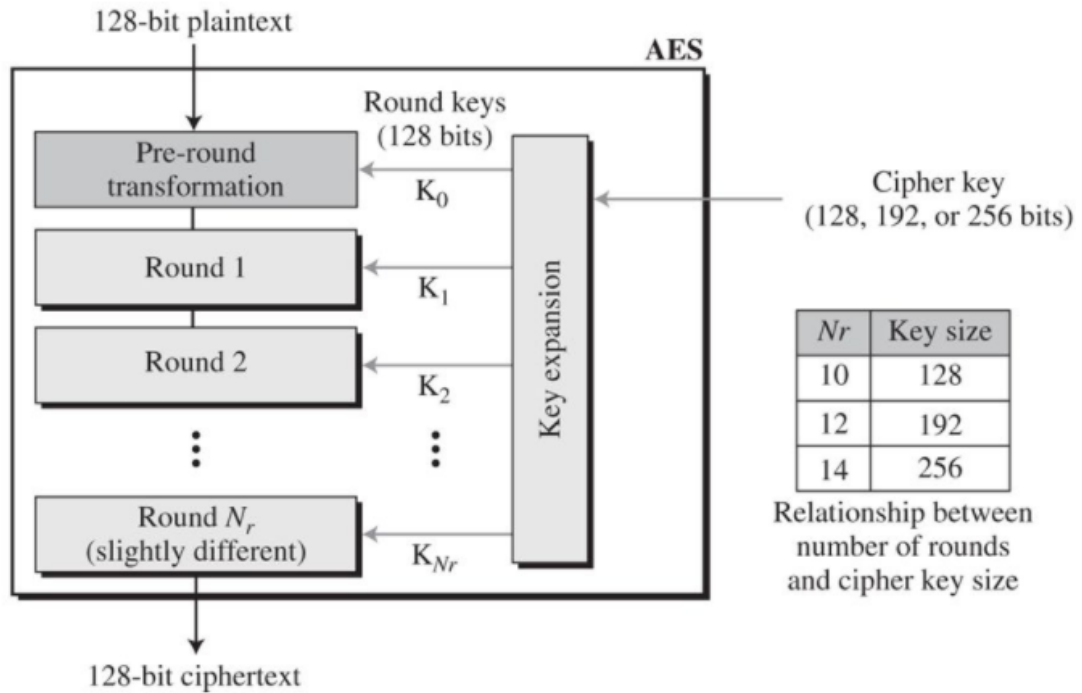
AES ABD (Amerika Birleşik Devletleri) vasıtasıyla onaylanmış bir şifreleme protokolüdür. Bilinen diğer ismi Rijndael algoritmasıdır, blok sistemine dayalı bir protokolüdür. DES algoritmasının gelişen teknoloji karşısında yetersiz kalması üzerine, güncel bir şifreleme sisteminin oluşturulması amaçlanarak NIST (National Institute of Standards and Technology) bünyesinde yapılan yarışmanın sonucunda iki Belçikalı bilim adamı olan Joan Daemen ve Vincent Rijmen çalışmaları sonucunda geliştirilen protokol, yeni bir şifreleme olarak yerini almıştır. Süre olarak bayağı vakit geçtikten sonra, ideal ve doğrulama yapılmasının ardından 26 Kasım 2001 itibariyle AES FIPS 197 standardı olarak NIST bünyesinde tanıtılmıştır. DES protokolüyle kıyaslayacak olursak AES algoritması daha güvenilir olmasının yanı sıra, kolay gerçekleylebilir oluşundan kaynaklı artıları vardır. Şifreleme standardının değişken anahtar ve veri blok boyutlarına uyumlu olmasıyla birlikte standart, sabit 128-bit'lik veri bloğu ile 128-bit,

192-bit ya da 256-bit'lik anahtar boyutlarını kullanmak için uyumludur. AES içerisinde 128-bit'lik veri blokları vardır bu bloklar kendi içinde her biri 32 bit den oluşan 4 kelime olarak düşünülür. AES ile şifreleme işlemine başlanırken 128-bit veya başka bir ifade ile 4 kelimeden oluşan veri bloğu durum dizisi içerisine kaydedilir ve algoritma sırasındaki gerekli işlemlerin tümü bu dizi vasıtasıyla uygulanır. Şifreleme için zaruri olan en son prosesin de bitimiyle durum dizisinin yeni hali çıkış olarak verilir [19].

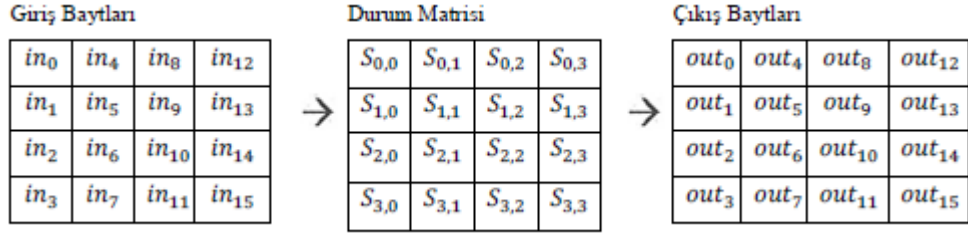
Örneğin;  $in_0, in_1, \dots, in_{15}$  den oluşan veri bloğu varsayım dizisine kaydedilir bunun akabinde gerekli hesaplamaların tamamı bu dizi esasında uygulanır. Gerekli hesaplamalarının son bulması ile birlikte şifrelenmiş veri çıkışa  $out_0, out_1, \dots, out_{15}$  şeklinde bir dizi olarak sonuç verir. Algoritmanın genel akış şeması Şekil 4.12'de gösterilmiştir.

AES protokolü çoğunlukla iki bloktan meydana gelir, birinci blok tur dönüşüm ikinci blok ise anahtar üretimi için kullanılır. Protokol tekrarlı bir şekilde çalışır, anahtar bloğunun boyutuna göre tekrar sayıları değişir. Örneğin; 128-bit, 192-bit veya 256-bit, tur dönüşüm işlemi sırasıyla 10, 12, 14 tekrar etme suretiyle işleme girer [4].

Tur dönüşüm sayılarının veri boyutuna göre değişimi Şekil 4.10'da gösterilmiştir.



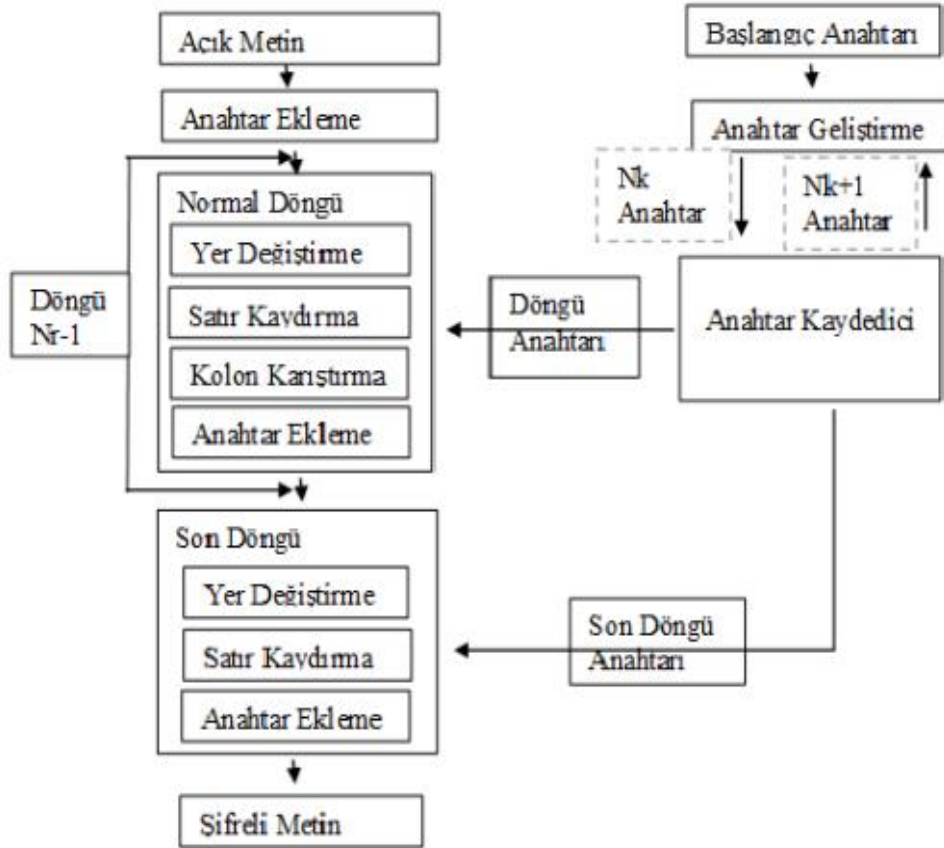
Şekil 4.10. AES tur sayısına göre anahtar sistemi.



Şekil 4.11. AES durum dizisi.

Algoritma çalışmaya başladığında ilk olarak şifreleme işlemine tabii tutulan blok, Şekil 4.11'e göre durum dizisine aktarılır. Şifreleme işlemi, giriş anahtarıyla durum dizisini toplama işlemiyle başlar. Anahtar bloğunun boyutuyla ilişkili tur dönüştürme işlemi olarak 10, 12, 14 defa olmak şartıyla çalışır. Algoritmaya ait blok sema Şekil 4.12'de gösterilmiştir.

Tur dönüştürme işlemi sırasında durum dizisi üzerinde satır sütun işlemlerinin yanı sıra, tur anahtarını toplama gibi alt işlemler dizisi kullanılır [16]. Tur dönüştürme işleminin çıktısı olarak ortaya çıkan 128-bitlik veri Anahtar imalat hesaplaması akabinde ortaya çıkan anahtar verisiyle toplama işlemine sokulur.



Şekil 4.12. AES akış diyagramı.

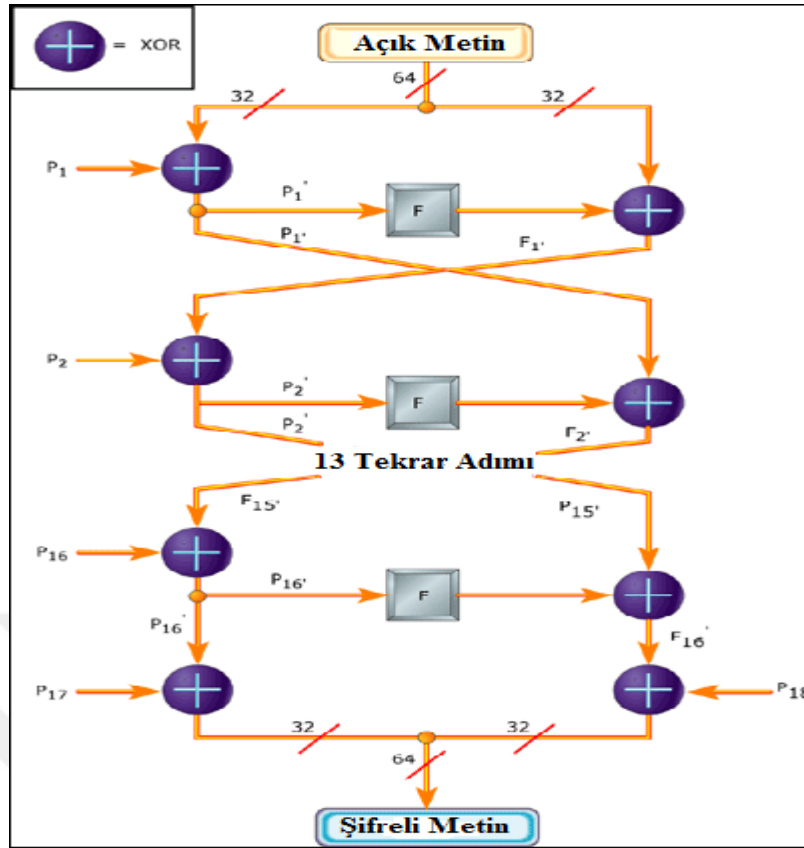
Anahtar imalat hesaplaması akabinde ortaya çıkan anahtar verisiyle toplama işlemine sokulur. En son çevrim işleminin de yapılmasıyla ve anahtar bloğu ile toplananın bir sonucu olarak şifrelenmiş blok üretilmiş olur. Bunun yanı sıra son turda gerçekleşen hesaplamalar, önceki turdaki hesaplama işlemlerinin çalışma mantığından farklıdır. Bununla birlikte son turda sütun karıştırma işlemi yapılmaz. Satırları kaydırma işleminin çıktısı tur anahtarı ile toplanır [19].

#### **4.3. BLOWFISH ALGORİTMASI**

Feistel ağı yapısını içinde barındıran blok şifrelemeye dayalı bir algoritmadır. Blowfish, Bruce Schneier tarafından 1993 yılında üretilmiş ve anahtar mantığı ile çalışan simetrik bir Block Cipher (öbek şifreleyici)'dir [23]. Blowfish algoritmasının çözümüne dair şimdiye kadar literatürde herhangi bir analiz yapılamamasına rağmen AES ya da Twofish gibi daha büyük boyutlardaki şifreleyicilere önemli derecede ilgi duyulmaktadır. Schneier; Blowfish'i bir algoritma sıfatıyla patentsiz bir şekilde modası geçen DES' in bir alternatifi olması amacıyla ve diğer algoritmalarındaki olumsuz yönlere çözüm odaklı olması için ortaya çıkarmıştır. Bahsedilen dönemlerde bulunan çoğu şifreleme protokolü o dönemlerde patentli ve devlet sırrı olarak saklanmaktaydı.

Blowfish, güvenli bir şekilde uygulama kolaylığı sağlayan simetrik bir blok şifreleyicisi olduğundan, verilerin şifrelenmesi ve korunması için. 32 bit ile 448 bit arasında değişen uzunluklarda bir anahtar kullanır ve bu durum verilerin güvenliğini sağlamak için yeterli gelmektedir. Blowfish patent ve lisans gibi birtakım prosedürlere bağlı değildir ve tüm kullanımlar için ücret talebi yoktur. Zayıf anahtar gibi bir dezavantajı olmasına rağmen, hiçbir saldırıya karşı kırıldığına dair herhangi bir çalışma yoktur [22].



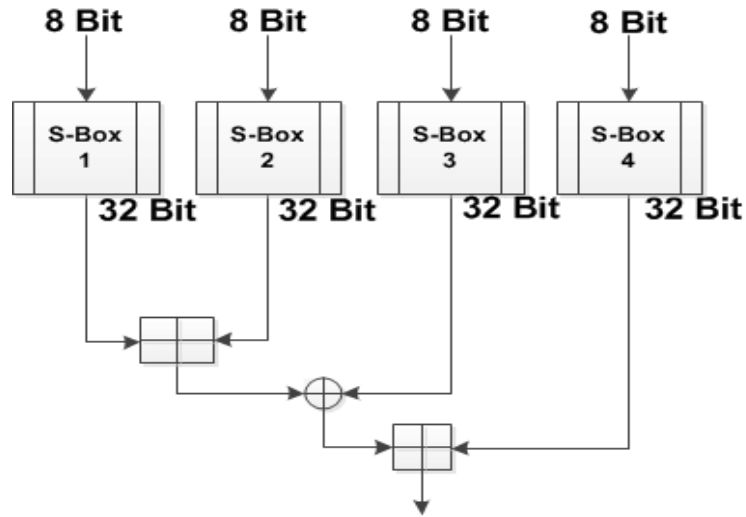


Şekil 4.13. Blowfish akış diyagramı.

Şekil 4.13’de totalde 16 adımdan meydana gelen Blowfish algoritmasının akış şeması gösterilmiştir. Şemaya göre her turda 32 bitlik proses devreye girer. Algoritma iki alt anahtar sırası (subkey array) tutar: 18-girisli P-sırası ve dört adet 256-girişli S- kutuları vardır. S- kutular 8-bit girdi kabul eder ve 32-bit çıktı verir. P-sırasının bir girişi her turda işleme girer ve son turdan sonra veri bloğunun her bir yarısı geri kalan kullanılmamış iki P-girişinden biri tarafından XOR yapılır.

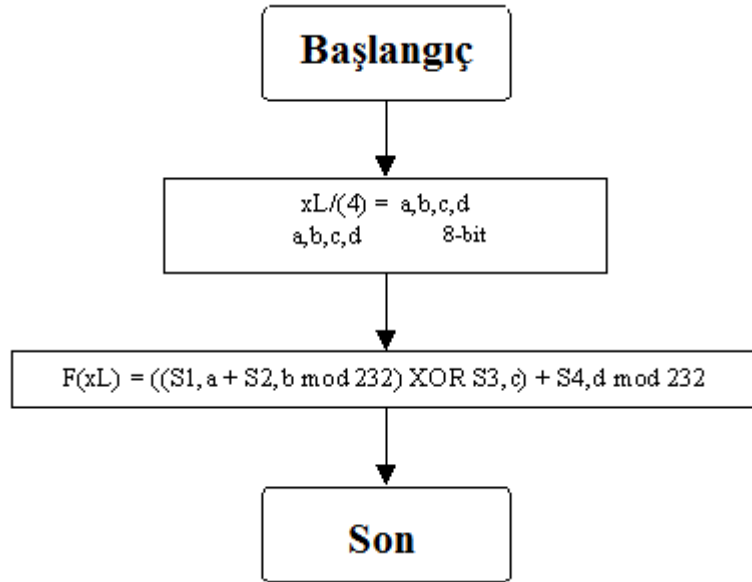
Her bir adım için değiştirme dizilerinden biri kullanılır. Son adımdan sonra veri bloğunun her iki yarısının veyası (özel veyası (XOR)) yapılır ve bu işlem sırasında artan iki adet yerine koyma dizisi de işleme sokulur.

Şekil 4.14’de her F-fonksiyonu için kullanılan metot gösterildi. Bu durumda metnin yarısı olan 32 bit boyutundaki veri her biri 8 bit olacak şekilde 4 parçaya bölünerek aşağıdaki S-kutularına giriş olarak girer ve bu kutulardan gelen çıktı aşağıda gösterildiği gibi işleme girer.



Şekil 4.14. Blowfish S-Kutuları.

Sonuçlar  $2^{32}$  esasında modulo işlemi yardımıyla hesaplanır daha sonra 32 bitlik sonucu elde etmek amacıyla özel veya (XOR) işlemine sokulmaktadır. Şifreleme işleminin çözülmesi için 17. ve 18. adımlarda kullanılan permutasyonun terslenmesi ve her adımda bulunan permutasyonun sırasıyla geri gidilmesi esasına dayalıdır. Algoritmada bulunan F fonksiyonu Şekil 4.15’deki gibidir.



Şekil 4.15. Blowfish F fonksiyonu.

Bu yöntemde baz alınan permutasyon dizileri ve ikame kutuları  $\Pi$  (pi) sayısından elde edilen sayılarla oluşturulur. Bilindiği üzere bu yöntem,  $\Pi$  sayısının tekrar etmeyen doğası nedeniyle güvenli kabul edilebilir.

Blowfish’in genel olarak duyulmuş güçlü bir kriptanalizi bulunmamaktadır. 64 bit öbek

büyüklüğü günümüzde çok kısa olarak görünse de  $2^{32}$  den fazla veri öbeğini şifrelemek doğum günü saldırıları nedeniyle açık metine ilişkin bilginin korunmasını tehdit edebilir. Yine de Blowfish şimdiye kadar güvenli bilinmektedir. Kısa blok boyutu, e-posta gibi günlük kullanıcı uygulamaları konusunda önemli kaygılar çıkarmasa da Blowfish veri saklama ya da daha büyük boyuttaki verilerin saklanmasıda problem olabilir.

Vincent Rijmen, doktora tezinde dört turdan fazlasını kıramayan ikincil-sıra diferansiyel saldırıyı sunmuştur.. Günümüzde, tam 16-tur'u kırabilecek, "brute-force search" haricinde bir yol görülmemiştir.

2005 yılında Dieter Schmidt, Blowfish anahtar tablosunu araştırmıştır, üçüncü ve dördüncü turların alt anahtarlarının ilk 64-bit kullanıcı anahtarından bağımsız olduğunu ortaya çıkarmıştır.

Blowfish anahtar değişikliği hariç; geniş kullanımda en hızlı küme şifrelerinden biri olarak kabul edilir. Her yeni anahtar diğer küme şifreleyicilerinden çok daha yavaş olan yaklaşık 4 KB'lık metni şifrelemek için ön işleme eşdeğerini gerektirir. Bu, bazı uygulamalarda kullanılmasını imkânsız kılar.

Bazı uygulamalarda Blowfish 4 kilobayt RAM'den biraz fazla büyük bellek alanına ihtiyaç duyar. Bu durum masa üstünde kullanılan ya da diz üstünde kullanılan ilk bilgisayarlar için bile sorun teşkil etmez ancak en birincil tip yani ilk üretim akıllı kartlarda ve gömülü sistemlerde çalışma konusunda engelleme yapar.

#### **4.4. TWOFISH ALGORİTMASI**

Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson tarafından üretilmiş ve 1998 yılında yayınlanan bu algoritma analiz edilmiştir.

Simetrik blok şifreleme algoritmalarından ve AES tipi şifreleyicilerin son hallerinden biridir ve AES kadar hızlıdır. Feistel iskeleti bulunur, DES'den farklarından biri anahtar kullanılarak üretilen değişen S-Kutularını (Substitution Box – Değiştirme kutuları) içinde barındırmasıdır. Bununla beraber 128 bitlik düz metni dört eşit parçaya bölerek işlemlerin büyük kısmını 32 bitlik veriler üzerinde uygular. Yüzyirmisekiz, 192 ya da 256 bit değişken anahtar uzunluğu vardır. AES'den farklı olarak eklenen iki adet 1 bitlik rotasyon, şifreleme ve şifre çözme algoritmalarını, birbirinden farklı yapmıştır. Bunun

AES standardı olarak NIST tarafından tasarımını başarılı bir şekilde desteklemesinden dolayı Twofish algoritması AES'in yerine önerilen algoritmalarından biridir. Twofish algoritmasının genel yapısı Şekil 4.16'da tanımlanmıştır.



#### 4.5. IDEA ALGORİTMASI

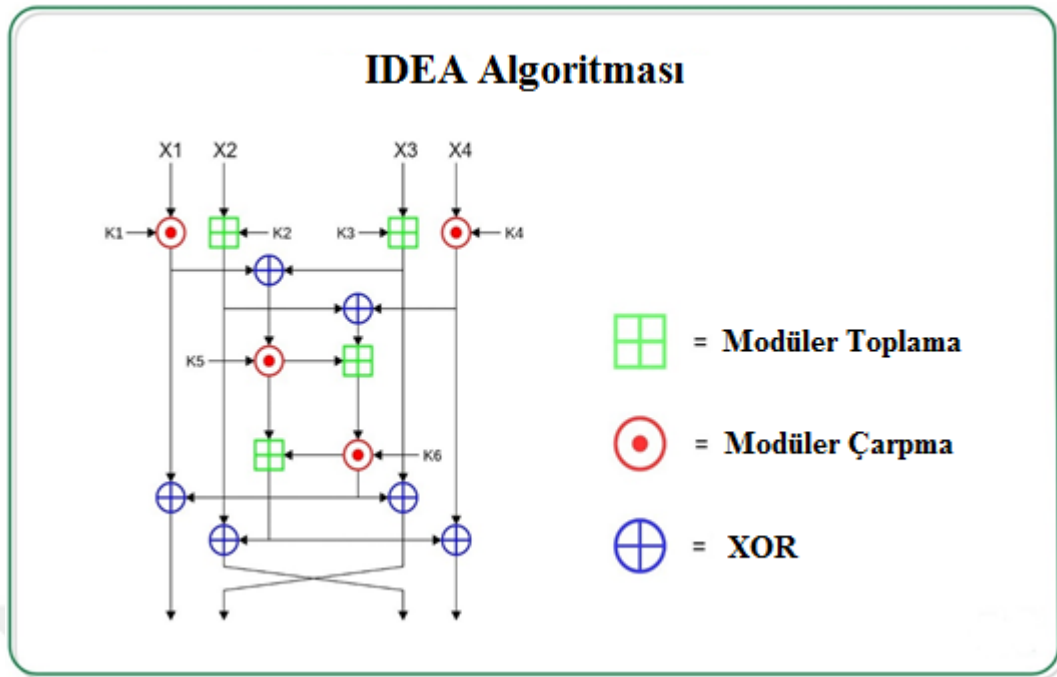
Uzun ismi “International Data Encryption Algorithm” olan IDEA, 1990’da Xuejia Lai ve James Massey tarafından tasarlanmıştır. 1991 yılında tasarlanan bir blok şifreleme algoritmasıdır. Bu algoritmaya DES alternatifi olarak üretilen PES’in geliştirilmiş hali olarak nitelendirilebilir. Aynı zamanda “Ascom Tech” firmasının patent sağladığı algoritmadır. IDEA algoritmasının popüler olmasındaki en büyük etkilerden biri de PGP uygulamasıdır. Bilinen en güçlü algoritmalarından biridir. Daha önce PES olarak bilinen bu algoritma modifikasyonlardan sonra bugünkü halini almıştır. İsviçre Zürih’de geliştirilmiştir. “Ascom Systec Ltd” adlı bir şirket tarafından patenti alınmıştır. Fakat ticari alanlar hariç lisans için ücreti istememektedir. PGP’nin yapı taşlarını oluşturan 2 ana algoritmadan bir tanesi IDEA’dır.

Altmış dört bit metini şifrelemek için 128 bit anahtar kullanan bir şifreleme algoritmasıdır. Bu anahtar, şifreleme ve şifre çözme işleminde aynı anahtar kullanır başka bir ifadeyle, simetrik bir algoritmadır ve 52 tane 16-bit anahtar üretir.

Algoritma 8 tanesi özdeş olan 9 fazdan oluşur. Altmış dört bitlik blok ilk 8 fazın her biri boyunca hızlı bir şekilde işleme dahil olur. Burada blok 16 bitlik dört parçaya bölünür her bir faza karşılık gelen altı alt anahtar metin yapısı baştan sıralanır. Sekizinci aşamanın çıktısı elde edildiğinde, blok son 4 alt-anahtarı kullanan bir son aşamaya girer.

Şifre çözme benzer bir modeli izler ancak duruma bağlı olarak ve alt anahtarların kullanım sırasını değiştirerek, her bir alt anahtarın toplamını veya çarpımını hesaplar [30].

IDEA’nın patentli bir algoritma olduğu bilinmektedir. IDEA algoritması çok yönlü matematiksel süreçlerin karışımından meydana gelmektedir. Algoritmanın çalışma şeması Şekil 4.17’de gösterilmiştir.



Şekil 4.17. IDEA yapısı.

Yardımcı anahtarların oluşumu, var olan 8 anahtarın kaydırma işlemine sokularak buradan türeyen anahtarlarla oluşturulur. Bu durum zayıflık olarak görünse bile bununla beraber matematiksel olarak herhangi bir olumsuz durumla karşılaşılmamıştır. Algoritmanın artı ve eksi yönlerini aşağıdaki gibi tanımlayabiliriz.

Olumlu yönleri;

- Algoritmalar süre açısından efektif derecede hızlıdır
- Algoritmaların donanımla kullanılması son derece basittir
- Gizlilik gibi güvenlik ile ilgili hizmetleri yapar

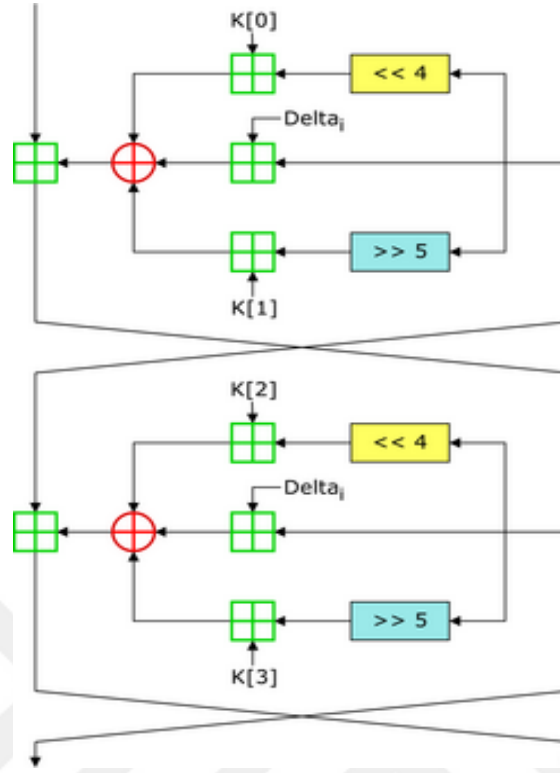
Olumsuz yönleri;

- Ölçeklenebilir değil
- Güvenli bir şekilde anahtar dağıtımı yapmak zordur
- Bütünlük ve kimlik doğrulama hizmetlerini uygulamak zordur.

#### 4.6. TEA ALGORİTMASI

Uzun ismi “Tiny Encryption Algorithm”dir. Tiny Encryption Algorithm (TEA), Cambridge Bilgisayar Laboratuvarın’da David Wheeler ve Roger Needham vasıtasıyla oluşturulmuştur. İlk olarak 1994 yılında Leuven’de Fast Software Algorithm (Hızlı Yazılım Şifreleme) şirketinde tanıtılmıştır. Tiny şifreleme protokolü blok şifrelemesine dayalı bir metin şifreleme uygular. Bunun yanında DES, Blowfish algoritmaları gibi Fiestel ağını kullanan şifreleme algoritmasıdır. En büyük avantajlarından bahsederek basitliği ve birçok şifreleme algoritmasından daha kısa satırdan meydana gelen sistemi önemli bir etkidir. Boyut olarak küçük olan kod satır uzunluğuna ve basit algoritma yapısına sahiptir. Özellikle kod uzunluğunun çok kısa olduğu yerleşik sistemlerde ve sağlık izleme sistemlerinde oldukça dikkat çeken ve kullanımı rahat olan bir şifreleme algoritmasıdır. TEA var olan şifreleme algoritmaları arasında eşsiz bir hız ve basitliğe sahiptir. TEA hafızada ters orantılı bir şekilde hızını artırır ve bunu yaparken 64 bitlik bloklar ile işlemlerini sürdürür. Bu 64 bit olan veri bloğu, 128 bit anahtar uzunluğu olan bir anahtar yardımıyla şifreleme yapar. 128 bitlik K anahtarı 4 eşit parçaya ayrılır, bu parçaların boyutu 32 bit olur.  $K = (K[0], K[1], K[2], K[3])$  değişebilmesine rağmen 64 adet Fiestel turu için 32 döngü önermektedir. (2 Fiestel turu = 1 döngü) Terminolojide 2’li yani çift tura “Cycle = Döngü” denmektedir. TEA algoritması Shannon’un desteklediği güçlü ve emniyetli bir blok şifreleme için yapılması gereken karıştırma (confusion) ve yayılma (diffusion) gibi özellikleri içinde barındırdığını söylenilebilir. Karıştırma gizli metin ve düz metin arasındaki bağlantıyı gizleme gayesiyle çalışırken, yayılma düz mesajdaki belli noktaların şifreli mesajda fark edilmesini ön plana çıkarmak için uygulanır. (Blok şifreler, Shannon’un önerdiği karıştırma ve yayılma tekniklerine dayanır.) Tam bir yayılma sağlar. Düz metnindeki tek bir bit girişi şifrelenmiş mesajda 32 bitlik bir değişikliğe neden olur. Modern bir bilgisayar veya çalışma alanındaki performansı etkileyici düzeydedir.

TEA için “2 Fiestel turu – 1 çevrim” yapısı aşağıdaki Şekil 4.18’de gösterilmiştir.



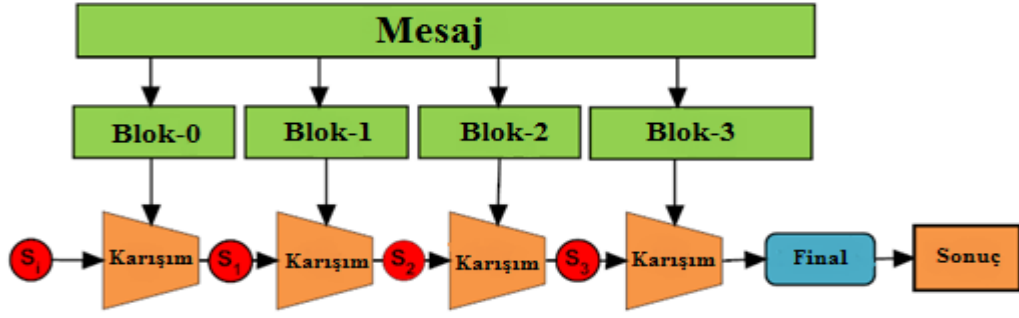
Şekil 4.18. TEA yapısı.

TEA algoritması, güvenlik seviyesi yüksek olan algoritmalarından biri olduğu söylenebilir. Massey ve Xuejia Lai vasıtasıyla oluşturulan IDEA algoritması ile denk olduğu varsayılabilir. IDEA’da kullanılan aynı karışık cebirsel kümeleri ve benzer bir çalışma prensibine sahip olsa bile, daha hızlı ve basit bir yapısı vardır [33].

#### 4.7. HASH ALGORİTMALARI

Hash (Karma) algoritması karma işlevidir, rastgele boyuttaki verileri sabit boyuttaki bir karmaya eşleyen matematiksel bir algoritmadır. Şifrelenecek verinin bir özetini kullanarak bir değer üretir ve metni parçalara ayırır, her parça için farklı bir değer kullanır ve başlangıçtaki değerın son hali mesaj özetini oluşturur. Ayrıca tek yönlü bir işlev olarak tasarlanmıştır tersine çevrilemez. Şekil 4.19’da Hash Fonksiyonu gösterilmiştir. Ancak, son yıllarda çeşitli karma algoritmalar güvenlik zafiyeti göstermiştir. Bu duruma örnek vermek gerekirse MD5 algoritması artık tersine çevrilmesi çok kolay olan bir kriptografik karma işlevi olmasının yansıya yaygın olarak bilinen bir karma işlevi olmuştur.



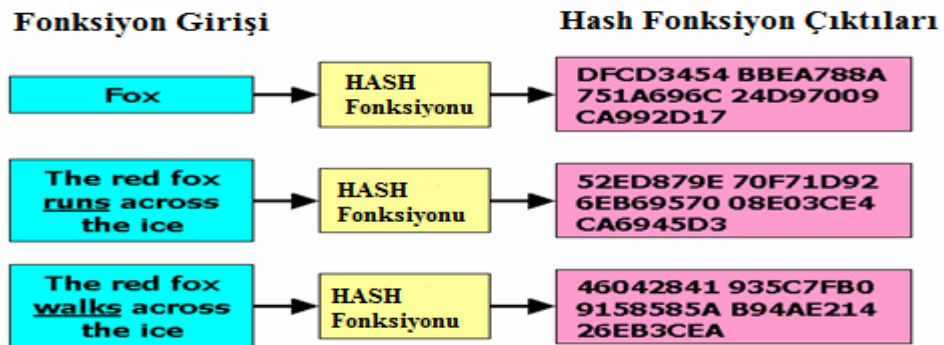


Şekil 4.19. Hash fonksiyonu.

İdeal şifreleme ve karma işlevinin nasıl olması gerektiğini ve kullanım özelliklerini sıralayabiliriz;

1. Her türlü veri için hash değerini hesaplamak hızlı olmalıdır.
2. Bir mesajı karma değerinden yeniden oluşturmak imkânsız olmalıdır (tek seçenek olarak kaba kuvvet saldırısı).
3. Karma çarpışmalardan kaçınmalıdır, her mesajın kendi karma değeri vardır.
4. Mesaj özetinden mesajı tekrar elde etmek imkânsız olmalı, bu özellikten dolayı hash fonksiyonları tek yönlüdür.
5. Hash fonksiyonunun çıkışı sabit uzunlukta olmalıdır.
6. Mesajdaki her değişiklik, hatta en küçük olanı bile karma değerini değiştirmelidir. Tamamen farklı olmalı buna çığ etkisi denilir.

Hash fonksiyonun çıktı boyutu açık metine göre daha uzundur, örnek çıktı değerleri Şekil 4.20’de gösterilmiştir.



Şekil 4.20. Hash fonksiyonu çıktıları.

## 5. DNA TABANLI KRİPTOLOJİ UYGULAMASI

Günümüzde hızla gelişen teknolojinin getirdiği imkân ve olanaklar birçok uygulamayı bilgisayar ve internet aracılığıyla yapmak noktasında ilerlediği söylenilebilir. Hem zaman hem de kolaylık sağladığı için bu durum git gide yaygınlaşmıştır. Bu durum sağladığı kolaylıkların yanı sıra bir takım güvenlik problemlerini de beraberinde getirmektedir. Bunlar kişisel verilerin, askeri ve kurumsal bilgilerin gizliliğini riske atmaktadır, bu noktada verilerin gizliliğini ve güvenliğini sağlamak açısından kriptografik işlemler önem kazanmaktadır.

Çalışmanın bu bölümünde, dördüncü bölümde bahsedilen simetrik şifreleme algoritmalarından Blowfish ve DES algoritmaları temel alınarak bir DNA tabanlı şifreleme yapıldı. İki algoritmanın hız, RAM, CPU açısından performans değerlendirilmesi gösterildi. Bu uygulama yapılırken PYTHON dili kullanılıp iki algoritmaya DNA tabanlı şifreleme modeli eklenip, algoritmaların yapısına eklenen DNA modeli detaylı olarak açıklanıp değerlendirildi.

### 5.1. ÖNERİLEN DNA TABANLI MODEL

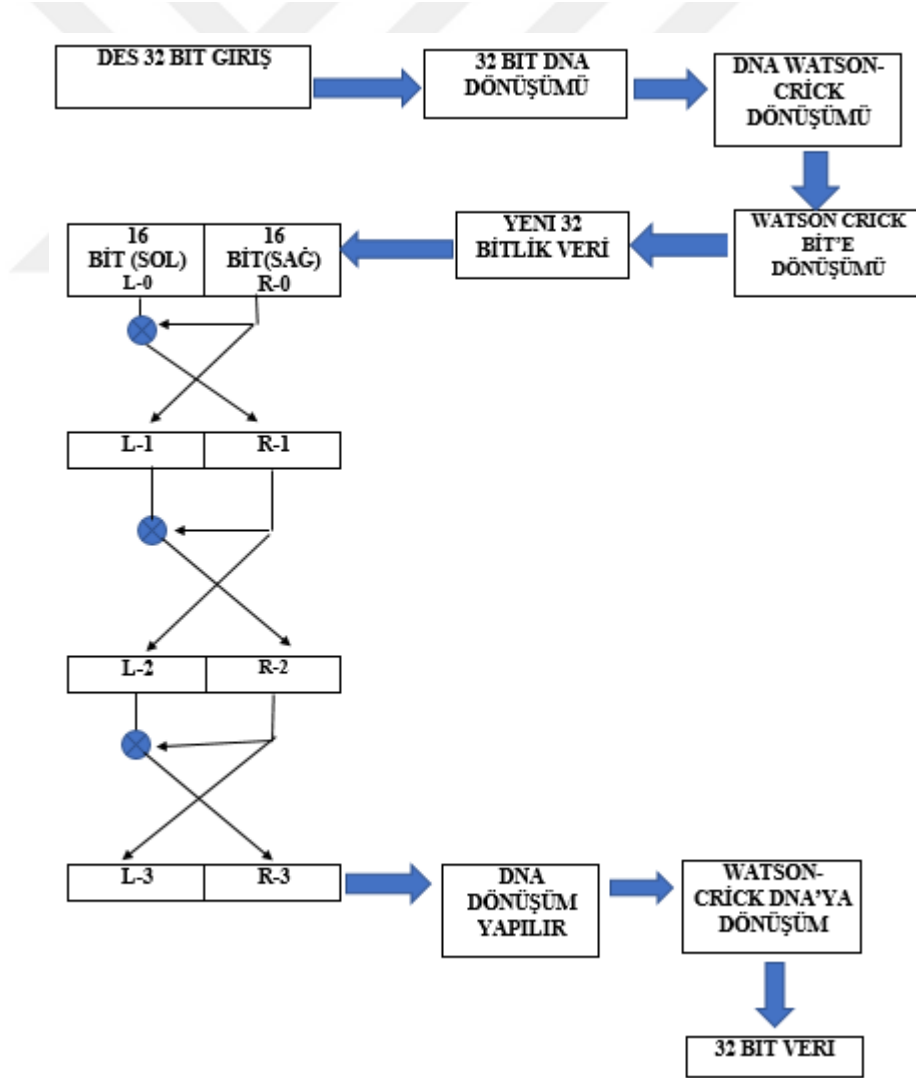
Kriptografi sektörü günümüz itibari ile çok önem kazanmıştır, her geçen gün yeni yöntemler ve metotlar ortaya çıkmıştır. Bunlardan bir tanesi olan DNA tabanlı şifreleme ise son geliştirilen popüler modelleme yöntemlerindendir. Bu model insan DNA yapısında bulunan dört bazın bilgisayar ortamında simule edilmesi temeline dayanan bir şifreleme metodudur.

Bu çalışmada, dört DNA bazı DES ve Blowfish algoritmalarına eklendi. İki simetrik şifreleme algoritmasının hız, RAM, CPU performansı açısından değerlendirmesi yapıldı. Algoritmaların bu modelleme sonucunda performans analizleri tablo ve grafikler ile detaylandırıldı. Ayrıca algoritmaların modellenmiş yapıları ile orijinal yapıları arasındaki performans farkı incelendi.

## 5.2. DES ALGORİTMASI İÇİN ÖNERİLEN DNA MODELİ

Tasarımı yapılan DNA tabanlı sistem simetrik bir yapıya sahip olup, tercih edilen iki şifreleme algoritmasının simetrik yapısını bozmayan bir DNA modeli içerir. Ayrıca önerilen sistem Feistel yapısına benzer 4 adımlı bir yapı olacak şekilde tasarlandı.

DES algoritmasının yapısı incelendiğinde işlem yükünün ve güvenliğin sağlandığı kısım algoritmanın F fonksiyonudur. Çünkü algoritma şifreleme işlemlerini ve anahtarı bu fonksiyon içinde kullanır. Bu fonksiyonun içi ne kadar karmaşık hale getirilirse algoritma o kadar güvenli ve dayanımı yüksek olur. Bundan dolayı algoritmaya yapılacak iyileştirme yada güvenliği artırma adımlarının bu aşamada uygulanması çok önemlidir. Ayrıca algoritma simetrik bir yapıda olduğu için simetrinin bozulmaması ve F fonksiyonun terslenebilir formunun bozulmaması önemlidir. Aksi durumda algoritma tek yönlü çalışıp sadece şifreleme gerçekleştirir ve geri açık metni veremez.



Şekil 5.1. DES algoritması için tasarlanan akış şeması.

Yukarıda Şekil 5.1’de DES algoritmasının içine eklediğimiz algoritmik yapının akış şeması gösterildi.

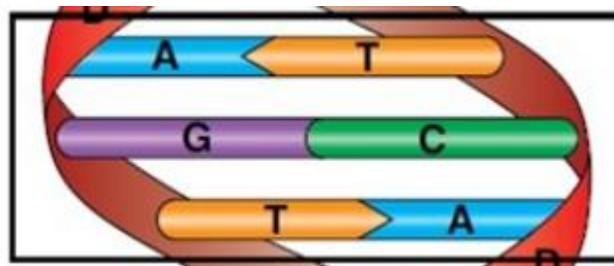
Bu yapıda DES algoritması çalışmaya başladıktan sonra ilk tur işlemini başlatmak için giriş permutasyonuna girer. Atmış dört bitlik açık metni permutasyon ile karıştırdıktan sonra ortadan ikiye bölerek sağdaki 32 bitlik veriyi F fonksiyonuna sokulur. Burada ilk olarak genişletme işlemine maruz bırakılır. Bu işlem 32 bit verinin boyutunu arttırıp 48 bit olacak şekilde gerçeklemesiyle oluşur. Bu gerçeklemeyi yaparken belirli bitler tekrar eder ve tekrar sonucunda 32 bit veri 48 bit veri olarak genişletilmiş olur.

Genişletilen veri bu aşamada DNA bazlarına dönüştürülür, bu dönüşüm esnasında her iki bit için 1 adet olmak üzere 24 adet DNA bazı üretilir. DNA şifrelemesi başlatılır ve ardından DNA bazları “Watson-Crick’s” diye isimlendirilen dönüşüme sokulur (Şekil 5.3). Bu dönüşüm DNA biyolojik yapısında bazların birbirleri ile kurduğu hidrojen bağını temsil etmektedir. Bu hidrojen bağlarına göre bazlar birbirlerine bağlanır.



Şekil 5.2. DNA bağları.

Yukarıda Şekil 5.2’de gösterildiği üzere A-T ve C-G şeklinde birbirlerine bağlanır. Bilgisayar dilinde bu yöntem kullanılarak 24 bitlik baz dizisi yeniden sıralanır. Bu sıralama sonucunda A-T ve C-G bazlarının yerleri değiştirildi.



Şekil 5.3. Watson Crick’s DNA bağları.

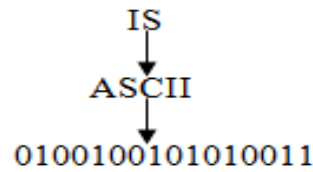
24 bitlik yeni baz dizilimi elde edilerek ilk karıştırma işlemi yapılır ve bazların yeri değiştirilir. Bununla beraber doğru orantılı olarak da yeni bit sırası oluşturuldu. Yeni sıralama 48 bit veriyi, algoritmada bulunan ve akış şemasında gösterilen Feistel yapısı kullanılarak bit sırası değiştirildi. Bu işlem sonucunda bitler değişiklik gösterir ve dört adım sonunda bit dizisi tekrar DNA dönüşüm işlemine tabii tutuldu. Ardından “Watson-Crick’s” dönüşümü uygulanarak işlem sonlandırıldı. Bu işlemin ardından algoritma rutin çalışmasına devam ederek işlemini bitirir ve bu işlem algoritmanın yaptığı 16 tekrar işleminin her adımında yapılır. Yapılan işlem sonucunda algoritma normalde bir adet “XOR” işlemi yaparken, yapılan modelleme sonucu F fonksiyonu içinde Feistel yapısını da içine alarak ekstra üç “XOR” işlemi daha yapılarak, algoritmayı veri erişimi konusunda daha karmaşık bir yapıya sokmaktadır.

DES algoritması için önerilen adımları daha anlaşılır bir şekilde göstermek amacıyla aşağıda örnek üzerinde işlemler gösterildi. Bunun yanı sıra veri boyutu değişkenlik gösterse bile mantık olarak algoritmanın simetrik yapısının bozulmadığını göstermek için, aşağıda 16 bitlik bir veri için önerilen yöntem adım adım anlatıldı.

Şekil 5.1’deki akış diyagramı için şifreleme işlemi adımlar halinde açıklanmıştır.

#### 1. Adım: Düz metnin ikili dönüşümünün yapılması;

Şifrelenecek düz metin ilk adımda Şekil 5.4’de gösterildiği gibi ASCII değerine dönüştürülüp ardından ikili (Binary) tabana dönüştürülür.



Şekil 5.4. Şifrelenecek veri.

#### 2. Adım: DNA dönüşüm işlemlerinin yapılması;

İkili tabana dönüştürülen metin burada DNA bazlarına çevrilerek 8 adet DNA bazı elde edilir. Ardından DNA bazları “Watson Crick’s” tamamlayıcı tablosu kullanılarak yeni bir baz dizilimi elde edilip bit sırası karıştırılmış olur.

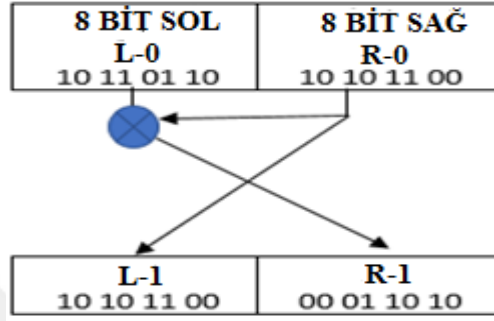
01 00 10 01 01 01 00 11 = C-A-G-C-C-C-A-T

C-A-G-C-C-C-A-T = G-T-C-G-G-G-T-A (Watson Crick’s)

G-T-C-G-G-G-T-A = 10 11 01 10 10 10 11 00

### 3. Adım: Şifreleme yapılması;

Şifreleme yapmak için karıştırılan veri Şekil 5.5’deki Feistel yapısı kullanılarak işleme sokulur. Feistel yapısı ile bit sırasını değiştirmek amacıyla ilk olarak sağdaki bitler soldakiler ile yer değiştirir ve sağdaki bit bloğu yeni sol blok olur. İkinci adım olarak sağ ve sol kısımlardaki bitler “XOR” işlemine sokularak yeni sağ blok oluşturulur ve Şekil 5.5’de görüldüğü üzere sıralamanın sağ kısmına eklenerek işlem sonlandırılır.

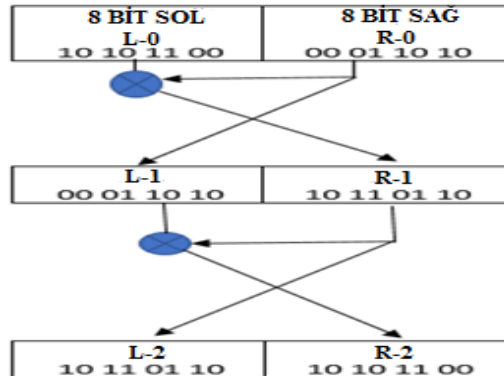


Şekil 5.5. Şifreleme işlemi.

Şekil 5.1’deki akış diyagramını için şifre çözme işlemi adımlar halinde açıklanmıştır

### 1. Adım: Şifre çözme işlemlerinin yapılması;

Şifreli metni geri dönüştürebilmek için şifreli mesaj çıktısı Şekil 5.6’daki Feistel yapısı kullanılarak aşağıdaki işlemler uygulandı. İşlemin ilk adımında f fonksiyonu içine yerleştirilmiş olan şifre çözme adımları çalışmaya başlar. 8 bitlik bloklardan sağ blok çaprazlanarak yeni sol blok olur. İşlemin ikinci adımında ise sağ ve sol bit blokları “XOR” işlemi yapıp metnin sağ tarafına yerleştirilir ve 16 bitlik yeni bir bit sırası elde edilir. Oluşan yeni sıralama yukarıda bahsedilen ilk iki adımı ikinci kez tekrarlayarak çalışmasını sonlandırır.



Şekil 5.6. Şifre çözme işlemi.

2. Adım: DNA dönüşüm işlemlerinin yapılması;

Elde edilen ikili veriler öncelikle DNA formuna dönüştürüldü. Ardından elde edilen 8 DNA bazı Şekil 5.3’de gösterilen “Watson Crick’s” tamamlayıcı tablosu kullanılarak yeni bir baz sıralaması oluşturuldu.

10 11 01 10 10 10 11 00 = G-T-C-G-G-G-T-A

G-T-C-G-G-G-T-A = C-A-G-C-C-C-A-T (Watson Crick’s)

Watson Crick’s modeli kullanılarak DNA sıralaması son halini alır.

3. Adım: Düz metnin elde edilmesi;

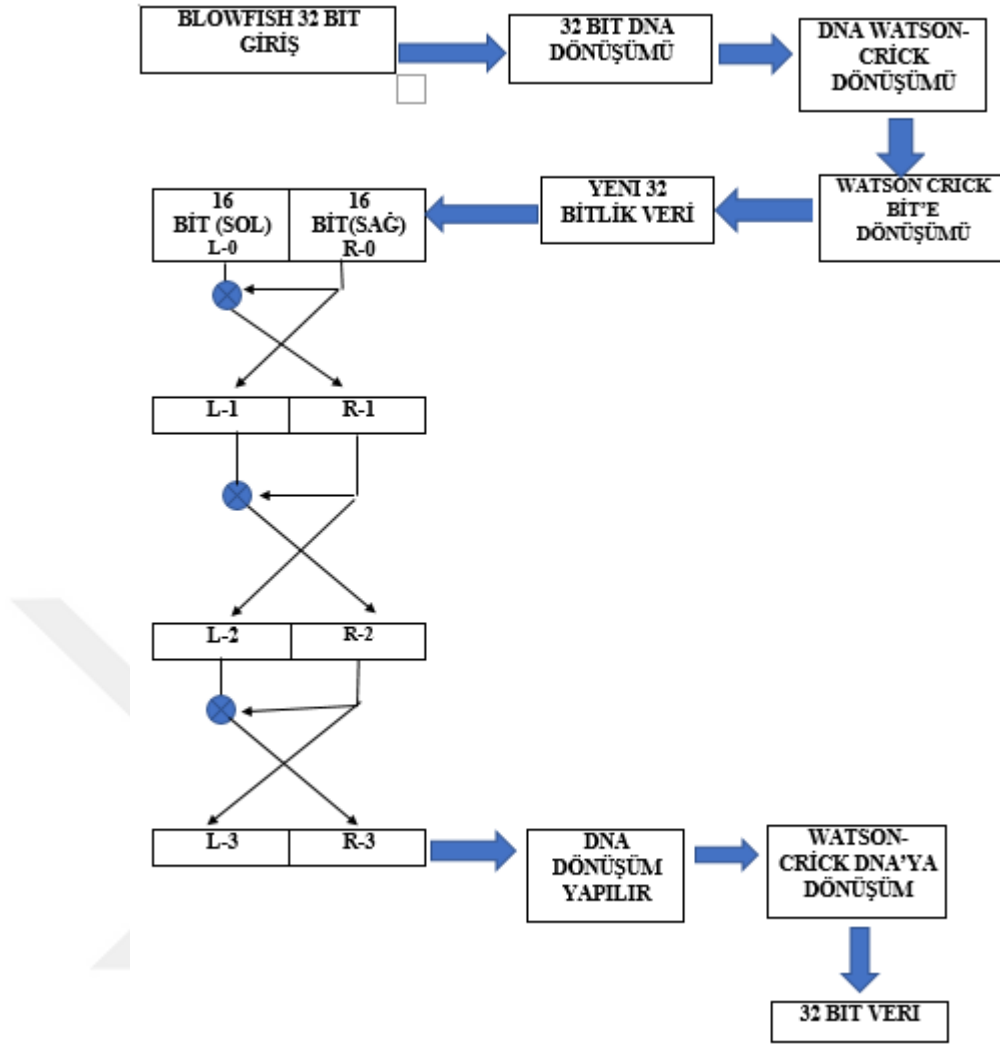
Son adım olarak da tamamlayıcı tablo (Watson Crick’s) temel alınarak yeni bir dizilime sahip olan DNA bazları tekrar ikili tabana dönüştürüldüğünde şifre çözme işlemi son bulur.

C-A-G-C-C-C-A-T = 01 00 10 01 01 01 00 11 = IS

Algoritmanın F fonksiyonu içine yerleştirilmiş olan DNA ve Feistel yapısını içeren bu adımlar sayesinde algoritmaya ek bir karmaşıklık sağlandı. DNA modelini etkin bir şekilde uyguladıktan sonra, DES algoritması üzerindeki performans (süre, Ram, CPU) değişimlerini analiz etmek amaçlanmıştır. Sonuçlar tablo ve grafikler yardımıyla ortaya koyuldu.

### 5.3. BLOWFISH ALGORİTMASI İÇİN ÖNERİLEN DNA MODELİ

Blowfish algoritmasında da DES algoritması gibi bir F fonksiyonu vardır. F fonksiyonu ve DES içinde bulunan fonksiyon aynı amaca hizmet etmektedirler. İkisi de algoritmanın karmaşıklık işlemlerinin yapıldığı bölümdür, ayrıca yapısal olarak da benzerlikleri bulunmaktadır. Tamamen farklı fonksiyonlar değillerdir, özünde aynı mantık ile çalışmaktadır. Bundan dolayı DES algoritmasının modellenmesinde kullandığımız akış şemasını Blowfish algoritması içinde kullanılabilir olduğu söylenilebilir. Blowfish algoritması için önerilen akış şeması Şekil 5.7’de gösterilmiştir.



Şekil 5.7. Blowfish için önerilen akış şeması.

Blowfish algoritması, DES algoritması gibi bir F fonksiyonuna sahiptir, bu fonksiyon temelde aynı mantık içerse bile birbirinden farklıdır. En basit şekilde anlatmak gerekirse Blowfish algoritması DES gibi bir genişletme tablosu kullanmaz, yine 64 bit veri ortadan ikiye iki eşit parçaya bölünür ve 32 bit şeklinde F fonksiyonuna giriş sağlar. Giriş ve çıkış arası işlemler yapılırken algoritma sürekli 32 bit veri kullanır, herhangi bir artış ya da düşüş olmaz. Her adımın sonunda 32 bit veri elde edilir bundan dolayı Blowfish üzerinde uyguladığımız akış şeması 32 bit veri ile başlayıp sonucunda 32 bit veri ile işlemi bitirir. Bundan kaynaklı olarak Blowfish algoritması DES algoritmasıyla karşılaştırıldığında 16 adet DNA bazı ile işlem yapmaktadır.

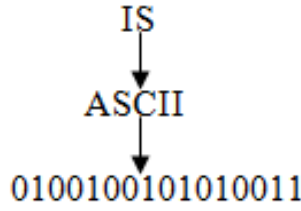
Şekil 5.7'deki akış diyagramı için şifreleme işlemi adımlar halinde açıklanmıştır.

1. Adım: Düz metnin ikili dönüşümünün yapılması;

Şifrelenecek düz metin ilk adımda Şekil 5.8'de gösterilen ASCII değerine



dönüştürölüp ardından ikili (Binary) tabana çevrilir.



Şekil 5.8. Şifrelenecek veri.

2. Adım: DNA dönüşüm işlemlerinin yapılması;

İkili tabana dönüştürülmüş metin burada DNA bazlarına dönüştürülerek 8 adet DNA bazı elde edilir. Ardından DNA bazları Şekil 5.3’de gösterilen “Watson Crick’s” tamamlayıcı tablosu kullanılarak yeni bir baz dizilimi elde edilip bit sırası karıştırılmış olur.

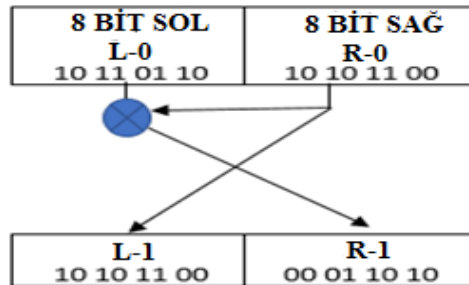
01 00 10 01 01 01 00 11 = C-A-G-C-C-C-A-T

C-A-G-C-C-C-A-T = G-T-C-G-G-G-T-A (Watson Crick’s)

G-T-C-G-G-G-T-A = 10 11 01 10 10 10 11 00

3. Adım: Şifreleme yapılması;

Şifreleme yapmak için karıştırılmış veri Şekil 5.9’daki Feistel yapısı kullanılarak işleme sokulur. Feistel yapısı ile bit sırasını değiştirmek amacıyla ilk olarak sağdaki bitler soldakiler ile yer değiştirir. İkinci adım olarak sağ ve sol kısımlardaki bitler “XOR” işlemine sokularak yeni bit sırası oluşturulur ve Şekil 5.9’da görüldüğü üzere sıralamanın sağ kısmına eklenerek işlem sonlandırılır.

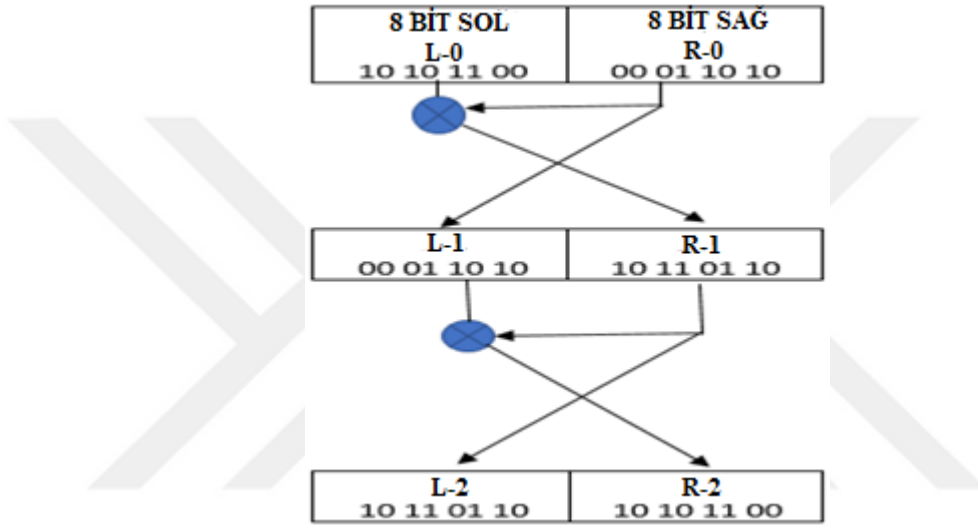


Şekil 5.9. Düz metnin şifrlenmesi.

Şekil 5.7’deki akış diyagramı için şifre çözme işlemi aşağıda adımlar halinde açıklandı.

1. Adım: Şifre çözme işlemlerinin yapılması;

Şifreli metni geri dönüştürebilmek için şifreli mesaj çıktısı Şekil 5.10’da gösterilen Feistel yapısı kullanılarak işlemler uygulanır. İşlemin ilk adımında f fonksiyonu içine yerleştirilmiş olan şifre çözme adımları çalışmaya başlar. 8 bitlik bloklardan sağ blok çaprazlanarak yeni sol blok olur. İşlemin ikinci adımında ise sağ ve sol bit blokları “XOR” işlemi yapıp metnin sağ tarafına yerleştirilir ve 16 bitlik yeni bir bit sırası elde edilir. Oluşan yeni sıralama yukarıda bahsedilen ilk iki adımı ikinci kez tekrarlayarak çalışmasını sonlandırır.



Şekil 5.10. Şifre çözümü.

2. Adım: DNA dönüşüm işlemlerinin yapılması;

Elde edilen ikili veriler öncelikle DNA formuna dönüştürülür ardından elde edilen 8 DNA bazı “Watson Crick’s” tamamlayıcı tablosu kullanılarak yeni bir baz sıralaması oluşturuldu.

10 11 01 10 10 10 11 00 = G-T-C-G-G-G-T-A

G-T-C-G-G-G-T-A = C-A-G-C-C-C-A-T (Watson Crick’s)

Watson Crick’s modeli kullanılarak DNA sıralaması son halini alır.

3. Adım: Düz metnin elde edilmesi;

Son adım olarak da tamamlayıcı tablo temel alınarak yeni bir dizilime sahip olan DNA bazları tekrar ikili tabana dönüştürüldüğünde şifre çözme işlemi son bulmaktadır.

C-A-G-C-C-C-A-T = 01 00 10 01 01 01 00 11

01 00 10 01 01 01 00 11 = IS

Algoritmanın F fonksiyonu içine yerleştirilen DNA ve Feistel yapısını içeren bu adımlar sayesinde algoritmaya ek bir karmaşıklık sağlandı. DNA modelini etkin bir şekilde uyguladıktan sonra Blowfish algoritması üzerindeki performans (Süre, RAM, CPU) değişimlerini analiz etmek amaçlandı ve sonuçlar tablo ve grafikler yardımıyla ortaya koyuldu.

#### 5.4. DES ALGORİTMASININ DNA İLE PERFORMANS ANALİZİ

Bu bölümde DES algoritmasının üç farklı açıdan performans değerlendirilmesi yapıldı, bu üç metot hız, hafıza karmaşıklığı ve CPU kullanımı üzerine değerlendirildi. Performans analizleri yapılırken kullanılan en yaygın metotlar bunlardır. Algoritmaların çalışma hızları bilgisayarın barındırdığı donanıma göre değişiklik göstereceği için bu çalışmada donanım olarak iki farklı bilgisayar kullanıldı ve bu iki farklı bilgisayara göre sonuçların değerlendirilmesi yapıldı.

##### 5.4.1. Performans Test Sonuçları

Şifreleme algoritmalarının performans analizleri farklı donanım özellikleri ve farklı işletim özelliklerine sahip bilgisayarlarda hesaplandı. Bilgisayar donanımları Çizelge 5.1’de gösterildi. Bu şifreleme işlemleri farklı boyutlardaki verilerin 200 defa şifrelenmesi ve şifrelerin çözülmesi ile elde edildi. Elde edilen sonuçlar 200 iterasyonun ortalama sonuçları şeklinde verildi.

Çizelge 5.1. Bilgisayar donanımları.

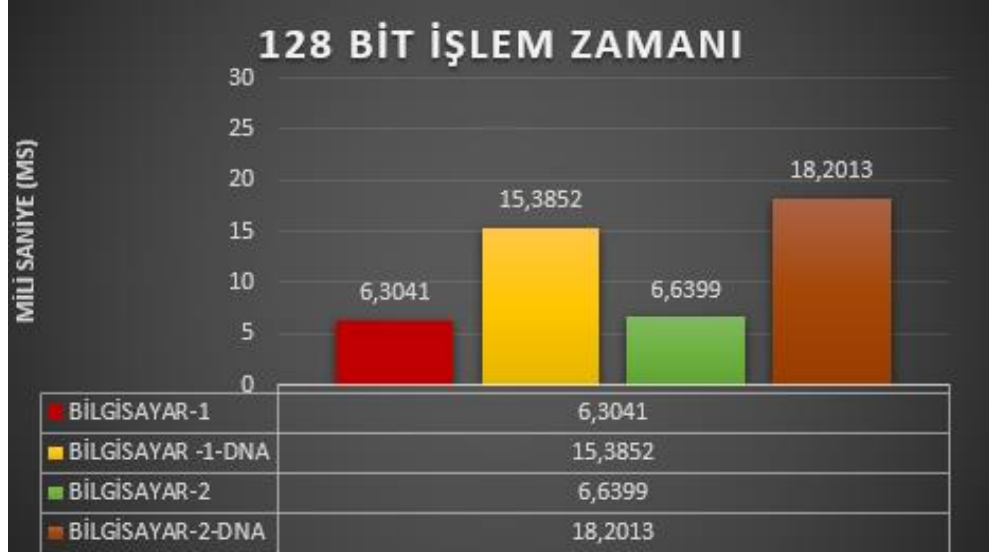
|                 | Bilgisayar-1                           | Bilgisayar-2                            |
|-----------------|--|---|
| CPU             | Intel® Core™ i5-3210M<br>CPU @ 2.50GHz | Intel® Core™ i7-4712MQ<br>CPU @ 2.30GHz |
| RAM             | 8 GB                                   | 12 GB                                   |
| İşletim Sistemi | Microsoft Windows 8.1                  | Microsoft Windows 10                    |

#### 5.4.1.1. DES Algoritmasının Şifreleme ve Şifre Çözme Sonuçları

128-256-512-1024-2048-4096-8192 Bit veri şifrelemesi yapılırken DNA modellenmesi ve ek adımlar yerleştirildi, Şifreleme metinleri katlanarak artacak şekilde gösterilerek, sonuçlar arasındaki ilişki bağlamı daha net belirtilmek istendi ve bu amaç doğrultusunda grafikler ile sonuçlar desteklendi. Grafikler tasarlanırken Bilgisayar-1 için kırmızı ve sarı renkler, Bilgisayar-2 için yeşil ve kahverengi olmak üzere her bilgisayar için iki renk kullanıldı. Kırmızı renk Bilgisayar-1 de yapılan orijinal şifreleme ve şifre çözme performans değerlerinin toplamını, sarı renk ise DNA tabanlı şifreleme ve şifre çözme performans değerlerinin toplamını göstermektedir. Bilgisayar-2 için yeşil renk algoritmanın orijinal halinin şifreleme ve şifre çözme performans değerlerini göstermektedir. Kahverengi ile gösterilen performans değerleri ise Bilgisayar-2’de yapılan DNA tabanlı şifreleme ve şifre çözme sonuçlarının toplamını göstermektedir. Algoritmaların işlem performans değerleri, şifreleme ve şifre çözme işlemlerinin toplamı alınarak gösterildi. Çizelge 5.2’de gösterilen değerler zaman açısından saniye, işlemci açısından kullanılan değerler yüzde (%) şeklinde ve hafıza kullanımı için gösterilen değerler megabayt (MB) türünden gösterildi.

Çizelge 5.2. 128 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 15,38800  | 26,99450  | 7,421000  | 16,01449  |
| RAM<br>Kullanımı<br>(MB) | 0,000506  | 0,000457  | 0,000314  | 0,000609  |
| İşlem Zamanı<br>(S)      | 0,006304  | 0,015385  | 0,006639  | 0,018201  |



Şekil 5.11. 128 Bit işlem zaman grafiği.

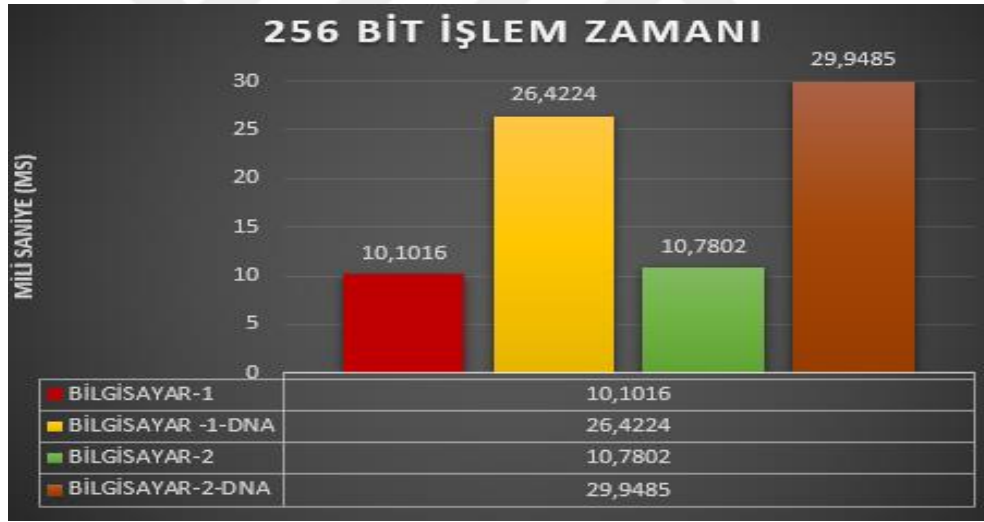
Şifreleme sonuçları Şekil 5.11’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan ve en küçük şifreleme boyutumuz olan 128 bit şifreleme sonucunda DES algoritmasının orijinal durumlardaki şifreleme ve şifre çözme süreleri iki bilgisayarda da birbirine çok yakın olduğu açıkça görüldü. Bununla birlikte DNA modeli uygulanmış DES algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %20 oranında donanımdan kaynaklı bir mutlak fark ortaya gözlemlendi.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.2’de gösterilen CPU kullanım değerleri arasındaki farktır, DES algoritmasının orijinal CPU kullanımı iki bilgisayar için oransal olarak  $\frac{1}{2}$  düzeyinde bir fark oluşturmuştur. Bu farkın sebebi veri boyutunun küçük olması ve bunun akabinde donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.2 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar olduğu, gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Çizelge 5.3. 256 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 22,538000   | 26,994500   | 12,162499   | 14,937500   |
| RAM<br>Kullanımı<br>(MB) | 0,0004730   | 0,0004774   | 0,0007494   | 0,0006498   |
| İşlem<br>Zamanı (S)      | 0,0101016   | 0,0264224   | 0,0107802   | 0,0299485   |



Şekil 5.12. 256 Bit işlem zaman grafiği.

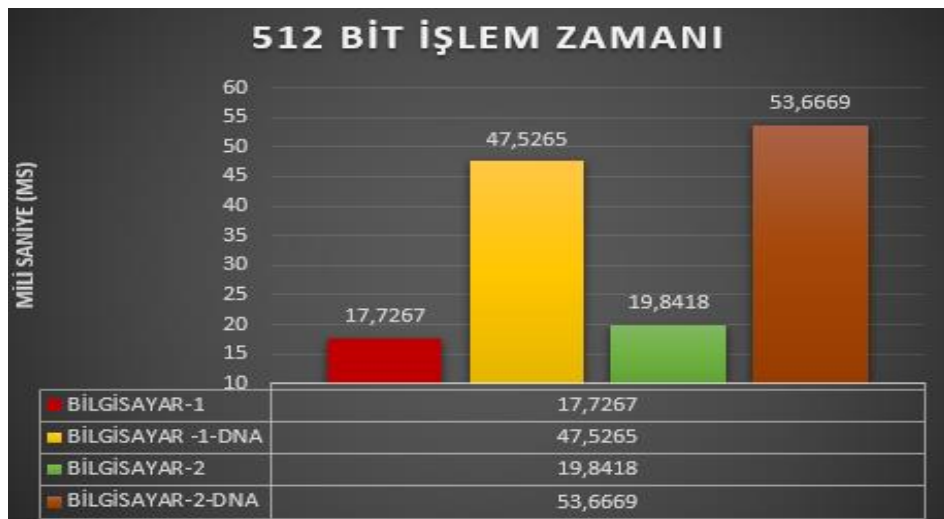
Şifreleme sonuçları Şekil 5.12’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 256 bit şifreleme sonucunda, DES algoritmasının orijinal durumlardaki, şifreleme ve şifre çözme süreleri birbirine çok yakın olduğu açıkça görüldü. Bununla birlikte DNA modeli uygulanmış DES algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %12 oranında donanımdan kaynaklı bir mutlak fark oluşturduğu gözlemlendi.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.3’de gösterilen CPU kullanım değerleri arasındaki farktır. DES algoritmasının orijinal hali ve DNA modeli uygulanması sonucunda CPU kullanımı, iki bilgisayar için yaklaşık olarak %45 bir fark oluşturduğu görüldü. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.3 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar olduğu, gerçekçi ortamlarda fark edilmeyecek düzeyde küçük farklar gözlemlendi.

Çizelge 5.4. 512 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 26,552000   | 25,764500   | 15,164999   | 14,997000   |
| RAM<br>Kullanımı<br>(MB) | 0,0004361   | 0,0004968   | 0,0006910   | 0,0007522   |
| İşlem<br>Zamanı (S)      | 0,0177267   | 0,0475265   | 0,0198418   | 0,0536669   |



Şekil 5.13. 512 Bit işlem zaman grafiği.

Şifreleme sonuçları Şekil 5.13’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 512 bit şifreleme sonucunda DES algoritmasının orijinal durumlardaki, şifreleme ve şifre çözme süreleri açısından yaklaşık %10 dan biraz daha az bir fark görüldü. Bununla birlikte DNA modeli uygulanmış DES algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %12 oranında donanımdan kaynaklı bir mutlak fark oluşturduğu gözlemlendi.

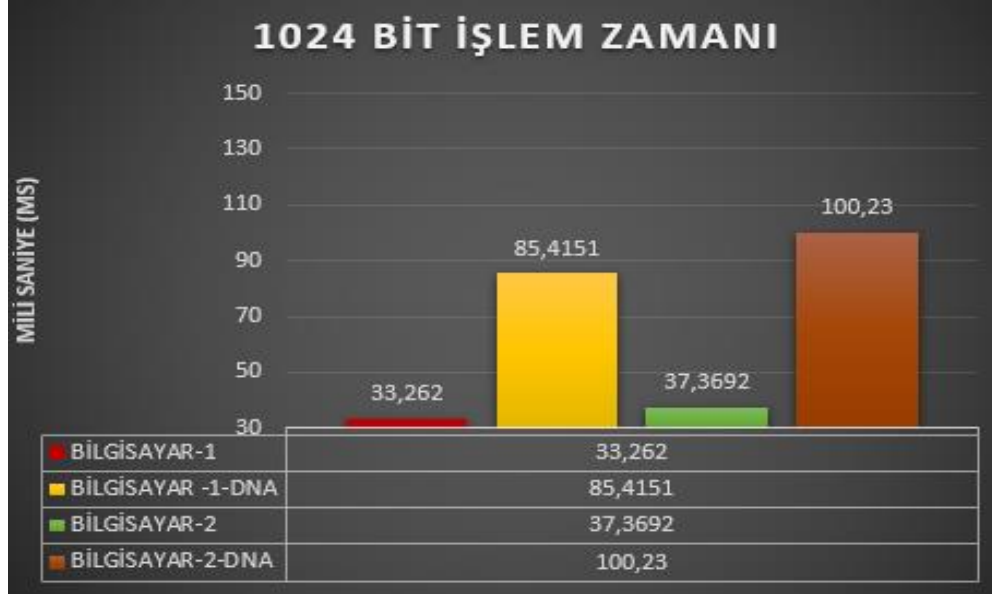
Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.4’de gösterilen CPU kullanım değerleri arasındaki farktır. DES algoritmasının orijinal ve DNA modeli uygulanması sonucunda CPU kullanımı iki bilgisayar için yaklaşık olarak %40 bir fark oluşturmuştur. Bu farkın sebebi veri boyutunun küçük olması bunun akabinde donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.4 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar, gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Çizelge 5.5. 1024 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 26,307499   | 25,212500   | 13,4834999  | 13,519500   |
| RAM<br>Kullanımı<br>(MB) | 0,0005587   | 0,0005576   | 0,0007712   | 0,0007917   |
| İşlem<br>Zamanı (S)      | 0,0332620   | 0,0854151   | 0,0373692   | 0,1002305   |





Şekil 5.14. 1024 Bit işlem zaman grafiği.

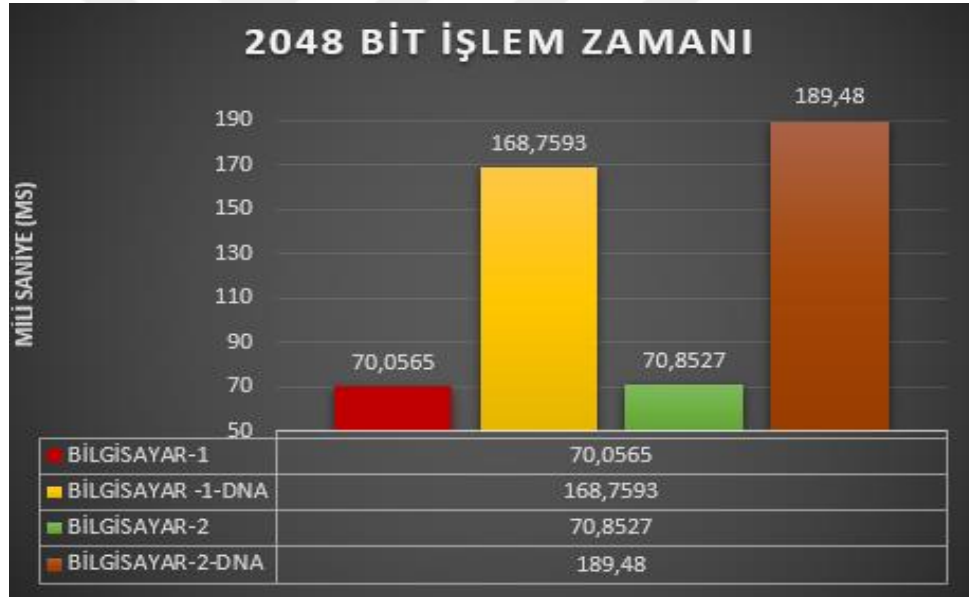
Şifreleme sonuçları Şekil 5.14’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 1024 bit şifreleme sonucunda, DES algoritmasının orijinal durumlardaki, şifreleme ve şifre çözme süreleri yaklaşık %12 den biraz daha az bir fark görüldü. Bununla birlikte DNA modeli uygulandı. DES algoritmasının işlem süresi, iki bilgisayar için yaklaşık olarak %18 oranında donanımdan kaynaklı bir mutlak fark oluşturduğu gözlemlendi.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.5’de gösterilen CPU kullanım değerleri arasındaki farktır. DES algoritmasının orijinal ve DNA modeli uygulanması sonucunda, CPU kullanımı iki bilgisayar için yaklaşık olarak %48 bir fark oluşturmuştur. Bu farkın sebebi, donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.5 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar, gerçekçi ortamlarda fark edilmeyecek düzeyde küçük farklar gözlemlendi.

Çizelge 5.6. 2048 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 33,117999   | 25,811500   | 15,143499   | 13,804500   |
| RAM<br>Kullanımı<br>(MB) | 0,0007294   | 0,0010907   | 0,0007524   | 0,0009763   |
| İşlem<br>Zamanı (S)      | 0.0700565   | 0,1687593   | 0,0708527   | 0,1894808   |



Şekil 5.15. 2048 Bit işlem zaman grafiği.

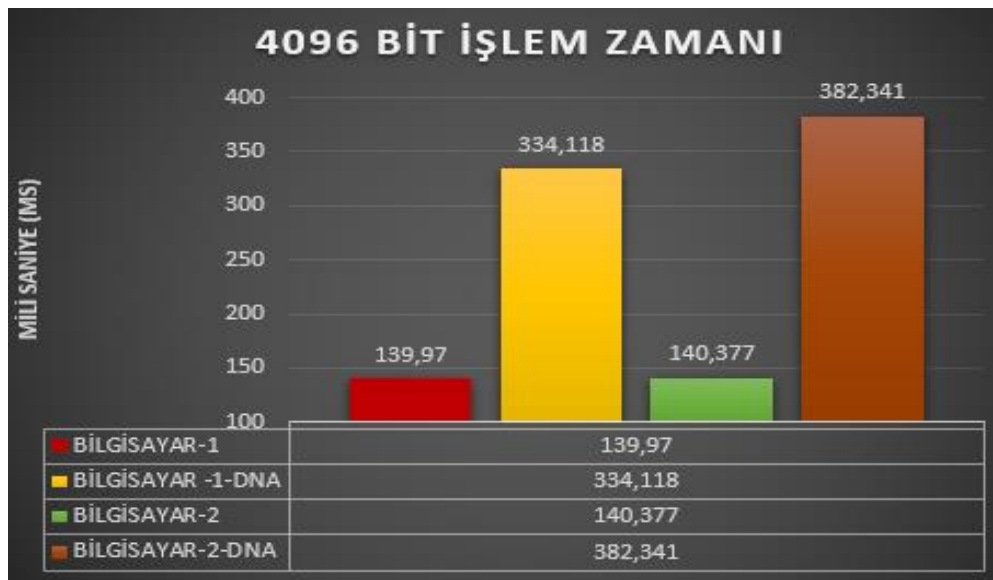
Şifreleme sonuçları Şekil 5.15’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 2048 bit şifreleme sonucunda; DES algoritmasının orijinal durumlarında, şifreleme ve şifre çözme süreleri çok minimal düzeyde etkilendiği saptandı. Bununla birlikte DNA modeli uygulandı. DES algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %11 oranında donanımdan kaynaklı bir mutlak fark gözlemlendi.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.6’da gösterilen CPU kullanım değerleri arasındaki farktır. DES algoritmasının orijinal hali ve DNA modeli uygulanması sonucunda, CPU kullanımı iki bilgisayar için yaklaşık olarak %50 bir fark oluşturmuştur. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.6 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar, gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Çizelge 5.7. 4096 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı (%)     | 35,998500   | 26,656999   | 13,147500   | 12,911500   |
| RAM<br>Kullanımı<br>(MB) | 0,0013876   | 0,0013783   | 0,0011395   | 0,0013031   |
| İşlem Zamanı<br>(S)      | 0,1399708   | 0,3341185   | 0,1403775   | 0,3823414   |



Şekil 5.16. 4096 Bit işlem zaman grafiği.

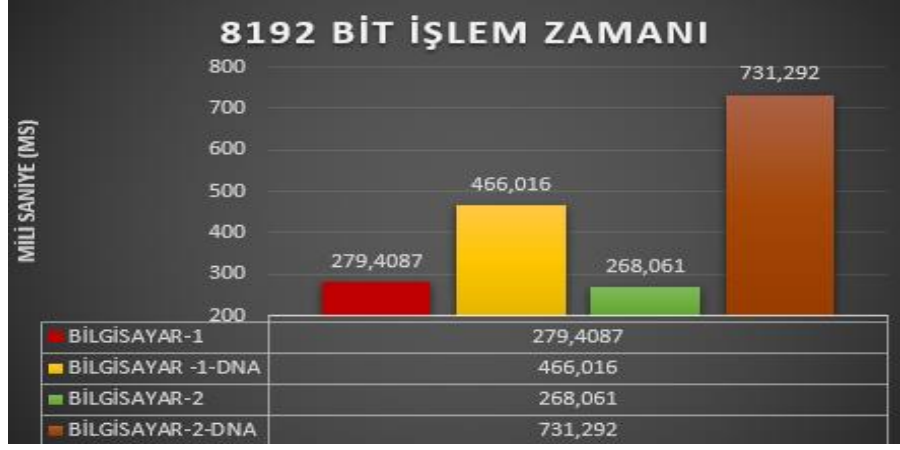
Şifreleme sonuçları Şekil 5.16’da gösterildiği üzere süre açısından incelenip grafiklerle görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 4096 bit şifreleme sonucunda DES algoritmasının orijinal durumlarındaki şifreleme ve şifre çözme süreleri çok minimal fark etmiştir. DNA modeli uygulanan. DES algoritmasının işlem süresi iki bilgisayar arasında yaklaşık olarak %14 oranında donanımdan kaynaklı bir mutlak fark oluşturduğu gözlemlendi.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.7’de gösterilen CPU kullanım değerleri arasındaki farktır. DES algoritmasının orijinal ve DNA modeli uygulanması sonucunda, CPU kullanımı iki bilgisayar için yaklaşık olarak %60 bir fark oluşturdu. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.7 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar olduğu ve gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Çizelge 5.8. 8192 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 40,779500   | 25,400500   | 13,286000   | 13,474500   |
| RAM<br>Kullanımı<br>(MB) | 0,0025598   | 0,0010698   | 0,0027371   | 0,0037817   |
| İşlem<br>Zamanı (S)      | 0,2794087   | 0,4660163   | 0,2680619   | 0,731292  |



Şekil 5.17. 8192 Bit işlem zaman grafiği.

Şifreleme sonuçları Şekil 5.17’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 8192 bit şifreleme sonucunda, DES algoritmasının orijinal durumlarında şifreleme ve şifre çözme süreleri arasında çok minimal farklar görüldü. Bununla birlikte DNA modeli uygulanmış DES algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %58 oranında donanımdan kaynaklı bir mutlak fark tespit edildi.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.8’de gösterilen CPU kullanım değerleri arasındaki farktır. DES algoritmasının orijinal hali ve DNA modeli uygulanması sonucunda, CPU kullanımı iki bilgisayar arasında yaklaşık olarak %60 bir fark oluşturduğu gözlemlendi. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.8 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar olduğu, gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Genel olarak yapılan tüm şifreleme ve şifre çözme işlemlerinin performans kriterlerini kıyaslırsak bilgisayar-1’de çalışan DES algoritması ve DNA ile modellenmiş DES algoritması süre açısından bilgisayar-2’ye göre daha iyi performans göstermiştir.

Bilgisayar-2 DES algoritması ve DNA ile modellenmiş DES algoritmasını çalıştırırken ise bariz olarak CPU kullanımı olarak bilgisayar-1’e üstünlük sağladığı görüldü. RAM kullanımı açısından her iki bilgisayarda çalışırken çok minimal farklar görüldü ve birbirlerine karşı yüksek farklar görülmedi.

## 5.5. BLOWFISH ALGORİTMASININ DNA İLE PERFORMANS ANALİZİ

Bu bölümde Blowfish algoritmasının üç farklı açıdan performans değerlendirilmesi yapıldı. Bu üç metot hız, hafıza karmaşıklığı ve CPU kullanımı üzerine değerlendirildi. Performans analizleri yapılırken kullanılan en yaygın kullanılan metotlar bunlardır.

Algoritmaların performansları bilgisayarın barındırdığı donanımına göre değişiklik göstereceği için bu çalışmada donanım olarak Çizelge 5.9’da gösterilen iki farklı bilgisayar kullanılıp ve bu iki farklı bilgisayara göre sonuçlar değerlendirildi.

### 5.5.1. Performans Test Sonuçları

Şifrelenecek düz metinleri katlanarak artacak şekilde gösterilerek sonuçlar arasındaki ilişki bağlamı daha net belirtilmek istendi ve bu amaç doğrultusunda grafikler ile sonuçlar desteklendi. Grafikler tasarlanırken Bilgisayar-1 için kırmızı ve sarı renkler, Bilgisayar-2 için yeşil ve kahverengi olmak üzere her bilgisayar için iki renk kullanıldı. Kırmızı renk Bilgisayar-1 de yapılan orijinal şifreleme ve şifre çözme performans değerlerini, sarı renk ise DNA tabanlı şifreleme ve şifre çözme performans değerlerini göstermektedir. Bilgisayar-2 için yeşil renk algoritmanın orijinal halinin şifreleme ve şifre çözme performans değerlerini göstermektedir. Kahverengi ile gösterilen performans değerleri ise Bilgisayar-2’de yapılan DNA tabanlı şifreleme ve şifre çözme sonuçlarını göstermektedir. Algoritmaların işlem performans değerleri, şifreleme ve şifre çözme işlemlerinin toplamı alınarak gösterildi. Çizelgelerdeki değerler algoritmanın DNA modellemesi ve orijinal hali arasındaki performans farkını göstermek amacıyla tasarlandı. Çizelgelerde gösterilen değerler zaman açısından saniye, işlemci açısından kullanılan değerler yüzde (%) şeklinde hafıza kullanımı için gösterilen değerler megabayt (MB) türünden gösterildi. Blowfish algoritmasının performans analizleri, farklı donanım özellikleri ve farklı işletim sistemi özelliklerine sahip bilgisayarlarda hesaplandı.

Çizelge 5.9. Bilgisayar donanımları.

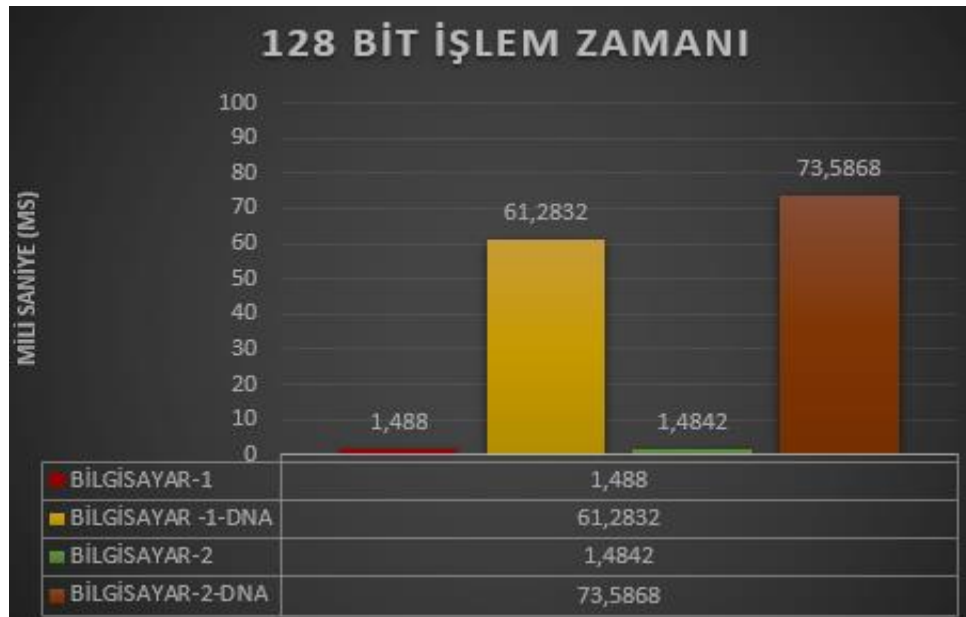
|                 | Bilgisayar-1                           | Bilgisayar-2                            |
|-----------------|--|---|
| CPU             | Intel® Core™ i5-3210M<br>CPU @ 2.50GHz | Intel® Core™ i7-4712MQ<br>CPU @ 2.30GHz |
| RAM             | 8 GB                                   | 12 GB                                   |
| İşletim Sistemi | Microsoft Windows 8.1                  | Microsoft Windows 10                    |

#### 5.5.1.1. Blowfish Algoritmasının Şifreleme ve Şifre Çözme Sonuçları

Bu bölümde Blowfish algoritması için yapılan şifreleme işlemleri, farklı boyutlardaki verilerin 200 defa şifrenmesi ve şifrelerin çözülmesi ile elde edildi. Elde edilen sonuçlar donanım özellikleri farklılık gösteren iki bilgisayarın 200 iterasyonunun ortalama sonuçları şeklinde verildi.

Çizelge 5.10. 128 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 5,1660000   | 25,499499   | 1,4360000   | 12,846000   |
| RAM<br>Kullanımı<br>(MB) | 0,0003328   | 0,0003736   | 0,0005748   | 0,0002993   |
| İşlem<br>Zamanı<br>(S)   | 0,0014880   | 0,0612832   | 0,0014842   | 0,0735868   |



Şekil 5.18. 128 Bit işlem zaman grafiği.

Şifreleme sonuçları Şekil 5.18’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan en küçük şifreleme boyutumuz olan 128 bit şifreleme sonucunda Blowfish algoritmasının orijinal durumlardaki, şifreleme ve şifre çözme süreleri birbirine çok yakın olduğu açıkça görüldü. DNA tabanlı şifreleme uygulanmış Blowfish algoritmasının işlem süresi, iki bilgisayar için yaklaşık olarak %20 oranında donanımdan kaynaklı bir mutlak fark oluşturduğu saptandı. Algoritmanın süre açısından DNA şifrelemesi sonucunda çok büyük performans kayıpları görüldü.

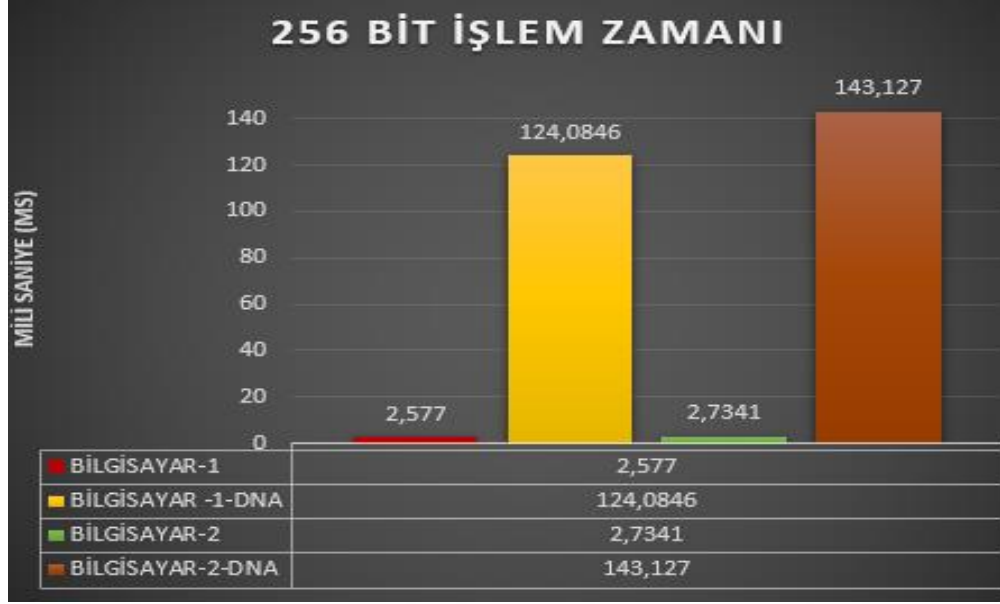
Bu donanım farkının ortaya çıkardığı diğer bir fark ise, Çizelge 5.10’da gösterilen CPU kullanım değerleri arasındaki farktır. Blowfish algoritmasının orijinal halinin CPU kullanımı, Bilgisayar-2 için daha verimli olduğu gözlemlendi, DNA modellemesi sonucu bu farkın 2 kat olduğu görüldü. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip Bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.10 incelendiğinde, iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar olduğu, gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Çizelge 5.11. 256 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 7,7654999   | 25,202500   | 2,945000  | 12.978000   |
| RAM<br>Kullanımı<br>(MB) | 0,0003354   | 0,0003733   | 0,0006583   | 0.0003406   |
| İşlem<br>Zamanı (S)      | 0,0025770   | 0,1240846   | 0,0027341   | 0.1431277   |





Şekil 5.19. 256 Bit işlem zaman grafiği.

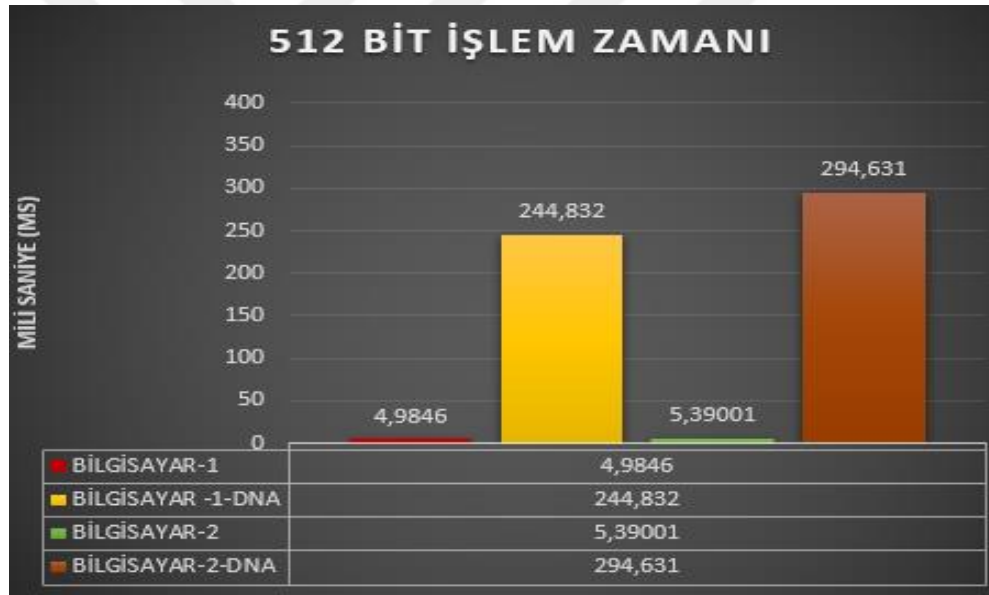
Şifreleme sonuçları Şekil 5.19’da gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 256 bit şifreleme sonucunda Blowfish algoritmasının orijinal durumlardaki, şifreleme ve şifre çözme süreleri birbirine çok yakın olduğu açıkça görüldü. DNA modeli uygulanmış Blowfish algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %16 oranında donanımdan kaynaklı bir mutlak fark oluşturdu. Algoritmanın süre açısından DNA modellemesi sonucunda çok büyük performans kayıpları görüldü.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.11’de gösterilen CPU kullanım değerleri arasındaki farktır, Blowfish algoritmasının orijinal CPU kullanımı yaklaşık 2,6 kat fark ile bilgisayar-2’nin daha performanslı olduğu görüldü. DNA modellemesi sonucu bu fark 2 kat oldu. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.11 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar olduğu, gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Çizelge 5.12. 512 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı (%)     | 10,430499   | 25,215000   | 9,1164999   | 12,692499   |
| RAM<br>Kullanımı<br>(MB) | 0,0003381   | 0,0005781   | 0,0006439   | 0,0005656   |
| İşlem Zamanı<br>(S)      | 0,0049846   | 0,2448324   | 0,0053900   | 0,2946314   |



Şekil 5.20. 512 Bit işlem zaman grafiği.

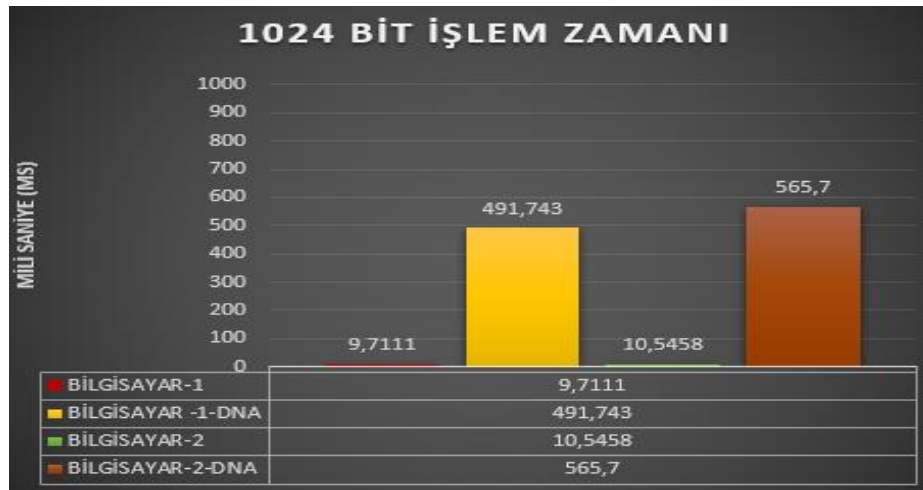
Şifreleme sonuçları Şekil 5.20’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 512 bit şifreleme sonucunda Blowfish algoritmasının orijinal durumlardaki, şifreleme ve şifre çözme süreleri birbirine çok yakın olduğu açıkça görüldü. DNA modeli uygulanmış Blowfish algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %20 oranında donanımdan kaynaklı bir mutlak fark oluşturdu. Algoritmanın süre açısından DNA modellemesi sonucunda çok büyük performans kayıpları saptandı.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.12’de gösterilen CPU kullanım değerleri arasındaki farktır. Blowfish algoritmasının orijinal CPU kullanımı çok küçük bir fark ile bilgisayar-2’nin daha performanslı olduğu tespit edildi. DNA modellemesi sonucu bu farkın 2 kat olduğu görüldü ve bilgisayar-2’nin yaklaşık 2 kat daha performanslı olduğu tespit edildi. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.12 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar olduğu, gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Çizelge 5.13. 1024 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 21,687000   | 25,180499   | 10,508999   | 12,744500   |
| RAM<br>Kullanımı<br>(MB) | 0,0003903   | 0,0004347   | 0,0006044   | 0,0004837   |
| İşlem<br>Zamanı (S)      | 0,0097111   | 0,4917431   | 0,0105458   | 0,5657002   |



Şekil 5.21. 1024 Bit işlem zaman grafiği.

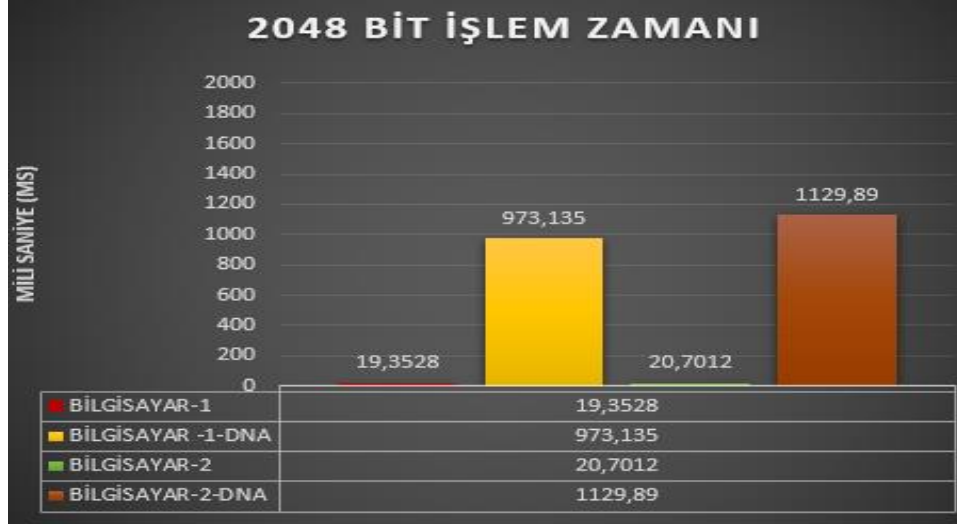
Şifreleme sonuçları Şekil 5.21’de gösterildiği üzere süre açısından incelenip grafiklerle görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 1024 bit şifreleme sonucunda, Blowfish algoritmasının orijinal durumlardaki, şifreleme ve şifre çözme süreleri birbirine çok yakın olduğu açıkça görüldü. DNA modeli uygulanmış Blowfish algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %15 oranında donanımdan kaynaklı bir mutlak fark oluşturduğu saptandı. Algoritmaların süre açısından DNA modellemesi sonucunda çok büyük performans kayıpları yaşandığı tespit edildi.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.13’de gösterilen CPU kullanım değerleri arasındaki farktır. Blowfish algoritmasının orijinal hali ve DNA tabanlı CPU kullanımı, yaklaşık 2 kat fark ile bilgisayar-2’nin daha performanslı olduğu tespit edildi. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.13 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar olduğu, gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Çizelge 5.14. 2048 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı (%)     | 27,399500   | 25,121500   | 14,768499   | 13,026499   |
| RAM<br>Kullanımı<br>(MB) | 0,0005189   | 0,0005985   | 0,0006496   | 0,0002383   |
| İşlem Zamanı<br>(S)      | 0,0193528   | 0,9731356   | 0,0207012   | 1,1298971   |



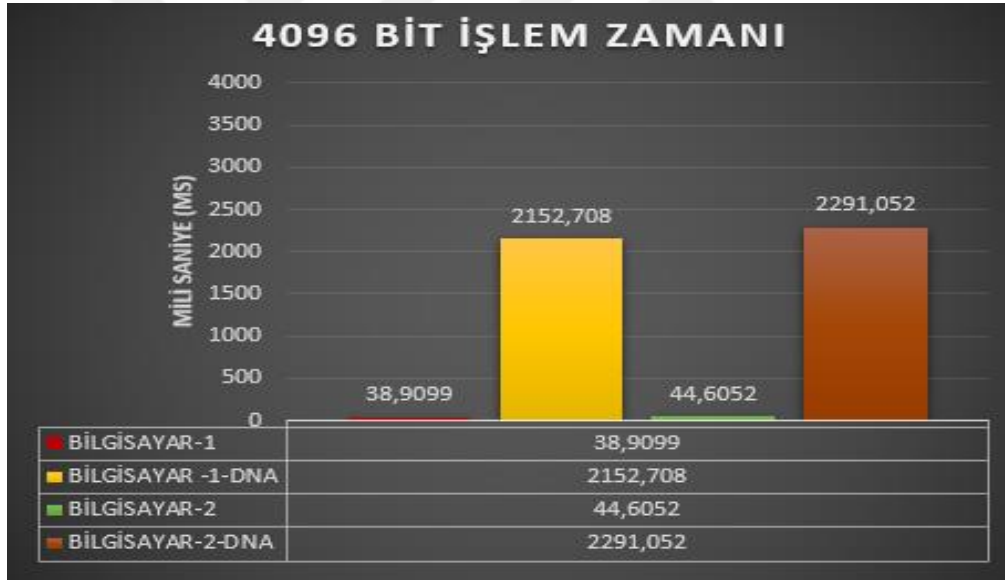
Şekil 5.22. 2048 Bit işlem zaman grafiği.

Şifreleme sonuçları Şekil 5.22’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 2048 bit şifreleme sonucunda, Blowfish algoritmasının orijinal durumlardaki, şifreleme ve şifre çözme süreleri birbirine çok yakın olduğu açıkça görüldü. Bununla birlikte DNA modeli uygulanmış Blowfish algoritmasının işlem süresi, iki bilgisayar için yaklaşık olarak %16 oranında donanımdan kaynaklı bir mutlak fark tespit edildi. Algoritmanın süre açısından DNA modellemesi sonucunda çok büyük performans kayıpları yaşandığı görüldü.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.14’de gösterilen CPU kullanım değerleri arasındaki farktır. Blowfish algoritmasının orijinal CPU kullanımı yaklaşık 1,9 kat fark ile bilgisayar-2’nin daha performanslı olduğu tespit edildi. DNA modellemesi sonucu bu fark yaklaşık 2 kat olduğu görüldü. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin, daha az CPU gereksiniminde bulunmasıdır. RAM açısından Çizelge 5.14 incelendiğinde iki bilgisayar arasında göz ardı edilebilecek çok küçük farklar olduğu gözlemlendi.

Çizelge 5.15. 4096 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 26,7895000  | 31,055000   | 14,210999   | 13,340499   |
| RAM<br>Kullanımı<br>(MB) | 0,0009688   | 0,0012393   | 0,0006081   | 0,0007915   |
| İşlem<br>Zamanı (S)      | 0,0389099   | 2,1527080   | 0,0446052   | 2,291052  |



Şekil 5.23. 4096 Bit işlem zaman grafiği.

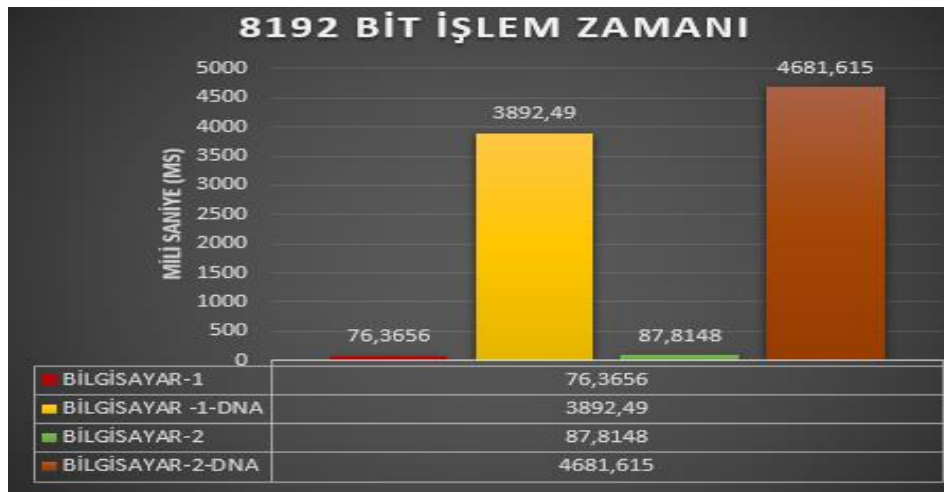
Şifreleme sonuçları Şekil 5.23’de gösterildiği üzere süre açısından incelenip grafikte görselleştirilmeye çalışıldı.. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 4096 bit şifreleme sonucunda Blowfish algoritmasının orijinal durumlardaki şifreleme ve şifre çözme süreleri %15 fark belirlendi. DNA modeli uygulanmış Blowfish algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %6 oranında donanımdan kaynaklı bir mutlak fark tespit edildi. Algoritmanın süre açısından DNA modellemesi sonucunda çok büyük performans kayıpları görüldü.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.15’de gösterilen CPU kullanım değerleri arasındaki farktır. Blowfish algoritmasının orijinal CPU kullanımı %45 fark ile bilgisayar-2 daha performanslı olduğu ve DNA modellemesi sonucu bu fark %55 olduğu belirlendi. Bu farkın sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.15 incelendiğinde iki bilgisayarında aralarında göz ardı edilebilecek çok küçük farklar olduğu ve gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

Çizelge 5.16. 8192 Bit performans değerleri.

|                          | Bilgisayar-1<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-1-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2<br>Orijinal<br>Şifreleme ve<br>Deşifreleme | Bilgisayar-2-<br>DNA Tabanlı<br>Şifreleme ve<br>Deşifreleme |
|--------------------------|---|---|---|---|
| CPU<br>Kullanımı<br>(%)  | 25,7820000  | 25,123500   | 13,325500   | 16,008500   |
| RAM<br>Kullanımı<br>(MB) | 0,0013569   | 0,0009672   | 0,0006277   | 0,0018387   |
| İşlem<br>Zamanı (S)      | 0,0763656   | 3,8924907   | 0,0878148   | 4,6816159   |



Şekil 5.24. 8192 Bit işlem zaman grafiği.

Şifreleme sonuçları Şekil 5.24’de gösterildiği üzere süre açısından incelenip grafiklerle görselleştirilmeye çalışıldı. İki farklı bilgisayarda yapılan şifreleme boyutumuz olan 8192 bit şifreleme sonucunda, Blowfish algoritmasının orijinal durumlardaki şifreleme ve şifre çözme süreleri arasında %15 fark tespit edildi. DNA modeli uygulanmış Blowfish algoritmasının, işlem süresi iki bilgisayar için yaklaşık olarak %20 oranında donanımdan kaynaklı bir mutlak fark olduğu belirlendi. Algoritmanın süre açısından DNA modellemesi sonucunda çok büyük performans kayıpları görüldü.

Bu donanım farkının ortaya çıkardığı diğer bir fark ise Çizelge 5.16’da gösterilen CPU kullanım değerleri arasındaki farktır, Blowfish algoritmasının orijinal CPU kullanımı %45 fark ile bilgisayar-2’nin daha performanslı ve DNA modellemesi sonucu bu fark %35 olduğu görüldü. Bu farkların sebebi donanım olarak daha yüksek özelliklere sahip bilgisayar-2’nin daha az CPU gereksiniminde bulunmasıdır.

RAM açısından Çizelge 5.16 incelendiğinde iki bilgisayarında aralarında göz ardı edilebilecek çok küçük farklar olduğu, gerçekçi ortamlarda fark edilmeyecek düzeyde minimal farklar gözlemlendi.

## **5.6. BLOWFISH VE DES ALGORİTMALARININ PERFORMANS ANALİZİ**

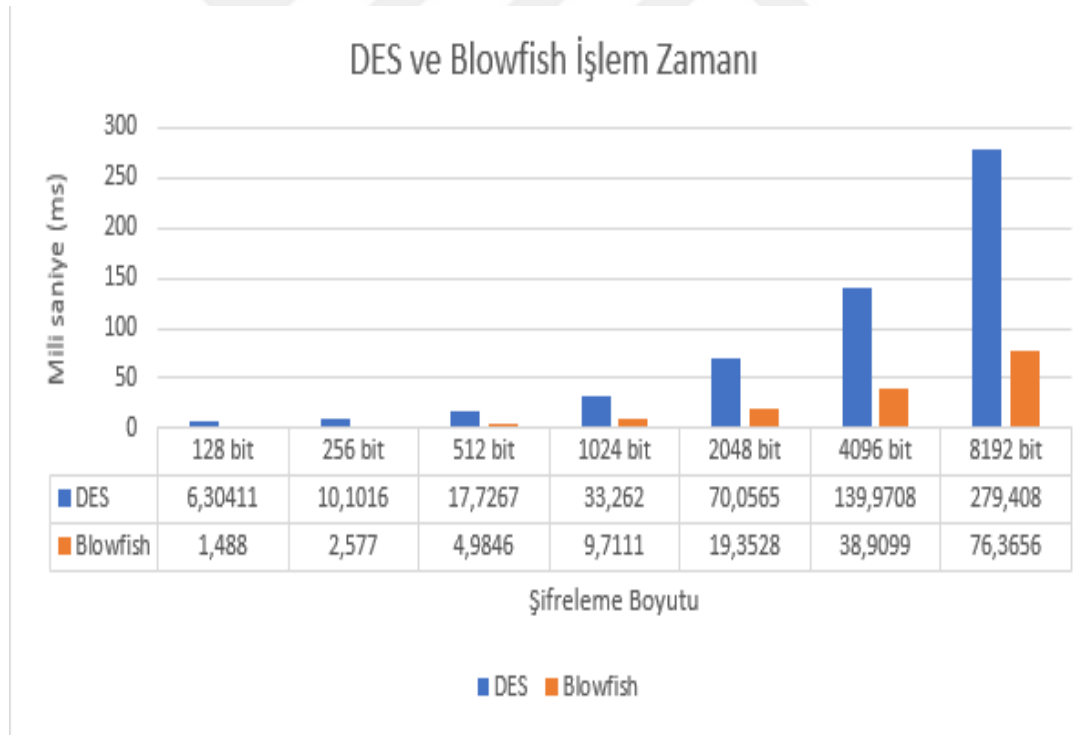
DES ve Blowfish algoritmalarının şifreleme ve şifre çözme işlemleri sonrasındaki performans değerleri aşağıdaki Çizelge 5.17’de gösterildi. Bu değerler hesaplanırken şifreleme algoritmalarının 200 iterasyon sonucundaki ortalama değerleri alınarak hesaplandı, sonuçlar şifreleme ve şifre çözme performans değerlerinin toplamı alınarak gösterildi. Analiz sonuçları Ram, zaman ve CPU üzerinden değerlendirildi. Sonuçlarda zaman olarak saniye, Ram kullanımı olarak megabayt ve CPU kullanımı olarak da yüzde (%) baz alındı.



### 5.6.1. Bilgisayar-1 Performans Analizi

Çizelge 5.17. DES ve Blowfish şifreleme algoritmalarının karşılaştırılması.

| Veri Boyutu (BİT) | DES CPU Kullanım (%) | DES RAM Kullanım (MB) | DES Zaman Kullanım (S) | B.F CPU Kullanım (%) | B.F RAM Kullanım (MB) | B.F Zaman Kullanım (S) |
|-------------------|----------------------|-----------------------|------------------------|----------------------|-----------------------|------------------------|
| 128               | 15,38800             | 0,000506              | 0,006304               | 5,166000             | 0,000332              | 0,001488               |
| 256               | 22,53800             | 0,000473              | 0,010101               | 7,765499             | 0,000335              | 0,002577               |
| 512               | 26,55200             | 0,000436              | 0,017726               | 10,43049             | 0,000338              | 0,004984               |
| 1024              | 26,30749             | 0,000558              | 0,033262               | 21,68700             | 0,000390              | 0,009711               |
| 2048              | 33,11799             | 0,000729              | 0,070056               | 27,39950             | 0,000518              | 0,019352               |
| 4096              | 35,99850             | 0,001387              | 0,139970               | 26,78950             | 0,000968              | 0,038909               |
| 8192              | 40,77950             | 0,002559              | 0,279408               | 25,78200             | 0,001356              | 0,076365               |



Şekil 5.25. DES ve Blowfish işlem zaman grafiği.

DES ve Blowfish (B.F) algoritmalarının farklı boyutlardaki, şifreleme ve şifre çözme performansları, Bilgisayar-1'e göre sonuçları Çizelge 5.17'de gösterildi. Ayrıca Şekil 5.25'de süre açısından performansı daha net anlaşılması adına grafik şeklinde ifade edildi.

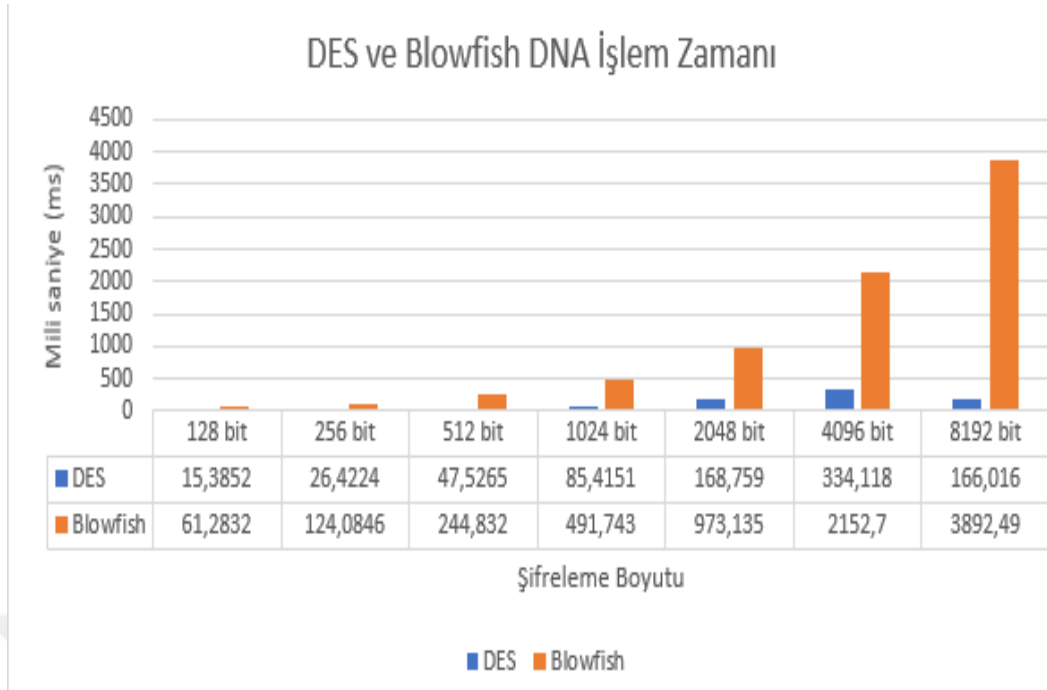
Şekil 5.25’de iki algoritmanın süre açısından farkları gösterilmiş olup, 200 iterasyonun ortalaması alınarak sonuçlar elde edildi. Blowfish algoritmasının süre konusunda yaklaşık olarak DES algoritmasına göre yaklaşık olarak 4 kat daha performanslı olduğu saptandı.

CPU kullanımı yüzdelik olarak incelendiğinde iki algoritmadan Blowfish performans açısından DES’in önünde olduğu görüldü. Ayrıca genel olarak performans olarak önde gözüксе dahi Çizelge 5.17’de görüldüğü üzere şifreleme boyutu arttıkça iki algoritma arasındaki performans farkının azalma gösterdiği tespit edildi.

RAM kullanımı açısından algoritmalarından Blowfish algoritması DES algoritmasına göre az RAM kullandığı ve aralarında yaklaşık olarak 1,5 katlık bir RAM kullanım farkı olduğu saptandı. Şifreleme boyutu 8192 bit olduğunda bu fark biraz daha artarak yaklaşık olarak 2 katlık bir fark gözlemlendi.

Çizelge 5.18. DES ve Blowfish tabanlı DNA şifreleme algoritmalarının karşılaştırılması.

| Veri Boyutu (BİT) | DES CPU Kullanım (%) | DES RAM Kullanım (MB) | DES Zaman Kullanım (S) | B.F CPU Kullanım (%) | B.F RAM Kullanım (MB) | B.F Zaman Kullanım (S) |
|-------------------|----------------------|-----------------------|------------------------|----------------------|-----------------------|------------------------|
| 128               | 26,99450             | 0,000457              | 0,015385               | 25,49949             | 0,000373              | 0,061283               |
| 256               | 26,86050             | 0,000477              | 0,026422               | 25,20250             | 0,000373              | 0,124084               |
| 512               | 25,76450             | 0,000496              | 0,047526               | 25,21500             | 0,000578              | 0,244832               |
| 1024              | 25,21250             | 0,000557              | 0,085415               | 25,18049             | 0,000434              | 0,491743               |
| 2048              | 25,81150             | 0,001090              | 0,168759               | 25,12150             | 0,000598              | 0,973135               |
| 4096              | 26,65699             | 0,001378              | 0,334118               | 31,05500             | 0,001239              | 2,152700               |
| 8192              | 25,40050             | 0,001069              | 0,166016               | 25,12350             | 0,000967              | 3,892490               |



Şekil 5.26. DES ve Blowfish tabanlı DNA işlem zaman grafiği.

DES ve Blowfish algoritmalarının DNA modellemesi sonucunda farklı boyutlardaki şifreleme ve şifre çözme performansları bilgisayar-1'e göre sonuçları Çizelge 5.18'de gösterildi. Ayrıca Şekil 5.26'da süre açısından performansı daha net anlaşılması adına grafik şeklinde ifade edildi.

Şekil 5.26'da iki algoritmanın süre açısından farkları gösterilmiş olup, 200 iterasyonun ortalaması alınarak sonuçlar elde edildi. Süre ölçümü yapıldığında, Blowfish algoritmasının süre konusunda DES algoritması karşısında veri boyutu arttıkça performans kaybettiği açıkça görüldü. DES özellikle büyük boyutlu işlem sürelerinde performans olarak bir adım öne çıktığı saptandı.

CPU kullanımı yüzdelik olarak incelendiğinde iki algortmada DNA kullanırken birbirine çok yakın performanslar sergilediği görüldü. Ayrıca algoritmaların çalışımı esnasında veri boyutunun CPU'yu çok minimal düzeyde etkilediği Çizelge 5.18'de gösterildi.

RAM kullanımı açısından algoritmalarından iki algoritma incelendiğinde Blowfish ile DES arasında, çok küçük farkla Blowfish algoritmasının daha verimli olduğu tespit edildi.

Bu farkın çok değişmemesinin en temel sebeplerinden birisi günümüz bilgisayarlarının donanımsal olarak bu sistemler karşısında yeterli teknolojiye sahip olmalarıdır. İki

algoritmada farklı iki bilgisayarda çok az Ram kullandığı gözlemlendi.

Çizelge 5.19. DNA tabanlı şifreleme sonuçları.

| ŞİFRELEME BOYUTU VE TOPLAM İŞLEM SÜRESİ (ms) | Literatür Çalışmaları |                          |                           |                    |
|--|-----------------------|--------------------------|---------------------------|--------------------|
|  | Şatır ve Talo<br>2021 | Kaundal ve Verma<br>2015 | Pramanik ve Setua<br>2004 | Paul Et Al<br>2016 |
|  | 128 Bit               | 80 Bit                   | 80 Bit                    | 80 Bit             |
|  | 9,0811                | 52,0                     | 69,0                      | 60,0               |
|  | 256 Bit               | 160 Bit                  | 160 Bit                   | 160 Bit            |
|  | 16,32                 | 71,0                     | 103,0                     | 85,0               |
|  | 512 Bit               | 320 Bit                  | 320 Bit                   | 320 Bit            |
|  | 29,798                | 148,0                    | 208,0                     | 165,0              |
|  | 1024 Bit              | 640 Bit                  | 640 Bit                   | 640 Bit            |
|  | 52,15                 | 340,0                    | 510,0                     | 394,0              |
|  | 2048 Bit              | 800 Bit                  | 800 Bit                   | 800 Bit            |
|  | 98,70                 | 461,0                    | 635,0                     | 527,0              |
|  | 4096 Bit              | 4000 Bit                 | 4000 Bit                  | 4000 Bit           |
|  | 194,14                | 2080                     | 2234                      | 2382               |
|  | 8192 Bit              |                          |                           |                    |
|  | 186,6                 |                          |                           |                    |

Bu kısımda DES tabanlı DNA şifrelemeye ait veriler Çizelge 5.19’da gösterildi, önerilen metot, literatürdeki diğer metotlar ile karşılaştırılmıştır. Karşılaştırma yapılırken Bilgisayar-1 verileri göz önüne alınarak yapıldı. Önceki kısımlarda grafik ve çizelgelerde belirtilen RAM ve CPU kullanımı Çizelge 5.19’da gösterilmemiştir. Bunun nedeni Çizelge 5.19’da gösterilen ilgili literatür çalışmalarında RAM ve CPU kullanımına dair hesaplamalar yapılmamıştır. Literatürdeki çalışmalarda (Çizelge 5.19) şifreleme boyutları değişiklik gösterse bile birbirine yakın boyutlardaki şifrelemelerde önerilen metodun süre açısından daha başarılı olduğu saptandı.

Çizelge 5.20. DNA tabanlı şifreleme sonuçları.

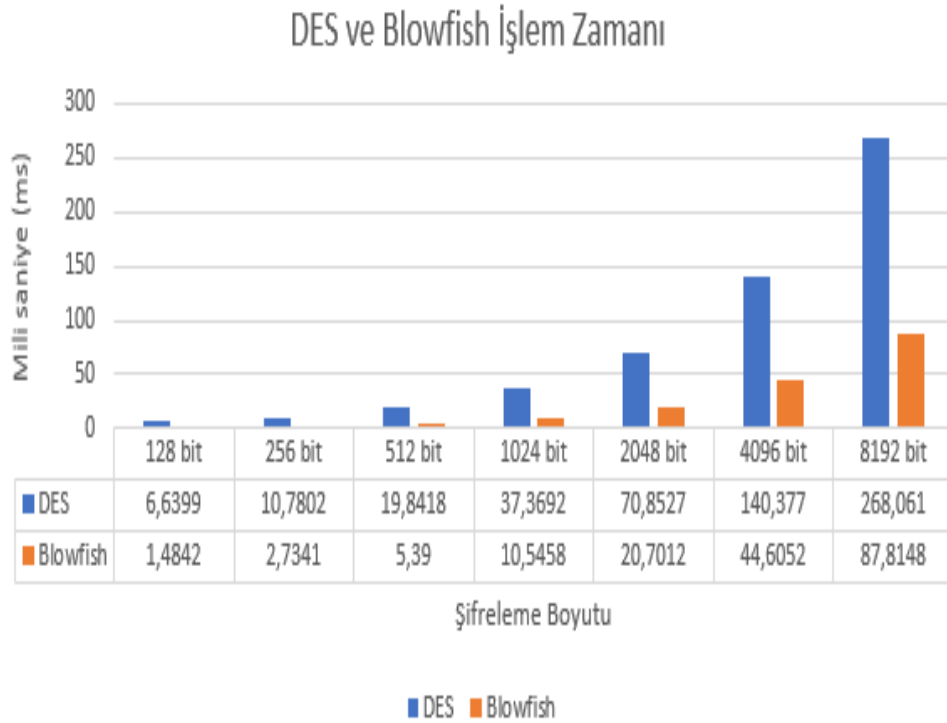
| ŞİFRELEME BOYUTU VE TOPLAM İŞLEM SÜRESİ (ms) | Literatür Çalışmaları |                          |                           |                    |
|--|-----------------------|--------------------------|---------------------------|--------------------|
|  | Şatır ve Talo<br>2021 | Kaundal ve Verma<br>2015 | Pramanik ve Setua<br>2004 | Paul Et Al<br>2016 |
|  | 128 Bit               | 80 Bit                   | 80 Bit                    | 80 Bit             |
|  | 59,42                 | 52,0                     | 69,0                      | 60,0               |
|  | 256 Bit               | 160 Bit                  | 160 Bit                   | 160 Bit            |
|  | 121,5                 | 71,0                     | 103,0                     | 85,0               |
|  | 512 Bit               | 320 Bit                  | 320 Bit                   | 320 Bit            |
|  | 239,84                | 148,0                    | 208,0                     | 165,0              |
|  | 1024 Bit              | 640 Bit                  | 640 Bit                   | 640 Bit            |
|  | 482,03                | 340,0                    | 510,0                     | 394,0              |
|  | 2048 Bit              | 800 Bit                  | 800 Bit                   | 800 Bit            |
|  | 953,78                | 461,0                    | 635,0                     | 527,0              |
|  | 4096 Bit              | 4000 Bit                 | 4000 Bit                  | 4000 Bit           |
|  | 2213,79               | 2080                     | 2234                      | 2382               |
|  | 8192 Bit              |                          |                           |                    |
|  | 3816,12               |                          |                           |                    |

Bu bölümde Blowfish tabanlı DNA şifrelemeye ait veriler Çizelge 5.20’de gösterildi, önerilen metot, literatürdeki diğer metotlar ile karşılaştırıldı. Karşılaştırma yapılırken Bilgisayar-1 verileri göz önüne alınarak yapıldı. Önceki kısımlarda grafik ve çizelgelerde belirtilen RAM ve CPU kullanımı Çizelge 5.20’de gösterilmemiştir. Bunun nedeni Çizelge 5.20’de gösterilen ilgili literatür çalışmalarında RAM ve CPU kullanımına dair hesaplamalar yapılmamıştır. Literatürdeki çalışmalarda (Çizelge 5.20) şifreleme boyutları değişiklik gösterse bile birbirine yakın boyutlardaki şifrelemelerde önerilen metodun süre açısından paralellik ve denklik gösterdiği saptandı. Benzer boyutlardaki şifreleme ve şifre çözme sürelerinin toplamı karşılaştırıldığında DNA tabanlı Blowfish algoritmasının DNA tabanlı DES algoritmasına göre daha yavaş olduğu gözlemlendi.

### 5.6.2. Bilgisayar-2 Performans Analizi

Çizelge 5.21. DES ve Blowfish şifreleme algoritmalarının karşılaştırılması.

| Veri Boyutu (BİT) | DES CPU Kullanım (%) | DES RAM Kullanım (MB) | DES Zaman Kullanım (S) | B.F CPU Kullanım (%) | B.F RAM Kullanım (MB) | B.F Zaman Kullanım (S) |
|-------------------|----------------------|-----------------------|------------------------|----------------------|-----------------------|------------------------|
| 128               | 7,421000             | 0,000314              | 0,006639               | 1,436000             | 0,000574              | 0,0014842              |
| 256               | 12,16249             | 0,000749              | 0,010780               | 2,945000             | 0,000658              | 0,0027341              |
| 512               | 15,16499             | 0,000691              | 0,019841               | 9,116499             | 0,000643              | 0,0053900              |
| 1024              | 13,48349             | 0,000771              | 0,037369               | 10,50899             | 0,000604              | 0,0105458              |
| 2048              | 15,14349             | 0,000752              | 0,070852               | 14,76849             | 0,000649              | 0,0207012              |
| 4096              | 13,14750             | 0,001139              | 0,140377               | 14,21099             | 0,000608              | 0,0446052              |
| 8192              | 13,28600             | 0,002737              | 0,268061               | 13,32550             | 0,000627              | 0,0878148              |



Şekil 5.27. DES ve Blowfish işlem zaman grafiği.

DES ve Blowfish algoritmalarının farklı boyutlardaki, şifreleme ve şifre çözme performansları bilgisayar-2'ye göre sonuçları Çizelge 5.19'da gösterildi. Şekil 5.27'de süre açısından performansı daha net anlaşılması adına grafik şeklinde ifade edildi.

Şekil 5.27’de iki algoritmanın süre açısından farkları gösterilip, 200 iterasyonun ortalaması alınarak sonuçlar elde edildi. Blowfish algoritmasının süre konusunda yaklaşık olarak DES algoritmasına göre yaklaşık olarak 4 kat daha performanslı olduğu saptandı.

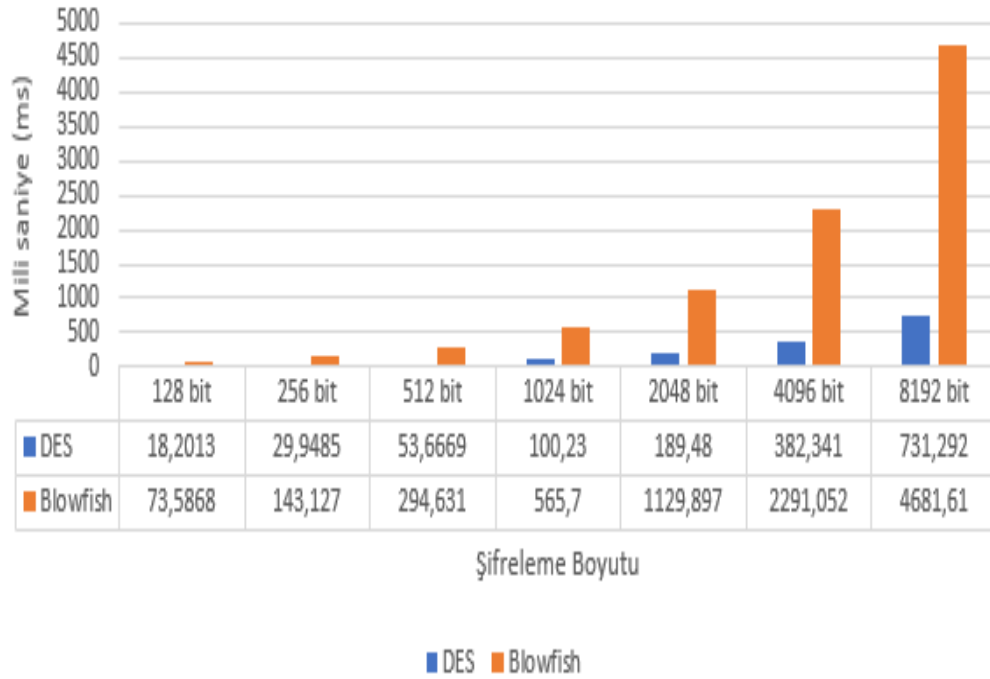
CPU kullanımı yüzdelik olarak incelendiğinde, iki algoritmadan Blowfish performans açısından DES ile karşılaştırıldığında, küçük boyutlu verilerde Blowfish’in çok daha performanslı olduğu görüldü. 1024 bit ve üzeri boyutlardaki verilerde Çizelge 5.19’da görüldüğü üzere şifreleme boyutu arttıkça iki algoritma arasındaki performans farkı neredeyse yok denecek kadar az olduğu söylenilebilir.

RAM kullanımı açısından algoritmalarından Blowfish algoritması DES algoritmasına göre az RAM kullandığı görüldü. 1024 bit ve üzeri çalışmalarda Blowfish algoritmasının daha verimli olduğu tespit edildi.

Çizelge 5.22. DES ve Blowfish tabanlı DNA şifreleme algoritmalarının karşılaştırılması.

| Veri Boyutu (BİT) | DES CPU Kullanım (%) | DES RAM Kullanım (MB) | DES Zaman Kullanım (S) | B.F CPU Kullanım (%) | B.F RAM Kullanım (MB) | B.F Zaman Kullanım (S) |
|-------------------|----------------------|-----------------------|------------------------|----------------------|-----------------------|------------------------|
| 128               | 16,01449             | 0,000609              | 0,018201               | 12,84600             | 0,000299              | 0,073586               |
| 256               | 14,93750             | 0,000649              | 0,029948               | 12,97800             | 0,000340              | 0,143127               |
| 512               | 14,99700             | 0,000752              | 0,053666               | 12,69249             | 0,000565              | 0,294631               |
| 1024              | 13,51950             | 0,000791              | 0,100230               | 12,74450             | 0,000483              | 0,565700               |
| 2048              | 13,80450             | 0,000976              | 0,189480               | 13,02649             | 0,000238              | 1,129897               |
| 4096              | 12,91150             | 0,001303              | 0,382341               | 13,34049             | 0,000791              | 2,291052               |
| 8192              | 13,47450             | 0,003781              | 0,731292               | 16,00850             | 0,001838              | 4,681615               |

## DES ve Blowfish DNA İşlem Zamanı



Şekil 5.28. DES ve Blowfish DNA işlem zaman grafiği.

DES ve Blowfish algoritmalarının DNA modellenmesi ile farklı boyutlardaki şifreleme ve şifre çözme performansları Bilgisayar-2'ye göre sonuçları Çizelge 5.20'de gösterildi. Ayrıca Şekil 5.28'de süre açısından performansı daha net anlaşılması adına grafik şeklinde ifade edildi.

Şekil 5.28'de iki algoritmanın süre açısından farkları gösterilmiş olup, 200 iterasyonun ortalaması alınarak sonuçlar elde edildi. Süre ölçümü yapıldığında Blowfish algoritmasının süre konusunda DES algoritması karşısında veri boyutu arttıkça performans kaybettiği açıkça görüldü. DES özellikle büyük boyutlu işlem sürelerinde performans olarak bir adım öne çıktığı saptandı.

CPU kullanımı yüzdelik olarak incelendiğinde iki algoritmada DNA kullanırken birbirine çok yakın performanslar sergilediği tespit edildi. Ayrıca algoritmaların çalışımı esnasında veri boyutunun CPU'yu çok minimal düzeyde etkilediği Çizelge 5.20'de gösterildi.

RAM kullanımı açısından algoritmalarından Blowfish algoritması DES algoritmasına göre az RAM kullandığı görüldü. Veri boyutu arttıkça Blowfish algoritmasının daha verimli olduğu tespit edildi.



Çizelge 5.23. DNA tabanlı şifreleme sonuçları.

| ŞİFRELEME BOYUTU VE TOPLAM İŞLEM SÜRESİ (ms) | Literatür Çalışmaları |                          |                           |                    |
|--|-----------------------|--------------------------|---------------------------|--------------------|
|  | Şatır ve Talo<br>2021 | Kaundal ve<br>Verma 2015 | Pramanik ve Setua<br>2004 | Paul Et Al<br>2016 |
|  | 128 Bit               | 80 Bit                   | 80 Bit                    | 80 Bit             |
|  | 11,56                 | 52,0                     | 69,0                      | 60,0               |
|  | 256 Bit               | 160 Bit                  | 160 Bit                   | 160 Bit            |
|  | 19,16                 | 71,0                     | 103,0                     | 85,0               |
|  | 512 Bit               | 320 Bit                  | 320 Bit                   | 320 Bit            |
|  | 33,82                 | 148,0                    | 208,0                     | 165,0              |
|  | 1024 Bit              | 640 Bit                  | 640 Bit                   | 640 Bit            |
|  | 62,86                 | 340,0                    | 510,0                     | 394,0              |
|  | 2048 Bit              | 800 Bit                  | 800 Bit                   | 800 Bit            |
|  | 118,62                | 461,0                    | 635,0                     | 527,0              |
|  | 4096 Bit              | 4000 Bit                 | 4000 Bit                  | 4000 Bit           |
|  | 241,96                | 2080                     | 2234                      | 2382               |
|  | 8192 Bit              |                          |                           |                    |
|  | 563,23                |                          |                           |                    |

Bu kısımda DES tabanlı DNA şifrelemeye ait veriler Çizelge 5.23’de gösterildi, önerilen metot, literatürdeki diğer metotlar ile karşılaştırıldı. Karşılaştırma yapılırken Bilgisayar-2 verileri göz önüne alınarak yapıldı. Önceki kısımlarda grafik ve çizelgelerde belirtilen RAM ve CPU kullanımı Çizelge 5.23’de gösterilmemiştir. Bunun nedeni Çizelge 5.19’da gösterilen ilgili literatür çalışmalarında RAM ve CPU kullanımına dair hesaplamalar yapılmamıştır. Literatürdeki çalışmalarda (Çizelge 5.23) şifreleme boyutları değişiklik gösterse bile birbirine yakın boyutlardaki şifrelemelerde önerilen metodun süre açısından daha başarılı olduğu saptandı.

Çizelge 5.24. DNA tabanlı şifreleme sonuçları.

| ŞİFRELEME BOYUTU VE TOPLAM İŞLEM SÜRESİ (ms) | Literatür Çalışmaları |                          |                           |                    |
|--|-----------------------|--------------------------|---------------------------|--------------------|
|  | Şatır ve Talo<br>2021 | Kaundal ve Verma<br>2015 | Pramanik ve Setua<br>2004 | Paul Et Al<br>2016 |
|  | 128 Bit               | 80 Bit                   | 80 Bit                    | 80 Bit             |
|  | 72,10                 | 52,0                     | 69,0                      | 60,0               |
|  | 256 Bit               | 160 Bit                  | 160 Bit                   | 160 Bit            |
|  | 140,39                | 71,0                     | 103,0                     | 85,0               |
|  | 512 Bit               | 320 Bit                  | 320 Bit                   | 320 Bit            |
|  | 289,24                | 148,0                    | 208,0                     | 165,0              |
|  | 1024 Bit              | 640 Bit                  | 640 Bit                   | 640 Bit            |
|  | 555,15                | 340,0                    | 510,0                     | 394,0              |
|  | 2048 Bit              | 800 Bit                  | 800 Bit                   | 800 Bit            |
|  | 1109,18               | 461,0                    | 635,0                     | 527,0              |
|  | 4096 Bit              | 4000 Bit                 | 4000 Bit                  | 4000 Bit           |
|  | 2246,44               | 2080                     | 2234                      | 2382               |
|  | 8192 Bit              |                          |                           |                    |
|  | 4593,80               |                          |                           |                    |

Bu bölümde Blowfish tabanlı DNA şifrelemeye ait veriler Çizelge 5.24’de gösterildi, önerilen metot, literatürdeki diğer metotlar ile karşılaştırıldı. Karşılaştırma yapılırken Bilgisayar-2 verileri göz önüne alınarak yapıldı. Önceki kısımlarda grafik ve çizelgelerde belirtilen RAM ve CPU kullanımı Çizelge 5.24’de gösterilmemiştir. Bunun nedeni Çizelge 5.24’de gösterilen ilgili literatür çalışmalarında RAM ve CPU kullanımına dair hesaplamalar yapılmamıştır. Literatürdeki çalışmalarda (Çizelge 5.24) şifreleme boyutları değişiklik gösterse bile birbirine yakın boyutlardaki şifrelemelerde önerilen metodun süre açısından paralellik ve denklik gösterdiği saptandı bu küçük performans kayıplarına, bilgisayarların yazılımsal ve donanımsal özelliklerinin de etkisi olduğu tespit edildi. Benzer boyutlardaki şifreleme ve şifre çözme sürelerinin toplamı karşılaştırıldığında DNA tabanlı Blowfish algoritmasının DNA tabanlı DES algoritmasına göre daha yavaş olduğu gözlemlendi.

## 6. SONUÇLAR VE ÖNERİLER

Bilgi güvenliği göz önüne alındığında değerlendirilen tüm sistematik olguları ve bu güvenliği sağlamak için şifreleme tekniklerinden istifade ederler. Bilgi teknolojilerindeki kalkınmalarla doğru orantılı olarak bilgi güvenliğinin sadece tek yönlü değil çeşitli alanları da içinde barındıran bir alan olduğundan dolayı önemi bir noktaya gelmiştir. Bu çalışmada bilgi güvenliği için çoğunlukla kullanılan simetrik ve asimetrik algoritmalar genel olarak açıklandı. Simetrik algoritmalarından DES ve Blowfish zaman karmaşıklığı, işlemci karmaşıklığı ve bellek karmaşıklığı açısından incelendi ve yeni bir DNA modellemesi yapıldı. Ayrıca bu çalışma genel olarak kriptografide kullanılan genel algoritmaların çalışma prensiplerini ve teknik detaylarını anlatacak şekilde oluşturuldu.

Bu çalışmada, genel olarak son yıllarda popülaritesi artan ve kriptoloji alanında yeni bir alan diyebileceğimiz DNA tabanlı şifreleme kullanıldı. Bu kullanım hali hazırda var olan simetrik algoritmalarından DES ve Blowfish algoritması üzerinde bu çalışmada uygulandı. DNA tabanlı şifreleme yapılırken algoritmaların teknik yapıları ve özellikleri incelenerek simetrik iki algoritma seçildi, bunun nedeni var olan asimetrik şifreleme sistemleri teknik yapılarından dolayı simetrik şifreleme algoritmalarına göre daha yavaş çalışan algoritmalar. DNA modellemesi ise algoritmaya ek bir yük olacağı ve asimetrik algoritmalarının verimliliği düşüreceğini öngörerek bu iki simetrik algoritma üzerinde çalışma yapıldı.

DNA modellemesi yapılan bu iki şifreleme algoritması süre RAM ve CPU performansları açısından incelendi. Algoritmaların orijinal ve DNA tabanlı şifreleme yapıldıktan sonra karşılaştırılması yapıldı. Çalışma sonucunda elde edilen verilere göre; DES ve Blowfish algoritmalarının orijinal halleri arasında süre, RAM, CPU açısından farklı sonuçlar elde edildi. Bu sonuçlar iki farklı bilgisayar tarafından alındı. Elde edilen sonuçlara göre, literatürdeki kaynaklarla doğru orantılı olarak Blowfish algoritmasının performans analizi yaptığımız 3 performans kriterine göre daha başarılı sonuçlar verdiği görüldü. DNA tabanlı şifreleme yapıldıktan sonraki sonuçlar incelendiğinde algoritmaların güncel halleri, orijinal hallerine göre özellikle süre açısından yavaşladığı tespit edildi. RAM ve CPU açısından incelendiğinde iki algoritmada da genel olarak

şifrelenecek veri boyutu 1024 bit üzeri olduğu durumlarda DNA şifrelemesi sonucu verimliliğinde çok küçük performans farkları gözlemlendi. Bununla beraber genel olarak algoritmaların orijinal halleri ile kıyas yapıldığında Blowfish algoritmasının performans açısından daha iyi durumda olduğu görülse bile, DNA modellemesi sonucunda şifrelenecek metin boyutunu daha büyük verilere çıkarıldığında Blowfish algoritmasının süre açısından önemli derecede performans kaybettiği gözlemlendi. DES algoritmasının veri boyutu arttıkça şifreleme işlemlerinde daha iyi performans sağladığı görüldü. Bu sonuçlar ışığında şifrelemelerde Blowfish algoritmasının DNA modellemesi sonucu daha başarısız olduğu, DES algoritmasının da veri boyutu yükseldikçe daha verimli olarak çalıştığı sonucuna varıldı. Genel olarak bilgisayar donanımları farklı olsa bile iki bilgisayarda da sonuçların birbirine paralel olarak değiştiği saptandı.

Literatür çalışmaları incelendiğinde, DNA tabanlı şifreleme yapıldıktan sonra DES tabanlı DNA şifrelemesinin süre açısından kıyaslaması yapılan literatürdeki diğer DNA tabanlı şifreleme algoritmalarına karşı bariz bir üstünlük sağladığı görüldü. Blowfish için uygulanan DNA tabanlı şifreleme sonucunda süre performansı açısından incelemesi yapılan literatür çalışmalarına paralel ve denk sonuçlar elde edildi.

İleri dönemde yapılacak çalışmalarda DES ve Blowfish algoritmaları gibi simetrik algoritmalar üzerinde yapılacak DNA modellemeleri, performans kriterleri yerine algoritmaların güvenilirliğini arttıracak şekilde uygulamalar yapılabilir. Bu çalışmalarda DES gibi sabit anahtar uzunluğuna sahip algoritmaların anahtar boyutu DNA ile arttırılabilir veya tek kullanımlık pedler (OTP) şifreleme için kullanılabilir. Anahtar boyutu arttıkça algoritma daha karmaşık bir yapıya bürünecektir ve güvenlik açısından daha güvenli olacaktır. Blowfish gibi değişken anahtar uzunluğuna sahip algoritmalar için de F fonksiyonu içinde karmaşıklık işlemleri oluşturulup daha karmaşık ve güvenli hale algoritma getirilebilir. Bunun dışında DNA yapısından esinlenerek kriptoloji üzerine efektif fikirler ortaya atılabilir.

## 7. KAYNAKLAR

- [1] E. Güvenoğlu, “Görüntü şifreleme ve performans analizleri,” Yüksek lisans tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Trakya Üniversitesi, Tekirdağ, Türkiye, 2006.
- [2] S. Pramanik ve S. K. Setua, “DNA cryptography,” *2012 7th International Conference on Electrical and Computer Engineering*, Bangladeş, 2012, ss. 551–554.
- [3] M. E. Hellman, “An overview of public key cryptography,” *IEEE Communications Magazine*, c. 40, sayı 5, ss. 42–49, 2002.
- [4] Ü. Günden, “Şifreleme algoritmalarının performans analizi,” Yüksek lisans tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, Sakarya, Türkiye, 2010.
- [5] N. Kumar, J. Thakur ve A. Kalia, “Performance analysis of symmetric key cryptography algorithms: DES, AES and Blowfish,” *An International Journal of Engineering Sciences*, c. 6, sayı 1, ss. 28–37, 2011.
- [6] T. Yerlikaya, “Şifreleme algoritmalarının analizi,” Doktora tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Trakya Üniversitesi, Tekirdağ, Türkiye 2006.
- [7] M. Misbahuddin ve M. H. N. P, “DNA for information security: A survey on DNA computing and a pseudo DNA method based on central dogma of molecular biology,” *International Conference on Computing and Communication Technologies*, Hindistan, 2014, ss. 1–6.
- [8] B. D. Patnala, “DNA cryptography life blood for new ERA computers,” *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing*, Hindistan, 2017, ss. 1178–1184.
- [9] M. Tektas, F. Baba, ve M. Çalışkan, “Şifreleme algoritmalarının sınıflandırılması ve bir kredi kartı uygulaması,” *3rd International Advanced Technologies Symposium*, Türkiye, 2003, ss 18-20.
- [10] K. E. Donald, “Volume 2 Seminumerical Algorithms,” *The Art of Computer Programming*, 4. Baskı, New Jersey, ABD: The Society for Industrial and Applied Mathematics, 1969, böl. 2, ss. 306-308.
- [11] B. Kaliski, “The mathematics of the RSA public-key cryptosystem,” *RSA Laboratories*, c. 1, sayı 4, ss. 1-9, 1989.
- [12] *Standard Specifications For Public–Key Cryptography*, IEEE P1398, 1998.
- [13] T.C. TÜBİTAK Teknoloji ve Yenilik Destek Programları Başkanlığı. (2018, 24 Ekim). *Elektronik imza hakkında bilgiler* [Online]. Erişim: <https://eteydeb.tubitak.gov.tr/eimzabilgisayfasi.html>.
- [14] K. Erkoç, “Kriptoloji ve bilgi güvenliği,” Yüksek lisans tezi, Bilgisayar ve Bilişim Mühendisliği, Fen Bilimleri Enstitüsü, Sakarya Üniversitesi, Sakarya, Türkiye, 2004.

- [15] Ö. Ökten, “Standard digital signature algorithm (DSA) and secure hash Algorithm (SHA) in public key cryptology,” Yüksek lisans tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Orta Doğu Teknik Üniversitesi, Ankara, Türkiye, 1997.
- [16] E. Andiç, “Bilgisayar haberleşmesinde şifreleme (Kripto) yazılımıyla güvenliğin sağlanması,” Yüksek lisans tezi, Bilgisayar Mühendisliği, Fen Bilimleri Enstitüsü, Marmara Üniversitesi, İstanbul, Türkiye, 2002.
- [17] M. Yıldırım, “DES ve DES benzeri şifreleme sistemlerinin diferansiyel kripto analizi,” Yüksek lisans tezi, Elektrik ve Elektronik Mühendisliği, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 1995.
- [18] S. Kansal, “Performance evaluation of various symmetric encryption algorithms,” *2014 International Conference on Parallel, Distributed and Grid Computing*, Hindistan, 2014, ss. 105–109.
- [19] A.Y. Doğan, “AES algoritmasının FPGA üzerinde düşük güçlü tasarımı,” Yüksek lisans tezi, Elektronik Mühendisliği, Fen Bilimleri Enstitüsü, Türkiye, İstanbul Teknik Üniversitesi, İstanbul, 2008.
- [20] L. R. Knudsen, V. Rijmen, R. L. Rivest, ve M. J. B. Robshaw, “On the design and security of RC2,” *Lecture Notes in Computer Science*, 1. baskı, Paris, Fransa : Springer, 1998, böl. 4, ss. 206–221.
- [21] A. Kaundal ve A. Verma, “DNA based cryptography: A review,” *International Journal of Information and Computation Technology*, c. 4, sayı 7, ss. 693–698, 2014.
- [22] N. Kumar, J. Thakur, ve A. Kalia, “Performance analysis of symmetric key cryptography algorithms: DES, AES and Blowfish,” *An International Journal of Engineering Sciences*, c. 6, sayı 1, ss. 28–37, 2011.
- [23] M. Faheem, S. Jamel, A. Hassan, Z. A., N. Shafinaz, ve M. Mat, “A survey on the cryptographic encryption algorithms,” *International Journal of Advanced Computer Science and Applications*, c. 8, sayı 11, ss. 333-344, 2017.
- [24] A. Kumar, “Feistel inspired structure for DNA,” Yüksek lisans tezi, Bilgi Güvenliği Departmanı, Thapar University, Patiala, Hindistan, 2014.
- [25] J. Chen, “A DNA-based, biomolecular cryptography design,” *Proceedings - IEEE International Symposium on Circuits and Systems*, Tayland, 2003, ss. 822–825.
- [26] A. Gehani, T. LaBean ve J. Reif, “DNA Based Cryptography,” In *Aspects of Molecular Computing*, 1. Baskı, Berlin, Almanya: Heidelberg, 2003, böl. 11, ss. 167-188.
- [27] M. A. Muslim, B. Prasetyo, ve Alamsyah, “Implementation twofish algorithm for data security in a communication network using library chilkat encryption activex,” *Journal of Theoretical and Applied Information Technology*, c. 84, sayı 3, ss. 370–375, 2016.
- [28] K. Tanaka, A. Okamoto, ve I. Saito, “Public-key system using DNA as a one-way function for key distribution,” *BioSystems*, c. 81, sayı 1, ss. 25–29, 2005.

- [29] S. T. Amin, M. Saeb, ve S. El-Gindi, "A DNA-based implementation of yaea encryption algorithm," *Proceedings of the 2nd IASTED International Conference on Computational Intelligence*, Mısır, 2006, ss. 116–120.
- [30] S. Aradhyamath ve J. Paulose, "Multi-key Modified Tiny Encryption Algorithm for health care," *International Journal of Engineering & Technology*, c. 7, sayı 2.14, ss. 559-563, 2018.
- [31] A. K. Verma, M. Dave, ve R. C. Joshi, "DNA cryptography: A novel paradigm for secure routing in mobile ad hoc networks (MANETs)," *Journal of Discrete Mathematical Sciences and Cryptography*, c. 11, sayı 4, ss. 393–404, 2008.
- [32] G. Cui, L. Qin, Y. Wang, ve X. Zhang, "An encryption scheme using DNA technology," *2008 3rd International Conference on Bio-Inspired Computing: Theories and Applications*, Avustralya, 2008, ss. 37–41.
- [33] X. Zheng, "Research for the application and safety of MD5 algorithm in password authentication," *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, Çin, 2012, ss. 2216-2219.
- [34] X. J. Lai, M. X. Lu, L. Qin, J. S. Han, ve X. W. Fang, "Asymmetric encryption and signature method with DNA technology," *Science in China, Series F: Information Sciences*, c. 53, sayı 3, ss. 506–514, 2010.
- [35] D. Kumar ve S. Singh, "Secret data writing using DNA sequences," *2011 International Conference on Emerging Trends in Networks and Computer Communications*, Hindistan, 2012, ss. 402–405.
- [36] Y. Zhang, B. Fu, ve X. Zhang, "DNA cryptography based on DNA fragment assembly," *2012 8th International Conference on Information Science and Digital Content Technology*, Güney Kore, 2012, ss. 179-182.
- [37] O. Tornea ve M. E. Borda, "Security and complexity of a DNA-based cipher," *Proceedings - RoEduNet IEEE International Conference*, Romanya, 2013, ss. 1-5.
- [38] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, c. 266, sayı 5187, ss. 1021-1024, 1994.
- [39] R. J. Lipton, "DNA solution of hard computational problems," *Science*, c. 268, sayı 5210, ss. 542–545, 1995.
- [40] R. J. Lipton, "DIMACS," *DNA Based Computers*, 4. Baskı, New Jersey, ABD: Princeton University, 1995, böl. 27, ss. 15-39.
- [41] Q. Ouyang, P. D. Kaplan, S. Liu ve A. Libchaber, "DNA solution of the maximal clique problem," *Science*, c. 278, sayı 5337, ss. 446–449, 1997.
- [42] S. Kalsi, H. Kaur, ve V. Chang, "DNA Cryptography and Deep Learning using Genetic Algorithm with NW algorithm for key generation," *Journal of Medical Systems*, c. 42, sayı 1, ss. 1-12, 2018.
- [43] B. B., J. Frank, ve T. Mahalakshmi, "Secure data transfer through DNA cryptography using symmetric algorithm," *International Journal of Computer Applications*, c. 133, sayı 2, ss. 19–23, 2016.
- [44] S. Bismi Beegom ve S. Jose, "An enhanced cryptographic model based on DNA approach," *Proceedings of the International Conference on Electronics, Communication and Aerospace Technology*, Arjantin, 2017, ss. 317–322.

- [45] S. Cherillath Sukumaran ve M. Mohammed, “DNA cryptography for secure data storage in cloud,” *International Journal of Network Security*, c. 20, sayı 3, ss. 447–454, 2018.
- [46] M. R. Biswas, K. M. R. Alam, S. Tamura, ve Y. Morimoto, “A technique for DNA cryptography based on dynamic mechanisms,” *Journal of Information Security and Applications*, c. 48, sayı 63, ss. 10-23, 2019.
- [47] P. T. Akkasaligar ve S. Biradar, “Selective medical image encryption using DNA cryptography,” *Information Security Journal*, c. 29, sayı 2, ss. 91–101, 2020.





# ÖZGEÇMİŞ

## KİŞİSEL BİLGİLER

Adı Soyadı : Furkan TALO

Yabancı Dili : İngilizce

## ÖĞRENİM DURUMU

| Derece    | Alan   | Okul/Üniversite         | Mezuniyet Yılı |
|-----------|--|-------------------------|----------------|
| Y. Lisans | Elektrik Elektronik ve<br>Bilgisayar Mühendisliği. | Düzce Üniversitesi      | 2021           |
| Lisans    | Bilgisayar Mühendisliği.                           | Tunceli Üniversitesi    | 2016           |
| Lise      | Düz Lise   | Elâzığ Balakgazi Lisesi | 2012           |

## YAYINLAR