**Şifreleme Algoritması Metodolojisi**

Bu çalışma, görüntü şifrelemesi için DNA tabanlı kodlama, kaotik logistic harita ve PBKDF2-HMAC-SHA256 ile parola tabanlı anahtar türetmeyi birleştiren bir sistem üzerine kuruludur. Aşağıdaki adımlar, algoritmanın matematiksel ve teknik işleyişini ayrıntılarıyla açıklamaktadır.

**1. Girdi ve Anahtar Türetimi**

**Girdi:**

* RGB formatında bir görüntü dosyası
* Kullanıcıdan alınan bir parola (password)

**İşlem:**

* PBKDF2 algoritması SHA256 ile birlikte kullanılır:

Key=PBKDF2(password,salt,iterations=100000,keylen=32)\text{Key} = \text{PBKDF2}(\text{password}, \text{salt}, \text{iterations}=100000, \text{keylen}=32)

* 16 bayt salt rastgele üretilir, key 32 bayttır.
* Bu key, byte dizisinden bit dizisine dönüştürülür:

key\_bits=unpackbits(key)\text{key\\_bits} = \text{unpackbits}(\text{key})

**Çıktı:**

* Anahtar bit dizisi (key\_bits)
* Salt (görüntüyle birlikte kaydedilir)

**2. Görüntü Dönüşümü ve DNA Kodlama**

**Girdi:**

* RGB görüntü (M x N x 3)

**İşlem:**

* Görüntü byte dizisine, ardından bit dizisine dönüştürülür:

img\_bits=unpackbits(image)\text{img\\_bits} = \text{unpackbits}(\text{image})

* 2-bit gruplar DNA harflerine dönüştürülür (kural 1-8 aralığında seçilir):

00→A,01→T,10→C,11→G00 \rightarrow A,\quad 01 \rightarrow T,\quad 10 \rightarrow C,\quad 11 \rightarrow G

* Aynı kodlama key için de uygulanır.
* Ardından DNA seviyesinde XOR yapılır:

DNAxor1=DNAimg⊕DNAkey\text{DNA}\_{xor1} = \text{DNA}\_{img} \oplus \text{DNA}\_{key}

**Çıktı:**

* DNA dizisi: DNA\_xor1

**3. Logistic Map Tabanlı Kaotik XOR**

**Girdi:**

* DNA\_xor1 dizisi
* Logistic parametreleri: x0=[x1,x2,x3], r=[r1,r2,r3]x0 = [x\_1, x\_2, x\_3],\ r = [r\_1, r\_2, r\_3]

**İşlem:**

Her bir logistic dizisi için:

Xn+1=r⋅Xn⋅(1−Xn)X\_{n+1} = r \cdot X\_n \cdot (1 - X\_n)

* X1,X2,X3X1, X2, X3 dizileri oluşturulur. Uzunluk: DNA dizisi kadar.
* Her biri 0-255'e ölçeklenir, byte'a çevrilir, unpack edilip DNA dizisine dönüştürülür.

DNAX=binary\_to\_dna(unpackbits(X⋅255))\text{DNA}\_{X} = \text{binary\\_to\\_dna}(\text{unpackbits}(X \cdot 255))

* Ardışık 3 XOR adımı:

DNA1=DNAxor1⊕DNAX1\text{DNA}\_1 = \text{DNA}\_{xor1} \oplus \text{DNA}\_{X1} DNA2=DNA1⊕DNAX2\text{DNA}\_2 = \text{DNA}\_1 \oplus \text{DNA}\_{X2} DNA3=DNA2⊕DNAX3\text{DNA}\_3 = \text{DNA}\_2 \oplus \text{DNA}\_{X3}

**Çıktı:**

* DNA\_3 dizisi

**4. Scrambling (Karıştırma)**

**Girdi:**

* DNA\_3 dizisi
* Logistic dizisi X1

**İşlem:**

* X1 dizisi sıralanır: X1\_sorted
* Orijinal indeksler toplanarak scramble dizisi oluşturulur

DNAscrambled=scramble(DNA3,argsort(X1))\text{DNA}\_{scrambled} = \text{scramble}(\text{DNA}\_3, \text{argsort}(X1))

**Çıktı:**

* Karıştırılmış DNA dizisi

**5. DNA → Binary → Görüntü**

**Girdi:**

* Scrambled DNA dizisi

**İşlem:**

* DNA harfleri tekrar 2-bit formatına, oradan byte formatına çevrilir.
* Görüntü şekline (M x N x 3) yeniden şekillendirilir.

**Çıktı:**

* PNG formatında şifreli görüntü dosyası
* salt ve meta.json dosyası (içinde dna\_rule, x0\_list, r\_list)

**6. Decryption (Ters İşlem)**

**Girdi:**

* Şifreli görüntü
* Kullanıcı şifresi
* Salt ve meta.json dosyası (otomatik okunur)

**İşlem:**

1. Salt + parola ile key tekrar üretilir.
2. DNA kodlama ile şifreli görüntü DNA'ya çevrilir.
3. Scramble tersine çevrilir (unscramble).
4. Logistic DNA dizileri tekrar üretilir.
5. XOR ters sırada uygulanır: X3, X2, X1
6. Key DNA ile tekrar XOR yapılır
7. Bit çevirme ve reshape ile orijinal görüntü kurtarılır.

**Çıktı:**

* Orijinal görüntü dosyası (decrypted.png)

Bu metodoloji, DNA kodlaması ile biyoinspirasyon, logistic haritalarla kaotik dağılım ve parola tabanlı anahtar türetmeyle klasik AES/RSA sistemlerine alternatif bir şifreleme mimarisi sunar. Yüksek güvenlik ve parametre kontrolü sağlar.