Proje Ana Alanı **:Matematik**

Proje Tematik Alanı **:Özgün Algoritma Tasarımı**

Proje Adı (Başlığı) **:Faktöriyel Sayılarla Oluşturulmuş Tabanla Kriptografi Algoritması**

Özet

Kriptoloji kelimesinin Yunancadan geldiği ve "kryptos" , "logos" kelimelerinin birleşimi olduğu bilinmektedir. Kryptos kelimesi "gizli"; logos ise "sebep-sonuç ilişkisi kurma, mantıksal çözümleme alanı" anlamı taşımaktadır. Şifreleme, hassas bilgilerin güvenilir olmayan bir araçla, genellikle bir elektronik ortamda alıcıya gönderilirken metnin değiştirilmesi şeklinde de tanımlanabilir. Şifre çözme ise şifrelenmiş bilgilerin orijinal metne dönüştürülmesidir. Şifreleme, veriler için bir şema oluşturarak çalışır. Bu şema daha sonra belirli anahtar kelimeleri ve kavramları o belirli mesajla ilişkilendirir. Ticari uygulamalarda, devlet işlerinde, askeri işlerde ve dijital ortamların aktarımında güvenli bilgi akışının sağlanması için şifreleme önemlidir.

İnsanlar eski çağlardan itibaren sayılara ve gösterimlerine ihtiyaç duymuşlardır. Kemiklerin üzerine atılan çentikler insanlığın ilk zamanlarından itibaren sayılara ihtiyaç duyduğun bir kanıtıdır. Günümüzde Hint-Arap sayı sisteminde geçen on tabanlı sayı sistemini kullanmaktayız. Bilgisayarlar ikilik taban kullanmaktadır. Şifreleme bilimi matematiksel yapılar ve dönüşümleri sıklıkla kullanır. Projemizde faktöriyel sayılar kullanılarak yeni bir sayı tabanı ve bu tabanın kullanıldığı şifreleme algoritması oluşturulacaktır. Ayrıca oluşturulan algoritmaya ait yazılım kodları ve uygulama örnekleri verilecektir.

Anahtar kelimeler: Şifreleme , Sayı Sistemleri, Taban Aritmetiği, Faktöriyel, Kodlama

Amaç

Projemizde faktöriyel sayıların yer aldığı yeni bir sayı tabanı kullanılarak şifreleme algoritması oluşturulacaktır. Örnek kelimelerin ASCII tablosundaki değerleri faktöriyel tabanda bulunacak ve yeni değerler şifreleme algoritmamızda kullanılacaktır. Ayrıca anahtarlarda faktöriyel tabanda dönüştürülerek kullanılacaktır. Oluşturduğumuz sayı tabanının ve şifreleme algoritmasının C++ yazılımında kodları verilecektir. Oluşturduğumuz algoritmayla şifrelenmiş kelimelerin deşifreleri ve kodları verilen yazılım sonuçları ile karşılaştırılacaktır.

Giriş

Eski zamanlardan itibaren insanlar sayıları ifade etmek için türlü simgeler kullanılmıştır. Matematikte sayıları yazma ve adlandırmaya sayıtlama sistemi denir. Bazen kullandığımız sayıtlama sistemi, aritmetik işlemleri kolaylaştırabileceği gibi zorlaştırabilir. Günlük hayatımızda Hint-Arap sayı sisteminden referans alan on tabanlı sayı sistemini kullanırız. Binary (İkili) Sayı Sistemi, Gottfried Leibniz tarafından icat edilen ve yalnızca iki sayıdan (0 ve 1) oluşan bir sayı sistemidir. Bu sayı sistemi, her yerde kullanılan bilgisayar işlemci talimatları gibi verileri yazmak için kullanılan tüm ikili kodların temelidir.

Bilgi ve teknolojinin hızlı bir şekilde geliştiği çağımızda, güvenli iletişimde rekabet artmış; bilgi aktarımı konusunda gizlilik ve güvenlik son derece önem kazanmıştır. Bu kapsamda bilginin iletiminde gizliliğin sağlanmasını amaçlayan şifreleme hızla gelişerek yaygınlaşmıştır. Şifre bilimi, kriptoloji ve kripto analiz olarak iki ana bölüme ayrılır. Gerçek metnin şifreli metne dönüştürülmesi için yapılan işlemler, oluşturulan sistemler, fonksiyonlar ve algoritmalar ile yani şifreleme ile ilgilenen çalışma alanına kriptografi adı verilir. Kripto analiz ise şifreli mesajları okumaya çalışmaktır (Özdemir, Erdoğan,2011).

Bilgi güvenliği için geliştirilen sistemler içerisinde şifreleme ve şifreleme algoritmaları önemli bir işleve sahiptir. Ticari ilişkilerde, devlet işlerinde, askeri işlerde ve personel ilişkilerinde güvenli bilgi akışının sağlanması için şifreleme önemlidir. Günlük yaşamın temel gereksinimleri olan güvenlik ve gizliliğin sağlanmasında vazgeçilmez bir unsur olan şifreleme ve şifreleme algoritmalarının, matematiksel bir temeli vardır. Şifreleme algoritmalarının matematiksel modellemelerinde bir çok matematik kavram ve konusundan yararlanılır. Modüler aritmetik, asal sayılar, fonksiyonlar, obeb örnek olarak gösterilebilir.

Şifreleme eski tarihlere dayanmaktadır. Roma İmparatoru Julius Caesar, Sezar şifreleme olarak bilinen en eski ve simetrik anahtar şifrelemenin klasik bir örneği olan basit bir yerine koyma şifrelemesini kullanmıştır (Stallings, 1998). Bu yöntemde alfabedeki her bir harf 3 sonraki harf ile şifrelenir. Bu şifre yönteminde her harf kendinden sonraki 3. Harfe dönüştürülmüştür. Şifreleme ve şifre çözme işlemleri yapılırken açık metin ile birlikte bir de anahtar kullanılmaktadır. Bir metni şifrelerken kullanılan değiştirme veya dönüştürme metodu anahtardır ve açık metin bu anahtardan yararlanılarak şifrelenmektedir. Simetrik anahtar kriptografisi (ya da simetrik şifreleme) mesajların hem şifrelenmesi hem de çözülmesinde aynı anahtarın kullanıldığı bir şifreleme düzenidir. Bu tip bir bilgi kodlama yöntemi geçtiğimiz on yıl içinde devletler ve ordular arasındaki gizli iletişimi sağlamak için sıklıkla kullanılmıştır. Günümüzde ise simetrik anahtar algoritmaları çeşitli bilgisayar sistemlerinde veri güvenliğini arttırmak için geniş çapta uygulanmaktadır.

10 Tabanlı Sayıtlama Dizgesi: Günlük yaşamda çoğunlukla sayıları 10 tabanı ile temsil ederiz. İnsanlar, parmak sayarak aritmetik yaptığı için, Onlu (10 tabanlı) Sayıtlama Dizgesini kullanmaya alışıktır. Bu taban 0,1,2,3,4,5,6,7,8,9 elamanlarıyla oluşur. Örneğin 4578 sayısını düşünelim. Bu bir simgedir ve aşağıdaki toplamı (10 tabanına göre açılımını) temsil eder:

$$4578 = 4x10^3 + 5x10^2 + 7x10^1 + 8x10^0 = 4000 + 500 + 70 + 8$$

2 Tabanlı Sayıtlama Dizgesi: Sayısal bilgisayarlar ikili sayıtlama dizgesini kullanır. İkilik taban 0 ile 1 elamanlarıyla oluşur. Bu tabandaki hanelere İngilizce'de bit denilir. Bu sözcük, Binary digIT deyiminden türetilmiştir. Bilgisayarın elektronik devrelerinde 0 bit'i alçak voltaj, 1 bit'i ise yüksek voltaj ile temsil edilir.

Tanım:

$a_{0,}a_1 \dots a_{n-1,}a_n$ kat sayılar ve t tabanı göstermek üzere; (a doğal sayı)

$$a_n t^n + a_{n-1}t^{n-1} + \cdots + a_2 t^2 + a_1 t^1 + a_0 \text{.....(1)}$$

biçiminde yazılabilir. Bu durumda

$$a = (a_n a_{n-1} \dots a_2 a_1 a_0)_t \text{.....(2)}$$

(1) ifadesine, t tabanında yazılmış a sayısının çözümlenmiş biçimi denir.

(2) ifadesinde, rakamların bulunduğu yerlere basamak; bir rakamın, bulunduğu basamaktaki değerine basamak değeri adı verilir.

Tanım :
faktöriyel n : n! = n × (n − 1) × ... × 3 × 2 × 1 olmak üzere

Yöntem

Bu bölümde faktöriyel tabanda sayıların nasıl yazıldığını ve şifreleme algoritmamızın nasıl uygulanacağından bahsedilecektir.

Tanım :(Faktöriyel Taban) t tabanını faktöriyel sayılar olarak tanımlayalım. Taban aritmetiğinin tanımı gereği faktöriyel tabanında yazacağımız sayılar kuvvetlerin yerine sırasıyla

…, 5 !, 4 !, 3 !, 2 !, 1 !.

şeklinde tanımlanabilir. Her sütunda izin verilen rakamlar sütun numarasına bağlıdır: ilk sütun (1!) 0,1; ikinci sütun (2!) 0,1 veya 2 katsayılarına sahip olabilir ve genel olarak n . faktöriyel sütun, n sütun numarasına kadar 0,1 içerebilir.

Faktöriyel tabanında bazı sayıları yazımı verilmiştir.

$1 = 1\times1! = (1)_!$
$6 = 1\times3! = (1,0,0)_!$
$8 = 1\times3! + 1\times2! = (1,1,0)_!$
$9 = 1\times3! + 1\times2! + 1\times1! = (1,1,1)_!$
$10 = 1\times3! + 2\times2! = (1,2,0)_!$
$24 = 1\times4! = (1,0,0,0)_!$

Tablo -1 de İlk 15 asal sayının faktöriyel tabanındaki gösterimleri verilmiştir.

$2=(1\ ,0\ )_!$
$3=(1\ ,1\ )_!$
$5=(2\ ,1\ )_!$
$7=(1\ ,0\ ,1\ )_!$
$11=(1\ ,2\ ,1\ )_!$
$13=(2\ ,0\ ,1\ )_!$
$17=(2\ ,2\ ,1\ )_!$
$19=(3\ ,0\ ,1\ )_!$
$23=(3\ ,2\ ,1\ )_!$
$29=(1\ ,0\ ,2\ ,1\ )_!$
$31=(1\ ,1\ ,0\ ,1\ )_!$
$37=(1\ ,2\ ,0\ ,1\ )_!$
$39=(1\ ,2\ ,1\ ,1\ )_!$
$41=(1\ ,2\ ,2\ ,1\ )_!$
$47=(1\ ,3\ ,2\ ,1\ )_!$

Tablo-1: İlk 15 Asal Sayının Faktöriyel Tabanda Gösterimi

Herhangi bir sayıyı faktöriyel tabanına çevirmek için sırasıyla 2,3,4,5… sayılarına bölünür. Örneğin 71 sayısını faktöriyel tabanında yazalım

71/n			
2	34	kalan = 1	71 = 2 × 34 + 3
3	11	kalan = 1	34 = 3 × 11 + 1
4	2	kalan = 3	11 = 4 × 2 + 3
5	0	kalan = 2	2 = 5 × 0 + 2

Tablo-2: Onluk Tabanda Verilen Bir Sayının faktöriyel Tabanda Yazımı

Bu durumda oluşan sayı , $71 = (2313)_!$ olarak faktöriyel tabanında yazılır.

n basamaklı faktöriyel tabanında verilen bir sayıyı onluk tabanda yazmak için basamak değerinin faktöriyeli ile sayılar çarpılarak toplanır.

$$(a_n, a_{n-1}, \dots, a_2, a_1)_! = a_n n! + a_{n-1}(n-1)! + \cdots + a_2 2! + a_1 1! \quad \text{.....(1)}$$

Örneğin, yukarıdaki örnekle aynı sayıları kullanarak: (2311)! Sayısını onluk tabanda yazalım.

$$\underset{4!}{(2}\ ,\ \underset{3!}{3}\ ,\ \underset{2!}{1}\ ,\ \underset{1!}{3}\ \underset{!}{)} = 2.4! + 3.3! + 1.2! + 3.1! = 48 + 18 + 2 + 3 = 71$$

olarak bulunur. Alan yazında faktöriyel tabanın kullanılan bir taban sistemi olmadığı görülmüştür.

Şifreleme Yöntemi

Şifreleme yöntemimizde şifrelenecek metnin yanında 3 tane kapalı anahtar yer alacaktır. 1.anahtar olarak 29,31,37,41,43,47,53,59,61,67,71,73,79,83 ve 89 olmak üzere toplam 15 asal sayıdan, şifrelenecek kelimenin harf sayısı kadarı kullanılacaktır çünkü 29'dan 89'a kadar asalları faktöriyel tabana çevirdiğimizde 0,1,2 veya 3 rakamlarından oluşan 4 basamaklı sayılar oluşacaktır, şifreleme yöntemimizde 4'lük sayı sisteminde de işlem yapılacağı için faktöriyel taban sayılarımızın 0,1,2 veya 3 rakamlarından oluşması gerekmektedir. En fazla 4 basamaklı sayıların oluşmasını isteme sebebimiz işlem kolaylığı açısındandır.

2.anahtar olarak rastgele seçilmiş karakter dizisi kullanılacaktır, bu karakter dizisi kelime veya sayı da olabilir. Rastgele anahtar seçiminin şifreleme algoritmasının gücünü arttırdığı bilinmektedir, ayrıca anahtar uzunluğu arttıkça şifre kırıcılara karşı, algoritmanın gücü artmaktadır.

3.anahtar ise şifreleme yönteminin son aşamasında oluşacak olan bölüm dizisi olacaktır.

Şifrelenecek metnin her bir kelimesi ayrı şifrelenecektir fakat kelimenin harf sayısı 15'den fazla ise 15'i aşan her harf grubu ayrı bir kelime gibi şifrelenecektir. Örneğin 18 harfli bir kelime 15+3 şeklinde bölünüp; ilk 15 harf birlikte, son 3 harf birlikte şifrelenecektir.

Şifrelenecek metnin(açık metin) harfleri ve rastgele oluşturulan anahtarın karakterleri, ASCII tablosundan sayı karşılıklarına dönüştürülerek şifrelemeye başlanır ve bu sayılar faktöriyel tabana çevrilir. Bu aşamada açık metin, büyük harflerden oluşacağı için başlangıçta ASCII tablosundaki 65-90 arasındaki sayılar kullanılacaktır.

Devamında açık metnin harf sayısı kadar 29 dan büyük ve eşit asal sayılar faktöriyel tabana çevrilir.

Elde edilen 3 grup faktöriyel taban sayıları,4'lük sayı sistemine uygun olduğu için , sırasıyla alt alta yazılarak 4'lük tabanda toplama işlemi yapılır, bu aşama tam olarak şifreleme dönüşümünün gerçekleştiği kısımdır. Ancak buradaki toplama işlemi standart toplamadan biraz farklı yapılacaktır.(Literatürde bu yöntemi kullanan şifreleme algoritmaları mevcuttur. Ör: Vernam Şifreleme).Toplama yapılırken ; toplam sayı 4 den küçükse aynen yazılır,4 den büyük veya eşitse 4'e bölümünden kalan yazılır, elde bir üst basamağa verilmez yani ihmal edilir. Bu toplama işlemine alternatif toplama diyelim.

Son aşamada, alternatif toplama işlemi sonucu elde edilen en çok 4 basamaklı sayılara (mod93) işlemi uygulanır ve kalan sayılara 33 eklenir. Burada şifreli metnin, ASCII tablosunun 33-126 arasında bulunan karakterlerden oluşması istendiği için (mod93) işlemi uygulanıp 33 eklenmiştir. Elde edilen sayılar ASCII tablosundan karakterlere dönüştürülür ve şifreli metin elde edilir. Bölüm sayıları ise 3.anahtar olarak ilk 2 anahtar ve şifreli metin ile birlikte alıcıya gönderilir.

Deşifre Yöntemi:

- Alıcıda şifreli metin ve 3 anahtar grubu bulunmaktadır. Şifreli metnin karakterleri ASCII tablosundan sayılara dönüştürülerek deşifreye başlanır. Bu sayıların her birinden 33 çıkarılır, elde edilen sayıları kalan, bölüm dizisi anahtarındaki sayıları sırasıyla bölüm ve 93 sayısını bölen kabul ederek bölünen sayılara ulaşılır.

- Rastgele anahtar dizisindeki karakterler, ASCII tablosundan sayılara dönüştürülür ve bu sayılarla birlikte, şifreli metindeki karakter sayısı kadar 29 ve 29 dan büyük asallar faktöriyel tabana çevrilir.
- Bu adımda 2 aşamalı, alternatif bir çıkarma işlemi yapılır.

Şifrelenmiş metinden elde ettiğimiz bölünen sayı gruplarından, rastgele anahtar dizisinin faktöriyel taban karşılıklarını 4'lük tabanda çıkarırız ancak buradaki çıkarma işlemi de şifrelemedeki toplama işlemi gibi yapılır; örneğin 12-23 işleminde, 2-3 yapılışı sırasında,2, komşusu 1'den dörtlük alır ve

6-3=3 olur fakat 1'in değeri azaltılmadan işleme devam edilir. Böylece hem işlem tersine dönmüş olur hem de görünüşte küçük sayıdan, büyük sayı çıkarıldığında sonuç negatif işaretli çıkmamış olur.

- Bir önceki maddede işlem sonucundan elde edilen sayılardan yine sırasıyla asal anahtarların faktöriyel taban karşılıkları bir önceki adımdaki gibi çıkarılır. Rastgele ve asal anahtar dizini önce toplayıp sonucu şifreli metinden elde edilen bölünen sayılardan çıkarmak bizi istediğimiz sonuca götürmemektedir.
- Son işlemle elde edilen sayılara faktöriyel tabandan onluk tabana çevirme işlemi uygulanır ve onluk sistemdeki bu sayılar ASCII tablosundan harflere dönüştürülür ve açık metne ulaşılır.

Şifreleme Algoritmasının Yazılım Basamakları

Gerekli kütüphaneleri ekleyip kullanacağımız fonksiyonları yazalım.

```c
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>

//faktoriyel alma fonksiyonu
int faktoriyel(int sayi) {
    int faktoriyel = 1;
    for (int i = 2; i <= sayi; i++) {
        faktoriyel *= i;
    }
    return faktoriyel;

}
```

Şifreleme yaparken sayıların ASCII kodlarını kullanacağımız için 3. anahtarın sayılarını ASCII kodlarına dönüştüren fonksiyonu yazalım.

```c
void rakamAscii(int dizi[], int Ascii[]) {


    for (int i = 0; i < 8; i++)
    {
        char a = dizi[i];


        Ascii[i] = a + '0';

    }

}
```

Girilen kelimenin harflerinin ASCII kodlarını bulan fonksiyonu yazalım.

```
//kelimenin harflerinin ASCII kodlarını bulup diziye atan fonksiyon.
void Ascii_Cevirme(char kelime[], int Asciikarsiligi[]) {

    int i = 0;

    while (kelime[i] != '\0') {


        Asciikarsiligi[i] = int(kelime[i]);

       i++;

    }


}
```

Sayıları onluk tabandan faktöriyel tabana çevirirken sırasıyla 2,3,4,5... sayılarına bölüp bölümün tam kısmı 0 olana kadar devam edelim ve kalanları tutalım. Kalanları tersten yazdığımızda faktöriyel tabandaki sayıyı elde ederiz.(Tablo-2)

```
int onluk_taban_faktoryel_cevirme(int sayi) {
    int  a = 2, i = 0;
    int kalan[30];
    int sonuc = 0;
    //sayinin 2,3,4,5... sayılarıyla bölümünün tam kısmı 0'dan büyük olduğu sürece devam edecek.
    while (sayi > 0) {
        //bölümün kalanlarını bir dizide tutalım

        kalan[i] = sayi % a;

        //kalanları tersten yazınca faktoriyel tabandaki sayıyı elde ederiz.
        //bunu hafızda dizi olarak değil de sayı olarak tutabilmek için kalanları 10'un artan üsleriyle çarpıp topluyoruz.

        sonuc += kalan[i] * pow(_x:10, _y:i);

        //sayiyi 2'ye bölüp her döngüde böleni arttıralım. Döngü bölümün tam kısmı sıfır olana kadar devam eder

        sayi /= a;
        a++;
        i++;

    }
    return sonuc;
}
```

Şifreleme yaparken harfin ASCII kodunun faktöriyel tabandaki değeri, asal sayının faktöriyel tabandaki değeri ve rastgele sayı dizisinin ASCII kodunun faktöriyel tabandaki değerini 4'lük tabanda eldesiz bir biçimde toplayan fonksiyonu yazalım.

```
//4'lük tabanda eldesiz toplama fonksiyonu
int toplama(int a, int b, int c) {

    int x = 0;
    int sonuc[4];
    int toplam = a + b + c;
    //bize gelen 3 sayıyı onluk tabanda toplayalım
    //sonra sayıyı basamaklarına ayırıp her basamağı dizinin bir elemanına atalım.
    for (int i = 0;i < 4;i++) {
        sonuc[3 - i] = toplam % 10;
        toplam /= 10;
    }
    //sayının basamaklarında 4'ten büyük bir rakam varsa 4 ile modunu alalım.
    for (int i = 0;i < 4;i++) {
        if (sonuc[i] >= 4) {
            sonuc[i] = sonuc[i] % 4;


        }
        //sayının basamaklarını değiştirdikten sonra 10'un azalan üsleriyle çarpıp sayıyı elde edelim.
        x += sonuc[i] * pow(10, 3 - i);
    }
    return x;
}
```

Asal sayı dizisini ve rastgele sayı dizisini tanımlayalım ve gerekli işlemleri yapalım.

```c
int main()
{
    int a, b, c, d, e, i = 0;
    //kullandığımız rastgele sayı dizisi.
    int anahtar[] = { 2,9,1,0,1,9,2,3 };
    int bolum[20];
    //kullandığımız asal sayılar dizisi.
    int asallar[] = { 29,31,37,41,43,47,53,59,61,67,71,73,79,83,89 };
    int anahtarascii[20];
    char harf[20];
    gets_s(harf);
    int Asciikarsiligi[20];
    //girilen kelimenin ASCII kodlarını bulup Asciikarsiligi dizisne atar.
    Ascii_Cevirme(harf, Asciikarsiligi);
    //rastgele sayı dizisinin ASCII karşılıklarını bulur.
    rakamAscii(anahtar, anahtarascii);
```

Şifrelemeyi yapalım.

```c
while (harf[i] != '\0')
{
    //harflerin ASCII karşılığını faktoriyel tabana çevirelim.
    a = onluk_taban_faktoryel_cevirme(Asciikarsiligi[i]);
    //asal sayıları faktoriyel tabana çevirelim.15 elemanlı olduğu için 15 ile modunu alalım.
    b = onluk_taban_faktoryel_cevirme(asallar[i % 15]);
    //rastgele sayı dizisinin elemanlarının ASCII karşılığını faktoriyel tabana çevirelim.
    c = onluk_taban_faktoryel_cevirme(anahtarascii[i%8]);

    //elde ettiğimiz sayılara 4'lük tabanda eldesiz toplama işlemini uygulayalım.
    d = toplama(a, b, c);

    //deşifre için sayının 93'e kaç kere bölündüğünü bulalım.
    e = d / 93;
    bolum[i] = e;

    //değerlerin istediğimiz aralıkta gelmesi için modunu alıp 33 ekleyelim.
    sifre[i] = d % 93 + 33;

    i++;
}
printf("Sifrelenmis hali: ");
for (int i = 0; i < strlen(harf); i++)
{
    printf("%c", sifre[i]);
}
printf("\n");
int k = 0;
printf("Bolumler: ");
while(k < i)
{
    printf("%d ,", bolum[k]);
    k++;
}
```

Deşifre Algoritması:

Önceden kullandığımız ve tekrar işimize yarayacak olan fonksiyonları çağırıp ek olarak faktöriyel tabandan onluk tabana çevirme ve sayıları dörtlük tabanda çıkartma algoritmasını yazalım.

```
//sayıyı faktoriyel tabandan 10'luk tabana çeviren algoritma.
int faktoryel_onluk_taban_cevirme(int sayi) {
    int i = 1;
    int a = 0, x;
    int faktoryel_tabandaki_sayi = 0;
    int k = 0;
    int sayibasamaklari[10];

    //sayının kaç basamaklı olduğunu bulalım.
    while (sayi / i > 0) {
        i *= 10;
        k++;
    }

    //sayının basamaklarındaki rakamları bulup bir diziye atalım.
    for (int i = 1; i <= k; i++)
    {
        sayibasamaklari[k - i] = sayi % 10;
        sayi /= 10;
    }

    //faktoriyel fonksiyonunu kullanarak sayının rakamlarını sırasıyla 1!,2!,3!.. ile çarpalım ve sayıyı elde edelim.
    for (int i = 0; i < k; i++) {
        x = sayibasamaklari[i];
        faktoryel_tabandaki_sayi += x * faktoriyel(k - i);
    }

    return faktoryel_tabandaki_sayi;
}
```

Çıkartma algoritması. Önceki toplama fonksiyonunu kullandık. Ama bu sefer ikinci sayıyı negatif yolladık. Yani – ile toplama işlemi yapmış olduk.

```
//çıkartma algoritması. 1. sayı pozitif 2. sayı negatif değerlidir.
int cikarma(int a, int b) {
    int x = 0;
    int sonuc[4];
    int c = 4444;
    //bize verilen 2 sayıyı topluyoruz.Yani - ile toplama işlemi yapmış oluyoruz.
    //sayının negatif gelmemesi için 4444 ekledik.
    //Zaten devamında bir rakamı 4'ü geçerse 4 ile modunu aldık.Bu sayade doğru bir şekilde çıkrtma işlemini yapmış olduk
    int toplam = a + b + c;

    //sayının basamklarını diziye atalım.
    for (int i = 0;i < 4;i++) {
        sonuc[3 - i] = toplam % 10;
        toplam /= 10;
    }

    //sayının basamklarında 4'ten büyük bir değer varsa 4 ile modunu alalım.
    for (int i = 0;i < 4;i++) {
        if (sonuc[i] >= 4) {
            sonuc[i] = sonuc[i] % 4;

        }
        x += sonuc[i] * pow(10, 3 - i);
    }
    return x;
}
```

Deşifre için gereken bölümü ,asalları ve rastgele sayı dizisini tanımlayalım.

```
int main()
{

    int b,x,i=0;
    //bize verilen anahtar.
    int anahtar[] = { 2,9,1,0,1,9,2,3 };
    int anahatarascii[20];
    //asal sayılar.
    int asallar[] = { 29,31,37,41,43,47,53,59,61,67,71,73,79,83,89 };
    //Önceki sayının 93'e kaç kere bölündüğü.
    int bolum[] = { 21 ,12 ,24 ,23 ,12 ,14 ,23 ,12 ,22 ,24 ,32 ,22 ,2 ,2 ,33 ,22 };
    char sifre[20];
    int a[20];

    //şifrelenmiş halini alalım.
    gets_s(sifre);

    //rsatgele sayı dizisinin ASCII kodlarını bulup bir diziye atalım.
    rakamAscii(anahtar, anahatarascii);
```

Deşifre işlemini yapalım.

```
while (sifre[i] != '\0') {
    //kelimenin harflerinin int değerini yani ASCII kodunu a dizisine atalım.
    a[i] = int(sifre[i]);

    //Gelen değerden 33 çıkarıp 93'e kaç kere bölündüğüyle çarpalım.
    a[i] = a[i] - 33;
    a[i] = a[i] + bolum[i] * 93;

    //İlk değere ulaşmak için değerleri birbirinden çıkartalım.
    x = cikarma(a[i], -onluk_taban_faktoryel_cevirme(asallar[i%15]));
    x = cikarma(x, -onluk_taban_faktoryel_cevirme(anahatarascii[i%8]));

    //en son gelen değeri onluk tabana çevirelim
    b = faktoryel_onluk_taban_cevirme(x);

    //gelen değeri char biçiminde yazdıralım
    printf("%c", b);
    i++;
}
```

Proje İş-Zaman Çizelgesi

	AYLAR									
İşin Tanımı	**Nisan**	**Mayıs**	**Haziran**	**Temmuz**	**Ağustos**	**Eylül**	**Ekim**	**Kasım**	**Aralık**	**Ocak**
Literatür Taraması					**x**	**x**	**x**	**x**	**x**	**x**
Verilerin Toplanması ve Analizi						**x**	**x**	**x**	**x**	**x**
Proje Raporu Yazımı								**x**	**x**	**x**

Bulgular

Bu bölümde yöntemde verilen algoritmaya ait metnin şifreleme ve deşifre işlemlerine ait örneklere yer verilecektir.

Örnek Uygulama

Cumhuriyetimizin kuruluşunun 100. yılı onuruna CUMHURIYET kelimesini şifreleyelim. Anahtar olarak ta 29 Ekim 1923 tarihine atfen 29101923 sayısını kullanalım.

CUMHURIYET kelimesini şifreleyelim.

- 1.Anahtar, açık metin 10 harften oluştuğu için 29 ve 29 dan büyük 10 tane asal sayı 2.Anahtarımız 29101923 olsun. Anahtarımız 8 karakter, açık metin 10 harfli olduğu için aradaki fark kadar sol baştan anahtarın karakterleri tekrar edilerek anahtar uzatılır 3.Anahtarımız ise şifrelemenin son aşamasında oluşacak olan bölüm dizisi olacaktır.

C:67=(2301)!	29=(1021)!	2:ASCII=50=(2010)!
U=85(3201)!	31=(1101)!	9:ASCII=57=(2111)!
M=77(3021)!	37=(1201)!	1:ASCII=49=(2001)!
H=72(3000)!	41=(1221)!	0:ASCII=48=(2000)!
U=85(3201)!	43=(1301)!	1:ASCII=49=(2001)!
R=82(3120)!	47=(1321)!	9:ASCII=57=(2111)!

I=73(3001)!	53=(2021)!	2:ASCII=50=(2010)!
Y=89(3221)!	59=(2121)!	3:ASCII=51=(2011)!
E=69(2311)!	61=(2201)!	2:ASCII=50=(2010)!
T=84(3200)!	67=(2301)!	9:ASCII=57=(2111)!

- **2. Adım**

2301	3201	3021	3000	3201	3120	3001	3221	2311	3200
1021	1101	1201	1221	1301	1321	2021	2121	2201	2301
2010	2111	2001	2000	2001	2111	2010	2011	2010	2111

1332	2013	2223	2221	2103	2112	3032	3313	2122	3212

Elde edilen sayıların (mod93) ü alınır, kalan sayılara 33 eklenir ve ASCII tablosundan karakter karşılıklarına dönüştürülüp ; şifrelenmiş metin elde edilir.

1332≡30 (mod93)	Bölüm:14 →	30+33=63 →	?
2013≡60 (mod93)	Bölüm:21 →	60+33=93 →	]
2223≡84 (mod93)	Bölüm:23 →	84+33=117 →	u
2221≡82 (mod93)	Bölüm:23 →	82+33=115 →	s
2103≡57 (mod93)	Bölüm:22 →	57+33=90 →	Z
2112≡66 (mod93)	Bölüm:22 →	66+33=99 →	c
3032≡56 (mod93)	Bölüm:32 →	56+33=89 →	Y
3313≡58 (mod93)	Bölüm:35 →	58+33=91 →	[
2122≡76 (mod93)	Bölüm:22 →	76+33=109 →	m
3212≡50 (mod93)	Bölüm:34 →	50+33=83 →	S

Sonuç olarak metnimiz

CUMHURIYET →?]usZcY[mS olacak şekilde şifreli metne dönüşür.

Örnek Deşifre

Alıcıda ?]usZcY[mS şifreli metni , {29 ,31,37,41,43,47,53,59,61,67} asal anahtarı ve {14,21,23,23,22,32,35,22,34} bölüm anahtarı mevcuttur.

Alıcı şifrelenmiş metnin karakterlerini, ASCII tablosundan sayılara çevirerek deşifre işlemine başlar.

?	**]**	**u**	**s**	**Z**	**c**	**Y**	**[**	**m**	**S**
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
63	93	117	115	90	99	89	91	109	83

Öncelikli olarak bu sayılardan 33 çıkartılır ve çıkan sonuçlar kalan, 93 ü bölen ve 3. Anahtar dizindeki sayıları bölünen kabul ederek sayılara ulaşılır.

63-33=30	→	**14.93+30=1332**
93-33=60	→	21.93+60=2013

117-33=84	→	23.93+84=2223
115-33=82	→	23.93+82=2221
90-33=57	→	22.93+57=2103
99-33=66	→	22.93+66=2112
89-33=56	→	32.93+56=3032
91-33=58	→	35.93+58=3313
109-33=76	→	22.93+76=2122
83-33=50	→	34.93+50=3212

Şimdi ;(29101923) anahtarının her bir karakterini ASCII tablosunda sayı karşılıklarına dönüştürüp ve bulduğumuz değerleri faktöriyel tabana çevirelim.

2:ASCII=50=(2010)!
9:ASCII=57=(2111)!
1:ASCII=49=(2001)!
0:ASCII=48=(2000)!
1:ASCII=49=(2001)!
9:ASCII=57=(2111)!
2:ASCII=50=(2010)!
3:ASCII=51=(2011)!
2:ASCII=50=(2010)!
9:ASCII=57=(2111)!

Şimdi 2 aşamalı olarak şifreli metnin faktöriyel taban karşılıkları sırasıyla 4 lük tabanda çıkarılır, ancak buradaki çıkarma işlemi şifreleme işlemin sırasında yapılan toplama gibi yapılır.

1332	**2013**	**2223**	**2221**	**2103**	**2112**	**3032**	**3313**	**2122**	**3212**
2010	2111	2001	2000	2001	2111	2010	2011	2010	2111

3322	0302	0222	0221	0102	0001	1022	1302	0112	1101

Oluşan sayılardan asal anahtarın faktöriyel taban karşılıkları yine 4 lük tabanda çıkarılır.

3322	**0302**	**0222**	**0221**	**0102**	**0001**	**1022**	**1302**	**0112**	**1101**
1021	1101	1201	1221	1301	1321	2021	2121	2201	2301

2301	3201	3021	3000	3201	3120	3001	3221	2311	3200

Bu adımda çıkarma işleminin iki aşamalı yapılması çok önemlidir fakat çıkarma sırası önemsizdir. Şifreli metnin faktöriyel taban karşılıklarından önce hangi anahtarın faktöriyel tabandan ilk olarak çıkarıldığı sonucu değiştirmez. Ancak çıkarmayı tek aşamada yapmaya çalışırsak sonuç değişecektir.

Son aşamada elde ettiğimiz sayıları faktöriyel tabandan 10 luk tabana çevirir; sayıları ASCII tablosundan karakterlere dönüştürerek açık metne ulaşırız.

(2301)!	= 48+18+1=67	→	C
(3201)!	**=72+12+1=85**	→	**U**
(3021)!	**=72+4+1=77**	→	**M**
(3000)!	**=72**	→	**H**
(3201)!	**=72+12+1=85**	→	**U**
(3120)!	**=72+6+4=82**	→	**R**
(3001)!	**=72+1=73**	→	**I**
(3221)!	**72+12+4+1=89**	→	**Y**
(2311)!	**48+18+2+1=69**	→	**E**
(3200)!	**72+12=84**	→	**T**

Böylece deşifre tamamlanmış olur.

Uygulama Örneği Yazılım Çıktıları

Şifreleme ve deşifre algoritmalarının çıktıları.

```
Sifrelenecek kelimeyi giriniz: CUMHURIYET
Sifrelenmis hali: ?]usZcY[mS
Bolumler: 14 ,21 ,23 ,23 ,22 ,22 ,32 ,35 ,22 ,34 ,

...Program finished with exit code 0
Press ENTER to exit console.
```

```
Sifreyi giriniz: ?]usZcY[mS
Cozulmus hali: CUMHURIYET

...Program finished with exit code 0
Press ENTER to exit console.
```

Sonuç ve Tartışma

Bu bölümde proje çalışması ile elde edilen bulgular araştırma sorusuna veya problemine uygun olarak yorumlanır.

Çalışmamızda faktöriyel taban kullanarak bir şifreleme algoritması kullanarak C++ dilinde bir yazılım geliştirdik. Faktoriyel taban matematikte kullanım alanı fazla olmaması , birçok kişi tarafından bilinmemesi ve şifrelemeyi gerçekleştirirken 4'lük tabanda eldesiz toplama yapılması algoritmayı daha da karmaşık hale getirmektedir. Bu yüzden bu algoritma kullanılarak yapılan şifrelemenin deşifresi zorlaşmaktadır. Günümüzde birçok alanda kullanılan şifreleme algoritmalarının deşifre edilmesinin zor olması istenilen bir durum olduğu için yazdığımız bu algoritma güvenli bir şifreleme için kullanılabilir.

Öneriler

Bu bölümde benzer çalışmalar yapacak olanlara yol göstermesi bakımından öneriler varsa belirtilir.

➢Tasarladığımız algoritma şu an için sadece büyük harfler için çalışabilmektedir. Bu algoritmayı küçük harfler ve simgeler için de çalışabilir hale getirilebil

Kaynaklar

Özdemir, A. Ş.,& Erdoğan, F. (2011). Şifreleme etkinlikleriyle faktöriyel ve permütasyon konusunun öğretimi. *Batı Anadolu Eğitim Bilimleri Dergisi*, 2(3), 19-43.

Özdemir, F., & Özdemir, H. (2017). Matematik eğitiminde sayıların önemi: özel sayı ve sistemlerinin keşfedilmesi örneği.

Stallings, W. (1998). *Cryptography and network security:Priciples and practice*. New Jersey: Prentice Hall.

Yerlikaya, T. , Gençoğlu, H. , Emir, M. K. , Çankaya, M. & Buluş, E. (2011). Rsa Şifreleme Algoritması Ve Aritmetik Modül Uygulaması . İstanbul Aydın Üniversitesi Dergisi , 3 (9) , 95-104 . Retrieved from https://dergipark.org.tr/en/pub/iaud/issue/30054/324501

Katrancı, A. G. Y., & Özdemir, A. Ş. (2013). Rsa şifrelemesi yardımıyla modüler aritmetik konusunun pekiştirilmesi.

http://www.baskent.edu.tr/~tkaracay/etudio/ders/math/topology/odev/binary.html Erişim Tarihi:15.12.2022