PROJE RAPORU

PROJE ADI: E-Günce: Şifreli Günlük Yazılımı

1. Giriş

Kriptoloji, şifre bilimidir. Çeşitli iletilerin, yazıların belli bir sisteme göre şifrelenmesi, bu mesajların güvenli bir ortamda alıcıya iletilmesi ve iletilmiş mesajın deşifre edilmesidir. Geçmiş dönemlerde, örneğin 2. Dünya Savaşı sırasında, devletler birbirleriyle haberleşirken şifreli mesajları kullanmışlardır. Bu da bir kriptoloji uygulamasıdır. Günümüzde ise kriptoloji, bilgi güvenliği, ağ güvenliği, veri iletimi gibi farklı amaçlar için kullanılmaktadır. Bilgi güvenliği uzun yıllardır insanların önemli uğraşı alanlarından biri olmuştur. Elde olan veriyi şifrelemek, yabancılardan korumak, zamanı gelince de şifresini çözmek adına birçok çalışma yapılmıştır.

E-Günce projesinde de C# dilini ve çeşitli şifreleme tekniklerini kullanan, üst düzey güvenliğe sahip bir günlük yazılımı geliştirilmiştir. Günlük metinlerinin şifrelenmesi, Rijndael olarak da bilinen Advanced Encryption Standard algoritması ile sağlanmıştır.

1.1) Amaç

İnsanlar uzun yıllardır günlük yazmaktadırlar. İnsanlar günlüklerinde ruh hallerinden günlük yaşamlarına kadar birçok konuda yazı yazabilirler. Ancak günlüklerin kişiye özel olduğu unutulmamalıdır. Yani günlüğü yazan kişi aynı zamanda günlüğünü de yabancı ellerden koruyabilmelidir. E-Günce projesinde de yazılımsal yollardan koruma altına alınan bir günlük uygulamasının geliştirilmesi amaçlanmıştır. Günlüğü şifrelemek için kişiye özel bir anahtar da girilmesi gerekmektedir. Aynı şekilde şifrelenen günlüğün şifresini çözebilmek için de şifrelerken kullanılan anahtarın uygulamaya girilmesi gerekmektedir.

1.2) C#

C#, Microsoft'un geliştirmiş olduğu yeni nesil programlama dilidir. Yine Microsoft tarafından geliştirilmiş .NET Teknolojisi için geliştirilmiş dillerden biridir. Özellikle nesne yönelimli programlama kavramının gelişmesine katkıda bulunan en aktif programlama dillerinden biridir. Uygulama tamamen C# dili kullanarak geliştirilmiştir. C# dilinin kullanılma sebebi, dilin hem kolay kullanılabilir olmasından dolayı hem de birçok sistemle uyumlu çalışabilmesinden dolayıdır.

1.3) Visual Studio

Microsoft Visual Studio, Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır (IDE). Visual Studio, değişik programlama dillerini destekler, bu da kod editörü ve hata ayıklayıcısının neredeyse tüm programlama dillerini desteklemesini sağlamaktadır. Projedeki C# kodları Visual Studio 2017 Community sürümü ile geliştirilmiştir.

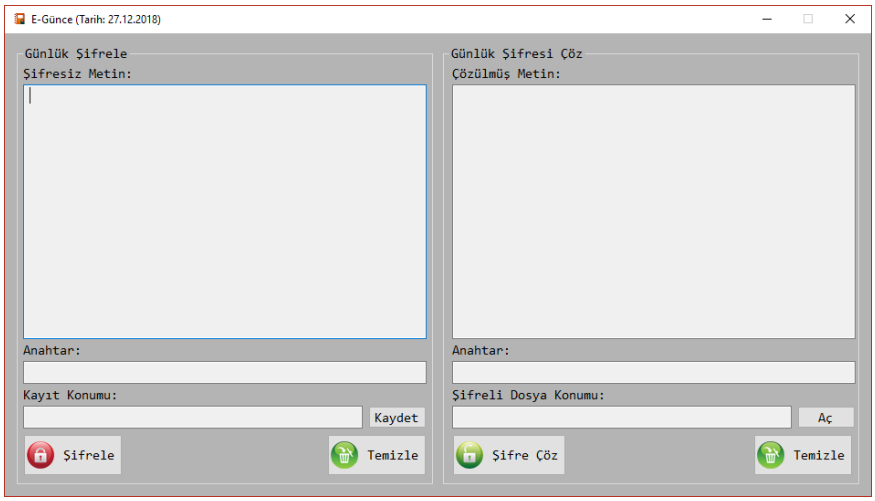
Ayrıca Microsoft, Visual Studio 2017 Community sürümünü öğrenciler ve araştırmacılar için ücretsiz sunmaktadır.

1.4) AES

AES (Gelişmiş Şifreleme Standardı), elektronik verinin şifrelenmesi için sunulan bir standarttır. Projede dosya şifreleme algoritması olarak kullanılmıştır. AES algoritması, 128 bit veri bloklarını 128, 192 veya 256 bit anahtar seçenekleri ile şifreleyen bir blok şifre algoritmasıdır. Anahtar uzunluğu bit sayıları arasındaki farklılık AES tur döngülerinin sayısını değiştirmektedir.

2. Yöntem

Proje, tek bir Windows formu üzerinden çalışmaktadır. Projede sadece C# dili kullanılmıştır. Takip eden bölümlerde sırasıyla şifreleme ve şifre çözme adımları teknik olarak izah edilecektir.



Görsel 1. E-Günce Şifreleme ve Şifre Çözme Ekranı

2.1) Şifreleme

Şifreleme işlemi içim ilk önce şifrelenecek günlük metninin "Şifresiz Metin" metin kutusuna yazılması, ardından şifre çözerken de kullanılacak olan anahtarın "Anahtar" kutusuna yazılması gerekmektedir. Son olarak şifreli dosyanın kaydedileceği konum da seçildikten sonra "Şifrele" butonu ile günlüğün şifrelenmesi tamamlanacaktır. Şifreleme adımları şu şekilde gerçekleşmektedir:

a) Metnin, Anahtarın ve Konumun Kullanıcı Tarafından Girilmesi

Görsel 1'de görüldüğü üzere, açılış ekranında "Şifresiz Metin" ve "Çözülmüş Metin" başlıklı iki metin kutusu bulunmaktadır. Kullanıcı, "Şifresiz Metin" kutusu aracılığıyla şifrelenmesini istediği metni uygulamaya gönderir.

Daha sonra "Anahtar" başlıklı metin kutusuna da şifrelemede ve şifre çözmede kullanılacak olan *gizli* anahtarı girer.

En son olarak da "Kaydet" butonuna basarak OpenFileDialog fonksiyonu yardımıyla şifrelenen metnin saklanması gereken yeri seçtikten sonra, "Şifrele" butonuna basar.

b) Girilen Metnin Şifrelenmesi

Uygulamada metni şifrelemek için "*sifreleFunc(string kaynak, string hedef, string anahtar)*" şeklinde bir fonksiyon oluşturulmuştur. Bu fonksiyonda sırasıyla "kaynak" değişkeni girilen şifresiz metin için, "hedef" değişkeni şifreli dosyanın konumu için, "anahtar" değişkeni ise gizli anahtar için kullanılmıştır.

Kullanıcı "Şifrele" butonuna bastığı an *sifrele_Click()* fonksiyonu devreye girer. Bu fonksiyonda da "*tarih*", "*temp*", "*tempAdres*", "*tarihMetin*", "*kullanici*" ve "*kisatarih*" isimli değişkenler bulunmaktadır.

Bu değişkenlerin görevleri şu şekildedir: "*tarih*" değişkeni, günlüğün kaydedildiği tarihi gün/ay/yıl/saat/dakika/saniye şekline getirip şifreli dosyanın başlığına ekler, "*temp*" değişkeni, sistemin geçici dosya dizininin adresini alır, "*tempAdres*" değişkeni, "*temp*" değişkeni ile "*tarih*" değişkenini bağlar, "*tarihMetin*" değişkeni, şu anki zamanı, string veri tipi şeklinde yazar, "*kullanici*" değişkeni, günlüğü yazan kullanıcının kullanıcı adını alır ve "*kisatarih*" değişkeni, gün, ay, yıl bilgileri ile saat, dakika, saniye bilgilerini bir araya getirir.

Eğer anahtar, şifresiz metin ve yeni şifreli metnin bulunacağı konum için kullanılan metin kutularında geçerli metinler varsa, *try* metodu ile dosyanın şifrelenmesine ve değişkenlerin değerlendirilmesine geçilir.

En nihayetinde, değişkenlerin değerleri atandıktan sonra, *sifreleFunc* fonksiyonu çalışır ve kullanıcının girdiği şifresiz metin ve anahtar, AES ile şifrelemeye girer. Bunun sonucunda kullanıcı tarafından seçilen konumda ismi gün/ay/yıl/saat/dakika/saniye formatında olan bir dosya oluşur. İşte bu dosya girilen günlüğün şifrelenmiş ve koruma altına alınmış halidir. Eğer süreçte herhangi bir sorun çıkarsa, *catch* yapısı ile "Bir sorun oluştu." şeklindeki hata mesajı kullanıcıya gösterilir.

```
private void sifrele_Click(object sender, EventArgs e)
{
    if (sifreleAnahtar.Text != "" && sifreleMetin.Text != "" && sifreleKonum.Text != "")
    {
        try
        {
            string tarih = DateTime.Now.ToString("ddMMyyyyHHmmss");
            string temp = System.IO.Path.GetTempPath();
            string tempAdres = temp + tarih;
            string tarihMetin = DateTime.Now.ToString("dd.MM.yyyy");
            string kullanici = System.Security.Principal.WindowsIdentity.GetCurrent().Name;
            File.WriteAllText(tempAdres, "Tarih:" + tarihMetin + "\r\n\n\n" + "Yazan:" + kullanici + "\r\n\n\n" + "Metin:" + sifreleMetin.Text);
            sifreleFunc(tempAdres, sifreleKonum.Text + "/" + tarih, sifreleAnahtar.Text);
            File.Delete(tempAdres);
            string kisatarih = DateTime.Now.ToString("dd.MM.yyyy") + " " + DateTime.Now.ToString("hh.mm.ss");
            MessageBox.Show(kisatarih + " tarihine ait günlük kaydı oluşturuldu.");
            sifreleMetin.Text = ""; sifreleAnahtar.Text = "";


        }
        catch
        {
            MessageBox.Show("Bir sorun oluştu.");
        }
    }
    else
    {
        MessageBox.Show("Tüm alanları doldurun.");
    }
    }
```

Görsel 2. sifrele_Click Fonksiyonu ve İşlevlerini Gösteren Kod Bloğu

2.2) Şifre Çözme

Şifre çözme adımında, şifrelemeden farklı olarak şifreli dosyaya da ihtiyaç duyulmaktadır. Şifreleme adımında oluşturulan şifreli dosya ve şifreleme adımında kullanılan anahtar yardımıyla şifreli günlük dosyasının şifresi çözülmektedir. Şifreli günlük dosyası ana ekrandaki "Şifreli Dosya Konumu" başlığı altındaki "Aç" butonu sayesinde seçildikten sonra yine **şifrelemede kullanılan** anahtar da "Anahtar" başlığı altındaki metin kutusuna girilir ve "Şifre Çöz" butonuna tıklanır. Şifre çözme adımları şu şekilde gerçekleşmektedir:

a) Metnin, Anahtarın ve Konumun Kullanıcı Tarafından Girilmesi

Ana ekrandaki "Günlük Şifresi Çöz" başlığı altında bulunan "Şifreli Dosya Konumu" kısmındaki "Aç" butonu ile şifreli günlük dosyası seçildikten sonra, "Anahtar" başlığı altındaki metin kutusuna da anahtar girilir ve "Şifre Çöz" butonuna tıklanır.

b) Girilen Şifreli Dosyanın Şifresinin Çözülmesi

Uygulamada metni şifrelemek için "*sifreleFunc(string kaynak, string hedef, string anahtar)*" şeklinde bir fonksiyon oluşturulduğunu üst bölümlerde belirtmiştik. Yine aynı şekilde, şifre çözmek amacıyla da *sifreleFunc(string kaynak, string hedef, string anahtar)* şeklinde bir fonksiyon oluşturulmuştur. Kullanıcı şifre çöz butonuna bastığı an *sifrecoz_Click()* fonksiyonu devreye girer. Tıpkı *sifrrele_Click()* fonksiyonunda olduğu gibi, bu fonksiyonda da birçok değişken görev almaktadır; "*tarih*", "*temp*", "*tempAdres*".

Bu değişkenlerin de görevleri şu şekildedir; "*tarih*" değişkeni, şu anki zamanı gün/ay/yıl/saat/dakika/saniye şeklinde alır, "*temp*" değişkeni, sistemin geçici dosya dizininin adresini alır, "*tempAdres*" değişkeni, temporal (geçici) hafızada bulunan ve şifre çözme aşamasında kullanılan *.GUNCE dosya formatının (proje için böyle bir dosya formatı oluşturulmuştur.) adresini çeker.

Eğer anahtar ve şifresiz metnin oluşturulacağı konum girildiyse ve şifreli metnin dosyasında herhangi bir sorun yoksa *try* metodu ile dosyanın şifre çözme işlemine ve değişkenlerin değerlendirilmesine geçilir ve *sifrecozFunc* fonksiyonu ile dosyanın şifresi çözülüp "Çözülmüş Metin" başlıklı metin kutusuna yazdırılır.

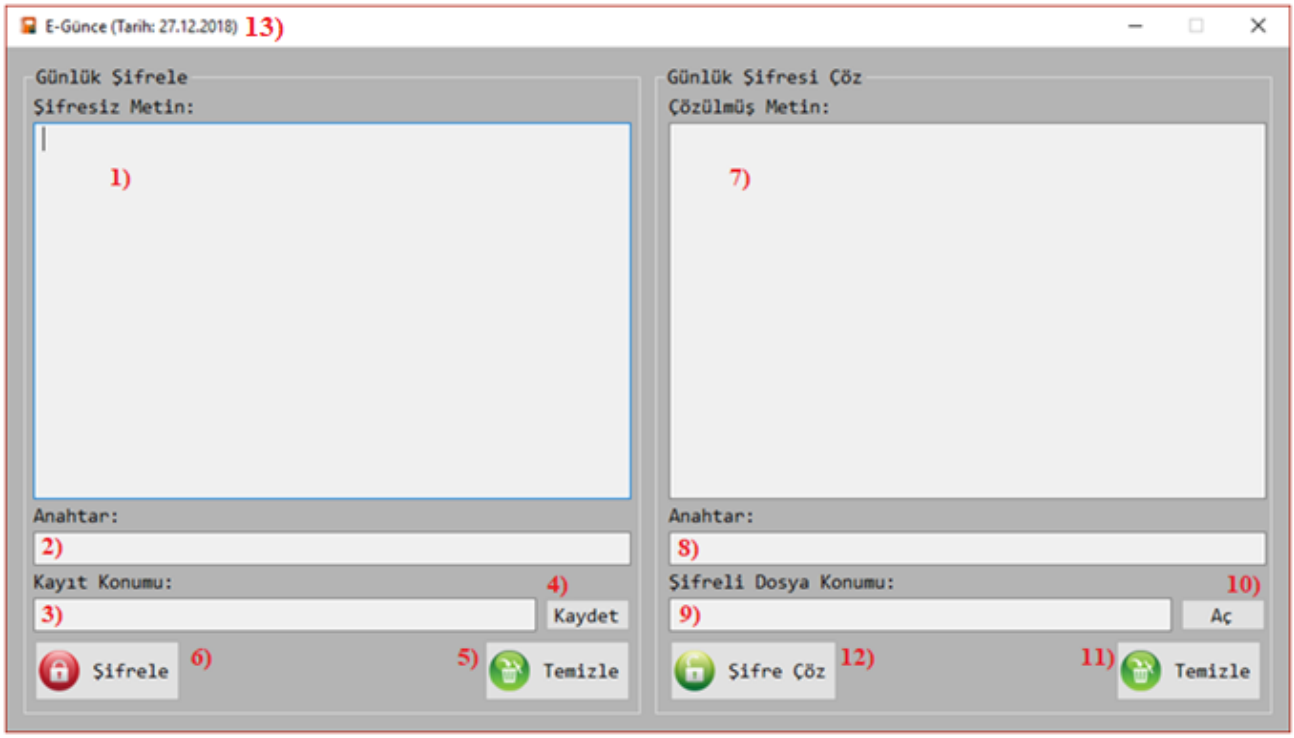
```
private void sifrecoz_Click(object sender, EventArgs e)
{
    if (sifrecozAnahtar.Text != "" && sifrecozKonum.Text != "")
    {
        try
        {

            string tarih = DateTime.Now.ToString("ddMMyyyyHHmmss");
            string temp = System.IO.Path.GetTempPath();
            string tempAdres = temp + tarih + ".GUNCE";
            sifrecozFunc(sifrecozKonum.Text, tempAdres, sifrecozAnahtar.Text);
            sifrecozMetin.Text = File.ReadAllText(tempAdres);
            MessageBox.Show("Dosyanın şifresi çözüldü.");
            File.Delete(@tempAdres);
        }
        catch
        {
            MessageBox.Show("Bir sorun oluştu.");
        }
    }
    else
    {
        MessageBox.Show("Tüm alanları doldurun.");
    }
}
```

Görsel 3. sifrecoz_Click Fonksiyonu ve İşlevlerini Gösteren Kod Bloğu

3. Bulgular ve Gerçekleşme

3.1) Windows Formunun Tasarımı

Projenin C# ile programlanıp, Visual Studio geliştirildiğini daha önceden belirtmiştik. Proje, tek formdan oluşmaktadır. E-Günce başlıklı form, Visual Studio 2017 Community sürümünde geliştirilmiştir.



Görsel 4. Formun Tanıtılması

1. Şifrelenecek metnin yazıldığı metin kutusudur.
2. Şifreleme işlemi sırasında kullanılan anahtarın yazıldığı metin kutusudur.
3. Şifreli dosyanın kaydedileceği konumun yazıldığı metin kutusudur.
4. *savefiledialog* fonksiyonuyla, şifreli dosyanın kayıt konumunu seçer.
5. Formu temizler.
6. Şifresiz metni şifreler.
7. Şifresiz metnin yazıldığı metin kutusudur.
8. Şifre çözme işlemi sırasında kullanılan anahtarın yazıldığı metin kutusudur.
9. Şifreli dosyanın konumunun yazıldığı metin kutusudur.
10. *openFileDialog* fonksiyonuyla, şifreli dosyanın açılmasını sağlar.
11. Formu temizler.
12. Şifreli dosyanın şifresini çözer.
13. Şu anki tarihi gün/ay/yıl şeklinde kullanıcıya verir.

3.2) Şifreleme Adımları

a. Kullanıcıdan anahtar, şifresiz metin ve kayıt konumu alınır. Eğer eksik bilgi varsa *if-else* metoduyla kullanıcıya uyarı mesajı verilir. Tüm alanlar doluysa diğer adıma geçilir.

b. O anki tarih *tarih*, değişkenine, geçici dosya dizini, *temp* değişkenine, kullanıcı adı *kullanici* değişkenine aktarılır.

c. Kullanıcıdan alınan şifresiz metin, kullanıcı bilgileri ve tarih içeriği, *File.WriteAllText* fonksiyonu ile geçici dosya dizininde şifreli bir metin dosyası biçimde geçici bir süre için kaydedilir.

d. Kaydedilen bu geçici dosya, *sifreleFunc* fonksiyonu ile kalıcı olarak kullanıcının istediği konuma 2. kez AES şifreleme algoritmasıyla şifrelendikten sonra taşınır.

e. Oluşturulan geçici dosya, *File.Delete* fonksiyonu ile silinir.

f. Metnin şifrelendiğine dair bilgilendirme mesajı, tarih bilgisiyle beraber kullanıcıya verilir.

g. Eğer eksik kısımlar varsa bu konuda kullanıcı uyarılır. Eğer herhangi bir problem gerçekleşirse, *try-catch* metodu ile kullanıcı uyarılır.

3.3) Şifre Çözme Adımları

a. Kullanıcıdan anahtar ve şifreli dosyanın kayıt konumu alınır. Eğer eksik bilgi varsa *if-else* metoduyla kullanıcıya uyarı mesajı verilir. Tüm alanlar doluysa diğer adıma geçilir.

b. O anki tarih *tarih*, değişkenine, geçici dosya dizini, *temp* değişkenine, geçici dosya format protokolü (.GUNCE dosya formatı), *tempAdres* değişkenine aktarılır.

c. Daha sonra *sifrecozFunc* isimli fonksiyon ile *sifrecozKonum* adresindeki şifreli dosya, anahtar ile AES şifre çözme algoritması ile şifre çözme işlemine tabi tutulur, geçici dosya dizininde .GUNCE dosya formatıyla bir şifreli dosya oluşturulur. Şifresiz metin, şifresiz metin başlıklı metin kutusuna yazdırılır.

d. Metnin şifresinin çözüldüğüne dair bilgilendirme mesajı, kullanıcıya verilir.

e. Geçici olarak oluşturulan .GUNCE formatlı dosya, *File.Delete* fonksiyonu ile silinir.

f. Eğer eksik kısımlar varsa bu konuda kullanıcı uyarılır. Eğer herhangi bir problem gerçekleşirse, *try-catch* metodu ile kullanıcı uyarılır.

3.4) Proje Yazılımının Kodları

Raporun bu bölümünde, projede kullanılan ve proje için geliştirilen kodlar paylaşılmış ve açıklanmıştır. Projenin C# ile geliştirildiğini aktarmıştık. Bu nedenle açıklanan kodların tamamı C# dilinde yazılmış olan kodlardır.

Ayrıca proje için zorunlu olan bir diğer yapı da .NET Framework'tür. Framework içinde projeler için gerekli kod bloklarını içeren bir pakettir.

Projede Microsoft tarafından yayımlanan .NET Framework'ün 4.5 versiyonu kullanılmıştır. Ancak bu kod blokları (framework) ilerleyen bölümlerde açıklanmayacak olup, zaten standart yapıda olan içeriklerdir.

a. Açılış Kodları

```
namespace E_Günce
{
    public partial class egunce : Form
    {
        public egunce()
        {
            InitializeComponent();
        }
```

Kod Bloğu 1. Açılış Kodları

Bu kodlar, projenin başlaması için şarttır. İlgili ad alanlarını yükler, formun ismini oluşturur ve gerekli komponentleri yükler.

b. sifreleFunc Fonksiyon Kodları

```
private void sifreleFunc(string kaynak, string hedef, string anahtar)
{
    AesManaged AES = new AesManaged();
    using (MD5CryptoServiceProvider MD5 = new MD5CryptoServiceProvider())
    {
        AES.KeySize = MD5.HashSize;
        AES.BlockSize = MD5.HashSize;
        AES.IV = MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(anahtar));
        AES.Key = MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(anahtar));
    }
    using (FileStream reader = new FileStream(kaynak, FileMode.Open, FileAccess.Read))
    {
        using (FileStream writer = new FileStream(hedef, FileMode.OpenOrCreate, FileAccess.Write))
        {
            using (CryptoStream cs = new CryptoStream(writer, AES.CreateEncryptor(), CryptoStreamMode.Write))
            {
                int bufferSize = 4096;
                byte[] buffer = new byte[bufferSize];
                int bytesRead;
                do
                {
                    bytesRead = reader.Read(buffer, 0, bufferSize);
                    if (bytesRead != 0)
                    {
                        cs.Write(buffer, 0, bytesRead);
                    }
                }
                while (bytesRead != 0);
                cs.FlushFinalBlock();
            }
        }
    }
}
```

Kod Bloğu 2. sifreleFunc Fonksiyon Kodları

Projede kullanılan bu fonksiyon, şifresiz metnin şifrelenmesini sağlamaktadır. Projede AES algoritmasının kullanıldığı daha önce de belirtilmişti.

Bu kod bloğunda da, AES ile şifreleme yapılmaktadır.

Kodları kısaca açıklayacak olursak; ilk önce AesManaged tipinde AES isminde bir nesne oluşturulmuştur. Daha sonra bu nesnenin özellikleri ve anahtarları tanımlanmıştır.

Ardından *FileStream* yapısının *reader* isimli yapı kullanarak yazılan şifresiz metnin içeriği okunmuştur. Okunan bu içerik, *writer* yapısıyla *hedef* değişkeninde AES algoritması ile şifrelenmiştir.

c. sifrecozFunc Fonksiyon Kodları

```
private void sifrecozFunc(string kaynak, string hedef, string anahtar)
{
    AesManaged AES = new AesManaged();
    using (MD5CryptoServiceProvider MD5 = new MD5CryptoServiceProvider())
    {
        AES.KeySize = MD5.HashSize; // they are 128 bit compatible
        AES.BlockSize = MD5.HashSize; // they are 128 bit compatible
        AES.IV = MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(anahtar));
        AES.Key = MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(anahtar));
    }
    using (FileStream reader = new FileStream(kaynak, FileMode.Open, FileAccess.Read))
    {
        using (FileStream writer = new FileStream(hedef, FileMode.OpenOrCreate, FileAccess.Write))
        {
            using (CryptoStream cs = new CryptoStream(reader, AES.CreateDecryptor(), CryptoStreamMode.Read))
            {
                int bufferSize = 4096;
                byte[] buffer = new byte[bufferSize];
                int bytesRead;
                do
                {
                    bytesRead = cs.Read(buffer, 0, bufferSize);
                    if (bytesRead != 0)
                    {
                        writer.Write(buffer, 0, bytesRead);
                    }
                }
                while (bytesRead != 0);
            }
        }
    }
}
```

Kod Bloğu 3. sifrecozFunc Fonksiyon Kodları

Bu kodlarda ise, bir önceki bloktaki kodların tam tersi şeklinde şifre çözme işlemi yine AES algoritması ile gerçekleştirilmektedir.

NOT: AES algoritmasının nasıl çalıştığı, hangi değişkenin ne işe yaradığı gibi bilgiler, bu fonksiyonlar standart olduğu için bu raporda verilmemektedir. Ayrıntılı bilgi için tıklayınız.

d. egunce_Load Fonksiyon Kodları

```
private void egunce_Load(object sender, EventArgs e)
{
    string date = DateTime.Now.ToString("dd.MM.yyyy");
    this.Text = "E-Günce (Tarih: "+date+")";
    sifrecozMetin.ReadOnly = true;
}
```

Kod Bloğu 4. Egunce_Load Fonksiyon Kodları

C# dili ile geliştirilen *Windows Form* projelerinin neredeyse tamamında bu fonksiyonu (load) görmeniz muhtemeldir. Bu fonksiyon projenin formunun başlatıldığı an gerçekleştirmesi gereken görevleri içerir.

Projedeki load fonksiyonunda *date* değişkenine *DateTime.Now* yapısı kullanılarak anlık zaman bilgisi tanımlanmıştır. Daha sonra *this.Text* deyimiyle bu tarih bilgisi formun başlığına yazdırılmıştır.

Ayrıca, şifresi çözülmüş metindeki içeriğin değiştirilmesini engellemek amacıyla, *sifrecozMetin* isimli metin kutusunun *ReadOnly* olayı *true* hale getirilerek yalnızca okunması sağlanmıştır.

e. sifreleKaydet_Click Fonksiyon Kodları

```
private void sifreleKaydet_Click(object sender, EventArgs e)
{
    DialogResult result = folderBrowserDialog1.ShowDialog();
    if (result == DialogResult.OK)
    {
        sifreleKonum.Text = folderBrowserDialog1.SelectedPath;
    }
}
```

Kod Bloğu 5. sifreleKaydet_Click Fonksiyon Kodları

Bu kodlarda kullanıcının şifreli dosyanın konumunu seçmesi sağlanmaktadır. Kullanıcı "Aç" butonuna bastığı an, folderBrowserDialog nesnesi ile şifreli dosyanın konumu seçilir.

f. ac_Click Fonksiyon Kodları

```
private void ac_Click(object sender, EventArgs e)
{
    OpenFileDialog dosyaac = new OpenFileDialog();
    if (dosyaac.ShowDialog() == DialogResult.OK)
    {
        sifrecozKonum.Text = dosyaac.FileName;
    }
}
```

Kod Bloğu 6. ac_Click Fonksiyon Kodları

Bu kodlarda ise şifreli dosyayı şifre çözme işlemine tabi tutmak üzere açmak için kullanılan OpenFileDialog nesnesi yer almaktadır.

g. temizle_Click Fonksiyon Kodları

```
private void temizle1_Click(object sender, EventArgs e)
{
    sifreleAnahtar.Text = "";
    sifreleMetin.Text = "";
    sifreleKonum.Text = "";
}

private void temizle2_Click(object sender, EventArgs e)
{
    sifrecozAnahtar.Text = "";
    sifrecozKonum.Text = "";
    sifrecozMetin.Text = "";
}
```

Kod Bloğu 7. temizle_Click Fonksiyon kodları

Bu fonksiyon, gerekli olduğu zamanlarda kullanıcıya hız kazandırmak amacıyla temizle1 ya da temizle2 butonlarına basıldığı zaman anahtar, konum ve metin isimli metin kutularının içeriğini sıfır yapmak suretiyle temizlemek için kullanılmıştır.

4. Sonuçlar ve Tartışma

Projemizde C# dilini kullanarak insan yaşamında uzun yıllardır bulunan bir nesne olan günlüğün yazılımını geliştirdik. Bu günlüğün güvenliğini Rijndael Algoritması olarak da bilinen AES ile sağladık.

Neden C# dilinin kullanıldığına gelecek olursak, C# gayet hızlı ve kolay kullanılabilir bir dildir, çoklu platform desteği sayesinde birçok işletim sistemi tarafından desteklenmektedir. Ayrıca halen yüksek güvenlik sunan AES algoritması, C# diliyle çalışmamızı mümkün kılmıştır.

5. Öneriler

Projenin ilerleyen dönemlerinde, bulut depolama desteği sayesinde bilgisayarda herhangi bir şifreli veri depolamaya gerek kalmaksızın, her yerden dosya şifrelemeyi ve şifrelenen verilere erişebilmeyi mümkün kılmak istiyoruz. Bunun için Server geliştirme ve PHP diline ihtiyaç duyacağız.

Ayrıca halen üzerinde çalıştığımız kendi güvenlik standardımızı da uygulama kapsamında kullanabilmek, ülkemize özgü kriptoloji çalışmalarına katkıda bulunmak, gelecek planlarımız dahilinde.

6. Kaynakça

a. https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.rijndael?view=netframework-4.7.2

b. https://docs.microsoft.com/tr-tr/dotnet/api/system.security.cryptography?view=netframework-4.7.2

c. https://www.lri.fr/~fmartignon/documenti/systemesecurite/5-AES.pdf

d. https://www.linkedin.com/pulse/kriptolojinin-tarihçesi-nihal-kindap

e. https://scholar.google.com.tr/scholar?hl=tr&as_sdt=0%2C5&q=cryptography&btnG=&oq=cryptoh

PROJE ÖZETİ

PROJE ADI: E-Günce: Şifreli Günlük Yazılımı

İnsanlar uzun yıllardır günlük yazmaktadırlar. Günlüklerde kişisel ve özel bilgiler de bulunabilmektedir. Bu nedenle günlüğün sahibinin günlüğünü koruması gerekmektedir. Bu projede de, güvenliğini Advanced Encryption Standard isimli algoritma (*Rijndael*) ile sağlayan bir günlük yazılımı geliştirilmiştir. Hızlı ve kolay kullanılabilir olmasından dolayı proje, C# dili kullanılarak geliştirilmiştir.

Projenin şifreleme adımları şu şekildedir:

1) Kullanıcı şifrelenmesini istediği günlük metnini Windows form içerisinde bulunan ilgili alana yazar,
2) Güvenliğin sağlanması için kullanılacak olan anahtarı da (*bu anahtar çok önemlidir zira şifreli içerik yalnızca aynı anahtar ile çözülebilir.*) ilgili alana yazar,
3) Şifreli dosyanın bulunacağı konumu seçer,
4) Şifrele butonuna basar.
5) Dosya AES algoritması ile şifrelenir.

Eğer herhangi bir hata oluşmazsa, kullanıcının seçtiği konumda günlüğün şifrelendiği tarihin de yazılı olduğu şifreli bir dosya oluşturulur.

Şifre çözme adımlarını da şu şekilde sıralayabiliriz:

1) Kullanıcı şifreli dosyayı Windows formda bulunan “Aç” butonu ile seçer,
2) Şifrelemede kullanılan *aynı* anahtarı da anahtar kısmına yazar,
3) Şifre çöz butonuna basar.
4) Dosya AES algoritması ile geri-şifrelenir. (Şifresi çözülür.)

Eğer dosya hasar almamışsa ve anahtar da doğruysa, şifre çözülür ve şifresiz metin Windows formdaki ilgili bölümde otomatik olarak yazılır.

Gelecekte, halen üzerinde çalıştığımız kendi güvenlik standardımızı da uygulama kapsamında kullanabilmek, ülkemize özgü kriptoloji çalışmalarına katkıda bulunmak, gelecek planlarımız dahilindedir.

Anahtar Kelimeler: Şifreleme, AES, Rijndael, günlük, kriptoloji, kriptografi, yazıbilim, şifrebilim, C#, C Sharp, Advanced Encryption Standard, E-Günce

PROJE PLANI

PROJE ADI: E-Günce: Şifreli Günlük Yazılımı

1) Amaç ve Kapsam:

İnsanlar uzun yıllardır günlük yazmaktadırlar. İnsanlar günlüklerinde ruh hallerinden günlük yaşamlarına kadar birçok konuda yazı yazabilirler. Ancak günlüklerin kişiye özel olduğu unutulmamalıdır. Yani günlüğü yazan kişi aynı zamanda günlüğünü de yabancı ellerden koruyabilmelidir. E-Günce projesinde de yazılımsal yollardan koruma altına alınan bir günlük uygulamasının geliştirilmesi amaçlanmıştır. Günlüğü şifrelemek için kişiye özel bir anahtar da girilmesi gerekmektedir. Aynı şekilde şifrelenen günlüğün şifresini çözebilmek için de şifrelerken kullanılan anahtarın uygulamaya girilmesi gerekmektedir.

2) Yöntem ve Gereçler:

Hâlihazırda var olan şifreleme algoritmaları araştırılmıştır. Amaca en uygun olan algoritmanın AES olduğuna karar verilmiştir. Çünkü AES dosya şifrelemede yüksek performans sağlamaktadır.

Aynı şekilde, proje için en uygun programlama dili taranmış, C# dilinin gayet hızlı ve kolay kullanılabilir bir dil olmasından ve çoklu platform desteği sayesinde birçok işletim sistemi tarafından desteklenmesinden dolayı çalışmanın C# diliyle gerçekleştirilmesine karar verilmiştir.

3) İş-Zaman Tablosu

İşin tanımı	Aylar									
	NİSAN	MAYIS	HAZİRAN	TEMMUZ	AĞUSTOS	EYLÜL	EKİM	KASIM	ARALIK	OCAK
LİTERATÜR TARAMASI	x	x	x						x	
VERİ TOPLANMASI		x	x		x			x	x	
ARAZİ ÇALIŞMASI		x	x	x	x		x			
PROJE RAPORU YAZIMI					x		x	x	x	