Hill Şifreleme Algoritması (Hill Cipher Algorithym)

Hazırlayanlar: Ali Emre Nebiler – 19011070 | Emre Arslanoğlu – 18011061

Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği, 1. Sınıf, Yapısal Prog. Giriş Dersi

Şifreleme Algoritmalarına Genel Bakış

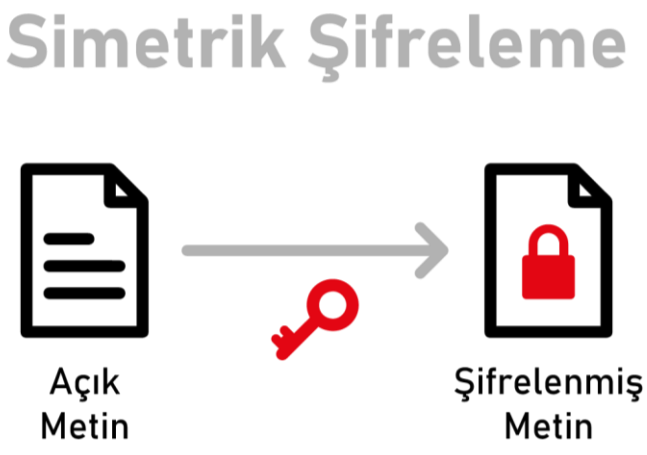
Günümüzde duyulan en büyük ihtiyaçlardan birisi "bilginin doğru ve güvenli bir şekilde saklanıp, gerektiği yer ve zamanda ilgili kişiler tarafından" kullanılmasıdır. Bu kapsamda karşımıza "Kripto" yani "Şifreleme" çıkmaktadır. Şifreleme en eski tarihten beri kullanılan bir sistemdir. Tarihte şifrelemeyi ilk kullanan bilinen kaynaklarda Sezar'dır. Şifreleme; verinin, bilginin her şartta istenilen kişilere verinin güvenli bir şekilde iletilmesi için kullanılmaktadır. Şifreleme işlemi yaparken kullandığımız temel algoritmalar ise kriptosistemi oluşturur ki, bu algoritmalar kriptosistemin temel yapı taşlarıdır.

Kriptografide şifreleme algoritmaları, bir bilgi parçasını şifrelemek ya da bu parçanın şifresini çözmek için kullanılan ve bitler serisi şeklinde olan anahtarları meydana getirirler. Bu anahtarların nasıl kullanıldığı simetrik ve asimetrik şifreleme arasındaki farkı yaratır.

Şifreleme algoritmaları genellikle iki kategoriye bölünür; simetrik ve asimetrik şifreleme. Bu iki şifreleme yönteminin arasındaki en temel fark, simetrik şifreleme algoritmasında tek anahtar kullanılırken, asimetrik şifrelemede iki farklı fakat birbiriyle bağlantılı anahtar kullanılmasıdır. Böylesi bir farklılık görünüşte basit olsa da iki şifreleme tekniğinin arasındaki işlevsel farklılıkları ve tekniklerin kullanım şekillerini belirler.

Simetrik Şifreleme:

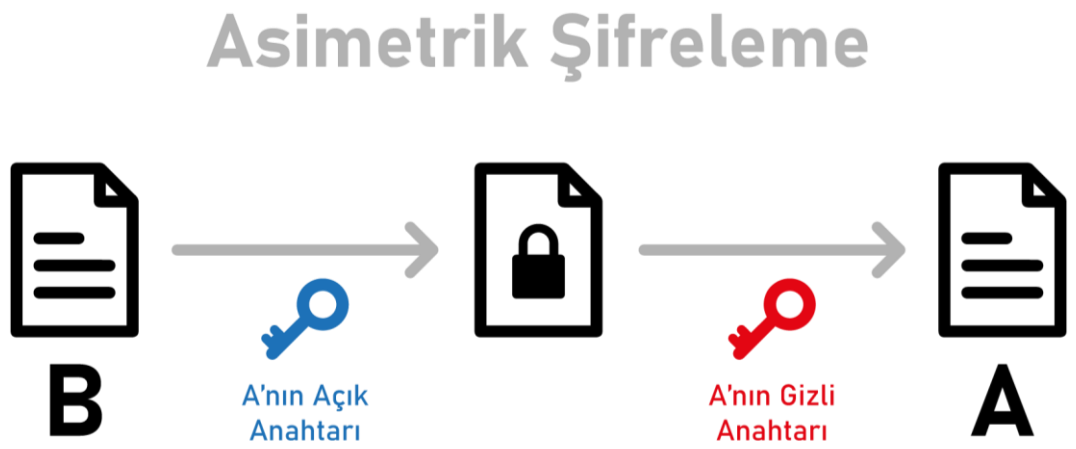
Simetrik şifreleme algoritmalarının temelinde tek bir gizli anahtar ile veri şifrelenir ve aynı anahtarla veri çözülür. İki kullanıcıda da aynı anahtar bulunur. Bu şifreleme yöntemleri matematiksel olarak karmaşık olmayan işlemlerde çalışır ve oldukça hızlı çalışır fakat metin kullanıcıya gönderilmesi kadar anahtarında güvenli bir biçimde diğer kullanıcıya ulaştırılması gerekir. Hızı, basitliği ve güvenliği sayesinde simetrik şifreleme, internet trafiğinin güvenliğini sağlamaktan, bulut sunucularda depolanan veriyi korumaya kadar geniş çapta uygulamalarda sıklıkla kullanılmaktadır. Anahtarları güvenle transfer etme sorununu çözmek için sıklıkla asimetrik şifrelemeyle eş zamanlı olarak kullanılsa da simetrik şifreleme düzenleri modern bilgisayar güvenliğinin önemli bir öğesi olmaya devam etmektedir.



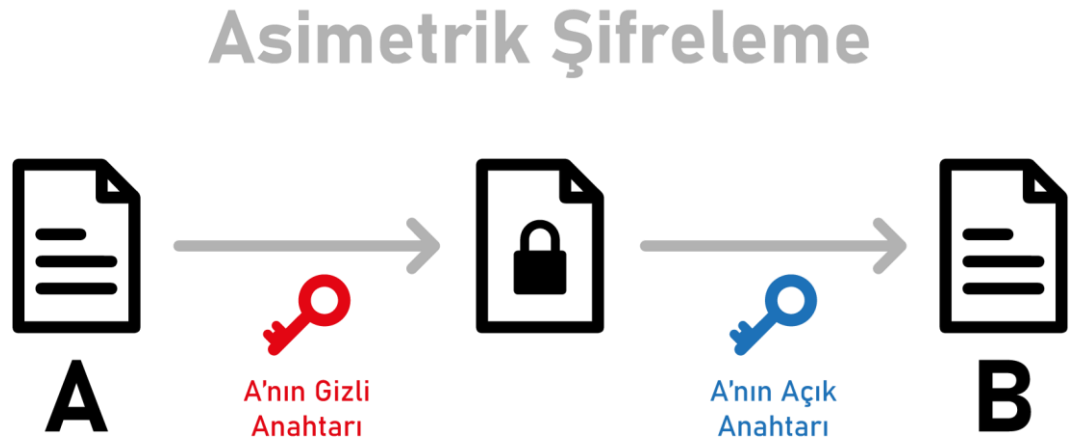
Asimetrik Şifreleme:

Asimetrik Şifreleme Algoritmaları, açık anahtarlı şifreleme olarak adlandırılmaktadır. Simetrik şifreleme algoritmasından farklı olarak asimetrik şifreleme algoritmasında açık ve gizli anahtar olmak üzere iki farklı anahtar bulunmaktadır. Şifreleme anahtarına açık anahtar, şifre çözüm anahtarına ise özel anahtar ismi verilmektedir. Şifre çözme işlemi için kullanılan anahtar ile şifreleme işlemi için kullanılan anahtarlar birbirinden farklıdır. Şifre anahtarının herkese açık olması gerektiğinden bu algoritmalara açık anahtarlı algoritmalar denilmiştir. Bu sebepten dolayı bir kullanıcının açık anahtarı ile şifrelenen bir metin sadece bu kullanıcıya ait olan gizli anahtar ile metnin şifresi çözülebilmektedir.

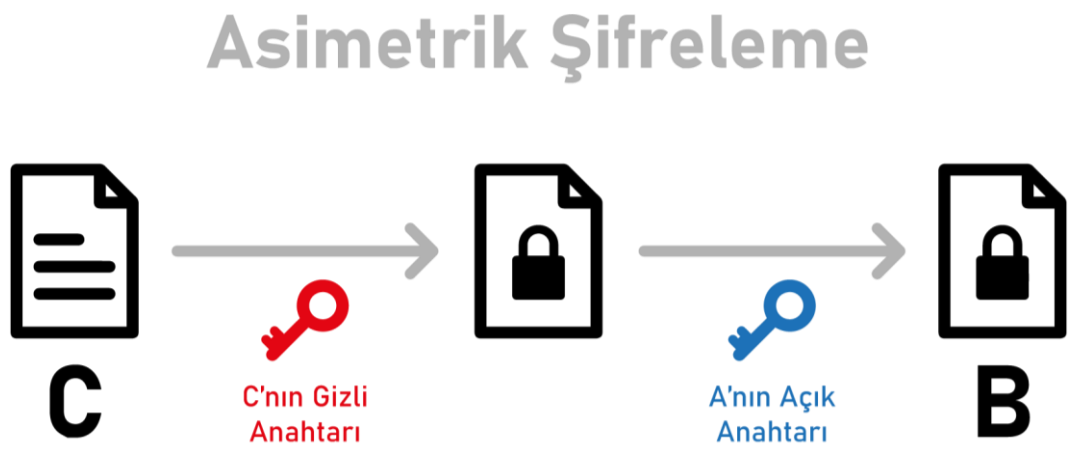
Ayrıca asimetrik şifrelemelerde kişi göndereceği metni önce kendi gizli anahtarı ile sonra karşıdaki kişinin açık anahtarı ile şifreleyerek gönderebilir. Bu durumda metni alan kullanıcı kendi gizli anahtarı ve gönderen kişinin açık anahtarı ile metni açarak hem güvenli bir şekilde metne ulaşır hem de metni gönderen kişinin kimlik doğrulamasını yapar.



(Bu örnekte B kişisi sadece A kişisinin görebileceği bir metin yollamak için A'nın açık anahtarı ile metni şifreliyor. Sadece A kişisi bu metni kendi gizli anahtarı ile çözebiliyor.)



(Bu örnekte asimetrik şifrelemenin kimlik doğrulama amacı ile kullanımını görüyoruz. B kişisi A'nın açık anahtarı ile şifreli metni çözebildiği için metnin A kişisinden geldiğini anlıyor.)



(Bu örnekte ise B kişisi A'nın açık anahtarını kullandığı için C kişisinin şifreli metnini çözememiştir. Çünkü C'nin gizli anahtarı ile şifrelenen metni ancak C'nin açık anahtarı çözebilir.)

Simetrik ve asimetrikleri şifreleme algoritmalarının tanımını yaptıktan sonra bu algoritmaların genel özellikleri, avantajları ve dezavantajlarını belirtmeliyiz.

Özellikler	Simetrik Şifreleme	Asimetrik Şifreleme
Gizlilik	✓	✓
Bütünlük	✗	✓
Kimlik Doğrulama	✗	✓
İnkâr Edilemezlik	✗	✓
Performans	Hızlı	Yavaş
Güvenlik	Anahtar uzunluğuna bağlı	Anahtar uzunluğuna bağlı

Simetrik Şifreleme Algoritmalarının Avantajları:

1. Simetrik şifreleme oldukça hızlıdır.
2. Donanımlarla birlikte çalışabilirler.
3. Hızlı ve basit olmasına rağmen yüksek seviyede güvenlik sunarlar.
4. Anahtarları oldukça kısadır ve basit matematiksel işlemlerden oluşur.
5. Daha güçlü şifrelerin oluşturulmasında aracı olarak kullanılabilirler.

Simetrik Şifreleme Algoritmalarının Dezavantajları:

1. Anahtarların saklanması ve taraflara ulaştırılması zordur, güvenlik riski içerir.
2. Büyük ağlarda çok fazla anahtara ihtiyaç duyulur, ölçeklenebilir değildir. N kullanıcılı bir sistemde [N*(N-1)/2] anahtar vardır.
3. Bütünlük sağlaması yoktur. Anahtarı ele geçiren kişi veriyi değiştirmiş olabilir.
4. Kimlik doğrulama (authenticity) sağlamaz. Aynı anahtara sahip olan herhangi birisi tarafından veri şifrelenmiş olabilir.

Asimetrik Şifreleme Algoritmalarının Avantajları:

1. Şifrelerin kırılması nispeten daha zordur.
2. Kimlik doğrulama, bütünlük ve gizlilik ilkelerini gerçekleştirmek için güveli bir yoldur.
3. Şifrelemede iki anahtar kullanıldığı için sayısal imza ile inkâr edememeyi sağlar.

Asimetrik Şifreleme Algoritmalarının Dezavantajları:

1. Simetrik şifrelemeye göre çok daha yavaştır.
2. Sistemde daha çok CPU harcanmasına neden olur.

NOT: Bir şifreleme algoritmasının performansı;

- sistemin kırılabilme süresinin uzunluğuna,
- şifreleme ve çözme işlemlerine harcanan süreye,
- şifreleme ve çözme işleminde ihtiyaç duyulan bellek miktarına ve
- algoritmanın kurulacak sisteme uygunluğuna bağlıdır.

Raporun devamında, bir simetrik şifreleme yöntemi olan

Hill Şifreleme algoritmasından bahsedeceğiz.

Hill Şifreleme Algoritmasının Tarifi

Şifreleme:

Şifreleme işlemine başlamadan önce, anahtar olarak kullanacağımız bir kare matris belirlemeliyiz. Kare matrisin boyutunu istediğimiz boyutta ayarlayabiliriz. Matrisimizin boyutuna N diyelim. (Biz kendi örneğimizde N değerini 2 alarak, 2x2'lik bir matris kullanacağız.)

$$N=2 \quad \begin{bmatrix} 1 & 3 \\ 8 & 13 \end{bmatrix}$$

İkinci adımda şifrelemek istediğimiz metni, N uzunluğundaki parçalara ayıracağız. Bu aşamada bir problemle karşılaşıyoruz: her metin eşit parçalara bölünemeyebilir. Örneğin N değerini 2 aldığımızda ve "MERHABA" metnini şifrelemek istediğimizde, sondaki A harfi tek kalmaktadır. Bu problemi ortadan kaldırmak adına kalan boşlukları kendi belirlediğimiz bir değer ile dolduracağız. (Örneğimizde boş kalan yerlere A harfini koyacağız.)

"MERHABA"

M E R H A B A A

Sonra şifrelemek istediğimiz metnin parçalarını sayılara çevirmeliyiz. Bu işlemi kendi belirlediğimiz değerlerle veya ASCII tablosundaki değerlerle yapabiliriz. (Örneğimizde kendi belirlediğimiz değerleri kullanacağız.)

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13

O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25

M E	R H	A B	A A
[12, 4]	[17, 7]	[0, 1]	[0, 0]

Sonrasında ise anahtar matrisimiz ile bu parçaların her birini çarpacağız, yani matris çarpımı yapacağız.

Fakat çarpım sonuçları, harflerin sayı karşılıkları için belirlediğimiz aralıkta olmayabilir, yani çok daha büyük sayılar çıkabilir. Bu problemi ortadan kaldırmak için de elde ettiğimiz tüm çarpımların modunu almalıyız. (Örneğimizde İngiliz alfabesini kullandığımız için 26 karakterimiz var, bu sebeple 26'nın modunu alacağız.)

M E
[12, 4]* $\begin{bmatrix} 1 & 3 \\ 8 & 13 \end{bmatrix}$ = [12*1 + 4*8, 12*3 + 4*13]
= [44, 88]

[44, 88] mod 26 = [18, 10]

R H
[17, 7]* $\begin{bmatrix} 1 & 3 \\ 8 & 13 \end{bmatrix}$ = [17*1 + 7*8, 17*3 + 7*13]
= [73, 142]

[73, 142] mod 26 = [21, 12]
.....
[8, 13] mod 26 = [8, 13]
.....
[0, 0] mod 26 = [0, 0]

Son olarak elde ettiğimiz tüm değerleri sırayla yan yana ekleyeceğiz ve sayı değerlerinin hangi harflere karşılık geldiğini bulacağız.

[18, 10] [21, 12] [8, 13] [0, 0]
S K V M I N A A

"SKVMINAA"

Böylece "MERHABA" kelimesini şifrelemiş olduk.

Deşifreleme:

Şifreyi çözme işlemi de şifreleme adımlarının aynılarını içermektedir, temel olarak yine matris çarpımı ve mod hesabı yapılır.

Deşifreleme işleminin mantığını anlayalım:

Şifrelemek istediğimiz metnin parçalarını T, anahtar matrisini K, şifrelenmiş halin parçalarını C ile gösterelim. Buna göre yaptığımız işlemi

$T * K \bmod 26 = C$ yani basitçe $T * K = C$

olarak gösterebiliriz. Bir matrisin kendisinin tersi ile çarpımının birim matris olduğunu biliyoruz. Yani:

$K * K^{-1} = 1$

Buna göre

$T * K * K^{-1} = C * K^{-1}$

$T = C * K^{-1}$ yani $T = C * K^{-1} \bmod 26$ olur.

Buradan anlıyoruz ki şifreyi çözmemiz için anahtar matrisinin tersine ihtiyacımız var.

<u>Anahtar Matrisinin Tersini Bulma:</u>

Matrisin tersini bulmak için kofaktör ile tersini bulma yöntemi kullanmalıyız. Ancak tersini bulduğumuzda matris elemanları kesirli çıkabilir ve bu da hesaplama sorunlarına, yanlış hesaplamalara sebep olabilir. Bu sebeple Hill Şifreleme algoritmasında kofaktör yöntemi kullanılmalı ve ek işlemler ile anahtar matrisinin tersi bulunmalıdır.

Anahtar matrisimiz ve kofaktör ile tersini bulma yöntemi:

$$K = \begin{bmatrix} 1 & 3 \\ 8 & 13 \end{bmatrix} \qquad K^{-1} = \frac{1}{|K|} * Adj(K)$$

Fakat ters matriste tam sayı değerleri elde edebilmek için kofaktör formülünü farklı ifade etmeliyiz.

$K = 1 / K^{-1}$ olduğunu, aynı zamanda $|K| = 1 / |K^{-1}|$ olduğunu biliyoruz.

O halde

$$K^{-1} = \frac{1}{|K|} * Adj(K) \qquad K^{-1} = |K^{-1}| * Adj(K)$$

formülünü elde ederiz. O halde $|K^{-1}|$ değerini bulmalıyız.

$|K^{-1}|$ değerini hesaplarken ihtiyacımız olacağı için determinantı ve kofaktör matrisini buluyoruz.

$$|K| = 1*13 - 8*3 = -11 \qquad Adj(K) = \begin{bmatrix} 13 & -3 \\ -8 & 1 \end{bmatrix}$$

$|K| = 1 / |K^{-1}|$ ifadesinde $|K^{-1}|$ ifadesini karşıya atarsak

$|K|*|K^{-1}| = 1$ olduğunu buluruz.

İşlemlerimizde mod kullandığımız için, burada da mod kullanmamızın bir sakıncası olmayacaktır. Böylece

$|K|*|K^{-1}| = 1 \pmod{26}$ ifadesini elde ederiz.

$|K| = -11$ olduğunu hesaplamıştık.

$(-11)*|K^{-1}| = 1 \pmod{26}$ ifadesini hesapladığımızda $|K^{-1}|$ değerimizi -19 buluruz.

Son olarak bulduğumuz $|K^{-1}|$ değeri ile anahtar matrisimizi çarpıp, bulduğumuz her değerin modunu alırsak, ters matrisimizi bulmuş olacağız.

$$K^{-1} = |K^{-1}| * Adj(K)$$

$$= (-19) * \begin{bmatrix} 13 & -3 \\ -8 & 1 \end{bmatrix} = \begin{bmatrix} -247 & 57 \\ 152 & -19 \end{bmatrix}$$

$$\begin{bmatrix} -247 & 57 \\ 152 & -19 \end{bmatrix} \bmod 26 = \begin{bmatrix} 13 & 5 \\ 22 & 7 \end{bmatrix}$$

Artık deşifre işlemini gerçekleştirebiliriz.

Şifrelenmiş metni N uzunluğundaki parçalara ayıracağız ve sayısal karşılıklarını bulacağız.

S K V M I N A A
[18, 10] [21, 12] [8, 13] [0, 0]

Parçaların her birini anahtar matrisinin tersi ile çarpacağız ve her çarpımın modunu alacağız.

$$\begin{matrix} & \text{S K} \\ [\ 18, 10\ ]* & \begin{bmatrix} 13 & 5 \\ 22 & 7 \end{bmatrix} \end{matrix} = [\ 454, 160\ ]$$

[454, 160] mod 26 = [12, 4]
.....
[537, 189] mod 26 = [17, 7]
[390, 131] mod 26 = [0, 1]
[0, 0] mod 26 = [0, 0]

Son olarak elde ettiğimiz tüm değerleri yan yana ekleyeceğiz ve sayı değerlerinin hangi harflere karşılık geldiğini bulacağız.

[12, 4] [17, 7] [0, 1] [0, 0]
M E R H A B A A

“MERHABAA”

Böylece yeniden “MERHABAA” kelimesini yeniden elde etmiş olduk.

Algoritmanın Karmaşıklığı

Metin uzunluğunu M, anahtar matris genişliğini de N ile göstererek karmaşıklığı hesaplayalım:

İlk olarak metnimizi N uzunluğundaki parçalara ayıracağız, yani M/N adet parçamız olacak. M, N sayısına tam bölünmeyebilir, bu durumda da M değerine, N sayısının bir katı olacak şekilde değerler ekleyeceğiz. O zaman parça sayımızı ⌈ M/N ⌉ olarak gösterebiliriz.

Her parça için matris çarpımı yapacağız, yani Nx1 genişliğindeki bir dizi ile NxN genişliğindeki bir matrisi çarpacağız. Buradaki işlem sayısını 1xNxN ile, yani N^2 ile gösterebiliriz.

Matris çarpımını her parça için tekrarlayacak olursak ⌈ M/N ⌉ * N^2 kere işlem gerçekleştirmeliyiz. M değerini N değerinin bir katı kabul edersek de karmaşıklığı M*N olarak gösterebiliriz.

Kısıtlar ve Dikkat Edilmesi Gereken Noktalar

1- Şifrelenen metnin deşifre edilmesi için anahtar matrisinin tersini bulmamız şarttır ancak her matrisin tersi bulunamaz. Matrisin tersinin bulunması için gereken şart matris determinantının sıfırdan farklı olmasıdır. Bu sebeple anahtar olarak her matrisi kullanamayız.
2- Anahtar matrisinin determinantı ile mod değeri (örneğimiz için mod değeri 26) aralarında asal olmalıdır. Aksi taktirde deşifreleme sırasında yanlış sonuçlar elde edilebilir. Bu sebeple anahtar olarak seçebileceğimiz matrisler kısıtlıdır.
3- Belirlediğimiz metin, matris genişliği ile eşit parçalara bölünemediğinde şartı sağlaması için metnin sonuna ekleme yapıyoruz, örneğimizde “MERHABA” metnini “MERHABAAA” olarak şifrelememiz gibi. Bazı metinler için bu durum alıcı için yanlış anlaşılmalara sebep olabilir. Bu sebeple algoritma, anlaşılması yönünden kısıtlıdır.

Uygulama Alanları

Simetrik şifreleme, daha hızlı olması nedeniyle birçok modern bilgisayar sisteminde veri koruması için yaygın olarak kullanılır. Örneğin, Gelişmiş Şifreleme Standardı (AES) Amerika Birleşik Devletleri hükümeti tarafından gizli ve hassas bilgileri şifrelemek için kullanılmaktadır. Hill Şifreleme algoritması da bir tür simetrik şifrelemedir fakat ilkel bir şifreleme yöntemi olduğu için günümüzde tercih edilmemektedir.

Asimetrik şifreleme, birçok kullanıcının bir mesaj ya da veri setini şifrelemesi ya da bunların şifresini çözmesi gerektiği, özellikle de hız ve hesaplama gücünün başlıca önem teşkil etmediği sistemlerde uygulanabilir. Böylesi sistemlere örneklerden biri mesajı şifrelemek için açık anahtarın, mesajın şifresini çözmek için de özel anahtarın kullanıldığı şifreli e-postadır.

Hibrid sistemler; simetrik ve asimetrik şifrelemenin bir arada kullanılmasıdır. Bu tip sistemlerin en tipik örnekleri, internet içinde güvenli iletişim sağlamak üzere tasarlanmış Güvenli Yuva Katmanı (SSL) ve Taşıma Katmanı Güvenliği (TLS) kripto grafik protokolleridir. SSL protokollerinin aksine TLS protokolleri güvenli kabul edilirler ve başlıca tüm web tarayıcılarında yaygın olarak kullanılırlar.

Rakipler ve Kıyaslama

Diğer simetrik algoritma türlerinden Sezar, Vigenere, Vernam ve Playfair Şifreleme algoritmaları, Hill Şifreleme algoritmasının rakipleridir. Aşağıdaki verilerle birbirlerine göre avantajlarını ve dezavantajlarını yorumlayacağız.

2.5KB büyüklüğündeki bir görüntünün şifrelenmesi sırasında algoritmaların performans analiz değerleri:

Algoritma	İşleme Zamanı (MS)		RAM Kullanımı (Byte)		CPU Kullanımı (%)	
	Şifreleme	Çözme	Şifreleme	Çözme	Şifreleme	Çözme
Sezar	22.36	25.59	127.94	127.76	90.75	92.99
Vigenere	24.68	26.52	241.20	241.20	88.91	91.15
Vernam	22.23	23.44	126.63	126.63	87.81	91.15
Hill	205.81	186.84	227.38	227.38	0.36	0.12
Playfair	160.84	139.43	194.43	194.43	0.53	0.37

- Tabloya bakıldığında zaman harcama bakımından Hill Şifreleme algoritması oldukça dezavantajlı bir algoritmadır. Diğer algoritmalar arasında en geç şifreleme ve çözme süresine sahiptir.
- Ram kullanımında ise aynı şekilde dezavantajlı durumda olan Hill Şifreleme algoritması en fazla yer kaplayan algoritmadır.
- CPU üzerinde yük oluşturmada ise en başarılı algoritma Hill Şifreleme algoritmasıdır. Tablo incelediğinde bellek kullanımı ile işlemci yükü arasında ters bir orantı bulunmaktadır.

Şimdi de başka bir şifreleme deneyini inceleyelim.

Şifrelenecek Görsel: Lenna

Boyutu: 1.10KB

Base 64 Karakter Sayısı: 5728

Tip: JPEG

Uzunluk: 50x29

Algoritma	Şifreleme	Çözme	Veri Artışı		Güven Oranı
			Şifreleme	Çözme	
Sezar	Uzunluk: 50x29 Boyut: 1.85 KB	Uzunluk: 58x58 Boyut: 1.66 KB	0.75 KB	0.56 KB	%100
Vigenere	Uzunluk: 58x58 Boyut: 1.83 KB	Uzunluk: 58x58 Boyut: 1.66 KB	0.73 KB	0.56 KB	%100
Vernam	Uzunluk: 58x58 Boyut: 1.83 KB	Uzunluk: 58x58 Boyut: 1.66 KB	0.73 KB	0.56 KB	%99
Hill	Uzunluk: 58x58 Boyut: 1.85 KB	Uzunluk: 58x58 Boyut: 1.66 KB	0.75 KB	0.56 KB	%100
Playfair	Uzunluk: 58x58 Boyut: 1.82 KB	Uzunluk: 58x58 Boyut: 1.66 KB	0.72 KB	0.56 KB	%97

Hill Şifreleme algoritmasında şifrelenen görüntünün tekrar elde edilmesi ve oranına olan güven %100 iken Vernam ve Playfair algoritmalarına bu oran %99 ve %97'dir.

Özetlemek gerekirse Hill Şifreleme algoritması daha fazla bellek ve zaman harcama dezavantajına sahipken daha az işlemci kullanır. Ek olarak Hill Şifreleme algoritmasında frekans, kasiski testi, çakışma indeksi gibi şifre çözme adımları işe yaramaz. Çünkü aynı karakter farklı sayılarla ifade edilir.

Algoritmanın Kodu:

(Yazdığımız kod algoritmanın temel mantığını anlatmak için yazılmıştır. Sorunsuz şekilde çalışan bir program elde etmek için kullandığınız dilin özelliklerine göre düzenlenmeli ayrıca hata vermemesi adına gerekli kontroller yapılmalıdır.)

```
float keyA[2][2] = {{1 , 3 },{8 , 13}};
char harftablosu[26] = {A, B, C, D, E, F, ... X, Y, Z};
char metin[12] = "YILDIZTEKNIK";
char sifrelimetin[12];
int sifrelimetinsayisal[12];
int sayisal[2];
harfsayisi = metindeki karakter sayisi;
n = matris boyutu;
```

```
m = 0;
if (harfsayisi % n != 0 ){
        for (i = harfsayisi; i<harfsayisi%n+harfsayisi ; i++){
                metin[i] = 'A';
        }
        harfsayisi = harfsayisi + harfsayisi%n;
}
```

Extra Harf Ekleme

```
for (i = 0 ; i < harfsayisi ; i=i+1){
        j=0;
        k=0;
        while ( k < n ){
                if ( harftablosu[j] == metin[i] ){
                        sayisal[k] = j;
                        k=k+1;
                        i=i+1;
                        j=0;
                }else{
                        j=j+1;
                }
        }
```

Sayısal Karşılığını Bulma

```
for (j = 0 ; j<n ; j++){
        int x = 0;
        for (k = 0 ; k<n ;k++){
                x = x + sayisal[k]*A[k][j];
        }
        sifrelimetinsayisal[m]=x % 26;
        m=m+1;
        }
}
```

Matris Çarpımı ve Mod Alma

```
for (i = 0 ; i > harfsayisi; i=i+1){
        sifrelimetin[i] = harftablosu[sifrelimetinsayisal[i]];
}
```

Sayısal Metni Harflere Çevirme

Kaynaklar

https://www.researchgate.net/publication/309435285_KLASIK_KRIPTOLOJI_YONTEMLERININ_KARSILASTIRILMASI KLASİK KRİPTOLOJİ YÖNTEMLERİNİN KARŞILAŞTIRILMASI | Zainab Obaid, Arkan Saboncchi, Bahriye Akay, Erciyes University

http://static.dergipark.org.tr/article-download/b085/3f50/6ee8/imp-JA54BZ36FZ-0.pdf? Modern Blok Şifreleme Algoritmaları doi: 10.17932/ IAU.IAUD.m.13091352.2015.7/26.15-21 Fatih ŞAHİN

https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/07/%C5%9Fifreleme-y%C3%B6ntemleri İTÜ Bilgi İşlem Daire Başkanlığı | Şifreleme Yöntemleri

https://www.researchgate.net/publication/328693056_Application_of_Hill_Cipher_Algorithm_in_Securing_Text_Messages Application of Hill Cipher Algorithm in Securing Text Messages | Muhammad Donni Lesmana Siahaan, Andysah Putera Utama Siahaan, Faculty of Science and Technology, Universitas Pembangunan Panca Budi, Medan, Indonesia

http://static.dergipark.org.tr/article-download/d71c/a751/9924/5db2a8d33216c.pdf? ATIF/REFERENCE: Beşkirli, A., Özdemir, D. & Beşkirli, M. (2019). Şifreleme Yöntemleri ve RSA Algoritması Üzerine Bir İnceleme. European Journal of Science and Technology, (Special Issue), 284-291.

https://en.wikipedia.org/wiki/Hill_cipher Hill Cipher | Wikipedia

http://bilgisayarkavramlari.sadievrenseker.com/2008/11/19/hill-sifrelemesi-hill-cipher/ Şadi Evren Şeker | Hill Şifrelemesi

https://www.youtube.com/watch?v=2uungCVsd1c Şadi Evren Şeker | Asimetrik Şifreleme Youtube

https://ferdigul.com/hill-cipher/ Ferdi Gül | Hill Cipher

https://www.turcademy.com/tr/kitap/algoritma-gelistirme-ve-programlamaya-giris-9789750244308 Algoritma Geliştirme ve Programlamaya Giriş | Fahri Vatanseven

https://kerteriz.net/modern-sifreleme-yontemleri-simetrik-asimetrik-sifreleme/ Modern Şifreleme Yöntemleri (Simetrik ve Asimetrik Şifreleme) | Kerteriz

https://www.binance.vision/tr/security/symmetric-vs-asymmetric-encryption Biance Academy | Simetrik ve Asimetrik Şifreleme Karşılaştırması

http://acikerisimarsiv.selcuk.edu.tr:8080/xmlui/handle/123456789/10812 Simetrik ve Asimetrik Şifreleme Algoritmalarının Karşılaştırılması | Halife KODAZ, Fatih M. BOTSALI, Selçuk-Teknik Dergisi ISSN

Ek olarak Yıldız Teknik Üniversitesi Matematik Bölümü öğretim üyelerinden,

Arş. Gör. Dr. İsmail AYDOĞDU'ya yardımlarından dolayı teşekkür ederiz.

Görev Dağılımı

Rapor:

Şifreleme Algoritmalarına Genel Bakış – Emre Arslanoğlu

Hill Şifreleme Algoritmasının Tarifi – Ali Emre Nebiler

Algoritmanın Karmaşıklığı – Emre Arslanoğlu

Kısıtlar ve Dikkat Edilmesi Gereken Noktalar – Ali Emre Nebiler

Uygulama Alanları – Ali Emre Nebiler

Rakipler ve Kıyaslama – Emre Arslanoğlu

Algoritmanın Kodu – Emre Arslanoğlu

Görseller – Ali Emre Nebiler

Video:

Simetrik ve Asimetrik Şifreleme – Ali Emre Nebiler

Hill Algoritmasında Şifreleme – Ali Emre Nebiler

Hill Algoritmasında Deşifreleme – Emre Arslanoğlu

Algoritmanın Kodu ve Karmaşıklığı – Emre Arslanoğlu

Rakipler ve Kıyaslama – Emre Arslanoğlu

Video Editleme – Ali Emre Nebiler