İSTANBUL TİCARET ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİLGİSAYAR SİSTEMLERİ LABORATUVARI

ŞİFRELEME, ŞİFRE ÇÖZME VE ŞİFRE KIRMA

1. DENEYİN AMACI

Bu deney, gizliliğin ve güvenliğin sağlanması için bir düz metinin üçüncü kişilerin anlayamayacağı biçime nasıl dönüştürülebileceğini (şifreleme), alıcı tarafın şifrelenmiş metni nasıl anlaşılır biçime sokabileceğini (şifre çözme) ve üçüncü kişinin (düşman) büyük emek sarf ederek dahi olsa, şifreyi nasıl kırabileceğini, yani anahtarı nasıl elde edebileceğini, klasik kripto sistemler üzerinde açıklamayı amaçlar. Klasik şifreleme yöntemleri basit olduğundan, burada şifrelemeye fazla ağırlık verilmemiştir. Ama şifre kırma, yani anahtarın eldesi genellikle zor bir işlem olduğundan burada üzerinde durulan en önemli konu olmuştur. Şifreleme için Vegenere yöntemi kullanılacaktır. Şifrenin kırılmasında ise harflerin bir dilde kullanılma olasılığından ve korelasyon teoreminden yararlanılacaktır.

Mesajların düşmanlara anlaşılmaz yapılması, tarih boyunca çok önemli olmuştur. Burada bilgisayarın icadından önce kullanılan bazı eski kripto sistemler anlatılacaktır. Bu sistemler bugün kullanılamayacak kadar zayıftır, ama kriptolojinin bazı önemli yönlerini çok iyi gösterirler. Bu basit kripto sistemler için aşağıdaki kabuller yapılacaktır.

- Düz-metin küçük harflerle, şifreli metin ise büyük harflerle yazılacaktır.
- Alfabedeki harflere aşağıdaki gibi sayılar atanacaktır.

a	b	c	d	e	f	g	…	v	w	x	y	z
0	1	2	3	4	5	6	…	21	22	23	24	25

* Boşluklar ve noktalama işaretleri ihmal edilecektir. Kriptolojinin esası sayı teorisine dayandığı için bu teorinin önceden öğrenilmesinde yarar vardır.

2. KAYDIRMA ŞİFRELEMESİ

Bu şifrelemede her harf aşağıdaki bağıntı ile bir miktar kaydırılarak şifreli metin elde edilir. Buna Sezar şifreleme denir. p düz-metin, C şifreli-metin ve k anahtar olmak üzere aşağıdaki özdeşlik ile şifreli metin üretilir.

$C \equiv p + k \pmod{26}$

Şifreli metinden düz-metine geçiş de yine benzer bir özdeşlikle aşağıdaki gibi yapılabilir.

$p \equiv C - k \pmod{26}$

3. AFİN ŞİFRELEME

Kaydırma şifrelemesi aşağıdaki gibi genellenip biraz da güçlendirilebilir. gcd (a,26)=1 olacak şekilde bir a ve bir de b tamsayısı seçilirse, aşağıdaki afin fonksiyon şifreleme yapmak amacıyla kullanılabilir. Burada a ve b tamsayıları anahtar kabul edilir ve 12*26=312 anahtar seçeneği sunarlar. (mod26 kullanıldığından dolayı, $0 \leq a,b \leq 26$ olduğu düşünülür.)

$C \equiv ap + b \pmod{26}$

afin $\Rightarrow$ CVVWPM (a=9, b=2)

4. VİGENERE ŞİFRELEME

Kaydırma şifrelemesinde önemli bir değişiklik 16. yüzyılda Vigenere'den geldi. Bu kripto sistem uzun yıllar çoğu kişi tarafından güvenli bulundu, ama 19. Yüzyılda Babbage bu sisteme bir atak gerçekleştirdi. İşte bu deneyin amacı da bu atağın gerçekleştirilmesinde kullanılan matematiksel yöntemleri tanıtmaktır.

Vigenere enkripsiyonun anahtarı aşağıdaki gibi seçilen bir vektördür. Önce bir anahtar uzunluğu, örneğin 6, seçilir. Daha sonra elemanları 0-25 arasında değerler alabilen bu boyutta bir anahtar vektörü alınır, örneğin k=(21,4,2,19,14,17). Genellikle bu anahtar kolayca hatırlanabilen bir kelimeye karşı düşürülür, buradaki k anahtarı "**vecto**r" sözcüğüdür. Sistemin güvenliği hem anahtar kelimesinin uzunluğunun hem de kendisinin bilinmemesine dayanır.

Örnekteki k anahtarı kullanılarak mesajı şifrelemek için önce düz-metnin birinci harfi alınır ve 21 kaydırılır. Daha sonra ikinci harf alınır ve 4 kaydırılır, üçüncü harf 2 kaydırılır, ve bu şekilde anahtar elemanlarının gösterdiği değerler kadar kaydırmaya devam edilir. Anahtar elemanlarının sonuna gelindiği zaman anahtarın ilk elemanından tekrar başlanır. Enkripsiyon sürecinin diyagramı Şekil 1'de verilmiştir.

Düz-metin	h	e	r	e	i	s	h	o	w	i	t	w	o	r	k	s
Anahtar	21	4	2	19	14	17	21	4	2	19	14	17	21	4	2	19
Şifreli-metin	C	I	T	X	W	J	C	S	Y	B	H	N	J	V	M	L

Şekil 1. *Vigenere Şifreleme*

Eğer yeterli sayıda karakter biliniyorsa bilinen düz-metin atağı başarılı olacaktır, çünkü düz-metinden şifreli-metin mod26'ya göre çıkarılarak anahtar elde edilebilir. aaaaaaa ….. düz-metnini kullanan seçilmiş düz-metin atağı hemen anahtarı verecektir, buna rağmen seçilen AAAAA.. şifreli-metin atağı bu anahtarın negatifini verecektir. Yalnız şifreli-metinin olması durumunda bu yöntemin yalnız şifreli-metin atağına karşı güvenli olduğu uzun süre düşünüldü. Ama bu durumda da anahtarı bulmak kolaydır.

Kripto analiz, İngilizce metinlerdeki harflerin frekanslarının eşit olmamasından faydalanır. Örneğin e harfi, x harfinden daha sık tekrarlanır. Harflerin frekansları Tablo 1'de verilmiştir.

Tablo 1. *İngilizcede harflerin frekansları*

a	b	c	d	e	f	g	h	i	J
.082	.015	.028	.043	.127	.022	.020	.061	.070	.002
k	l	m	n	o	p	q	r	s	t
.008	.040	.024	.064	.075	.019	.001	.060	.063	.091
u	v	w	x	y	z				
.028	.010	.023	.001	0.020	.001				

Eğer basit bir kaydırma şifrelemesi kullanılsaydı, e harfine karşı düşen harf şifreli metinde en çok karşılaşılan harf olacaktı. Bu yüzden frekans analizi muhtemel anahtarı ortaya çıkaracaktı. Ama önceki Vigenere şifreleme örneğinde, e harfi I ve X harfi olarak görünür. Daha uzun düz-metin kullanılsaydı, anahtar elemanlarının değerine bağlı olarak e harfi başka harflere dönüştürülmüş olarak görünecekti. Bu yüzden frekans sayısından bir şeyler çıkarmak çok zorlaşır. Daha doğrusu frekans sayıları genelde birbirine yaklaşır ve frekans eğrisi düzgün biçim almaya başlar, yani şifreli-metinin her bir harfi için olasılık 1/26 değerine yaklaşır.

Şekil 2'de bir şifreli metin ve Tablo 2'de de bu metindeki karakterlerin frekansları verilmiştir. Şimdi bu metnin şifresinin nasıl çözülebileceğini inceleyeceğiz. Bu iş iki adımdan oluşur: Birincisi anahtar uzunluğunun bulunması ve ikincisi de anahtar elemanlarının bulunmasıdır.

VVHQWVVRHMUSGJGTHKIHTSSEJCHLSFCBGVWCRLRYQTFSVGAHW
KCUHWAUGLQHNSLRLJSHBLTSPISPRDXLJSVEEGHLQWKASSKUWE
PWQTWVSPGOELKCQYFNSVWLJSNIQKGNRGYBWLWGOVIOKHKAZKQ
KXZGYHCECMEIUJOQKWFWVEFQHKIJRCLRLKBIENQFRJLJSDHGR
HLSFQTWLAUQRHWDMWLGUSGIKKFLRYVCWVSPGPMLKASSJVOQXE
GGVEYGGZMLJCXXLJSVPAIVWIKVRDRYGFRJLJSLVEGGVEYGGEI
APUUISFPBTGNWWMUCZRVTWGLRWUGUMNCZVILE

Şekil 2. *67 karakterden oluşan bir şifreli metin örneği*

Tablo 2. *Şekil 2'de verilen örnek şifreli metindeki karakterlerin frekansları*

A	B	C	D	E	F	G	H	I	J	K	L	M
8	5	12	4	15	10	27	16	13	14	17	25	7
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
7	5	9	14	17	24	8	12	22	22	5	8	5

5. ANAHTAR UZUNLUĞUNUN BULUNMASI

Anahtar uzunluğunun bulunması için şifreli metin uzun bir şerit üzerine yazılır ve daha sonra bu şeridin bir kopyası oluşturulur. Orijinal şerit bir miktar ötelenerek kopyası ile üst üste getirilir (muhtemel anahtar uzunluğu kadar). İki karakterlik kaydırma yapılırsa Şekil 3'teki durum elde edilir.

		V	V	H	Q	W	V	V	R	H	M	U	S	G	J	G
V	V	H	Q	W	V	V	R	H	M	U	S	G	J	G *	T	H
T	H	K	I	H	T	S	S	E	J	C	H	L	S	F	C	B
K	I	H	T	S	S	E	J	C	H	L	S	F	C	B	G	V
G	V	W	C	R	L	R	Y	Q	T	F	S	V	G	A	H	…
W	C	R	L	R *	Y	Q	T	F	S	V	G	A	H	W	K	…

Şekil 3. *Şifreli metin ve kopyasının kaydırılarak çakıştırılması*

Üst üste gelen (çakışan) harflerden aynı değerli olanlar (çakışanlar) sayılır. Şekil 3'te listelenen metinde çakışan karakterlerin sayısı ikidir. Eğer Şekil 2'deki tüm şifreli metin için bu işler yapılsaydı 14 sayısı bulunacaktı. Farklı miktarda ötelemeler için Tablo 3'deki değerler elde edilir.

Tablo 3. *Farklı ötelemeler için çakışma sayıları*

Öteleme	1	2	3	4	5	6
Çakışma	14	14	16	14	24	12

En büyük çakışma 5 ötelemesi için olur ve 24 olarak bulunur. Bu değer, anahtar uzunluğu için en iyi tahmindir. Bu yöntem bilgisayar olmadan da hızlı olarak çalışır ve anahtar uzunluğunu da oldukça sağlıklı şekilde verir.

6. ANAHTAR ELEMANLARININ BULUNMASI

Örnekte olduğu gibi, şimdi anahtar uzunluğunun 5 olarak belirlendiği varsayılsın. Anahtarın ilk elemanını bulmak için, 1., 6., 11., ….. konumlarda bulunan harflerin frekansları sayılır (Tablo 4) ve bir **v** vektörüne yerleştirilir. Şekil 2'deki şifreli metindeki harflerin toplam sayısı 67'dir. Eğer her bir harfin sayısı bu toplam sayıya oranlanırsa aşağıdaki **w** vektörü bulunur.

Tablo 4. *Örnek şifreli metinin 1., 6., 11., …. Konumlarındaki harflerin sıklıkları.*

A	B	C	D	E	F	G	H	I	J	K	L	M
0	0	7	1	1	2	9	0	1	8	8	0	0
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
3	0	4	5	2	0	3	6	5	1	0	1	0

v=(0, 0, 7, 1, 1, ……., 1, 0)

w=(0, 0, .1045, .0149, .0149, .0299, …, .0149, 0)

İngilizce harflerin frekansları bir vektör şeklinde düzenlenirse aşağıdaki $\mathbf{A}_0$ vektörü elde edilir (Tablo 1'e bakınız). $0 \leq i \leq 25$ olmak üzere

$\mathbf{A}_0$ = (.082, .015, .028, …, .020, .001)

$\mathbf{A_0}$ vektörünün i eleman sağa kaydırılmışı $\mathbf{A}_i$ ile gösterilirse, $\mathbf{A}_{i=2}=\mathbf{A}_2$ vektörü aşağıdaki gibi yazılabilir.

$\mathbf{A_2}$=(.020, .001, .082, .015, ……)

$\mathbf{A_0}$'ın kendisi ile skaler çarpımı veya $\mathbf{A}_i$'nin $\mathbf{A}_i$ ile skaler çarpımı hep aynı sonucu verir.

$\mathbf{A_0}.\mathbf{A_0} = (.082)^2+(.015)^2+\ldots=.066$

Ama i≠j olduğu zaman $\mathbf{A}_i.\mathbf{A}_j$ skaler çarpımı daha küçük sonuç verir.

Tablo 5. *Farklı miktarda kaydırılmış* $\boldsymbol{A_0}$ *vektörleri ile* $\boldsymbol{A_0}$*'ın skaler çarpımları.*

$\lvert i-j \rvert$	0	1	2	3	4	5	6
$\mathbf{A}_i.\mathbf{A}_j$	.066	.039	.032	.034	.044	.033	.036
$\lvert i-j \rvert$	7	8	9	10	11	12	13
$\mathbf{A}_i.\mathbf{A}_j$	.039	.034	.034	.038	.045	.039	.042

0≤i≤25 olmak üzere $\mathbf{w}.\mathbf{A}_i$ skaler çarpımları hesaplanırsa, maksimum değer i'nin doğru değerinden gelmelidir. Bu skaler çarpımlar Tablo 6'da verilmiştir.

Tablo 6*.* $\boldsymbol{w.A_i}$ *skaler çarpımları*

.0250, .0391, .0713, .0388, .0275, .0380, .0512, .0301, .0325,
.0430, .0338, .0299, .0343, .0446, .0356, .0434, .0502,
.0392, .0296, .0326, .0392, .0366, .0316, .0488, .0349

Bu tabloda en büyük değer üçüncü değerdir, yani $\mathbf{w}.\mathbf{A}_2$ = .073 değeridir. Bundan dolayı birinci ötelemenin 2 olduğu tahmin edilir ve bu da 'c' anahtar harfine karşı düşer.

Anahtarın üçüncü elemanını hesaplamak için aynı yöntem kullanılabilir. Daha önce tablo halinde verilen 3., 8., 13., … harflere ilişkin frekanslar kullanılarak yeni bir **w** vektörü hesaplanır.

w = (0, .0152, 0, .0454, .0454, 0.152, …, 0, .0152)

0≤i≤25 olmak üzere $\mathbf{w}.\mathbf{A}_i$ skaler çarpımşları Tablo 7'de verilmiştir.

Tablo 7*.* $\boldsymbol{w.A_i}$ *skaler çarpımları*

.0372, .0267, .0395, .0624, .04741, .0279, .0319, .0504, .0378,
.0351, 0.367, 0,395, .0264, .0415, .0427, .0362, .0322, .0457
.0526, .0397, .0322, .0299, .0364, .0372, .0352, .0406

Bu tabloda en büyük değer dördüncü değerdir, yani $\mathbf{w.A}_3$ =.0624 değeridir. Bundan dolayı en iyi tahmindir, bu da d anahtar harfine karşı düşer. Anahtarın kalan üç elemanı da aynı yöntemle bulunursa anahtar harfleri c, o, d, e, s olarak ortaya çıkar.

En büyük skaler çarpımın, her iki durumda da diğerlerinden önemli derecede büyük olduğuna dikkat edilmelidir. Bu yüzden doğru olanı bulmak için birkaç tahmin yapılmasına gerek kalmamıştır. Bu sebeple bu yöntem diğerlerine göre daha üstündür.

7. DENEYİN YAPILIŞI

7.1. Deneyde Kullanılacak dosyalar

Sifrele.cpp, Anahtarbul.cpp, ve Ornek_Metin.txt isimli üç dosya

7.2. Vigenere şifreleme yöntemi ile bir düz metnin şifrelenmesi

Aşağıdaki tabloya yaklaşık 20 karakterlik bir düz metin ve 5-10 karakterlik bir anahtar yazınız. Ardından Vigenere şifreleme yöntemi ile şifrelenmiş metni oluşturunuz.

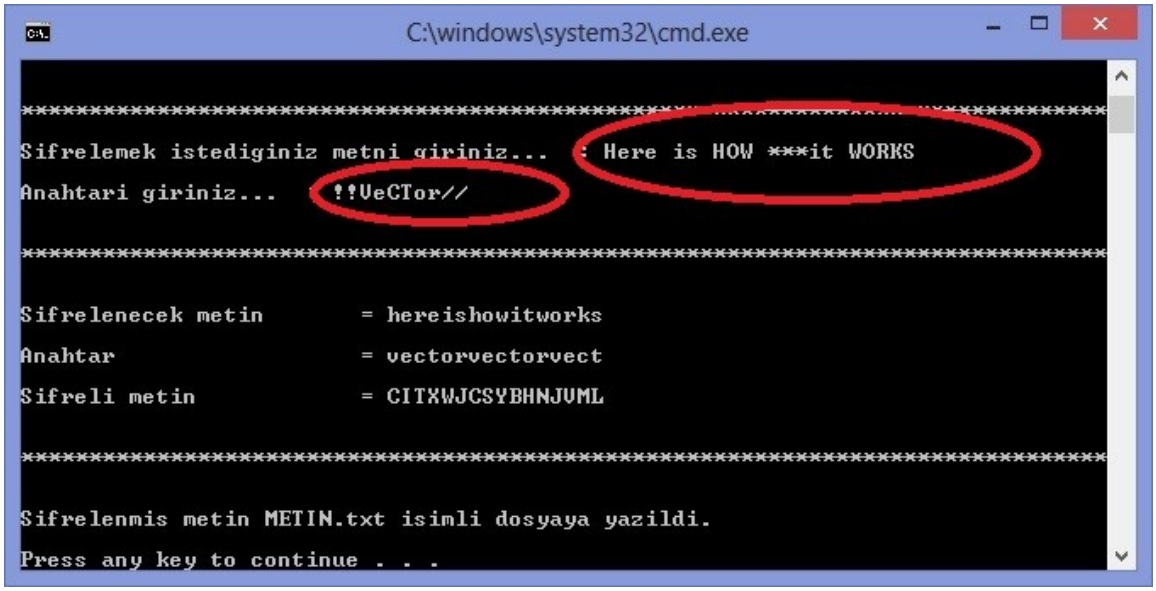
Düz metin

Anahtar

Şifreli metin

Sifrele.cpp programını bir **C** derleyicisi kullanarak çalıştırınız. Yukarıdaki verileri giriniz ve programın bulduğu sonuçları kendi bulduğunuz sonuçlar ile karşılaştırınız.

Sifrele.cpp programı çalıştırıldığında program şifrelenecek metni ve anahtarı klavyeden okur, ve varsa metin içindeki boşluk ve özel karakterleri çıkararak şifrelenecek metin ve anahtarın sadece İngiliz alfabesindeki küçük harflerden oluşmasını sağlar. Dönüşümler için ASCII tablosundan faydalanılır. Metin içinde Türkçe'ye özgü karakterler varsa atılır. Düz metnin ve anahtarın 97-122 arasında ASCII kodlarına sahip '**a**'-'**z**' arası karakterlerden oluşması garanti edildikten sonra şifreleme işlemi gerçekleştirilir.



ÖRNEK PROGRAM ÇIKTISI

Verilen **"Ornek_Metin.txt"** dosyasını bir text editöründe açınız. Bu text dosyasında yer alan ilk düz metni **"theory"** anahtar kelimesi ile şifreleyiniz. Şifreli metin **"Metin.txt"** dosyasına yazılacaktır. Bu dosyanın oluşup oluşmadığını programın bulunduğu dizinden kontrol ediniz.

Sifrele.cpp dosyasında bulunan kodları inceleyiniz.

7.3. Anahtar yardımıyla şifreli metinden düz metne dönüşüm

"Sifrele.cpp" dosyasını farklı kaydedip program üzerinde gerekli güncellemeleri yapınız, ve Vigenere Şifrelemede şifreli metnin bilinen bir anahtar ile düz metne dönüştürülebilmesi için program üzerinde gerekli modifikasyonları yapınız.

Hazırladığınız dosyayı mckasapbasi@ticaret.edu.tr adresine gönderiniz. Gönderilen dosyalar posta kutusuna ulaşma zamanına ve programın çalışıp çalışmamasına göre değerlendirilecektir. Posta kutusuna ilk ulaşan ve çalışan 5 gruba ait maile not verilecektir. Her gruptan yalnız 1 adet mail gönderilmelidir. NOT: Mailler gelme anında değerlendirilir.

Programınızı test etmek için yaklaşık 20 karakterlik bir şifreli metin ve 5 karakterlik bir anahtar giriniz. Ardından düz metni oluşturunuz.

Şifreli metin

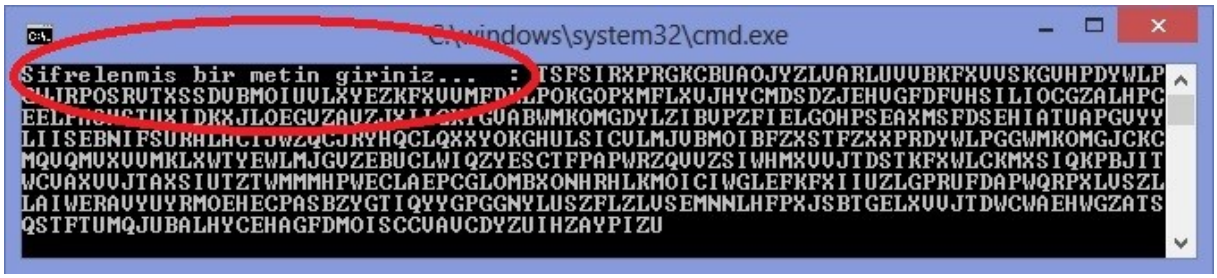
Anahtar

Düz metin

Yukarıdaki verileri ve üzerinde modifikasyon yaptığınız programın bulduğu sonuçları kendi bulduğunuz sonuçlar ile karşılaştırınız.

7.4. Anahtar bilinmeden şifreli metinden düz metnin bulunması

C dilinde yazılmış **Anahtarbul.cpp** programını çalıştırınız. Daha önce **Metin.txt** dosyasında yarattığınız şifreli metne ait karakterleri programa giriniz.



7.4.1 Anahtar Uzunluğunun Bulunması

Her seferinde bir karakter olmak üzere şifreli metni 10 karaktere kadar kaydırılıp, her kaydırma için karakterlerin çakışma miktarlarını aşağıdaki tabloya kaydediniz.

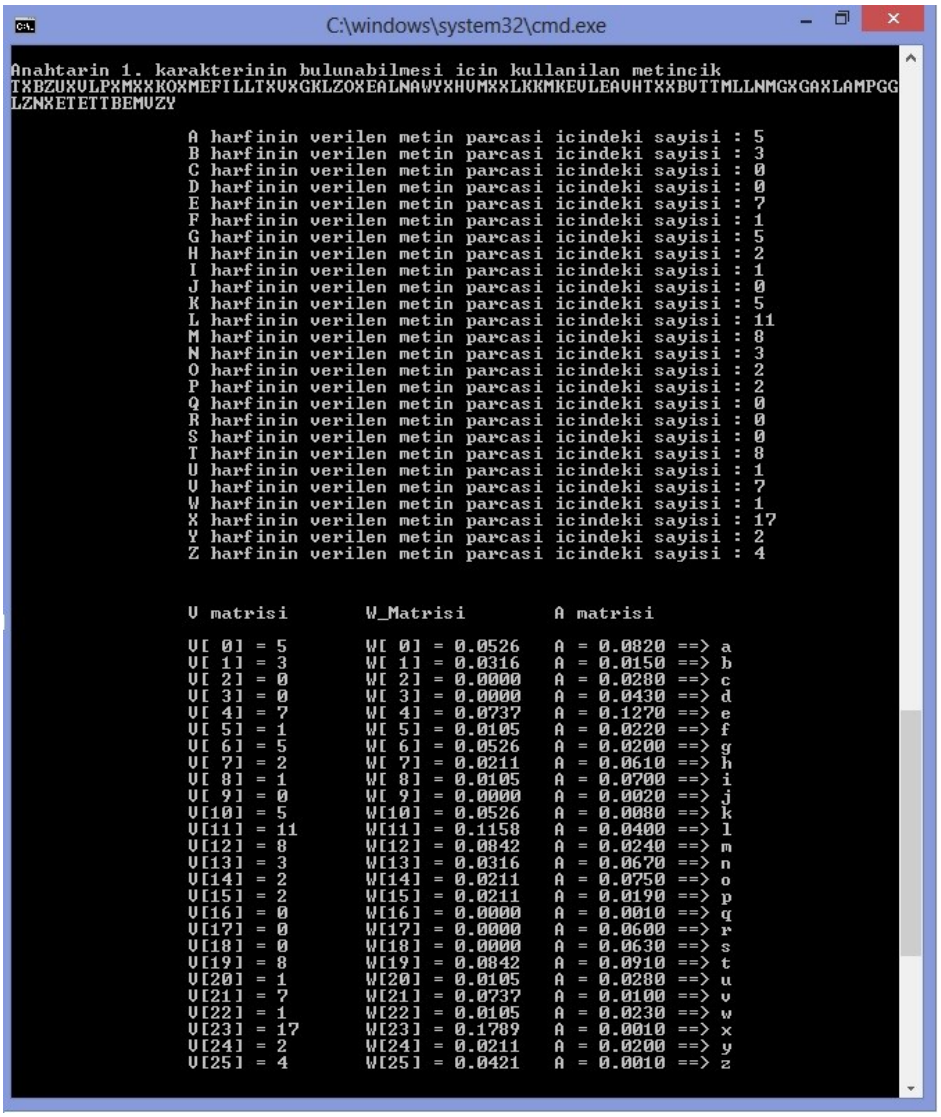
-

Öteleme	1	2	3	4	5	6	7	8	9	10
Çakışma										

E

En fazla eşleşme …… kaydırma ile elde edildiğinden anahtar …… karakterden ibarettir.

7.4.2 Anahtar karakterlerinin Bulunması



Anahtarın 1. karakterinin bulunması için yukarıdaki matrislerin skaler çarpımlarından elde edilen değerleri aşağıya kaydediniz.

En büyük $\mathbf{w}.\mathbf{A}_i$ skaler çarpım değeri : ……..

Bu değerin dizideki indeksi : ……..

Bu değere karşılık gelen karakter : ……..

Anahtarın diğer karakterleri için de aynı işlem yapıldığında bu örnek için bulunan anahtar harflerini aşağıya yazınız.

ANAHTAR :

Bu kez şifreli metnin yarısını deneyerek anahtar kelimenin ne olduğunu bulmaya çalışınız. Sonra tekrar şifreli metni küçültünüz.

(Şifreli Metin / 2) uzunluğundaki veri için bulunan anahtar uzunluğu : ……

ANAHTAR

(Şifreli Metin / 4) uzunluğundaki veri için bulunan anahtar uzunluğu : ……

ANAHTAR

(Şifreli Metin / 8) uzunluğundaki veri için bulunan anahtar uzunluğu : ……

ANAHTAR

Eğer değerlerini karşılaştırınız. Bulunan anahtarları kendi yazdığınız programda deneyiniz.

EK :

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

128	Ç	144	É	160	á	176	▒	192	└	208	╨	224	α	240	≡			
129	ü	145	æ	161	í	177	▓	193	┴	209	╤	225	ß	241	±			
130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥			
131	â	147	ô	163	ú	179	|	195	├	211	╙	227	π	243	≤			
132	ä	148	ö	164	ñ	180	┤	196	—	212	╘	228	Σ	244	⌠			
133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡			
134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷			
135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈			
136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°			
137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙			
138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·			
139	ï	155	¢	171	½	187	╗	203	╦	219	█	235	δ	251	√			
140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	n			
141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²			
142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■			
143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255				

Source: www.LookupTables.com