T.C.

BEYKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ

VİDEO VERİSİ ÜZERİNDE GÖRSEL GÜVENLİK MODELİNİN GELİŞTİRİLMESİ

(Yüksek Lisans Tezi)

Tezi Hazırlayan :

MAHMUT TÜLÜ

İstanbul, 2016

T.C.

BEYKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ

VİDEO VERİSİ ÜZERİNDE GÖRSEL GÜVENLİK MODELİNİN GELİŞTİRİLMESİ

(Yüksek Lisans Tezi/Projesi)

Tez/Projeyi Hazırlayan:

MAHMUT TÜLÜ

Öğrenci No:

140820002

DANIŞMAN:

Yrd.Doç.Dr. Ediz ŞAYKOL

İstanbul, 2016

YEMIN METNI

Yüksek lisans tezi olarak sunduğum “Görsel Güvenlik Sisteminin Geliştirilmesi” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmanın içinde kullanıldıları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım.

(imza)

Aday: MAHMUT TÜLÜ

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi.1408222002.no'lu MAHMUT TÜLÜ'nün ..... 'in 11.5.2016 tarihinde yapılan tez savunma sınavı[1] sonucunda...30...dakika süreyle sunduğu ve savunduğu tezi hakkında[2] oybirliğiyle,KABUL...kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : ....BİLGİSAYAR MÜHENDİSLİĞİ
Programı : ....BİLGİSAYAR MÜHENDİSLİĞİ
Tez Başlığı[3] : .....VİDEO VERİSİ ÜZERİNDE GÖRSEL GÜVENLİK MODELİNİN GELİŞTİRİLMESİ

Tez Sınav Jürisi	Öğretim Üyesi	İmza
Danışman	: Yrd. Doç. Dr. Ediz ŞAYKOL	
Üye	: Doç. Dr. Gökhan SİLAHTAROĞLU	
Üye	: Doç. Dr. Kemal SAFİ	

---

[1] Jüri üyeleri söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez savunma sınavına alır. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda jüri en geç onbeş gün içinde toplanarak adayı tez savunma sınavına alır. Tez savunma sınav süresi en az 45 dakikadır. Yüksek lisans tez savunma sınavı, tez çalışmasının sunulması ve bunu izleyen soru-yanıt bölümlerinden oluşur ve dinleyiciye açıktır. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-3)

[2] Tez sınavının tamamlanmasından sonra jüri, tez hakkında "kabul", "düzeltme" veya "red" kararı verir. Jüri başkanı, jüri üyelerince imzalanmış sınav tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi başarısız bulunan öğrencinin Enstitü ile ilişiği kesilir. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve yönetmelikte belirtilen usullere uygun olarak tezini aynı jüri önünde yeniden savunur. Bu savunma sınavında da tezi kabul edilmeyen öğrencinin enstitü ile ilişiği kesilir. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-4)

[3] İleride doğabilecek aksaklıkların engellenmesi için tezin başlığının yazılması gerekmektedir.

VİDEO VERİSİ ÜZERİNDE GÖRSEL GÜVENLİK MODELİNİN GELİŞTİRİLMESİ

Tezi Hazırlayan:

MAHMUT TÜLÜ

Özet

Bu çalışmamızda simetrik şifreleme algoritmalarını kullanarak mp4 formatındaki videoların şifreleyerek avi formatında oluşturmak ve simetrik şifre ile şifrelenmiş avi formatındaki videoların orjinal mp4 formatındaki haline nasıl dönüştürüleceğini anlatılmıştır. Ayrıca simetrik şifreleme algoritmalarının farklı modlarda çalışabilen, video tabanlı şifreleme ve deşifreleme imkanı sağlayan bir modellenmesi gerçekleştirilmiştir. Gerçekleştirilen bu modellemede programlama dili olarak C# ve Emgu-cv seçilmiştir.

Anahtar Kelimeler: Simetrik Şifreleme Algoritmalar, Video Şifreleme

A VISUAL PRIVACY MODEL FOR VIDEO DATA

Presented By :

MAHMUT TÜLÜ

Abstract

In this thesis, the steps of converting mp4 formatted videos into encrypted avi formatted videos by using symmetric encryption algorithms and also the steps of converting encrypted avi formatted videos (by using symmetric encryption algorithms) into decrypted (original) mp4 formatted videos has been worked and explanied in details. Furthermore, a modelling of this symmetric encryption algorithms, which operates in different modes and gives permission to the use of video based encryption and decryption, has been implemented. In this modelling, C# and Emgu-cv was chosen as the programming language

Key Words: Symmetric encryption algorithms, Video encryption

RESİMLER LİSTESİ

ŞEKİLLER LİSTESİ

GİRİŞ

A. AMAÇ

Son zamanlarda insanların kendilerini güvende hissetmek için evlerine, işyerlerine daha birçok yere görsel güvenlik sistemi olan kameraları bağlamaktadırlar. Bu kameralardan gelen görüntüler bir makinaya kaydedilmektedir. Bu makinaya kimi zaman yetkisiz kişilerin ulaşabileceği bilinmektedir. Bu tezimizde bu makinalara gelen görüntülerin ya da kişisel videolarımızın görsel güvenliği arttırmak adına bilinen resim şifreleme yöntemlerinin dışında metin ile resmin uygulama farklılıklarına rağmen metin için olan simetrik şifreleme algoritmalarını kullanacağız.

Videoların mp4 formatında alınarak seçeceğimiz simetrik şifreleme algoritmalarından birine bağlı olarak istenen şifreleri vererek bütün görüntüleri şifreleme ya da belirli bir alanı şifrelemesini sağlayacağız. Şifrelenen bu görüntüler avi formatında video olarak kaydedilecektir. Avi formatında olan şifreli videolarımızın istediğimiz zaman şifrelerini girerek orjinal mp4 formatına çevrilmesini sağlayacağız.

B. LİTERATÜR TARAMASI

[3],[5] numaralı çalışmalarda metin ile resim arasındaki farklılıklara dikkat çekilmiştir. Bu çalışmada bu farklılıklara göre görsel güvenlik sistemlerinin geliştirilmesinden yararlanılmıştır.

[1],[6],[8] numaralı tezlerde simetrik şifreleme algoritmalarına dikkat çekilmiştir. Bu çalışmada dikkat çekilen konular incelenerek simetrik şifreleme modelleri hazırlanmıştır.

BÖLÜM 1

ŞİFRELEME VE SİMETRİK ŞİFRELEME SİSTEMLERİ

1. ŞİFRELEME

Sistemler arası bağlantıda gidip gelen verinin güvenli olması gerekir. Bunu sağlamak için şifrelenmiş veri olması gerekir. Bu şifreleme işi ile uğraşan bilim dalına Kriptoloji denir. Kriptoloji iki bölüme ayrılır: Kriptografi (Şifreleme) ve Kriptoanaliz (Şifre Çözme). Gönderilen verinin şifrelenmemiş haline plain text şifrelenmiş haline cipher text denir. Gönderilen şifrelenmiş verinin yetkisiz kişiler tarafından orjinal haline döndürülmemesi için şifreleme sırasında kullanılan tüm yöntem ve bilgilerin gizli kalmasına bağlıdır. Şifreleme sistemleri iki başlık altında incelenir:

- Simetrik Şifreleme Algoritmaları
- Asimetrik Şifreleme Algoritmaları

Asimetrik Şifreleme algoritmalarına bu çalışmamızda girmeyeceğiz.[8]

2. SİMETRİK ŞİFRELEME SİSTEMLERİ

"Blok şifrelerin tasarımı shannon tarafından ortaya atılan karıştırma ve yayılma tekniklerine dayanır. Karıştırma şifrenin tek doğrusal olmayan parçası S-Kutuları ile sağlanırken, yayılma bit tabanlı ya da byte tabanlı doğrusal dönüşümlere dayanır."[6]

2.1. DES Şifreleme Algoritması

Des şifreleme algoritması 1974 yılında IBM tarafından geliştirilen Amerika birleşik devletleri standartlar enstitüsü tarafından standartlaştırılan, küçük anahtar boyutuna sahip ve büyük anahtar boyutları için herhangi bir seçeneği olmayan, 64 bit metni 56 bit anahtar ile şifreleyen, şifreleme piyasasında yaygın kullanılan şifreleme sistemidir.[2][6]

Algoritmanın ilk kısmında açık metinin bit sıralamasının değiştirildiği giriş permütasyonu (Initial Permutation) $IP(x)=L^0R^0$ (x açık metin), sonunda da şifreli metinin bit sıralamasının değiştirildiği ve giriş permütasyonunun tersi olan çıkış permütasyonu(Final Permutation) $y=IP^{-1}(R^{16}L^{16})$ (y şifreli metin) bulunur. Bu permütasyonların DES'in güvenliğine herhangi bir katkısı bulunmamaktadır. DES feistel mimarisine sahiptir. Algoritmanın ana gövdesi, veriyi 32 bitlik sağ ve sol yan veri olarak yinelemeli çevrimlerden (round) oluşur. 16 döngüden oluşan algoritmanın temel işlemi, her döngüde sağ ve sol yan verileri ile anahtar düzenlemek algoritmasından elde edilen 48 bit anahtarlar kullanılarak yeni sağ ve sol yan verileri elde edilir. Her döngünün sırasıyla sağ yan verisi ile 48 bit anahtar kullanılarak bir f fonksiyonu hesaplanarak sol yan verisi ile XOR'lama işlemi yapılır. İlk döngü öncesi ile son döngü sonrası hariç iki döngü arasında verinin sol ve sağ yan verileri yer değiştirir. Aşağıda ifadelerde matematiksel anlatımı vardır.[2][6]

$R^{i-1}: \{0,1\}^{32}$

$K^{i:} \{0,1\}^{48}$

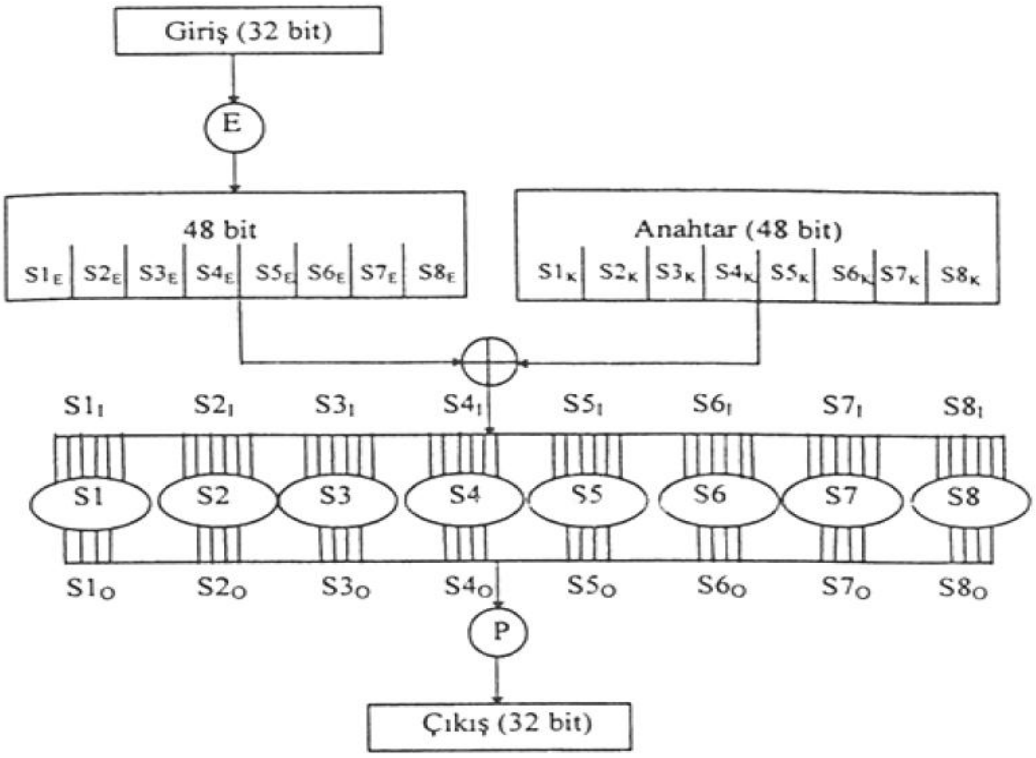
$f : R^{i-1} \times K^i \rightarrow \{0,1\}^{32}$

$R^i = L^{i-1} \oplus f(R^{i-1},K^i)$

DES'deki f fonksiyonun çalışma mantığı şu şekildedir[2][6]:

1. Gelen 32 bitlik sağ yan verisi 48 bit olacak şekilde genişletme (Expansion) permütasyonuna sokulur.
2. Üretilen 48 bitlik değer 48 bit anahtar ile XOR işlemine tabi tutulur.
3. Buradan elde edilen 48 bitlik değer her biri kendine özel tabloyu kullanan 6 biti 4 bite indirgeyen S-kutuları kullanılır.
4. S kutularından çıkan bitler sıralanarak 32 bit değer elde edilir. Bu 32 bit değer 32 bitlik P permütasyonuna sokularak 32 bit uzunluğunda yeni bir bit dizisi elde edilir.

Şekil 1 de 32 bit giriş sağ verisinin expansion alınarak 48 bit anahtar ile XOR işlemi ve S-Kutularından geçerek 32 bit değer ile permütasyonuna sokularak yeni bit dizisinin elde edilmesi gösterilmektedir.



Şekil 1 : DES'in F Fonksiyonu[2]

K anahtar bitlerinin elde edilmesi şu şekildedir[2][4]:

1. Başlangıçtaki anahtar bitleri, bit numarası sekize bölünenler (8,16,24,...) eşlik biti olacak şekilde 1'den 64'e kadar numaralandırılır.
2. Elde edilen anahtar bitleri Şekil 3'deki PC-1 permütasyonu kullanılarak yeniden sıralanarı ve 28 bitlik C ve D kaydedicilerine yüklenir.
3. Şekil 2'de göründüğü gibi her döngüde C ve D kaydedicileri bir ya da iki bit sola kaydırılır.
4. C ve D kaydediciler birleştirilerek 1'den 56'ya kadar numaralandırılır.
5. Şekil 4'deki PC-2 permütasyonuna sokularak 48 bit anahtar elde edilir.

Çevrim Numarası :	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Kaydırma Miktarı :	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Şekil 2 : Anahtar düzenleme algoritmasında döngü numarasına göre sola kaydırma sayıları

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

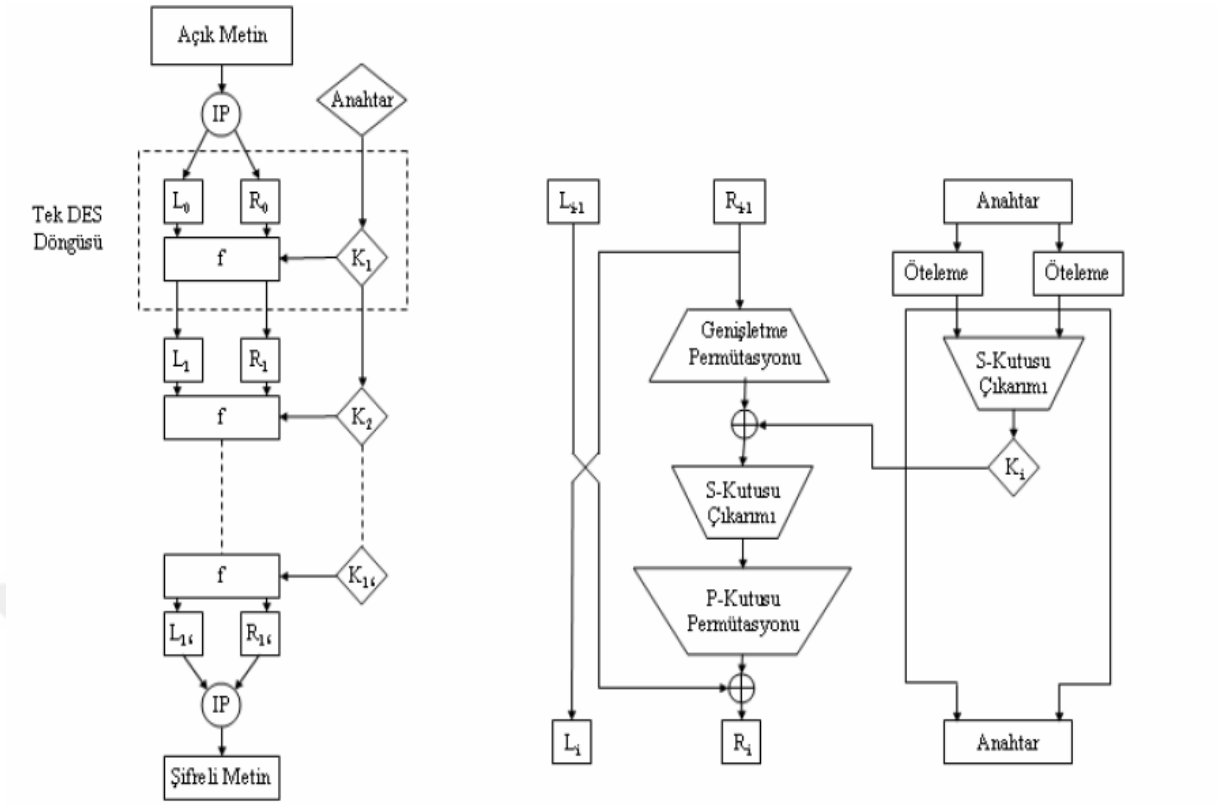
Şekil 3 : PC-1 Permütasyonu

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	35
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Şekil 4 : PC-2 Permütasyonu

DES algoritmasının çalışmasını aşağıdaki gibi özetlenebilir[6]:

1. 56 bitlik anahtardan 48 bitlik (K1,K2,K3..K16) olmak üzere 16 adet anahtar üretilir.
2. 64 bitlik açık metni (x) başlangıç permütasyonuna sokulur.
3. Elde edilen 64 bit dizisiyle sol ve sağ parçalara ayrılacak şekilde 32 bitler haline getirilir.
4. Sağ parça ile 48 bitlik anahtar işleme sokarak F fonksiyonunu elde et. Daha sonra F fonksiyonunu sol parça ile XOR'lamaya tabi tut.
5. Sağ parçayı sol parçaya ata.
6. 4 işlemde elde edilen değeri sağ parçaya ata.
7. Elde edilen 64 bit değeri diğer döngünün başına yerleştir.
8. 4-7 arasındaki işlemleri 16 kez tekrarla
9. 16 döngü bittiğinde y şifreli metni elde et.



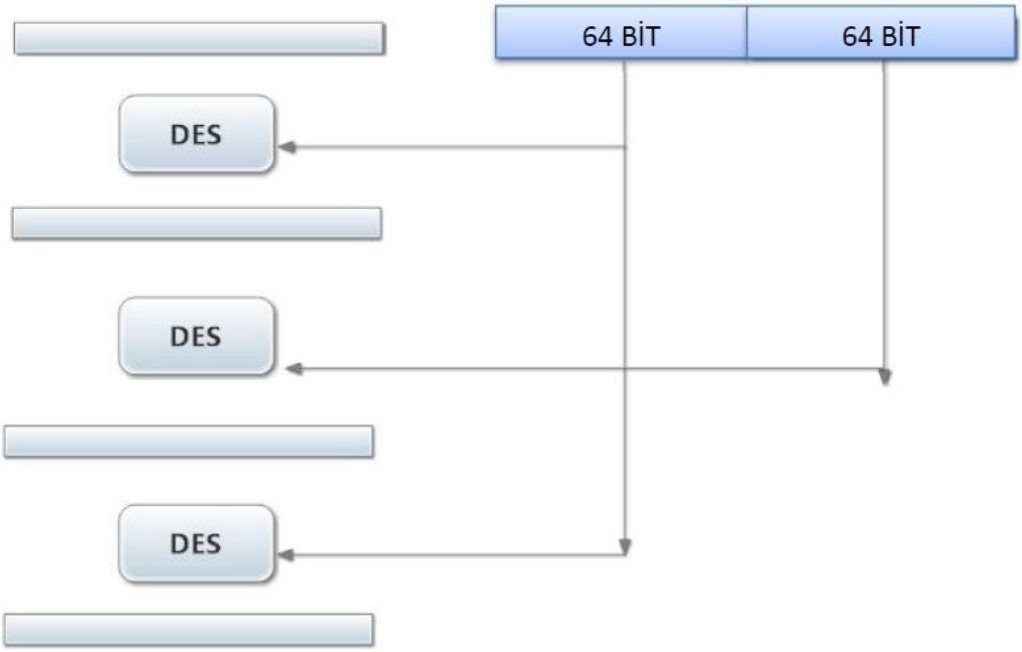
Şekil 5 : DES algortimasının çalışma şekli[6]

2.2. Triple DES Algoritması

DES algoritmasının brute force yöntemine karışı koymakta zorlandığından dolayı IBM tarafından 1978 yılında geliştirilen şifreleme algoritmasıdır. DES algoritmasının en büyük dez avantajının 56 bit olmasıdır.[2][7]

Özellikleri[2][7]:

1. Çift yönlü çalışır. Şifreleme ve çözme işlemini yapabilir.
2. DES şifrelenmesinin 3 kez tekrar edilmesi şeklinde çalışır.
3. DES algoritmasına göre 3 kez daha yavaştır.
4. Şifreleme yapmak için 24 byte uzunluğunda anahtar kullanılır. Her byte için 1 bit eşlik için kullanıldığından 168 bit anahtar uzunluğu vardır. Şifreleme anahtar seçenekleri arasında 8 byte,16 byte,24 byte vardır.
5. Veri, Triple DES anahtarının 8 byte ile şifrelemeye tabi tutulur. Daha sonra ortadaki 8 byte ile çözülür. Son 8 byte ile de tekrar şifrelenir.



Şekil 6 : Triple DES algoritmasının akış Diyagramı

2.3. AES Şifreleme Algoritması

DES algoritmasının gelişen teknoloji ve işlemci hızları karşısında güvenirliliğini yitirmeye başladığı için ocak 1997'de NIST (National Institute of Standarts and Technology) yeni bir şifreleme standartının geliştirilmesi için çalışmalara başladı.[1][2]

NIST tarafından Eylül 1997'de yeni bir şifreleme standartı için yarışma düzenlendi. Ağustos 1998'de 15 aday algoritması (1.Tur) değerlendirilmeye alındığını duyurdu. Nisan 1999 da 2. Tur değerlendirilmesinin sonucu 5 aday algoritması kaldı.[1]

2000 yılının Ekim ayında NIST Gelişmiş şifreleme standartı'nın (AES) geliştirilmesi konulu bir rapor yayınlamış ve Rijndael algoritmasına AES şifreleme adı verilerek tavsiye edilmiştir.[1]

AES, Rijndael algoritmasının bir bölümünü gerçekleştirebilmektedir. Rijndael algoritmasında anahtar ve blok 128 bitten 256 bite kadar 32 bit aralıklarla birbirinden

bağımsız olarak değişebildiği halde AES 128 bit ve 192 bit 256 bit olarak anahtarda blokta ise 128 bit olarak kullanılmaktadır.[1]

AES SPN(Substitution Permutation Network) algoritmasının geniş bir çeşidir. 128 bit veri bloklarını 128 bit,192 bit,256 bit anahtar seçenekleri ile şifreleyen bir blok şifreleme algoritmasıdır.[1][2][6][7]

128 bit'lik veri blokları her biri 32 bit'ten oluşan 4 kelime olarak düşünülmektedir. AES ile şifreleme işlemi başlarken 4 kelimeden oluşan veri bloğu durum dizisi içerisine yazılır. Giriş bloğu, çıkış bloğu, durum uzunluğu 'N_b' ile ifade edilir. N_b değeri sabittir ve bu değer 4 eşittir.[1][2]

Algoritmanın anahtar uzunluğu N_k ile gösterilir ve N_k anahtar kelime sayısını belirtirken 4, 6,8 değerlerinden birini gösterir.[1]

N_r AES anahtar uzunluğuna bağlı olarak Tur sayısının değişimini ifade etmektedir.[1][2] N_r,N_k,N_b'nin kısaca anlatımı Şekil 7'de ifade edilmiştir.

AES Anahtar uzunluğu	Anahtar uzunluğuna karşı gelen kelime sayısı (N_k)	Blok uzunluğuna karşı gelen kelime sayısı (N_b)	Tur sayısı (N_r)
128 bit	4	4	10
192 bit	6	4	12
256 bit	8	4	14

Şekil 7 : Tur sayısının anahtar uzunluğuna göre değişimi ve Anahtar uzunluğunun kelime sayısının değişimi

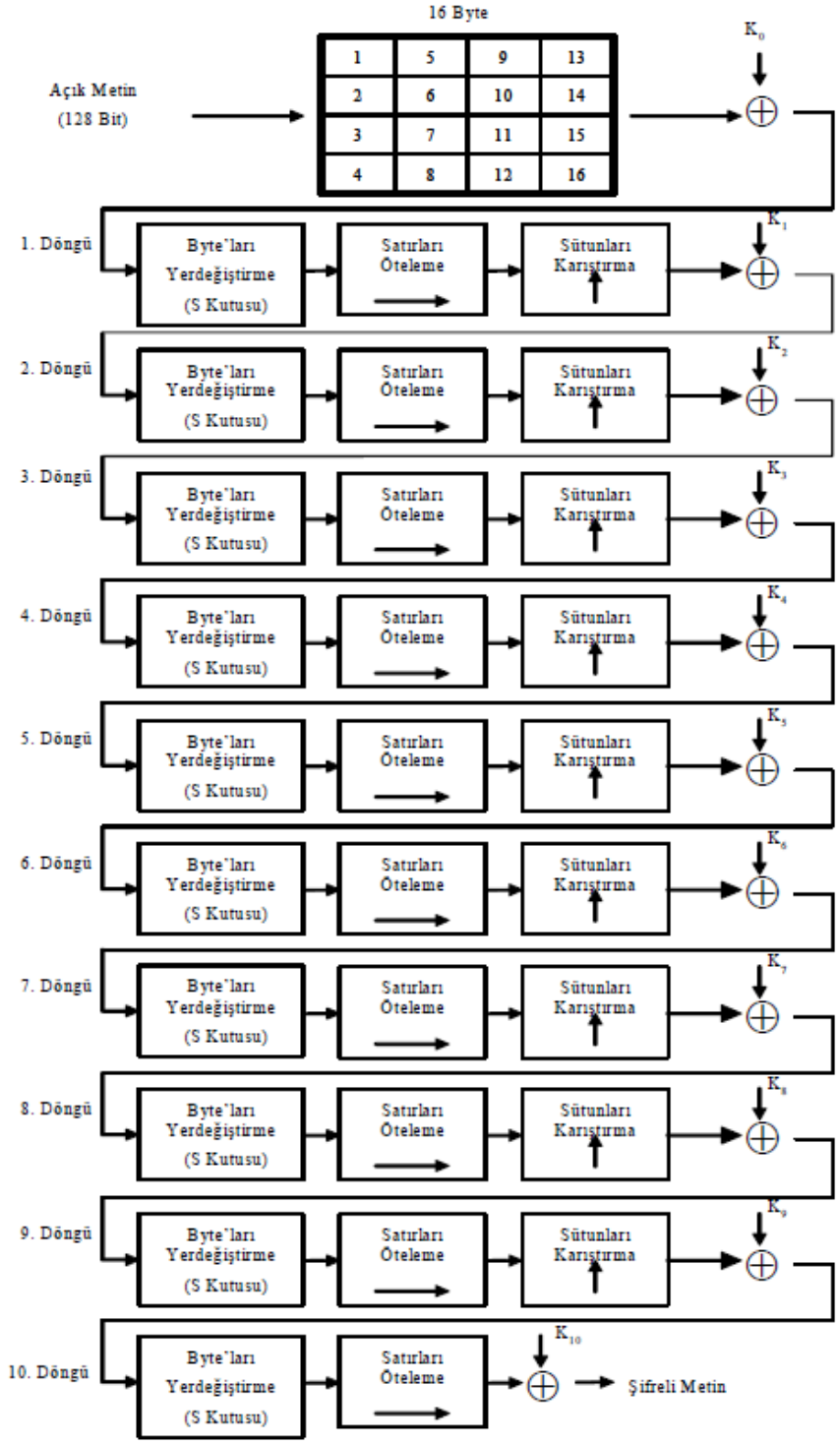
"AES Algoritması iki bloktan oluşur. İlk blok tur dönüşümü ikinci blok ise anahtar üretim bloğudur."[2]

AES algoritmasında her döngü 4 katmandan oluşmaktadır. Bunlar sırasıyla [1][2][6][7].

1. SubBytes dönüşümü (S kutusu kullanılarak) olarak bilinen byte'ların yer değiştirmesi
2. ShiftRows dönüşümü olarak bilenen satırların ötelenmesi
3. MixColumns dönüşümü olarak bilenen Sütunları Karıştırılması
4. AddRoundKey dönüşümü olarak bilinen anahtarla XOR'lama işlemi

AES şifreleme algoritması tekrarlanan bir döngü yapısına sahip olduğu için yukarıda 4 işlem her döngüde tekrar eder. Yalnızca son döngüde MixColumns dönüşümü olarak bilenen Sütunlarun Karıştırılması işlemi yapılmaz.[1][2][6][7]

128 bit veri 4x4 byte matrisine dönüştürülür. Her satırda 4 sütündan oluşur ve her satır 32 bit değerinde veri tutar. AES şifreleme algoritması Byte'lar üzerinden işlem yapmaktadır. 4x4 matriks Master anahtar ile XOR'lama işlemine tabi tutulduktan sonra döngü işlemi başlatılır. Bu döngü işlemi yukarıda belirtmiş olduğum işlemdir.[1][2][6][7]



Şekil 8 : AES Şifreleme algoritmasının çalışma şekli (128 bit anahtar için)

2.3.1. SubBytes Dönüşümü (Byte'ların Yer Değiştirmesi)

Byte'ların yer değiştirilmesi işlemi lineer olamayan bir işlemdir. Bir S kutusu kullanılarak her byte üzerinde işlem yapılır. Her byte için karşılık gelen byte birbirinden farklı baytlardır. Bu dönüşüm işlemi iki aşamada gerçekleşir[1][2][6]:

1. Çarpmaya göre ters alma işlemidir. GF(2^8)'de bir polinom olarak ele alınır. Polinom "$x^8 + x^4 + x^3 + x^2 + 1$" kullanılarak ters alma işlemi gerçekleşir.
2. Birinci dönüşüm sonucu elde edilen değer,Affine dönüşüm GF(2) üzerinde uygulanır. Affin dönüşümün girişinde x_i biti dönüşüm sonrası elde edilen y_i biti olacak şekilde aşağıdaki matematik işlemi ile ifade edilir.

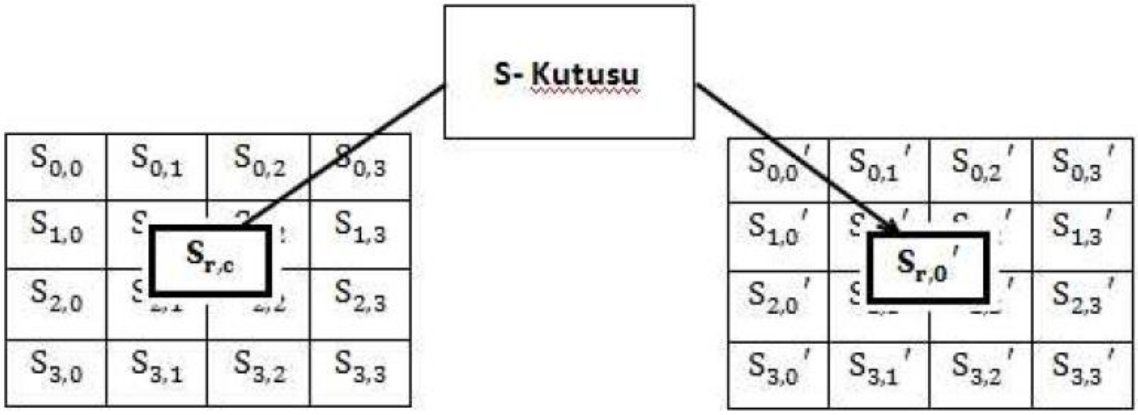
$$y_i = x_i \oplus x_{(i+4) \bmod 8} \oplus x_{(i+5) \bmod 8} \oplus x_{(i+6) \bmod 8} \oplus x_{(i+7) \bmod 8} \oplus c_i$$

Burada c_i, {63} değerinin i'inci bitidir[1]. Affine dönüşümü Şekil-9'daki gibi ifade edilebilir.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Şekil 9 : Affine dönüşüm işlemi

Şekil 10 ve Şekil 11 da S kutusunun Byte değiştirme gösteren resim ile xy byte değerlerinin değiştirilmesine karşılık gelen değerlerin olduğu tablo vardır.



Şekil 10 : S-Kutusu Byte değiştirme işlemi[2]

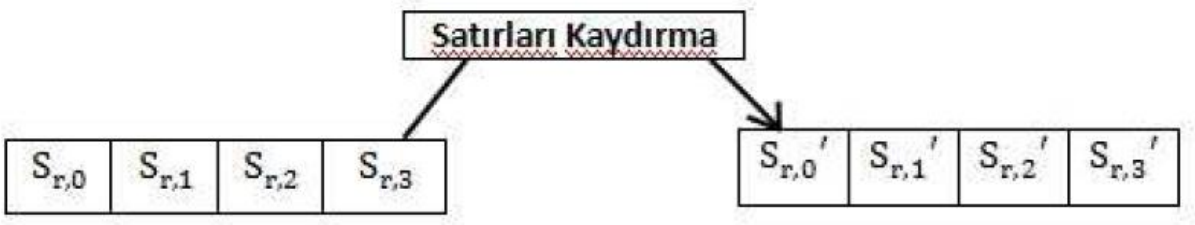
		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Şekil 11 : xy byte değerlerinin değiştirilmesine karşılık gelen değerler tablosu[6]

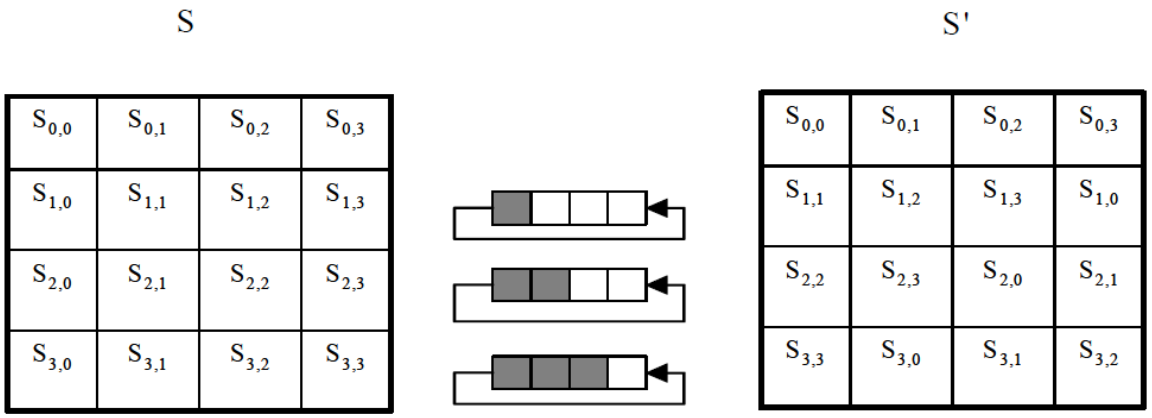
2.3.2. ShiftRows Dönüşümü (Satırların Ötelenmesi)

128 bit giriş verisi 4x4 byte matris olacak şekilde oluşturulmuştu. Satırları öteleme işlemi, birinci satır hariç diğer satırların sağdan sola doğru hareket işlemidir.[1][2][6]

Şifreleme için ikinci satır sağdan sola doğru bir pozisyon değiştirecek şekilde dairesel olarak ötelenir. Üçüncü satır sağdan sola doğru iki pozisyon değiştirecek şekilde dairesel olarak ötelenir. Dördüncü satır sağdan sola doğru üç pozisyon değiştirecek şekilde dairesel olarak ötelenir. Aşağıdaki Şekil 12, Şekil 13 ve Şekil 14 de bu işlemin gösterimi vardır.[1][2][6]



Şekil 12 : Satırları Kaydırmanın ifade edilmesi



Şekil 13 : Satırları Öteleme İşlemi

F0	87	5E	42
1C	3A	7B	EC
97	A6	C3	6E
E1	B2	12	16

F0	87	5E	42
3A	7B	EC	1C
C3	6E	97	A6
16	E1	B2	12

Şekil 14 : Byte'ların Satır ötelemesinin gösterimi

2.3.3. MixColumns Dönüşümü (Sütunların Yer Değiştirilmesi)

MixColumns Dönüşümü, 4x4'lük durum matrisi üzerindeki sütünlarda tek tek bağımsız olarak işlem yapılır. 4 Byte Doğrusal bir dönüşümdür. Giriş matrisinin her bir sütununa MixColumns dönüşümü uygulanarak çıkış durum matrisi elde edilir. MixColumns dönüşümü, algoritmaya nüfuz etme (diffusion) özelliği kazandırır.[1][2][6]

Sütunlar katsayıları GF(2^8)'in birer elemanı olan üçüncü derecede bir polinom olarak kabul edilirler. Bu polinomlar sabit bir polinomla(a(x)), x^4+1'e göre mod alınarak, çarpılarak MixColumns dönüşümü gerçekleştirilir. "Bu sabit polinomun katsayıları mümkün olan en basit işlem karmaşıklığını sağlayacak şekilde seçilmiştir." [1] Sabit a(x) polinumu ve sütunların MixColumns dönüşümü Şekil 15 ve Şekil 16 de verilmiştir. . [1][2][6]

$a(x) = \{0,3\}x^3 + \{0,1\}x^2 + \{0,1\}x + \{0,2\}$

Şekil 15 : Sabit a(x) polinumu

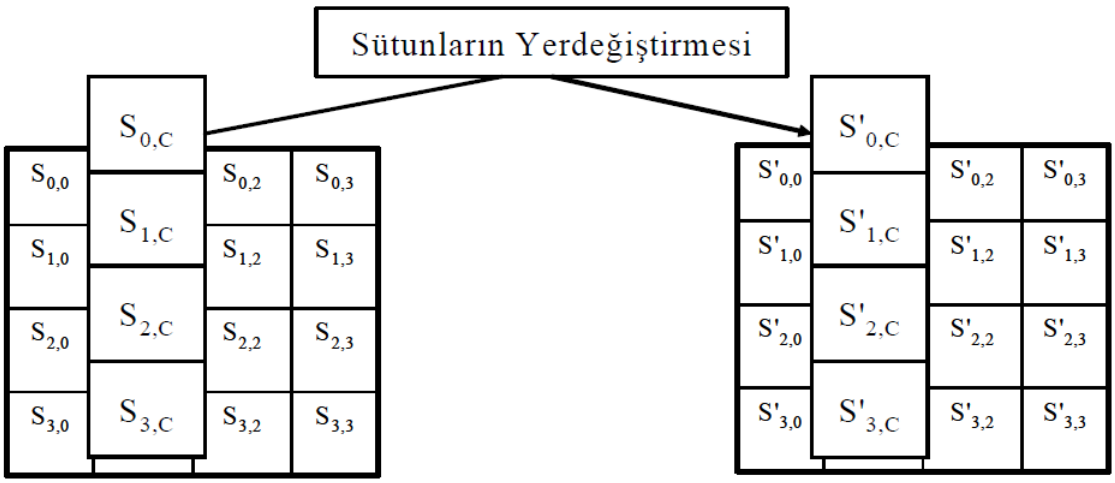
$s(x) = a(x) * s(x)$

Şekil 16 : Sütunların karıştırma işlemi

Sütunların MixColumns dönüşümü durum matrisi üzerinde gösterimi Şekil 17'de ve Şekil 18 da görünmektedir. [1][2][6]

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

Şekil 17 : Sütunların Durum matrisi üzerinde MixColumns dönüşümü

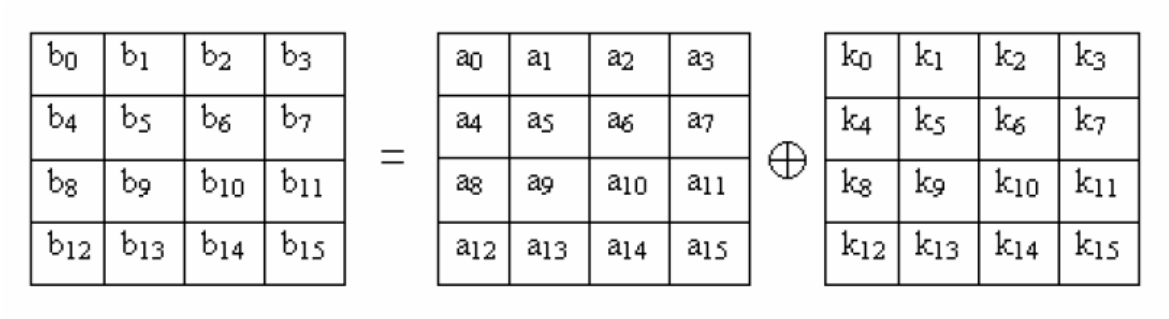


Şekil 18 : Sütunların MixColumns Dönüşümü (Sütun Sütun değişikliğin gösterilmesi)

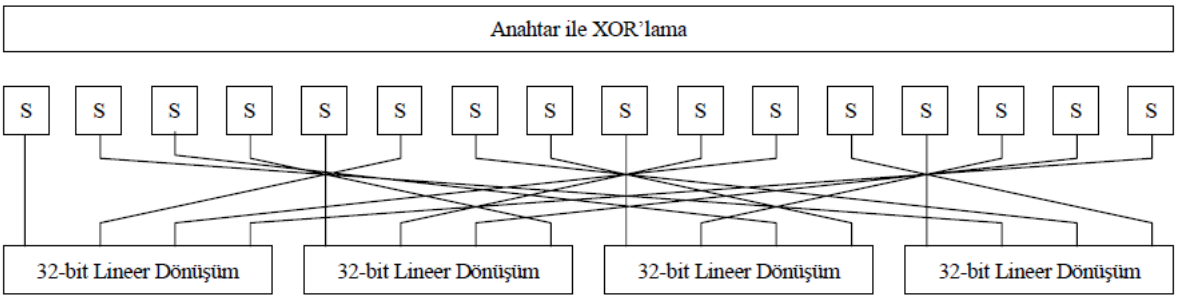
2.3.4. AddRound Key Dönüşümü (Döngüye Anahtar Ekleme Dönüşümü)

Master (ana) anahtardan o döngü için üretilen anahtar ile 4x4 matrisi olan durum matrisi arasında XOR'lama işlemi yapılır. Bu işlem GF(2) üzerinden doğrusal bir dönüşümdür. Bu katmandaki yapı şifreleme ve şifre çözme durumunun her

ikisinde de kendisine eşittir. Durum matrisi üzerinde yapılan XOR'lama işlemi Şekil 19'da ve Şekil 20'de anlatılmıştır. [1][2][6]



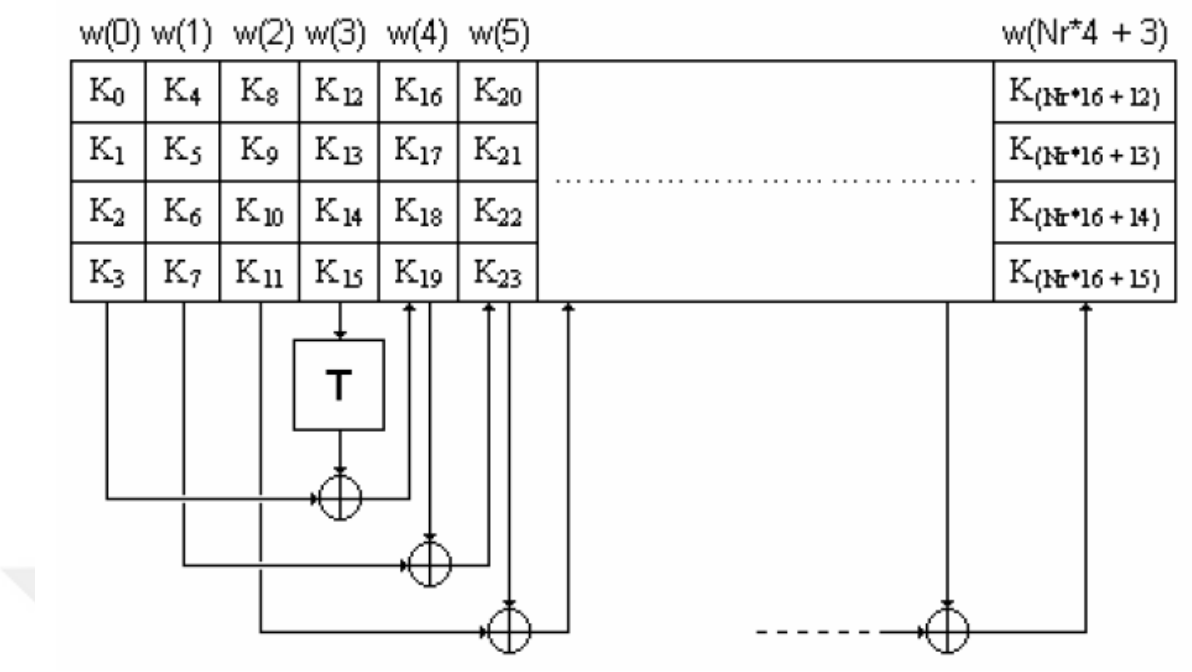
Şekil 19 : Durum matrisi üzerinde XOR'lama işleminin gösterimi



Şekil 20 : Anahtar ile XOR'lama işlemi

2.3.5. Anahtar Üretim İşlemleri

AES algoritması 128 bit,192 bit,256 bit anahtar uzunlukları çeşidinden birini kullanır. Algoritma her bir döngü için gerekli olan K anahtar bloklarını üretir. Çeşitli anahtar uzunluklarına sahip olmasına rağmen veri blokları 128 bit olduğu için oluşturacağı K anahtar bloklarının uzunluğu 128 bit olmaktadır. Her bir döngüde de 128 bitlik anahtar kullanılır. Anahtar üretici, her döngü için uzunlukları farklı olan anahtar bloklarından $4*N_k$ boyutunda, Elemanları byte'lardan oluşan birer matris olarak kaydeder. Anahtar üretici bu matrisin üzerinde çalışırken tüm anahtar matrisleri (Şifreleme anahtarı ile döngü anahtarları) birleştirerek, $4*(N_r + 1)$ boyuntunda bir matris elde eder. 128 bitlik giriş anahtarı için anahtar üreticinin şekli, Şekil 21'de gösterilmektedir. [1][2]



Şekil 21 : Nr=10, Nk=4, 128 bitlik Giriş anahtarı için Anahtar üretici

Matrisin ilk N_k Sütunu, Master (ana) anahtarın byte'ları ile doldurulur. Sonraki Anahtar sütunları şu yöntemler izlenerek oluşturulur[1].

1. Yeni oluşturulacak olan sütunun numarası, N_k'nın katıysa; oluşturulacak olan sütundan bir önceki sütun (T) dönüştürme işleminden geçirildikten sonra yeni oluşturulacak olan sütundan N_k önceki sütun ile XOR'lama işlemine tabi tutulur.
2. N_k'nın 6'dan büyük olması (256 bit anahtar blok uzunluğuna sahip olması) durumunda oluşturulacak yeni sütun numarası N_k'ya göre mod alındığında 4 sonucunu veriyorsa kendisinden bir önceki sütun S-Kutusundan geçirildikten sonra, kendisinden N_k önceki Sütunla XOR'lama işlemine tabi tutularak oluşturulur.
3. Yukarıdaki iki durum oluşmaması durumda yeni oluşacak Sütun, kendisinden bir önceki sütun ile N_k önceki Sütunun XOR'lama işlemi sonucu oluşur.

T Dönüşüm işlemi üç aşamadan meydana gelir[1]:

1. Dört byte'lık olan Sütunun dairesel olarak ötelenir.
2. Öteleme sonucu oluşan dört byte'lık sütun her byte'ı S-Kutusuna sokulur.

3. Elde edilen yeni Sütun bir döngü sabit vektörüyle XOR'lama işlemine tabi tutulur. Döngü sabit vektörünün ilk Byte'ı her döngü için farklı bir değer alır diğer byte'ları 0 değerine sahiptir.

2.3.6. AES Algoritmasında Deşifreleme İşlemi

Deşifreleme işlemi sırasında şifreleme işleminde yapılan bütün işler ters sırası ile yapılmalı ve anahtar planlama ters sırası ile oluşturularak kullanılmalıdır. ShiftRows , Subbytes, MixColumns dönüşümleri tersinden yapılmalıdır.[6]

2.3.6.1. Ters MixColumns (Sütunları Karıştırma) Dönüşümü

Ters MixColumns dönüşümü MixColumns dönüşümünün tersidir. Sütunlar GF(2^8)'de polinomlar olarak düşünülür. Sabit bir $a^{-1}(x)$ polinomu ile modulo x4 + 1'e göre çarpılır. Şekil 22'de sabit a polinomu gösterilmektedir. Şekil 23'de sütunların karıştırma işlemi gösterilmektedir. Şekil 24'de Sütunların matris üzerinde ters MixColumns dönüşümü gösterilmektedir.[1][6]

$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$

Şekil 22 : Sabit a(x) polinomu

S` = a-1(x) ⊕ S(x)

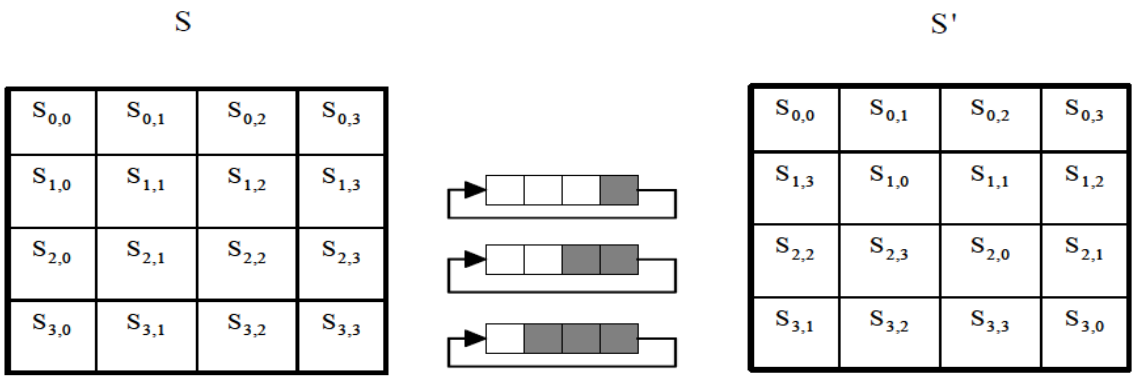
Şekil 23 : Sütunların karıştırma işlemi

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

Şekil 24 : Sütunların matris üzerinde ters MixColumns dönüşümü

2.3.6.2. Ters ShiftRows (Satırları Öteleme) Dönüşümü

Ters ShiftRows Dönüşümü, durum verisinin birinci satır hariç soldan sağa doğru hareket etmesidir. İkinci satır soldan sağa doğru bir kez, üçüncü satır soldan sağa doğru iki kez ve son satır soldan sağa doğru üç kez dairesel olarak ötelenmelidir. Şekil 25 ve Şekil 26'de Ters ShiftRows dönüşümünü gösterilmektedir.[1][6]



Şekil 25 : Ters ShiftRows (Satır öteleme) Dönüşümü

F0	87	5E	42
3A	7B	EC	1C
C3	6E	97	A6
16	E1	B2	12

F0	87	5E	42
1C	3A	7B	EC
97	A6	C3	6E
E1	B2	12	16

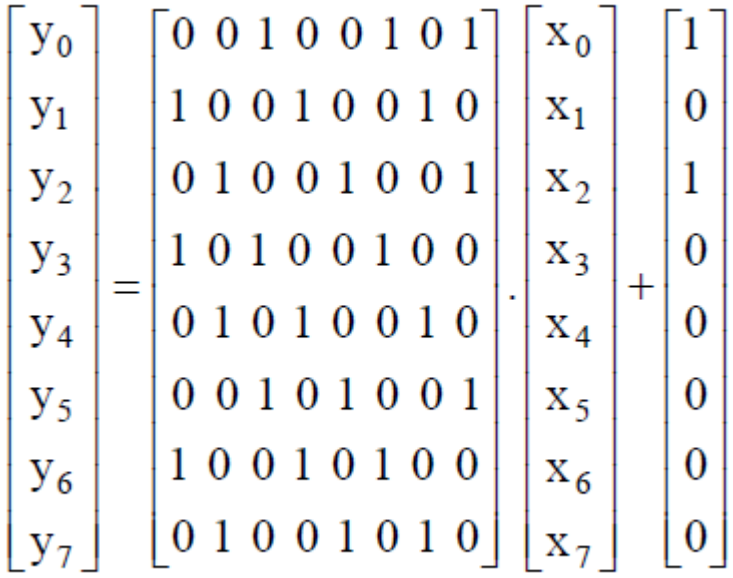
Şekil 26 : Byte'ların Ters Satır ötelemesinin gösterimi

2.3.6.3. Ters SubBytes (Byte'ların Yer Değiştirme) Dönüşümü

Ters Subbytes dönüşümü yer değiştirme dönüşümünün tersidir. Ters S kutusu kullanılarak her byte'a karşılık gelen byte değeri ile yer değiştirilir. Ters S kutusu Affine dönüşümünün tersi uygulanarak elde edilen durumun GF(2^8) de çarpmaya göre ters alma işlemi uygulanarak elde edilir.Ters S kutusu Şekil 27'de gösterilmektedir. Affine dönüşümün tersi Şekil 28'de gösterilmektedir. [1][6]

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Şekil 27 : Ters S kutusu



Şekil 28 : Affine Dönüşümün Tersi

2.4. RC2 Algoritması

1987 yılında Ron Rivest tarafından RSA'nın güvenliği için RC2 algoritması tasarlanmıştır. RC2 gizli anahtarla şifreleme yapan klasik blok şifreleme algoritmalarından biridir. Giriş ve çıkış blokları her biri 64 bittir. Mevcut uygulamada anahtar 8 byte olarak kullanılmaktadır. Anahtara ek olarak 40 ile 88 bit arası uzunluklarda değişen "salt" adı verilen ayrı ek bir anahtar daha kullanılabilmektedir. IBM AT'ye göre uygun anahtar atandığında DES şifreleme algoritmasına göre iki kat daha hızlı çalışmaktadır.[2]

BÖLÜM 2

GELİŞTİRİLEN YAZILIM SİSTEMİ, KULLANIM SENARYOLARI VE ÖRNEK UYGULAMALAR

1. Geliştirilen Yazılım Sistemi

Visual Studio'da yazılan program c# dilinde olup emgu-cv framework'unu kullanarak Simetrik şifreleme algoritmalarından yararlanarak hazırlanmıştır. Videodan gelen ya da Kameradan gelen her bir görüntü simetrik şifreleme algoritmasına bağlı olarak 64 byte'lara ya da 128 byte'lara bölünmüştür. Kullanılan Padding Mode PKCS7 olması durumunda 64 byte'dan 8 byte, 128 byte'dan 16 byte eksiltilmiştir. Bu eksiltme işlemi gelen byte miktarı 8 byte'lara bölünüp içinden beşinci byte silinerek beşinci byte'ın yerine altıncı byte, altıncı byte'ın yerine yedinci byte ve yedinci byte'ın yerine sekinci byte koyularak yeniden oluşturulmuştur. Seçilmiş olan simetrik şifreleme algoritmalardan birine bu oluşturulan yeni hali verilmiş çıkan şifrelenmiş veriler birleştirilerek şifrelenmiş frame'ler oluşturulmuştur. Bu frame'ler avi formatında bir video olarak oluşturulmuştur. Şifrelenmiş frame'leri bulunduran avi formatındaki video şifreleri girilerek orjinal haline getirilirken tekrar Padding Mode PKCS7 olarak seçilmelidir. Gelen şifrelenmiş frame'ler simetrik şifreleme algoritmasına bağlı olarak 64 byte ya da 128 byte'lara bölünmüştür. Bölünmüş olan frame'in her bir parçası simetrik şifreleme algoritmasına sokularak 64 byte girdiyse 56 byte, 128 byte girdiyse 112 byte çıkacaktır. Bu çıkan byte miktarı 7 byte'lara bölünmüştür. 7 byte'lık 8 byte'lık veriye dönüşüm işlemi yapılırken ilk dört byte sabit kalmış beşinci byte boş bırakılmıştır. Altıncı byte 7 byte'lık olan verinin beşincisi verilmiştir. Yedinci byte 7 byte'lık olan verinin altıncısı verilmiştir. Sekizinci byte 7 byte'lık olan verinin yedincisi verilmiştir. Padding Mode None olan şifrelemelerde her bir frame 64 byte ya da 128 byte'lara bölünmüştür. Bu 64 byte ya da 128 byte bölünen frame'ler olduğu gibi simetrik şifreleme algoritmalarından biri seçilerek şifrelenmiştir. Bu şifrelenen veriler şifrelenmiş frame'leri oluşturarak avi formatında şifrelenmiş videolar oluşturulmuştur. Oluşturulan bu şifrelenmiş avi formatındaki videolar aynı şekilde Padding Mode None olarak seçilerek şifresi girildiğinde orjinal haline getirilmiştir.

2. Yazılımın Kullanım Senaryoları

Yazılımın birkaç senaryosu bulunmaktadır. Bunlar kısa şu şekildedir.

A. Dosya aç butonu ile mp4 formatında video seçilir. Dosyayı kaydet butonu ile de videonun avi formatında nereye kaydedileceği seçilir. DES,TripleDES,AES,RC2 şifreleme algoritmalarından biri kullanarak ECB veya CBC Mode seçilir. Padding Mode'larından biri seçilir. Key ve IV değerleri girilir. Koordinatlar girilmeyerek video şifreleme başlat butonuna basılması durumda hiçbir sayısal değer girmediğinizi söyler ve devam etmek istiyor musunuz diye sorar. Devam butonuna basmanız durumunda videodaki her bir frame'in bütün datasını şifreleme sokar.

B. Dosya aç butonu ile mp4 formatında video seçilir. Dosyayı kaydet butonu ile de videonun avi formatında nereye kaydedileceği seçilir. DES,TripleDES,AES,RC2 şifreleme algoritmalarından biri kullanarak ECB veya CBC Mode seçilir. Padding Mode'larından biri seçilir. Key ve IV değerleri girilir. Koordinatlar girilerek video şifreleme başlat butonuna basılması durumda videodaki her bir frame'in belirlediğiniz koordinatlar arasında kalan alanını şifrelemeye sokar.

C. Kamera şifreleme yapmak için ise Dosyayı kaydet butonu ile videonun avi formatında nereye kaydedileceği seçilir. DES,TripleDES,AES,RC2 şifreleme algoritmalarından biri kullanarak ECB veya CBC Mode seçilir. Padding Mode'larından biri seçilir. Key ve IV değerleri girilir. Koordinatlar girilmeyerek kamera şifrelemeyi başlat butonuna basılması durumda hiçbir sayısal değer girmediğinizi söyler ve devam etmek istiyor musunuz diye sorar. Devam butonuna basılır. Kamera açılarak kameradan gelen görüntülerin şifrelenmesi başlar. Kamera şifreleme durdur ile kamera şifrelenmesi durdurulur. Avi formatında kameradan gelen görüntülerle şifrelenmiş dosyanız olur.

D. Kamera şifreleme yapmak için ise Dosyayı kaydet butonu ile videonun avi formatında nereye kaydedileceği seçilir. DES,TripleDES,AES,RC2 şifreleme algoritmalarından biri kullanarak ECB veya CBC Mode seçilir. Padding

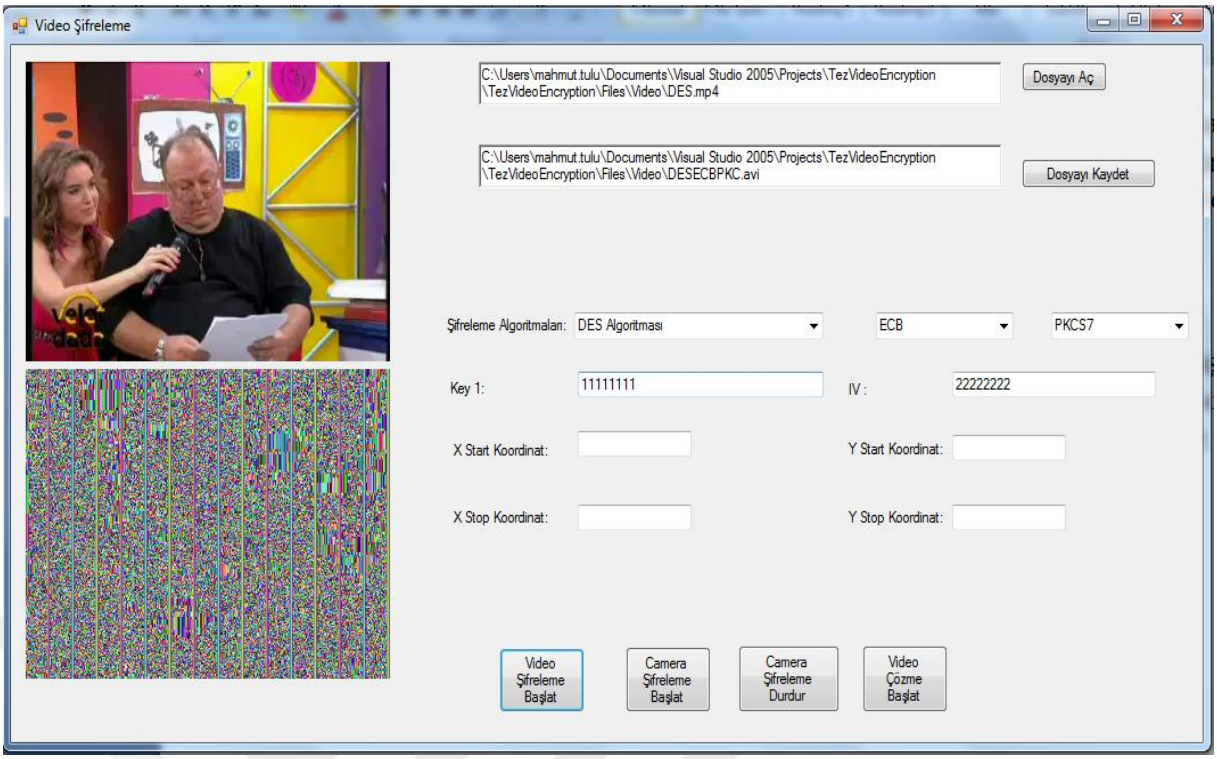
Mode'larından biri seçilir. Key ve IV değerleri girilir. Koordinatlar girilerek kamera şifrelemeyi başlat butonuna basılması durumda kamera açılarak kameradan gelen görüntülerin belirlediğiniz koordinatlar arasında kalan alanın şifrelenmesine başlar. Kamera şifreleme durdur ile kamera şifrelenmesi durdurulur. Avi formatında kameradan gelen görüntülerle şifrelenmiş dosyanız olur.

E. Dosya aç butonu ile şifrelenmiş avi formatındaki video seçilir. Dosyayı kaydet butonu ile de mp4 formatında şifresi çözülmüş videomuzun nereye kaydedileceği seçilir. Seçtiğimiz şifrelenmiş avi formatındaki videomuzun hangi simetrik şifre ile ve hangi mode ve hangi padding mode ile şifrelendiği seçilir. Daha sonra belirli bir koordinat verdiysek koordinatlar girilir. Key ve IV değerleri girilir. Video çözme başlat butonu ile videodaki şifreli alanlar çözülür.

3. Örnek Uygulamalar

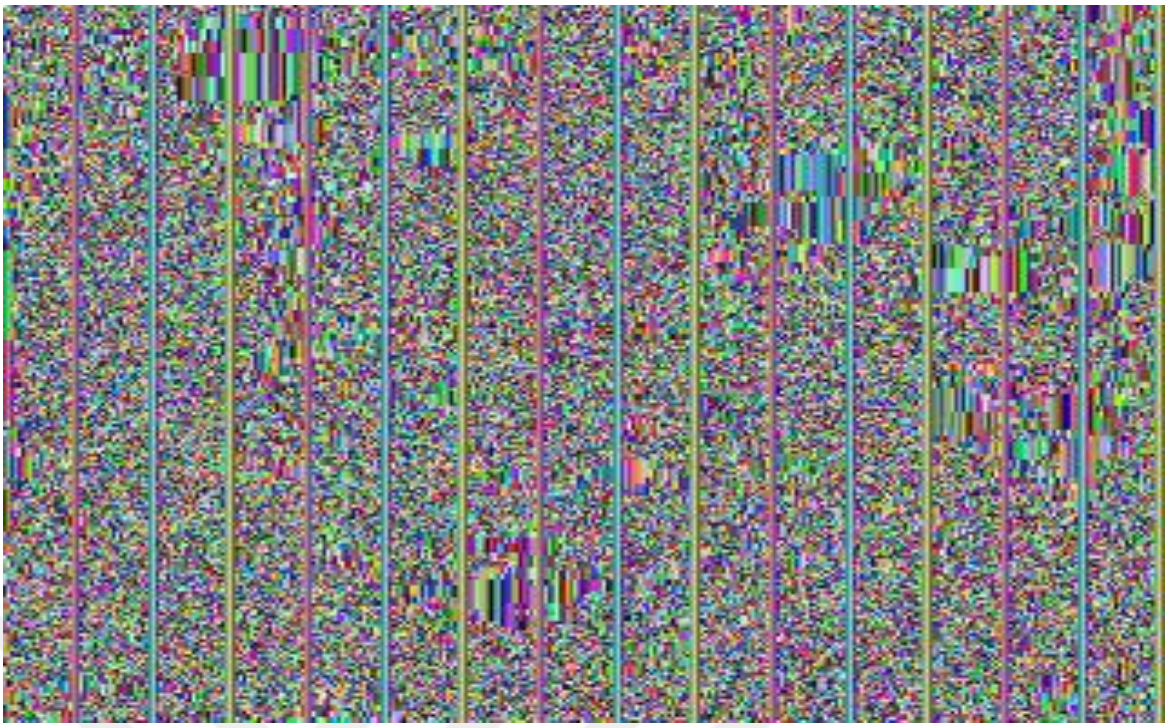
Yukarıda anlatmış olduğum senaryoların sırası ile uygulamasının resimsel gösterimi aşağıda anlatılmaktadır.

A. Birinci senaryonun kullanımı Resim 1 ile Resim 12 arasında kalan resimlerle gösterilmiştir. Koordinatlar girilmemiş hali bulunmaktadır. Resimler Sırası ile ilk Şifreleme algoritmasının olduğu uygulama, ikincisi Orjinal Resim, üçüncüsü ise Şifrelenmiş resim yer almaktadır. Her şifreleme algoritması Mode veya Padding farklı olacak şekilde verilmektedir.

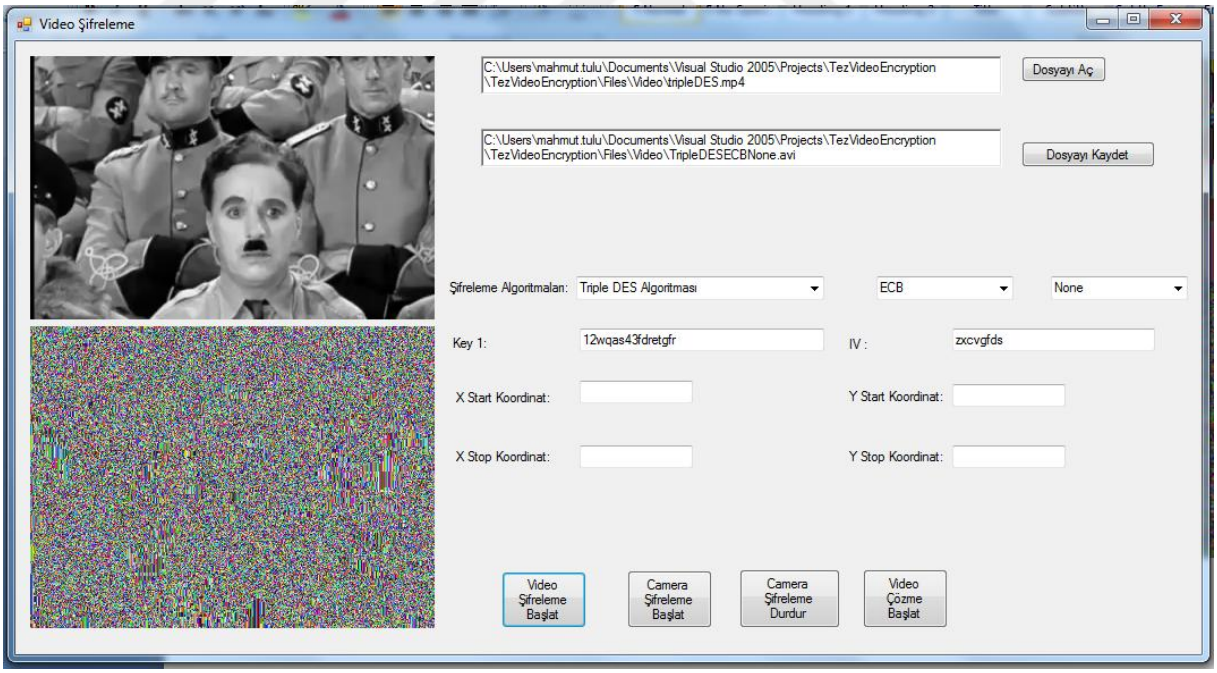


Resim 1 : DES algoritması ECB Mode PKCS7 Padding Mode

Resim 2 : PlainImage

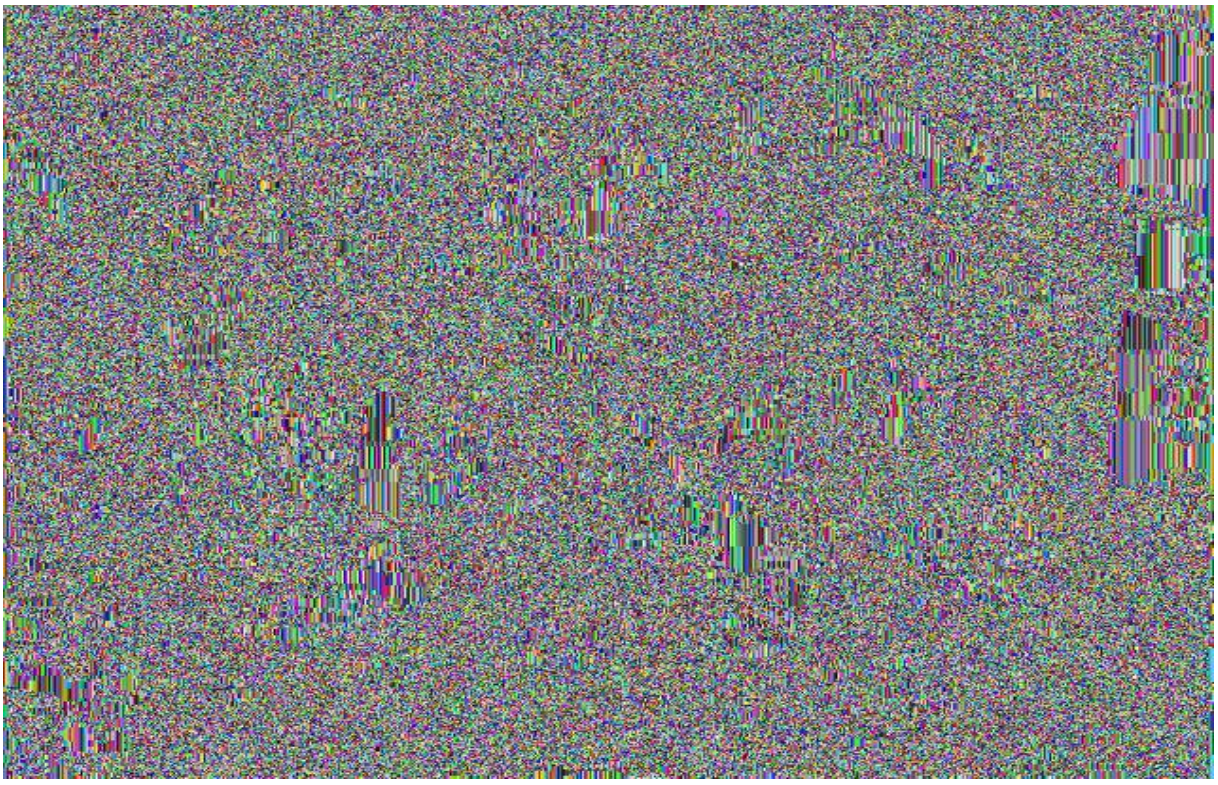


Resim 3 : CipherImage

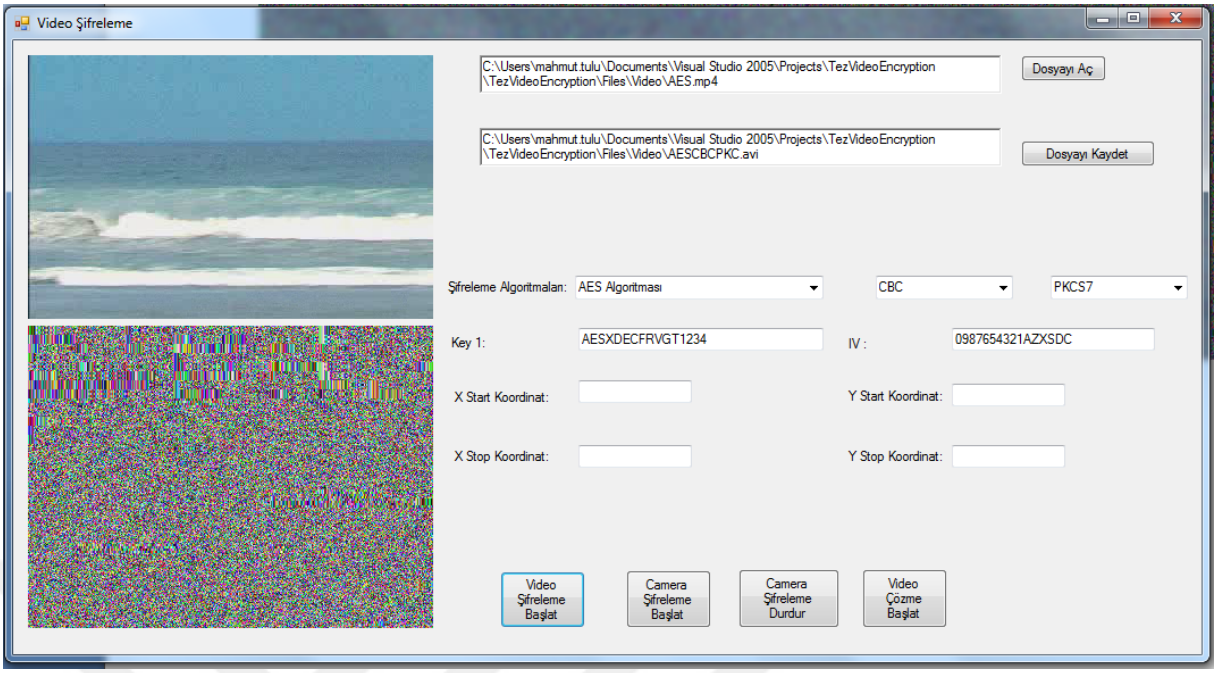


Resim 4: TripleDES algoritması ECB Mode None Padding Mode

Resim 5 : PlainImage

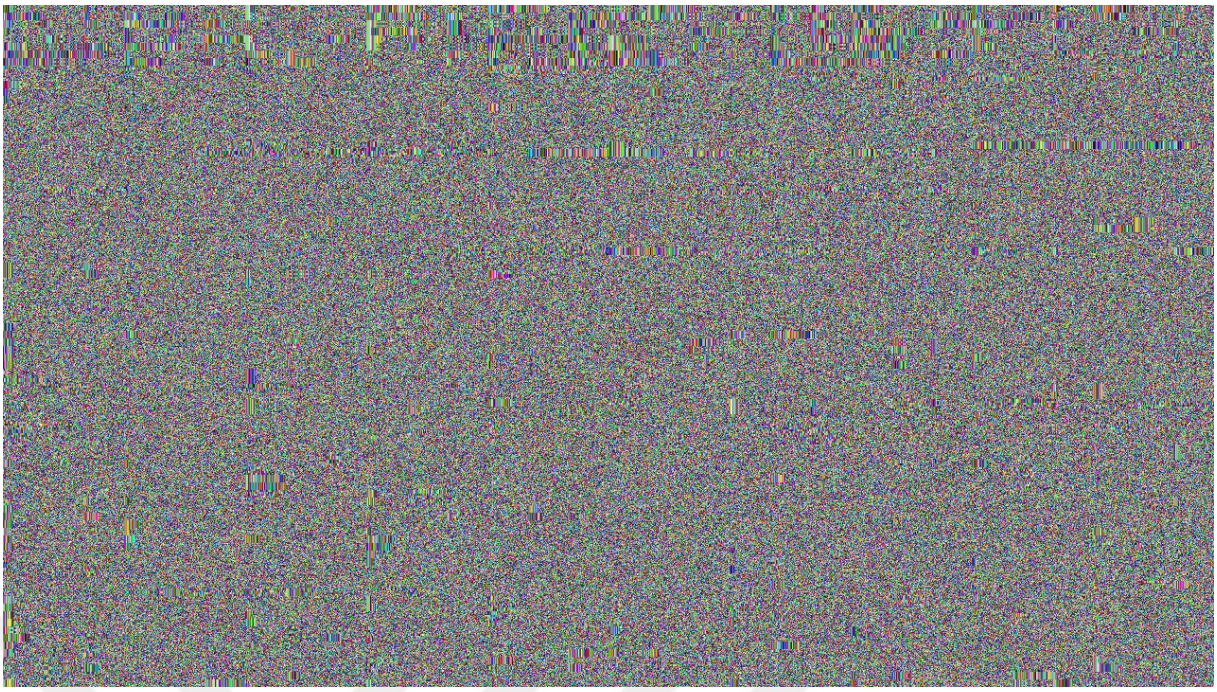


Resim 6 : CipherImage

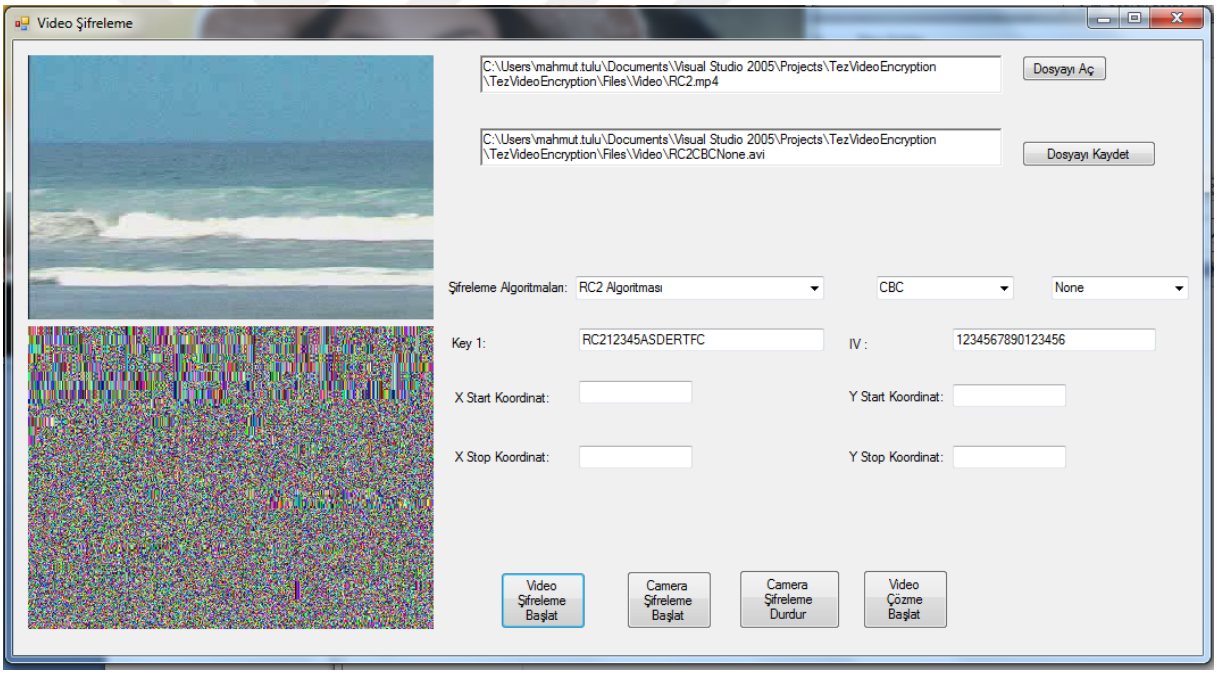


Resim 7 : AES algoritması CBC Mode PKCS7 Padding Mode

Resim 8 : PlainImage

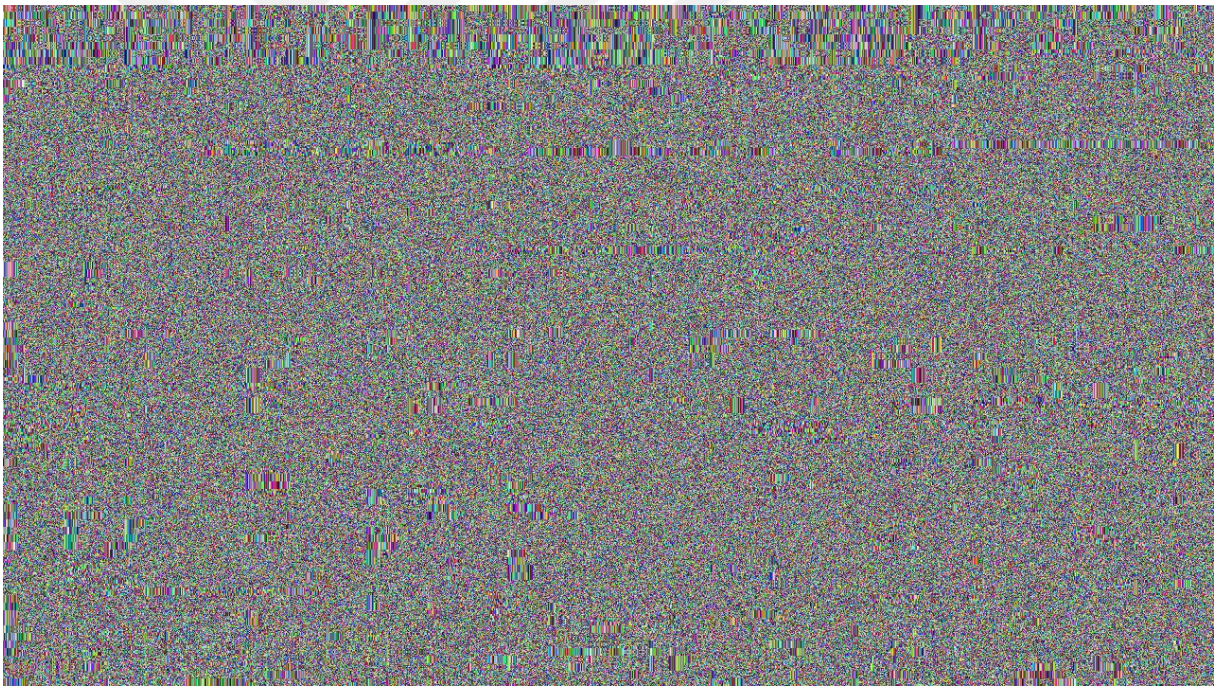


Resim 9 : CipherImage



Resim 10 : RC2 algoritması CBC Mode None Padding Mode

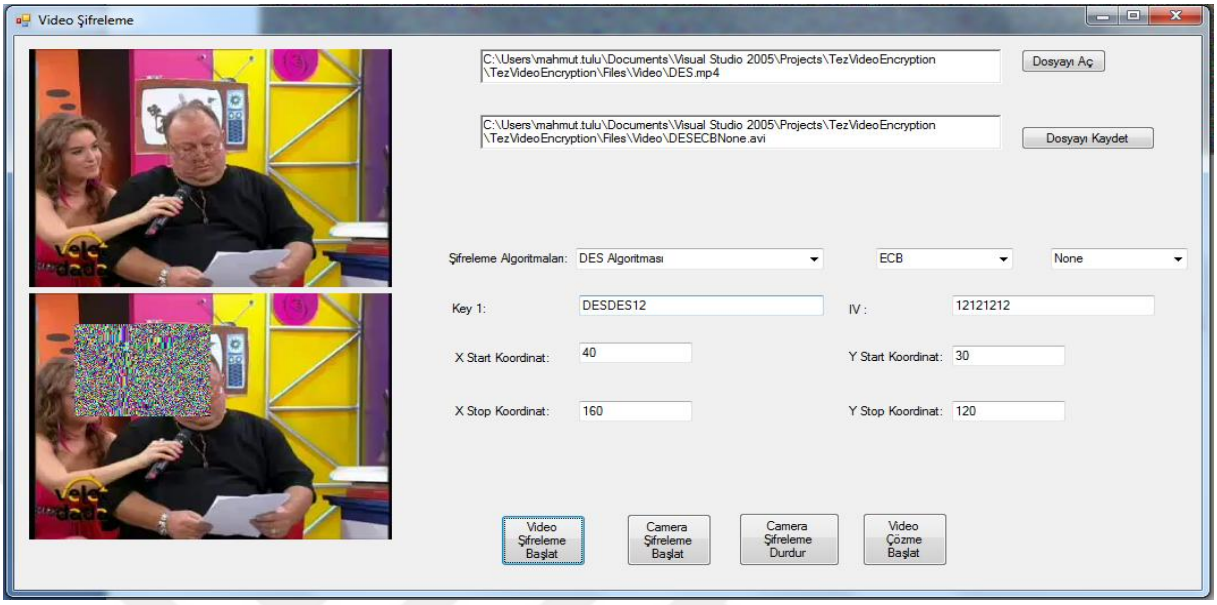
Resim 11 : PlainImage



Resim 12 : CipherImage

B. İkincisi senaryonun kullanımı Resim 13 ile Resim 24 arasındaki resimlerle gösterilmiştir. Bu senaryoda gireceğiz koordinatların arasında kalan alanı şifreleyecek geri kalan alanı şifrelemeden bize gösterecek. Resimler Sırası ile ilk Şifreleme algoritmasının olduğu uygulama, ikincisi Orjinal Resim,

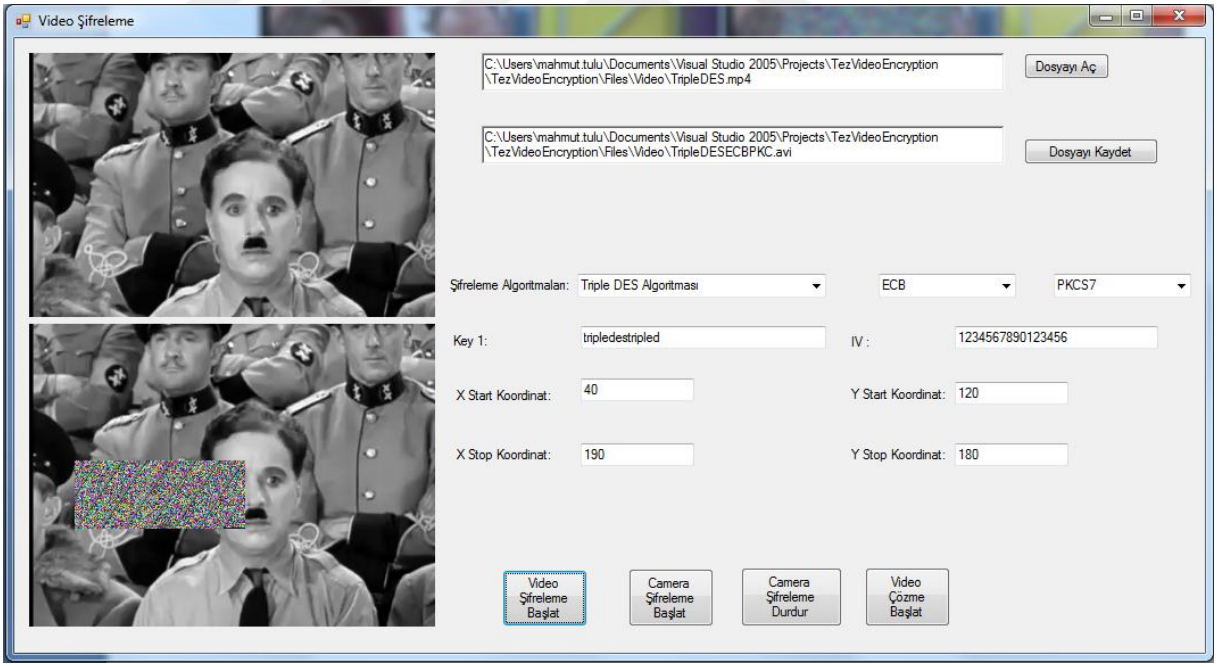
üçüncüsü ise Şifrelenmiş resim yer almaktadır. Her şifreleme algoritması Mode veya Padding farklı olacak şekilde verilmektedir.



Resim 13 : DES Şifreleme Algoritması ECB Mode None Padding

Resim 14 : PlainImage

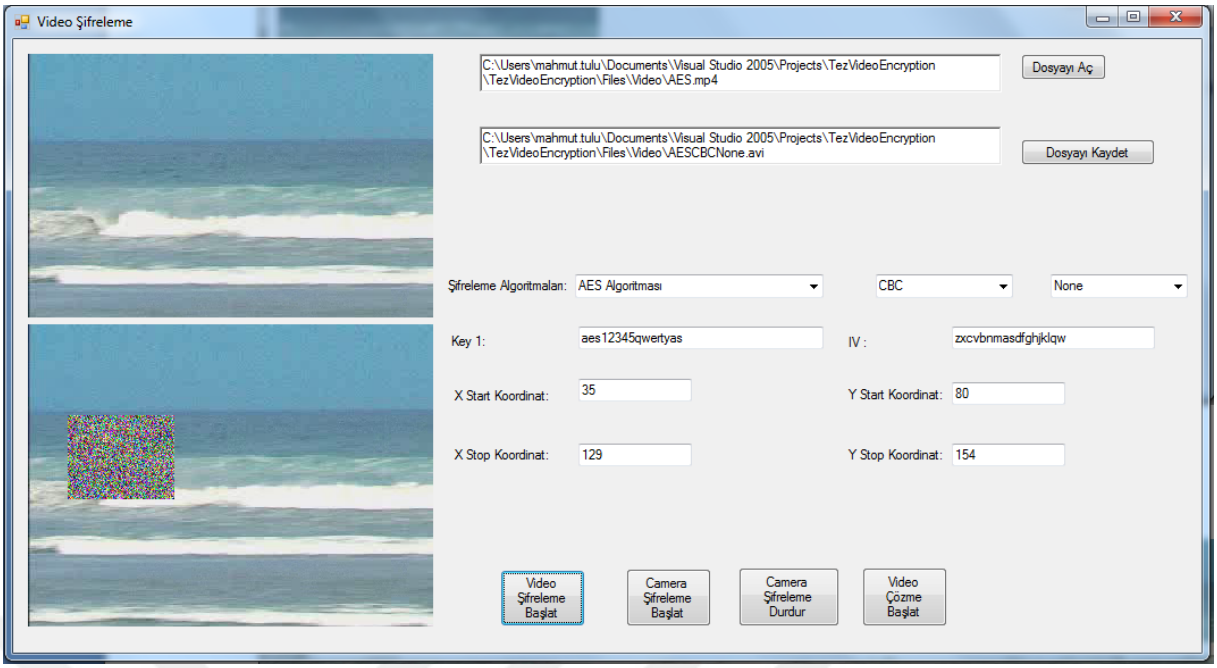
Resim 15 : CipherImage



Resim 16 : TripleDES şifreleme algoritması ECB Mode PKCS7 Padding Mode

Resim 17 : PlainImage

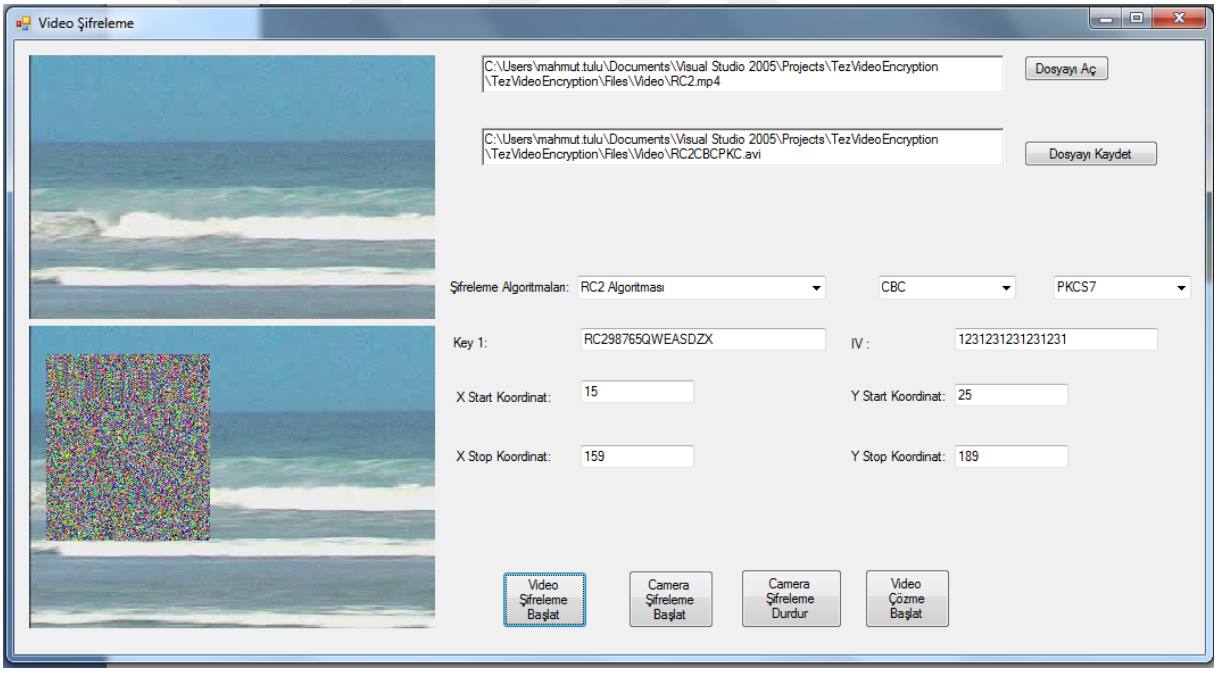
Resim 18 : CipherImage



Resim 19 : AES şifreleme algoritması CBC Mode None Padding Mode

Resim 20 : PlainImage

Resim 21 : CipherImage



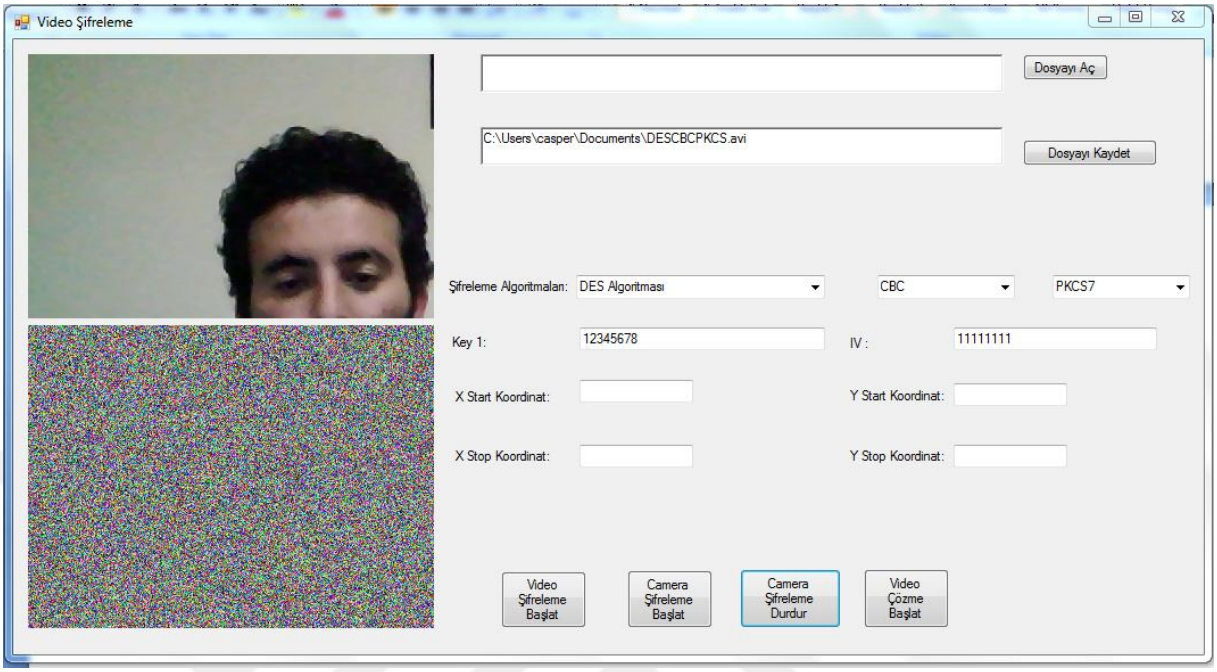
Resim 22 : RC2 şifreleme algoritması CBC Mode PKCS7 Padding Mode

Resim 23 : PlainImage

Resim 24 : CipherImage

C. Üçüncü senaryonun kullanımı Resim 25 ile Resim 36 arasındaki resimlerle gösterilmiştir. Bu senaryoda koordinatlar girilmeyecek. Kamera ile görüntüler alınarak şifrelenecektir. Resimler Sırası ile ilk Şifreleme algoritmasının olduğu uygulama, ikincisi Orjinal Resim, üçüncüsü ise Şifrelenmiş resim yer

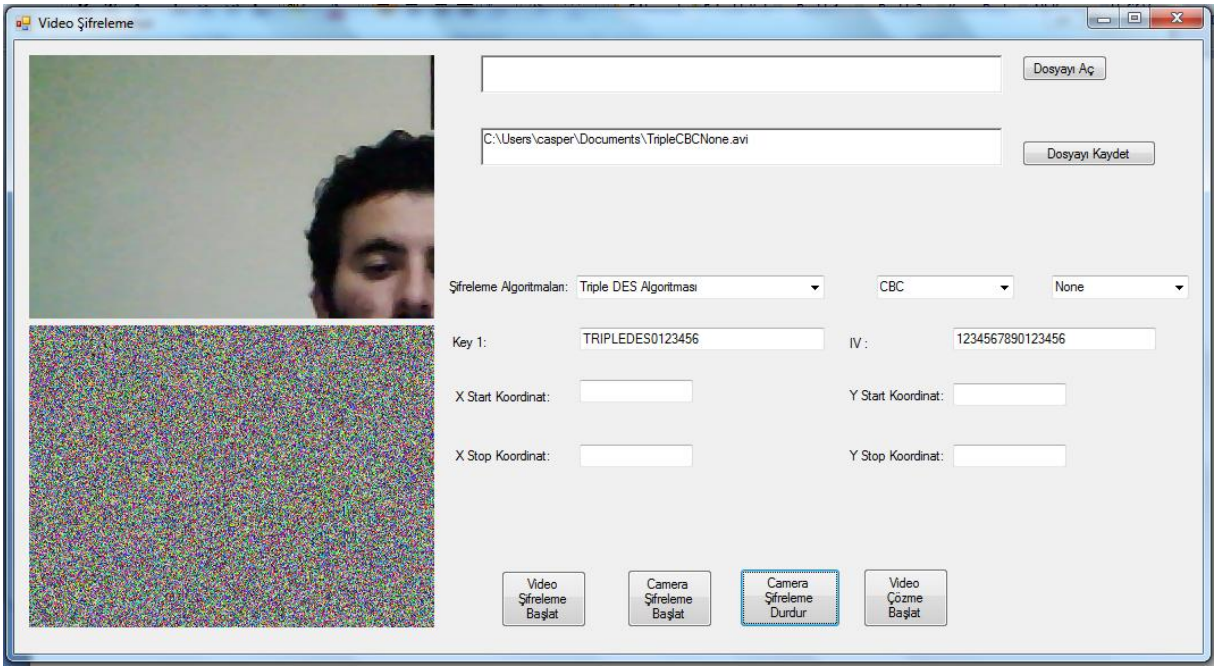
almaktadır. Her şifreleme algoritması Mode veya Padding farklı olacak şekilde verilmektedir.



Resim 25 : DES şifreleme algoritması CBC Mode PKCS7 Padding Mode

Resim 26 : PlainImage

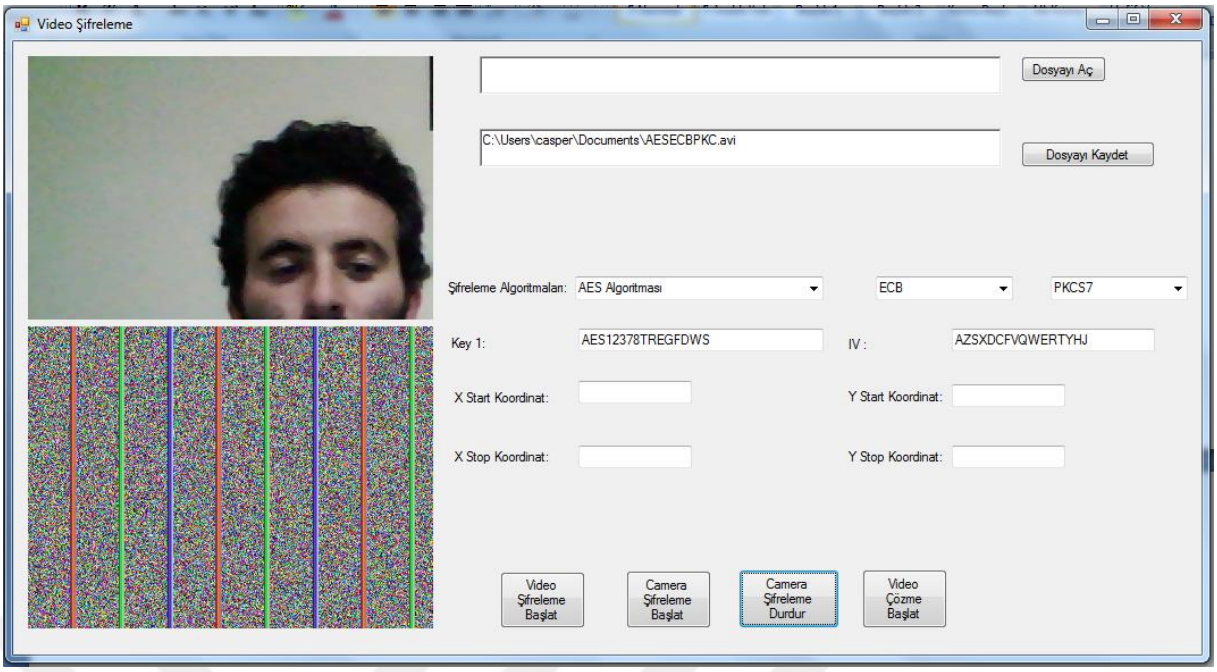
Resim 27 : CipherImage



Resim 28 : Triple şifreleme algoritması CBC Mode None Padding Mode

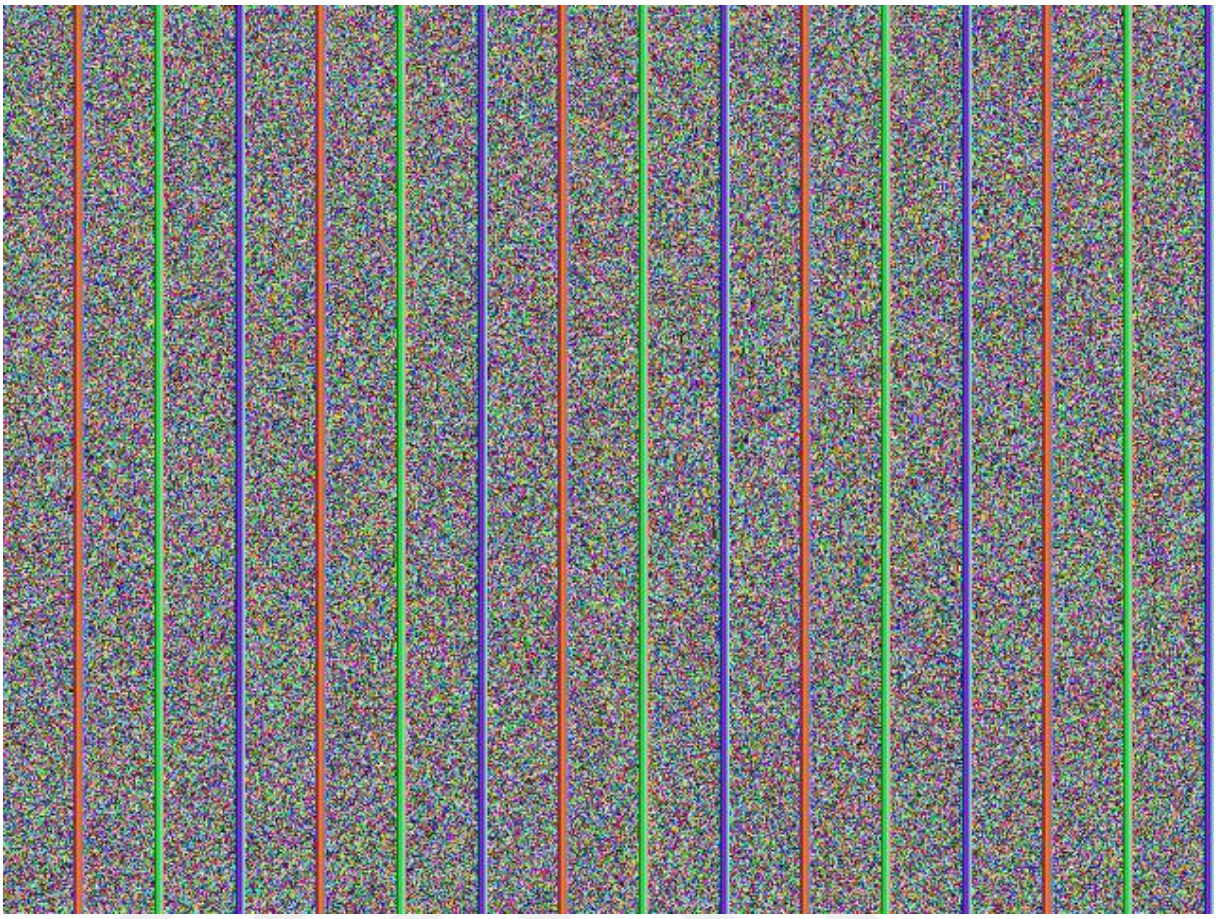
Resim 29 : PlainImage

Resim 30 : CipherImage

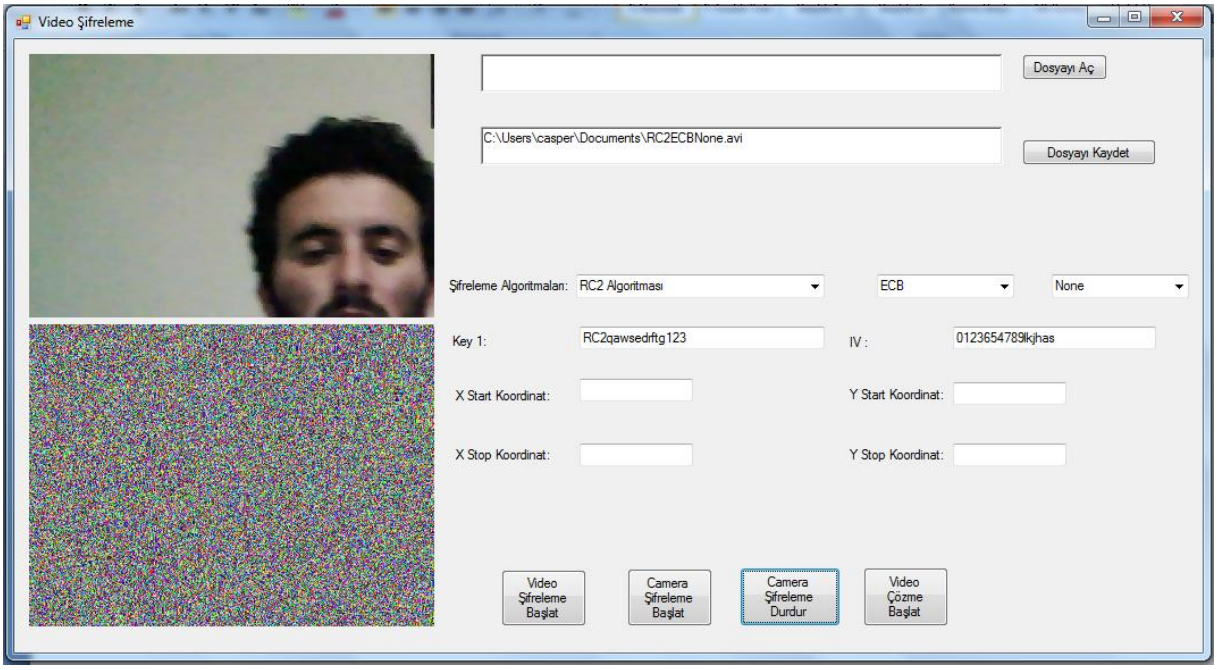


Resim 31 : AES şifreleme algoritması ECB Mode PKCS7 Padding Mode

Resim 32 : PlainImage



Resim 33 : CipherImage

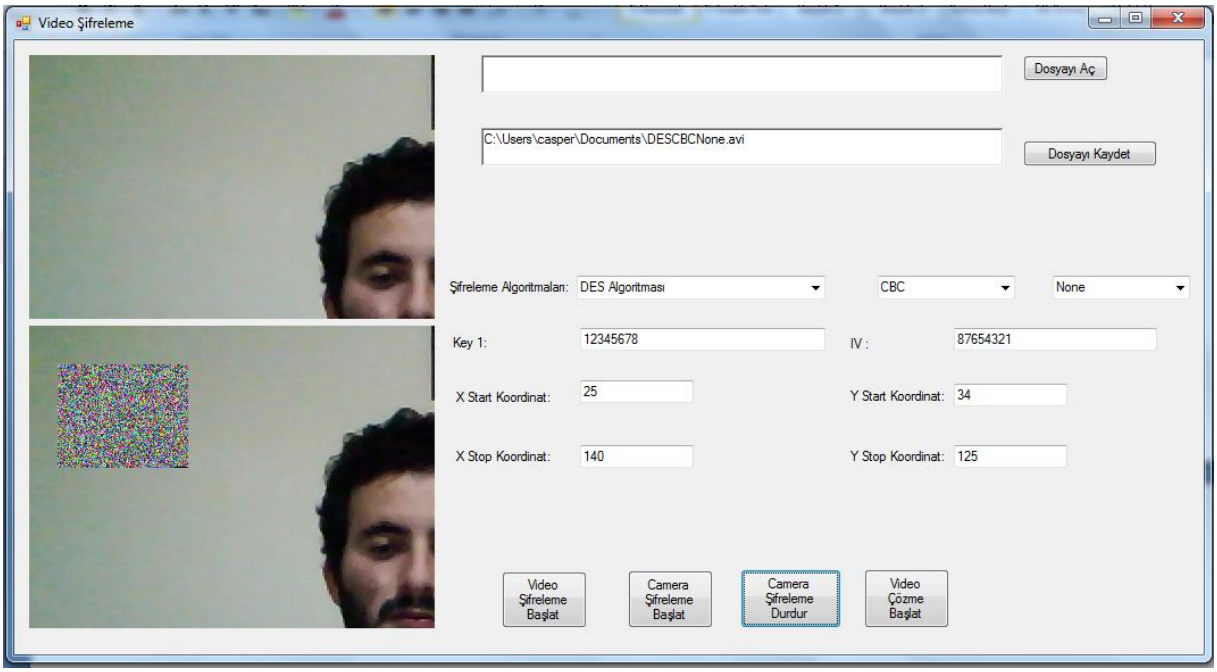


Resim 34 : RC2 şifreleme algoritması ECB Mode None Padding Mode

Resim 35 : PlainImage

Resim 36 : CipherImage

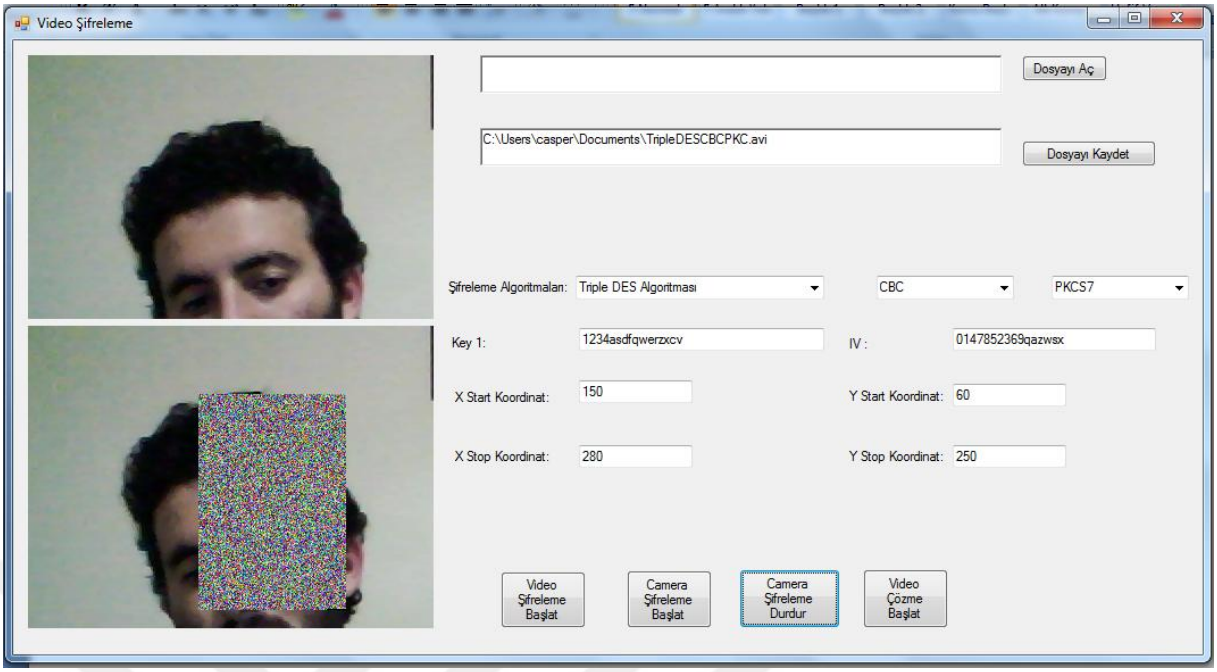
D. Dördüncü senaryonun kullanımı Resim 37 ile Resim 48 arasındaki resimlerle gösterilmiştir. Bu senaryoda koordinatlar girilerek. Kamera ile görüntüler alınarak şifrelenecektir. Resimler Sırası ile ilk Şifreleme algoritmasının olduğu uygulama, ikincisi Orjinal Resim, üçüncüsü ise Şifrelenmiş resim yer almaktadır. Her şifreleme algoritması Mode veya Padding farklı olacak şekilde verilmektedir.



Resim 37 : DES şifreleme algoritması CBC Mode None Padding Mode

Resim 38 : PlainImage

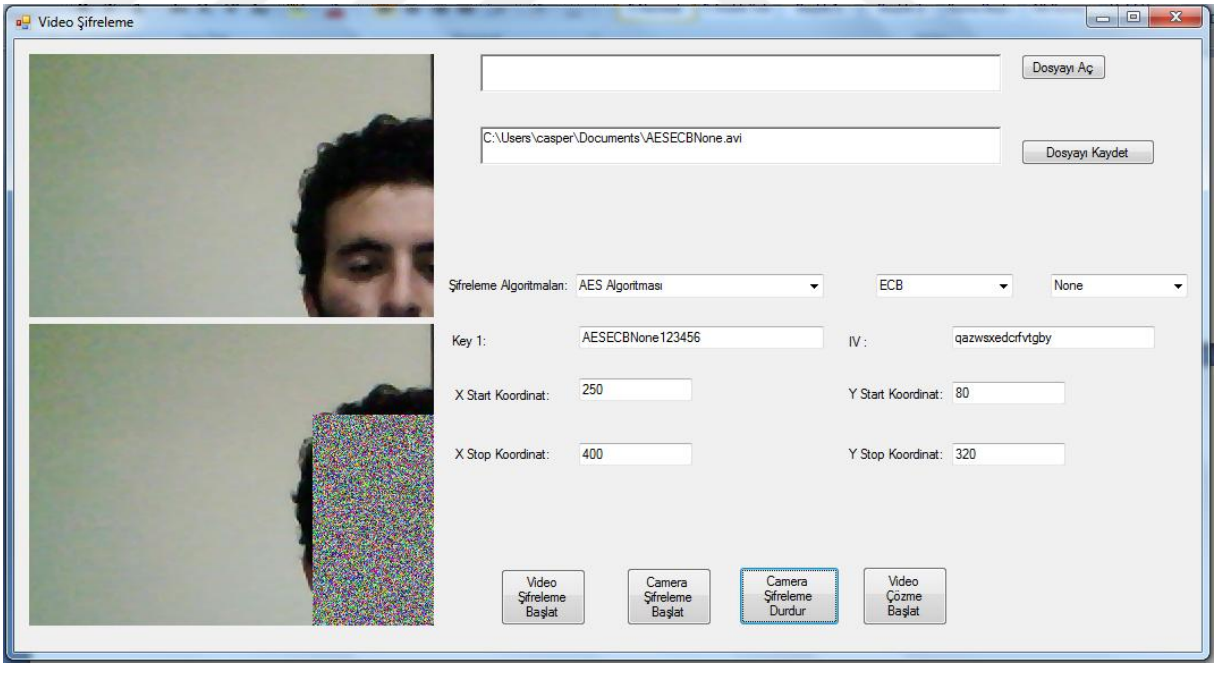
Resim 39 : CipherImage



Resim 40 : Triple DES şifreleme algoritması CBC Mode PKCS7 Padding Mode

Resim 41 : PlainImage

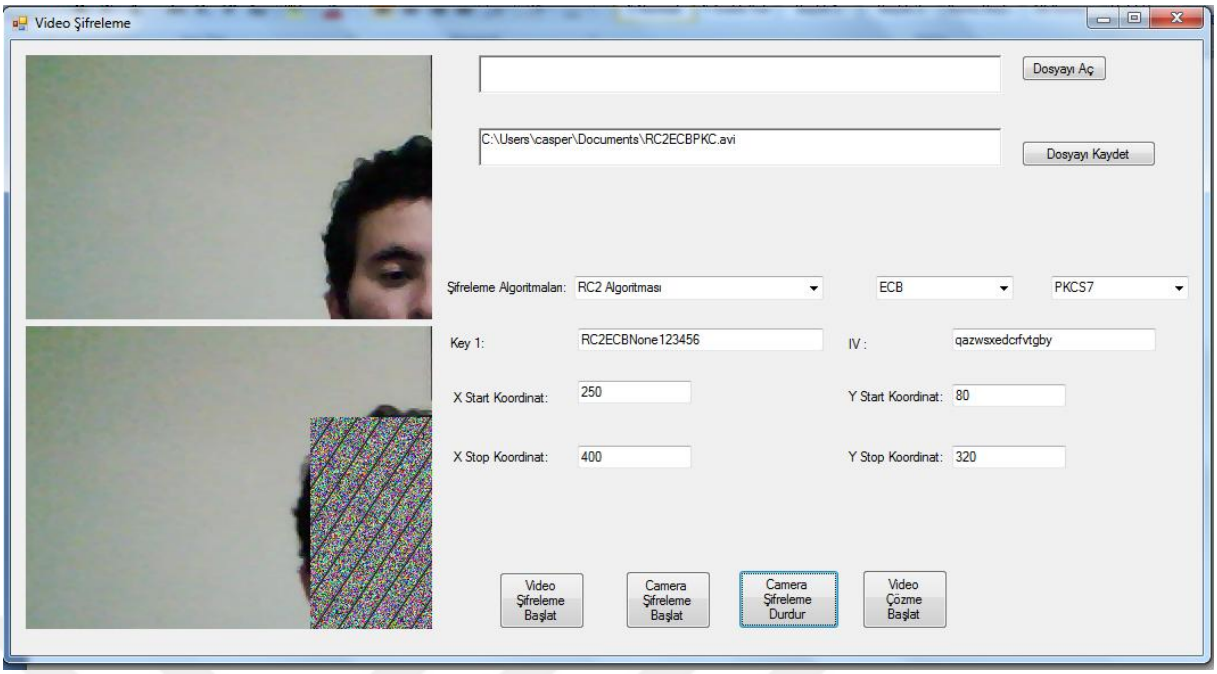
Resim 42 : CipherImage



Resim 43 : AES şifreleme algoritması ECB Mode None Padding Mode

Resim 44 : PlainImage

Resim 45 : CipherImage

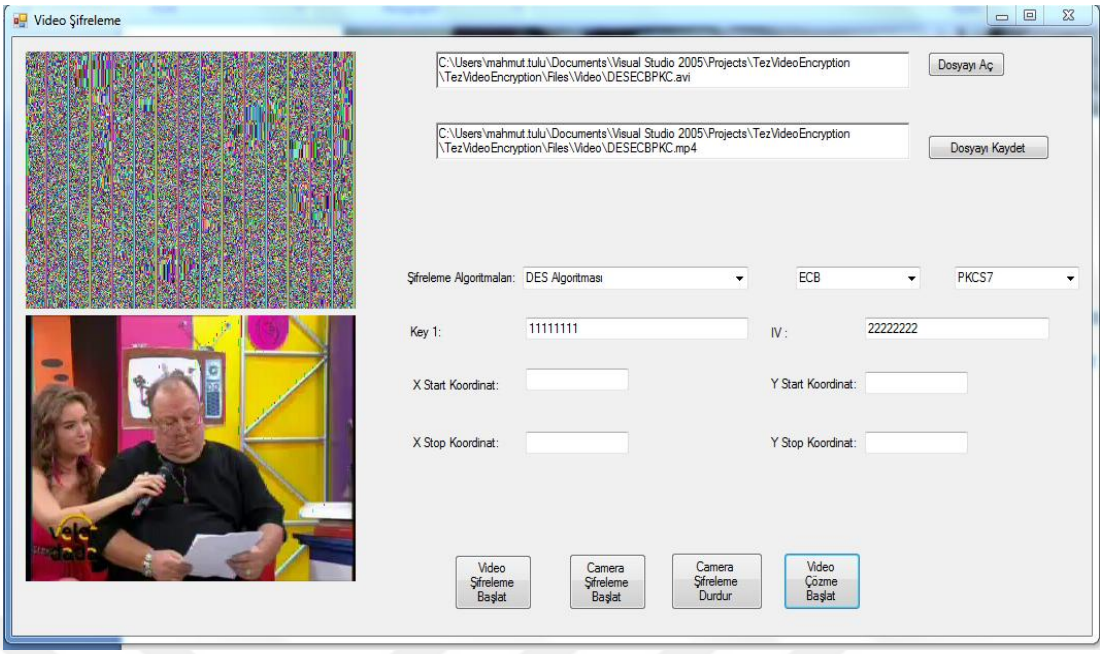


Resim 46 : RC2 şifreleme algoritması ECB Mode PKCS7 Padding Mode

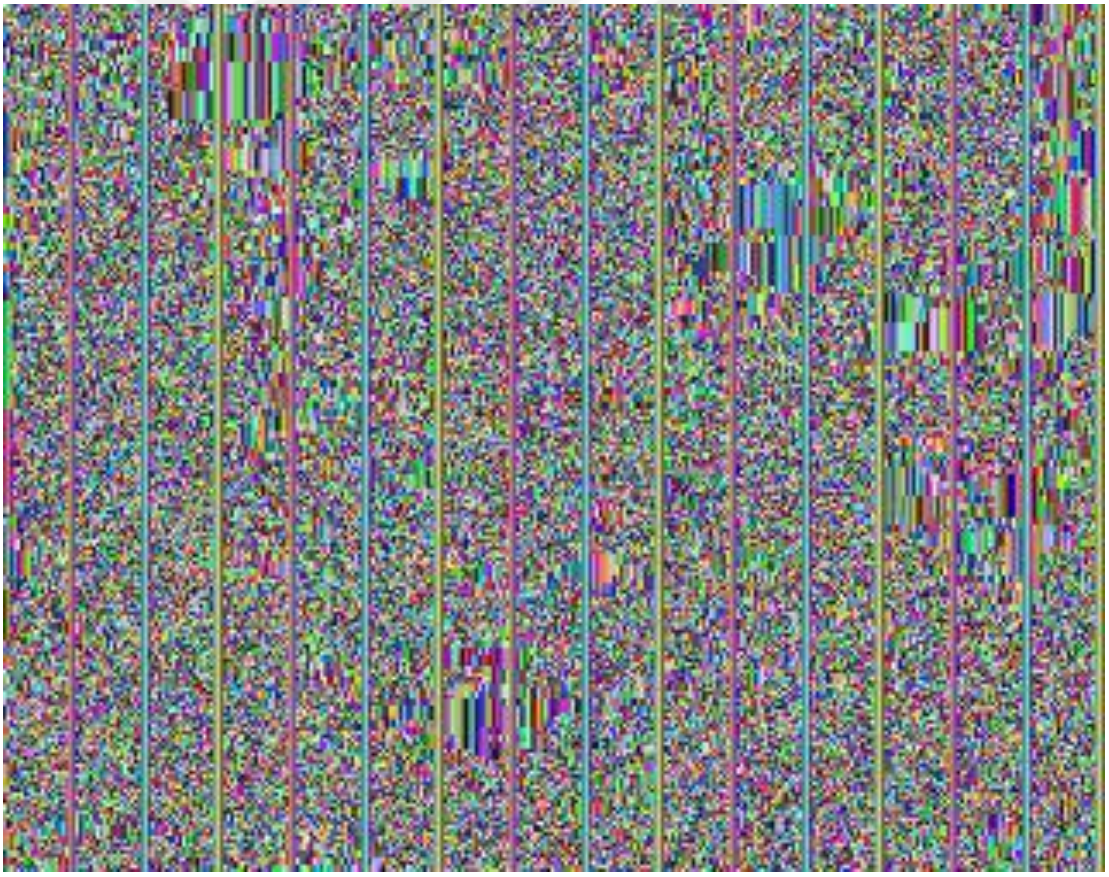
Resim 47 : PlainImage

Resim 48 : CipherImage

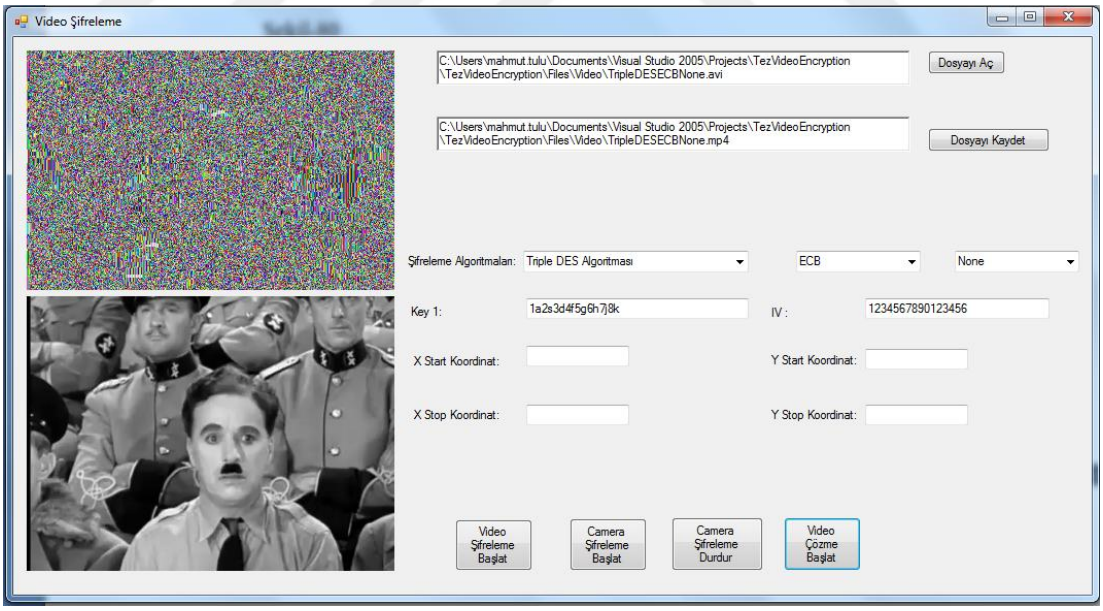
E. Beşinci senaryonun kullanımı Resim 49 ile Resim 60 arasındaki resimlerle gösterilmiştir. Bu senaryoda şifrelenmiş olan resimler orjinal haline getirilecektir. Sırası ile ilk Şifreleme algoritmasının olduğu uygulama, ikincisi Şifrelenmiş Resim, üçüncüsü ise Orjinale yakın resim yer almaktadır. Her şifreleme algoritması Mode veya Padding farklı olacak şekilde verilmektedir.



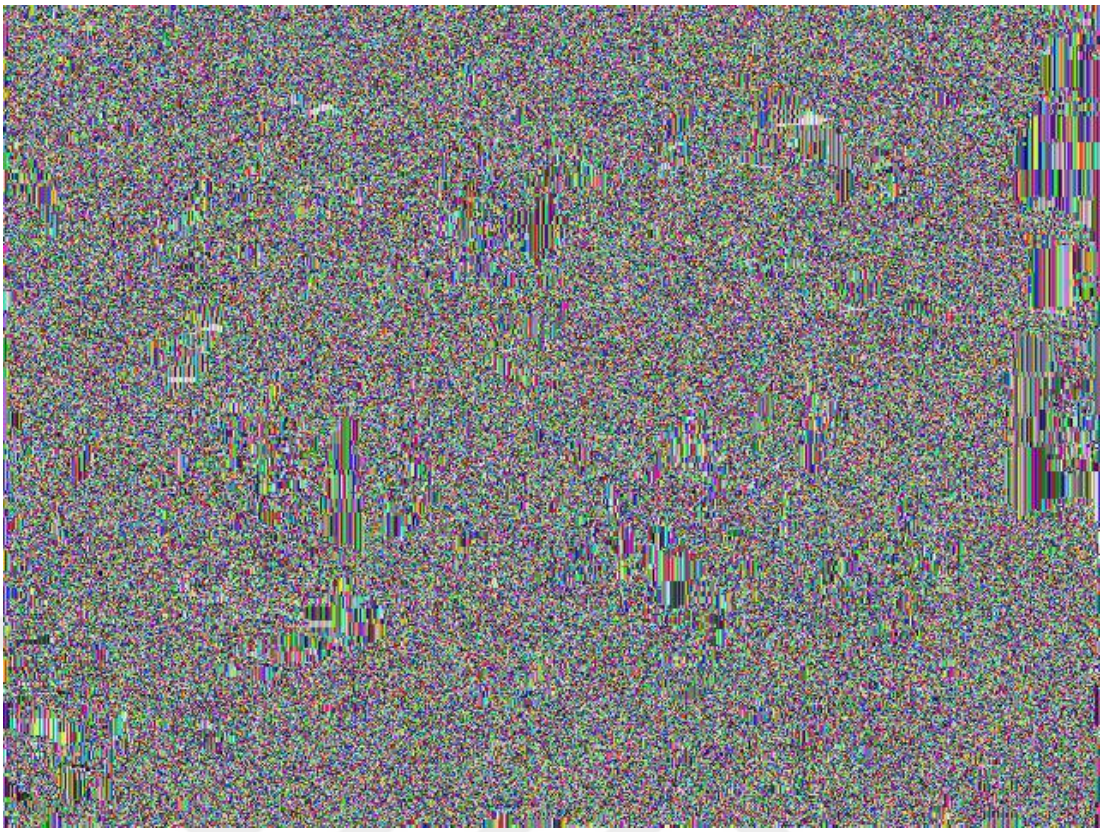
Resim 49 : DES şifreleme algoritması ECB Mode PKCS7 Padding Mode


Resim 50 : CipherImage

Resim 51 : PlainImage

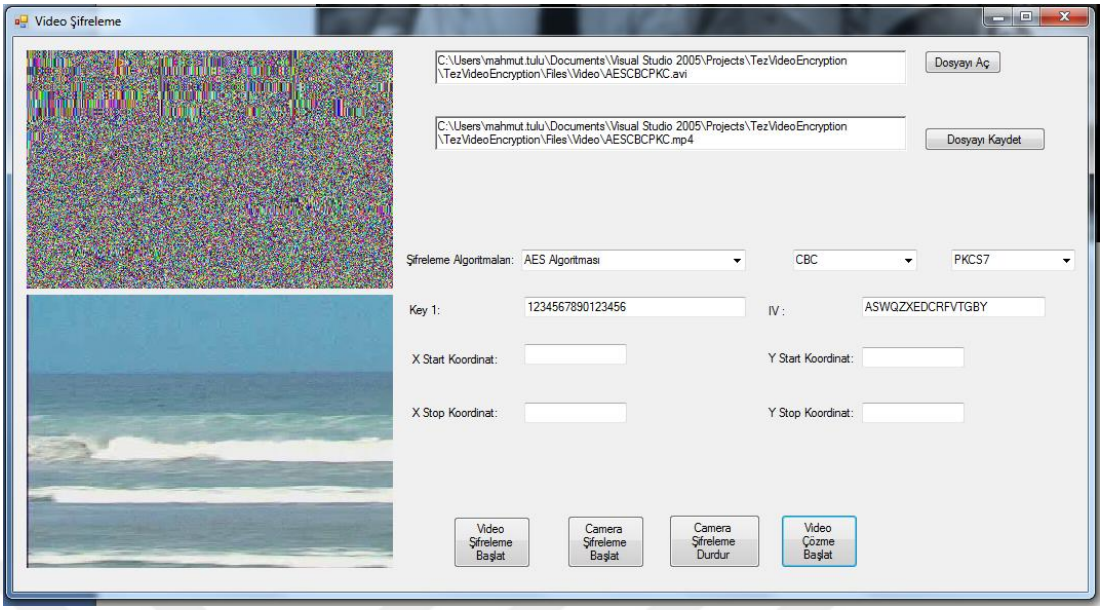


Resim 52 : TripleDES şifreleme algoritması ECB Mode None Padding Mode

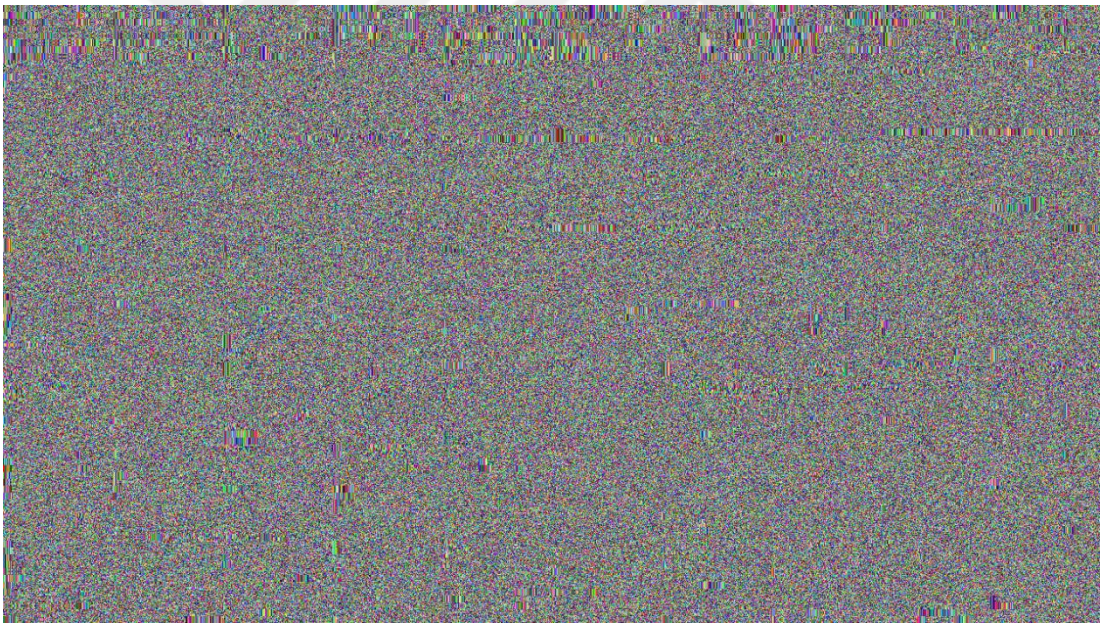


Resim 53 : CipherImage

Resim 54 : PlainImage

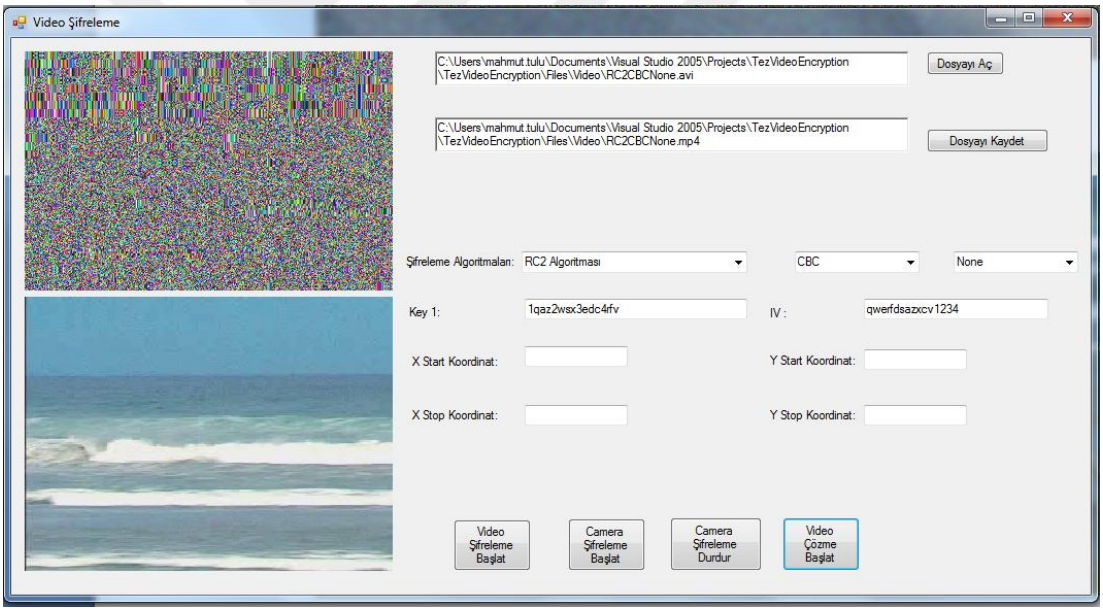


Resim 55 : AES şifreleme algoritması CBC Mode PKCS7 Padding Mode

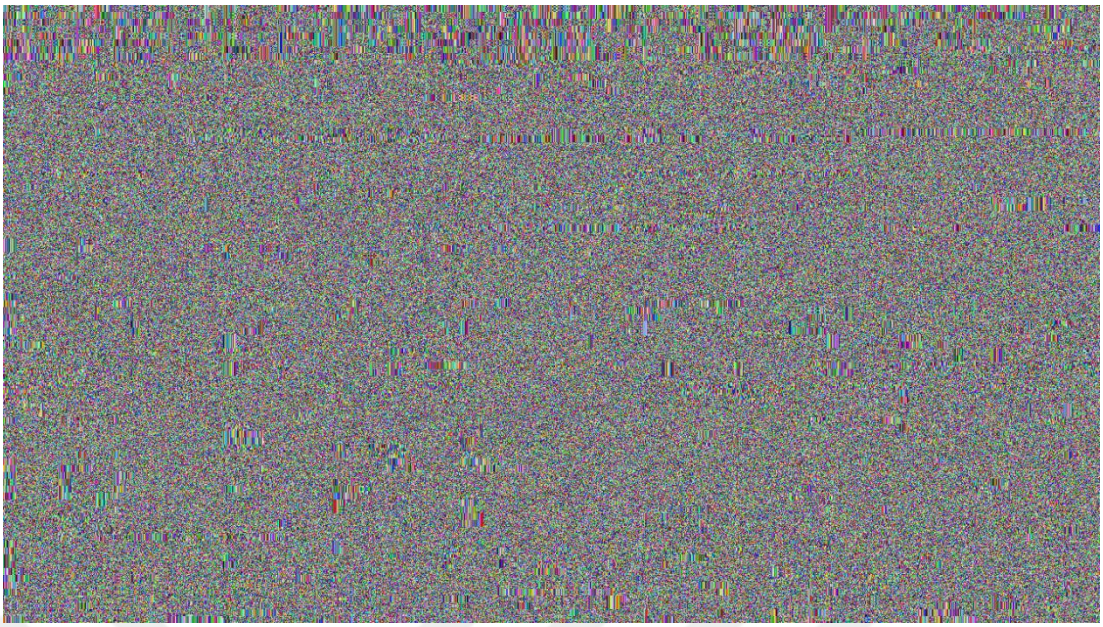


Resim 56 : CipherImage

Resim 57 : PlainImage



Resim 58 : RC2 şifreleme algoritması CBC Mode None Padding Mode



Resim 59 : PlainImage

Resim 60 : CipherImage

SONUÇLAR

Bu tezde yapmış olduğumuz çalışma simetrik şifreleme algoritmalarını kullanarak videoları veya kemaradan gelen görüntülerin şifrelemesini sağladık. Şifreleme sırasında AES şifreleme algoritmasının hızı diğer şifreleme algoritmalarına oranla daha hızlı olduğunu gördük. Simetrik şifreleme algoritmalarından AES kullanmak daha iyi bir güvenlik sağlayacak olsa da boyut açısından şifrelenmiş video verisinin çok büyük boyutlara çıktığı görülmektedir.

Simetrik şifreleme algoritmalarında Padding Mode PKCS7 olması durumunda byte kaybına uğranılmasına ve belli bir miktar görüntü bozukluğu olmasına rağmen şifrelenmiş görüntünün çözüldüğünde neyle ilgili olduğu anlaşılmaktadır.

Simetrik şifreleme algoritmalarından AES kullanılacak olması durumunda resmin belli bir oranda sıkıştırmaya yapılması durumunda ve şifrelenmiş resmin datası bozulmadan belli bir oranda sıkıştırma olması durumunda AES şifreleme algoritmasının kullanılması görsel güvenlik sistemlerinin geliştirilmesini belirli bir oranda sağlar.

KAYNAKÇALAR

1. BAŞKÖK D. M., 2007, AES Şifreleme Algoritmasının Modellenmesi
2. GÜNDEN Ü., 2010, Şifreleme Algoritmalarının Performans Analizi
3. GÜVENOĞLU E., 2006, Görüntü Şifreleme Algoritmaları ve Performans Analizleri
4. KARAKOÇ F., 2008, Kripto Analizde Melez Bir Yöntem: Çakışma Saldırısı
5. ÖZTÜRK İ., 2003, Görüntü Şifreleme
6. SAKALLI M. T., 2006, Modern Şifreleme Yöntemlerinin Gücünün İncelenmesi
7. ŞİŞMAN Ç., 2014, Veri Sıkıştırma ve Veri Şifreleme Algoritmalarının Ağ Üzerinde Beraber Kullanımı ve Performanslarının İncelenmesi
8. YERLİKAYA T., 2002, Şifreleme Teknikleri ve Güncel Uygulama Olanakları

EKLER

- C# ve Emgu cv ile yazılmış olan Simetrik Şifreleme algoritmalarını kullaranak şifreleme yapan program
- Tez içerisinde kullanılan videolar

ÖZGEÇMİŞ

19.11.1988 tarihi, İstanbul ili Bayrampaşa ilçesinde doğumluyum. İlköğretimi, Kocaeli ili Gebze ilçesinde Sultanorhan ilköğretim okulunda okudum. Liseyi, İstanbul ili Ümraniye ilçesinde Ümraniye İmam-Hatip Lisesinde okudum. Süleyman Demirel Üniversitesi Yalvaç Kılıçarslan Kampüsünde Bilgisayar Programcılığı ve Teknolojileri ÖnLisans bölümünü 2007 – 2009 yıllarında okudum. 2009-2012 yıllarında Süleyman Demirel Üniversitesi Batı Yerleşkesinde Bilgisayar Mühendisliğini okudum. 2012 yılından beri özel bir şirkette Analist Programcı olarak çalışmaktayım. 2015 yılında da Beykent Üniversitesi, Bilgisayar Mühendisiliği Anabilim Dalında yüksek lisans eğitimine başladım.

Özel ilgili alanım, bilgisayar programlama dilleri ve veritabanlarıdır.

Aday : MAHMUT TÜLÜ