1. GİRİŞ

1.1. Giriş ve Çalışmanın Amacı

Kriptografi bilgi ve haberleşme güvenliği bilimidir, güvenli olmayan bir kanal üzerinde haberleşmenin güvenliğini sağlar [1]. Genel olarak bahsedersek, kriptografik bir sistem kullanarak şifresiz metni şifreli metine çoğunlukla bir anahtar kullanarak çevirir.

Bazı şifreleme tekniklerinde anahtar çok basittir. Bunların ilk bilinen örneği ROT13 (Rotation 13'ün kısaltmasıdır)'dür [1]. Ceaser tarafından kullanılmış olan algoritma metnin her harfini alfabede 13 harf aşağıya kaydırır. Tekrar şifrelenmiş metini şifrelenirse orijinal metin elde edilir.

Açık anahtarlı şifrelemenin kullanıldığı şifrelemeye ise Rivest-Shamir-Adleman (RSA) örnek verilebilir[1] RSA metini şifrelemek için açık bir anahtar kullanır ve şifresini çözmek için gizli bir anahtar kullanır.

Kriptografi bilgisayar bilimi alanında bir çok uygulamasıyla önemli bir yer tutmaktadır. Kriptografinin en önemli örneklerinden biri Enigma makinesidir. 2. Dünya Savaşı'nda Alman 3. Reich tarafından mesajlarını şifrelemekte kullanılan algoritmanın kırılmasıyla denizaltı kuvvetleri yenilgiye uğramıştır [2].

Anahtarının küçük boyutu ve mikroişlemci gücünde teknolojik ilerlemeler sonucu zayıf kalan Data Encryption Standart (DES) 'tan sonra Amerika hükümeti tarafından yeni şifreleme standardı seçilmeye karar verilmiştir [4]. Bu standart Advanced Encryption Standart (AES) olarak adlandırılır [4]. 1998'de elli aday kabul edilmiştir ve 1999'da elenerek beş aday kalmıştır. Ekim 2000 tarihinde Rijndael'in hafif değiştirilmiş versionu olan AES standart haline gelmiştir [4].

Bu çalışmada amaç AES algoritması'nın incelenerek 8 bitlik bir mikroişlemci üzerinde gerçeklenmesi ve bilgisayarla haberleşme kurabilmesidir. Bilgisayarda yapılan şifrelemenin güvensiz olduğu düşünüldüğünden şifreleme yapmak için ayrıca bir mikroişlemci kullanılmıştır. Bilgisayarda sadece şifrelenecek veya şifresi

çözülecek metin bulunacaktır. Bu metin uygun bir haberleşme protokolü seçilerek içinde şifreleme algoritması olan mikroişlemciye gönderilecektir. Mikroişlemci düz metini alırsa şifreleyip, şifrelenmiş metin alırsa düz metine çevirip bilgisayara yollayacaktır. AES algoritması ve haberleşmeyi gerçeklemek için C dili kullanılacaktır.

2. SONLU UZAY ARİTMETİĞİ

AES algoritmasındaki bütün baytlar sonlu uzay elemanı olarak değerlendirilir. Sonlu uzay elemanları toplanabilir ve çıkarılabilir fakat bu, sayılarla yaptığımız işlemlerden farklıdır.

2.1.Toplama

Sonlu alanda iki elemanın toplanması, iki elemanın polinomlarının aynı üstel kuvvete sahip x'lerin katsayılarının toplanması ile bulunur. Bu toplama XOR işlemi ile gerçekleştirilir. $1\oplus1=0$, $1\oplus0=1$, $0\oplus1=1$ ve $0\oplus0=0$ olduğundan dolayı katsayıları toplamı 2 olan terimler yok olur. Toplamanın polinominal olarak, ikilik düzende ve hexadecimal düzende gösterim şekilleri aşağıdaki gibidir .

$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2$ (polinominal gösterim)

$\{01010111\}\oplus\{10000011\} = \{11010100\}$ (İkilik düzende gösterim)

$\{57\}\oplus\{83\} = \{d4\}$ (heksadesimal düzende gösterim)

2.2.Çarpma

$GF(2^8)$'de çarpım, iki polinomun çarpımlarının 8. dereceden indirgenmez polinom modülünün alınmasıyla bulunur. Bir polinomun bölenleri yalnız bir ve kendisiyse indirgenmezdir. AES algoritması için indirgenmez polinomu denklem (2.1)'de verilmiştir.

$m(x) = x^8 + x^4 + x^3 + x + 1$ (2.1)

Bu polinomun heksadesimal karşılığı ise {01} {1b}'dir.

Örneğin $\{57\}\oplus\{83\} = \{c1\}$ yapar çünkü;

$(x^6 \; x^4 \; x^2 \; x \; 1)\,(x^7 \; x \; 1) = x^{13} \; x^{11} \; x^9 \; x^8 \; x^7$

$x^7 \; x^5 \; x^3 \; x^2 \; x$

$x^6 \; x^4 \; x^2 \; x \; 1$

$= x^{13} \; x^{11} \; x^9 \; x^8 \; x^6 \; x^5 \; x^4 \; x^3 \; 1$

$x^{13} \; x^{11} \; x^9 \; x^8 \; x^6 \; x^5 \; x^4 \; x^3 \; 1 \bmod(x^8 + x^4 + x^3 + x + 1) = x^7 + x^6 + 1$

m(x) polinomuyla yapılan indirgeme sayesinde ikilik düzendeki polinomun derecesi 8'den az olur ve de bir bayt ile temsil edilebilir. Toplamadaki gibi bayt seviyesinde çarpmaya karşılık gelen işlem yoktur.

2.3 Bölme İşlemi

Klasik bölme işlemi Şekil 2.1'de gösterilmiştir:

$$
\begin{array}{rl}
 & (x^{13} + x^8 + x^7 + x^4 + x^3 + 1) : (x^8 + x^4 + x^3 + x + 1) = x^5 - x \\
\hline
- & (x^{13} + x^9 + x^8 + x^6 + x^5) \\
\hline
 & -x^9 + x^7 - x^6 - x^5 + x^4 + x^3 + 1 \\
- & (-x^9 - x^5 - x^4 - x^2 - x) \\
\hline
 & x^7 - x^6 + 2x^4 + x^3 + x^2 + x + 1
\end{array}
$$

Şekil 2.1. Klasik Bölme İşlemi

Bölünen polinomun en büyük üstel değeri (x^{13}) , bölenin en yüksek üstel değeri (x^8) ile bölünür ve sonuç (x^5) olarak elde edilir. Daha sonra (x^5) bölenin tüm değerleri ile çarpılarak bölünen polinomunda çıkarılır ve yeni bir bölünen polinomu elde edilir.

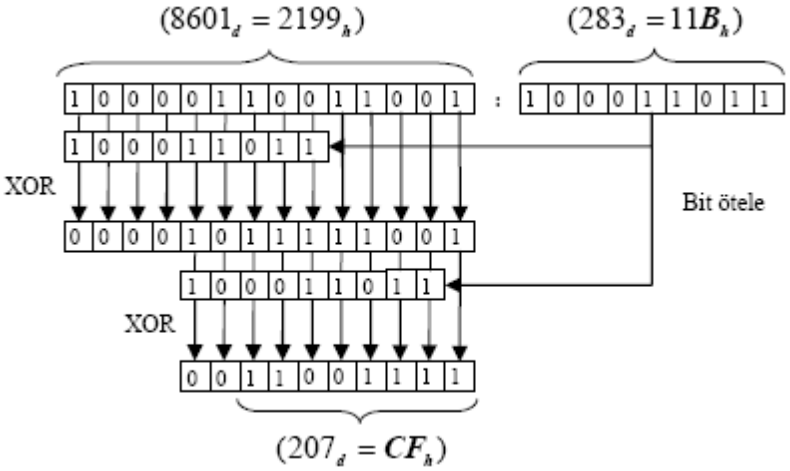
Daha sonra aynı işlemler tekrar yapılır ve bu işlem bölünen polinomunun en yüksek üstel değeri bölenin en yüksek üstel değerinden küçük değerde oluncaya kadar devam eder. En sonda kalan bölünen, işlem sonucunda kalan polinomunu oluşturur:

$(-x^9 + x^7 - x^6 - x^5 + x^4 + x^3 + 1)$

Sonuç polinomuna genel xor işlemi uygulanarak (tek katsayılar > 1 , çift katsayılar > 0 işlemi) bayt gösteriliminde sonuç elde edilir:

$(x^7 + x^6 + x^3 + x^2 + x + 1)$

Bit seviyesinde işlemler Şekil 2.2 'de gösterilmiştir:



Şekil 2.2. Bit Seviyesindeki İşlemler

3. AES – GELİŞMİŞ ŞİFRELEME STANDARDI

3.1. Giriş

Blok denilen belirli bir uzunluğa sahip bit grupları üzerinde çalışan bir Blok Şifreleme Algoritması olan Rijndael ismini iki Belçikalı muciti Joan Daemen ve Vincent Rijmen'dan almaktadır. Giriş olarak belirli bir büyüklükte olan bloğu alır ve çıkış olarak aynı büyüklükte bir blok üretir. Dönüşüm ikinci bir giriş olan gizli anahtarı gerektirir. Gizli anahtar herhangibir boyutta olabilir. AES üç farklı anahtar boyunu kullanmaktadır: 128, 192 ve 256 bit.

3.1.1. Simetrik Ve Asimetrik Algoritmalar

Tüm modern algoritmalar şifreleme ve çözme işlemlerini kontrol etmek için bir anahtar kullanırlar. Bir mesaj sadece kullanılan anahtar şifreleme anahtarıyla uyuştuğunda çözülebilir. Anahtar temelli algoritmaların iki çeşidi vardır. Bunlar simetrik (veya gizli-anahtar) ve asimetrik (veya açık-anahtar) algoritmalardır [3]. Aralarındaki fark, simetrik algoritmalar şifreleme ve çözme işlemleri için aynı anahtarı kullanırken (veya çözme anahtarı şifreleme anahtarından kolayca türetilebilir), asimetrik algoritmalar şifreleme ve çözme için farklı anahtar kullanırlar ve çözme anahtarı şifreleme anahtarından elde edilemez. Simetrik algoritmalar dizi şifreleyiciler ve blok şifreleyiciler olarak ikiye ayrılabilir. Dizi şifreleyiciler belli bir anda bir bitlik düz-metni şifreleyebilirken, blok şifreleyiciler pek çok biti alıp bunları tek bir birim olarak şifrelerler. Asimetrik şifreleyiciler (açık-anahtar algoritmaları veya açık-anahtar kriptografisi olarak da adlandırılırlar) şifreleme anahtarının halka açık olmasına isteyen herkesin bu anahtarı kullanarak şifreleme yapmasına izin verirken, sadece uygun veya istenilen alıcı mesajı çözebilir [3]. Şifreleme anahtarı açık-anahtar, ve çözme anahtarı da özel veya gizli anahtar olarak da adlandırılır. Güçlü kriptografik algoritmalar bilgisayarla veya özelleştirilmiş cihazlarda çalıştırılmak üzere tasarlanmaktadır. Pek çok uygulamada kriptografi, bilgisayar yazılımlarıyla yapılmaktadır. Genel olarak, simetrik algoritmalar bilgisayarda asimetrik olanlardan çok daha hızlıdırlar. Uygulamada bunlar sık sık beraber kullanılırlar. Örneğin açık-anahtar algoritmalar rasgele üretilmiş bir şifreleme anahtarını şifrelemek için kullanılır ve rasgele anahtar hakiki mesajı simetrik bir algoritma kullanarak şifrelemek için kullanılır. Bu bazen,

hibrid(melez) şifreleme olarak adlandırılır. En çok çalışılan ve muhtemelen en yaygın simetrik şifreleme DES tir; yeni geliştirilen AES en yaygın şifreleme algoritması olarak DES in yerini alabilir. RSA muhtemelen en iyi bilinen asimetrik şifreleme algoritmasıdır [3].

3.2.2. Kriptografik Algoritmaların Güvenliği

İyi kriptografik sistemler kırılması zor olucak şekilde tasarlanmalıdırlar. Teoride, herhangi bir anahtarlı kriptografik metot olası tüm anahtarların denenmesi ile kırılabilir. Eğer tüm anahtarların denendiği kaba kuvvet kullanımı tek yolsa, gerekli hesaplama gücü anahtarın uzunluğu ile üstel olarak artar. N anahtarın kaç bitlik olduğunu belirtirse; 2^N olabilecek bütün anahtar kombinosyanlarının sayısıdır. Öyleyse 32 bitlik bir anahtar, 2^{32} = 4,294,967,296 adım alır. Bu herhangi bir ev bilgisayarı ile yapılabilecek bir şeydir. 40 bitlik anahtarlar $1{,}09.10^{12}$ adım alır. Bu tür bir hesaplama (kullanılan algoritmanın etkinliğine bağlı olarak) modern bir ev bilgisayarında bir hafta gibi bir zaman gerektirir. 56 bit anahtarlı bir sistem (DES gibi) esaslı bir zahmet gerektirir (çok sayıda ev bilgisayarının güç paylaşımı ile bunu kırmak birkaç ay alır), ama özel donanımlarla kolayca kırılabilir. Özel donanımların maliyetleri de doğal olarak yüksektir, ama organize suç örgütleri, büyük hükümet ve şirketler bunları alabilirler. 64 bitli anahtarlarda şimdiden kırılabilir durumdadırlar. 80 bitli anahtarlar bir kaç yıl sonra kırılabilecekken ve 128 bitli anahtarlar kaba kuvvet ile kırılması zordur. Ancak anahtar uzunluğu tek önemli konu değildir. Pek çok şifreleme olası tüm anahtarlar denenmeden de kırılabilir. Genelde, diğer metotların daha da etkili kullanımı ile bile kırılamayacak şifreleme tasarlamak çok zordur. Çoğunlukla, algoritmanın gizliliğine dayanan algoritmalar güvenli değillerdir. Açık-anahtarlı kriptografide kullanılan anahtarların uzunluğu simetrik cipher'larda kullanılanlardan daha uzundur. Bunun nedeni, kriptanalistler için kullanılan ekstra yapıdır. Burada problem doğru anahtarın tahmin edilmesi değil, gizli anahtarın açık-anahtardan türetilmesidir. RSA da bu işlem iki asal çarpanı olan bir tamsayının üretilmesi ile yapılmaktadır. RSA kripto-sisteminin karmaşıklığı hakkında biraz bilgi vermek gerekirse, 256 bitlik bir modulus evde kolayca ve 512 bitlik anahtarlar üniversitedeki araştırma grupları tarafından birkaç ay içinde kırılabilir. 768 bitlik anahtarlar muhtemelen uzun vadede güvende sayılmazlar. 1024 ve daha büyük bitli anahtarlar RSA ya karşı büyük kriptografik ilerlemeler kaydedilmedikçe güvende sayılırlar [3].

3.3.Gelişmiş Şifreleme Standardı Algoritmasının Tanımı

3.3.1. Giriş Ve Çıkışlar

AES algoritmasında giriş ve çıkışlar 128 bitlik dizilerden oluşur. Bu diziler bazen blok olarak değerlendirilir ve içerdikleri bit sayısı uzunlukları olur. AES algoritması için şifreleme anahtarı 128, 192 veya 256 bitlik dizilerden oluşur. Diğer giriş, çıkış ve şifre anahtarı uzunluklarının kullanılmasına izin verilmez. Dizilerdeki bitler numaralandırılırken sıfırdan başlanır ve dizi uzunluğunun bir azı ile biter. Bite ilişkilendirilmiş i sayısı indekstir ve dizinin uzunluğuna göre $0 < i < 128$, $0 < i < 192$ veya $0 < i < 256$ arasındadır.

3.3.2. Baytlar

AES algoritması için en basit parça bayttır. Bayt sekiz bitlik bir dizidir ve bütün olarak ele alınır. Giriş, çıkış ve şifreleme anahtarları bit dizilerinde, sekizer bitler biraraya gelerek bayt dizilerini oluşturur. Giriş, çıkış veya şifreleme anahtarı bir a[n] dizisi olarak tanımlanırsa. "n" sayısı bit uzunluğuna göre değişiklik gösterir.

Anahtar uzunluğu = 128 bit, $0< n < 16$;

Anahtar uzunluğu = 192 bit, $0< n <24$;

Anahtar uzunluğu = 256 bit, $0< n <32$;

AES algoritmasına bayt değerleri $\{b7, b6, b5, b4, b3, b2, b1, b0\}$ şeklinde oluştuğu bitler ile temsil edilir. Bu baytlar polinom temsili ile sonlu alan elemanı olarak değerlendirilir.

$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_ix^i$ (3.1)

Örneğin {01100011} baytı sonlu alan elemanı x^6+x^5+x+1'i tanımlar. Ayrıca bayt değerlerini hexadesimal değerler ile gösterme mümkündür. İlk dört bit ve son dört bit birer hexadesimal karakterle gösterilir. Heksadesimal karakterlerin ikilik düzendeki karşılıkları aşağıdaki gibi olur.

$(0000)_2 = (0)_{16}$

$(0001)_2 = (1)_{16}$

$(0010)_2 = (2)_{16}$

$(0011)_2 = (3)_{16}$

$(0100)_2 = (4)_{16}$

$(0101)_2 = (5)_{16}$

$(0110)_2 = (6)_{16}$

$(0111)_2 = (7)_{16}$

$(1000)_2 = (8)_{16}$

$(1001)_2 = (9)_{16}$

$(1010)_2 = (a)_{16}$

$(1011)_2 = (b)_{16}$

$(1100)_2 = (c)_{16}$

$(1101)_2 = (d)_{16}$

$(1110)_2 = (e)_{16}$

$(1111)_2 = (f)_{16}$

3.3.3.Algoritmanın Tanımı

AES 128 bit sabit blok büyüklüğüne sahip ve değişen anahtar uzunluğuna sahip bir blok şifrelemesidir[5] Farklı dönüşümler ara sonuçlar olan durumda üzerinde çalıştırılır. Durumlar baytlardan oluşan 4x4 boyutunda dikdörtgen dizilerdir. 128 bit yani 16 bayt büyüklüğündedirler. Rijndael versiyonunda blok büyüklükleri değişir; sıra sayısı dörtle sabittir fakat sütun sayısı değişebilir. Şifre anahtarı benzer bir şekilde dört satırı olan dikdörtgen diziden oluşmaktadır. Sütun sayısı ise N_k ile temsil edilir ve anahtar uzunluğunun 32'ye bölünmesiyle bulunur. Durum ve anahtar dizileri tablo 3.2'de verilmiştir.

Tablo 3.2. Durum ve Anahtar Dizileri

Durum:

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

Anahtar:

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

Şifrelemenin giriş baytları, durum baytları üzerine $a_{0,0}$, $a_{1,0}$, $a_{2,0}$, $a_{3,0}$, $a_{0,1}$, $a_{1,1}$, $a_{2,1}$, $a_{3,1}$, ... sırasıyla, şifre anahtarı baytları ise $k_{0,0}$, $k_{1,0}$, $k_{2,0}$, $k_{3,0}$, $k_{0,1}$, $k_{1,1}$, $k_{2,1}$, $k_{3,1}$ ... sırasıyla diziyle işleme girer. Şifreleme işleminin sonunda çıkış, durumdan baytlar aynı sırayla alarak oluşturulur. AES değişen tur sayıları kullanır. 128 bitlik anahtar için 10, 192 bitlik anahtar için 12, 256 bitlik anahtar için 14 tur kullanılır. Her tur boyunca aşağıdaki işlemler duruma uygulanır.

1. BaytYerDeğiştirme: Durumun her baytı Rijndael'in S-Matrisi kullanılarak yeni değeriyle değiştirilir.

2. SatırKaydırma: 4x4 dizideki her satır belli bir miktar sola kaydırılır.

3. SütunKarıştırma: Durumdaki sütunların lineer bir dönüşümüdür.

4. TurAnahtarıEkleme: Durumun her baytı Rijndael'in anahtar sürecinden çıkartılan ve her tur için farklı olan tur anahtarıyla birleştirilir.

3.4. AES İşlemleri : Bayt Yer değiştirme, Satır Kaydırma, Sütun Karıştırma, Tur Anahtarı Ekleme

3.4.1.Tur Anahtarı Ekleme İşlemi

Bu işlemde Tur Anahtarı duruma bit bit XOR'lanır. Tur Anahtarı Şifreleme Anahtarından anahtar üreteci sayesinde türetilir. Tur anahtarının uzunluğu blok anahtarının uzunluğuna eşittir (=16 bayt). Bu işlemi Tablo 3.3'teki gibi gösterebiliriz.

Tablo 3.3: $b(i,j)=a(i,j) \oplus k(i,j)$ işlemi

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$\oplus$	$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$	=	$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$		$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$		$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$		$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$		$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$		$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$		$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

3.4.2. Satır Kaydırma işlemi

Bu işlemde, durumun her satırı döngüsel olarak satır indeksine göre sola kaydırılır.

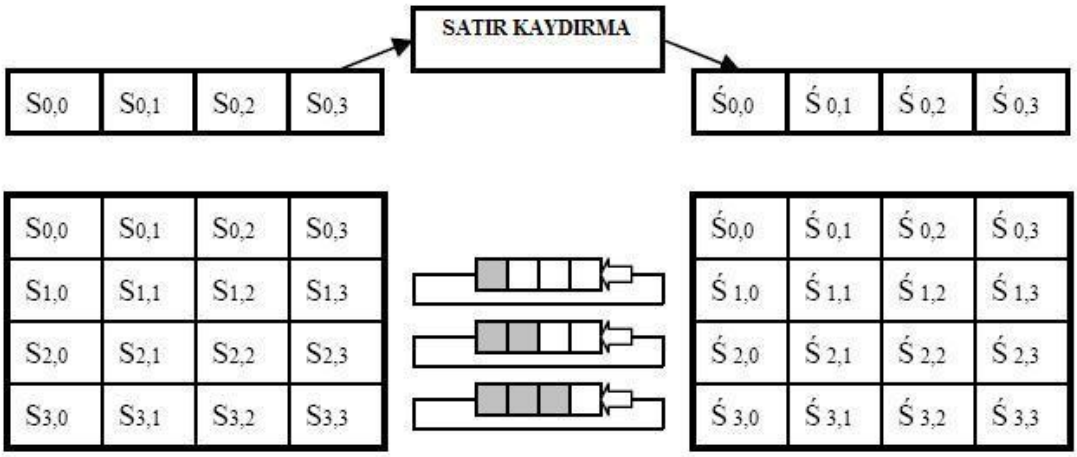
1.Satır 0 pozisyon sola kaydırılır.

2.Satır 1 pozisyon sola kaydırılır.

3. Satır 2 pozisyon sola kaydırılır.

4. Satır 3 pozisyon sola kaydırılır.

Bu işlem grafiksel olarak şekil 3.1.'deki gibi temsil edilebilir.



Şekil 3.1. Satır Kaydırma İşlemi [3]

3.4.3. Bayt Yer Değiştirme İşlemi:

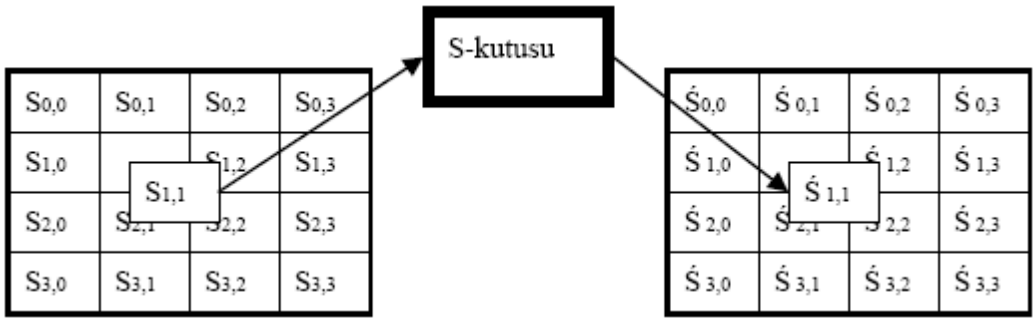
Durumun her bayt'ı üzerinde uygulanan Bayt Yer Değiştirme işlemi lineer değildir. S-Matrisi tersi alınabilir bir matristir ve iki dönüşümün karışımından oluşmuştur:

1. Rijndael'in sonlu alanında çarpmaya göre ters alınır.

2. Rijndael 'de açıklanan ilgin dönüşüm yapılır.

Eğer yeteri kadar hafıza varsa (S-Matrisi için 256 bayt) S-Matrisi herhangibir girişten bağımsız olduğu için önceden hesaplanmış tablolar kullanılır. Durumun her bayt'ı S-Matrisi'ndeki indekse karşılık düşen değerle değiştirilir. Denklem 3.1.'teki işlem şekil 3.2. ile gösterilebilir.

$a_{(i,j)}=Sbox[a_{(i,j)}]$ (3.1.)

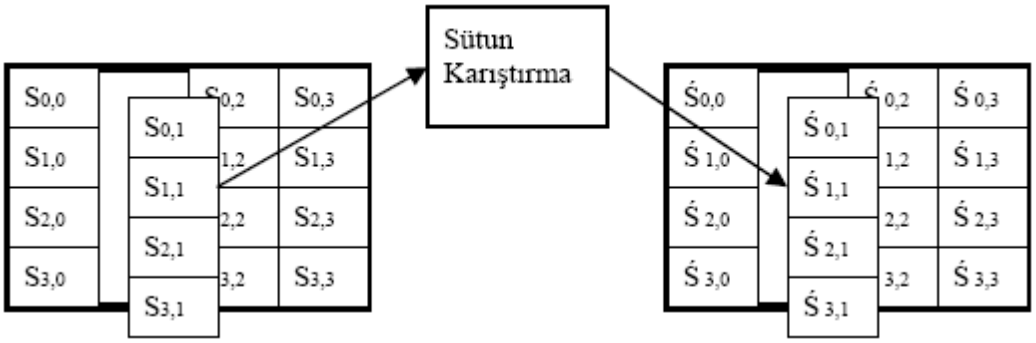


Şekil 3.2. Bayt Yer Değiştirme İşlemi

3.4.4. Sütun Karıştırma İşlemi:

Hesaplamalar Rijndael'in sonlu alanında yapılır. Aşağıdaki matrisle çarpılmaya eş düşer. Şekil 3.3.'teki işlem yapılır.

$$\begin{vmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{vmatrix}$$



Şekil 3.3. Sütun Karıştırma

3.5. Rijndael Anahtar Süreci:

Anahtar Süreci kısa anahtarı parçaları farklı iterasyonlar sırasında kullanılan daha geniş anahtara genişletmekten sorumludur. Her anahtar boyutu farklı bir boyutuna genişletilir:

128 bit anahtar 176 byte anahtara

192 bit anahtar 208 byte anahtara

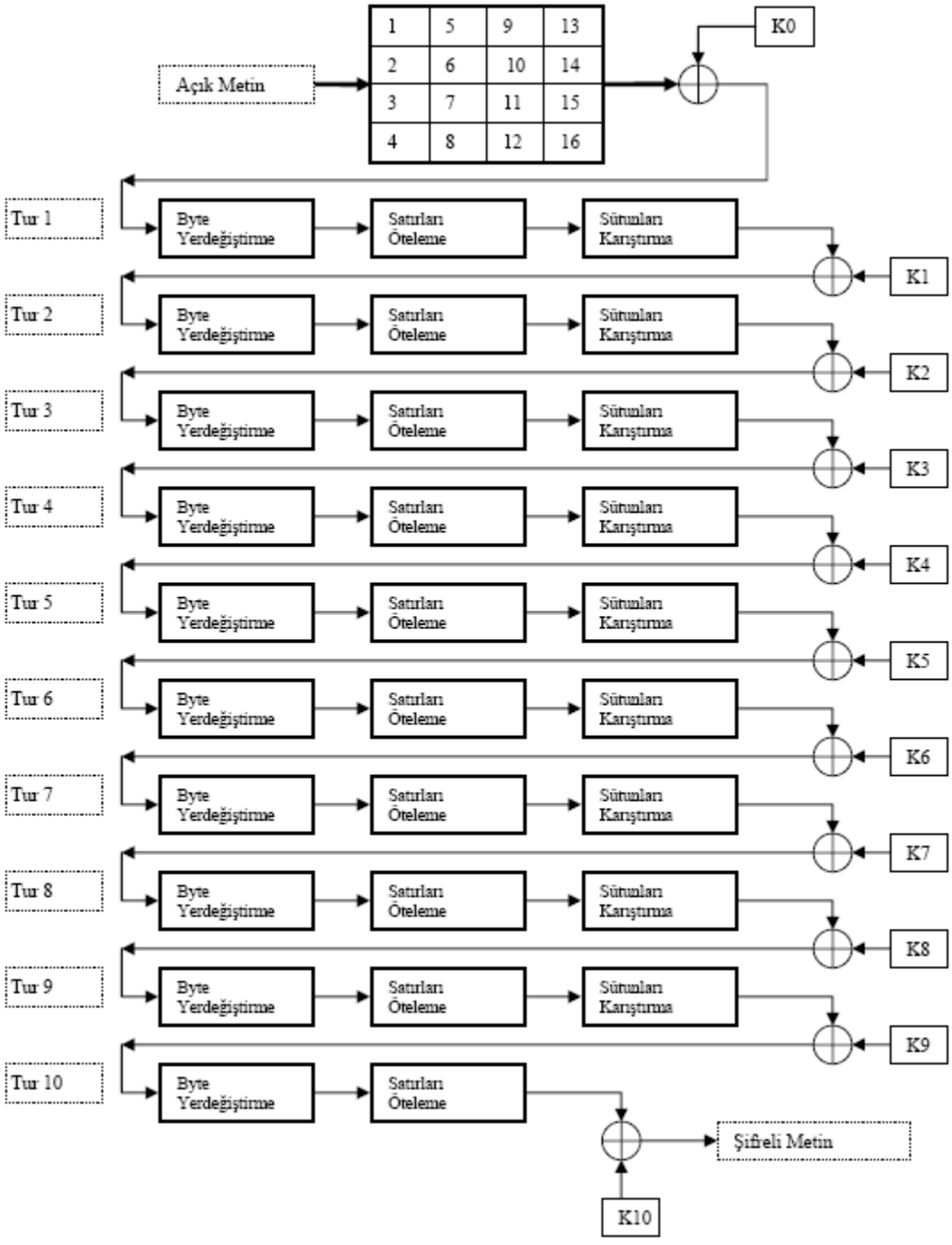
256 bit anahtar 240 byte anahtara

genişletilir.

Şifre anahtarı boyutuyla , tur sayısı ve Genişletilmiş Anahtar boyutu arasında bir ilişki vardır. 128 bitlik anahtar için ilk olarak bir kere Tur Anahtarı Eklemesi işlemi ve sonrasında 10 tur yapılır. Her turda yeni 16 byte'lık anahtara ihtiyaç duyulur bu nedenle 176 byte'a eşit olan 10+1 adet 16 byte'lık tur anahtarı kullanılır bu da 176 byte eder. Aynı mantık diğer iki şifre anahtarı boyutlarına da uygulanabilir. Genel formul denklem 3.2.'deki gibidir.

GenişletilmişAnahtarBoyutu = (TurSayısı+1)*AnahtarBoyutu (3.2.)

AES algoritmasın tüm işlemleri incelendikten sonra, bütün işlemleri bir arada gösteren şekil 3.2. verilmiştir.



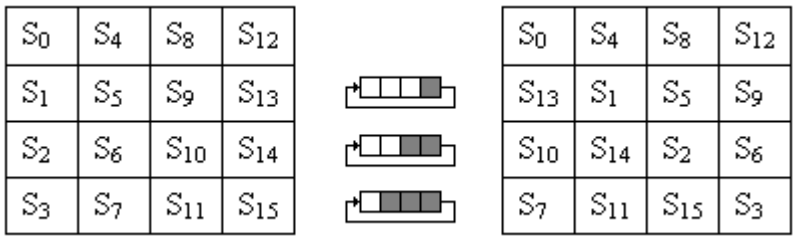
Şekil 3.2. Tüm Aes Algoritması [4]

3.9.AES'in Şifre Çözümü

Şifreleme dönüşümlerini tersine çevrilip, ters sırayla gerçeklenirse AES algoritmasının Ters Şifrelemesi elde edilebilir. Ters Şifrelemede Ters Sütun Kaydırma, Ters Bayt Yer Değiştirme, Ters Sütun Karıştırma ve Ters Tur Anahtarı Ekleme dönüşümleri kullanılır.

3.9.1.Ters Sütun Kaydırma İşlemi

Ters sütun kaydırma işlemi sütun kaydırma işleminin tersidir. Durum matrisinin son üç satırı dairesel olarak sağa kaydırılır. İlk satır, r=0 kaydırılmaz. İkinci satır bir kere, üçüncü satır iki kere, dördüncü satır üç kere sağa doğru bir eleman kaydırılır. Şekil 3.4.'te ters sütun kaydırma işlemi gösterilmiştir.



Şekil 3.4. Ters Sütun Kaydırma İşlemi [5]

3.9.2.Ters Bayt Yer değiştirme İşlemi

Bu işlem afin dönüşümün tersini uygular ve GF(2^8)'de tersini alır. Ters BaytYer değiştirme işlemi bayt yer değiştirme işleminin tersidir ve durum matrisinin her baytına S-Matrisinin tersini uygular. S-Matrisi şekil 3.5.'te gösterilmiştir.

		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 3.5. Ters S-Matrisi

3.9.3. TersSütunKarıştırma

TersSütunKarıştır işlemi SütunKarıştırma() işleminin tersidir. Durumun sütunları üstünde işlem görür. Her sütun GF(2^8) alanında dört terimli bir polinom olarak düşünülür ve modülü x^4 olan sabit polinom a^-1(x) ile çarpılır.

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\} \quad (3.2.)$$

$$s'(x) = a^{-1}(x) \cdot s(x) \quad (3.3.)$$

$$\begin{vmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{vmatrix} = \begin{vmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{vmatrix} \begin{vmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{vmatrix}$$

Dönüşümün sonucu olarak sütundaki dört bayt aşağıdakilerle değiştirilir:

$$s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c}) \quad (3.4.)$$

$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c}) \quad (3.5)$$

$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c}) \quad (3.6)$$

$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c}) \quad (3.7)$$

4. MİKROİŞLEMCİ VE SİSTEM

4.1. Seri Haberleşme

Seri port üzerinde paralel portta olduğu gibi bir çok uygulama vardır. İşlemci bile aldığı veriyi paralel olarak işlediğinden bir çok uygulama paralel iletişim ile daha kolay gerçekleştirilebilir. Aslında seri ve paralel veri iletimi arasında çok ufak bir fark bulunmaktadır. Paralel portta, bitler yani lojik 1 yada 0 değerleri 8 tane ayrı kablo ile aynı anda iletilir. Seri portta ise bu lojik değerler tek bir kablo aracılığı ile sırayla iletilir. Seri port iletişiminin kullanımı ve programlaması paralel porta göre daha zordur fakat bir çok avantajı da bulunmaktadır:

4.1.1. Seri portun Avantajları

1. Seri kablolar, paralel kablolara göre daha uzun olur. Seri port, lojik değerleri -3 volt ile +25 volt arasında iletebilir. Paralel portta ise "0", 0 volt ile, "1" ise +5 Volt ile iletilir. Dolayısı ile, seri portun 50V maksimum voltaj değişim aralığına sahiptir. Paralel portta ise bu aralık 5 volttur. Bu nedenle kabloda oluşan kayıp, seri portlarda, paralel portlardaki gibi önemli değildir.

2. Seri iletişimde, paralel porta göre çok daha az tel kullanılır. Cihaz ile bilgisayar arasındaki 3 telli kablo seri iletişim için yeterlidir. 3 telli bir kablo, 25 telli bir kabloya göre daha ucuz olacaktır.

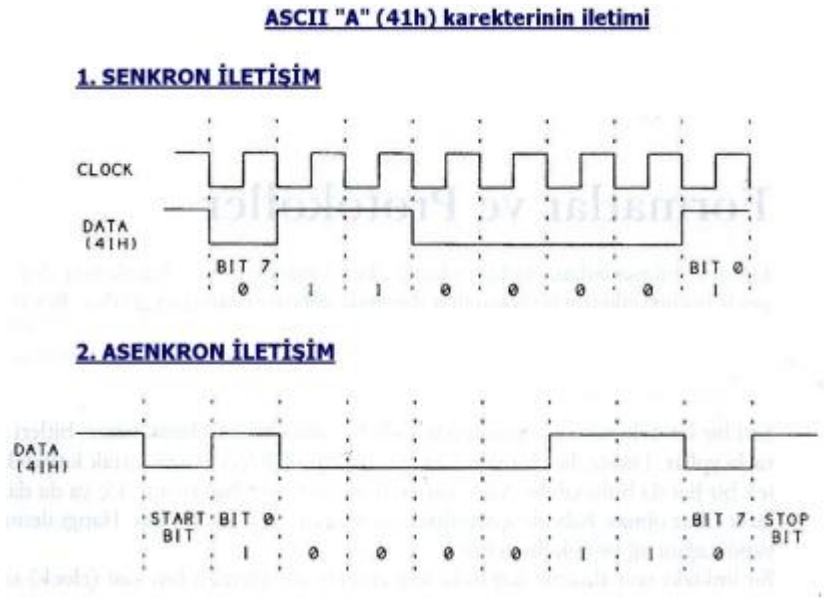
3. Seri haberleşmeyi kullanan kızıl ötesi cihazlar veriyi ancak seri olarak iletebilirler. Böyle bir haberleşmenin paralel olarak gerçekleştirilemez.

Seri haberleşmede, gönderici kısmında 8-bit veri, paralelden seriye çevrilir ve daha sonra tek bir hattan karşıya gönderilir. Alıcı, seri veriyi paralele çevirerek 8 bit veriyi oluşturur.

4.1.2. Seri Port İle Veri Aktarımı

Veri akışının kontrolü için, gerekli sinyallerden biri saat (clock) sinyalidir. Hem gönderici, hem de alıcı cihazda, bir bitin ne zaman gönderileceğine veya alınacağına karar verilirken bir saat sinyali kullanılır. Veri gönderen ve alan uçların belli kurallar çerçevesinde haberleşmesi gerekir. Verinin nasıl paketleneceği, bir karekterdeki bit sayısını, verinin ne zaman başlayıp biteceği gibi bilgileri bu kurallar belirler. Bu kurallar çerçevesine, Protokol adı verilir.

Eğer veri sadece bir yönde aktarılıyor ise, half duplex, aynı anda her iki yönde aktarılıyorsa, full duplex olarak adlandırılır. İki çeşit seri iletim formatı vardır. Senkron ve Asenkron. Herbiri saatleri farklı şekilde kullanırlar.



Şekil 4.1. ASCII “A”(41h) Karakterinin İletimi

Senkron gönderimde, her cihaz, kendisi yada dışarıdan bir cihaz tarafından üretilen aynı saat sinyali darbelerini kullanırlar. Saatin frekansı sabit yada düzensiz aralıklarda değişkende olabilir. Şekil 4.1’de gösterildiği gibi iletilen her bit, bir saat darbesi geçişinden, yani şekildeki yükselen veya alçalan kenardan sonraki belirli bir zamanda geçerli olur. Senkron formatlar, iletimi başlatırken yada bitirirken, çok çeşitli formatlar kullanırlar. Bunlara start-stop bitleri denir. Fakat uzun mesafeli linklerde senkron format uygun değildir. Saat sinyalinin iletimi, parazit nedeni ile, ek bir hat gerektirebilir. Bu durumda, Asenkron gönderim kullanılır.

Asenkron iletişimlerde, linkte saat hattı bulunmaz. Her uç kendi sinyalini sunmaktadır. Bu iletişimde de, uçların saat frekansında anlaşmaları gerekir. Bu nedenle iletilen her byte 'ta saatleri eşlemek üzere bir start biti ve iletimin bittiğini bildirmek üzere bir stop biti bulunur. Seri iletişimde veri aktarım hızı, saniyedeki bit sayısı olarak belirtilir.

4.2. C8051F060 Mikroişlemcisinin Haberleşme Birimleri

Bitirme projesinde Silicon Laboratories firması tarafından üretilen C8051F060DK isimli kiti kullanılmıştır. Kitte 25 Mhz sistem saati olan 8051 mikrokontrolör, 4 kB'lık iç oku-yaz belleği, 64 kb'lık dış oku-yaz belleği ve yazılan programı saklayan 64 kB'lık Flash belleği bulunmaktadır. C8051 işlemcisi saniyede 25 milyon komutu çalıştırabilmektedir. Cihazın 2 adet UART bağlantısı ve 5 tane 16 bitlik zamanlayıcısı bulunmaktadır. UART bağlantılarından biri haberleşmek için RS-232 protokolünü diğeri ise CAN protokolünü kullanmaktadır [6]. Çalışmada bilgisayarla haberleşmek için RS-232 protokolü ve UART0 haberleşme modu kullanılmıştır.

4.2.1. UART0

UART0 çerçeve hatası algılama ve adres tanıma donanımı olan gelişmiş bir seri porttur. UART0 tam-dupleks asenkron veya yarım-dupleks senkron moda çalışabilir. Birden çok mikroişlemciyle haberleşmeyi destekler. Alınan veri tutma registerında tamponlanır, bu sayede UART0 yazılım önceki veriyi okumayı bitirmeden ikinci gelen veriyi almaya başlar. Yeni alınan veri alış tamponuna önceki alınan baytın okuması bitmeden yazılırsa aşma biti bunu gösterir.

UART0'a SFR, Seri Kontrol(SCON0) ve Seri Veri Tamponu(SBUF0) aracılığıyla erişilir. Tek SBUF0 yeri hem gönderme ve alma registerlarına erişim sağlar. SCON0'ı okumak alma registerına ve SCON0'a yazmak gönderme registerına erişim sağlar.

UART0 polled veya kesme modunda çalışır. UART0'ın iki kesme kaynağını vardır: Gönderme Kesme bayrağı TI0 (SCON0.1) ve Alma Kesme bayrağı RI0 (SCON0.0). TI0 bayt verilerinin gönderimi tamamlandığında 1 olur. RI0 ise bayt verilerinin alımı tamamlandığında 1 olur. UART0 kesme bayrakları donanım tarafından sıfırlanmaz, yazılım tarafından sıfırlanması gerekmektedir. Bu yazılımın UART0'ın göndermeyi veya almayı bitirdiğini saptamasına yarar.

4.2.1.1. UART0'ın Çalışma Modları

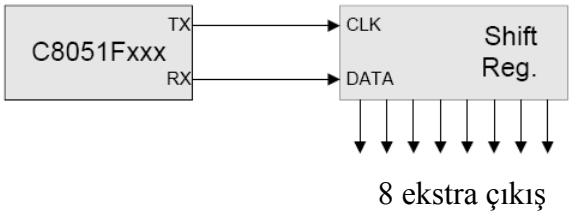
UART0 dört adet çalışma modu(bir senkron ve üç asenkron) sunar. Çalışma modları SCON0 registerındaki ayarlama bitleriyle seçilir. Bu dört mod farklı baud değerleri ve haberleşme protokolleri sunar. Çalışma modları tablo 4.1'de gösterilmişir.

Tablo 4.1. UART0'ın Çalışma Modları

MOD	SENKRONİZASYON	Baud Saati	Veri Bitleri	Başma/Bitirme Bitleri
0	Senkron	SİSTEMSAATİ/12	8	Yok
1	Asenkron	Zamanlayıcı 1, 2, 3, 4 veya Taşma	8	1 Başlama, 1 Bitirme
2	Asenkron	SİSTEMSAATİ/32 veya SİSTEMSAATİ/64	9	1 Başlama, 1 Bitirme
3	Asenkron	Zamanlayıcı 1, 2, 3, 4 veya Taşma	9	1 Başlama, 1 Bitirme

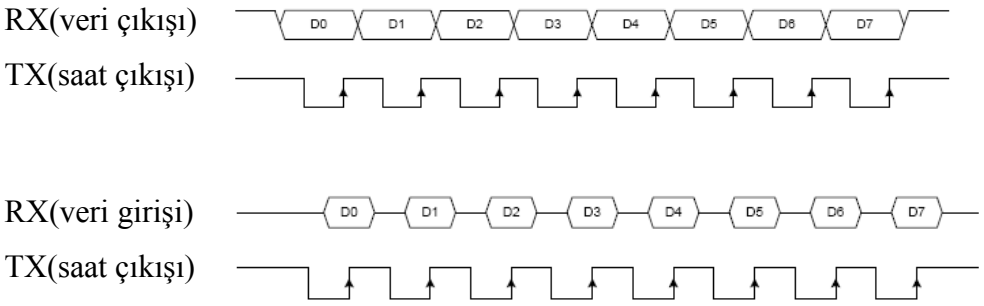
MOD 0: Senkron Mod

Mod 0 senkron, yarım – dupleks haberleşme sağlar. Seri veri RX0 pininden alınır ve gönderilir. TX0 pininin gönderim ve alım için kaydırma saati vardır. MCU iki yöndede kaydırma saati yarattığından öncelikli olarak seçilmelidir. MOD0 bağlantısı şekil 4.2'deki gibidir:



Şekil 4.2 MOD0 Bağlantısı

Veri haberleşmesi bir talimatın SBUF0 registarına bayt verisini yazmasıyla başlar. Sekiz veri biti öncelikle LSB'ye aktarılır(şekil 4.3.) ve işlem bittikten sonra TI0 gönderme kesme bayrağı (SCON0.1) 1 olur. Veri alımı ise REN0 Alma Etkinleştirme biti (SCON0.4) 1 ve RI0 alma kesme bayrağı 0 olduğu zaman başlar. Sekiz bit alımından sonra RI0 bayrağı 1 olur ve yazılım RI0 bitini 0 yapana kadar bit alımı olmaz.



Şekil 4.3. UART0 MOD0 Zamanlama Diyagramı

4.3. Bilgisayarı ve C8051F060'ı Uart Haberleşmesine Hazırlama

İki cihaz için UART haberleşmesi kullanılmak istendiği zaman ikisi de aynı baud hızına, 8 – bit veya 9-bit veri moduna sahip olmalıdır ve çift veya çift olmamalıdır. 8-bit veri, çift olmayan ve baud hızı 115200 bit/sn olarak kullanılacaktır. Terminal programı kullanılırsa aşağıdaki gibi ayarlanmalıdır.

Tablo 4.2. : Terminal Programı Ayarları

Saniyedeki bit sayısı	115200
Veri bitleri	8
Çiftlik	Yok
Dur Bitleri	1
Akış Kontrolu	Yok

Zamanlayıcı 1 gereken baud rate'i yaratmak için kullanılır. Mikroişlemcinin UART haberleşmesini yapabilmesi için aşağıdaki registerların ayarlanması gerekmektedir.

- TMOD
- SCON
- TH1
- TL1
- TCON

4.3.1. C8051 Mikroişlemcisi Özel Registerları

TMOD (zamanlayıcı modu registeri, adres 89h)

TMOD registeri standart iki zamanlayıcının hangi modda çalışacağını kontrol etmek için kullanılır. Zamanlayıcılar bu register ile 16-bit zamanlayıcı, 8-bit tekrar yüklenen zamanlayıcı veya 13-bit zamanlayıcı olarak programlanabilir. Aynı zamanda zamanlayıcılar sayıcı olarak ta programlanabilmektedirler. Bu sayede harici bir sinyalin her değişiminde zamanlayıcı değeri 1 artar. Tablo 4.3.'te TMOD registerının bitleri gösterilmiştir.

Tablo 4.3. TMOD Bitleri

7	6	5	4	3	2	1	0
Geçit	C/T	M1	M0	Geçit	C/T	M1	M0

Bit7: Geçit VEYA kapısı aktifleme bitidir. Zamanlayıcı 1 çalışmaya başlamak için bu bitin değerinin 0 olmasına gerek duyar.

Bit6: C/T Sayıcı veya zamanlayıcı seçme bitidir. Bu bit 1 olduğu zaman zamanlayıcı/sayıcı 1 sayıcı modunda çalışmaya başlar ve T0 pinine bağlı sinyal sayılır.

Bit 5: M1 Zamanlayıcı/sayıcı 1 mod seçme biti

Bit 4: M0 Zamanlayıcı/sayıcı 1 mod seçme biti.

Bit3: Geçit VEYA kapısı aktifleme bitidir. Zamanlayıcı 1 çalışmaya başlamak için bu bitin değerinin 0 olmasına gerek duyar.

Bit 2: C/T Sayıcı veya zamanlayıcı seçme bitidir. Bu bit 1 olduğu zaman zamanlayıcı/sayıcı 1 sayıcı modunda çalışmaya başlar ve T0 pinine bağlı sinyal sayılır.

Bit 1 M1 Zamanlayıcı/sayıcı 0 mod seçme bitidir.

Bit 0 M0 Zamanlayıcı/sayıcı 0 mod seçme bitidir.

M1	**M0**	**MOD**
0	**0**	**0**
0	**1**	**1**
1	**0**	**2**
1	**1**	**3**

SCON (seri kontrol, adres 98h, bit adreslenebilir)

SCON registeri 8051'in seri giriş ve çıkış biriminin davranışını kontrol etmek için kullanılır. Bu register kullanılarak seri haberleşme hızı belirlenebilir. Seri olarak bir karakter başarıyla yollandığında aktiflenen bayrakları bulunmaktadır. SCON bitleri tablo 4.4.'te gösterilmiştir. Tablo 4.5.'te ise SM0 ve SM1 bitleri ve anlamları verilmiştir.

Tablo 4.4. SCON Bitleri

Bit	**İsim**	**Bit Adresi**	**Açıklama**
7	SM0	9Fh	Seri port mod bit 0
6	SM1	9Eh	Seri port mod bit 1.
5	SM2	9Dh	Birden çok işlemci ile haberleşme modu
4	REN	9Ch	Alıcı aktif biti. Karakter okuması için bu bit aktif yapılmalıdır.
3	TB8	9Bh	Yolla biti, bit 8. Mod 2 ve 3 de karakter yollamak için kullanılan bit.
2	RB8	9Ah	Mod 2 ve 3 de karakter okumak için kullanılan bit.
1	TI	99h	Yolla bayrağı. Bir karakter yollandığı zaman bu bit aktif olur.
0	RI	98h	Alıcı bayrağı. Bir karakter okunduğu zaman bu bit aktif olur.

Tablo 4.5. SM0 ve SM1 bitleri ve anlamları.

SM0	**SM1**	**Seri Mod**	**Açıklama**	**Haberleşme hızı**
0	0	0	8-bit kaydırma Registeri	Osilatör frekansı / 12
0	1	1	8-bit UART	Zamanlayıcı 1 ile ayarlanır.(*)
1	0	2	9-bit UART	Osilatör frekansı / 32
1	1	3	9-bit UART	Zamanlayıcı 1 ile ayarlanır. (*)

(*) Zamanlayıcı 1 kullanılarak haberleşme hızı seçildiği durumda, eğer PCON.7 biti aktiflenirse haberleşme hızı iki katına yükselir.

SCON registerının bitleri adreslenebilirdir. Registerın bitleri isimleriyle kullanılabilmektedirler. SCON registerının yüksek dört biti(7.bit-4.bit) ayarlama bitleridir.

SM0 ve SM1 bitleri: Seri haberleşme modunu seçmek için kullanılır. 4 farklı haberleşme modu bulunur. Mode 0 ve Mod 2'de haberleşme hızı kristal frekansıyla orantılıdır ve sabittir. Mod 1 ve 3 te ise haberleşme hızı zamanlayıcı 1'in birim zamanda taşma sayısına bağlı olarak değişir.

SM2 biti: Birden fazla işlemciyle haberleşmek için kullanılır. Seri olarak bir karakter okunduğunda RI(alındı kesmesi) bayrağı aktiflenir. Bu sayede program karakterin alındığını ve işlenmeye hazır olduğunu bilebilir. Ancak SM2 biti aktiflenirse RI bayrağı okunan 9. bitten sonra aktiflenir. Gelişmiş seri haberleşme için kullanılan bir bittir.

REN(alıcı aktifleme) biti: Seri porttan okuma yapılmak isteniyorsa aktiflenir.

SCON registerının son 4 biti seri haberleşmede kullanılan işlem bitleridir. Yazma ve okuma işlemlerini gerçekleştirmek için kullanılırlar.

TB8 biti: Mod 2 ve mod 3'te kullanılmaktadır. Mod 2 ve mod 3'te 9 bitlik veri gönderilip okunmaktadır. İlk 8 bit karakter değeridir, 9. Bit ise TB8'den okunarak gönderilir.

RB8 biti : Mod 2 ve mod 3 de kullanılır. Mod 2 ve mod 3 de okuma yapıldığında toplam 9 bit data okunmaktadır. Okunan ilk 8 bit SBUF registerine kaydedilir. 9'uncu bit ise RB8 bitine kaydedilir.

TI (transmit interrupt) yollandı kesmesi biti : Program seri porttan bir bilgi yolladığında, değerin seri porttan tamamen yollanması için belli bir zaman vardır. Eğer bu karakterin yollanması bitmeden, yeni bir karakter yollanmak üzere SBUF registerine yazılırsa, veriler birbirine karışır. Bunu engellemek için yollama tamanlandı (TI) biti kullanılır. TI biti "1" ise bir önceki karakterin yollandığı anlaşılır ve yeni bir karakterin yollanmasında bir sakınca yoktur.

RI (receive interrupt) alma kesmesi biti: Bu bit de TI bitine benzer bir görev görmektedir. Ancak bu sefer dışardan bir karakter okunduğunda, okumanın bittiğini bildirmek için bu bit "1"değerini almaktadır.

TCON (Zamanlayıcı kontrol registeri, adres 88h, bit adreslenebilir)

TCON registeri 8051 entegresindeki iki adet zamanlayıcının ayarlanmasında kullanılmaktadır. Bu register ile zamanlayıcılar çalıştırılabilir veya durdurulabilirler. TCON registerının bir biti zamanlaıcının taşma biti olarak kullanılmaktadır. Bu sayede her zamanlayıcı veya sayıcı taşmasında bu bit aktiflenir. Bu registerın bazı bitleri ise zamanlayıcı ve sayıcının kesme üretmesi için kullanılır. TCON registeri bit adreslenebilirdir. Bu registerin bitleri zamanlayıcı/sayıcıları kontrol etmek için kullanılmaktadır. Tablo 4.6.'da TCON bitleri verilmiştir.

Tablo 4.6. TCON Bitleri

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit 7 TF1 Zamanlayıcı 1 taşma bayrağı: Zamanlayıcı taştığı zaman bu bayrak aktiflenir. Mikroişlemci ilgili kesme programına sıçradığında bu bayrak tekrar temizlenir. Eğer kesme programı yoksa bu bayrak program tarafından temizlenmelidir.

Bit 6 TR1 Zamanlayıcı 1 çalışma kontrol biti: Zamanlayıcı 1 çalışmaya başlatılmak istendiğinde bu bayrak aktiflenir. Bu bayrak aktif olduğu sürece zamanlayıcı 1 çalışmaktadır.

Bit 5 TF0 Zamanlayıcı 0 taşma bayrağı: Zamanlayıcı taştığı zaman bu bayrak aktiflenir. Mikroişlemci ilgili kesme programına sıçradığında ise bu bayrak tekrar temizlenir. Eğer kesme programı yoksa bu bayrak program tarafından temizlenmelidir.

Bit 4 TR0 Zamanlayıcı 0 çalışma kontrol biti: Zamanlayıcı 0 çalışmaya başlatılmak istendiğinde bu bayrak aktiflenir. Bu bayrak aktif olduğu sürece zamanlayıcı 0 çalışmaktadır.

Bit 3 IE1 Harici kesme 1 kenar bayrağı: INT1 pininde yüksekten alçağa düşen bir sinyal görüldüğünde, program INT1 kesme adresi 0013h'e sıçrar.

Bit 2 IT1 Harici kesme 1 INT1 tip belirleme biti: Eğer sinyal yüksekten düşüğe geçtiğinde kesme aktiflenmesi isteniyorsa bu bit SET edilir. Bu bit 0 olduğunda pindeki bir 0 sinyali kesmeyi aktifler.

Bit 1 IE0 Harici kesme 0 kenar bayrağı: INT0 pininde yüksekten alçağa düşen bir sinyal görüldüğünde, program INT0 kesme adresi 0003h'e sıçrar.

Bit 0 IT0 Harici kesme 0 INT0 tip belirleme biti: Eğer sinyal yüksekten düşüğe geçtiğinde kesme aktiflenmesi isteniyorsa bu bit SET edilir. Bu bit 0 olduğunda pindeki bir 0 sinyali kesmeyi aktifler.

TL1/TH1 (Zamanlayıcı 1 düşük ve yüksek, adres 8Ch ve 8Dh)

Bu registerlar zamanlayıcı 1'i temsil ederler. Zamanlayıcı/sayıcı 1'in sayma değerleri bu registerlerde tutulur. Herhangi bir anda sayma değeri okunmak istendiğinde, TL1 ve TH1 sayıcılarının değerleri değişkenlere atanır.