**AES ŞİFRESİ**

**Ters İkame Bayt Dönüşümü**

InvSubBytes adı verilen ters ikame bayt dönüşümü, Tablo 5.4b'de gösterilen S-box'ın tersini kullanır. Örneğin, {2a} girişinin {95} çıkışını ürettiğine ve S-kutusuna giden {95} girişinin {2a} ürettiğine dikkat edin. Ters S-kutusu, (5.1)'deki dönüşümün tersi uygulanarak ve ardından GF(2 8 )'de çarpımsal ters alınarak oluşturulur.

ters dönüşüm



burada d={05} veya 0000 0101. Bu dönüşümü şu şekilde gösterebiliriz:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B 0 ' |  | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |  | B 0 |  | 1 |  |
| B 1 ' |  | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |  | B1 \_ |  | 0 |  |
| 2 ' \_ |  | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |  | B2 \_ |  | 1 |  |
| 3 ' \_ | = | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | X | B3 \_ | + | 0 |  |
| 4 ' \_ |  | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |  | B4 \_ |  | 0 |  |
| B 5 ' |  | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |  | B5 \_ |  | 0 |  |
| B 6 ' |  | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |  | B6 \_ |  | 0 |  |
| B 7 ' |  | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |  | B7 \_ |  | 0 |  |

InvSubBytes'in SubBytes'in tersi olduğunu görmek için, SubBytes ve InvSubBytes içindeki matrisleri sırasıyla X ve Y olarak ve c ve d sabitlerinin vektör sürümlerini sırasıyla C ve D olarak etiketleyin. 8 bitlik bir B vektörü için denklem (5.2) şöyle olur:

 (\*)

varsayalım ki

: YX=E

burada E birlik matrisidir.

(\*)'nın her iki kısmını da Y ile çarparsak,



Y'nin X'in tersi olduğunu kısmen kontrol edelim: Çarpımın köşegen elemanları 1, diğer elemanlar – 0 olmalıdır. Örneğin 2. köşegen elemanı hesaplayalım ( Y'nin 2. sırasını X'in 2. sütunuyla çarp, numaralandırmaya 0'dan başla) :

0\*0+1\*0+0\*1+0\*1+1\*1+0\*1+0\*1+1\*0=1

Y'nin 2. satırı ile X'in 1. sütununu çarparsak ,

**Ters İkame Bayt Dönüşümü (Devamı 1)**

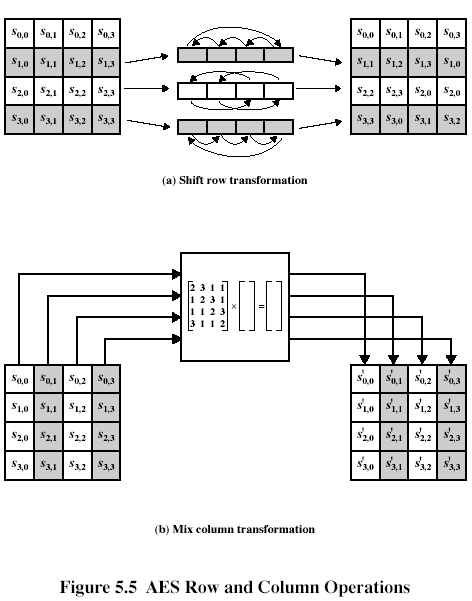
0\*0+1\*1+0\*1+0\*1+1\*1+0\*1+0\*0+1\*0=1+1=0.

Yani, köşegen üzerinde 1 ve köşegenin dışında 0 var.

S-box, bilinen kriptanalitik saldırılara karşı dayanıklı olacak şekilde tasarlanmıştır. Giriş bitleri ile çıkış bitleri arasında düşük korelasyon sağlar, çıkış girişin basit matematiksel fonksiyonu olarak ifade edilemez, sabit noktaları yoktur (S-box(a)=a).

**Satır Dönüşümünü Kaydır**

ShiftRows adı verilen ileri kaydırma satırı dönüşümü, Şekil 5.5a'da gösterilmektedir.



**Satır Dönüşümünü Kaydırma (Devam 1)**

1. satır (0 sayısı) değiştirilmez, satır numarası i, i-byte dairesel sola kaydırma ile sola kaydırılır, i=1, 2, 3. Aşağıda bu tür kaydırma örneği verilmiştir :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 87 | F2 | 4D | 97 |  | 87 | F2 | 4D | 97 |
| AK | 6E | 4C | 90 | => | 6E | 4C | 90 | AT |
| 4A | C3 | 46 | E7 |  | 46 | E7 | 4A | C3 |
| 8C | D8 | 95 | A6 |  | A6 | 8C | D8 | 95 |

InvShiftRows adı verilen ters kaydırmalı satır dönüşümü, i'inci satırın sağa dairesel kaydırmasını i bayt, i=0,1,2,3 gerçekleştirir.

Satır kaydırma dönüşümü, bir sütunun 4 baytının dört farklı sütuna yayılmasını sağlar (Şekil 5.3 bu etkiyi göstermektedir).

**Karışık Sütun Dönüşümü**

MixColumns adı verilen forward mix sütun dönüşümü, her sütunda ayrı ayrı çalışır. Her bayt, sütundaki dört baytın hepsinin bir işlevi olan yeni bir değere eşlenir. Dönüşüm, Durum üzerinde aşağıdaki matris çarpımı olarak tanımlanabilir (Şekil 5.5b):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 02 | 03 | 01 | 01 |  | S00 | S01 | S02 | S03 |  | S00' | S01' | S02' | S03' |  |
| 01 | 02 | 03 | 01 | \* | S10 | Ö11 | Ö12 | Ö13 | = | S10' | S11' | S12' | S13' | (5.3) |
| 01 | 01 | 02 | 03 |  | S20 | S21 | S22 | S23 |  | S20' | S21' | S22' | S23' |  |
| 03 | 01 | 01 | 02 |  | S30 | S31 | S32 | S33 |  | S30' | S31' | S32' | S33' |  |

Çarpım matrisindeki her eleman, bir satır ve bir sütundaki elemanların çarpımlarının toplamıdır. Bu durumda GF(2 8 ) de çarpma ve toplama işlemleri yapılır .

Aşağıda MixColumns örneği verilmiştir;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 87 | F2 | 4D | 97 |  | 47 | 40 | A3 | 4C |
| 6E | 4C | 90 | AK | => | 37 | D4 | 70 | 9F |
| 46 | E7 | 4A | C3 |  | 94 | E 4 | 3 A | 42 |
| A6 | 8C | D8 | 95 |  | ED | A5 | A6 | M.Ö |

Sonucun 1. sütunu şu şekilde elde edilir:

{02){87}+{03}{6E}+{46}+{A6} ={47}

{87}+{02}{6E}+{03}{46}+{A6} ={37}

{87}+{6E}+{02}{46}+{03}{A6} ={94}

{03}{87}+{6E}+{46}+{02}{A6} ={ED}

1. denklem için {02}{87}=(0000 0010)(1000 0111)=

=

(0001 0101)={15}

**Karma Sütun Dönüşümü (Devam 1)**

{03}{6E}=(0000 0011)(0110 1110)= =

(1011 0010) = {B2}

{02){87}+{03}{6E}+{46}+{A6}={15}+{B2}+{46}+{A6}=

(0001 0101)+

(1011 0010)+

(0100 0110)+

(1010 0110)=

(0100 0111)={47}

InvMixColumns adı verilen ters karışım sütunu dönüşümü, aşağıdaki matris çarpımı ile tanımlanır:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0E | 0B | 0D | 09 |  | S00 | S01 | S02 | S03 |  | S00' | S01' | S02' | S03' |  |
| 09 | 0E | 0B | 0D | \* | S10 | Ö11 | Ö12 | Ö13 | = | S10' | S11' | S12' | S13' | (5.5) |
| 0D | 09 | 0E | 0B |  | S20 | S21 | S22 | S23 |  | S20' | S21' | S22' | S23' |  |
| 0B | 0D | 09 | 0E |  | S30 | S31 | S32 | S33 |  | S30' | S31' | S32' | S33' |  |

8 )'deki çarpımının bir birim matris olduğunu kontrol edeceğiz . S00' ( 0. satır ile 0. sütunun çarpımı ) için böyle bir kısmi kontrol yapalım :

S00'=(0E}{02}+{0B}{01}+{0D}{01}+{09}{03}=(0E}{02}+{0B}+{0D}+{09}{ 03}

{0E}{02}=(0000 1110)(0000 0010)= =(0001 1100)={1C}

{09}{03}=(0000 1001)(0000 0011)= =(0001 1011)={1B}

(0E}{02}+{0B}+{0D}+{09}{03}={1C}+{0B}+{0D}+{1B}=

(0001 1100)+

(0000 1011)+

(0000 1101)+

(0001 1011)=

(0000 0001)={01}

Diğer unsurlar da benzer şekilde doğrulanır.

AES belgesi, MixColumns'u polinom aritmetiği açısından açıklar. Standartta, MixColumns, State'in her sütununun GF(2 8 )'deki katsayılara sahip dört terimli bir polinom olduğu düşünülerek tanımlanır. Her sütun, modulo tarafından verilen sabit polinom a (x) ile çarpılır.

**Karışık Sütun Dönüşümü (Devamı 2)**

 (5.7)

Polinom (5.7) ile bu tür bir çarpmanın (5.3) ile temsil edilen matris çarpımına eşdeğer olduğunu gösterelim. Durum matrisinin her bir sütunu, ilgili polinomun katsayıları kümesi olarak görüntülenir, örneğin, Durum'un 1. sütunu , polinomu karşılar:



Daha sonra

Son polinomda, katsayılar (5.3)'e göre matris çarpımında kullanılanlarla aynıdır. Aslında (5.3)'teki sonucun 1. sütunu aşağıdaki gibi yazılabilir:

S00'={02}S00+{03}S10+S20+S30

S10'=S00+{02)S10+{03}S20+S30

S20'=S00+S10+{02}S20+{03}S30

S30'={03}S00+S10+S20+{02}S30

Benzer şekilde, (5.5)'teki dönüşümün, her sütunu 4 terimli bir polinom olarak ele almaya ve her sütunu b(x) ile çarpmaya karşılık geldiği gösterilebilir.

 (5.8)

**Karma Sütun Dönüşümü (Devamı 3)**

gösterilebilir ki

Satır kaydırma dönüşümüyle birleştirilen karma sütun dönüşümü, birkaç turdan sonra tüm çıkış bitlerinin tüm giriş bitlerine bağlı olmasını sağlar.

**Yuvarlak Anahtar Dönüşümü Ekle**

AddRoundKey adı verilen ileriye dönük yuvarlak anahtar dönüşümünde, 128 bitlik Durum, yuvarlak anahtarın 128 biti ile bit düzeyinde XORlanır.

AddRoundKey dönüşümü olabildiğince basittir ve State'in her bitini etkiler. Yuvarlak anahtar genişletmenin karmaşıklığı, AES'nin diğer aşamalarının karmaşıklığıyla birlikte güvenliği sağlar.

**AES Anahtar Genişletme**

AES anahtar genişletme algoritması, girdi olarak 4 kelimelik (16 bayt) bir anahtar alır ve 44 kelimelik (156 bayt) doğrusal bir dizi üretir. Aşağıdaki sözde kod, genişletmeyi açıklar:

KeyExpansion(byte key[16], word w[44]){

Kelime sıcaklığı;

For(i=0;i<4;i++) w[i]=(tuş[4\*i], tuş[4\*i+1], tuş[4\*i+2], tuş[4\*i+3 ]);

(i=4;i<44;i++){ için

Sıcaklık=w[i-1];

If(I mod 4 = 0) temp = SubWord(RotWord(temp)) XOR Rcon[i/4];

W[i]=w[i-4] XOR sıcaklığı;

}

}

, genişletilmiş anahtarın 1. dört sözcüğüne kopyalanır . Genişletilmiş anahtarın geri kalanı her seferinde dört sözcükle doldurulur. Eklenen her w[i] sözcüğü, hemen öncesindeki w[i-1] sözcüğüne ve dört konum gerideki w[i-4] sözcüğüne bağlıdır. Dört vakanın üçünde basit bir XOR kullanılır. w dizisindeki konumu 4'ün katı olan bir sözcük için daha karmaşık bir işlev kullanılır. Şekil 5.6 , karmaşık işlevi temsil etmek için g sembolü kullanılarak genişletilmiş anahtarın 1. sekiz kelimesinin oluşturulmasını göstermektedir . g işlevi aşağıdaki alt işlevlerden oluşur:

1. RotWord, bir kelime üzerinde 1 baytlık dairesel bir sola kaydırma gerçekleştirir. Bu, bir giriş kelimesinin [b0, b1, b2, b3] [b1, b2, b3, b0]'a dönüştürüldüğü anlamına gelir.
2. Alt Kelime, S-kutusunu kullanarak giriş kelimesinin her bir baytında bir bayt ikamesi gerçekleştirir (Tablo 5.4a)
3. 1. ve 2. adımların sonucu, yuvarlak bir sabit olan Rcon[j] ile XORlanır.

**AES Anahtar Genişletme (Devamı 1)**

Yuvarlama sabiti, en sağdaki üç baytın her zaman 0 olduğu bir sözcüktür. Bu nedenle, Rcon'lu bir sözcüğün XOR'unun etkisi, yalnızca sözcüğün en soldaki baytında bir XOR gerçekleştirmektir. yuvarlak sabit

her tur için farklıdır ve Rcon[j]=(RC[j],0,0,0), RC[1]=1, RC[j]=2 RC[j-1] ve çarpma tanımlı olarak tanımlanır GF(2 8 ) alanı üzerinde . Onaltılık olarak RC[j] değerleri şöyledir:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| J | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| RC[j] | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 80 | 1b | 36 |

Örneğin, 8. tur için yuvarlak anahtarın olduğunu varsayalım.

EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F

Daha sonra 9. tur için yuvarlak anahtarın 1. dört baytı (1. sütun ) aşağıdaki gibi hesaplanır:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ben(ondalık) | sıcaklık | RotWord'dan sonra | Alt Kelimeden Sonra | Rkon(9) | XOR'dan sonra  Rcon ile | W[i-4] | W[i]=temp XOR w[i-4] |
| 36 | 7f8d292f | 8d292f7f | 5da515d2 | 1b000000 | 46a515d2 | Ead27321 | Ac7766f3 |

Yuvarlamaya bağlı bir sabitin dahil edilmesi, yuvarlak anahtarların farklı döngülerde üretilme biçimleri arasındaki simetriyi veya benzerliği ortadan kaldırır. Tersine çevrilebilir bir dönüşümdür. Her anahtar biti, birçok yuvarlak anahtar bitini etkiler. Doğrusal olmayan bir dönüşümdür.