BÖLÜM 4

RIJNDAEL

4.1 Matematiksel Özellikler

8 bitten oluşan bir byte 16'lık tabanda yazılabildiği gibi polinom olarak ifade ediliyor. İki byte'ı toplamak, çarpmak ve bir byte'ın tersini almak polinomlarca ifade edilecektir.Buna göre bir $b = (b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0)$ bytının *polinom gösterimi*:

$$p(a) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

Bir *byte'ın değeri* 10'luk tabanda kendisine karşılık gelen sayıdır.

Örnek 4.1.1 $(53)_{16}$ byte'ının değeri: $(53)_{16} = \text{`}53\text{`} = 3 + 5.16 = 83$
Bİt olarak ifadesi yani ikilik düzende: $83 = (0\,1\,0\,1\,0\,1\,1\,1)$
$p_a(x) = x^6 + x^4 + x^2 + x + 1$

4.1.1 Toplama İşlemi

Byte'ların polinomlarını modulo 2'de toplamaktır. Bu işlem aynı zamanda iki byte'ı XOR'lamaya da denktir.

Örnek 4.1.2

$$\begin{aligned} a = (1\,0\,0\,1\,1\,0\,1\,0) \Rightarrow p_a(x) &= x^7 + x^4 + x^3 + x \\ b = (1\,0\,1\,0\,1\,0\,1\,1) \Rightarrow p_b(x) &= x^7 + x^5 + x^3 + x + 1 \\ p_a(x) + p_b(x) &= x^7 + x^4 + x^3 + x + x^7 + x^5 + x^3 + x + 1 \bmod 2 \\ &= x^5 + x^4 + 1 \quad \Rightarrow \quad (0\,0\,1\,1\,0\,0\,0\,1) \\ a \oplus b &= (0\,0\,1\,1\,0\,0\,0\,1) \end{aligned}$$

4.1.2 Çarpma İşlemi

1. İki Byte'ı Çarpma:

   çarpma işleminde, iki byte polinom olarak ifade edilir. İki polinom çarpılır, çarpma işlemi mod $m(x) = x^8 + x^4 + x^3 + x + 1$ de yapılır. $m(x)$ modulo 2'de çarpanlarına ayrılamayan bir polinomdur. Modulo 2'de çarpanlarına ayrılamamak demek katsayıları 1 veya 0 olan polinomların çarpımı şeklinde yazılamamak demektir. Bir polinomun modulo $m(x)$'deki değeri polinomun $m(x)$'e bölümünden kalana denktir.

   Bir polinomun $m(x)$'e bölümünden kalanı bulmak için polinomda $x^8+x^4+x^3+x+1$ görülen yere 0 koymaktır. Bu aynı zamanda x^8 görülen yere x^4+x^3+x+1 koymakla aynıdır.

   Bir byte'ın polinom gösteriminde en fazla yedinci dereceden bir terim olacağından iki byte'ın çarpımının yine bir byte olabilmesi polinom ifadesinde derecesi sekiz ve sekizden büyük terimlerin yok edilmesi gerekiyor. Bu nedenle çarpma işlemi modulo

2'de çarpanlarına ayrılamamyan bir polinom olan mod $m(x)$ de yapılıyor

Not: Bu işlem için derecesi 8 olan ve modulo 2'de çarpanlarına ayrılamamyan başka bir polinom da seçilebilirdi.

Kısaca:

$$\begin{aligned} a &\rightarrow p_a(x) \\ b &\rightarrow p_b(x) \\ (a).(b) &= p_a(x).p_b(x) \bmod m(x) \end{aligned}$$

Örnek 4.1.3

$$\begin{aligned} a = (0\,1\,0\,1\,0\,1\,0\,1) \Rightarrow p_a(x) &= x^6 + x^4 + x^2 + 1 \\ b = (1\,0\,0\,0\,0\,0\,1\,1) \Rightarrow p_b(x) &= x^7 + x + 1 \end{aligned}$$

$$
\begin{aligned}
(a).(b) &= p_a(x).p_b(x) \mod m(x) = x^8 + x^4 + x^3 + x + 1 \\
&= (x^6 + x^4 + x^2 + 1)(x^7 + x + 1) \\
&= x^{13} + x^7 + x^6 + x^{11} + x^5 + x^4 + x^9 + x^3 + x^2 + x^7 + x + 1 \\
&= x^{13} + x^{11} + x^9 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \\
&= x^5x^8 + x^3x^8 + x + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \\
&= x^5(x^4 + x^3 + x + 1) + x^3(x^4 + x^3 + x + 1) + x(x^4 + x^3 + x + 1) \\
&\quad +x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \\
&= x^9 + x^8 + x^6 + x^5 + x^7 + x^6 + x^4 + x^3 + x^5 + x^4 + x^2 + x \\
&\quad +x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \\
&= x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + 1 \\
&= xx^8 + x^8 + x^7 + x^6 + x^5 + x^4 + 1 \\
&= x(x^4 + x^3 + x + 1) + x^4 + x^3 + x + 1 + x^7 + x^6 + x^5 + x^4 + 1 \\
&= x^5 + x^4 + x^2 + x + x^4 + x^3 + x + 1 + x^7 + x^6 + x^5 + x^4 + 1 \\
&= x^7 + x^6 + x^4 + x^3 + x^2 \\
\Rightarrow &\quad (a)(b) = (1\,1\,0\,1\,1\,1\,0\,0)
\end{aligned}
$$

2. 4 Byte'lık Vektörleri Çarpma:

   4 byle'lık bir vektör olan $\vec{a} = (a_3, a_2, a_1, a_0)$ polinom olarak ifade edilir.

$$p_{\vec{a}}(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

   Burada a_3, a_2, a_1, a_0'ın byte oldukları unutulmamalıdır.

$\vec{a}$ ve $\vec{b} = (b_3, b_2, b_1, b_0)$ vektörleri için:

$$\vec{a}.\vec{b} = p_{\vec{a}}(x).p_{\vec{b}}(x) \bmod M(x) = x^4 + 1$$

Aynı zamanda

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} . \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Not: 4 byte'lık bir $\vec{b} = (b_3, b_2, b_1, b_0)$vektörünü $\vec{a} = (\text{`00`}, \text{`00`}, \text{`01`}, \text{`00`}) \Rightarrow p_{\vec{a}}(x) = x$ ile çarpmak bir sola kaydırmaya denktir. Yani

$$(b_3 b_2 b_1 b_0).(0010) = (b_2 b_1 b_0 b_3)$$

3. $a = (a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0)$ byte'ının çarpmaya göre tersi

$$p_a(x).p_b(x) = 1 \mod m(x) = x^8 + x^4 + x^3 + x + 1$$

eşitliğini sağlayan $p_b(x)$ polinomuna karşılık gelen byte'tır.

$$p_a^{-1}(x) = p_b(x) \mod m(x) = x^8 + x^4 + x^3 + x + 1.$$

Buna göre $a = (0\,0\,0\,0\,0\,0\,0\,0)$ nın tersi kendisidir.

4. 4 byte'lık $\vec{a} = (a_3, a_2, a_1, a_0)$ vektörünün çarpmaya göre tersi

$$p_{\vec{a}}(x).p_{\vec{b}}(x) = 1 \mod M(x) = x^4 + 1$$

eşitliğini sağlayan $p_{\vec{b}}(x)$ polinomuna karşılık gelen 4 byte'lık $\vec{b}$ vektörüdür.

$$p_{\vec{a}}^{-1}(x) = p_{\vec{b}}(x) \mod M(x) = x^4 + 1.$$

4.2 Algoritma

Rijndael Algoritması

Metin uzunluğu: 128,192,256 bit olabilir.

Anahtar uzunluğu: 128,192,256 bit olabilir.

Döngü (*round*) sayısı: Anahtar uzunluğu ve metin uzunluğuna göre değişiklik göstermektedir. Aşağıdaki tabloda gösterilmiştir. Satırlar metin uzunluklarını, sütünlar anahtar uzunlukarını göstermektedir.

	128	192	256
128	10	12	14
192	12	12	14
256	14	14	14

Her döngüde üç ayrı bölüm vardır.

1. Doğrusal (linear) işlemlerin olduğu bölüm. Bu katmanlarda (*layer*) difüzyon sağlanmaktadır.
2. Doğrusal olmayan bölüm. S kutularından (S-*box*) oluşmaktadır.
3. Anahtarın XOR'landığı katman.

Bit olarak ifade edilen mesajı byte'lara ayrılır.

$$a_{00}\ a_{10}\ a_{20}\ a_{30}\ a_{01}\ a_{11}\ a_{21}\ a_{31}\ a_{02}\ a_{12}\ a_{22}\ a_{32}\ a_{03}\ a_{13}\ a_{23}\ a_{33}\ \ldots$$

Metin 4 byte'lık sütün vektörleri şeklinde, yani 128 bit için 4×4, 192 bit için 4×6, 256 bit için 4×8'lik matrislerle ifade edilir. 128 bitlik bir metin için aşağıdaki gibidir.

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

4.2.1 Byte Sub

S-kutusunun olduğu katmandır. Matristeki her byte'ın modulo $m(x) = x^8+x^4+x^3+x+1$'e göre tersi bulunur. $a \longrightarrow a^{-1} = b = (b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0)$ S-kutusunun çıktısı $y = (y_7\ y_6\ y_5\ y_4\ y_3\ y_2\ y_1\ y_0)$ olmak üzere:

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

Burdaki 8×8'lik matrisin özelliği modulo 2'de tersi olan bir matris olmasıdır.

4.2.2 Shift Row

128 bit için:

$$\begin{matrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{matrix} \longrightarrow \begin{matrix} 1 & 5 & 9 & 13 \\ 6 & 10 & 14 & 2 \\ 11 & 15 & 3 & 7 \\ 16 & 4 & 8 & 12 \end{matrix}$$

Bu permütasyona göre 6. pozisyondaki byte 2. pozisyona, 3. pozisyondaki byte 11. pozisyona geçmiş.

192 bit için:

$$\begin{matrix} 1 & 5 & 9 & 13 & 17 & 21 \\ 2 & 6 & 10 & 14 & 18 & 22 \\ 3 & 7 & 11 & 15 & 19 & 23 \\ 4 & 8 & 12 & 16 & 20 & 24 \end{matrix} \longrightarrow \begin{matrix} 1 & 5 & 9 & 13 & 17 & 21 \\ 6 & 10 & 14 & 18 & 22 & 2 \\ 11 & 15 & 19 & 23 & 3 & 7 \\ 16 & 20 & 24 & 4 & 8 & 12 \end{matrix}$$

256 bit için:

$$\begin{matrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 \end{matrix} \longrightarrow \begin{matrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 \\ 6 & 10 & 14 & 18 & 22 & 26 & 30 & 2 \\ 15 & 19 & 23 & 27 & 31 & 3 & 7 & 11 \\ 20 & 24 & 28 & 32 & 4 & 8 & 12 & 16 \end{matrix}$$

4.2.3 Mix Column

Byte Sub ve Shift Row işlemlerinden çıkan, her sütünü 4 bytle'lık vektör olan matris, bu katmanda modulo $M(x) = x^4 + 1$'de $c(x) = \text{`}03\text{`}x^3 + \text{`}01\text{`}x^2 + \text{`}01\text{`}x + \text{`}02\text{`}$ polinomuyla

çarpılır. Buna göre:

$$\begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} \text{`02`} & \text{`03`} & \text{`01`} & \text{`01`} \\ \text{`01`} & \text{`02`} & \text{`03`} & \text{`01`} \\ \text{`01`} & \text{`01`} & \text{`02`} & \text{`03`} \\ \text{`03`} & \text{`01`} & \text{`01`} & \text{`02`} \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

4.2.4 Anahtarla XOR'lama

Anahtarlar da aynı şekilde matrislerle ifade edilir, anahtarla metnin karşılıklı byte'ları XOR'lanır.

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \oplus \begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{bmatrix}$$

$$= \begin{bmatrix} a_{00} \oplus k_{00} & a_{01} \oplus k_{01} & a_{02} \oplus k_{02} & a_{03} \oplus k_{03} \\ a_{10} \oplus k_{10} & a_{11} \oplus k_{11} & a_{12} \oplus k_{12} & a_{13} \oplus k_{13} \\ a_{20} \oplus k_{20} & a_{21} \oplus k_{21} & a_{22} \oplus k_{22} & a_{23} \oplus k_{23} \\ a_{30} \oplus k_{30} & a_{31} \oplus k_{31} & a_{32} \oplus k_{32} & a_{33} \oplus k_{33} \end{bmatrix}$$

4.3 Anahtar Algoritması

Anahtar olarak üretilen toplam bit sayısı (blok uzunluğu) × (Döngü sayısı+1) kadardır. Eğer 4 bytle'lık vektörlerin her birine kelime (1 kelime=32 bit) dersek, kelime olarak üretilen anahtar sayısı (bloktaki kelime sayısı) × (döngü sayısı+1) dir.

$N_b = 1$ bloktaki kelime sayısı

$N_k =$ Anahtardaki kelime sayısı

$N_r =$ Döngü sayısı

Bu anahtar algoritmasından çıkan her kelimeye $\begin{bmatrix} w_{0i} \\ w_{1i} \\ w_{2i} \\ w_{3i} \end{bmatrix}$ dersek,

$0 \leq i < N_k$ için

$$\begin{bmatrix} w_{0i} \\ w_{1i} \\ w_{2i} \\ w_{3i} \end{bmatrix} = \begin{bmatrix} k_{0i} \\ k_{1i} \\ k_{2i} \\ k_{3i} \end{bmatrix}$$

$N_k \leq i < N_b(N_r + 1)$ için

$N_k \mid i$ ise

$$w_i = w_{i-N_k} \oplus Subbyte(RotByte(w_{i-1})) \oplus RoundConstant[i/N_k]$$

$N_k \nmid i$ ise

$$w_i = w_{i-N_k} \oplus w_{i-1}$$

Subbyte: Byte sub katmanındaki işlemlerden oluşur.

Rotbyte: Rotbyte((a,b,c,d))=(b,c,d,a).

Round Constant:

RC[1]='01'

RC[j]=x.RC[j-1]

$$RoundConstant[\text{j}] = \begin{bmatrix} \text{RC}[j] \\ \text{'00'} \\ \text{'00'} \\ \text{'00'} \end{bmatrix}$$

Anahtar algoritmasında ilk alınan anahtar 32 bitlik kelimeler halinde hiç değişikliğe uğramadan kullanılır. Eğer algoritmadan çıkan anahtar kelimeler (32 bit uzunluğunda 4 byte'lık vektörler) $w_0\, w_1\, w_2\, w_3\, w_4\, w_5 \ldots$ ise 128 bitlik blok uzunluğu için:

$$\begin{aligned} w_0\, w_1\, w_2\, w_3 &\rightarrow \textit{Round } 0 \text{ Round'a girmeden} \\ w_4\, w_5\, w_6\, w_7 &\rightarrow \textit{Round } 1 \\ w_8\, w_9\, w_{10}\, w_{11} &\rightarrow \textit{Round } 2 \\ &\vdots \end{aligned}$$

192 bitlik blok uzunluğu için:

$$\begin{aligned} w_0\, w_1\, w_2\, w_3\, w_4\, w_5 &\rightarrow \textit{Round}\ 0 \text{ Round'a girmeden} \\ w_6\, w_7\, w_8\, w_9\, w_{10}\, w_{11} &\rightarrow \textit{Round}\ 1 \\ w_{12}\, w_{13}\, w_{14}\, w_{15}\, w_{16}\, w_{17} &\rightarrow \textit{Round}\ 2 \\ &\vdots \end{aligned}$$

256 bitlik blok uzunluğu için:

$$\begin{aligned} w_0\, w_1\, w_2\, w_3\, w_4\, w_5\, w_6\, w_7 &\rightarrow \textit{Round}\ 0 \text{ Round'a girmeden} \\ w_8\, w_9\, w_{10}\, w_{11}\, w_{12}\, w_{13}\, w_{14}\, w_{15} &\rightarrow \textit{Round}\ 1 \\ w_{16}\, w_{17}\, w_{18}\, w_{19}\, w_{20}\, w_{21}\, w_{22}\, w_{23} &\rightarrow \textit{Round}\ 2 \\ &\vdots \end{aligned}$$

4.4 ALGORİTMANIN TERSİ

Algoritmaın Tersi

Algoritmanın tersinde katmanların tersi uygulanır.

4.4.1 Mix Column

Mix Column'da kullanılan polinomun tersi kullanılır. $c^{-1}(x) = d(x) = \text{`}0B\text{`}x^3 + \text{`}0D\text{`}x^2 + \text{`}09\text{`}x + \text{`}0E\text{`}$ çarpma işlemi yine modulo $M(x) = x^4 + 1$'de yapılır.

4.4.2 Byte Sub

Byte Sub katmanında yapılan işlemlerin tersi yapılır.

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \left(\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \right)$$

Bulunan $x = (x_7\, x_6\, x_5\, x_4\, x_3\, x_2\, x_1\, x_0)$'in modulo $m(x) = x^8 + x^4 + x^3 + x + 1$'de tersi alınır.