Der Advanced Encryption Standard im Detail

Im Standard wurde eine fixe Blocklänge von 128 Bits gewählt. Ein Eingangsblock besteht also aus 128 binären Ziffern, die gemäss der Beziehung

$$in_n = \{input_{8\cdot n}, input_{8\cdot n+1}, \ldots, input_{8\cdot n+7}\}, \quad 0 \le n < 16$$

auf die 16 Bytes in_0 bis in_{15} aufgeteilt werden. Diese wiederum werden in einer 4×4 Zustandsmatrix abgelegt.

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{3,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} in_0 & in_4 & in_8 & in_{12} \\ in_1 & in_5 & in_9 & in_{13} \\ in_2 & in_6 & in_{10} & in_{14} \\ in_3 & in_7 & in_{11} & in_{15} \end{bmatrix}.$$

Die weiteren Berechnungen finden innerhalb dieser Zustandsmatrix statt, wobei die einzelnen Bytes jeweils als Elemente des endlichen Körpers GF(2^8) interpretiert werden. Das heisst, jedem Byte

$$\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$$

wird formal ein Polynom

$$b_7 \cdot x^7 + b_6 \cdot x^6 + b_5 \cdot x^5 + b_4 \cdot x^4 + b_3 \cdot x^3 + b_2 \cdot x^2 + b_1 \cdot x^1 + b_0$$

vom Grad 7 zugeordnet. Zwei Bytes werden addiert, indem die Koeffizienten der dazugehörigen Polynome XOR-verknüpft werden, also beispielsweise

$$\{01010111\} + \{10000011\} = \{11010100\}.$$

Für die Multiplikation zweier Bytes werden die entsprechenden Polynome zunächst miteinander multipliziert. Der Grad des resultierenden Polynoms ist unter Umständen grösser als 7. Deshalb wird anschliessend durch das Polynom

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

dividiert. Der Rest, der aus dieser Division resultiert, ist ein Polynom, dessen Grad kleiner als 8 ist und dem folglich ein Byte zugeordnet werden kann. Dieses Byte ist das gesuchte Produkt.

Beispiel

Beispielsweise werden die beiden Bytes {01010111} und {10000011} durch die Polynome

$$x^6 + x^4 + x^2 + x + 1$$

und

$$x^7 + x + 1$$

repräsentiert. Die Multiplikation liefert das Ergebnis

$$\left(x^6 + x^4 + x^2 + x + 1\right)\left(x^7 + x + 1\right) = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1.$$

Die Division durch m(x) lässt den Rest

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

was dem Byte {11000001} entspricht. ■

Für die Verschlüsselung werden vier Transformationen definiert, die jeweils die Zustandsmatrix verändern:

SubBytes

Dabei handelt es sich um eine nichtlineare Substitution. Jedes Byte der Zustandsmatrix wird mit Hilfe einer Tabelle durch einen anderen Wert ersetzt. Im Gegensatz zum DES ist beim AES genau bekannt, wie diese Substitutionstabelle erzeugt wurde. Der Wert des Bytes wird nämlich im Körper $GF(2^8)$ invertiert und danach affin[1] transformiert. Da beide Schritte invertierbar sind, ist die dadurch definierte Substitution eineindeutig und damit umkehrbar.

Beispiel

Es lässt sich leicht überprüfen, dass {0 1 0 1 0 1 0 1} der Kehrwert von {0 0 1 0 0 1 0 0} ist, denn es gilt

$$\left(x^5+x^2\right)\cdot\left(x^6+x^4+x^2+1\right) \bmod \left(x^8+x^4+x^3+x+1\right) = 1.$$

Die affine Transformation

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

liefert das Ergebnis {0 0 1 1 0 1 1 0}. Folglich ersetzt SubBytes das Byte {0 0 1 0 0 1 0 0} durch das Byte {0 0 1 1 0 1 1 0}. ■

ShiftRows

Bei der ShiftRows-Transformation werden die vier Zeilen der Zustandsmatrix um jeweils unterschiedlich viele Stellen zyklisch nach links verschoben. Die erste Zeile der Matrix bleibt unverändert, die zweite Zeile wird um ein Byte, die dritte Zeile um zwei Bytes und die vierte Zeile um drei Bytes nach links verschoben.

---

[1] Eine affine Transformation ist von der Form: $y = A \cdot x + b$

MixColumns

Hierbei handelt es sich um die komplexeste Operation. Aus den vier Bytes einer Spalte der Zustandsmatrix wird ein Polynom vom Grad 3 gebildet

$$s(x) = s_{3,c} \cdot x^3 + s_{2,c} \cdot x^2 + s_{1,c} \cdot x + s_{0,c}\,.$$

Dieses Polynom wird zunächst mit dem fixen Polynom

$$\begin{aligned} a(x) &= a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0 \\ &= \{00000011\} \cdot x^3 + \{00000001\} \cdot x^2 + \{00000001\} \cdot x + \{00000010\} \end{aligned}$$

multipliziert. Zu beachten ist dabei, dass die Koeffizienten der Polynome aus dem Körper GF(2^8) stammen und dass deshalb bei Addition und Multiplikation die oben beschriebenen Regeln zur Anwendung kommen. Das Produkt von s(x) und a(x) ist im Allgemeinen vom Grad 6, kann also nicht mehr mit vier Bytes dargestellt werden. Deshalb wird anschliessend durch $x^4 + 1$ dividiert und nur noch der Rest betrachtet. Die Berechnung dieses Restes wird durch die Beziehung

$$x^j \bmod \left(x^4 + 1\right) = x^{j \bmod 4}$$

erleichtert. Schliesslich erhält man ein Polynom vom Grad 4

$$d(x) = d_3 \cdot x^3 + d_2 \cdot x^2 + d_1 \cdot x + d_0\,,$$

dessen vier Koeffizienten die transformierte Spalte der Zustandsmatrix bilden.

Da die Koeffizienten des Polynoms a(x) konstant sind, kann die gesamte Transformation mit Hilfe einer Matrix beschrieben werden

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_2 \end{bmatrix} \cdot \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}.$$

Das Polynom a(x) respektive dessen Koeffizienten a_i wurden so gewählt, dass die Transformation invertierbar ist. Es gilt nämlich

$$a^{-1}(x) = \{00001011\} \cdot x^3 + \{00001101\} \cdot x^2 + \{00001001\} \cdot x + \{00001110\}\,.$$

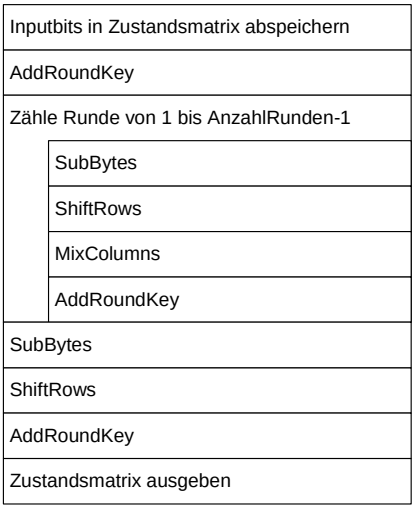
AddRoundKey

Bei der AddRoundKey-Transformation wird die Zustandsmatrix bitweise mit einem Rundenschlüssel XOR-verknüpft. Diese Rundenschlüssel werden nach bestimmten Regeln aus den Schlüsselbits berechnet.

Ablauf der Verschlüsselung

Bei der Verschlüsselung werden zunächst die Inputbits in die Zustandmatrix übertragen. Danach wird die Zustandsmatrix zum ersten Mal mit einem Rundenschlüssel verknüpft. Anschliessend folgen 9, 11 oder 13 identische Runde, je nachdem, ob die Schlüssellänge 128, 192 oder 256 Bits

beträgt. In jeder Runde werden die Transformationen SubBytes, ShiftRows, MixColumns und AddRoundKey nacheinander ausgeführt. Es folgt eine letzte Runde, die sich von den vorangegangenen nur dadurch unterscheidet, dass die Transformation MixColumns wegfällt. Abschliessend werden die Bytes der Zustandsmatrix als Geheimtext ausgegeben.



Figur 1: Verschlüsselung mit AES. Die Anzahl Runden ist von der Länge des Schlüssels abhängig.

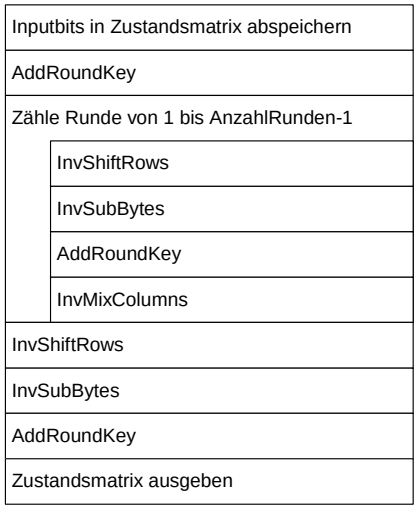
Ablauf der Entschlüsselung

Alle in der Verschlüsselung verwendeten Transformationen sind invertierbar. Die Transformation AddRoundKey ist ihre eigene Umkehrfunktion. In der Transformation InvSubBytes werden die Substitutionen rückgängig gemacht, die Transformation InvShiftRows verschiebt die Zeilen der Zustandsmatrix um entsprechend viele Zeilen zyklisch nach rechts und bei InvMixColumns wird das aus den Spalten der Zustandsmatrix gebildete Polynom mit

$$a^{-1}(x) = \{00001011\} \cdot x^3 + \{00001101\} \cdot x^2 + \{00001001\} \cdot x + \{00001110\}$$

multipliziert und anschliessend der Rest bei Division durch $x^4 + 1$ bestimmt.

Um die Entschlüsselung zu realisieren, müssen diese Umkehrtransformationen in umgekehrter Reihenfolge und mit dem jeweils richtigen Rundenschlüssel abgearbeitet werden.



Figur 2: Ablauf der Entschlüsselung bei AES.