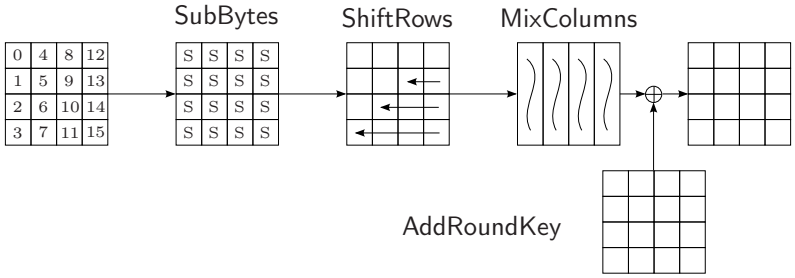
4. Übungsblatt
Kryptographie und Mediensicherheit (SoSe 2018)

Bauhaus-Universität Weimar, Professur für Mediensicherheit (Prof. Lucks)
Betreuer: Eik List
URL: `http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching`
Abgabe: 29.05.2017, 11:00 Uhr vor Beginn der Übung oder an `eik.list(at)uni-weimar.de`.
LaTeX-Template: template.tex

Aufgabe 1 – Aufbau des AES' (6 Punkte)
Wiederholen Sie den Aufbau und die Diffusionseigenschaften des AES' aus Kapitel 4 der Vorlesung. Unten finden Sie eine Grafik mit dem Aufbau einer AES-Runde:



Die Spalten 0, 1, 2, 3 werden von links nach rechts, die Zeilen 0, 1, 2, 3 von oben nach unten nummeriert. Betrachten Sie folgende modifizierte Varianten des AES.

a) In jeder Runde ist die Operation ShiftRows ersetzt mit einer Operation ShiftColumns, welche die Bytes der i-ten Spalte um i Positionen nach oben statt nach links rotiert. Alle anderen Operationen bleiben unverändert.

b) In jeder Runde werden die Bytes die i-te Zeile in der ShiftRows-Operation um $2i \bmod 4$ (statt um i) Positionen nach links rotiert. Alle anderen Operationen bleiben unverändert.

c) In jeder Runde wird jede S-box S ersetzt durch $S(x_i) \stackrel{\text{def}}{=} x_i \oplus \texttt{01010101}$. Alle anderen Operationen bleiben unverändert.

Geben Sie für jede der Varianten entweder eine kurze Erläuterung warum der AES mit dieser Modifikation immer noch sicher ist *oder* zeigen Sie jeweils kurz warum nicht. Sie finden eine vereinfachte Implementation des AES, SimpleAES, als `simple_aes.py` auf der Übungswebseite die Sie anpassen können.

Aufgabe 2 – SimpleAES (4 Punkte)
SimpleAES (die vereinfachte Implementierung aus der vorigen Aufgabe) nutzt nicht den originalen Key Schedule des AES sondern XORt in jeder Runde den gleichen geheimen 16-byte-Schlüssel K auf den Zustand. Wählen Sie einen Klartext $P = (x_0, \ldots, x_{15})$ der in jedem Byte gleich ist, d. h. $x_0 = x_1 = \ldots = x_{15}$ und verschlüsseln Sie ihn mit SimpleAES unter einem Schlüssel $K = (k_0, \ldots, k_{15})$ der ebenfalls in jedem Byte gleich ist: $k_0 = k_1 = \ldots = k_{15}$.

a) Was beobachten Sie?

b) Zeigen Sie dass diese Eigenschaft für alle derartige Klartext-Schlüssel-Paare für den SimpleAES gilt.

Hinweis: Sie können folgende Eigenschaften der MixColumns-Operation verwenden: Für alle Werte $x \in \mathbb{GF}(2^8)$ gilt $(3 \cdot x) = (2 \cdot x) \oplus (1 \cdot x)$.

Aufgabe 3 – Difference-Distribution Table (Programmieraufgabe) (4 Punkte)
Sei $S : \{0,1\}^n \to \{0,1\}^n$ eine n- zu n-bit-Sbox. Für zwei beliebige Differenzen $\Delta_{\text{in}}, \Delta_{\text{out}} \in \{0,1\}^n$ schreiben wir $\Delta_{\text{in}} \xrightarrow{S} \Delta_{\text{out}}$ für das Differential durch S. Die Kardinalität $C_S(\Delta_{\text{in}}, \Delta_{\text{out}})$ eines Differentials ist die Anzahl der Paare (x, x') mit Eingabedifferenz $x \oplus x' = \Delta_{\text{in}}$, die auf Ausgaben $(S(x), S(x'))$ mit Differenz $S(x) \oplus S(x') = \Delta_{\text{out}}$ abgebildet werden. Die Wahrscheinlichkeit ist dann einfach diese Anzahl geteilt durch 2^n:

$$C_S(\Delta_{\text{in}}, \Delta_{\text{out}}) \stackrel{\text{def}}{=} |\{x \in \{0,1\}^n : S(x) \oplus S(x \oplus \Delta_{\text{in}}) = \Delta_{\text{out}}\}|$$
$$\Pr\left[\Delta_{\text{in}} \xrightarrow{S} \Delta_{\text{out}}\right] \stackrel{\text{def}}{=} \frac{C_S(\Delta_{\text{in}}, \Delta_{\text{out}})}{2^n}.$$

Eine Difference-Distribution Table $\text{DDT}_S : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_{2^n}$ enthält für jedes Differential $(\Delta_{\text{in}}, \Delta_{\text{out}})$ dessen Kardinalität:

$$\text{DDT}_S[\Delta_{\text{in}}][\Delta_{\text{out}}] \stackrel{\text{def}}{=} C_S(\Delta_{\text{in}}, \Delta_{\text{out}}).$$

Schreiben Sie ein Programm in Python 3, welches eine S-Box S entgegennehmen kann und DDT_S erzeugt. Die betrachtete S-Box soll dabei als Inhalt einer Datei (Pfad, Kommandozeilenparameter) übergeben werden. Dabei steht jener Wert x_i an i-ter Stelle in der Datei, für den gilt: $S(i) = x_i$.
Beispiel: Sei $S = [5, 1, 3, 6, 4, 0, 2, 7]$ eine 3×3-bit S-box. D. h., $S(0) = 5$, $S(1) = 1$, …, $S(7) = 7$. Der Inhalt von `sbox.txt` ist:

```
5,1,3,6,4,0,2,7
```

Dann soll die Ausgabe Ihres Programms liefern:

```
$ python3 ddt_12345.py -i sbox.txt
    00 01 02 03 04 05 06 07
00   8  0  0  0  0  0  0  0
01   0  0  0  0  8  0  0  0
02   0  0  0  4  0  0  0  4
03   0  0  0  4  0  0  0  4
```

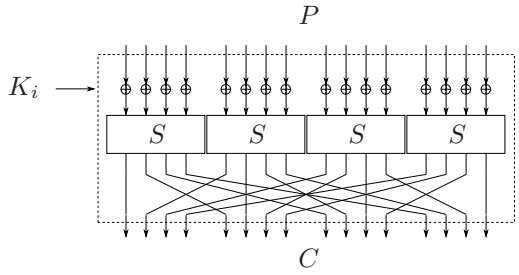
```
04   0  4  0  0  0  4  0  0
05   0  4  0  0  0  4  0  0
06   0  0  4  0  0  0  4  0
07   0  0  4  0  0  0  4  0
```

Die Eingabedifferenzen stehen in horizontaler Richtung, d. h. $C_S(\texttt{0x04} \to \texttt{0x01}) = 8$. Schicken Sie Ihre Lösung als Anhang einer E-Mail `ddt_<MatrNr>.py` mit dem Betreff `[Krypto SS18] Beleg 4` an `eik.list(at)uni-weimar.de`. Es reicht dabei eine Matrikelnummer aus der Gruppe anzugeben. Die vollständigen Namen und Matrikelnummern sollen als Kommentar im Python-Skript stehen.

Aufgabe 4 – Differentielle Charakteristiken (6 Punkte)
Sei $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ eine Rundenfunktion. Eine differentielle Charakteristeristik über r Runden ist ein $(r+1)$-Tupel $(\Delta_0, \Delta_1, \ldots, \Delta_r)$, wobei jedes $\Delta_i \in \{0,1\}^n$ die Differenz nach i Runden repräsentiert. Wir nehmen an dass alle Rundenschlüssel unabhängig zufällig gewählt sind. Die Wahrscheinlichkeit einer Charakteristik berechnet sich dann durch das Produkt der Wahrscheinlichkeiten der einzelnen Differentiale durch je eine Runde:

$$\Pr\left[\Delta_0 \xrightarrow{F_{K_r} \circ \cdots \circ F_{K_2} \circ F_{K_1}} \Delta_r\right] = \prod_{i=1}^{r} \Pr\left[\Delta_{i-1} \xrightarrow{F_{K_i}} \Delta_i\right]$$

Gegeben sei eine 16-bit-Chiffre $E : \{0,1\}^k \times \{0,1\}^{16} \to \{0,1\}^{16}$. E besitzt r Runden, von denen jede aus dem XOR mit einem Rundenschlüssel K_i, einem S-Box-Layer und einer Permutation der Bitpositionen besteht (siehe Abbildung). Nach der letzten Runde findet ein XOR mit einem finalen Rundenschlüssel K_{r+1} statt, bevor der Chiffretext C ausgegeben wird.



1 Runde

x	0	1	2	3	4	5	6	7
$S(x)$	E	7	1	A	3	2	B	D
x	8	9	A	B	C	D	E	F
$S(x)$	6	0	F	8	5	C	9	4

a) Ermitteln Sie für die gegebene S-box $S : \{0,1\}^4 \to \{0,1\}^4$ die Wahrscheinlichkeiten aller 1-bit-Differentiale, d. h., $\Pr\left[\Delta_{\text{in}} \xrightarrow{S} \Delta_{\text{out}}\right]$ für $\Delta_{\text{in}}, \Delta_{\text{out}} \in \{1000, 0100, 0010, 0001\}$.

b) Seien $\Delta_{\text{in}}, \Delta_{\text{out}} \in \{0,1\}^{16}$ zwei von Ihnen gewählte 16-bit-Differenzen mit der einzigen Einschränkung dass $\Delta_{\text{in}}, \Delta_{\text{out}} \neq 0^{16}$. Finden Sie eine Charakteristik $\Delta_{\text{in}} \to \Delta_{\text{out}}$ mit Wahrscheinlichkeit $\geq 1/2^8$ über **vier Runden** und geben Sie die Wahrscheinlichkeit Ihrer Charakteristik an. Sie können Ihre Erkenntnisse aus Aufgabenteil a) nutzen.

Sie finden eine Implementation der Chiffre als `minipresent.py` auf der Webseite der Übung.