2. 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: 08.05.2018, 11:00 Uhr vor Beginn der Übung oder an `eik.list(at)uni-weimar.de`.
LaTeX-Template: template.tex

Hinweis: Denken Sie daran, stets nachvollziehbare Lösungswege anzugeben. Für Lösungen, deren Herkunft nicht nachvollziehbar ist, werden 0 Punkte vergeben. Schreiben Sie mindestens eine Matrikelnummer in den Dateinamen Ihrer Abgaben.

Definition 1 (PZBG-Vorteil)**.** Ein PZBG-Angreifer $\mathbf{A}$ interagiert mit einem Orakel, dass einen geheimen Schlüssel $K \in \{0,1\}^k$ hat und eine faire Münze b wirft. Das Orakel gibt einen n-Bit-Wert $X \in \{0,1\}^n$ an $\mathbf{A}$.

- $b = 1$: Orakel gibt $X \in \{0,1\}^n$ als Ergebnis eines PZBGs $F : \{0,1\}^k \to \{0,1\}^n$ aus.
- $b = 0$: Orakel wählt X als Ergebnis von n Würfen mit einer fairen Münze.

Das Ziel von $\mathbf{A}$ ist es, b korrekt zu erraten. Wir bezeichnen mit $\mathbf{A} \Rightarrow 1$ das Ereignis, dass $\mathbf{A}$ rät dass $b = 1$ sei. Der PZBG-Vorteil von $\mathbf{A}$ auf F ist

$$\mathbf{Adv}_F^{\mathrm{PZBG}}(\mathbf{A}) = \left|\Pr\left[\mathbf{A} \Rightarrow 1 | b = 1\right] - \Pr\left[\mathbf{A} \Rightarrow 1 | b = 0\right]\right| .$$

Aufgabe 1 – PZBG-Sicherheit (4 Punkte)
Sei $F : \{0,1\}^k \to \{0,1\}^n$ ein sicherer PZBG und $\wedge$ die bitweise logische AND-Verknüpfung. Seien K_1 und K_2 zwei unabhängige geheime Schlüssel und der PZBG $F' : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}^n$ definiert als $F'(K_1, K_2) = F(K_1) \wedge F(K_2)$.

a) Sei $n = 1$ Bit. Geben Sie einen effizienten PZBG-Angreifer $\mathbf{A}$ mit signifikantem Vorteil auf F' an. Was ist der Vorteil von $\mathbf{A}$?

b) Sei $n = 3$ Bits. Geben Sie einen effizienten PZBG-Angreifer $\mathbf{A}$ mit höherem Vorteil auf F' an als in a) und berechnen Sie seinen Vorteil.

Aufgabe 2 – PRF-Sicherheit (7 Punkte)
PRF-Sicherheit ist sehr ähnlich zu PZBG-Sicherheit. Bei Ersterer darf der Angreifer Eingaben wählen und erhält als Antworten Y entweder stets (bei $b = 1$) echte Berechnungen einer Funktion F deren Schlüssel geheim ist, $Y = F_K(X)$, oder stets Zufallswerte $Y \leftarrow \{0,1\}n$ (bei $b = 0$). Das Ziel eines PRF-Angreifers ist wieder b korrekt vorherzusagen.
Sei $F : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^n$ eine sichere Pseudozufallsfunktion (PRF). Das heißt, es existiert kein effizienter Algorithmus, der die Ausgaben von F mit signifikantem Vorteil von aus

n unabhängigen Würfen einer fairen Münze erhaltene Zufallsbits unterscheiden kann. Geben Sie für die folgenden Funktionen F' jeweils an ob auch sie sichere PRFs sind *oder* beschreiben Sie jeweils kurz einen effizienten Angriff mit max. 2 Anfragen:

a) $F'_K(X) = F_K(X)[0..n-2]$ (das letzte Bit wird nicht ausgegeben)

b) $F'_K(X) = F_K(X) \oplus c$, wobei c eine öffentlich bekannte Konstante ist

c) $F'_K(X) = F_K(X) \oplus \overline{F_K(X)}$, wobei $\overline{X}$ das bitweise Inverse von X bezeichnet

d) $F'_{K_1,K_2}(X) = F_{K_1}(X) \oplus F_{K_2}(X)$, wobei K_1 und K_2 unabhängige und zufällig gewählte geheime Schlüssel sind.

e) $F'_K(X) = F_K(X) \oplus F_K(\overline{X})$

f) $F'_K(X) = \text{reverse}(F_K(X))$ (die Reihenfolge der Bits wird umgedreht)

g) $F'_K(X) = F_K(X) \,\|\, c$ (der Ausgabe wird ein konstanter m-Bit-Wert c angehangen)

Aufgabe 3 – Entropie (4 Punkte)
Das ASCII-Alphabet bietet 95 druckbare Zeichen: 26 Kleinbuchstaben, 26 Großbuchstaben, 10 Ziffern und 33 druckbare Sonderzeichen. Berechnen Sie die Shannon-Entropie der folgenden Passwortgenerierungsverfahren:

a) Viermaliges zufälliges Ziehen (mit Zurücklegen) aus einem Wörterbuch von 2^{12} Worten.

b) Die Aneinanderreihung von jeweils zufällig gleichverteilt gewählten vier Ziffern, gefolgt von vier Klein-, danach von vier Großbuchstaben und danach vier Sonderzeichen (mit Zurücklegen).

c) Die Aneinanderreihung von vier zufällig gewählten Ziffern, gefolgt von vier Kleinbuchstaben, vier Großbuchstaben und vier Sonderzeichen (ohne Zurücklegen, d.h. jedes Zeichen kommt maximal einmal im Passwort vor).

d) Die Kombination 16 zufällig gewählter unterschiedlicher ASCII-Zeichen (d.h. ohne Zurücklegen).

Aufgabe 4 – Password-Cracking (Programmieraufgabe) (4 Punkte)
Die folgenden sechs Hashwerte wurden mit Hilfe der Hashfunktion SHA-1 erzeugt

(1) `d6f2d9d65fe3acdceed29f17062b07c4e427f9ad`
(2) `c283e375ed8cebf3b8d1b5101fd51bb522961656`
(3) `54435a836007fcf7d77b44e3dc82ad68a39ad852`
(4) `1572bd30ac06678a82df42b5913e5e52e27f9a12`
(5) `0fe74481e67876a75541e5341659839104a44b11`
(6) `cc5a0c506761ce686f1f145da868c162f269480f`

Implementieren Sie ein Programm in Python 3, welches zu gegebenen Hashwerten die dazugehörigen Passwörter rekonstruiert. Ihr Programm soll

a) die Hashwerte aus einer Textdatei entgegennehmen (ein Hash pro Zeile, Textdatei als Kommandozeilenparameter),

b) die Passwörter (bestehend aus je genau sechs Kleinbuchstaben) für die oben angegebenen Hashwerte effizient ermitteln,

c) und Passwörter zusammen mit ihren Hashwerten ausgeben.

Sie müssen die Funktion SHA-1 nicht selbst implementieren, sondern können eine vorhandene Implementierung nutzen. Zum Beispiel bietet Ihnen das Modul `hashlib` eine Implementierung von SHA-1 an.
Schicken Sie die Lösungen als Anhang einer E-Mail

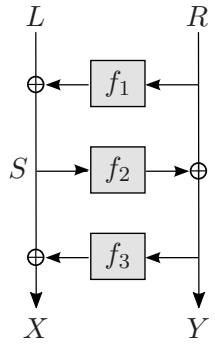
`recover_passwords_<MatrNr>.py`

an `eik.list(at)uni-weimar.de` mit dem Betreff `[Krypto SS18] Beleg 2` bis zum 08. Mai 2018, 11:00 Uhr. Es reicht dabei eine Matrikelnummer aus der Gruppe anzugeben.

Hinweis: Die untere(n) Aufgabe(n) sind Teil des 3. Aufgabenblatts und bis zum 15.05.2018 abzugeben. Sie sind hier bereits veröffentlicht damit ausreichend Zeit für ihre Bearbeitung ist.

Aufgabe 5 – 3-Runden-Feistel (3 Punkte)
Gegeben ein P3. Seien $f_1, f_2, f_3 : \{0,1\}^4 \to \{0,1\}^4$ sichere unabhängige Zufallsfunktionen. Gegeben seien die Klartexte (L_i, R_i) und die dazugehörigen Chiffretexte (X_i, Y_i). Geben Sie nachvollziehbar einen neuen Chiffretext (X, Y) und mindestens die Hälfte des dazugehörenden Klartextes an.

i	L_i	R_i	X_i	Y_i
1	0000	0000	1011	0001
2	0000	0001	1001	1000
3	0001	0000	0110	0101
4	0001	0001	1011	1100



Aufgabe 6 – 4-Runden-Feistel (6 Punkte)
Gegeben seien die folgenden Luby-Rackoff-Konstruktionen P4 mit den Funktionen $f_1, f_2, f_3, f_4 : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ und einem zufällig gewählten geheimen Schlüssel k. Beschreiben Sie für jede der untenstehenden Konstruktionen einen CPA- *oder* CCA-Angreifer mit jeweils maximal zwei Anfragen. Geben Sie zudem jeweils den Vorteil des Angreifers mit an.

a) Erster $P4$:

- $f_2(k, x) := f_1^{-1}(k, x)$
- $f_3(k, x) := f_1(k, x)$
- $f_4(k, x) := f_1(k, x)$

b) Zweiter $P4$:

- $f_1(k, x) := f_4(k, x)$
- $f_2(k, x) := f_3(k, x)$