6. 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: 26.06.2018, 11:00 Uhr vor Beginn der Übung oder an `eik.list(at)uni-weimar.de`.
LaTeX-Template: template.tex

Aufgabe 1 – Betriebsmodi (4 Punkte)
Informieren Sie sich, (z. B. in Kapitel 8.1ff. der Vorlesung) zu den Betriebsmodi Counter (CTR), Chaining Block Cipher (CBC), sowie selbständig zu Output Feedback (OFB) und Ciphertext Feedback (CFB). Vergleichen Sie sie im Hinblick auf die folgenden Kriterien:

- Parallel (ja/nein): Ist die Ver- bzw. die Entschlüsselung blockweise parallelisierbar?
- Schlüsselstrom (ja/nein): Kann man einen Schlüsselstrom vorabberechnen der beim Eintreffen der Nachricht nur noch auf diese XORt werden muss?
- Inversfrei (ja/nein): Kommt die Entschlüsselung von Chiffretexten ohne die Implementierung der inversen Blockchiffrenoperation aus?
- Längenerhaltend (ja/nein): Sind Chiffretexte stets so lang wie die entsprechenden Klartexte, insbesondere für Klartexte deren Länge kein Vielfaches der Blockgröße ist?
- #Blöcke (Zahl): Angenommen, in einem Chiffretext $(C_1, \ldots, C_m)$ wird ein Bit in C_1 fehlerhaft übertragen. Auf wieviele Klartextblöcke breitet sich der Fehler aus?

		Verschlüsselung			Entschlüsselung	
Modus	#Blöcke	Parallel	Schlüsselstrom	Längenerhaltend	Parallel	Inversfrei
CTR						
CBC						
OFB						
CFB						

Nonce- und IV-basierte Verschlüsselung. Das Sicherheitsmodell für Verschlüsselungsverfahren ist Ununterscheidbarkeit von zufälligen Bitstrings gleicher Länge gegen Chosen-Plaintext-Angriffe (RoR-CPA, siehe z. B. in Kapitel 8, Folie 255ff. der Vorlesung). Um zu verhindern dass gleiche Nachrichten gleiche Chiffretexte produzieren benötigen Verschlüsselungsverfahren einen zusätzlichen Input. Die üblichen Ansätze sind Nonce-basierte, randomisierte und zustandsbehaftete Verschlüsselung.

Seien $\mathcal{K}$, $\mathcal{M}$ und $\mathcal{C}$ eine Menge an Schlüsseln, Klartexten und Chiffretexten, $\mathcal{N}$ eine Menge an Nonces und $\mathcal{IV}$ eine Menge an Initialwerten. Eine nonce-basierte Verschlüsselungsfunktion $\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{M} \to \mathcal{C}$ erhält einen Schlüssel $K \in \mathcal{K}$, eine Nonce $N \in \mathcal{N}$ und eine Nachricht $M \in \mathcal{M}$, berechnet $C = \mathcal{E}_K(N, M)$ und gibt C aus. Die Nonce N kann beliebig gewählt werden solange für jede Verschlüsselung eine andere Nonce gewählt wird.

Eine IV-basierte Verschlüsselungsfunktion $\mathcal{E} : \mathcal{K} \times \mathcal{IV} \times \mathcal{M} \to \mathcal{C}$ erhält einen Schlüssel K und eine Nachricht M, wählt den $IV \twoheadleftarrow \mathcal{IV}$ zufällig und unabhängig von vorigen IV's, berechnet $C = \mathcal{E}_K(IV, M)$ und gibt den Chiffretext (IV, C) aus. Der IV wird von der Verschlüsselungsfunktion (d. h., nicht vom Nutzer und nicht vom Angreifer) gewählt.

Bei Nonce- und IV-basierter Verschlüsselung wird die Nonce bzw. der IV mit dem Chiffretext übertragen. Bei zustandsbehafteten Verfahren wird ein Zustand bei jeder Verschlüsselung aktualisiert und intern verwaltet. Dieser braucht bei synchronen Sendern und Empfängern nicht verschickt zu werden.

Aufgabe 2 – Sicherheit von Nonce- und IV-basierten Betriebsmodi (8 Punkte)
Bei blockchiffrenbasierten Betriebsmodi werden Nonce bzw. IV oft als C_0 bezeichnet oder C_0 wird aus ihnen abgeleitet. Sofern nicht anders vorgegeben nehmen wir in der Folge $\mathcal{N} = \mathcal{IV} = \{0,1\}^n$ an. $E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ sei eine sichere Blockchiffre. Wir definieren die Verschlüsselung einer Nachricht $M = (M_1, \ldots, M_m)$ im CTR-Modus als

$$I_i = (C_0 + \mathsf{tostr}_n(i-1)) \bmod 2^n \quad \text{und} \quad C_i = M_i \oplus \mathsf{msb}_{|M_i|}(E_K(I_i)), \quad \text{für alle } 1 \le i \le m.$$

$\mathsf{msb}_k(x)$ gibt die k most-significant bits von x aus, $\mathsf{tostr}_k(x)$ wandelt eine Ganzzahl $x \ge 0$ in einen k-bit-String um. Wir definieren die Verschlüsselung einer Nachricht $M = (M_1, \ldots, M_m)$ im CBC-Modus als

$$I_i = M_i \oplus C_{i-1} \quad \text{und} \quad C_i = E_K(I_i), \quad \text{für alle } 1 \le i \le m.$$

Für **vier** der folgenden Teilaufgaben, begründen Sie jeweils **entweder** kurz warum der Modus sicher ist **oder** geben Sie einen effizienten RoR-CPA-Angreifer mit möglichst wenigen Anfragen an.

a) Nonce-basierte CTR-Verschlüsselung: $C_0 = N$.

b) Nonce-basierte CBC-Verschlüsselung: $C_0 = N$.

c) Nonce-basierte CTR-Verschlüsselung mit $C_0 = E_K(N)$. C_0 wird nicht ausgegeben.

d) Nonce-basierte CBC-Verschlüsselung mit $C_0 = E_K(N)$. C_0 wird nicht ausgegeben.

e) Nonce-basierte CTR-Verschlüsselung, aber ein k-bit-Counter wird an eine kürzere $(n-k)$-bit Nonce angehangen statt XORt. D. h., $\mathcal{N} = \{0,1\}^{n-k}$ und $I_i = N \,\|\, \mathsf{tostr}_k(i-1)$. Jede Nachricht besteht aus maximal 2^k Blöcken.

f) Zustandsbasierte CBC-Verschlüsselung: Der Startwert C_0^j für die Verschlüsselung der j-ten Nachricht (nicht j-ten Block!) M^j ist der Chiffretext des letzten Blocks der vorigen Nachricht: $C_0^j = C_m^{j-1}$; der erste Initialwert $C_0^1 \twoheadleftarrow \{0,1\}^n$ ist zufällig. C_0 wird für keine Nachricht jemals ausgegeben.

Bonus (+1/Teilaufgabe): Bearbeiten Sie mehr als vier Teilaufgaben.

Aufgabe 3 – Authentizität von Betriebsmodi (2 + 2 Punkte)
Alice möchte die 32-Zeichen-Nachricht

$$M = (M_1, M_2) = \texttt{Zahle␣Bob␣100EUR␣von␣Konto␣12345}$$

verschlüsselt an Ihre Bank schicken. Dazu kodiert Sie die Nachricht M byteweise in 8-bit ASCII, verschlüsselt sie mit AES in einem Betriebsmodus und erhält den Chiffretext $C = (IV, C_1, C_2)$, den sie anschließend verschickt:

$$\begin{aligned} IV &= \texttt{0xdabe785fb5a688a6 1cc77c3035d754fa} \\ C_1 &= \texttt{0x5973ecca8de35880 102bd68ba0ede12b} \\ C_2 &= \texttt{0x42f2dc27a8a676d6 9e1825d323016d38} \end{aligned}$$

Eve fängt nun den Chiffretext von Alice ab. Sie möchte stattdessen den Chiffretext $C' = (IV', C_1', C_2')$ zur Nachricht $M' = (M_1', M_2') =$ "`Zahle␣Eve␣500EUR␣von␣Konto␣12345`" in Alice' Namen weiterleiten.

a) Geben Sie einen gültigen Chiffretext C' zu M' an, wenn Alice und Eve Counter-Mode verwenden.

b) Geben Sie einen gültigen Chiffretext C' zu M' an, wenn Alice und Eve CBC-Mode verwenden.

Begründen Sie Ihre Antworten. Beachten Sie, dass der IV im Klartext übertragen wird und Eve ihn beliebig verändern darf.

Aufgabe 4 – Geburtstags-Angriff auf Textbook-RSA (5 Punkte)
Informieren Sie sich (z. B. in Kapitel 7.3 der Vorlesung) zum Geburtstags-Angriff auf Textbook-RSA. Sei $C = M^e \bmod n$ und M das Produkt zweier kleiner Primteiler: $M = M_1 \cdot M_2$ für $M_1, M_2 \leq b$.
Implementieren Sie den Geburtstags-Angriff in Python 3. Erreichen Sie 5/10 in `pylint` bzw. `pylint3`. Finden Sie für ein gegebenes (n, e, C, b) mit $b = 2^{21}$ die dazugehörende Nachricht M, die genauen Werte sind als Textdatei auf der Übungsseite gegeben. Ihr Skript soll (n, e, C, b) als Parameter einlesen und M ausgeben:

```
$ python3 rsa_mitm_attack.py -i params.txt
```

Schicken Sie Ihre Lösung als Anhang einer E-Mail

`rsa_mitm_attack_<matrikelnr>.py`

an `eik.list(at)uni-weimar.de` mit dem Betreff `[Krypto SS18] Beleg 6`. Es reicht dabei eine Matrikelnummer aus der Gruppe anzugeben. Die vollständigen Namen und Matrikelnummern sollen als Kommentar im Python-Skript stehen.
Hinweis: Eine geeignete Bibliothek für modulare Arithmetik mit großen Ganzzahlen ist `gmpy`.