Institut für Theoretische Informatik
Jun.-Prof. Dr. D. Hofheinz

Stammvorlesung Sicherheit im Sommersemester 2014

Übungsblatt 3

Hinweis: Übungsblätter können freiwillig bei Florian Böhl, Raum 255, Geb. 50.34 („Info-Bau“) bis zur Übung am 22.5.14 zur Korrektur abgegeben werden. Die Korrektur dient nur der Selbstkontrolle; es gibt keine Punkte und keinen Klausur-Bonus.

Aufgabe 1. Beurteilen Sie für die folgenden Konstruktionen jeweils, ob es sich für beliebige kollisionsresistente Hashfunktionen $H, H' : \{0,1\}^* \to \{0,1\}^k$ ebenfalls um kollisionsresistente Hashfunktionen handelt. Falls ja, beweisen Sie diese Aussage indem Sie zeigen, wie sich aus einer Kollision für $\hat{H}$ eine Kollision für H oder H' entwickeln ließe. Falls nicht, geben Sie konkrete Wahlen für H und H' und eine Kollision an.

(a) $\hat{H}(x) := H(x||x)$

(b) $\hat{H}(x) := H(x) \oplus H'(x)$

(c) $\hat{H}(x) := H(H'(x))$

(d) $\hat{H}(x) := F(x)||H(x)$ für eine beliebige Funktion $F : \{0,1\}^* \to \{0,1\}^k$

(e) $\hat{H}(x) := H(F(x))$ für eine beliebige Einwegfunktion $F : \{0,1\}^* \to \{0,1\}^k$

(f) $\hat{H}(x) := \begin{cases} x & \text{wenn } |x| = k \\ H(x) & \text{sonst} \end{cases}$

(g) $\hat{H}(x) := H(F(x))$ für eine beliebige injektive Funktion $F : \{0,1\}^* \to \{0,1\}^*$

Dabei bezeichne $||$ die Konkatenation von Bitstrings und $\oplus$ Verknüpfung durch bitweises XOR.

Aufgabe 2. Aus der Vorlesung ist bekannt, dass mithilfe einer kollisionsresistenten Kompressionsfunktion $\mathsf{F} : \{0,1\}^{2k} \to \{0,1\}^k$, mit $k \in \mathbb{N}$, eine kollisionsresistente Hashfunktion H_{MD} mittels des Merkle-Damgård-Verfahrens konstruiert werden kann. Eine mögliche Realisierung funkioniert wie folgt: Bei Eingabe von $M \in \{0,1\}^*$, mit $|M| = L < 2^k$, führe folgenden Algorithmus aus:

1. Setze $B := \lceil \frac{L}{k} \rceil$. (Dieses entspricht der Anzahl der Blöcke von M der Länge k.) Füge an das Ende der Nachricht M so viele Nullen an bis die Länge ein Vielfaches von k ergibt. Somit kann die Nachricht in die einzelnen k-Bit-Blöcke $M_1, \ldots, M_B$ aufgeteilt werden.
2. Setze $M_{B+1} := L$, wobei L mit exakt k Bits codiert wird.
3. Setze $Z_0 := 0^k$.
4. Für $i = 1, \ldots, B+1$ berechne $Z_i := \mathsf{F}(Z_{i-1}, M_i)$.
5. Gib Z_{B+1} aus.

Somit erhalten wir den Hashwert $\mathsf{H}_{\mathsf{MD}}(M) := Z_{B+1}$. Die Länge der Nachricht fließt also in den Hashwert ein. Sei nun $\mathsf{H}'_{\mathsf{MD}}$ eine Variante, die Z_B statt Z_{B+1} ausgibt. Geben Sie eine Kollision für $\mathsf{H}'_{\mathsf{MD}}$ an.

Betrachten wir nun nur Nachrichten, deren Länge ein Vielfaches der Blocklänge k ist. Lässt sich die Sicherheit von $\mathsf{H}'_{\mathsf{MD}}$ unter dieser Einschränkung für alle kollisionsresistenten Kompressionsfunktion F beweisen?

Aufgabe 3. Im ersten Teil der RSA-Schlüsselgenerierung wurden für einen Benutzer A die Primzahlen $P = 23$ und $Q = 11$ gezogen.

(a) Setzen Sie die RSA-Schlüsselgenerierung fort. Berechnen Sie einen zu P und Q gehörigen öffentlichen RSA-Schlüssel $pk_A = (N, e)$ und einen privaten RSA-Schlüssel $sk_A = (N, d)$.
Hinweis: Führen Sie den erweiterten euklidischen Algorithmus zur Übung von Hand durch.

(b) Senden Sie die Nachricht $M = 17$ RSA-verschlüsselt an Benutzer A. Benutzen Sie dazu die Lehrbuch-Variante von RSA (ohne Padding) und den öffentlichen Schlüssel aus (a).

(c) Nehmen wir an, Sie seien Benutzer A. Entschlüsseln Sie das empfangene Chiffrat aus (b).

Aufgabe 4. Auf der Webseite zur Vorlesung finden Sie die Datei `Sicherheit_UE03_Moduli.zip`, die 50.000 512-Bit RSA-Moduli enthält, von denen einige nicht vernünftig generiert wurden. Wie viele Moduli können Sie faktorisieren?

Hinweis: Mit Python können Sie die Liste beispielsweise wie folgt laden:

```
f = open('Sicherheit_UE03_Moduli.txt', 'r')
moduli = eval(f.read())
f.close()
```