# Computersystemsicherheit – Übungsblatt Nr. 3

Marc Fischlin, Jacqueline Brendel, Christian Janson
TU Darmstadt, 23 November 2018

**Gruppenübung.** Die Übungsaufgaben in diesem Bereich sind Gegenstand der Übungen in der Woche vom 26.11.2018 – 30.11.2018.

**Aufgabe 1 (Verständnisaufgaben).** In dieser Aufgabe prüfen wir unser Verständnis über den Inhalt der Vorlesung.

a) Erinnern Sie sich an die Schutzziele aus der ersten Vorlesung. Welches Ziel wird durch den Einsatz einer Signatur erreicht?

b) Wofür braucht man Signaturen? (Nennen Sie ein Beispiel aus der Praxis)

c) Welcher Teil des Schlüsselpaars sollte geheim bleiben und warum? Erklären Sie den Unterschied zwischen Verschlüsselung und digitalen Signaturen.

d) Welche Sicherheitseigenschaft sollen Signaturen intuitiv erreichen?

e) Erinnern Sie sich an das Hash-and-Sign Prinzip. Formalisieren Sie es.

f) Basiert jedes Signaturverfahren auf diesem Prinzip?

g) Was ist eine Hashfunktion?

h) Was ist eine Kollision?

i) Besitzen Hashfunktionen Kollisionen?

j) Was bedeutet es wenn wir sagen eine Hashfunktion ist kollisionsresistent?

k) Nennen Sie eine kollisionsresistente Hashfunktion aus der Praxis.

l) Wie und warum werden Hashfunktionen bei der Erzeugung von digitalen Signaturen eingesetzt?

**Aufgabe 2 (Hashfunktionen).** Es stehen folgende zwei Kandidaten für Hashfunktionen zur Verfügung:

$$H_1(n) = \lfloor 2^n \sin(\frac{1}{n}) \rfloor \quad \text{und} \quad H_2(n) = n^{n \bmod \varphi(99)} \bmod 99$$

Für beide Hashfunktionen können beliebige Zahlen aus $\mathbb{N}$ als Input verwendet werden.

a) Berechnen Sie für beide Kandidaten die Werte $n = 10, 11, 12$ aus.

b) Bewerten Sie beide Kandidaten bzgl. der Aspekte **Kompression**, **Umkehrresistenz**[1] und **Kollisionsresistenz**.
Sind die Kandidaten als kryptographische Hashfunktion nutzbar?

**Aufgabe 3 (RSA-Signaturen).** a) Erinnern Sie sich an das RSA-Signaturverfahren aus der Vorlesung und formalisieren Sie, wie mit einem gegebenen Schlüsselpaar die Signatur zu einer Nachricht $m$ erstellt wird. Wie kann diese Signatur verfiziert werden?

b) In der Vorlesung haben Sie gelernt, dass der Hashwert der Nachricht noch kodiert wird auf RSA-Länge. Sichere Optionen für die Kodierung können vom BSI in Erfahrung gebracht werden. Im Folgenden werden wir einfachheitshalber die Kodierung nicht weiter betrachten. Gegeben ist folgendes RSA-Schlüsselpaar mit $pk = (e, N) = (103, 143)$ und $sk = (d, N) = (7, 143)$. Signieren Sie die Nachricht $m = 7$ mit dem Schlüsselpaar unter der Hashfunktion $H\colon \{0,1\}^* \to \{0,1\}^3$ mit $x \longmapsto x \cdot 2 \bmod 8$.

c) In Aufgabe 5 der Gruppenübung von Übungsblatt 2 zum Thema Diffie-Hellman, Aufgabenteil d) haben Sie einen Angriff auf das Diffie-Hellman Verfahren kennen gelernt, der wegen mangelnder Authentifikation auftritt. Nehmen Sie an, dass Alice und Bob jeweils einen vertrauenswürdigen öffentlichen RSA-Public Key des jeweils anderen haben und erweitern Sie das Protokoll mit Ihrem neuen Wissen über digitale Signaturen.

d) Wenn bei dem RSA-Signaturverfahren keine Hashfunktion verwendet wird, die Message also einfach mit dem geheimen Exponenten $d$ potenziert wird und modulo $N$ reduziert wird, sind Angriffe auf das Verfahren möglich. Zum Beispiel kann man aus zwei gültigen Signaturen $s_1$ und $s_2$ ohne Kenntnis des geheimen Exponenten $d$ eine gültige Signatur für eine dritte Nachricht $m$ fälschen, die gar nicht unterschrieben worden ist. Demonstrieren Sie dies an einem Beispiel.

**Aufgabe 4 (DSA-Signaturen).** Der Assistent von Computersystemsicherheit signiert die Listen der Klausurnoten, bevor er diese an das Prüfungssekretariat schickt. Dazu verwendet er DSA-Signaturen. Allerdings besitzt er keinen guten Zufallsgenerator und beschließt daher, statt zufällige $k$-Werte zu verwenden, diese nach einem System zu bestimmen. Zufälligerweise erfährt ein Student, dass der Assistent für jede Nachricht das vorherige $k$ jeweils um eins erhöht und für die nächste Nachricht also den Wert $k + 1$ benutzt. Dieser Student ist ein Freund von Ihnen und gibt ihnen weitere Information über die verwendeten Parameter. Ihr Ziel ist es nun daraus den geheimen Schlüssel $x$ zu berechnen und im Anschluss eine korrekt signierte Liste von Klausurnoten an das Prüfungssekretariat zu schicken.

a) Erinnern Sie sich an das DSA-Signaturverfahren aus der Vorlesung und formalisieren Sie, wie mit einem gegebenen Schlüsselpaar die Signatur zu einer Nachricht $m$ erstellt wird. Wie kann diese Signatur verfiziert werden?

b) Gegeben seien $g = 72$, $p = 103$ und $q = 17$. Weiter kennen Sie die Hashwerte zweiter Notenlisten $m_1, m_2$ und die daraus berechneten Signaturen $S_1, S_2$ mit jeweils $H(m_1) = 5, S_1 = (r_1, s_1) = (0, 11)$ und $H(m_2) = 1, S_2 = (r_2, s_2) = (11, 9)$. Sie wissen, dass für $S_1$ ein zufälliges $k$ verwendet wurde und für $S_2$ ein $k + 1$. Berechnen Sie $k$ und $x$.

[1] D.h., ob es möglich ist ein Urbild von einem gegebenem Wert zu finden.

c) Nachdem Sie nun $k$ und $x$ kennen, können Sie selbst, ohne Verdacht zu schöpfen, eine andere Notenliste signieren. Signieren Sie die Notenliste $m$ mit $H(m) = 2$ und verwenden Sie als zufälligen $k$-Wert jetzt $k + 2$. Berechnen Sie dafür die DSA-Signatur $S = (r, s)$.

**Aufgabe 5 (Pseudozufallsgenerator).** Im Folgenden betrachten wir Pseudozufallsgeneratoren.

Gegeben seien folgende Java Klassen die einen PRNG darstellen sollen:

```
public class Random implements Runnable {
public int random = 0;
public void run() {
for (int i = 0; i < 256; i = (i + 1) % 256) {
random = i;
}
}
}
```

```
import java.util.Scanner;
public class Main{
public static void main(String[] args) {
Random random = new Random();
Thread myT = new Thread(random);
myT.start();
Scanner scan = new Scanner(System.in);
System.out.println("Press enter to get a random number");
while(scan.nextLine().length() == 0){ //exit if not enter
System.out.println(random.random);
}
myT.stop();
}
}
```

Threads laufen unabhängig vom restlichen Programm ab.

a) Diskutieren Sie wie die Ausgabe des Programms aussieht.

b) Woher kommt die Entropie in diesem Algorithmus?

c) Handelt es sich um einen guten PRNG?

---

**Hausübung.** Dieser Bereich ist dazu gedacht das Gelernte weiter zu vertiefen. Dazu werden je nach Themen weitere Übungsaufgaben, ergänzende Beweise oder ähnliche Aufgaben gestellt. Die Aufgaben sind freiwillig, können aber, bei erfolgreicher Bearbeitung, zu Bonuspunkten in der Klausur führen. Die Abgabe dieser Übungen erfolgt über **moodle** und kann in Gruppen mit bis zu vier Studenten (aus Ihrer **eigenen** Übungsgruppe) eingereicht werden. Abgaben werden nur als **.pdf-Dateien** akzeptiert. Denken Sie bitte daran, dass Ihre Lösungen nachvollziehbar und entsprechend ausführlich dargestellt werden sollen.

Für Gruppenabgaben ist folgendes zu beachten: Sie müssen in der Abgabe (.pdf-Datei) deutlich und eindeutig kennzeichnen mit welchen Gruppenpartnern die Aufgaben gelöst wurden.

Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Falls dieser Fall eintritt, behalten wir uns das Recht vor für diese Abgabe den jeweiligen Gruppen keine Punkte gutzuschreiben.

Bitte reichen Sie Ihre Abgabe bis **spätestens Freitag 07.12.2018 um 11:40 Uhr** ein. Verspätete Abgaben können **nicht** berücksichtigt werden.

---

**Hausübung 1 (Replay-Attacke (2 Punkte)).** Eine Bank möchte in ihrem Online-Banking-Verfahren auf digitale Signaturen zurückgreifen. Jedoch fürchtet die Bank, dass ein Angreifer eine mitgehörte Nachricht zeitversetzt ein zweites Mal an die Bank schickt (z.B. für eine abermalige Kreditierung eines Überweisungsbetrages, falls der Empfänger mit dem Angreifer in Verbindung steht). Hierbei handelt es sich um eine sogenannte Replay-Attacke.
Das Management der Bank überlegt sich, dass sie sich gegen eine solche Attacke schützen könnte, falls sie alle Nachrichten bzw. deren Hashwerte speichern würde, und neue Nachrichten mit den bereits empfangen (und damit gespeicherten) vergleicht. Dies ist jedoch in der Praxis enorm aufwendig.
Finden und beschreiben Sie ein alternatives Verfahren.

**Hausübung 2 (RSA Signatur (2 + 1 Punkte)).** In dieser Aufgabe beschäftigen wir uns mit RSA-Signaturen.

a) Es sei $pk = (N, e) = (77, 43)$ ein öffentlicher RSA-Schlüssel. Berechnen Sie eine gültige RSA-Signatur der Nachricht $m = 14234$ mit Hashwert $H(m) = 17$.

b) Betrachten Sie folgenden öffentlichen RSA-Schlüssel $pk = (N, e) = (91, 23)$. Überprüfen Sie, ob $s = 62$ eine gültige RSA-Signatur der Nachricht $m = 7$ mit Hashwert $H(m) = 3$ ist.

**Hausübung 3 (DSA-Signatur (2 + 1 Punkte)).** Gegeben seien folgende Basisparameter $p = 29$, $q = 7$ und $g = 16$ für ein DSA-Signaturverfahren.

a) Alices geheimer Schlüssel $x = 4$ und damit signiert sie den Hashwert $H(m) = 6$ einer Nachricht $m$. Berechnen Sie Alices DSA-Signatur $S = (r, s)$. W"ahlen Sie dafür das "zufällige" $k = 2$.

b) Berechnen Sie Alices public key $pk = (y, p, q)$. Prüfen Sie die Signatur aus Teilaufgabe a) mit Hilfe des public key.

**Hausübung 4 (Zertifikate (0,5 + 0,5 + 1 Punkte)).** Öffnen Sie im Webrowser Ihres Vertrauens die Webseite `https://www.tu-darmstadt.de` und finden Sie selbstständig heraus wie Sie sich das Zertikat anzeigen lassen können.

a) Wann läuft das Zertifiakt ab? Wie lautet der SHA-1 Fingerabdruck des Zertifikats?

b) Betrachten Sie nun das Root-Zertifikat an der Wurzel der Zertifizierungshierarchie. Von welcher Firma wurde es ausgestellt?

c) Begründen Sie intuitiv warum Zertifikate ein Ablaufdatum besitzen?