IT-Sicherheit
Prof. Dr. Michael Zohner
Wintersemester 2021/2022

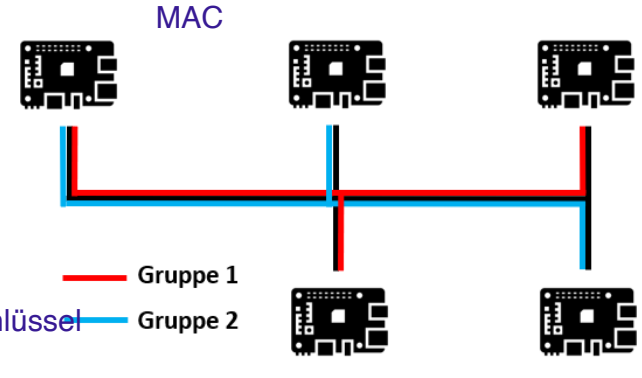
Hochschule Fulda
University of Applied Sciences

Übung 6: MACs und Zufallszahlengeneratoren v1.1*

Aufgabe 1 Gegeben seien die folgenden beiden Anwendungsfälle für Softwareupdates und Gruppenkommunikation:

Figure 1: Software Updates sollen an Endgeräte verteilt werden wobei die nutzenden Endgeräte stark fluktuieren.

Figure 2: Gruppenkommunikation einer statischen Rechnerarchitektur (z.B. im Fahrzeug) bei denen die Geräte über die Lebenszeit gleich bleiben.



Erstellen Sie für beide Anwendungsfälle je ein Sicherheitskonzept um das Sicherheitsziel Authentizität für (1) Software Updates sowie (2) Kommunikation zu erfüllen. Das Sicherheitskonzept sollte die folgenden Fragen beantworten:

A) Welches kryptographische Verfahren (Digitale Signaturen oder Message Authentication Codes) sollte eingesetzt werden? Begründen Sie die Wahl des kryptographischen Verfahrens.

B) Welche Schlüssel müssten auf welchen Entitäten gespeichert werden? Server

C) Welche Sicherheitsziele müssen für welche Schlüssel gelten? öffnt schlüssel auf smartphone

Aufgabe 2 Bewerten Sie die Gleichverteilung und Unvorhersagbarkeit der folgenden vier Zufallszahlengeneratoren (RNGs). Falls nötig, schlagen Sie Verbesserungen vor um eine Gleichverteilung bzw. Unvorhersagbarkeit für die Zufallszahlengeneratoren zu erhalten.

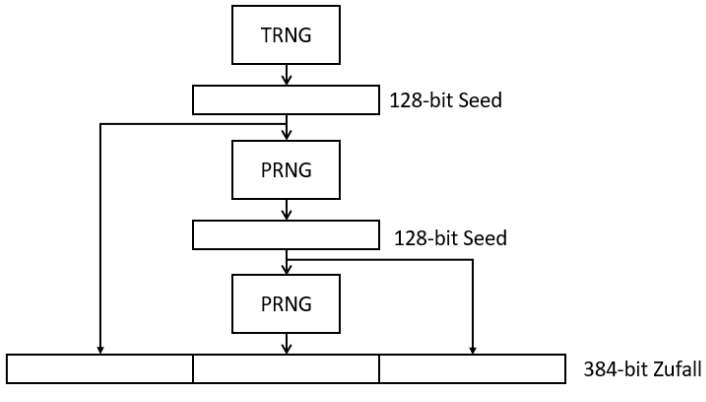
RNG 1 Zufälliger Würfelwurf 1-6.

1: $val = TRNG(3)$; //3 zufällige Bits
2: return $(val\%6) + 1$; //Wurf 1-6

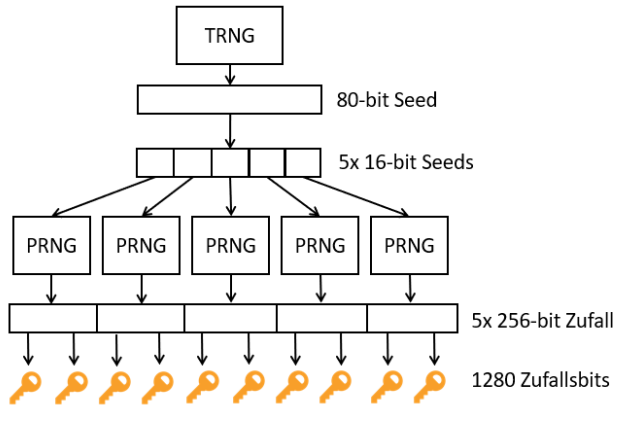
RNG 2 Zufälliger 128-bit Wert.

1: $PRNG.seed(time())$; //PRNG mit Zeit als Seed
2: return $PRNG(128)$; //128-bit aus dem PRNG

RNG 3: Zufälliger 384-bit Wert aus einem sequentiellen RNG.



RNG 4: Zufälliger 1280-bit Wert aus einem parallelen RNG.



Aufgabe 3 - Bonuspunkteaufgabe (TaskID 6A) Gegeben sei der folgende Zufallszahlengenerator:

RNG 3 Linearer Kongruenzgenerator

```
1: long long seed; //Updated intermediate seed
2: long long a, b; //Constants
3:
4: //Generiere 32 zufällige bits als unsigned integer
5: procedure UNSIGNED INT RAND_NEXTUINT()
6:     seed = (seed * a + b) & ((1L << 48) − 1);
7:     return (unsigned int) (seed >> 16)
```

Sie erhalten aus der Übungsumgebung die Konstanten a und b sowie zwei aufeinanderfolgende 32-bit Zufallszahlen $zufall_1$ und $zufall_2$ als unsigned Integer. Ihre Aufgabe ist es, den internen 48-bit Zustand *seed* des Zufallszahlengenerators nach der Ausgabe von $zufall_2$ zu berechnen. Laden Sie den Wert von *seed* als Dezimalzahl in das Moodle Element "Abgabe Übung 6: Zufallszahlengeneratoren" hoch.

Hinweise:

- Sie können den internen Zustand als Dezimalzahl mit der Aufgaben-ID "6AV" verifizieren (Antwort als OK/NOK).
- Die Abgabefrist ist der 13/Dezember, 23:59 Uhr. Ein erfolgreiches Bestehen wird via Moodle mitgeteilt.

***Versionshistorie** v1.0: Initialversion. v1.1: Bild zu Aufgabe 2, RNG 4 geupdated und Beschreibung für Bonusaufgabe präzisiert.