IT-Security Lab 3

Abgabegruppe 22 (Kai Werk, Jan Hinrichs)

Aufgabe 3.1 Theorie

A und B sind abgefangene Public Keys. Wir sind nun der Man in the Middle und jegliche Kommunikation geht über uns. Normalerweise werden A = g^a mod p und B = g^b mod p, wobei a und b die jeweiligen private keys sind, berechnet. Der gemeinsame Key wird bei den Nutzern jeweils durch k = g^a*b mod p bzw. k = g^b*a mod p berechnet. Wir als man in the middle berechnen zwei unterschiedliche gemeinsame keys K* und K~ mit b* und a~. Diese berechnen wir wie folgt, K~ = B**a~ und K* = A**b* .a~ und b* werden einfach von 1 an hochgezählt und ein Paar von a~ und Hash von K~ bzw. b* und Hash von K* in einer Tabelle gespeichert. Sobald wir eine ausreichend lange Hashkollision gefunden haben übergeben wir die jeweiligen a~ und b* an Bob und Alice weiter. Diesen Angriff nennt man den Geburtstagsangriff.

Abschätzung Speicher und Zeit:
Bei einer Erfolgswahrscheinlichkeit von p = ½ braucht man 1,18* sqrt(m) wobei m die Anzahl möglicher Hashwerte ist.

Aufgabe 3.2 Warm Up

Aufruf über:

```
./diffie.sh a b
```

Ausgabe erfolgt gemäß der Aufgabenstellung.

Aufgabe 3.3 Kollision

Aufruf über:

```
./collision.py A B
```

Hinweis: Es werden Kollisionen der Länge 10 erzeugt. Die Berechnungszeit für Kollisionen mit höherer Länge steigt enorm, wenngleich sie gefunden und berechnet werden. Um die Länge der gewünschten Kollision anzupassen, muss in Zeile 7 die Konstante COLLISION_LEN verändert werden.