Prof. Dr. Holger Schmidt
Professur für IT-Sicherheit, Informatik
Fachbereich Informatik – Fachhochschule Dortmund

Fachhochschule Dortmund
University of Applied Sciences and Arts

Aufgabe 8.1

a) Erzeugen Sie mit OpenSSL einen *privaten RSA-Schlüssel*. Der Schlüssel soll eine Größe von *1024 Bit* haben. Speichern Sie den Schlüssel in einer *Datei*. Benutzen Sie dazu OpenSSL und dokumentieren Sie den genutzten Befehl mit Parametern.

b) Recherchieren Sie: Wie lautet das *Format* in dem der Schlüssel gespeichert wird? In welchem *Standard* ist das Format spezifiziert?

c) Aus der Vorlesung ist Ihnen bekannt, dass der private Schlüssel aus dem Wertepaar n und d besteht. Ermitteln Sie diese *Werte* Ihres zuvor selbst erzeugten Schlüssels und dokumentieren Sie diese. Benutzen Sie dazu OpenSSL und dokumentieren Sie den genutzten Befehl mit Parametern.

d) Welche *weiteren Parameter* neben n und d werden in der Schlüsseldatei gespeichert? Geben Sie die *Namen* und *zugehörigen Werte* an. Benutzen Sie dazu OpenSSL und dokumentieren Sie den genutzten Befehl mit Parametern.

e) Recherchieren Sie: Erläutern Sie für die weiteren Parameter warum diese im privaten Schlüssel abgespeichert werden.

f) Erzeugen Sie mit OpenSSL aus Ihrem privaten Schlüssel den öffentlichen Schlüssel und lassen Sie sich die im öffentlichen Schlüssel enthaltenen Werte anzeigen. Wie lauten die *Namen* und *zugehörigen Werte*? Wie *unterscheiden* sich die gespeicherten Werte im privaten und öffentlichen Schlüssel? Benutzen Sie dazu OpenSSL und dokumentieren Sie den genutzten Befehl mit Parametern.

g) *Verschlüsseln* Sie nun einen kurzen Klartext und lassen Sie sich den Geheimtext anzeigen. Benutzen Sie dazu OpenSSL und dokumentieren Sie den genutzten Befehl mit Parametern.

h) Warum kann es bei zu *langen Nachrichten* zu einem Fehler kommen?

i) *Entschlüsseln* Sie nun den zuvor erzeugten Geheimtext. Benutzen Sie dazu OpenSSL und dokumentieren Sie den genutzten Befehl mit Parametern.

Aufgabe 8.2

Wenn RSA-Schlüssel zu klein gewählt sind, dann besteht die Möglichkeit eines Public-Key-Only Angriffs. Stellen Sie zunächst den privaten Schlüssel und dann damit den Klartext wieder her mithilfe der in ILIAS/sciebo zur Verfügung gestellten Datei `my.pub` mit dem öffentlichen Schlüssel. Weiterhin steht Ihnen ein noch von Ihnen zu vervollständigendes Python-Skript `calculatekey.py` zur Verfügung. Benutzen Sie außerdem die in der vorherigen Aufgabe gewonnenen Erkenntnisse über OpenSSL.

a) Wie *lang* ist der öffentliche RSA-Schlüssel?

b) Finden Sie einen Dienst oder ein Programm, welcher/s den Modulus des öffentlichen Schlüssels in *Primfaktoren zerlegt*. Notieren Sie die Primfaktoren. *Wie lange* hat die Zerlegung in Primfaktoren gedauert?

c) Basierend auf den Primfaktoren können Sie nun $\varphi(n)$ und anschließend mit dem erweiterten Euklidischen Algorithmus das *modulare Inverse* berechnen. Vervollständigen Sie das Python-Skript `calculatekey.py` entsprechend. Ergänzen Sie im Anschluss die bestehenden Kommentare und fügen Sie gegebenenfalls neue Kommentare hinzu, um die Funktionsweise des Skripts zu erklären.

d) Stellen Sie nun den *privaten RSA-Schlüssel* wieder her. Notieren Sie sich den Wert des Exponenten.

e) *Entschlüsseln* Sie mit dem wiederhergestellten privaten RSA-Schlüssel den in ILIAS/scibo zur Verfügung gestellten Geheimtext in der Datei `message`. Wie lautet der Klartext?

Aufgabe 8.3

2011 hat das NIST die Hashfunktion SHA-1 als veraltet eingestuft.

a) Recherchieren Sie und dokumentieren Sie Ihre Rechercheergebnisse mit Quellenangaben: Was ist der Stand der Technik bzgl. SHA-1 bei

   (i) Kollisionsangriffen,

   (ii) Urbild-Angriffen,

   (iii) Zweites-Urbild-Angriffen?

b) Spielen Sie Angreifer: Sie haben eine signierte Nachricht (m, sig) mit $sig = e_{RSA}(SHA-1(m), k_{priv})$ abgefangen. Sie möchten die Nachricht ändern, ohne dass der Empfänger es merkt, aber Sie können den Signaturalgorithmus selbst nicht kompromittieren.

   Ausgehend von den momentan möglichen Angriffen, wie kann der beschriebene Angriff auf die signierte Nachricht gelingen?

c) Was könnten Sie im Allgemeinen (also nicht nur auf digitale Signaturen abzielend) machen, wenn Ihnen Urbild-Angriffe gegen SHA-1 gelingen würden?