Aufgabe 1

- Grande dafür, dass sich E-Mail-Verschlüsselung noch nicht flächendeckend etabliert hat.
  - Arbeitsaufwand + hörere Kosten für Provider
  - langsamere Übertragung, die es manchen Kunden womöglich nicht wert ist
  - weit verbreitetes Desinteresse der Bevölkerung an Datensicherheit
- $S/MIME$ und PGP im Vergleich
  1. Der wesentlichste Unterschied zwischen PGP und S/MIME liegt im Bereich der Authentifizierung. Um die Authentizität des Absenders festzustellen, ist bei S/MIME ein Zertifikat notwendig, welches die Authentizität des Schlüssel-Besitzers unzweifelhaft beglaubigt. Das heisst, dass ich meinen Schlüssel von einer dazu berechtigten Organisation zertifizieren lassen muss, bevor er daruch wirklich nutzbar wird. Diese Organisation wurde wiederum von einer höher stehenden Organisaiton zertifiziert usw. bis man zu einem Wurzel-Zertifikat kommt. Vertaut man nun diesem Wurzel-Zertifkat, so vertaut man automatisch allen darunter liegenden Zertfizierungen. Das nennt man hierarchisches Vertrauenskonzept.
     Im Gegensatz dazu erlaubt PGP neben dieser baum-artigen Zertifizierung zusätzlich auch eine direkte „peer-to-peer“ Zertifizierung (Anwender A zertfiziert Anwender B, B zertifiziert A und C usw.) und macht damit aus einem Zertifizierungs-Baum ein Zertifizierungs-Netz, das sogenannte Web-of-Trust. Im Fall der direkten Authentifizierung bei PGP hat man also die Möglichkeit, ohne eine Zertifizierung einer höheren Stelle verschlüsselte Daten und E-Mails auszutauschen. Dafür reicht es aus, wenn man der E-Mail-Adresse und dem dazugehörigen Zertifikat seines Kommunikationspartners vertraut.
  2. Schutzziel: Verschlüsseln und Signieren von Daten, also Geheimhaltung und Authentizität (genauer s. Punkt 1).
  3. Bei beiden Systemen wird die kryptographische Methode des Public Keys verwendet, bei der mit einem öffentlichen Schlüssel Nachrichten an den Empfänger kodiert werden, die dieser dann mit seinem privaten Schlüssel dekodieren kann (genauer s. Punkt 1).
- Angriff auf PGP
  Eine der gefährlichsten Angriffsmöglichkeiten besteht darin, dass der Public Keys gefälscht und Nachrichten somit umgeleitet werden können. Schafft man es also, einen falschen Key zu verbreiten, spart man sich gewissermaßen jeden Angriff auf Dekodierungs-Ebene und kann das ganze Verfahren aushelben, indem man einfach den eigentlichen Empfänger und seinen sicher sehr komplexen Schlüssel umgeht.

Aufgabe 2

Ein Server benutzt zur verschlüsselten Kommunikation das RSA-Verfahren mit folgendem Public- Key: (N, e) = (622579, 21113)

- Die Primzahlen bzw. das Produkt dieser ist zu klein (= zu einfach zu knacken).
- $\sqrt{n} \approx 789$
  Durch systematisches Ausprobieren mit ungeraden Zahlen ab 789 abwärts erhalten wir: $p = 751$, $q = 829$
  $\Rightarrow \varphi(N) = (751 - 1) \cdot (829 - 1) = 621000$
  Finde d mit
  $d \cdot e \equiv 1 \bmod \varphi(N)$
  $\Leftrightarrow d = 11177$
- Entschlüsselung:
  $380157^{11177} \bmod N = 21369$
  $615426^{11177} \bmod N = 29556$
  $92340^{11177} \bmod N = 25965$
  $57197^{11177} \bmod N = 25906$
- Die zu verschlüsselnden Zahlen dürfen maximal 16 Bit groß sein.

Aufgabe 3

Sicherheitslücken

1 http
  Triviales Abhören von Daten, da diese nicht verschlüsselt sind.

2 https mit selbstsignierten Zertifikaten
  Auch diese Variante ist nur bedingt sicher, da sie mit einem MITM-Angriff leicht ausgehebelt werden kann, wenn der Angreifer (statt dem eigentlichen Empfänger), dem Sender ein gefälschtes, selbstsigniertes Zertifikat zukommen lässt. Fortan kann der Datenverkehr abgehört/umgeleitet werden.

3 https mit CA signierten Zertifikaten
  An dieser Stelle kann ein Angriff auf das Zertifikationssystem dem Angreifer Zugang verschaffen, beispielsweise wenn er eine Kollision im MD5-Algorithmus herbeiführt, wodurch er sich selbst authentische Zertifikate ausstellen kann.