Aufgaben

6.1. In diesem Kapitel wurde gezeigt, dass asymmetrische Algorithmen für den Schlüsselaustausch und digitale Signaturen eingesetzt werden können, was beides nicht mit symmetrischer Kryptografie möglich ist. Darüber hinaus kann man mit asymmetrischen Verfahren auch klassische Datenverschlüsselung durchführen.

Die Frage ist nun, warum für die Praxis nach wie vor symmetrische Kryptoverfahren benötigt werden?

6.2. In dieser Aufgabe wird der Rechenaufwand für symmetrische und asymmetrische Algorithmen verglichen. Wir nehmen an, eine Bibliothek wie z. B. OpenSSL kann Daten mit einer Geschwindigkeit von 100 Kbit/s mittels des RSA-Verfahrens auf einem PC dechiffrieren. Auf dem gleichen Rechner entschlüsselt AES mit einer Datenrate von 17 Mbit/s. Wir möchten nun einen 1 GByte großen Film, der auf einer DVD gespeichert ist, dechiffrieren. Wie lange dauert dies mit den beiden Kryptoverfahren?

6.3. Gegeben sei ein kleineres Unternehmen mit 120 Mitarbeitern. Es wird eine neue Sicherheitsrichtlinie eingeführt, nach der sämtliche E-Mail-Kommunikation mit symmetrische Kryptografie verschlüsselt werden muss. Wie viele Schlüssel werden in dem Unternehmen benötigt, wenn jeder Mitarbeiter mit jedem anderen sicher E-Mail austauschen sollen kann?

6.4. Das Sicherheitsniveau asymmetrischer Algorithmen kann erhöht werden, indem größere Bitlängen gewählt werden. Allerdings wirken sich die längeren Parameter direkt auf die Laufzeit der Algorithmen aus. Den Zusammenhang zwischen Ausführungszeit und Sicherheitsniveau wird in dieser Aufgabe untersucht.

Ein Web-Server für einen Online-Shop kann entweder RSA oder ECC für das Erstellen von digitalen Signaturen verwenden. Die Signaturzeit für RSA-1024 beträgt 15,7 ms und für ECC 1,3 ms.

1. Wie hoch ist die Laufzeit für eine RSA-Signatur, wenn die Bitlänge aus Sicherheitsgründen von 1024 Bit auf 3072 Bit erhöht wird?
2. Wie erhöht sich die Laufzeit für RSA bei einer Erhöhung von 1024 Bit auf 15.360 Bit?
3. Berechnen Sie die Laufzeiten für ECC, wenn ECC das gleiche Sicherheitsniveau wie RSA-3072 und RSA-15.360 bieten soll.
4. Beschreiben Sie das unterschiedliche Verhalten von RSA und ECC wenn das Sicherheitsniveau erhöht wird.

Hinweis: Die Rechenlaufzeit von RSA und ECC wächst kubisch mit der Bitlänge. Tabelle Table 6.1 gibt die Sicherheitsniveaus von ECC und RSA an.

6.5. Verwenden Sie die den euklidischen Algorithmus (aber nicht den erweiterten euklidischen Algorithmus), um den größten gemeinsamen Teiler der folgenden Zahlenpaare zu bestimmen.

1. 7469 und 2464

2. 2689 und 4001

Verwenden Sie lediglich einen Taschenrechner. Zeigen Sie jede Iteration des Algorithmus. Zeigen Sie ebenfalls die Kette der ggT, die berechnet werden, in der folgenden Form:

$$\gcd(r_0, r_1) = \gcd(r_1, r_2) = \cdots$$

6.6. Verwenden Sie den erweiterten euklidischen Algorithmus um den ggT sowie die Koeffizienten s,t der folgenden Zahlenpaare zu berechnen.

1. 198 und 243
2. 1819 und 3587

Verifizieren Sie jeweils ob die Gleichung $s\,r_0 + t\,r_1 = \gcd(r_0, r_1)$ erfüllt ist. Zeigen Sie alle Zwischenergebnisse in jeder Iteration des Algorithmus.

6.7. Mit dem EEA steht uns (endlich) ein Verfahren zur Berechnung der multiplikativen Inversen in Z_m zur Verfügung, das effizient ist. Bestimmen Sie die Inversen a^{-1} modulo m:

1. $a = 7$, $m = 26$
   (Diese Inverse wurde für in Aufgabe 1.11 in Kapitel 1 für die affine Chiffre benötigt.)
2. $a = 19$, $m = 999$

Man beachte, dass die Inversen auch wieder in Z_m liegen müssen und dass man die Korrektheit der gefundenen Lösung durch einfaches Multiplizieren überprüfen kann.

6.8. Bestimmen Sie $\phi(m)$ für $m = 12, 15, 26$ unter Verwendung der Definition der eulersche Phi-Funktion: Überprüfen Sie für jede positiven ganze Zahl n, die kleiner m ist, ob $\gcd(n,m) = 1$ gilt. (Für die kleinen Zahlen, die hier auftreten, muss der euklidische Algorithmus nicht verwendet werden.)

6.9. Entwickeln Sie eine Formel für $\phi(m)$ für die beiden Spezialfälle:

1. m ist eine Primzahl
2. $m = p \cdot q$, wobei p und q Primzahlen sind. Dieser Fall ist für das RSA-Kryptoverfahren sehr wichtig. Verifizieren Sie die gefundene Formel für $m = 15$ und $m = 26$ mit den Lösungen von der vorherigen Aufgabe.

6.10. Berechnen Sie die Inverse a^{-1} mod n mittels des kleinen fermatschen Satzes bzw. mit dem Satz von Euler:

- $a = 4$, $n = 7$
- $a = 5$, $n = 12$
- $a = 6$, $n = 13$

6.11. Verifizieren Sie den Satz von Euler in Z_m für $m = 6, 9$ für alle Elemente a, für die $\gcd(a,m) = 1$ gilt. Zeigen Sie auch, dass der Satz nicht für Elemente a mit $\gcd(a,m) \neq 1$ gilt.

6.12. Die multiplikative Inverse für die affine Chiffre in Abschnitt 1.4.4 kann berechnet werden als

$$a^{-1} \equiv a^{11} \bmod 26.$$

Zeigen Sie die Herleitung dieses Ausdrucks unter Verwendung des eulerschen Satzes.

6.13. Der erweiterte euklidische Algorithmus hat die Anfangswerte $s_0 = 1, s_1 = 0, t_0 = 0, t_1 = 1$. Leiten Sie diese Initialwerte her. Es ist hilfreich, wenn man sich hierfür anschaut, wie die allgemeine Iterationsformel für den EEA hergeleitet wurde.