Aufgaben

7.1. Gegeben sei RSA mit den beiden Primzahlen $p = 41$ und $q = 17$.

1. Welcher der beiden Parameter $e_1 = 32$ und $e_2 = 49$ ist ein gültiger RSA-Exponent? Begründen Sie die Antwort.
2. Berechnen Sie den privaten Schlüssel $k_{pr} = (p,q,d)$ unter Verwendung des erweiterten euklidischen Algorithmus. Zeigen Sie alle Zwischenschritte der Berechnung.

7.2. Die effiziente Berechnung von Potenzen ist wichtig, um RSA in der Praxis einsetzen zu können. Berechnen Sie die folgenden Potenzen $x^e \bmod m$ mittels des Square-and-Multiply-Algorithmus:

1. $x = 2$, $e = 79$, $m = 101$
2. $x = 3$, $e = 197$, $m = 101$

Zeigen Sie nach jeder Iteration des Algorithmus den Exponenten in Binärdarstellung.

7.3. Führen Sie eine Ent- und Verschlüsselung mit RSA mit den folgenden Systemparametern durch:

1. $p = 3$, $q = 11$, $d = 7$, $x = 5$
2. $p = 5$, $q = 11$, $e = 3$, $x = 9$

Verwenden Sie nur einen Taschenrechner und zeigen Sie alle Zwischenschritte.

7.4. Ein Nachteil aller asymmetrischen Kryptoverfahren ist, dass sie vergleichsweise langsam sind. In Abschnitt 7.5.1 wurde gezeigt, wie RSA durch die Verwendung kurzer öffentlicher Schlüssel e beschleunigt werden kann. Wir untersuchen diesen Ansatz in dieser Aufgabe.

1. In einer gegebenen RSA-Implementierung benötigt eine modulare Quadrierung 75 % der Zeit einer modularen Multiplikation. Wie viel schneller ist eine Verschlüsselung im Durchschnitt, wenn anstatt eines öffentlichen Schlüssels mit 2048 Bit der kurze Exponent $e = 2^{16} + 1$ benutzt wird? In beiden Fällen wird der Square-and-Multiply-Algorithmus eingesetzt.
2. Die meisten kurzen Exponenten haben die Form $e = 2^n + 1$. Wäre es vorteilhaft, Exponenten der Form $2^n - 1$ zu benutzen? Begründen Sie die Antwort.
3. Berechnen Sie $x^e \bmod 29$ für $x = 5$ mit den beiden Varianten für e von oben, d. h. für $e = 2^n + 1$ und $2^n - 1$. Verwenden Sie den Square-and-Multiply-Algorithmus und zeigen Sie alle Zwischenschritte.

7.5. In der Praxis werden oft die kurzen öffentlichen Exponenten $e = 3$, 17 und $2^{16} + 1$ verwendet.

1. Warum kann man diese drei Werte nicht für den Exponenten d benutzen, wenn man die Entschlüsselung beschleunigen möchte?

2. Schlagen Sie eine untere Grenze für die Bitlänge des Exponenten d vor und begründen Sie die Antwort.

7.6. Verifizieren Sie RSA mit dem chinesischen Restsatz aus Beispiel 7.6, indem Sie $y^d = 15^{103} \bmod 143$ mit dem Square-and-Multiply-Algorithmus berechnen.

7.7. Eine RSA-Dechiffrierung hat die Parameter $p = 31$ und $q = 37$. Der öffentliche Schlüssel ist $e = 17$.

1. Entschlüsseln Sie das Chiffrat $y = 2$ unter Verwendung des chinesischen Restsatzes (CRT).
2. Verifizieren Sie die Antwort durch Entschlüsselung des Klartextes ohne Verwendung des CRT.

7.8. In der Praxis werden sehr oft RSA-Moduln mit einer Länge von 1024, 2048, 3072 und 4092 Bit verwendet.

1. Wie viele zufällige ungerade natürliche Zahlen muss man durchschnittlich testen, bevor man eine Primzahl findet?
2. Leiten Sie hierfür einen einfachen Ausdruck für beliebige Bitlängen her.

7.9. Eine der Hauptanwendungen der asymmetrischen Kryptografie ist der Austausch eines geheimen Sitzungsschlüssels über einen unsicheren Kanal, welcher dann für symmetrische Chiffren wie AES verwendet werden kann.

Bob hat den öffentlichen und privaten Schlüssel eines RSA-Kryptosystems. Zeigen Sie ein einfaches Protokoll, mit dem Alice und Bob nun ein gemeinsames Geheimnis austauschen können. Wer bestimmt in diesem Protokoll den Wert des Geheimnisses: Alice, Bob oder beide gemeinsam?

7.10. In der Praxis ist es manchmal wünschenswert, dass beide Parteien den Wert des gemeinsamen Sitzungsschlüssels beeinflussen. Dies verhindert z. B., dass eine der Parteien absichtlich einen schwachen Schlüssel (weak key) für die symmetrische Chiffre wählt. Manche Blockchiffren wie DES oder IDEA haben schwache Schlüssel, vgl. Aufgabe 3.7. Nachrichten, die mit einem solchen Schlüssel chiffriert werden, können von einem Angreifer leicht gebrochen werden.

Entwickeln Sie ein Protokoll ähnlich zu dem in Aufgabe 7.9, in dem beide Teilnehmer den Wert des auszuhandelnden Schlüssels beeinflussen. Nehmen Sie an, dass sowohl Alice als auch Bob ein RSA-System mit gültigem öffentlichen und privaten Schlüssel aufgesetzt haben. Es gibt verschiedene Möglichkeiten, dieses Problem zu lösen, zeigen Sie eine davon.

7.11. Das Ziel ist es, eine mit RSA verschlüsselte Nachricht zu brechen. Der Angreifer sieht auf dem Kanal das Chiffrat $y = 1141$. Der öffentliche Schlüssel ist $k_{pub} = (n, e) = (2623, 2111)$.

1. Wir schauen uns zunächst die RSA-Verschlüsselungsgleichung an. Alle Variablen mit Ausnahme von x sind bekannt. Warum kann man die Gleichung nicht einfach für x lösen?

2. Um den privaten Schlüssel d zu erhalten, muss der Ausdruck $d \equiv e^{-1} \bmod \Phi(n)$ bestimmt werden. Es gibt einen effizienten Weg, $\Phi(n)$ zu bestimmen. Können wir diese Formel hier anwenden?
3. Bestimmen Sie den Klartext x, indem zunächst der private Schlüssel d durch Faktorisierung von $n = p \cdot q$ bestimmt wird. Ist dieser Ansatz auch machbar, wenn der Modul 1024 Bit oder länger ist?

7.12. In dieser Aufgabe wird gezeigt, wie RSA mittels eines Angriffs mit gewähltem Chiffrat (chosen ciphertext attack) gebrochen werden kann.

1. Zeigen Sie, dass RSA die *multiplikative Eigenschaft* besitzt, d. h. das Produkt zweier Chiffrate ist gleich der Verschlüsselung des Produkts der beiden entsprechenden Klartexte.
2. Unter bestimmten Voraussetzungen kann diese Eigenschaft für einen Angriff ausgenutzt werden. Zunächst empfängt Bob von Alice das Chiffrat y_1, welches Oskar auf dem Kanal mitschneidet. Oskars Ziel ist es, x_1 zu erhalten. Wir nehmen an, dass Oskar zu einem späteren Zeitpunkt ein Chiffrat y_2 an Bob schicken kann und dass Oskar die Entschlüsselung von y_2 kennt, indem er sich kurzzeitig in Bobs Rechensystem hackt. Wie kann Oskar y_2 konstruieren, so dass er x_1 berechnen kann?

7.13. In dieser Aufgabe untersuchen wir die Probleme, die deterministische Kryptosysteme wie das Schulbuch-RSA-Verfahren besitzen. Im Gegensatz zu probabilistischen Kryptosystemen bilden deterministische Verfahren denselben Klartext immer auf dasselbe Chiffrat ab.

Ein Angreifer kann durch Beobachtung der Chiffrate Rückschlüsse auf den Klartext ziehen, beispielsweise kann er erkennen, wenn Klartexte wiederholt gesendet werden. Manchmal können deterministische Kryptoverfahren sogar vollständig gebrochen werden. Dies gilt insbesondere, wenn die Anzahl der möglichen Klartexte klein ist. Wir nehmen die folgende Situation an:

Alice sendet einen Text an Bob, der mit seinem öffentlichen Schlüssel (n,e) chiffriert ist. Sie verschlüsselt jeden Klartextbuchstaben einzeln, wobei jedes Klartextzeichen mit dem entsprechenden ASCII-Wert codiert wird (Leerzeichen $\rightarrow 32$, $! \rightarrow 33, \ldots, A \rightarrow 65, B \rightarrow 66, \ldots, \sim \rightarrow 126$).

1. Oskar hat Zugriff auf den Übertragungskanal und kann die Chiffrate mitschneiden. Beschreiben Sie, wie er die Nachricht dechiffrieren kann.
2. Bobs öffentlicher RSA-Schlüssel ist $(n,e) = (3763, 11)$. Dechiffrieren Sie den Geheimtext

$$y = 2514, 1125, 333, 3696, 2514, 2929, 3368, 2514$$

   mit dem vorgeschlagenen Angriff. Wir nehmen an, dass Alice nur Großbuchstaben A–Z verschlüsselt.
3. Ist der Angriff auch möglich, wenn Alice OAEP-Padding verwendet? Begründen Sie die Antwort.

7.14. In den vier Jahrzehnten seit RSA vorschlagen wurde wurde der Modul n immer länger gewählt, um neuen Faktorisierungsangriffen zu widerstehen. Wie zu erwarten verlängert sich die Laufzeit von RSA (und den meisten anderen asymmetrischen Kryptoverfahren), wenn längere Parameter gewählt werden. In dieser Aufgabe untersuchen wir den Zusammenhang zwischen Laufzeit und Länge des Moduls. Die Laufzeit von RSA wird durch die Geschwindigkeit für eine modulare Exponentiation bestimmt.

1. Wir nehmen an, dass eine Modulo-n-Multiplikation oder eine Quadrierung $c \cdot k^2$ Taktzyklen benötigen, wobei k die Bitlänge des Moduls und c eine Konstante ist. Um welchen Faktor ist eine RSA-Ver- oder -Entschlüsselung mit 1024-Bit-Modul langsamer als mit einem 512-Bit-Modul, wenn der Square-and-Multiply-Algorithmus angewendet wird? Wir betrachten nur die reine Ver- bzw. Entschlüsselung und nicht die Schlüsselerzeugung oder das Padding. Des Weiteren nehmen wir Exponenten mit voller Länge an.
2. Ein verbreiteter Algorithmus, um die Multiplikation und Quadrierung mit großen Zahlen zu beschleunigen, ist der Karatsuba-Algorithmus, dessen Laufzeit proportional zu $k^{\log_2 3}$ ist. Wir nehmen an, dass der Karatsuba-Algorithmus $c' \cdot k^{\log_2 3} = c' \cdot k^{1{,}585}$ Taktzyklen für eine Multiplikation oder Quadrierung benötigt, wobei c' eine Konstante ist. Was ist das Verhältnis der Laufzeiten für RSA mit 1024 Bit und 512 Bit? Wir nehmen wiederum Exponenten mit voller Länge an.

7.15. (Anspruchsvolle Aufgabe!) Es gibt Möglichkeiten, den Square-and-Multiply-Algorithmus zu beschleunigen. Die Anzahl der Quadrierungen kann zwar nicht ohne weiteres reduziert werden, allerdings gibt es verbesserte Algorithmen, die weniger Multiplikationen benötigen. Das Ziel ist es nun, einen solchen Algorithmus zu entwerfen, der weniger Multiplikationen benötigt. Beschreiben Sie genau, wie der Algorithmus abläuft und wie viele Operationen (Multiplikationen bzw. Quadrierungen) er benötigt.
Hinweis: Versuchen Sie, den Square-and-Multiply-Algorithmus so zu verallgemeinern, dass er pro Iteration mehr als nur ein Bit des Exponenten verarbeitet. Die Grundidee ist, dass in jeder Iteration k Bits (z. B. $k = 3$) des Exponenten betrachtet werden.

7.16. In dieser Aufgabe betrachten wir Seitenkanalangriffe auf RSA. In einer RSA-Implementierung, die nicht gegen Seitenkanalangriffe geschützt ist, kann man aus der Beobachtung des Stromverbrauchs, der bei der RSA-Entschlüsselung auftritt, den privaten Schlüssel extrahieren. Abbildung 7.5 zeigt die Stromverbrauchskurve eines Mikrocontrollers, der den Square-and-Multiply-Algorithmus ausführt. Die Bereiche mit hoher Aktivität sind die Zeiten, während der der Mikrocontroller eine Quadrierung oder Multiplikation berechnet. Wir erinnern uns daran, dass eine einzelne Quadrierung bzw.-multiplikation einer Langzahl aus Millionen von Instruktionen besteht, vgl. Abschnitt 7.9. Durch die kurzen Intervalle mit wenig Aktivität erkennt der Angreifer das Ende bzw. den Anfang einer neuen Operation. Für den Angriff ist es wichtig, dass man zwischen Quadrierung (kürzere Operation) und Multiplikation (längere Operation) unterscheiden kann.

1. Identifizieren Sie die Iterationen des Square-and-Multiply-Algorithmus und markieren Sie Iterationen, die nur aus einer Quadrierung bestehen, mit *S* und solche, die aus einer Quadrierung gefolgt von einer Multiplikation bestehen, mit *SM*.
2. Wie beim Square-and-Multiply-Algorithmus üblich wird der Exponent von links nach rechts verarbeitet. Was ist der Wert des privaten Schlüssels d?
3. Der Schlüssel gehört zu einem RSA-System mit den Parametern $p = 67$, $q = 103$ und $e = 257$. Verifizieren Sie die Antwort. (Man beachte, dass ein Angreifer in der Praxis die Werte von p und q natürlich nicht kennt.)

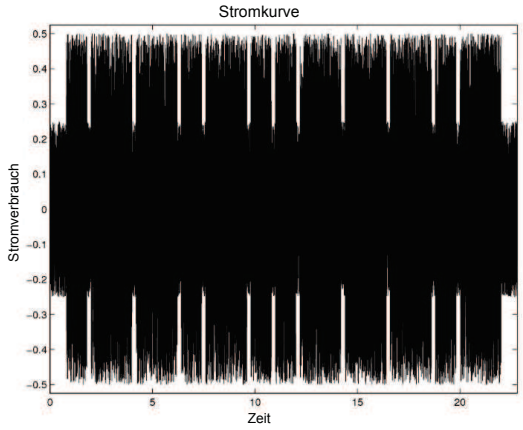

Abb. 7.5 Stromverbrauchskurve einer RSA-Entschlüsselung