Aufgaben

8.1. Ein Verständnis von Gruppen, zyklischen Gruppen und Untergruppen ist wichtig für die Verwendung von asymmetrischen Kryptosystemen basierend auf dem diskreten Logarithmusproblem. In dieser und den folgenden Aufgaben machen wir uns mit der Arithmetik in derartigen algebraischen Strukturen vertraut.

Bestimmen Sie die Ordnung aller Elemente der folgenden multiplikativen Gruppen:

1. $\mathbb{Z}_5^*$
2. $\mathbb{Z}_7^*$
3. $\mathbb{Z}_{13}^*$

Erzeugen Sie jeweils eine Liste mit zwei Spalten, wobei jede Zeile ein Element a und dessen Ordnung $\text{ord}(a)$ enthält.

(Hinweis: Um sich an zyklische Gruppen und deren Eigenschaften zu gewöhnen, ist es hilfreich, alle Ordnungen „von Hand", d. h. nur mit einem Taschenrechner, auszurechnen. Wenn man sein Kopfrechnen auffrischen möchte, kann man versuchen, so weit wie möglich ohne Taschenrechner auszukommen.)

8.2. Wir betrachten die Gruppe $\mathbb{Z}_{53}^*$. Welche möglichen Ordnungen können Elemente haben? Wie viele Elemente existieren für jede Ordnung?

8.3. Wir schauen uns nun die Gruppen aus Aufgabe 8.1 an.

1. Wie viele Elemente hat jede der multiplikativen Gruppen?
2. Teilen alle der obigen Ordnungen die Anzahl der Elemente in der zugehörigen multiplikativen Gruppe?
3. Welche der Elemente aus Aufgabe 8.1 sind primitive Elemente?
4. Zeigen Sie für diese Gruppen, dass die Anzahl der primitiven Elemente durch $\phi(|\mathbb{Z}_p^*|)$ gegeben ist.

8.4. In dieser Aufgabe wollen wir primitive Elemente (Generatoren) einer multiplikativen Gruppe finden. Generatoren spielen eine große Rolle für den Diffie-Hellman-Schlüsselaustausch und viele weitere DL-basierte asymmetrische Verfahren. Gegeben seien die Primzahl $p = 4969$ und die zugehörige multiplikative Gruppe $\mathbb{Z}_{4969}^*$.

1. Bestimmen Sie die Anzahl der Generatoren in $\mathbb{Z}_{4969}^*$.
2. Wie hoch ist die Wahrscheinlichkeit für ein zufälliges Element $a \in \mathbb{Z}_{4969}^*$, ein Generator zu sein?
3. Bestimmen Sie den kleinsten Generator $a \in \mathbb{Z}_{4969}^*$ mit $a > 1000$.
   Hinweis: Diese Aufgabe kann durch einfaches Testen *aller* möglichen Faktoren der Gruppenordnung $p-1$ gelösten werden. Eine alternative und viel effizientere Methode besteht darin, die Aussage $a^{(p-1)/q_i} \neq 1 \bmod p$ für alle Primfaktoren q_i mit $p-1 = \prod q_i^{e_i}$ zu testen. Starten Sie mit $a = 1001$ und wiederholen Sie diese Schritte, bis Sie einen Generator von $\mathbb{Z}_{4969}^*$ erhalten.

4. Welche Maßnahme bezüglich der Wahl von p kann man zur Vereinfachung der Suche nach Generatoren in $\mathbb{Z}_p^*$ durchführen?

8.5. Berechnen Sie die beiden öffentlichen Schlüssel und den gemeinsamen Sitzungsschlüssel für das Diffie-Hellmann-Protokoll mit den Parametern $p = 467$, $\alpha = 2$ und

1. $a = 3, b = 5$
2. $a = 400, b = 134$
3. $a = 228, b = 57$

Führen Sie in allen Fällen die Berechnungen für Alice *und* Bob aus. Dies ist auch eine gute Überprüfung der Ergebnisse.

8.6. Wir betrachten nun einen weiteren DHKE mit derselben Primzahl $p = 467$ wie in Aufgabe 8.5. Dieses Mal nutzen wir jedoch das Element $\alpha = 4$. Das Element 4 hat die Ordnung 233 und generiert eine Untergruppe mit 233 Elementen. Berechnen Sie k_{AB} für

1. $a = 400, b = 134$
2. $a = 167, b = 134$

Warum sind die Sitzungsschlüssel identisch?

8.7. Für das DHKE-Protokoll werden die privaten Schlüssel aus der Menge

$$\{2, \ldots, p-2\}$$

gewählt. Warum sind hier die Werte 1 und $p-1$ ausgenommen? Beschreiben Sie, warum diese beiden Werte Sicherheitsprobleme darstellen.

8.8. Gegeben sei ein DHKE-System. Der Modul p hat 1024 Bit und α ist ein Generator einer Untergruppe mit $\text{ord}(\alpha) \approx 2^{160}$.

1. Was ist die maximale Größe, die die privaten Schlüssel haben sollten?
2. Wie lange dauert im Durchschnitt die Berechnung der Sitzungsschlüssel, wenn eine modulare Multiplikation 700 μs und eine modulare Quadrierung 400 μs benötigt? Nehmen Sie dabei an, dass die öffentlichen Schlüssel bereits berechnet wurden.
3. Eine bekannte Technik zur Beschleunigung von diskreten Logarithmus-Systemen verwendet kleine primitive Elemente. Wir nehmen nun an, dass α ein solches kleines Element ist (z. B. eine natürliche Zahl mit einer Länge von 16 Bit). Nehmen wir weiterhin an, dass eine modulare Multiplikation mit α nun nur noch 30 μs benötigt. Wie lange benötigt nun die Berechnung des öffentlichen Schlüssels? Warum ist die Ausführungsdauer für eine modulare Quadrierung bei Anwendung des Square-and-Multiply-Algorithmus immer noch die gleiche wie oben?

8.9. Nun wollen wir die Bedeutung der richtigen Wahl von Generatoren in multiplikativen Gruppen betrachten.

1. Zeigen Sie, dass die Ordnung eines Elementes $a \in \mathbb{Z}_p$ mit $a = p-1$ immer 2 ist.
2. Welche Untergruppe wird durch a erzeugt?
3. Beschreiben Sie kurz einen einfachen Angriff auf den DHKE, welcher diese Eigenschaft ausnutzt.

8.10. Wir betrachten ein DHKE-Protokoll über einem endlichen Körper $GF(2^m)$. Sämtliche Berechnungen werden in $GF(2^5)$ mit dem irreduziblen Polynom $P(x) = x^5 + x^2 + 1$ durchgeführt. Das primitive Element für das Diffie-Hellman-Verfahren ist $\alpha = x^2$. Die privaten Schlüssel sind durch $a = 3$ und $b = 12$ gegeben. Wie lautet der Sitzungsschlüssel k_{AB}?

8.11. In Abschnitt 8.4 wurde gezeigt, dass das Diffie-Hellman-Protokoll so sicher wie das Diffie-Hellman-Problem ist, d. h. der DHKE ist so sicher wie das DLP in der Gruppe $\mathbb{Z}_p^*$. Diese Aussage gilt jedoch nur für passive Angreifer, d. h. Oskar kann den Kanal lediglich abhören. Ist Oskar auch in der Lage, Nachrichten zwischen Alice und Bob abzufangen und zu manipulieren, kann das Schlüsselaustausch-Protokoll einfach gebrochen werden. Entwickeln Sie einen aktiven Angriff gegen den Diffie-Hellman-Schlüsselaustausch, in welchem Oskar der sogenannte „Mann in der Mitte" ist.

8.12. Schreiben Sie ein Programm, welches den diskreten Logarithmus $\mathbb{Z}_p^*$ durch eine vollständige Suche ermittelt. Die Eingabeparameter für Ihr Programm sind p, α, β. Das Programm soll x berechnen, wobei $\beta = \alpha^x \bmod p$.

Berechnen Sie eine Lösung für $\log_{106} 12375$ in $\mathbb{Z}_{24691}$.

8.13. Verschlüsseln Sie die folgenden Nachrichten mit dem Elgamal-Verfahren, wobei $p = 467$ und $\alpha = 2$:

1. $k_{pr} = d = 105, i = 213, x = 33$
2. $k_{pr} = d = 105, i = 123, x = 33$
3. $k_{pr} = d = 300, i = 45, x = 248$
4. $k_{pr} = d = 300, i = 47, x = 248$

Entschlüsseln Sie nun jedes Chiffrat und zeigen Sie alle Zwischenschritte.

8.14. Nehmen wir an, Bob schickt eine mit dem Elgamal-Verfahren verschlüsselte Nachricht an Alice. Fälschlicherweise verwendet Bob denselben Parameter i für alle Nachrichten. Darüber hinaus wissen wir, dass jeder Klartext von Bob mit der Zahl $x_1 = 21$ (Bobs ID) beginnt. Wir erhalten nun folgende Chiffrate:

$$(k_{E,1} = 6, y_1 = 17),$$
$$(k_{E,2} = 6, y_2 = 25).$$

Die Parameter von Elgamal sind $p = 31, \alpha = 3, \beta = 18$. Bestimmen Sie den zweiten Klartext x_2.

8.15. Sei ein Elgamal-Kryptosystem gegeben. Bob versucht besonders schlau zu sein und wählt folgenden Pseudozufallszahlengenerator zur Berechnung neuer Werte i:

$$i_j = i_{j-1} + f(j) \ , \ 1 \leq j \tag{8.5}$$

wobei $f(j)$ eine "komplizierte", aber bekannte Pseudozufallsfunktion ist (z. B. könnte $f(j)$ eine kryptografische Hashfunktion wie SHA-1 sein). i_0 ist eine echte Zufallszahl, welche Oskar nicht bekannt ist.

Bob verschlüsselt n Nachrichten x_j wie folgt:

$$k_{E_j} \equiv \alpha^{i_j} \bmod p$$
$$y_j \equiv x_j \cdot \beta^{i_j} \bmod p,$$

wobei $1 \leq j \leq n$. Nehmen Sie an, dass Oskar neben allen Geheimtexten auch den letzten Klartext x_n kennt.

Geben Sie eine Formel an, mit welcher Oskar jede der Nachrichten x_j, $1 \leq j \leq n-1$ berechnen kann. Natürlich kennt Oskar nach dem Kerckhoffsschen Prinzip auch das obige Verfahren sowie die Funktion $f()$.

8.16. Sei ein Elgamal-Verschlüsselungsverfahren mit den öffentlichen Parametern $k_{pub} = (p, \alpha, \beta)$ und einem unbekannten privaten Schlüssel $k_{pr} = d$ gegeben. Durch eine fehlerhafte Implementierung des Zufallszahlengenerators der verschlüsselnden Partei gilt der folgende Zusammenhang zwischen zwei temporären Schlüsseln:

$$k_{M,j+1} = k_{M,j}^2 \mod p.$$

Seien n aufeinanderfolgende Chiffrate

$$(k_{E_1}, y_1), (k_{E_2}, y_2), ..., (k_{E_n}, y_n)$$

gegeben, die zu Klartexten

$$x_1, x_2, ..., x_n.$$

gehören. Sei darüber hinaus der erste Klartext x_1 bekannt (z. B. der Header einer Datei).

1. Beschreiben Sie, wie ein Angreifer die Klartexte $x_1, x_2, ..., x_n$ aus den gegebenen Informationen berechnen kann.
2. Kann ein Angreifer hieraus auch den privaten Schlüssel d berechnen? Begründen Sie Ihre Antwort.

8.17. Das Elgamal-Verfahren ist nichtdeterministisch, d. h. ein gegebener Klartext x hat viele gültige Chiffrate.

1. Warum ist die Elgamal-Verschlüsselung probabilistisch?
2. Wie viele gültige Chiffrate existieren für jede Nachricht x (allgemeiner Ausdruck)?
   Wie viele gibt es in Aufgabe 8.13 (numerische Antwort)?

3. Ist das RSA-Kryptosystem deterministisch, nachdem der öffentliche Schlüssel gewählt wurde?

8.18. Wir schauen uns nun die Schwächen der Elgamal-Verschlüsselung an, wenn ein öffentlicher Schlüssel mit einer kleinen Ordnung verwendet wird. Betrachten wir das folgende Beispiel: Angenommen, Bob verwendet die Gruppe $\mathbb{Z}_{29}^*$ mit dem primitiven Element $\alpha = 2$. Sein öffentlicher Schlüssel ist $\beta = 28$.

1. Was ist die Ordnung des öffentlichen Schlüssels?
2. Welche möglichen Maskierungsschlüssel k_M gibt es?
3. Alice verschlüsselt eine Nachricht, wobei jeder Buchstabe nach der einfachen Regel $\texttt{a} \to 0$, …, $\texttt{z} \to 25$ kodiert wird. Es gibt drei zusätzliche Symbole im Chiffrat, $\ddot{\texttt{a}} \to 26$, $\ddot{\texttt{o}} \to 27$, $\ddot{\texttt{u}} \to 28$. Alice überträgt die folgenden 11 Chiffrate (k_E, y):

$$(3,15),(19,14),(6,15),(1,24),(22,13),(4,7),$$
$$(13,4),(3,21),(18,17),(26,25),(7,17)$$

Entschlüsseln Sie die Nachricht, ohne Bobs privaten Schlüssel zu berechnen. Schauen Sie sich hierzu die Chiffrate genau an und raten Sie! Berücksichtigen Sie die Tatsache, dass es lediglich einige wenige Maskierungsschlüssel gibt.