Tutoriumsblatt 9 mit Musterlösung

Aufgabe 9.1: Symmetrische und asymmetrische Verschlüsselung

a) Nennen und erläutern Sie einen *wichtigen Unterschied* zwischen *symmetrischen* und *asymmetrischen* Verfahren.

b) Angenommen, N Personen wollen paarweise mittels symmetrischer Verschlüsselung kommunizieren. Alle Kommunikationspartner können die ausgetauschen, verschlüsselten Nachrichten mitlesen, aber keiner außer den zwei kommunizierenden Personen soll in der Lage sein, die mitgelesene Kommunikation zu entschlüsseln.

(i) *Wie viele Schlüssel werden benötigt?*

(ii) Nehmen Sie nun an, dass *asymmetrische Verschlüsselung* genutzt wird. *Wie viele Schlüssel werden benötigt?*

c) Warum verwenden die meisten Protokolle *sowohl symmetrische als auch asymmetrische* Verfahren?

d) Was ist ein *Man-in-the-Middle-Angriff (MITM)*? Kann dieser Angriff durchgeführt werden, wenn *symmetrische Schlüssel* benutzt werden?

e) Gegeben sei RSA mit $p = 5$ und $q = 13$.

i) Was sind n und $\varphi(n)$?

ii) Sei $e = 5$. Warum ist dies eine *gute* Wahl?

iii) *Finden* Sie ein d, so dass $d \cdot e = 1 \mod \varphi(n)$.

iv) *Verschlüsseln* Sie die Nachricht $m = 8$ mit dem Schlüssel $\langle e, n \rangle$.

Lösung 9.1

1.a) Ein wichtiger Unterschied zwischen symmetrischer und asymmmetrischer Verschlüsselung ist, dass bei symmetrischen Verfahren beide, der Sender und Empfänger, den (geheimen) Schlüssel kennen müssen und dieser zum Ver- und Entschlüsseln benutzt wird.

Bei asymmetrischer Verschlüsselung sind die Schlüssel für die Verschlüsselung und Entschlüsselung unterschiedlich. Alle anderen, auch der Angreifer, kennen den zur Verschlüsselung benutzten Schlüssel, aber nur der Empfänger kennt den Schlüssel zur Entschlüsselung.

1.b)

(i) Wenn jeder mit jedem sicher kommunizieren will, muss jedes Kommunkationspaar einen gemeinsamen Schlüssel haben. Es gibt $\frac{N \cdot (N-1)}{2}$ solcher Paare, also werden auch so viele Schlüssel benötigt.

(ii) Wenn asymmetrische Verschlüsselung benutzt wird, dann hat jede Person einen öffentlichen Schlüssel (zur Verschlüsselung), den jeder andere kennt, und einen geheimen, privaten Schlüssel (zur Entschlüsselung). Also werden $2 \cdot N$ Schlüssel benötigt. Das ist im Allgemeinen deutlich weniger.

1.c) Ein Vorteil asymmetrischer Verfahren steht schon in Teil b): es müssen weniger Schlüssel verteilt werden. Auch ist die Verteilung der Schlüssel einfacher: wir können ein Zertifikat unseres öffentlichen Schlüssels erzeugen und dieses zum Download bereitstellen. (Wobei unser Zertifikat authentifiziert sein

muss, durch eine CA.) Wir brauchen uns nicht erst Gedanken zu machen, wie wir eine authentifizierte Schlüsselaushandlung vornehmen können.

Ein Nachteil asymmetrischer Verschlüsselung ist, dass sie extrem langsam ist.

Wenn große Datenmengen ausgetauscht werden müssen, werden asymmetrische Verfahren zur Authentifizierung (Zertifikate) genutzt und auch dazu, einen symmetrischen Schlüssel auszutauschen, welcher nun effizient zur Verschlüsselung der Daten genutzt werden kann (mittels AES oder ähnlichem). (Anmerkung: auch Diffie-Hellman wird als asymmetrisches Verfahren bezeichnet, fällt also auch in diese Kategorie. Hier müssen die Nachrichten zusätzlich authentifiziert werden, sonst ist ein MITM-Angriff möglich.)

1.d) Bei einem MITM-Angriff setzt sich der Angeifer zwischen Alice und Bob und liest und/oder verändert die ausgetauschten Nachrichten. Er täuscht also beiden Seiten vor, der jeweils andere Kommunikationspartner zu sein.

Werden symmetrische Schlüssel benutzt, können Alice und Bob alle Nachrichten verschlüsseln (und sich durch korrekte Verschlüsselung im Prinzip auch authentifizieren). Man ist also gegen einen MITM geschützt. Aber Vorsicht: während der Schlüsselaushandlung kann sich eventuell schon ein MITM eingeklinkt haben, wenn sie nicht authentifiziert stattgefunden hat. In diesem Fall helfen auch symmetrische Schlüssel nicht mehr.

Also: symmetrische Schlüssel helfen, aber bei der Schlüsselverteilung muss man Vorsicht walten lassen.

1.e) Sieh Folie VI-39.

i) $n = p \cdot q = 65$, $\varphi(n) = (p-1) \cdot (q-1) = 48$

ii) $e = 5$ ist kleiner als n und hat keinen gemeinsamen Faktor mit $\varphi(n)$, also ist es eine korrekte Wahl. Die Wahl einer kleinen Zahl hat außerdem den Vorteil, dass die Rechenoperationen relativ schnell ausgeführt werden können, so dass die Anwendung des Schlüssels nicht viel Zeit kostet. Das ist vor allem bei der Verwendung asymmetrischer Verschlüsselung als digitales Signaturverfahren nützlich, da das Prüfen von Signaturen einfach wird und nur die Erstellung der Signatur lange dauert. Da eine Signatur eventuell nur ein mal erstellt aber mehrmals geprüft wird, ist dies sinnvoll.

iii) $d = 29$, da $d \cdot e = 1 \mod \varphi(n)$

Erweiterter euklidischer Algorithmus: Der euklidische Algorithmus wird in den Spalten q_i und r_i eingetragen. Dabei gilt: $r_{i-2} = q_i \cdot r_{i-1} + r_i$. u_i und v_i sind Faktoren von $\varphi(n)$ und e, so dass gilt: $r_i = u_i \cdot \varphi(n) + v_i \cdot e$. Für die Berechnung gilt: $u_i = u_{i-2} - (q_i \cdot u_{i-1})$ und $v_i = v_{i-2} - (q_i \cdot v_{i-1})$. Das gesuchte multiplikative Inverse (muss nicht positiv sein, daher eventuell ein weiterer Schritt erforderlich) steht in v_{N-1}, wobei N die Anzahl der Schritte des euklidischen Algorithmus' ist.

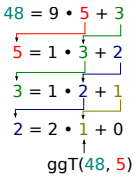
i	q_i	r_i	u_i	v_i
-2		48	1	0
-1		5	0	1
0	9	3	1	-9
1	1	2		10
2	1	1		-19
3	2	0		

Wir sind jetzt fast fertig. Allerdings soll das Ergebnis positiv sein. Es gilt $-19 \cdot 5 \mod 48 = 1$. Also addieren wir $48 \cdot 5$ auf beiden Seiten (ändert auf der rechten Seite natürlich nichts) und erhalten $29 \cdot 5 \mod 48 = 1$. Damit ist 29 das gesuchte multiplikative Inverse und damit ergibt sich der private, geheime Schlüssel:

$$\langle d, n \rangle = \langle 29, 65 \rangle$$

Hinweis: Hier noch eine andere Möglichkeit den erweiterten euklidischen Algorithmus zu präsentieren (in Bunt und Farbe).

Wir wollen die Inverse von 5 mod 48 berechnen. Dazu schreiben wir zunächst den euklidischen Algorithmus auf, so als wollten wir den ggT der beiden Zahlen ermittlen (der ist natürlich 1).



Der erweiterte euklidische Algorithmus: Ausgehend von der vorletzten Zeile rollen wir die Rechenschritte von unten nach oben auf, indem wir die einzelnen Zeilen nach den Resten auflösen und diese nacheinander einsetzen:

$$
\begin{aligned}
1 &= 3 - 1 \cdot 2 \\
&= 3 - 1 \cdot (5 - 1 \cdot 3) = 2 \cdot 3 - 1 \cdot 5 \\
&= 2 \cdot (48 - 9 \cdot 5) - 1 \cdot 5 = 2 \cdot 48 - 19 \cdot 5
\end{aligned}
$$

Dann haben wir $2 \cdot 48 - 19 \cdot 5 = 1$, daher gilt $-19 \cdot 5 \mod 48 = 1$. Die gesuchte Inverse soll positiv sein, also addieren wir noch $48 \cdot 5$ und erhalten $29 \cdot 5 \mod 48 = 1$. Daher ist 29 das gesuchte Inverse.

iv) $m = 8$, $m^e = 32.768$, Ciphertext $c = m^e \mod n = 8$

Dumm gelaufen, der Chiffretext ist identisch zum Klartext. Aber das ist wirklich Zufall. Vielleicht sollte man doch größere Zahlen verwenden.

Aufgabe 9.2: Diffie-Hellman

a) Zur gesicherten Kommunikation wollen Alice und Bob einen geheimen Schlüssel vereinbaren. Sie verwenden den Algorithmus von Diffie-Hellman mit den Parametern $p = 31$ und $g = 15$; diese Parameter sind Alice und Bob bereits bekannt. Als Geheimzahl generiere Alice die Zahl 3, Bob die Zahl 4.

Berechnen Sie den geheimen Schlüssel unter Verwendung des Algorithmus' von Diffie-Hellman. Geben Sie als Lösung an, welche Operationen/Berechnungen Alice und Bob jeweils ausführen und welche Informationen an den jeweiligen Kommunikationspartner übermittelt werden.

b) Die Wahl der Werte in Teil a) war nicht optimal – einer der Werte macht es einem Angreifer einfacher, den geheimen Schlüssel zu ermitteln. *Welcher der Werte war schlecht gewählt, und welche Probleme verursacht diese Wahl?*

Anmerkung: dass alle Werte zu klein sind, ist hier nicht gemeint.

Lösung 9.2

a) 1.) Alice wählt $a = 3$ und berechnet $A = g^a \bmod p = 15^3 \bmod 31 = 27$.
2.) Alice sendet A an Bob.
3.) Bob wählt $b = 4$, berechnet $B = g^b \bmod p = 15^4 \bmod 31 = 2$
4.) Bob sendet B an Alice.
5.) Bob berechnet den gemeinsamen Schlüssel $K = A^b \bmod p = 27^4 \bmod 31 = \mathbf{8}$.
6.) Alice berechnet den Schlüssel $K = B^a \bmod p = 2^3 \bmod 31 = \mathbf{8}$.

b) g ist schlecht gewählt.

g sollte Generator sein. Berechnet man alle Potenzen von $g = 15$, sieht man das Problem: g erzeugt nur eine Teilmenge der möglichen Elemente modulo 31. (Nämlich nur 10 Stück.) Damit ist der Raum der möglichen Schlüssel eingeschränkt und auch die Wahl der Geheimzahlen. (z.B. ist $15^2 = 15^{12}$ – für einen Angreifer kommt es also aufs selbe raus, ob er 2 oder 12 testet.) Die Sicherheit ist also eingeschränkt, da ein Angreifer einen Schlüssel schneller ermitteln kann

Aufgabe 9.3: Schlüsselvereinbarung

In der Vorlesung wurde der Diffie-Hellman-Algorithmus vorgestellt, mit dessen Hilfe zwei Parteien sich auf einen geheimen, gemeinsamen Schlüssel einigen können. In dieser Aufgabe soll jetzt alternativ ein sehr einfaches Protokoll zum Austausch geheimer Schlüssel bezüglich seiner Sicherheit analysiert werden. Das Protokoll arbeitet wie folgt ($\oplus$ steht für die XOR-Verknüpfung):

i) Alice wählt zufällig $k, a \in \{0,1\}^n$ und sendet $s = k \oplus a$ an Bob.

ii) Bob wählt eine Zufallszahl $b \in \{0,1\}^n$ aus und sendet $u = s \oplus b$ an Alice.

iii) Alice berechnet $w = u \oplus a$ und sendet w an Bob.

iv) Alice benutzt k und Bob $w \oplus b$ als geheimen Schlüssel.

Beantworten Sie nun folgende Fragen:

a) Zeigen Sie, dass Alice und Bob im Besitz des gleichen Schlüssels sind.

b) Bietet das Protokoll einen sicheren Schlüsselaustausch, falls ein Angreifer die Nachrichten zwar mitlesen, aber nicht modifizieren kann? Begründen Sie Ihre Antwort.

Lösung 9.3

Hier kann man ein Diagramm aufzeichnen - es hilft vielleicht, wenn man vor sich sieht, wie die Nachrichten hier ausgetauscht werden.

3.a) Alice benutzt k als geheimen Schlüssel. Setzt man nun einfach die übertragenen Werte ein, erhält man:

$$s = k \oplus a \tag{1}$$
$$u = s \oplus b = (k \oplus a) \oplus b \tag{2}$$
$$w = u \oplus a = [(k \oplus a) \oplus b] \oplus a \tag{3}$$

Bob benutzt $w \oplus b$ als geheimen Schlüssel. Es ergibt sich also:

$$w \oplus b = ([(k \oplus a) \oplus b] \oplus a) \oplus b = k$$

Alice und Bob verwenden also denselben Schlüssel.

3.b) Das Protokoll ist nicht sicher, da ein Angreifer (Eve) den geheimen Schlüssel durch Belauschen der ausgetauschten Nachrichten rekonstruieren kann. Natürlich muss Eve dazu das Protokoll bekannt sein.

1. Die Nachrichten s, u, w können belauscht werden.
2. Eve berechnet nun $s \oplus u$:
$$s \oplus u = (k \oplus a) \oplus (s \oplus b) = (k \oplus a) \oplus ((k \oplus a) \oplus b) = b.$$
3. Als nächstes kann Eve $w \oplus b$ berechnen:
$$w \oplus b = (u \oplus a) \oplus b = ((s \oplus b) \oplus a) \oplus b = ((k \oplus a) \oplus b) \oplus a) \oplus b = k$$

Also kann Eve den geheimen Schlüssel konstruieren – das Protokoll ist **nicht** sicher!

Man sollte es sich nie zu einfach machen, sondern gründlich über die Möglichkeiten eines Angreifers nachdenken. Oder noch besser: nie eigene Sicherheitslösungen entwickeln, sondern immer auf vernünftige Libraries zurückgreifen.