Grundkurs Codierung

Lösungsvorschläge zu den Fragen in den Unterkapiteln „Was blieb?“
Stand 23.04.2007

Unterkapitel 5.6, Seiten 294 und 295

Zu Frage 1:

Da bei einem symmetrischen Verschlüsselungsverfahren wie dem DES oder dem AES jeder der n Teilnehmer mit jedem der n-1 Partner einen geheimen Schlüssel austauschen muss, berechnet sich die Gesamtzahl nach der Formel für die ”2 aus n”-Binomialkoeffizienten

$$\binom{n}{2} = \frac{n \cdot (n-1)}{2}$$

Für 2 Teilnehmer ist es ein Austausch, für 3 Teilnehmer sind es drei Vorgänge, für 4 Teilnehmer sechs usw. (polynomiale Komplexität, hier quadratisch mit n^2).

Zu Frage 2:

Bei einem asymmetrischen Verfahren wie dem RSA braucht der öffentliche Schlüssel natürlich nicht geheim ausgetauscht zu werden. Der geheime Schlüssel ist nur jedem Teilnehmer selbst mitzuteilen, nicht einem zweiten Teilnehmer zusätzlich wie bei den symmetrischen Verfahren. Hierbei sind zwei Varianten zu unterscheiden:

- Werden die Schlüssel von einer – vertrauenswürdigen - Vergabestelle erzeugt, sind bei n Teilnehmern genau n Übermittlungen erforderlich, also nur proportional zur Teilnehmeranzahl.

- Der geheime Schlüssel kann aber auch bei jedem Teilnehmer selbst berechnet werden. In diesem Fall ist überhaupt kein Schlüsselaustausch erforderlich (Beispiel: https-Protokoll). Wenn der den Schlüssel erzeugende Rechner gegen unberechtigte Zugriffe, z. B. über Trojaner, geschützt ist, wird damit also das Problem der sicheren Schlüsselübergabe vollständig gelöst. Global betrachtet ist dann trotzdem nicht automatisch vollkommene Sicherheit gegeben, da auch die Vertrauenswürdigkeit jedes Teilnehmers garantiert sein muss.

Zu Frage 3:

Die Sicherheit des RSA-Verfahrens beruht ausschliesslich auf der z. Z. nicht gegebenen Möglichkeit der schnellen Zerlegung von Zahlen in Primfaktoren. Man kann also den Modul $n = p \cdot q$ nicht in kurzer Zeit (= Stunden, Tage) in seine Faktoren p und q zerlegen, obwohl es prinzipiell sehr einfache Wege hierfür gibt. Auch die ständige Leistungsteigerung der Rechnersysteme stellt vorläufig keine Gefahr dar, da durch Vergrößerung von n der Einfluss kürzerer Rechenzeiten beliebig zunichte gemacht werden kann. Sollte sich allerdings die Rechenleistung mit Hilfe neuartiger Technologien (= Quantencomputer) um beinahe beliebige Größenordnungen steigern lassen, wären alle Verfahren “tot”, deren Sicherheit auf dem Zeitbedarf für Primzahlenzerlegung beruht. Dann käme aus heutiger Sicht nur noch das one time pad-Verfahren in Form der Quantenkryptographie in Frage, für das es im Gegensatz zu Quantencomputern bereits realistische Labormodelle gibt.

Zu Frage 4:

Nein, gemäß dem Prinzip von Kerckhoff (siehe Unterkapitel 5.2, Seite 276) hängt die Sicherheit eines Verschlüsselungsverfahrens allein von der Qualität und Geheimhaltung des Schlüssels ab, nicht vom verwendeten Algorithmus (genauer: Die Sicherheit hängt so gut wie überhaupt nicht vom verwendeten Algorithmus ab, sieht man einmal von Triviallösungen ab, wie dem Einsatz von Verschiebeverfahren auf

Klartexte mit ausgeprägtem Verteilungsprofil).

Zu Frage 5:

Ausser der Qualität und der Geheimhaltung der Schlüssel gibt es weitere Anforderungen zur Gewährleistung der Sicherheit von Informatiksystemen. Gemäß Unterkapitel 4.3, Seite 255 ff, dürfen die Geheimtextzeichen kein ausgeprägtes und eindeutiges Verteilungsprofil aufweisen. Dies kann perfekt mit dem Verfahren der one time pad-Verschlüsselung erreicht werden, womit nach dem heutigen Stand der Technologien aber noch anderweitige Nachteile verbunden sind (z. B. eine aufwändige und deshalb unsichere Schlüsselübergabe). Ein weiterer Weg besteht in der Zusammenfassung möglichst vieler Klartextzeichen zu Superzeichen (mit der Folge der dann notwendigen längeren Schlüssel), wodurch sich das Verteilungsprofil der Klartextzeichen verflachen und gegen Entschlüsselungsversuche ohne Schlüssel unempfindlicher machen lässt.

Ein ganz anderer Gesichtspunkt liegt im Ablaufprotokoll der Geheimtexterzeugung und -übertragung. Schwachstellen wie z. B. eine ungenügende Verhinderung von “men in the middle”-Zugriffen können hier die mathematische Sicherheit eines Verfahrens vollständig unterlaufen. Deshalb sind diese Abläufe besonders sorgfältig zu analysieren und zu gestalten.

Zu Frage 6:

Die Vertraulichkeit lässt sich durch Verschlüsselung der Klartexte erreichen (siehe Unterkapitel 5.2, 5.3, 5.7 und 5.14), Datenintegrität durch Hash-Summen (siehe Unterkapitel 5.11) und elektronische Unterschrift, (siehe Unterkapitel 5.3 und 5.12). Diese Verfahren stellen die technischen Anforderungen sicher. Zu beachten ist aber ebenfalls, ob man den Partnern als Personen trauen kann. Einer der Wege, um dieses Vertrauen herzustellen oder zu verbessern, ist die Zertifizierung durch eine vertrauenswürdige Instanz oder durch einen möglichst großen Kreis vertrauenswürdiger Personen wie etwa bei GNUPP.

Zu Frage 7:

Gemäß Unterkapitel 5.2 werden beim DES (oder AES) -Verfahren nur XOR-Verknüpfungen und Bit-Verschiebungen benötigt, die sich als elementare Maschinenbefehle praktisch mit Zykluszeit-Geschwindigkeit abarbeiten lassen. Beim RSA-Verfahren sind im Vergleich hierzu aufwändige Algorithmen (Erzeugung von Pseudoprimzahlen, Euklidischer Algorithmus zur Prüfung auf Teilerfreiheit und zur Bestimmung modularer inverser Zahlen, Berechnung modularer Potenzen) durchlaufen zu lassen, siehe Unterkapitel 5.3, 5.4 und 5.5. Diese erfordern etwa die 10-fache Rechenzeit und verringern dementsprechend den Echtzeit-Durchsatz, der im online-Betrieb eine entscheidende Rolle spielt.

Zu Frage 8:

Die beiden verschieden Einsatzwege des RSA-Algorithmus sind gemäß Unterkapitel 5.3

- Erzeugung von Geheimtexten zum Versenden vertraulicher Nachrichten: Verschlüsselung des Klartextes mit dem öffentlichen Schlüssel des Adressaten. Er kann den Geheimtext mit seinem geheimen Schlüssel entschlüsseln

- Sicherstellung der Authentizität einer Nachricht (“stammt diese vom vorgegebenen Autor?”): Verschlüsseln eines Klartextes durch den Autor mit seinem geheimen Schlüssel, der entstandene “Geheimtext” kann dann von jedem Interessenten mit dem zugeordneten öffentlichen Schlüssel entschlüsselt werden. Da nur dieser passt, ist dies der Nachweis, dass die Nachricht tatsächlich vom Besitzer des Geheimschlüssels erstellt wurde. Doch *Vorsicht*: Es besteht damit noch keinerlei Garantie, dass man eine inhaltlich korrekte Nachricht erhalten hat, sie könnte betrügerisch sein. Auch hier benötigt man also zusätzlich das Vertrauen in die Integrität des Schlüsselbesitzers.

Zu Frage 9:

Nach Unterkapitel 5.5, Seite 289 ff, sinkt die Wahrscheinlichkeit dafür, dass es sich bei der gewählten Zufallszahl nicht um eine Primzahl handelt, bei jedem erfolgreichen Durchlaufen eines Testschrittes um den Faktor 4. Bei 50 erfolgreichen Testschritten beträgt die Wahrscheinlichkeit für eine Primzahl also

$$1 - 0.25^{50} = 1 - 7.89 \cdot 10^{-31} = 0.99999\ 99999\ 99999\ 99999\ 99999\ 99999\ 21.$$

Im Mittel würde man auf 2 mal 1 Billion x 1 Billion x 10 Millionen Versuche eine unerkannte Nicht-Primzahl erhalten.

Zu Frage 10:

Die anspruchsvollste Grundrechenoperation ist in diesem Fall die Multiplikation bzw. die Potenzierung MOD n. Bei letzterer kommen nach zweckmäßiger Zerlegung des Exponenten höchstens Quadrate vor, siehe Unterkapitel 5.4, Seiten 288 und 289. Eine Quadratzahl darf deshalb maximal 15 Dezimalstellen enthalten, die beiden Faktoren müssen auf 7.5 Stellen beschränkt bleiben. Dies gilt auch für den Modul n. Der Algorithmus muss also mit einer Ganzzahlgenauigkeit von 15 Dezimalstellen ausgeführt werden können. “Bessere” Taschenrechner arbeiten mit einer internen Genauigkeit von 12 – 16 Dezimalstellen.

Mit programmierbaren Geräten lässt sich auf Kosten der Rechenzeit natürlich jede erforderliche Genauigkeit erzielen.