Aufgaben

10.1. In Abschnitt 10.1.3 wurde behauptet, dass Nachrichtenauthentisierung immer (Nachrichten-) Integrität impliziert. Warum gilt diese Aussage? Gilt im Umkehrschluss auch, dass bei Integrität immer auch Nachrichtenauthentisierung gegeben ist? Begründen Sie beide Antworten.

10.2. In dieser Aufgabe werden einige grundlegende Eigenschaften von Sicherheitsdiensten betrachtet.

1. Folgt aus gegebener Vertraulichkeit immer Integrität? Warum?
2. In welcher Reihenfolge sollten bei der Nachrichtenübertragung Vertraulichkeit und Integrität sichergestellt werden, d. h. soll die Verschlüsselung oder die Integritätsmaßnahme zuerst berechnet werden? Begründen Sie Ihre Antwort.

10.3. Wir betrachten ein Kommunikationssystem mit zwei Teilnehmern und einem unsicheren Kanal. Entwerfen Sie ein System, dass die Dienste Geheimhaltung, Integrität und Nichtzurückweisbarkeit zur Verfügung stellt. Begründen Sie, warum die drei Sicherheitsdienste erreicht werden. (Hinweis: Betrachten Sie bei Ihrer Argumentation Angriffe gegen das System.)

10.4. Ein Maler hat eine neue Geschäftsidee: Er bietet Bilder an, die auf Fotos basieren. Sowohl die Fotos als auch die Bilder werden in digitaler Form über das Internet versendet. Es soll hierbei Diskretion gewahrt werden, da auch brisante Fotos wie z. B. Nacktfotos eingesandt werden können. Von daher sollten die Fotos während der Übertragung für Dritte nicht lesbar sein. Da die Erstellung eines Bildes eine erhebliche Zeitinvestition darstellt, muss der Maler sicherstellen, dass er den wahren Namen der Kunden kennt. Er muss auch sicherstellen, dass der Kunde nicht abstreitet, dass er das Bild bestellt hat.

1. Welche Sicherheitsdienste sind für die Übertragung der digitalen Fotos seitens des Kunden erforderlich?
2. Mit welchen kryptografischen Bausteinen (z. B. symmetrische Verschlüsselung) können die Sicherheitsdienste realisiert werden? Wir nehmen an, dass jedes Foto mehrere MByte groß ist.

10.5. Gegeben sei eine RSA-Signatur mit dem öffentlichen Schlüssel $(n = 9797, e = 131)$. Welche der folgenden Signaturen sind gültig?

1. $(x = 123, \mathrm{sig}(x) = 6292)$
2. $(x = 4333, \mathrm{sig}(x) = 4768)$
3. $(x = 4333, \mathrm{sig}(x) = 1424)$

10.6. Gegeben sei eine RSA-Signatur mit dem öffentlichen Schlüssel $(n = 9797, e = 131)$. Zeigen Sie anhand eines Beispiels, wie Oskar eine existentielle Fälschung durchführen kann.

10.7. Gegeben sei ein RSA-Signaturverfahren. Bob signiert Nachrichten x_i und sendet diese zusammen mit den Signaturen s_i und seinem öffentlichen Schlüssel an Alice. Bob hat den öffentlichen Schlüssel (n,e) und sein privater Schlüssel ist d. Oskar kann nun einen Mann-in-der-Mitte-Angriff ausführen, d. h. er kann Bobs öffentlichen Schlüssel durch seinen eigenen öffentlichen Schlüssel während der Übertragung ersetzen. Sein Ziel ist es, Nachrichten so zu manipulieren, dass Alice dies bei der Verifikation nicht erkennt. Zeigen Sie alle Schritte, die Oskar für einen erfolgreichen Angriff durchführen muss.

10.8. Gegeben sei eine RSA-Signatur mit EMSA-PSS-Padding, wie in Abschnitt 10.2.3 dargestellt. Beschreiben Sie jeden Schritt der Verifikation, der vom Empfänger der Signatur durchgeführt werden muss.

10.9. Ein wichtiger praktischer Aspekt digitaler Signaturen ist der benötigte Rechenaufwand zum (i) Signieren und (ii) Verifizieren. In dieser Aufgabe untersuchen wie die Rechenkomplexität des RSA-Signaturverfahrens.

1. Wie viele Multiplikationen werden durchschnittlich benötigt, um (i) eine Nachricht mit einem allgemeinen Exponenten zu signieren und (ii) eine Signatur mit mit dem kurzen Exponenten $e = 2^{16} + 1$ zu verifizieren? Wir nehmen an, dass n eine Länge von $l = \lceil \log_2 n \rceil$ Bit hat. Sowohl für das Signieren als auch für das Verifizieren wird der Square-and-Multiply-Algorithmus verwendet. Entwickeln Sie einen allgemeinen Ausdruck in Abhängigkeit von l für die Komplexität.
2. Welche Operation ist schneller, Signieren oder Verifizieren?
3. Wir entwickeln nun eine Geschwindigkeitsabschätzung für eine Software-Implementierung. Das folgende Zeitmodell für die Multiplikation wird verwendet: Ein Computer verwendet 32-Bit-Datenstrukturen. Daher wird jede Variable mit voller Länge, insbesondere n und x, durch ein Array mit $m = \lceil l/32 \rceil$ Elementen dargestellt (x ist hierbei die Basis bei der Exponentiation). Wir nehmen an, dass eine Multiplikation oder Quadrierung mit zwei solcher Variablen modulo n genau m^2 Zeiteinheiten benötigt (eine Zeiteinheit ist gleich der Taktperiode des Prozessors multipliziert mit einer Konstante größer als 1, die von der Implementierung abhängt). Man beachte, dass man niemals mit den Exponenten d und e multipliziert. Deswegen beeinflusst die Bitlänge des Exponenten nicht die Zeit, die eine modulare Quadrierung oder Multiplikation benötigt.
   Wie lange dauert eine Signatur bzw. eine Verifikation, wenn die Zeiteinheit auf einer bestimmten CPU 100 ns lang ist und n aus 512 Bit besteht? Was sind die Laufzeiten, um wenn l 1024 Bit lang ist?
4. Chipkarten sind eine wichtige Anwendungsdomäne für digitale Signaturen. Wir betrachten Karten, die mit einem 8051-Mikroprozessorkern ausgestattet sind. Es handelt sich hierbei um einen 8-Bit-Prozessor. Wie muss die Zeiteinheit gewählt werden, um eine Signaturerzeugung in 0,5 s durchzuführen, wenn n (i) 512 Bit und (ii) 1024 Bit besitzt? Sind diese Zeiteinheiten realistisch, wenn wir annehmen, dass die maximale Taktfrequenz bei 10 MHz liegt?

10.10. Diese Aufgabe beschäftigt sich mit Elgamal-Signaturen. Gegeben seien Bobs privater Schlüssel $K_{pr} = (d) = (67)$ und der dazugehörige öffentliche Schlüssel $K_{pub} = (p, \alpha, \beta) = (97, 23, 15)$.

1. Berechnen Sie die Elgamal-Signatur (r,s) und die entsprechende Verifikation für eine Nachricht, die von Bob an Alice gesandt wird. Die Nachricht x und der temporäre Schlüssel k_E haben die folgenden Werte:

   a. $x = 17$ und $k_E = 31$
   b. $x = 17$ und $k_E = 49$
   c. $x = 85$ und $k_E = 77$

2. Sie empfangen zwei Nachrichten x_1, x_2 zusammen mit den zugehörigen Signaturen (r_i, s_i) von Bob. Verifizieren Sie, ob die Nachrichten $(x_1, r_1, s_1) = (22, 37, 33)$ und $(x_2, r_2, s_2) = (82, 13, 65)$ beide von Bob kommen.
3. Vergleichen Sie das RSA-Signaturverfahren mit dem von Elgamal. Was sind die jeweiligen Vor- und Nachteile?

10.11. Gegeben sei eine Elgamal-Signatur mit $p = 31$, $\alpha = 3$ und $\beta = 6$. Die Nachricht $x = 10$ wird zweimal empfangen mit den Signaturen (r,s):

$$(i) \quad (17,5)$$
$$(ii) \quad (13,15)$$

1. Handelt es sich beide Male um gültige Signaturen?
2. Wie viele gültige Signaturen existieren für jede gegebene Nachricht x und die oben genannten Parameter?

10.12. Gegeben sei eine Elgamal-Signatur mit den öffentlichen Parametern $(p = 97, \alpha = 23, \beta = 15)$. Zeigen Sie, wie Oskar eine existentielle Fälschung erstellen kann, indem Sie ein Beispiel für eine gültige Signatur konstruieren.

10.13. Gegeben sei ein Elgamal-Signaturverfahren mit den öffentlichen Parametern $p, \alpha \in \mathbb{Z}_p^*$ und dem unbekannten privaten Schlüssel d. Aufgrund einer fehlerhaften Implementierung besteht zwischen zwei aufeinanderfolgenden temporären Schlüsseln der folgende Zusammenhang:

$$k_{E_{i+1}} = k_{E_i} + 2.$$

Des Weiteren sind zwei aufeinanderfolgende Signaturen

$$(r_1, s_1)$$
$$\text{und} \quad (r_2, s_2)$$

zu den Klartexten x_1 und x_2 gegeben. Zeigen Sie, wie ein Angreifer den privaten Schlüssel aus den obigen Werten berechnen kann.

10.14. Geben seien die DSA-Parameter $p = 59, q = 29, \alpha = 3$ und Bobs privater Schlüssel $d = 23$. Zeigen Sie die Prozesse der Signaturerstellung durch Bob und der Signaturverifikation durch Alice mit den folgenden Hashwerten $h(x)$ und temporären Schlüsseln k_E:

1. $h(x) = 17, k_E = 25$

2. $h(x) = 2, k_E = 13$
3. $h(x) = 21, k_E = 8$

10.15. Zeigen Sie, wie DSA gebrochen werden kann, wenn der temporärer Schlüssel mehrfach verwendet wird.

10.16. Die folgenden ECDSA-Parameter sind gegeben: Die Kurve $E : y^2 = x^3 + 2x + 2 \bmod 17$, der Punkt $A = (5,1)$ mit der Ordnung $q = 19$ und Bobs privater Schlüssel $d = 10$. Zeigen Sie die Prozesse der Signaturerstellung seitens Bob und der Signaturverifikation durch Alice mit den folgenden Hashwerten $h(x)$ und temporären Schlüsseln k_E:

1. $h(x) = 12, k_E = 11$
2. $h(x) = 4, k_E = 13$
3. $h(x) = 9, k_E = 8$