Public-Key-Infrastrukturen

16

Ein großer Vorteil asymmetrischer Kryptoverfahren besteht darin, dass die Schlüsselverwaltung einfacher ist als bei symmetrischen Verschlüsselungsverfahren. Die Schlüssel, die zum Verschlüsseln gebraucht werden, müssen nicht geheimgehalten werden, sondern sie können öffentlich sein. Die privaten Schlüssel müssen aber auch in Public-Key-Systemen geheim bleiben. Außerdem müssen die öffentlichen Schlüssel vor Fälschung und Missbrauch geschützt werden. Die Verteilung und Speicherung der öffentlichen und privaten Schlüssel geschieht in *Public-Key-Infrastrukturen* (PKI). In diesem Kapitel werden einige Organisationsprinzipien von Public-Key-Infrastrukturen erläutert. Mehr Details findet man in [21].

16.1 Persönliche Sicherheitsumgebung

16.1.1 Bedeutung

Will Bob mit einem Public-Key-System Signaturen erzeugen oder verschlüsselte Nachrichten entschlüsseln, braucht er einen privaten Schlüssel. Dieser Schlüssel muss geheim bleiben, weil jeder, der ihn kennt, Bobs Signatur fälschen oder geheime Nachrichten an Bob entschlüsseln kann. Die Umgebung, in der Bob seinen geheimen Schlüssel ablegt, wird *persönliche Sicherheitsumgebung* oder *Personal Security Environment (PSE)* genannt. Optimal ist es, wenn der private Schlüssel die PSE nicht verlässt, weil er sonst in eine unsichere Umgebung kommt. Dann entschlüsselt die PSE Chiffretexte und signiert Doumente, weil dazu der private Schlüssel nötig ist.

Oft werden in der PSE auch die privaten Schlüssel erzeugt. Werden sie anderswo erzeugt, dann sind sie wenigstens dem Erzeuger bekannt. Andererseits ist es bei der Schlüsselerzeugung besonders wichtig, dass keine technischen Fehler gemacht werden. Beispielsweise erfordert die Schlüsselerzeugung im RSA-Verfahren die zufällige Wahl zweier Primzahlen p und q. Ist der Zufallszahlengenerator der PSE schwach, so lassen

J. Buchmann, *Einführung in die Kryptographie*, Springer-Lehrbuch,
DOI 10.1007/978-3-642-39775-2_16

sich die verwendeten Primzahlen vielleicht rekonstruieren. Das kann dafür sprechen, die Schlüssel in einer vertrauenswürdigen Stelle zu erzeugen, wo der jeweils beste bekannte Zufallszahlengenerator verwendet wird.

16.1.2 Implementierung

Je sensibler die Dokumente sind, die verschlüsselt oder signiert werden, desto sicherer muss die PSE ausgelegt sein. Eine einfache PSE ist ein durch ein Passwort geschützter Speicherbereich auf der Festplatte von Bobs Computer. Sie wird *Software-PSE* genannt. Dieses Passwort kann zum Beispiel dazu verwendet werden, die Information in der PSE zu entschlüsseln. Die Sicherheit dieser PSE hängt sehr stark von der Sicherheit des verwendeten Betriebssystems ab. Man kann argumentieren, dass Betriebssysteme ohnehin hohe Sicherheit gewährleisten müssen und daher auch die PSE schützen können. Betriebssysteme müssen zum Beispiel verhindern, dass sich ein Unbefugter Administrator-Rechte verschafft. Andererseits ist bekannt, dass man mit entsprechendem Aufwand viele Schutzfunktionen des Betriebssystems umgehen kann und eine Software-PSE daher nicht sicher ist.

Für sehr sicherheitskritische Anwendungen reicht der Schutz durch das Betriebssystem nicht. Sicherer ist es, die PSE auf einer Chipkarte unterzubringen. Bob kann seine Karte immer bei sich haben. Wenn die Karte im Leser steckt, erlaubt sie nur sehr eingschränkte Zugriffe von außen. Die Manipulation ihrer Hard- oder Software ist extrem schwierig. Leider sind Berechnungen auf Chipkarten langsam. Darum lassen sich große Datenmengen auf einer Chipkarte nicht in vertretbarer Zeit ver- oder entschlüsseln. Man verschlüsselt daher mit dem öffentlichen Schlüssel des Kommunikationspartners nur Sitzungsschlüssel, die dann zur Verschlüsselung von Dokumenten verwendet und in verschlüsselter Form den Schlüsseltexten angehängt werden (siehe Abschn. 8.1). Der Sitzungsschlüssel wird auf der Chipkarte entschlüsselt. Die Entschlüsselung des gesamten Textes erfolgt dann in einem leistungsfähigen Computer. Entsprechend signieren Chipkarten nur vorberechnete Hashwerte.

16.1.3 Darstellungsproblem

Selbst wenn Alice zum Signieren eine Chipkarte verwendet, treten ernste Sicherheitsprobleme auf: Alice will ein Dokument signieren. Sie benutzt dazu ein Signierprogramm. Es schickt den Hashwert des zu signierenden Dokumentes an die Chipkarte. Dort wird die Signatur berechnet. Angenommen, der Angreifer Oskar kann den PC von Alice so manipulieren, dass ein anderes Dokument, als Alice glaubt, an die Chipkarte geschickt und dort signiert wird. Alice sieht nicht, was signiert wird, weil die Chipkarte keine Anzeige hat. Sie sieht nur das Dokument von dem sie glaubt, dass es signiert werde. Dies ist als *Darstellungsproblem* für Signaturen bekannt.

16.2 Zertifizierungsstellen

In Public-Key-Systemen ist es nicht nur wichtig, dass Alice ihre privaten Schlüssel schützen kann. Benutzt sie den öffentlichen Schlüssel von Bob, muss sie sicher sein, dass sie tatsächlich Bobs öffentlichen Schlüssel hat. Gelingt es nämlich dem Angreifer Oskar, Bobs öffentlichen Schlüssel gegen seinen eigenen auszutauschen, dann kann er Nachrichten entschlüsseln, die für Bob bestimmt waren, und er kann in Bobs Namen digitale Signaturen erzeugen.

Eine Möglichkeit zu garantieren, dass Teilnehmer in IT-Netze die richtigen öffentlichen Schlüssel der anderen Teilnehmer erhalten, besteht darin, dass eine *Zertifizierungsstelle* oder *Certification-Authority* (CA) mit ihrer Signatur bestätigt, dass ein öffentlicher Schlüssel zu einem Teilnehmer gehört. Wir erklären im Folgenden, wie das im einzelnen funktioniert.

16.2.1 Registrierung

Wenn Bob neuer Benutzer des Public-Key-Systems wird, lässt er sich bei der ihm zugeordneten CA registrieren. Er teilt der CA seinen Namen und andere erforderliche persönliche Daten mit. Die CA muss Bobs Informationen verifizieren. Am einfachsten ist es, wenn Bob persönlich zu der CA geht und seinen Ausweis vorlegt. Die CA weist Bob einen geeigneten Benutzernamen zu, der sich von allen anderen Benutzernamen unterscheidet. Unter diesem Namen wird Bob zum Beispiel Signaturen erzeugen. Wenn Bob nicht will, dass sein wirklicher Name bekannt wird, kann er auch ein Pseudonym verwenden. Dann ist nur der CA Bobs wirkliche Identität bekannt.

16.2.2 Schlüsselerzeugung

Bobs öffentliche und private Schlüssel werden in seiner PSE oder von seiner CA erzeugt. Es ist vorteilhaft, wenn Bob seine privaten Schlüssel nicht kennt. Dann kann er sie auch nicht preisgeben. Geschieht die Schlüsselerzeugung in Bobs PSE, so werden Bobs private Schlüssel in der PSE gespeichert. Die öffentlichen Schlüssel werden der CA mitgeteilt. Werden die Schlüssel von der CA erzeugt, so müssen die privaten Schlüssel auf Bobs PSE gelangen. Die Übertragung der Schlüssel muss natürlich entsprechend gesichert sein.

Für jeden Zweck, etwa Signieren, Verschlüsseln und Authentifizieren, muss ein eigenes Schlüsselpaar erzeugt werden, weil sonst die Sicherheit des Systems gefährdet ist. Das illustriert das folgende Beispiel.

Beispiel 16.1 Wenn Alice für Challenge-Response-Authentifikation und Signatur dasselbe Schlüsselpaar verwendet, dann können ihre Signaturen folgendermaßen gefälscht werden: Oskar gibt vor, er wolle die Identität von Alice prüfen. Als Challenge schickt er

Alice den Hashwert $h(m)$ eines Textes m. Alice signiert diesen Hashwert im Glauben, es handele sich um eine Zufallszahl. Der signierte Hashwert ist aber die Signatur des Textes m. Alice hat also, ohne es zu merken, ein Dokument signiert, das ihr von Oskar vorgelegt wurde.

16.2.3 Zertifizierung

Um eine verifizierbare Verbindung zwischen Bob und seinen öffentlichen Schlüsseln herzustellen, erstellt die CA ein *Zertifikat*. Dieses Zertifikat ist ein von der CA signiertes Dokument, das mindestens folgende Informationen enthält:

1. Bobs Benutzernamen oder sein Pseudonym,
2. die zugeordneten öffentlichen Schlüssel,
3. die Bezeichnung der Algorithmen, mit denen die öffentlichen Schlüssel von Bob benutzt werden können,
4. die laufende Nummer des Zertifikats,
5. Beginn und Ende der Gültigkeit des Zertifikats,
6. den Namen der ausstellenden CA,
7. Angaben, ob die Nutzung der Schlüssel auf bestimmte Anwendungen beschränkt ist.

16.2.4 Archivierung

Einige Schlüssel, die in Public-Key-Systemen verwendet werden, müssen archiviert werden. Die Dauer der Aufbewahrung hängt von der Verwendung der Schlüssel ab.

Öffentliche Signaturschlüssel müssen solange aufbewahrt werden, wie die entsprechenden Signaturen noch verifizierbar sein sollen. Signaturen in elektronischen Grundbüchern müssen zum Beispiel viele Jahrzehnte gültig sein. Für Signaturschlüssel werden Zertifikate gespeichert, damit später ihre Authentizität noch verifiziert werden kann. Private Entschlüsselungsschlüssel müssen solange aufbewahrt werden, wie die entsprechenden Chiffretexte noch entschlüsselbar sein sollen. Private Schlüssel werden aber nicht von der CA, sondern in der PSE der Benutzer archiviert. Für Authentifikationsschlüssel, private Signaturschlüssel und öffentliche Verschlüsselungsschlüssel ist keine Archivierung erforderlich. Sie werden nur solange benötigt, wie sie zur Authentifikation, zum Signieren oder zum Verschlüsseln gebraucht werden.

16.2.5 Personalisierung der PSE

Nach erfolgreicher Registrierung, Schlüsselerzeugung und Zertifizierung überträgt die CA die für den Teilnehmer relevanten Daten in seine PSE. Das sind insbesondere Bobs private

Schlüssel, sofern sie von der CA erzeugt wurden. Außerdem kann Bobs Zertifikat und der öffentliche Schlüssel der CA in der PSE gespeichert werden.

16.2.6 Verzeichnisdienst

Bei der CA gibt es ein Verzeichnis (Directory) der von der CA erzeugten Zertifikate mit den entsprechenden Teilnehmernamen. Will Alice die öffentlichen Schlüssel von Bob erfahren, so fragt sie bei dem Directory ihrer CA an, ob Bob dort registriert ist. Wenn ja, erhält sie Bobs Zertifikat. Da Alice den öffentlichen Schlüssel ihrer CA kennt, kann sie verifizieren, dass das Zertifikat von ihrer CA kommt und sie vertraut ihrer CA. Also hat sie den authentischen öffentlichen Schlüssel von Bob. Wenn Bob nicht bei der CA von Alice registriert ist, kann Alice versuchen, ein Zertifikat für Bob auf Umwegen zu erhalten. Das wird in Abschn. 16.3 genauer beschrieben.

Zertifikate, die Alice häufig braucht, kann sie in ihre PSE aufnehmen. Sie muss aber die Gültigkeit der Zertifikate regelmäßig überprüfen (siehe Abschn. 16.2.8).

Wenn eine CA für eine große Zahl von Benutzern zuständig ist, kann es bei den Anfragen an die CA Engpässe geben. Eine Möglichkeit, die Effizienz der Directory-Anfragen zu verbessern, besteht darin, das Directory zu replizieren und die Anfragen auf die verschiedenen Kopien aufzuteilen.

Beispiel 16.2 Eine Firma möchte alle 50.000 Mitarbeiter zu Teilnehmern einer PKI machen. Die Firma ist über fünf Länder verteilt. Sie möchte nur eine CA in England betreiben. Um die Anfrage beim Directory der CA effizienter zu gestalten, verteilt sie fünf Kopien des Directories auf die fünf Länder und führt selbst das Original. Die Kopien werden zweimal am Tag auf den neuesten Stand gebracht.

16.2.7 Schlüssel-Update

Alle Schlüssel, die in einem Public-Key-System benutzt werden, haben eine bestimmte Gültigkeitsdauer. Rechtzeitig, bevor ein Schlüssel seine Gültigkeit verliert, muss er durch einen neuen Schlüssel ersetzt werden. Dieser neue Schlüssel wird nach seiner Erzeugung von CA und Benutzer ausgetauscht. Der Austausch muss so ablaufen, dass der neue Schlüssel selbst dann nicht kompromittiert wird, wenn der alte Schlüssel bekannt wird.

Folgendes Verfahren, den Schlüssel zu erneuern, ist unsicher: Kurz bevor das Schlüsselpaar von Bob ungültig wird, erzeugt seine CA ein neues Schlüsselpaar für Bob. Sie verschlüsselt den neuen privaten Schlüssel mit Bobs altem öffentlichen Schlüssel und sendet ihn an Bob. Wenn Oskar diese Kommunikation protokolliert und wenn es ihm später gelingt, Bobs alten privaten Schlüssel zu finden, dann kann er auch den neuen privaten Schlüssel bestimmen. In diesem Fall hängt die Sicherheit des neuen Schlüssels von der

Sicherheit des alten Schlüssels ab, und es ist sinnlos, einen neuen Schlüssel auszutauschen.

Statt dessen kann man Varianten des Diffie-Hellman-Schlüsselaustauschs verwenden, der Man-In-The-Middle-Angriff unmöglich machen (siehe Abschn. 8.6).

16.2.8 Widerruf von Zertifikaten

Manchmal muss die CA ein Zertifikat für ungültig erklären, obwohl seine Gültigkeitsdauer noch nicht abgelaufen ist.

Beispiel 16.3 Bob hat seine Chipkarte bei einer Bootsfahrt verloren. Die Smartcard liegt jetzt auf dem Meeresgrund. Auf der Smartcard ist Bobs privater Entschlüsselungsschlüssel. Den kann er jetzt nicht mehr benutzen, weil der Schlüssel nur auf der Chipkarte gespeichert ist und sonst nirgendwo. Daher muss die CA das Zertifikat, das den entsprechenden öffentlichen Schlüssel enthält, für ungültig erklären.

Widerrufene Zertifikate schreibt die CA in eine Liste, die *Certificate Revocation List* (CRL). Diese ist Teil des Directories der CA. Ein Eintrag in der CRL enthält die Seriennummer des Zertifikats, den Zeitpunkt, zu dem das Zertifikat ungültig wurde und möglicherweise andere Informationen, wie z. B. die Gründe für die Ungültigkeit. Die CRL wird von der CA signiert.

16.2.9 Zugriff auf ungültige Schlüssel

Wird ein ungültiger, aber archivierter Schlüssel benötigt, wird dieser vom Archiv bereitgestellt. Der Benutzer, der diesen Schlüssel erhalten möchte, muss seine Berechtigung gegebenenfalls nachweisen.

Beispiel 16.4 Von der CA werden die Signaturschlüssel jeden Monat gewechselt. Bob signiert einen Auftrag an Alice, bestreitet aber drei Monate später, den Auftrag erteilt zu haben. Alice möchte beweisen, dass der Auftrag tatsächlich erteilt wurde. Sie braucht dafür den öffentlichen Schlüssel, der aber schon seit zwei Monaten ungültig ist und darum nicht mehr im Directory der CA steht.

16.3 Zertifikatsketten

Wenn Bob und Alice nicht zu derselben CA gehören, kann Alice den öffentlichen Schlüssel von Bob nicht einfach aus dem Directory ihrer CA erfahren. Sie kann den öffentlichen Schlüssel von Bob aber indirekt erhalten.

Beispiel 16.5 Alice ist bei der CA für Hessen registriert. Bob ist bei der CA für das Saarland registriert. Alice kennt also den öffentlichen Schlüssel der CA Hessen, aber nicht den öffentlichen Schlüssel der CA des Saarlandes. Von ihrer CA erhält Alice ein Zertifikat für den öffentlichen Schlüssel der CA des Saarlandes. Aus dem Directory der CA des Saarlandes erhält Alice ein Zertifikat für den öffentlichen Schlüssel von Bob. Unter Verwendung des öffentlichen Schlüssels ihrer CA kann Alice verifizieren, dass der öffentliche Schlüssel der CA des Saarlandes korrekt ist. Damit ist Alice also im Besitz des öffentlichen Schlüssels der CA des Saarlandes. Diesen Schlüssel kann sie nun verwenden, um das Zertifikat für Bobs öffentlichen Schlüssel zu verifizieren.

Wie in Beispiel 16.5 beschrieben, kann Alice eine *Zertifikatskette* verwenden, um den authentischen öffentlichen Schlüssel von Bob zu erhalten, selbst wenn Bob zu einer anderen CA gehört. Formal kann man eine solche Kette so beschreiben: Für eine Zertifizierungsstelle CA und einen Teilnehmer U bezeichne $\mathrm{CA}\{U\}$ ein Zertifikat, das den Namen von U an den öffentlichen Schlüssel von U bindet. Dabei kann U entweder ein Benutzer oder eine CA sein. Eine Zertifikatskette, die für Alice den öffentlichen Schlüssel von Bob zertifiziert, ist dann eine Folge

$$\mathrm{CA}_1\{\mathrm{CA}_2\}, \mathrm{CA}_2\{\mathrm{CA}_3\}, \ldots, \mathrm{CA}_{k-1}\{\mathrm{CA}_k\}, \mathrm{CA}_k\{\mathrm{Bob}\}.$$

Dabei ist CA_1 eine CA, deren Schlüssel Alice kennt. Alice verifiziert diese Zertifikatskette, indem sie mit dem öffentlichen Schlüssel von CA_1 das erste Zertifikat prüft und dabei den öffentlichen Schlüssel von CA_2 erhält, mit dem öffentlichen Schlüssel von CA_2 das Zertifikat für CA_3 prüft usw. und zum Schluss mit dem öffentlichen Schlüssel von CA_k das Zertifikat für Bob prüft.

Dieses Verfahren setzt voraus, dass Vertrauen transitiv ist, d. h. wenn sich U_1 auf U_2 verlässt und U_2 auf U_3, dann traut U_1 auch U_3.