# Lektion 3

## Asymmetrische Kryptosysteme

#### LERNZIELE

Nach der Bearbeitung dieser Lektion kennen Sie …

… die Vorteile und Grenzen der asymmetrischen Kryptografie.

… Falltürfunktionen und modulare Arithmetik.

… die gängigsten asymmetrischen kryptografischen Algorithmen und die ihnen zugrunde liegenden Falltürfunktionen.

DL-E-DLMCSEAITSC02-U03

1. Asymmetrische Kryptosysteme

### Einführung

Das große praktische Problem der Kryptografie mit nur einem Schlüssel ist die Schlüsselverteilung, d. h. die geheime Weitergabe desselben Schlüssels an alle Kommunikationsteilnehmer, die normalerweise weit voneinander entfernt sind, bevor sie sicher kommunizieren können. Problematisch ist zudem die große Anzahl von Schlüsseln, die eine Gruppe von Kommunikationsteilnehmern für eine sichere Kommunikation benötigt.

1976 fanden Difﬁe und Hellman heraus, dass das Problem der Schlüsselverteilung durch einen Algorithmus gelöst werden kann, der (rechnerisch) folgende Bedingungen erfüllt:

* + einfache Erstellung eines passenden Schlüsselpaares für die Ver- und Entschlüsselung,
  + einfache Verschlüsselung und Entschlüsselung,
  + unmögliche Wiederherstellung eines der Schlüssel trotz Kenntnis des Algorithmus, des anderen Schlüssels und einer beliebigen Anzahl übereinstimmender Klartext- und Geheimtext-Paare, und
  + unmögliche Wiederherstellung des Klartextes für nahezu alle Schlüssel k und Nachrichten x.

Als Benutzerin dieses Algorithmus hält Alice ihren Entschlüsselungsschlüssel geheim, macht aber ihren Verschlüsselungsschlüssel öffentlich, zum Beispiel in einem öffentlichen Verzeichnis. Jeder, der sicher mit Alice kommunizieren möchte, braucht nur ihren öffentlichen Schlüssel nachzuschlagen, um ihr einen Geheimtext zu senden, den nur sie entschlüsseln kann, d. h. eine Nachricht kann verschlüsselt werden, ohne dass eine Geheimhaltung erforderlich ist. Jeder, der den entsprechenden öffentlichen Schlüssel verwendet, kann einen mit dem geheimen Schlüssel von Alice verschlüsselten Geheimtext entschlüsseln, d. h. ein Absender kann identifiziert werden, ohne dass eine Geheimhaltung erforderlich ist.

Die Sicherheit kryptografischer Algorithmen mit zwei Schlüsseln hängt von der Schwierigkeit der Berechnung eines mathematischen Problems ab, z. B. der Faktorisierung einer Zahl, die das Produkt aus zwei großen Primzahlen ist. Im Idealfall ist die Berechnung des geheimen Schlüssels gleichbedeutend mit der Lösung des schwierigen Problems, so dass der Algorithmus mindestens so sicher ist, wie das zugrunde liegende mathematische Problem schwierig ist. Dies wurde für keinen der Standardalgorithmen bewiesen, obgleich angenommen wird, dass es für jeden von ihnen gilt.

###### Arithmetische Falltüren

Um die Nützlichkeit der (modularen) Arithmetik in der Kryptografie zu erkennen, erinnern wir uns daran, dass die asymmetrische Kryptografie auf einer Falltürfunktion basiert, die leicht berechenbar sein muss, deren Umkehrung aber ohne Kenntnis eines geheimen Schlüssels praktisch unberechenbar sein muss.

Die Einfachheit der Berechnung der Funktion entspricht der Einfachheit der Verschlüsselung, während die Schwierigkeit der Berechnung der Umkehrung der Schwierigkeit der Entschlüsselung, d. h. der Umkehrung der Verschlüsselung, entspricht. RSA verwendet zum Beispiel als Verschlüsselungsfunktion die Erhöhung auf eine n-te Potenz und als Entschlüsselungsfunktion deren Umkehrung, das Wurzelziehen.

Während sowohl die Funktion selbst als auch ihre Umkehrung leicht mit der üblichen Multiplikation von Zahlen berechnet werden können, verwenden kryptografische Algorithmen (wie RSA) die modulare Arithmetik, um die Berechnung der Umkehrfunktion ohne Kenntnis des Schlüssels komplexer zu gestalten.

Asymmetrische Kryptosysteme

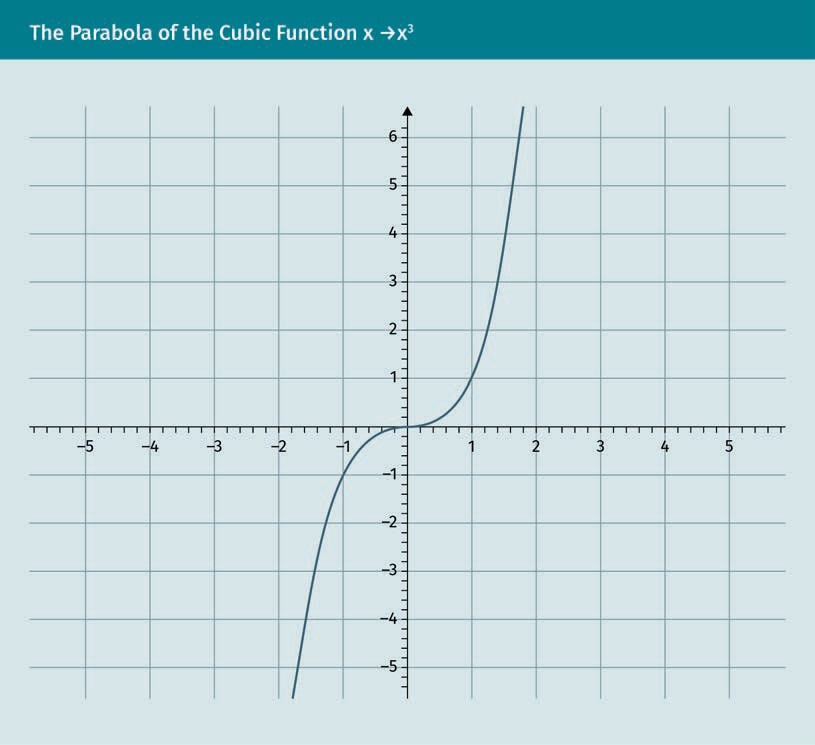
Die modulare oder zirkuläre Arithmetik kennen wir bereits aus der Arithmetik der Uhr, bei der m = 12 als gleich 0 angesehen wird. Da der Zeiger nach jeder Umdrehung wieder bei 0 beginnt zu zählen, ist es zum Beispiel drei Stunden nach 11 Uhr zwei Uhr, 11 + 3 = 14 = 12 + 2 = 2. In diesen Definitionsbereichen, die man als infinite Ringe bezeichnet, werden (die Graphen der) Funktionen unregelmäßig und praktisch unberechenbar, zumindest ohne Kenntnis einer Abkürzung (des Schlüssels).

Modulare Arithmetik als Randomisierung

Die Schwierigkeit, die Umkehrung zu berechnen, entspricht der Schwierigkeit der Entschlüsselung, d. h. der Umkehrung der Verschlüsselung. Bei einem asymmetrischen kryptografischen Algorithmus basieren die Einfachheit der Verschlüsselung einer Zahl und die Schwierigkeit der Entschlüsselung einer Zahl auf einer invertierbaren Funktion, so dass diese zwar leicht berechenbar ist, aber sich ihre Umkehrfunktion nur schwer berechnen lässt. Zum Beispiel sind die Umkehrungen der Falltürfunktionen Potenzierung x /xe (im RSA-Algorithmus) und Exponentiation x /gx (im Difﬁe-Hellman-Algorithmus) durch das Wurzelziehen x /x1/e und den Logarithmus log*g* gegeben. Wir können sie im Definitionsbereich der reellen Zahlen 0 leicht berechnen (z. B. mit der Bisektionsmethode für stetige Funktionen dank der Verknüpfung von 0), aber in ihren kryptografischen Definitionsbereichen sind sie nahezu unberechenbar. Wir wollen diese finiten Definitionsbereiche nun einführen.

Funktionen auf diskreten Mengen

Ihr Definitionsbereich ist nicht die Menge der ganzen Zahlen + (oder die der reellen Zahlen 0 , die sie einschließt), weil beide Funktionen, Exponentiation und Potenzierung, stetig über 0 sind.



Wenn der Definitionsbereich dieser Funktionen 0 wäre, könnten ihre Umkehrungen über 0 angenähert werden, zum Beispiel durch iterative Bisektion, bei welcher der Umkehrpunkt in Intervallen eingeschlossen wird, die bei jeder Iteration halbiert werden.

Endlicher Ring Dies ist eine endliche Menge, die 0 und 1 enthält und über die eine Summe erklärt ist, die dem Assoziativgesetz und dem Kommutativgesetz

gehorcht.

Endliche Ringe

Um die iterative Annäherung an die Null zu vermeiden und damit die Berechnung der Umkehrfunktion zu erschweren (neben der Erleichterung der Berechnung der eigentlichen Funktion), ist der Definitionsbereich einer Falltürfunktion ein endlicher Ring, angegeben durch

+/m+ = 0,1, . . . , m − 1

für eine natürliche Zahl m. In solch einem endlichen Ring gilt

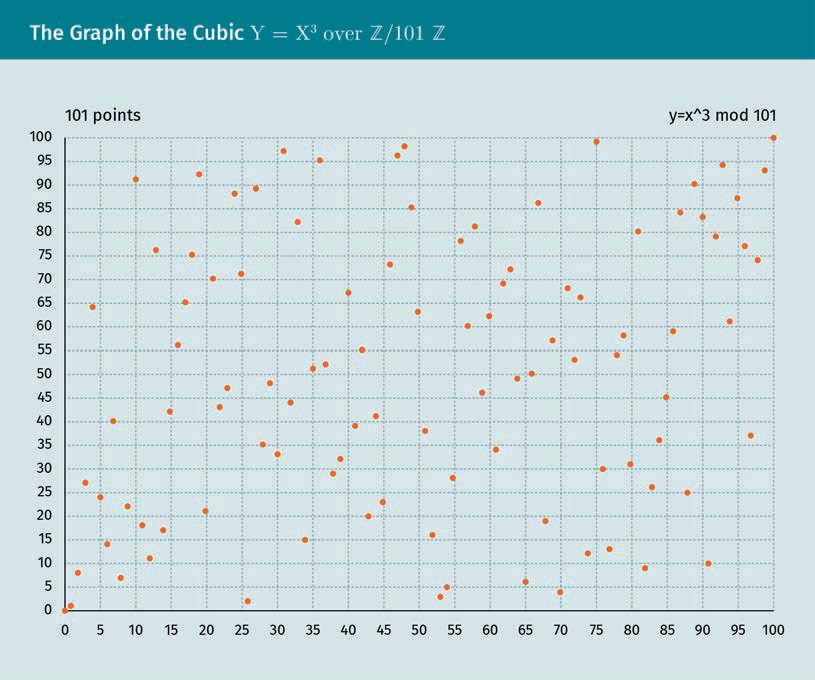
m = 1 + ! + 1 = 0

Asymmetrische Kryptosysteme

und notwendigerweise hat jede Addition (und damit jede Multiplikation und jede Potenzierung) ein Ergebnis < m. Also ist + / m + seine Addition anders als die von + (oder 0). Für m = 7 erhalten wir z. B.

22 = 2 · 2 = 4 und 32 = 3 · 3 = 7 + 2 = 0 + 2 = 2 .

Wir werden diese endlichen Ringe zunächst anhand der Beispiele + / 12 + und + / 7 + (der Ring, der durch die Zahlen der Stunden einer Uhr und der Wochentage gegeben ist) einführen, dann für jedes m.



Wenn wir uns die Graphen der Funktionen ansehen, die auf 0 so regelmäßig sind, stellen wir fest, dass die Parabel über dem endlichen Ring + /101 + zunächst genauso regelmäßig über + / 101 + ist wie über +. Ab x = 11 (wegen 112 = 121 > 100) beginnt sie sich jedoch unregelmäßig zu verhalten (mit Ausnahme der Symmetrie der Parabel auf der Mittelachse x = 50.5 aufgrund der Gleichheit ((–x)2 = x2).

Zwei ganze Zahlen a und b sind kongruent modulo m

a ≡ b mod m

wenn m | a – b, das heißt, wenn ihre Differenz a – b durch m teilbar ist.

Kongruenz

Zwei ganze Zahlen a und b sind kongruent modulo m, wenn sie nach der Division durch m den gleichen Rest übriglassen.

Die Zahl m ist der Modul. Oder anders formuliert, a ≡ b mod m , wenn a und

b beim Dividieren durch m den gleichen Rest lassen.

### Der Difﬁe-Hellman-Schlüsselaustausch

Difﬁe-Hellman-Schlüssel-

austausch Dieses Protokoll war eine offene Vereinbarung über einen gemeinsamen geheimen Schlüssel, dessen Sicherheit auf der Unmöglichkeit beruhte, den Logarithmus modulo einer großen Zahl zu berechnen.

Das erste veröffentlichte Protokoll, in dem ein gegenseitiger geheimer Schlüssel offen vereinbart wurde, ist der Difﬁe-Hellman-Schlüsselaustausch (Difﬁe & Hellman, 1976). Dies ist noch keine Zwei-Schlüssel-Kryptografie, da beide Korrespondenten den einzigen geheimen Schlüssel kennen. Die asymmetrischen kryptografischen Algorithmen, die auf diesem Protokoll aufbauen (z. B. Elgamal und ECC), erzeugen für jede Nachricht einen eindeutigen Schlüssel.

###### Schlüsselaustauschprotokoll

Bezeichnen wir in jeder asymmetrischen Verschlüsselung eine öffentliche Zahl mit einem Großbuchstaben und eine geheime Zahl mit einem Kleinbuchstaben. Damit sich zwei Parteien, Alice und Bob, offen auf einen geheimen Schlüssel einigen können, kombinieren sie zunächst eine geeignete Primzahl p (der Modul) und eine geeignete natürliche Zahl g (die Basis).

Zunächst generiert Alice eine Hälfte des Schlüssels, indem sie (1) eine Zahl a wählt, (2) A

≡ga mod p berechnet und (3) A an Bob überträgt. Danach generiert Bob die andere Hälfte des Schlüssels, indem er (1) eine Zahl b wählt, (2) B ≡gb mod p berechnet und (3) B an Alice übermittelt. Der gemeinsame geheime Schlüssel zwischen Alice und Bob lautet

b a

b a aab ba b

c ( A mod p = g mod p mod p = g mod p = g mod p = g mod p mod p = B

mod p .

Dieses Protokoll zeigt, wie man einen gemeinsamen geheimen Schlüssel offen erstellt. Dieser Schlüssel kann dann zur Verschlüsselung der gesamten weiteren Kommunikation verwendet werden, so wie AES. Er zeigt jedoch weder, wie man eine Nachricht verschlüsselt, noch wie man sie signiert.

###### Sicherheit

Die Sicherheit des Difﬁe-Hellman-Schlüsselaustauschs beruht auf der Schwierigkeit, den Logarithmus modulo p zu berechnen. Ein Abhörender würde den geheimen Schlüssel Ab = Ba aus A und B erhalten, wenn er folgendes berechnen könnte:

a = loggA oder b = loggB mod p .

Während eine Potenz leicht berechenbar ist, erst recht in der modularen Arithmetik, ist ihre Umkehrung, der Logarithmus, also der Exponent für eine bestimmte Potenz, bei geeigneter Wahl von p und g praktisch nicht berechenbar.

Asymmetrische Kryptosysteme

###### Logarithmus

Da zu Beginn (für x < logg p) die Werte gx über + / p + gleich den Werten gx über + sind, müssen die geheimen Zahlen a und b ausreichend groß sein, d. h. > logg p. Um dies sicherzustellen, werden diese Zahlen in der Praxis künstlich erhöht, das heißt, die Nachricht wird aufgefüllt.

Der derzeit schnellste Algorithmus zur Berechnung des Logarithmus x aus gx ist eine Adaption des allgemeinen Zahlenfeldsiebs (Gordon, 1993). Die Anzahl der Operationen zur Berechnung des Logarithmus einer ganzen Zahl mit n Bits entspricht grob

exp logn1/3 .

### Der RSA-Algorithmus

Der bekannteste Public-Key-Algorithmus ist der Rivest-Shamir-Adleman-Kryptoalgorithmus (kurz (RSA-Algorithmus) (Rivest et al., 1978). Ein Benutzer wählt im Geheimen ein Paar Primzahlen p und q , die so groß sind, dass die Faktorisierung des Produkts N = p q die geschätzte Rechenleistung für die gesamte Nutzungsdauer der Chiffre übersteigt. Die Zahl N wird der Modul sein, d. h. unsere Falltürfunktion basiert auf + / N + = { 0, 1, …, N-1 }.

N ist öffentlich, p und q jedoch nicht. Wenn die Faktoren p und q von N bekannt wären, dann könnte der geheime Schlüssel leicht berechnet werden. Damit RSA sicher ist, darf die Faktorisierung rechnerisch nicht durchführbar sein, was derzeit 2048 Bit entspricht. Der Schwierigkeitsgrad der Faktorisierung verdoppelt sich ungefähr für jeweils drei zusätzliche Ziffern in N.

Der RSA-Algorithmus erzeugt einen öffentlichen Schlüssel zum Verschlüsseln und einen privaten Schlüssel zum Entschlüsseln. Im Vergleich zum Difﬁe-Hellman-Protokoll hat dieser Algorithmus den Vorteil, dass er vollständig asymmetrisch ist. Es besteht keine Notwendigkeit, einen gemeinsamen geheimen Schlüssel zu teilen (und der geheime Schlüssel wird nur an einem einzigen Ort aufbewahrt). Stattdessen hat ein einziger Kommunikationspartner Zugriff auf den geheimen Schlüssel. In diesem Fall wird die Kommunikation jedoch nur gegenüber dem Besitzer des geheimen Schlüssels verschlüsselt. Um in beide Richtungen zu verschlüsseln, erstellt entweder jeder Kommunikationspartner einen asymmetrischen RSA-Schlüssel, oder der andere Kommunikationspartner verschlüsselt und sendet einen symmetrischen Schlüssel (bekannt als hybride Verschlüsselung, da sie ein asymmetrisches mit einem symmetrischen Kryptosystem kombiniert).

###### Zahlentheorie

Mit der Eulerschen Formel seien p und q verschiedene Primzahlen. Ist a nun weder durch p noch durch q teilbar, dann

a p − 1 q − 1 ≡ 1 mod pq .

RSA

Dieser Algorithmus verschlüsselt durch Potenzierung. Seine Sicherheit beruht auf der rechnerischen Unmöglichkeit, ein Produkt aus Primzahlen zu faktorisieren.

Wurzeln ziehen

Seien p und q verschiedene Primzahlen, N = p q und φ(N) = (p – 1)(q – 1). Für jeden Exponenten n so dass

n ≡ 1 mod 3 N

haben wir

an ≡ a mod N für jede ganze Zahl a .

Wenn m ≡1 mod φ(N), dann gilt nach der Eulerschen Formel am ≡a mod N, d. h. die Potenzierung ist die Identitätsfunktion,

m· ≡ id mod N .

Insbesondere, wenn m = E d das Produkt zweier ganzer Zahlen E und d ist, d. h.

Ed ≡ 1 mod 3 N ,

dann

a = am = aEd = aE d .

Das heißt • d = • 1/E mod N. Eine Potenz zu berechnen ist viel einfacher als eine Wurzel!

Beispiel

Wenn p = 3 und q = 11 dann

N = pq = 33 und 3 N = p − 1 q − 1 = 20 .

Wenn E = 7 und d = 3, dann n = Ed = 21 ≡1 mod 20. Für die Basis 2 prüfen wir zum Beispiel

2E = 27 = 128 = 29 + 3 · 33 ≡ 29 mod N

und

29d = 293 = − 4 3 ≡ − 64 = 2 − 2 · 33 ≡ 2 mod N .

Somit

E 29 = 2 = 29d mod N .

Asymmetrische Kryptosysteme

Euklidischer Algorithmus

Ein gemeinsamer Teiler von zwei ganzen Zahlen a und b ist eine natürliche Zahl, die diese beiden Zahlen teilt. Der größte gemeinsame Teiler von zwei ganzen Zahlen ist die größte natürliche Zahl, die diese beiden teilt. Der größte gemeinsame Teiler von 12 und 18 ist zum Beispiel 6. Bezeichnen wir mit mdc(a,b) den größten gemeinsamen Teiler von a und b,

mdc a, b = die größte natürliche Zahl, die a und b teilt.

Die wiederholte Division mit Rest ergibt einen efﬁzienten Algorithmus zur Berechnung des größten gemeinsamen Teilers, d. h. den euklidischen Algorithmus.

Erweiterter Euklidischer Algorithmus

Für die Berechnung des Exponenten der Entschlüsselungsfunktion benötigen wir mehr Informationen als den größten gemeinsamen Teiler (berechnet durch den euklidischen Algorithmus). Im erweiterten Algorithmus stellt man fest, dass in jedem Schritt des euklidischen Algorithmus der größte gemeinsame Teiler mdc(x, m) von x und m eine Linearkombination (oder Summe von Vielfachen) von x und m ist, d. h.

mdc x, m = λx + µm für die ganzen Zahlen x und m .

Als Beispiel nehmen wir mdc(528, 220) = 44 und tatsächlich

44 = 5 · 748 − 7 · 528 .

###### Verschlüsselungsalgorithmus

Wir erinnern uns, dass ein Großbuchstabe für eine öffentliche Zahl steht (und umgekehrt), während ein Kleinbuchstabe für eine geheime Zahl steht. Wir sehen uns zwei Kommunikationspartner an: Alice sendet die Nachricht m über einen unsicheren Kanal heimlich an Bob.

1. Für die Erzeugung des Schlüssels wählt Bob zwei Primzahlen p und q sowie einen Exponenten E,

der teilerfremd ist zu φ(N) := (p – 1)(q – 1).

1. Zur Übermittlung des Schlüssels sendet Bob Alice das Produkt N := pq (den Modul) und den Exponenten E (den öffentlichen Schlüssel).
2. Zur Verschlüsselung berechnet Alice M = mE mod N und übermittelt M an Bob.
3. Zur Entschlüsselung berechnet Bob (mit dem erweiterten euklidischen Algorithmus) d so, dass Ed

≡1 mod (p – 1)(q – 1) (da E teilerfremd ist zu φ(N)), und berechnet Md = mEd = m mod N (mit der Eulerschen Formel).

Zusammenfassend lässt sich sagen, dass die Potenzierung y = xE mod N verschlüsselt, wobei der Exponent E der öffentliche Schlüssel ist. Dementsprechend entschlüsseln wir mit der Umkehrung, dem Ziehen der E-ten Wurzel x = y1/E mod N. Sie ist praktisch nicht berechenbar. Aber modulo N, mit der Eulerschen Formel, gibt es d, so dass

y1/E = yd mod N

für eine Zahl d, die der euklidische Algorithmus aus E sowie p und q berechnet. Der geheime Schlüssel ist also d, oder, hinreichend genau, die Kenntnis der Primfaktoren p und q von N.

###### Sicherheit

Da der Modul N, der Exponent E und die verschlüsselte Nachricht M (= mE) alle öffentlich sind, basiert die rechnerische Sicherheit von RSA allein auf der Schwierigkeit, eine Wurzel aus einer großen Zahl zu finden:

m ≡ E M = M1/E mod N .

### Elliptische Kurven

Wir schreiben IFp := + / p +. Unter allen Kurven besteht der entscheidende Vorteil der elliptischen Kurven (gegeben durch die Gleichung y2 = x3 + ax + b) darin, dass man auf ihnen Punkte hinzufügen kann: p + q + r = 0 wenn eine Linie durch p*,* q und r verläuft.

Difﬁe-Hellman über elliptischen Kurven

Ein Analogon des Difﬁe-Hellman-Protokolls, in dem die iterative Multiplikation einer Zahl modulo p durch die iterative Addition eines Punktes auf einer elliptischen Kurve ersetzt wird. Das Difﬁe-Hellman-Protokoll (über IFp) hat ein Analogon über elliptischen Kurven. Anstatt wiederholt (n-mal) die Basis g in IF \* zu *multiplizieren*, d. h. zu berechnen

Es ist von Vorteil, den Logarithmus über einer endlichen elliptischen Kurve zu verwenden (die Funktion, die für einen gegebenen Punkt G und Y den Skalar x in 5 so bestimmt, dass Y = x G) statt des Logarithmus über IFp (die Funktion, die bei gegebenen Zahlen g und y den Exponenten x so bestimmt, dass y ≡gx mod p). Abhängig von der Anzahl der Bits n von p (in Bezug auf die schnellsten derzeit bekannten Algorithmen), steigt die Zeit zur Berechnung des Logarithmus über einer elliptischen Kurve linear an und benötigt etwa n/2 Operationen, während die Zeit zur Berechnung des multiplikativen Logarithmus sublinear ansteigt und etwa n1/3 Operationen benötigt.

Zum Beispiel entspricht die Sicherheit eines 2048-Bit-Schlüssels für den multiplikativen Logarithmus ungefähr der eines 224-Bit-Schlüssels für den Logarithmus über einer elliptischen Kurve. Ein 512-Bit-Schlüssel für eine elliptische Kurve entspricht einer Länge von 15360 Bit eines RSA-Schlüssels.

###### Elliptische Kurven

Eine elliptische Kurve E über einem endlichen Körper (in dem 0 ≠2, 3) ist eine Gleichung

y2 = x3 + ax + b

Asymmetrische Kryptosysteme

für die Koeffizienten a und b so, dass die Kurve nicht singulär ist, d. h. ihre Diskriminante ist ungleich Null, 4 a3 + 27 b2 ≠ 0.

Die Gleichung y2 = x3 + ax + b ist die Weierstraß-Normalform, aber es gibt einige andere, die sich als rechnerisch efﬁzienter erwiesen haben, wie Montgomery

By2 = x3 + Ax2 + x mit B A2 − 4 ≠ 0 .

Ist die Charakteristik 2, also IFq mit q = 2n, dann lautet die Gleichung y2 + cxy + dy = x3

+ ax + b.

Nach der Wahl eines Definitionsbereichs (z. B. +, \*, 0, 6 oder IFp für eine Primzahl p) bilden die Punkte (x, y) , die diese Gleichung lösen, E(x,y) = 0, eine Kurve in der Ebene. Diese Ebene, 0 steht für die übliche kartesische Ebene, + steht für ein Gitter von Punkten und + / m + steht für das endliche Gitter von Punkten innerhalb des Quadrats der Länge m , dessen untere linke Ecke im Ursprung liegt.

Zusätzlich zu den Punkten in der Ebene gibt es auch den Punkt in der Unendlichkeit (oder Idealpunkt), der mit 0 bezeichnet wird. Die Punkte der elliptischen Kurve sind also gegeben durch

E: = alle Punkte x, y so dass E x, y = 0 ∪ 0

wobei der Begriff des Punktes vom Definitionsbereich abhängt. Auf einem endlichen Körper IFq ist die Anzahl der Punkte # E begrenzt durch q + 1 – t mit t ≤ 2 √{q}, also asymptotisch gleich # IF \* = q –1. Der Algorithmus von Schoof (1999) kann dies in etwa n5 Operationen für n *=* log2 q die Anzahl der binären Ziffern von q berechnen.

q

Stetige und diskrete endliche Kurven

Für den Definitionsbereich 0 nehmen die Kurven in der Ebene der reellen Zahlen für verschiedene Parameter a

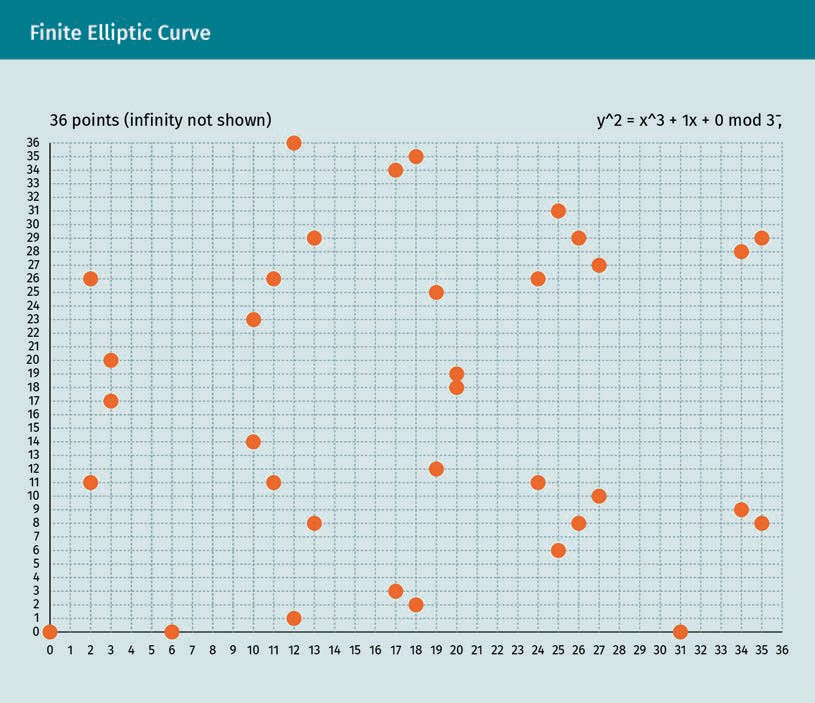
und b die folgenden Formen an:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Während wir auf endlichen Körpern eine diskrete Menge von Punkten erhalten (symmetrisch um die mittlere horizontale Achse).

Asymmetrische Kryptosysteme



Kurven in der Kryptografie

Damit der kryptografische Algorithmus auf dieser Kurve sicher ist, d. h. die Berechnung des Logarithmus auf ihr viel Zeit benötigt, gibt es Einschränkungen bei der Wahl von q = pn und der elliptischen Kurve (bei ihren definierenden Koeffzienten a und b). Als Richtwert gilt zum Beispiel, dass q ≥ 2224 , um bei einer Schlüssellänge von 2048 Bit genauso sicher zu sein wie RSA. Eine sichere Wahl ist zum Beispiel Curve25519, die durch

y2 = x3 + 486662x2 + x

über IFq mit q = p2 und p = 2255 – 19 (daher der Name) gegeben ist. Die Anzahl ihrer Punkte beträgt # E = 2252 + 277423177773585193790883648493. Diese Kurve wurde als neutrale Alternative zu den empfohlenen Kurven des NIST populär, denen man bald misstraute.

###### Schlüsselaustausch mit elliptischen Kurven

ECC verwendet den Difﬁe-Hellman-Schlüsselaustausch, um einen geheimen Schlüssel zu erstellen, ihn in einen kryptografischen Hash umzuwandeln und ihn zur Verschlüsselung der Kommunikation durch einen symmetrischen kryptografischen Algorithmus zu verwenden.

Die Verschlüsselung durch ECC ist durch das ECIES (Elliptic Curve Integrated Encryption Scheme), ein hybrides Verfahren (asymmetrische Kryptografie mit symmetrischer Kryptografie), standardisiert.

Sobald der gemeinsame geheime c-Schlüssel (ein Punkt auf der endlichen elliptischen Kurve) vereinbart ist, leiten Alice und Bob einen Schlüssel für eine symmetrische Chiffre wie AES oder 3DES ab. Die Ableitungsfunktion, die eine geheime Information in das entsprechende Format umwandelt, wird Schlüsselableitungsfunktion (Key Deviation Function KDF) genannt. Eine solche standardisierte Funktion ist ANSI-X9.63-KDF mit der Option SHA-1. Das TLS-Protokoll verwendet zum Beispiel die x-Koordinate des Punktes c, verknüpft sie mit Zahlen, die sich auf die Verbindung beziehen, und berechnet einen kryptografischen Hash dieser verknüpften Zahl.

Wir wollen nun das Difﬁe-Hellman-Protokoll von der Multiplikation auf einem endlichen Körper auf die Addition auf einer endlichen elliptischen Kurve übertragen. Bezeichne G einen Punkt auf der Kurve, und

xG = G + ! + G

die x-fache iterierte Addition über die endliche elliptische Kurve (anstelle von g und gx = g ·g für einen endlichen Körper).

Aufbau

Die kritische kryptografische Zahl ist die Ordnung n des Basispunktes G , die groß genug sein muss.

Hier ein Beispiel für einen Basispunkt. Die elliptische Kurve Curve25519 mit

y2 = x3 + 486662x2 + x

über IFq mit q = p2 mit p = 2255 – 19, verwendet als Basispunkt G = (xG, yG) eindeutig bestimmt durch

Schritte

1. Im ECDH-Protokoll (Elliptic Curve Difﬁe-Hellman) können Alice und Bob offen einen geheimen Schlüssel erstellen, indem sie eine Potenz q aus einer geeigneten Primzahl p , eine geeignete elliptische Kurve E über IFq und einen geeigneten Punkt G in E kombinieren.
2. Um eine Hälfte des Schlüssels zu generieren, wählt Alice eine Zahl a, berechnet A = a G und

übermittelt A an Bob.

1. Um eine weitere Hälfte des Schlüssels zu generieren, wählt Bob eine Zahl b, berechnet B ≡b G und sendet B an Alice.

Der gemeinsame geheime Schlüssel zwischen Alice und Bob lautet

c: = bA = baG = abG = aB .

Damit beide denselben Schlüssel c berechnen können, muss die Addition das Assoziativ- und Kommutativgesetz erfüllen, d. h. es ist unerlässlich, dass E eine Gruppe ist.

Das ECDHE-Protokoll, bei dem das zusätzliche E für „ephemeral“ (kurzlebig, flüchtig) steht, verwendet den gleichen Schlüsselaustausch wie das ECDH-Protokoll, verwirft aber die Schlüssel (die notwendigerweise durch permanente Schlüssel signiert werden, um die Identität zu bezeugen) direkt nach der Sitzung (Corbellini, 2015b).

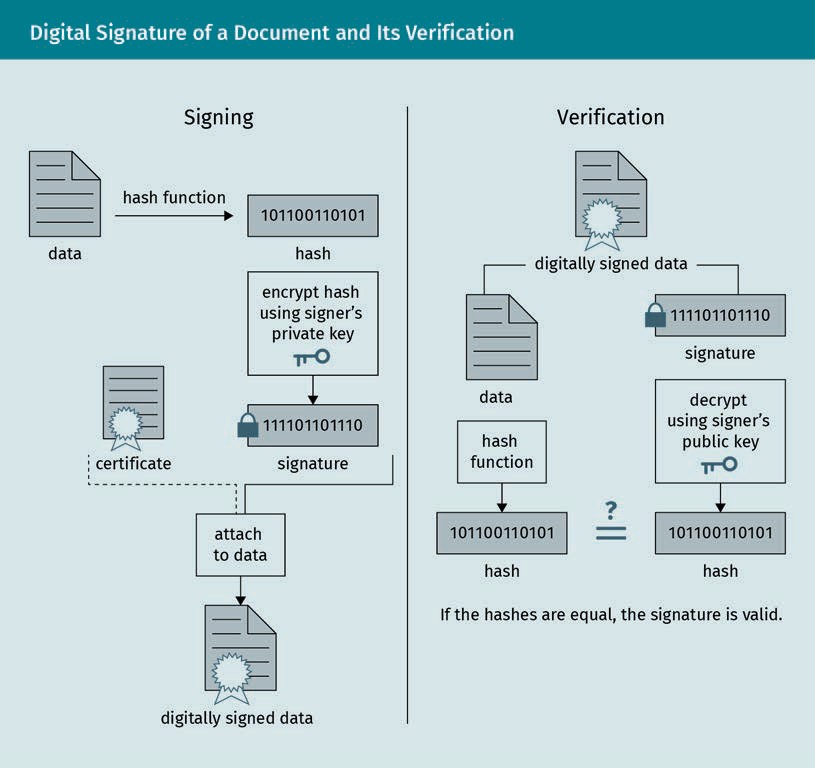
### Signaturen

Die Public-Key-Kryptografie verwendet zwei Schlüssel: einen öffentlichen Schlüssel und einen privaten Schlüssel. Normalerweise wird der öffentliche Schlüssel zum Verschlüsseln verwendet, während der private Schlüssel zum Entschlüsseln verwendet wird. So kann ein Text von der verschlüsselnden Person (Alice) nur an eine Person, die entschlüsselnde Person (Bob), übertragen werden. Die Rollen des öffentlichen und des privaten Schlüssels können vertauscht werden. Der private Schlüssel wird dann zum Verschlüsseln verwendet, während der öffentliche Schlüssel zum Entschlüsseln verwendet wird. Auf diese Weise kann der Verschlüsselnde allen Entschlüsselnden (denjenigen, die den öffentlichen Schlüssel besitzen) beweisen, dass er im Besitz des privaten Schlüssels ist: der digitalen Signatur.

Die Theorie hinter der Verschlüsselung mit dem öffentlichen Schlüssel (digitale Nachrichten) oder dem privaten Schlüssel (digitale Signatur) ist fast identisch. Lediglich die Rollen der Argumente der Falltürfunktion sind vertauscht. Beim RSA-Algorithmus zum Beispiel ist dieser Austausch von Variablen tatsächlich schon alles, was passiert. In der Praxis verschlüsseln der private Schlüssel und ein kryptografischer Hash in der Regel jedoch die Auffüllungen des Klartextes, um Schwachstellen zu vermeiden, die den Schlüssel offenbaren, wenn der Text zu kurz ist. Während bei der Verschlüsselung mit öffentlichen Schlüsseln die Funktion, mit welcher der Text (die Auffüllung) zunächst umgewandelt wird, leicht umkehrbar ist, ist die Funktion, mit welcher der Text zunächst umgewandelt wird (der Hash), bei der Verschlüsselung mit privaten Schlüsseln kaum umkehrbar.

Digitale Signatur Auf diese Verschlüsselung einer Nachricht mit dem privaten Schlüssel folgt die Entschlüsselung mit dem öffentlichen Schlüssel, um zu überprüfen, ob die ursprüngliche Nachricht

mit dem privaten Schlüssel verschlüsselt wurde.



###### RSA-Signaturalgorithmus

Beim RSA-Signaturalgorithmus besteht der einzige Unterschied darin, dass die Exponenten

E and d ihre Rollen tauschen, d. h. die signierte Nachricht ist M = md (statt mE).

Das Signieren und Dechiffrieren sind beide durch • d für den privaten Schlüssel d gegeben. Das Signieren eines verschlüsselten Dokuments (für den öffentlichen Schlüssel E , der d entspricht) ist also gleichbedeutend mit dessen Entschlüsselung! In der Praxis werden verschiedene Schlüsselpaare zum Verschlüsseln/Entschlüsseln, zum Signieren/Verifizieren verwendet und ein kryptografischer Hash h(d) des Dokuments d wird signiert, d. h. eine kleine Zahl, die das Dokument identiﬁziert.

Wir stellen fest, dass er anstelle der Originalnachricht einen kryptografischen Hash (z. B. mit dem Algorithmus MD5) der Originalnachricht signiert, und zwar mit zusätzlichen Informationen wie dem Namen des Signierenden und den Algorithmen, die zur Verschlüsselung und Berechnung des Hash verwendet wurden.

###### Digital Signature Algorithm (DSA)

Elgamal (1985) war der erste, der zeigte, wie man einen Verschlüsselungs- und Signaturalgorithmus auf der Grundlage des Difﬁe-Hellman-Protokolls entwickelt. Zwar wird der Verschlüsselungsalgorithmus nur selten verwendet (das Standard-Kryptografie-Befehlszeilentool GnuPG bietet ihn als erste Alternative zu RSA an), aber der Signaturalgorithmus steckt hinter dem des Digital Signature Algorithm (DSA), der im Digital Signature Standard (DSS) der US-Regierung verwendet wird, der 1994 vom NIST herausgegeben wurde. Elliptic Curve DSA (ECDSA) ist eine Variante des DSA, die Punkte auf endlichen (elliptischen) Kurven anstelle von ganzen Zahlen verwendet.

### Schlüsselaustausch und Public-Key-Infrastrukturen

Im Vergleich zur symmetrischen Kryptografie entfällt bei der asymmetrischen Verschlüsselung das Risiko der Kompromittierung des Schlüssels zur Entschlüsselung, das mit dem Austausch des Schlüssels mit dem Chiffrierenden verbunden ist. Diese sichere Kommunikation mit jedermann über einen unsicheren Kanal ist ein großer Vorteil gegenüber der symmetrischen Kryptografie. Wir wollen uns zunächst die klassischen Methoden zum Austausch eines symmetrischen Schlüssels ins Gedächtnis rufen, bevor wir das asymmetrische Gegenstück betrachten. Die asymmetrische Kryptografie machte es zwar möglich, einen geheimen Schlüssel offen auszutauschen, aber diese Annehmlichkeit verschleiert die Identität des Schlüsselbesitzers und macht sie anfällig für einen Man-in-the-Middle-Angriff, den die Public-Key-Infrastruktur durch die Verwendung von Zertifikaten (digitale Signaturen öffentlicher Schlüssel durch Dritte) unterbindet.

###### Symmetrische Kryptosysteme

Ein symmetrischer Schlüssel muss heimlich weitergegeben werden. Mögliche Methoden sind

Asymmetrische Kryptosysteme

* Ableitung von einem Basisschlüssel unter Verwendung einer Schlüsselableitungsfunktion (Key Derivation Function, KDF), einer kryptografischen Hashfunktion, die einen geheimen Schlüssel aus geheimen – und möglicherweise anderen öffentlichen – Informationen ableitet, z. B. einer eindeutigen Zahl,
* Erstellung eines Schlüssels aus Schlüsselteilen, die sich im Besitz verschiedener Personen befinden, z. B. als Analogie zum One-Time-Pad. Wenn *s* die geheime (binäre) Zahl ist, dann ist s = s1 ⊕s2 ⊕…⊕sn für die Teilgeheimnisse s1, s2, … Die Rekonstruktion von s ist nur möglich, wenn alle s1, s2, …

kombiniert werden.

* Übermittlung über einen anderen Kanal, z. B. persönlich, einen versiegelten Brief, per Telefon oder durch Quantenverschränkung, bei der Quantenpartikel so miteinander verbunden sind, dass sich der Zustand des einen Teilchens unabhängig von der räumlichen Trennung sofort auf das andere überträgt, obwohl sich Informationen nicht schneller als mit Lichtgeschwindigkeit bewegen können.

###### Man-In-The-Middle-Angriff (MITM)

Der große Vorteil der asymmetrischen Kommunikation besteht darin, dass die Geheimhaltung von der Authentifizierung getrennt werden kann, d. h. für einen erstellten Geheimtext wird nur der jeweilige öffentliche Schlüssel zur Entschlüsselung benötigt. Hinzu kommt, dass es keinerlei Informationen über den Dechiffrierenden gibt. Für die Kommunikation zwischen mehreren Parteien, z. B. Alice, Bob und Eve, müssen die öffentlichen Schlüssel authentifiziert werden, d. h. die Dechiffrierenden Bob und Eve müssen im Verzeichnis der Chiffrierenden Alice authentifiziert werden. Andernfalls könnte Eve Alice vorgaukeln, dass sie mit Bob kommuniziert, indem sie ihren öffentlichen Schlüssel durch den von Bob ersetzt.

Bei einem MITM platziert sich der Angreifer zwischen den Kommunikationspartnern und nimmt gegenüber jedem von ihnen die Identität des anderen an, um Nachrichten abzufangen. Ein Beispiel:

1. Bob sendet seinen öffentlichen Schlüssel an Alice. Eve fängt ihn ab und sendet Alice ihren eigenen öffentlichen Schlüssel, der Bob als Eigentümer angibt. Wenn Alice eine Nachricht an Bob sendet, dann verwendet sie, ohne es zu wissen, den öffentlichen Schlüssel von Eve!
2. Alice verschlüsselt eine Nachricht mit dem öffentlichen Schlüssel von Eve und sendet sie an Bob.
3. Eve fängt die Nachricht ab und entschlüsselt sie mit ihrem privaten Schlüssel. Sie kann die Nachricht lesen und sie verändern.
4. Eve verschlüsselt die Nachricht mit Bobs öffentlichem Schlüssel.
5. Bob entschlüsselt die Nachricht mit seinem privaten Schlüssel und schöpft keinen Verdacht.

Sowohl Alice als auch Bob sind davon überzeugt, dass sie den öffentlichen Schlüssel des jeweils anderen benutzt haben, aber in Wirklichkeit benutzen sie den von Eve!

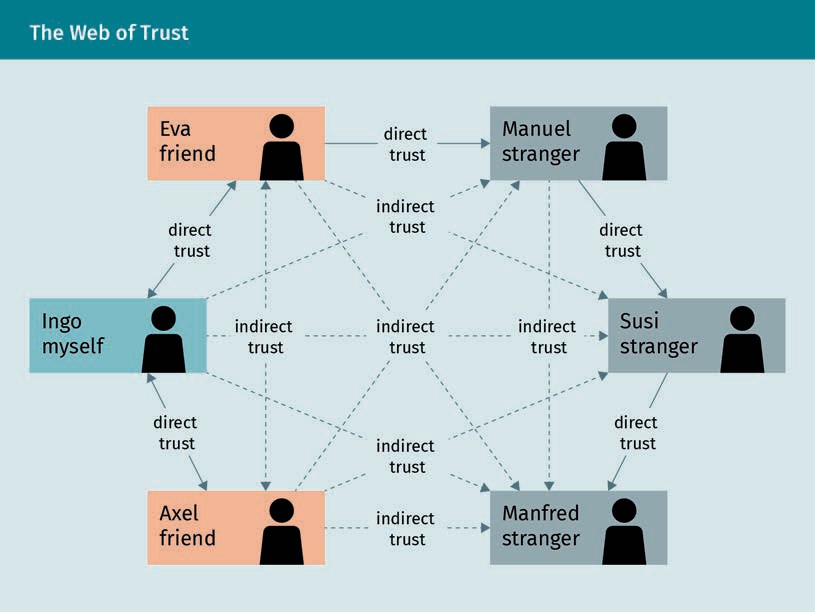
###### PKI

Eine Public-Key-Infrastruktur (PKI) eines Netzwerks stellt das Vertrauen zwischen den räumlich getrennten Benutzern her, indem sie diese zunächst authentifiziert und dann ihre öffentlichen Schlüssel durch Signieren (digitale Zertifikate) autorisiert und verteilt. In Institutionen und Unternehmen wird eine PKI oft als „Vertrauenshierarchie“ von Zertifizierungsstellen implementiert, während sie in weniger organisierten Gemeinschaften dezentralisiert sein kann und das Vertrauen von den Benutzern selbst aufgebaut wird. Eine PKI umfasst

* + (digitale) Zertifikate. Es handelt sich dabei um öffentliche Schlüssel, die signiert sind, um ihre Benutzer zu authentifizieren. Neben dem Namen und dem Schlüssel enthalten sie weitere persönliche Daten, wie z. B. eine E-Mail-Adresse und in der Regel ein Ablaufdatum.
  + Zertifikatssperrliste (Certiﬁcate Revocation List, CRL). Dies ist eine Liste von Zertifikaten, die vor Ablauf ihrer Gültigkeit widerrufen wurden, weil der Schlüssel kompromittiert wurde oder der Schlüsselinhaber aufgrund seines Verschwindens nicht mehr vertrauenswürdig ist.
  + Verzeichnisdienst. Dies ist eine durchsuchbare Datenbank mit den emittierten Zertifikaten. Beispiele hierfür sind eine Vertrauenshierarchie, ein LDAP-Server (Lightweight Directory Access Protocol, ein Standard, der von großen Unternehmen zur Verwaltung des Zugriffs von Benutzern auf Geräte, Drucker, Server und Anwendungsdaten verwendet wird) und das Web of Trust, ein Server, der eine Datenbank hostet, die über ein Webformular durchsucht werden kann.

###### Philosophie der Lösungen

Dritte, d. h. andere Identitäten mit privaten Schlüsseln, die mit ihren digitalen Signaturen bestätigen, bestätigen die Identität des Schlüsselinhabers. Es stellt sich jedoch erneut das Problem der Identität des öffentlichen Schlüssels: Wie können wir die Identitäten der Besitzer privater Schlüssel sicherstellen? Es gibt zwei Lösungen: hierarchische Autoritäten und das Web of Trust.



Während beim Web of Trust die durch Vertrauen aufgebauten Verbindungen einen Graphen bilden, bilden sie beim Ansatz der hierarchischen Autoritäten einen Baum.

Asymmetrische Kryptosysteme

Hierarchische Autoritäten

Beim Ansatz der hierarchischen Autoritäten werden die Besitzer privater Schlüssel durch hierarchische Ebenen unterschieden. Auf der höchsten Ebene befinden sich die Stammautoritäten, denen wir bedingungslos vertrauen. VeriSign, GeoTrust und Commode sind zum Beispiel wichtige US-Zertifizierungsunternehmen. Ein Blick in den Ordner /etc/ssl/certs in der Linux-Distribution openSUSE zeigt eine deutsche (TeleSec der Deutschen Telekom AG, dem ehemaligen staatlichen Telekommunikationsunternehmen), drei spanische (Firmaprofesional, ACCVRAIZ1-Agencia de Tecnología y Certiﬁcación Electrónica und ACC RAIZ FNMT-Fábrica Nacional de Moneda y Timbre) und viele amerikanische Autoritäten.

Web of Trust

Im Web of Trust sind die Besitzer privater Schlüssel nicht voneinander zu unterscheiden. Das Fehlen von Stammautoritäten, d. h. von Entitäten, denen man bedingungslos vertraut, wird durch das Vertrauen kompensiert, das ursprünglich dadurch aufgebaut wurde, dass man den öffentlichen Schlüssel persönlich erhalten hat (z. B. bei einer Schlüsselsignatur-Party) oder dass man den Schlüssel über einen anderen Kanal erhalten hat (Website, E-Mail usw.) und die Prüfsumme wieder über einen anderen Kanal (Telefon, SMS, Instant Messenger usw.) übermittelt hat. Vertrauen wird im aufgebauten Netzwerk transitiv weitergegeben.

###### Standardisierung von Philosophien im Internet

Im Internet wurde das System des Vertrauens durch hierarchische Autoritäten durch das Schema X.509 standardisiert, das hauptsächlich zur Verschlüsselung der Kommunikation zwischen einem Benutzer und einer (kommerziellen) Website (aber auch zwischen Benutzern in Unternehmensumgebungen, wie z. B. die S/MIME-E-Mail-Verschlüsselung) verwendet wird, sowie durch das OpenPGP-Schema (wie es durch das GnuPG-Programm implementiert wurde), dessen Hauptanwendung die Verschlüsselung von E-Mails ist. Dieses Schema lehnt jede Hierarchie radikal ab. Der Benutzer kann einen öffentlichen Schlüssel mit einer E-Mail-Adresse auf einem Public-Key-Server veröffentlichen, ohne auch nur zu bestätigen, dass er Zugriff auf dieses E-Mail-Konto hat.

###### DANE

Die Internet Engineering Task Force (IETF) hat (in RFC 63941: DANE Use Cases und RFC 66982: DANE Protocol) das DANE-Protokoll vorgeschlagen, das darauf abzielt, die Protokolle TLS, DTLS, SMTP und S/MIME mit DNSSEC kryptografisch abzusichern. Mit DNSSEC kann ein DNS-Resolver eine DNS-Auflösung authentifizieren, d. . feststellen, ob sie mit derjenigen auf dem maßgeblichen DNS-Server identisch ist, indem er seine Signatur (des verbindlichen DNS-Servers) überprüft. Anstatt sich wie bei diesen Protokollen vollständig auf Zertifizierungsstellen (CAs) zu verlassen, können die Inhaber von Definitionsbereichen die CAs einschränken, die die Zertifikate der Domain validieren, und können selbst Zertifikate ausgeben, ohne auf CAs zu verweisen.

Bei Verwendung von CAs gibt es keine Einschränkung bei den ausgestellten Zertifikaten. Wenn ein Angreifer die Kontrolle über eine einzige der vielen CAs erlangt, denen der Client vertraut, kann er gefälschte Zertifikate für jede Domain emittieren. DANE ermöglicht es Clients, die DNS-Server zu fragen, welche Zertifikate vertrauenswürdig sind, so dass der Domain-Inhaber den Geltungsbereich einer CA einschränken kann. Wenn dem Benutzer ein Domain-Name-Zertifikat übergeben wird (als Teil des anfänglichen TLS-Handshakes), kann der Client das Zertifikat mit einem TLSA-Ressourcendatensatz (TLSA RR) abgleichen, der im DNS für den vom maßgeblichen DNS-Server authentifizierten Dienstnamen veröffentlicht wurde.

Web of Trust Inhaber von privaten Schlüsseln

bestätigen die Identität der anderen, indem sie sich unter Gleichen gegenseitig Vertrauen schenken.

Schlüsselsignatur-Partys Das sind Treffen, bei denen die Teilnehmer ihre öffentlichen Schlüssel austauschen und gegenseitig signieren.

Der am weitesten verbreitete Standard für eine PKI ist die Hierarchie der X.509-Zertifizierungsstellen.

X.509 wurde erstmals 1998 veröffentlicht und wird von der Standardisierungsabteilung der Internationalen Fernmeldeunion (ITU-T) entwickelt. X.509 legt insbesondere ein Standardformat für elektronische Zertifikate und einen Algorithmus für die Validierung des Zertifizierungspfads fest. Das wichtigste Profil, PKIX-Zertifikat und CRL-Profil (PKIX), wurde von der IETF als Teil von RFC 3280, derzeit RFC 5280, entwickelt. Es wird von allen gängigen Webbrowsern wie Chrome und Firefox unterstützt, die über eine Liste vertrauenswürdiger X.509-Zertifizierungsstellen verfügen.

Im Einzelnen enthält die TLSA RR den Eintrag Certiﬁcate Usage, dessen Wert die Stelle einschränkt, die das Zertifikat für den Benutzer validieren darf. Je niedriger die Ebene, desto restriktiver ist sie.

* + Ebene 0: PKIX-TA (Zertifizierungsstellenzwang). Das Vertrauen des Clients beruht auf einer PKIX-Stelle.
  + Ebene 1: PKIX-EE (Servicezertifikatszwang). Das Vertrauen des Clients beruht auf einem PKIX-Zertifikat.
  + Ebene 2: DANE-TA (Trust-Anchor-Zusicherung). Das Vertrauen des Clients beruht auf einer Stelle, bei der es sich im Gegensatz zu PKIX-TA nicht um eine PKIX-Zertifizierungsstelle handeln muss.
  + Ebene 3: DANE-EE (Von Domain ausgestelltes Zertifikat). Das Vertrauen des Clients beruht auf einem Zertifikat, bei dem es sich, anders als bei PKIX-EE, nicht um ein PKIX-Zertifikat handeln muss.

Die DANE-Prüfung dient zur Bestätigung von Zertifikaten, die von öffentlichen Zertifizierungsstellen ausgestellt wurden. Mit den DANE-Werten (zwei und drei) hat der Domaininhaber die Möglichkeit, seine eigenen, selbst signierten Zertifikate für seine TLS-gesicherten Dienste zu erstellen, ohne eine dem Client bekannte Zertifizierungsstelle einbeziehen zu müssen. Durch die Wahl zwischen „Trust Anchor“ (TA) und „End Entity“ (EE) kann der Domaininhaber selbst entscheiden, ob er die DANE-Sicherheit an einem Zertifizierungsstellen- oder Server-Zertiﬁkat verankern möchte.

###### Hybride Kryptosysteme

Hybride Verschlüsselung Ein Zwei-Schlüssel-Algorithmus wird verwendet, um die Kommunikationsteilnehmer zu authentifizieren, indem die Nachrichten digital signiert werden, oder um einen Schlüssel für die Ein-Schlüssel-Kryptografie auszutauschen, und so eine efﬁziente Kommunikation zu

ermöglichen.

Zwischen zwei Parteien ist es üblich, einen Hash, z. B. einer Nachricht, als Verschlüsselungsschlüssel zu erstellen. MITM wird durch die Authentifizierung der entsprechenden öffentlichen Schlüssel bei einer Zertifizierungsstelle verhindert. Da kryptografische Algorithmen mit einem Schlüssel um einen beträchtlichen Faktor efﬁzienter sind als kryptografische Algorithmen mit zwei Schlüsseln, wird die Zwei-Schlüssel-Verschlüsselung hauptsächlich für die hybride Verschlüsselung verwendet; hierbei wird der Zwei-Schlüssel-Algorithmus verwendet, um die Kommunikationspartner zu authentifizieren, indem die Nachrichten digital signiert werden, oder um einen Schlüssel für die Ein-Schlüssel-Kryptografie auszutauschen, um eine efﬁziente sichere Kommunikation zu ermöglichen.

Im TLS-Protokoll (Transport Layer Security; früher SSL), das sichere Websites im World Wide Web verschlüsselt, verwendet ein kryptografisches Paket wie TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA (Identifizierungscode 0x00 0x0a) beispielsweise RSA zur Authentifizierung und zum Austausch der Schlüssel, 3DES im CBC-Modus zur Verschlüsselung der Verbindung und SHA als kryptografischen Hash.

Asymmetrische Kryptosysteme

Zusammenfassung

Die symmetrische Kryptografie leidet unter dem Problem der Schlüsselverteilung. Die asymmetrische Kryptografie löst dieses Problem scheinbar auf Anhieb, indem sie die Verwendung unterschiedlicher Schlüssel zum Ver- und Entschlüsseln ermöglicht. Allerdings muss dazu die Identität des Schlüsselinhabers bestätigt werden. Dies kann persönlich oder durch Dritte geschehen, z. B. durch Identitäten mit privaten Schlüsseln, die mit ihren digitalen Signaturen die Eigentümerschaft bestätigen. Es stellt sich jedoch erneut das Problem der Identität des privaten Schlüssels: Wie können wir die Identitäten dieser Besitzer privater Schlüssel sicherstellen? Dafür gibt es zwei Lösungen. Bei dem Ansatz über hierarchische Autoritäten werden die Besitzer privater Schlüssel durch hierarchische Ebenen unterschieden. Auf der höchsten Ebene befinden sich die Stammautoritäten, denen wir bedingungslos vertrauen. Im Web of Trust wird das Vertrauen von einem zum anderen übertragen.

Asymmetrische Kryptografie beruht auf einer Falltürfunktion, die einfach sein muss, deren Umkehrung aber ohne Kenntnis einer Abkürzung – des Schlüssels – praktisch nicht möglich sein darf.

Die Schwierigkeit, die Umkehrung zu berechnen, entspricht der Schwierigkeit der Entschlüsselung, d. h. der Umkehrung der Verschlüsselung. Die Berechnung der Umkehrfunktion wird durch modulare (oder zirkuläre) Arithmetik erschwert.

Die derzeit am weitesten verbreitete Kryptografie verwendet elliptische Kurven. Das Difﬁe-Hellman-Protokoll (über IFp) hat ein Analogon über elliptischen Kurven. Der Vorteil der Verwendung elliptischer Kurven sind kürzere Schlüssel. Kleine Schlüssel für ECC erreichen das gleiche Maß an Sicherheit wie große Schlüssel für RSA oder DH. So entspricht beispielsweise die Sicherheit eines 224-Bit-Schlüssels bei ECC der eines 2048-Bit-Schlüssels bei RSA oder DH. Dieser Faktor bei der Reduzierung der Schlüsselgröße entspricht einem ähnlichen Faktor bei der Reduzierung des Rechenaufwands.