Kryptographie

Einleitung

Über das Thema Kryptographie gibt es jede Menge Literatur und jede Menge Texte und Videos im Internet. Wozu also dieses kleine Skript? Meine Antwort dazu ist, dass ich es in erster Linie für mich selbst schreibe. Das Thema hat mich schon immer interessiert. Aber ist zu schwierig, um sich nur am Rande damit zu beschäftigen. Mir hilft es, wenn ich Zeit habe, es systematisch durchzuarbeiten, und mir auch die dazugehörigen mathematischen Hintergründe erarbeiten kann. Erst in den letzten Jahren bin ich dazu gekommen. Und da ich mich ganz und gar nicht zu den mathematischen Genies zähle, muss ich alles aufschreiben, um eine Struktur zu erarbeiten und alles halbwegs behalten zu können. Dabei haben meine Aufzeichnungen die Form eines Lehrtextes, der auch dazu dienen könnte, anderen Interessierten den Inhalt nahe zu bringen. Dies ist mir in der Vergangenheit immer recht gut gelungen und ich hoffe vielleicht, dass auch dieser Text dem einen oder der anderen beim Verständnis der Materie besser helfen kann als viele andere verfügbare Werke.

Das Thema ist deswegen besonders faszinierend, weil es um ein uraltes Problem geht, für das kluge Köpfe schon immer geniale Lösungen gefunden haben. Dabei machen wir einen Streifzug durch die Mathematik der letzten dreitausend Jahre und lernen ihre bedeutendsten Vertreter und deren Ideen kennen. Die Erkenntnisse von Euklid, Fermat, Euler und Gauß spielen auch in den modernsten kryptographischen Methoden eine große Rolle.

Definition

Als Kryptographie bezeichnet man die Wissenschaft von der Verschlüsselung und der Entschlüsselung von Informationen. Durch die Verschlüsselung soll die Information vor Unbefugten geheim gehalten werden.

Nachrichten wurden schon im Altertum hauptsächlich zu militärischen Zwecken verschlüsselt. In der heutigen Zeit werden die meisten Daten, die über das Internet ausgetauscht werden, verschlüsselt.

Bekannte Beispiele für Verschlüsselung von Nachrichten sind die von Julius Cäsar verwendete Methode, oder die mit der Enigma verschlüsselten Nachrichten, mit denen deutsche U-Boote im 2. Weltkrieg kommunizierten. In der heutigen Zeit ist es ein großes Thema, wie unsere Privatsphäre beim Austausch von Nachrichten auf Internetplattformen gewahrt wird oder wie das Bezahlen mit Kryptowährungen sicher gemacht werden kann.

Man unterscheidet zwischen der **symmetrischen und der asymmetrischen** Verschlüsselung. Bei der symmetrischen Verschlüsselung nutzen Absender und Empfänger den gleichen Schlüssel. Der Schlüssel muss beiden Personen bekannt sein und mündlich oder unverschlüsselt übermittelt werden.
Bei der asymmetrischen Verschlüsselung gibt es einen öffentlichen und einen privaten Schlüssel. Mit dem öffentlichen Schlüssel kann jeder Nutzer eine verschlüsselte Nachricht

zuschicken. Die Entschlüsselung der Nachricht ist allerdings nur mit dem privaten Schlüssel möglich.

A Symmetrische Verschlüsselungsverfahren

Der Schwerpunkt dieses Skripts liegt auf dem grundlegenden Verständnis der asymmetrischen Verschlüsselungsverfahren, die etwa seit den 60er Jahren des 20. Jahrhunderts aufgetaucht sind. Auf die symmetrischen Verfahren soll hier nur kurz eingegangen werden. Das Prinzip haben viele von uns bereits in der Kindheit kennengelernt, wenn es darum ging eine Geheimschrift oder eine Geheimsprache zu entwickeln. Meist haben wir dabei Buchstaben oder Worte durch andere Buchstaben oder Worte ersetzt. Damit haben wir bereits das Grundprinzip der symmetrischen Verschlüsselung angewendet.

Der Text des Senders (**Klartext)** wurde auf eine bestimmte Art verschlüsselt (**Schlüssel)**. Der daraus entstandene **Geheimtext** wurde dann dem Empfänger übermittelt, der denselben Schlüssel kennen musste, um den Geheimtext durch Umkehrung des Schlüssels wieder in den Klartext übersetzen zu können

1 Monoalphabetische Verschlüsselung

Jeder Buchstabe wird stets durch das gleiche Symbol ersetzt. Es gibt also ein Klartextalphabet (a – z) sowie ein Geheimtextalphabet. Die Anzahl der Möglichkeiten der Verschlüsselung ergibt sich aus 26!

Transposition: Die Buchstaben des Klartextes werden nach einer Regel an andere Positionen gesetzt.

Verschiebung: Die Buchstaben des Klartextes werden um eine best. Anzahl von Positionen verschoben. Berechnung per Modulo 26. Bsp**. Caesarverschlüsselung**. (Drehscheibe) Der Schlüssel ist eine Zahl zwischen 0 und 25 um die das Alphabet nach rechts verschoben wird. Rechts herausfallende Buchstaben werden auf der linken Seite wieder eingesetzt.
Das Verfahren ist nur dann absolut sicher, wenn lediglich ein einziges Zeichen verschlüsselt wird. Bei längeren Texten kann es leicht, auch ohne Computer, durch die statistische Analyse geknackt werden

Verbesserung: Der Verschiebschlüssel kann auch ein Wort oder eine längere Zahl sein, so dass die Verschiebung bei jedem Buchstaben anders erfolgt. Bsp. „abel“. D.h. beim ersten Buchstaben erfolgt eine Verschiebung um 0 Zeichen, beim zweiten um 1, beim dritten um 4 beim vierten um 11, dann wieder von vorn.

Statistische Analyse

Statistische Erfassung der Buchstabenhäufigkeit (siehe S. 12) Bei langen Texten ist i. d. R. ‚e‘ der häufigste Buchstabe gefolgt von ‘n‘ usw.

Bigramme: es gibt bestimmte Buchstabenfolgen, die in der dt. Sprache bes. häufig vorkommen. Z. B. ‚en‘ gefolgt von ‚er‘. (S. 21). Man kann den Text nach häufig vorkommenden Folgen von zwei Buchstaben durchsuchen.

Tauschchiffren

Jeder Buchstabe bekommt eine Nummer. Zunächst wird jeder Klartextbuchstabe mit einer Zahl s multipliziert. Zu dem sich daraus ergebenden Buchstaben wird dann eine weitere Zahl t addiert. Das

Ergebnis ist die Tauschchiffre die man mit [s,t] bezeichnet. S muss eine der Zahlen 1, 3 , 5, 7, 9 ,11, 17,19, 21, 23, 25 sein, da dies teilerfremd zu 26 sind. Wenn x die Nummer des Klartextbuchstabe ist, dann ist die Nummer des Geheimtextbuchstabens xs + t mod 26.

Bsp. x = c (3) und s = 17 und t = 5, dann gilt g = 3 .17 + 5 = 56 mod 26 = 4 also d!

Entzifferung: $x = (s^{-1} (g – t) \bmod 26$, wobei s^{-1}- die Inverse zu s ist, die sich aus einer Tabelle ergibt (siehe erw. euklidischer Algorithmus)

Schlüsselwortchiffre

Schlüssel besteht aus zwei Komponenten: einem Schlüsselwort und einem Schlüsselbuchstaben. Beim Schlüsselwort wird jeder mehrfach vorkommende Buchstabe in der Wiederholung gestrichen.

Bsp. geheimschrift → gehimscrft , der Schlüsselbuchstabe sei das ‚e'

Dann schreibt man den Geheimtext unter das Alphabet und beginnt beim Schlüsselbuchstaben. Die restlichen Buchstaben füllt man mit den restlichen Buchstaben des Alphabetes auf, indem man hinter dem letzten Schlüsselwortbuchstaben beginnt.

a b c d e f g h i j k l m n o p q r s t u v w x y z

w x y z g e h i m s c r f t a b d j k l n o p q u v

Ein Buchstabe des Klartextes wird durch den darunter stehenden Buchstaben verschlüsselt.

DES (Data Encryption Standard)

Der zu verschlüsselnde Text wird ein Blöcke zu jeweils 64 Bit umgewandelt (z. B. mit dem ASCII-Code). Die Schlüssel sind binäre Folgen aus 56 Bits. Es gibt dann $2^{56} = 7 \cdot 10^{16}$ Schlüssel. Das ist zwar schon eine ganze Menge, kann aber mit modernen Rechnern mühelos durch systematisches Durchprobieren aller Möglichkeiten innerhalb von Minuten geknackt werden.

2 Polyalphabetische Verschlüsselung

Bei diesen Verfahren sollen die unterschiedlichen Häufigkeiten der Buchstaben möglichst angeglichen werden, indem ein Klartextbuchstabe nicht immer mit demselben Geheimtext-buchstaben verschlüsselt wird.

Viginère-Verschlüsselung (häufig verwendete polyalphabetische Methode)

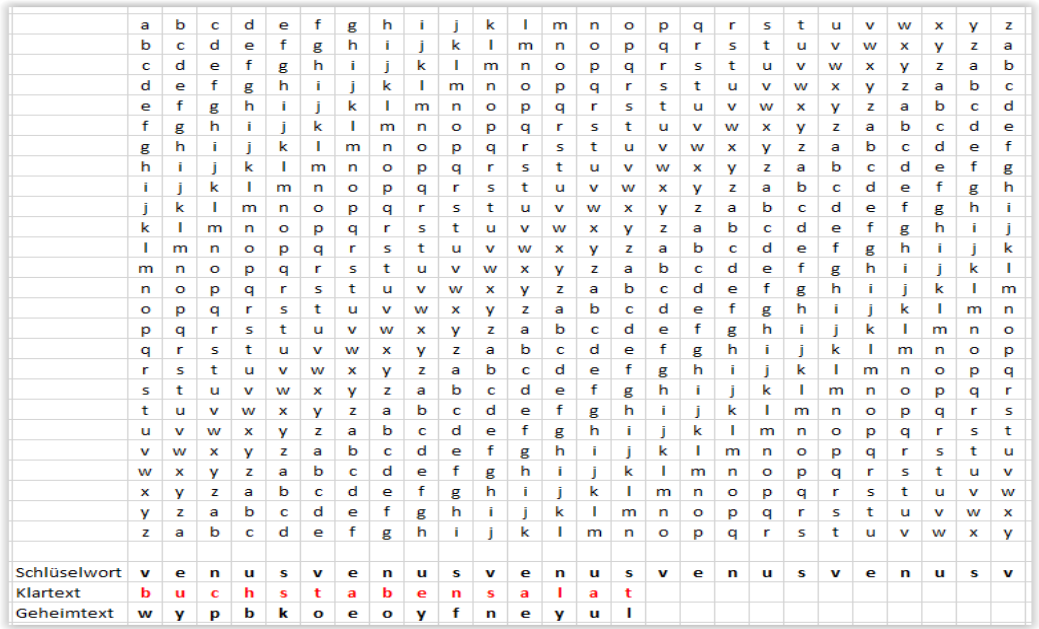
Erstellen eines Vigninère-Quadrates 26 x 26 in der Form das alle Buchstaben des Alphabetes jeweils um einen Buchstaben versetzt untereinandergeschrieben werden.

Dazu gibt es ein Schlüsselwort. Das Schlüsselwort wird fortlaufend über den Klartext geschrieben.

Bei der Verschlüsselung wird jeder Buchstabe mit Hilfe des über ihm stehenden Schlüsselwortbuchstabens verschlüsselt. Dazu wird in der Tabelle in der ersten Zeile nach dem Klartextbuchstaben gesucht und in der entsprechenden Zeile des Schlüsselwortbuchstabens der dazu gehörige Buchstabe des Geheimtextes ermittelt. Dadurch wird die Häufigkeit der Buchstaben viel gleichmäßiger verteilt.

Bsp.: Schlüsselwort: venus	Klartext: buchstabensalat	Geheimtext: wypbkoeoyfneyui

Wenn man ein Schlüsselwort aus einer zufällig erzeugten Buchstabenfolge wählt und einen sicheren Übertragungsweg für das Schlüsselwort findet, ist dieser Code nicht zu knacken. Selbst der schnellste Computer müsste für das Schlüsselwort alle möglichen Buchstabenkombinationen durchspielen und käme damit auf eine sehr, sehr große Anzahl an möglichen Klartexten, die einen Sinn ergäben, wüsste aber dann noch immer nicht, welches der richtige ist.



	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
Schlüselwort	v	e	n	u	s	v	e	n	u	s	v	e	n	u	s	v	e	n	u	s	v	e	n	u	s	v
Klartext	b	u	c	h	s	t	a	b	e	n	s	a	l	a	t											
Geheimtext	w	y	p	b	k	o	e	o	y	f	n	e	y	u	l											

AES (advanced encrypting signature)

Im Jahr 2002 wurde dieser Standard entwickelt, der auch heute noch als das sicherste symmetrische Verschlüsselungsverfahren gilt und in vielen Bereichen der Telefonie und WLAN usw. angewendet wird.
Es handelt sich dabei um eine Blockverschlüsselung, bei der der zu verschlüsselnde Text in Blöcke mit 128, 192 oder 256 Bit mit je 8 Bits eingeteilt wird. Man erhält bei 128 Bit dann 4x4 Felder à 8 Bit, die in einer Tabellenform geschrieben werden.

Die Verschlüsselung erfolgt in vier Stufen, die mehrfach durchlaufen werden.
1. Stufe: Jedes Byte wird durch eine anders Byte einer Substitutionsmatrix ersetzt.
2.Stufe: Jede Zeile wird um eine Anzahl Bytes nach links verschoben. Herausfallende Bytes werden auf der rechten Seite wieder eingesetzt.
3. Stufe: Jede Spalte wird mit einer Matrix multipliziert.
4. Stufe: Jedes Byte wird mit der XOR-Operation einer Schlüsselmatrix verknüpft.
Die Informationen für die Anzahl der Durchläufe und die Substitutionsmatrix usw. befinden sich im Schlüssel.

Nachteile der symmetrischen Verschlüsselung

Die Gefahren der symmetrischen Verschlüsselungsmethoden liegen auf der Hand. Sender und Empfänger müssen sich einmal treffen, um den Schlüssel auszutauschen. Dieses Treffen kann abgehört werden. Es gibt mindestens zwei Parteien, die den Schlüssel kennen. Der Schlüssel kann in der heutigen

Zeit schnell geknackt werden, indem per Computer alle Möglichkeiten simuliert werden. Das bedeutet, wenn eine verschlüsselte Nachricht abgehört wird und der Abhörenden über leistungsfähige Rechner verfügt, ist die Nachricht meist in Kürze entschlüsselt. Andernfalls könnte die Nachricht verändert werden, ohne, dass der Empfänger es merkt.

Wem diese kurzen Ausführungen nicht genügt haben, der sei auf eines der zahlreichen YouTube-Videos verwiesen, in denen jedes einzelne Verfahren ausführlich und mit Beispielen erklärt wird. Mein Schwerpunkt liegt eindeutig auf dem folgenden Kapitel, in dem ein faszinierendes Gebiet der Mathematik betreten wird.

B Asymmetrische Verschlüsselung

Seit etwa der Zeit nach dem 2. Weltkrieg verwendet man asymmetrische Verschlüsselungsmethoden. Diese haben den Vorteil, dass der Verschlüsselungsalgorithmus nicht mehr geheim gehalten werden muss, sondern lediglich der private Schlüssel des Senders oder Empfängers.

Zum Verständnis der asymmetrischen Verschlüsselung sind einige mathematische Grundkenntnisse nützlich, die im 1. Kapitel dargestellt werden. Anhand des RSA-Verfahrens wird die Funktionsweise der asymmetrischen Verschlüsselung ausführlich beschrieben. Es folgt dann die kurze Beschreibung des Diffie-Hellman- Verfahrens und des Verfahrens der elliptischen Kurven.

1. Mathematische Grundlagen

Eine wichtige Rolle in der Kryptographie spielen **Primzahlen**. Wir wissen, dass Primzahlen solche Zahlen sind, die sich nur durch 1 und durch sich selbst teilen lassen. Eine wichtige Aussage stammt von **Euklid**, die auch als der **Fundamentalsatz der Arithmetik** bezeichnet wird:
Jede natürliche Zahl kann eindeutig als das Produkt von Primzahlen dargestellt werden.
Beispiel: 24 = 2 x 2 x2 x 3

1.1 Modulo-Rechnen

In der Kryptographie wird sehr oft das Modulo-Rechnen verwendet, welches den Divisionsrest einer Zahl berechnet. Man schreibt: 17 = 2 mod 5 . D. h. 17 geteilt durch 5 hat den Rest 2. Für das Modulo-Rechnen gibt es eine Reihe von Rechenregeln.

Hier ein Link für einen Modulo-Rechner https://de.planetcalc.com/8326/

1.2 Teilerfremdheit

Zwei Zahlen sind teilerfremd, wenn es keine Zahl außer 1 gibt, die beide Zahlen teilt. Beispiel: 15 und 8 sind teilerfremd, 8 und 18 sind nicht teilerfremd, da beide durch 2 teilbar. Teilerfremde Zahlen zu 15 sind: 2, 7, 9, 11,13.

1.3 Kongruenz: Wenn zwei ganze Zahlen a und b bei der Division durch m denselben Rest ergeben, sind sie kongruent zueinander.
Den Divisionsrest bezeichnet man als modulo (mod), den Teiler als Modul.
Das Symbol für Kongruenz ist das dreifache Gleichheitszeichen $\equiv$

$$a \equiv b \ mod\, m \quad \text{oder} \quad 11 \equiv 25\, mod\, 7$$

Bei kongruenten Gleichungen kann man auf beiden Seiten dasselbe addieren, subtrahieren oder multiplizieren. Beide Seiten durch dieselbe Zahl dividieren funktioniert allerdings nur, wenn Divisor und Modul teilerfremd zueinander sind.

Beispiele:
$11 \equiv 25 \bmod 7 \rightarrow 4$
$11 + 4 \equiv 25 + 4 \bmod 7 \rightarrow 1$
$11 - 5 \equiv 25 - 5 \bmod 7 \rightarrow 6$
$11 \cdot 2 \equiv 25 \cdot 2 \bmod 7 \rightarrow 1$
aber: $10 : 2 \equiv 24 : 2 \bmod 6$ falsch, da 2 und 6 nicht teilerfremd sind
aber $15 : 3 \equiv 54 : 3 \bmod 7$ ist richtig → 5, da 3 und 7 teilerfremd sind.

1.4 Zyklische Gruppen (etwas aus der Gruppentheorie)

Eine Primzahl p hat genau p-1 Gruppenelemente. Diese sind in diesem Fall alle Zahlen, die teilerfremd zu p sind.
Gibt es einen Generator, der alle Gruppenelemente erzeugt?
Bsp. p = 11, die Gruppenelemente sind 1,2,3,4,5,6,7,8,9,10,
die Operation $2^n \bmod 11$ erzeugt alle Elemente, aber in nicht vorhersehbarer Reihenfolge.
Diese Gruppe ist deshalb zyklisch, weil man nach p-1 (hier 10) Schritten wieder auf dem 1. Element landet. D.h. $2^3 \bmod 11 \equiv 2^{13} \bmod 11$. Der Generator ist 2.

$2^1 \bmod 11 = 2$	$2^2 \bmod 11 = 4$	$2^3 \bmod 11 = 8$
$2^4 \bmod 11 = 5$	$2^5 \bmod 11 = 10$	$2^6 \bmod 11 = 9$
$2^7 \bmod 11 = 7$	$2^8 \bmod 11 = 3$	$2^9 \bmod 11 = 6$
$2^{10} \bmod 11 = 1$	$2^{11} \bmod 11 = 2$	

Neutrale Elemente verändern den Wert einer Gruppenoperation nicht. Z. B. die Multiplikation mit 1 oder die Addition von 0.

Inverse Elemente ergeben den Wert 1 bei einer Operation. Z.B. Multiplikation mit dem Kehrwert.

1.5 Die (schwache) Vermutung von Goldbach (noch unbewiesen)

Jede ungerade Zahl kann als die Summe dreier Primzahlen dargestellt werden.

1.6 Die starke Vermutung von Goldbach, auch Eulers Vermutung

Jede gerade Zahl kann als die Summe zweier Primzahlen dargestellt werden.
Z. B. 8 = 5 + 3 oder 22 = 17 + 5

Man kann mit einem schönen Beweis zeigen, dass die schwache Goldbach'sche Vermutung zutreffen muss, wenn die starke Vermutung gilt. D.h. die schwache Vermutung kann aus der starken Vermutung abgeleitet werden.

Sei z eine gerade Zahl, dann gilt $z = p1 + p2$
ziehen wir 1 ab, dann gilt $z - 1 = p1 + p2 - 1$
Damit muss z – 1 ungerade sein sowie p2 – 1 gerade sein. Jede gerade Zahl kann aber als Summe zweier Primzahlen dargestellt werden. $P2 - 1 = p3 + p4$
und dann gilt $z - 1 = p1 + p3 + p4$.
Damit ergibt sich die schwache Goldbach'sche Vermutung aus der Eulerschen Vermutung.

1.7 Diskrete Exponentialfunktion und diskreter Logarithmus (diskret = ganzzahlige Werte)

Die Exponentialfunktion $b^x \bmod m$ liefert den Divisionsrest bei der Division von bx durch m. Die Berechnung des Exponenten x bei gegebener Basis b und dem Modul m wird als diskreter

Logarithmus bezeichnet. Diese Berechnung ist für große Exponenten (100 Stellen und mehr) nicht möglich bzw. würde selbst für die schnellsten Computer länger dauern als das Universum existiert.

Bsp. $3^9 \bmod 5 = 3$ welche Lösung hat x bei $3^x \bmod 11 = 9$?

Man nennt Funktionen, die nicht umkehrbar sind, **Einwegfunktionen**. Zwei einfache Beispiele für solche Einwegfunktionen sind die Quersumme und das Produkt zweier Zahlen.
Die Zahl 2521 hat als Quersumme 10. Von der 10 kann man aber nicht wieder auf die ursprüngliche Zahl schließen. Oder: Das Produkt zweier Primzahlen beträgt 11349. Man kann aber diese beiden Primzahlen (117 und 97) nur schwer wieder herausfinden.

1.8 Der kleine Satz von Fermat

$a^p \equiv a \bmod p$ wobei a eine ganze Zahl und p eine Primzahl ist.
Wenn man eine Zahl mit einer Primzahl potenziert, ergibt sich diese Zahl wieder als Divisionsrest durch die Primzahl.

Oder auch $a^{p-1} \equiv 1 \bmod p$

Mit dieser Methode, dachte man, sei es möglich, verlässlich Primzahlen zu testen. Bis allerdings der amerikanische Mathematiker Robert Carmichael herausfand, dass es Zahlen gibt, die zwar dieses Kriterium erfüllen, diese aber keine Primzahlen sind. Deswegen ist bei Erzeugung von Schlüsseln mittels des RSA-Verfahrens darauf zu achten, dass man nicht durch eine Carmichael-Zahl getäuscht wird.

1.9 Die Eulersche Phi-Funktion Φ

Gegeben ist eine Zahl n, die das Produkt zweier Primzahlen ist. $n = p \cdot q$.
Man erhält die Φ-Funktion, indem man die Anzahl, der zu n teilerfremden Zahlen bestimmt.
Da eine Primzahl p p-1 teilerfremde Zahlen hat, ist $\Phi(n) = (p-1)(q-1)$.
Beispiel: p = 13 und q = 23 → $n = 13 \cdot 23 = 299$ und $\Phi(n) = 12 \cdot 22 = 264$.

1.10 Der Satz von Euler und der kleine Satz von Fermat

Wenn a und m zwei teilerfremde Zahlen sind (z. B 15 und 8), dann gilt $a^{\Phi(m)} \equiv 1 \bmod m$

$\Phi(8) = 4$ {1,3,5,7}

$15^4 \equiv 1 \bmod 4$ 59625 : 4 = 12656 Rest 1.

Aus dieser Aussage folgt der **kleine Satz von Fermat** für eine Primzahl p und eine zu p teilerfremde Zahl a $a^{p-1} \equiv 1 \bmod p$ $4^6 = 1 \bmod 7$

daraus folgt auch: $a^p = a \bmod p$ damit erhält man wieder die Ausgangszahl

Daraus kann auch folgender Zusammenhang abgeleitet werden:

$e \cdot d \equiv 1 \bmod \Phi(n)$

wenn e und n die public keys sind, kann der private key d daraus berechnet werden.
Bsp. e = 7und n = 187, dann hat d den Wert 23.

Dieser Zusammenhang spielt eine wichtige Rolle bei der RSA-Verschlüsselung.
Der private key d kann mit dem euklidischen Algorithmus errechnet werden.

1.11 Satz von der modularen Inversen

Wenn a und n teilerfremde Zahlen sind, so gibt es eine ganze Zahl b mit der Eigenschaft

$a \cdot b \bmod n = 1$ oder auch $a \cdot b = 1 \bmod n$

Bsp.: $5 \cdot b = 1 \bmod 7$ dann ist b = 3, da $(5 \cdot 3) : 7 = 2$ Rest 1

Dann ist b die modulare Inverse oder auch a^{-1}.
Dies gilt auch für alle anderen Elemente Restklasse 3 mod 7 . Z. B. b = 17, da $5 \cdot 17 \bmod 7 = 1$, oder b = 10, 24, 31 …).
Zur **Restklasse** gehören alle Zahlen, die denselben Rest ergeben mod n.

1.12 Diophantische Gleichungen

Diophantische Gleichungen sind Gleichungen mit mehr Variablen als Gleichungen. Oftmals handelt es sich um eine Gleichung mit 2 Variablen.
$a \cdot x + b \cdot y = c$ wobei a und b zu bestimmen sind.
$a \cdot x + b \cdot y = 1$, wenn a und b Primzahlen sind.

Es gibt verschiedene Verfahren, um solche Gleichungen zu lösen. In der Kryptographie verwendet man den euklidischen Algorithmus bzw. dem erweiterten euklidischem Algorithmus

1.13 Der euklidische Algorithmus

Der euklidische Algorithmus dient dazu, den ggT zweier Zahlen zu berechnen.
Beispiel: Wir suchen den ggT der beiden Zahlen 728 und 266. Das Verfahren geht so:

728 = 2 x 266 Rest 196
266 = 1 x 196 Rest 70
196 = 2 x 70 Rest 56
70 = 1 x 56 Rest 14
56 = 4 x **14** Rest 0 Damit ist 14 der ggT.

1.14 Berechnung der modularen Inversen durch den erweiterten euklidischen Algorithmus

Anhand der beiden Zahlen von oben und deren ggT von 14, kann man folgende Gleichung aufstellen:

14 = 728 x + 266 y Gesucht sind die beiden Zahlen x und y. Die Lösung erhält man durch Fortsetzen des euklidischen Algorithmus in umgekehrter Reihenfolge. Dazu ordne ich die Zeilen des obigen Rechenbeispiels etwas anders in einer Tabelle an:

Zeile	a	b	q	x	y
1	728	266	2	**-4**	**11**
2	266	196	1	3	-4
3	196	70	2	-1	3
4	70	56	1	1	-1
5	56	14	4	**0**	**1**
6	**14**	0			

Zunächst notierte man in Zeile 5 für x eine 0 und für y eine 1. Die 1 in der letzten Spalte wird in die Zeile davor in die vorletzte Spalte (Spalte x) übernommen. Der Wert in der y-Spalte der 4. Zeile wird berechnet aus dem Wert aus der x-Spalte der Zeile davor (also 0) minus dem Produkt des Wertes aus der y-Spalte der 5. Zeile und dem Wert aus der Q-Spalte der 4. Zeile. (Also 0-1x1 = -1). Dieser Wert wird in die x-Spalte der 3. Zeile übernommen und der Wert in der y-Spalte errechnet aus: 1 – (-1)x 2 = 3. So wird weiter verfahren, bis die oberste Zeile erreicht ist, und man erhält die Lösungen x = - 4 und y = 11.

Mit diesem Verfahren berechnet man den privaten Schlüssel bei der RSA-Verschlüsselung.
Es folgt noch ein 2. Beispiel, um insbesondere das rückwärtige Einsetzen nochmals zu üben.

Gegeben ist die Gleichung 1 = 30x + 47y oder 30b ≡ 1 mod 47
Wir suchen wieder x bzw. b.

30 = 0 x 47 + 30	30 = 30 – 0 x 47	(I)
47 = 1 x 30 + 17	17 = 47 – 1 x 30	(II)
30 = 1 x 17 + 13	13 = 30 – 1 x 17	(III)
17 = 1 x 13 + 4	4 = 17 – 1 x 13	(IV)
13 = 3 x 4 + 1	1 = 13 – 3 x 4	(V)
4 = 4 x 1 Rest 0	wenn kein Rest bleibt, ist die Rechnung beendet	

rückwärts

(V)	1 = 13 – 3 x 4	
(IV)	1 = 13 – 3(17 – 1 x 3)	= -3 x 17 + 4 x 13
(III)	1 = -3 x 17 + 4(30 – 1 x 17)	= 4 x 30 – 7 x 17
(II)	1 = 4 x 30 – 7(47 – 1 x 30)	= -7 x 47 + 11 x 30
(I)	1 = - 7 x 47 + 11(30 – 0 x 47)	= 11 x 30 – 7 x 47

die gesuchte Zahl ist 11
Probe: 30 x 11 + (-7 x 47) = 1 oder 30 x 11 ≡ 1 mod 47

1.15 Berechnen von großen Potenzen und deren Modulo

Um den Divisionsrest von großen Potenzen zu berechnen, helfen die Aussagen über Kongruenz und der Satz von Euler. Man zerlegt die großen Exponenten in kleinere und berechnet schrittweise den Divisionsrest. Beispiele:

$7^{98} \bmod 5 \equiv 2^{98} \bmod 5$	da 7 mod 5 = 2
$2^{98} \bmod 5 \equiv 2^5 \cdot 2^{93} \bmod 5 \equiv 2 \cdot 2^{93} \bmod 5$	da $2^5 \bmod 5 = 2$
$= 2 \cdot (2^{10})^9 \cdot 2^3 \equiv 2 \cdot 4^9 \cdot 3$	da $2^{10} \bmod 5 = 4$ und $2^3 \bmod 5 = 3$
$= 6 \cdot (4^3)^3 \bmod 5 \equiv 6 \cdot 4^3 \equiv 1 \cdot 64 \bmod 5 =$ **4**	

Beispiel 2:

$2^{47} \bmod 77 = 2^5 \cdot (2^7)^6 \bmod 77 = 32 \cdot 128^6 \bmod 77 \equiv 32 \cdot 51^6 \bmod 77$	da 51 mod 77 == 128 mod 77
$= 32 \cdot (51^2)^3 \bmod 77 = 32 \cdot 2601^3 \bmod 77 \equiv 32 \cdot 60^3 \equiv 32 \cdot (-17)^3 \bmod 77$	da 60 = - 17 mod 77
$32 \cdot (-17)^3 \bmod 77 = 32 \cdot (-17)(-17)*2 \bmod 77 \equiv 32 \cdot (-17) \cdot 58$	da $(-17)^2 \bmod 77 = 58$
$-544 \cdot 58 \bmod 77 \equiv -5 \cdot 58 \bmod 77 = -290 \bmod 77 = 18$	

natürlich sind auch andere Umformungen und Lösungswege denkbar.

kleine Hilfe, um den Divisionsrest mit dem TR zu ermitteln. Beispiel 551 mod 17.

Teile 551 durch 17, ziehe dann die Zahl vor dem Komma von dem Ergebnis ab und multipliziere dies mit 17.
551 : 17 = 32,41176… - 32 = 0,41176 x 17 = 7.

1.16 Rechenzeit

Die Rechenleistung von Supercomputern und Quantencomputern misst man in TeraFlops. Ein TeraFlop sind 1 Billion Rechnungen pro Sekunde (1 Billion ist 10^{12}).

Der derzeit schnellste Computer berechnet 400 000 TeraFlops pro Sekunde. Das sind 4 x 10^{17} Rechnungen pro Sekunde. Ein Quantencomputer ist angeblich 10^8 mal schneller als ein Supercomputer. Das sind dann 4 x 10^{25} Rechnungen pro Sekunde oder ca. 10^{33} Rechnungen pro Jahr. Wenn dann etwa 10^{45} Rechnungen für eine Entschlüsselung benötigt werden, so dauert das bereits länger als das Universum existiert (1,38 x 10^{10} Jahre).

Die RSA-Verschlüsselung verwendet RSA-Modulo von 1024 Bit oder gar 2056 Bit Länge. Es wird behauptet, dass es ca. $10^{80} – 10^{89}$ Atome im Universum gibt ($2^{83} – 2^{92}$) oder ca. 10^{28} Sandkörner auf der Erde.

1.17 Wie sicher ist der öffentliche Schlüssel?

Wie schon erwähnt, gehört das Modul n zum öffentlichen Schlüssel. Diese Zahl ist das Produkt zweier Primzahlen. Diese beiden Primzahlen dürfen aber auf keinen Fall bekannt sein, da man damit den privaten Schlüssel berechnen kann. Wie ist das möglich ?

Sei n beispielsweise die Zahl 91. Durch Probieren kann man relativ schnell herausfinden, dass 91 das Produkt der beiden Primzahlen 7 und 13 ist. Aber wenn n eine Zahl mit 200 Stellen ist, dann muss man alle Primzahlen zwischen 2 und 10^{97} durchprobieren. Und das dauert eine Weile, denn das sind mehr Zahlen als es Atome im Universum gibt.

2 RSA – Verschlüsselung

Nehmen wir an, Sara möchte Ben eine verschlüsselte Nachricht schicken. Ben denkt sich zwei Primzahlen aus, multipliziert diese und erhält n als p1 x p2. Dann berechnet er Φ(n) aus (p1-1)(p2-1). Ist p1 = 11 und p2 = 17, dann ist n = 187 und Φ(n) = 10 x 16 = 160.
Zu 160 sucht er eine teilerfremde Primzahl e, z. B 7.
n und e (187 und7) bilden den **öffentlichen Schlüssel**, den er Sara schickt.
Damit verschlüsselt Sara ihre Nachricht. Ihre Nachricht sei die Zahl 9 (z.B. 9. Buchstabe im Alphabet). Sie wendet folgende Formel an:

$9^7 \bmod 187 = 70$ 70 ist der verschlüsselte Text.

Ben berechnet nun seinen **privaten Schlüssel** d (als die modulare Inverse) mit der Gleichung
$e \cdot d \equiv 1 \bmod (p1-1)(p2-1)$. Also in diesem Fall $7 \cdot d = 1 \bmod 160$.
Diese Gleichung kann mit dem erweiterten Euklidischen Algorithmus gelöst werden. In diesem Fall ergibt sich für d = 23 da $7 \cdot 23 = 161$ und 161 : 160 hat den Rest 1.

In der Praxis kann d auch mit der Formel $d = \frac{1+(p1-1)(p2-1)}{e}$ (hier: $\frac{1+(11-1)(17-1)}{7} = 23$)

Mit seinem privaten Schlüssel kann Ben nun die Nachricht entschlüsseln.

$70^{23} \bmod 187 \equiv 9$ Damit ist die ursprüngliche Nachricht wieder vorhanden.

Man sieht: Wenn man die verschlüsselte Nachricht 70 liest und die öffentlichen Schlüssel 187 und 7 kennt, ist es unmöglich wieder auf die ursprüngliche Nachricht zu schließen, da man den privaten key nicht kennt. Dazu benötigt man die beiden ursprünglichen Primzahlen, die aber geheim bleiben. In der Realität handelt es sich bei p1 und p2 um Zahlen mit mehr als 512 Bits, die miteinander multipliziert wieder eine mehr als 1024 Bit lange Zahl ergibt (Dies entspricht etwa einer Dezimalzahl mit mehr als 300 Stellen) . Um daraus die beiden Primzahlen wieder zu errechnen, müssen selbst die modernsten Supercomputer sehr lange rechnen (siehe Abschnitt **Rechenzeit**). Selbstverständlich darf auch n keine Primzahl sein, was ja auch nicht sein kann, da es sich aus der Multiplikation von p und q ergibt. Aber wenn n eine Primzahl wäre, könnte man daraus Φ(n) als n-1 berechnen und damit den privaten key knacken.

Noch ein etwas schwierigeres Beispiel, welches aber noch von Hand zu berechnen ist.

Sara möchte die beiden Buchstaben B und x verschlüsseln. Für die Buchstaben wird deren ASCII-Wert als Zahl verwendet.

Klartext m = Bx Umrechnung in ASCII: 66 120

Primzahlen: p1 = 7 p2 = 23

Ben schickt den public key 161 und als zu (7-1)(23-1) = 132 teilerfremde Zahl e = 19 public (161,19).

Sara verschlüsselt nun $66^{19} \bmod 161 = 17$ und $120^{19} \bmod 161 = 99$
der so verschlüsselte **Geheimtext** lautet: 17 99

Ben berechnet seinen privaten Schlüssel d mit der Formel $e \cdot d \equiv 1 \bmod (7-1)(23-1)$.

Das führt zu der Gleichung: $19 \cdot d \equiv 1 \bmod 132$, die man auch schreiben kann als: $1 = 19d - 132c$.
($19d \equiv 1 \bmod 132$ → $19d - 1 = c \cdot 132$ (linke Seite muss ein Vielfaches (c) von 132 sein)
→ $19d - 132c = 1$) oder $-1 = 132c - 19d$ oder $1 = -132c + 19d$

Zur Berechnung von d wendet man wieder den erweiterten euklidischen Algorithmus, wie oben mit der Tabelle beschrieben:

Zeile	a	b	q	c	D
1	132	19	6	**-1**	**7**
2	19	18	1	1	-1
3	18	1	18	0	1
4	1	0			

Erwartungsgemäß ergibt sich als Lösung für den ggT 1, da 19 eine Teilerfremde Zahl zu 132 sein sollte.
Im 2. Schritt rechnen wir von unten nach oben und berechne die Zahl in der 2. Zeile der letzten Spalte aus 0 – 1 x 1 = -1 und in der ersten Zeile 1 – (-1 x 6) = 7. Damit ergibt sich als Lösung: d = 7. Dies ist der private Schlüssel von Ben, den nur er errechnen kann, da nur er die Zahl 132 als (p1-1)(ps-1) kennt.

Ben entschlüsselt: $17^7 \bmod 161 = 66$ und $99^7 \bmod 161 = 120$
Daraus ergeben sich wieder die beiden Zeichen Bx

Hier ein Link zu einem RSA-Rechner https://www.mathespass.at/uni/rsa.php?submit=1

Zusammenfassung:

1. Der erste Schritt besteht darin **das öffentliche Schlüsselpaar** zu erzeugen und diese den Sendern einer verschlüsselten Nachricht zur Verfügung zu stellen. Dafür werden zwei große Primzahlen p und q multipliziert. Das Produkt aus p und q n wird als RSA-Modul bezeichnet und stellt den ersten Teil des öffentlichen Schlüssels dar. Der zweite Teile besteht aus einer Zahl e, die teilerfremd zu dem Produkt (p-1)(q-1), welches als Eulersches phi bezeichnet wird, ist.
   Wichtig: Aus n kann nicht wieder auf die beiden Primzahlen geschlossen werden. ´(Einwegfunktion), da der Rechenaufwand zu groß ist.
2. Wer das öffentliche Schlüsselpaar besitzt kann nun eine verschlüsselte Nachricht c mit der Formel $c = m^e \bmod n$ berechnen und diese versenden.
3. Der Empfänger der Nachricht errechnet seinen **privaten Schlüssel d** mit der Formel
   $e \cdot d = 1 \bmod \Phi(n)$
   Daraus kann d mit dem erw. euklidischen Algorithmus als modulare Inverse berechnet werden.
   (falls bei einfachen Beispielen e und d identisch sein sollten, kann man zu d auch jeweils Φ(n) addieren, um einen anderen Wert zu bekommen.
   Bsp.: für $5 \cdot d = 1 \bmod 6$ gilt d=5 aber auch 11, 17, 23 usw. Dies stimmt deshalb, weil der Modulus ja nach n Schritten wieder auf dem ersten Wert ankommt. (siehe zyklische Gruppe))
4. Die verschlüsselte Nachricht c kann der Empfänger mit seinem privaten key entschlüsseln.
   $m = c^d \bmod n$

Man kann es nicht oft genug wiederholen: Die Grundidee der asymmetrischen Verschlüsselung liegt in der Idee des diskreten Logarithmus:

Nehmen wir an, du kennst die verschlüsselte Nachricht c, die den Wert 8 hat. Du kennst außerdem den public key n mit dem Wert 11 sowie den public key e mit dem Wert 2. Du weißt, dass du die Nachricht mit der Formel $m = c^d \bmod n$ wieder entschlüsseln kannst.

Also stehst du vor der Gleichung $m = 8^d \bmod 11$. Wie willst du diese Gleichung lösen, wenn du d nicht kennst. Man kann durch Probieren, alle Möglichkeiten für d durchprobieren, bis ein sinnvoller Wert für m herauskommt. Aber, allein wenn d eine 1024 Bit-Zahl ist, die umgerechnet über 300 Dezimalstellen hat, erkennt man die Unmöglichkeit dieses Lösungsweges. Nach Empfehlungen des Bundesamtes für Sicherheit in der Informationstechnik soll die Schlüssellänge mindestens 2000 Bit betragen.

Um sicherzugehen, dass eine Nachricht auch von einem bestimmten Absender kommt, verwendet man die digitale Signatur. Dabei geht man umgekehrt vor, wie bei der Verschlüsselung der Nachricht. Diese Informationen werden an die Nachricht angehängt, die er Empfänger dann mit dem öffentlichen Schlüssel wieder entschlüsseln kann.

Im Jahr 2010 wurde erstmals diese 768-Bit lange Zahl in Primfaktoren zerlegt.

1.230.186.684.530.117.755.130.494.958.384.962.720.772.853.569.595.334.792.197.322.
452.151.726.400.507.263.657.518.745.202.199.786.469.389.956.474.942.774.063.845.9
25.192.557.326.303.453.731.548.268.507.917.026.122.142.913.461.670.429.214.311.60
2.221.240.479.274.737.794.080.665.351.419.597.459.856.902.143.413

Das Ergebnis war

33.478.071.698.956.898.786.044.169.848.212.690.817.704.794.983.713.768.568.912.43
1.388.982.883.793.878.002.287.614.711.652.531.743.087.737.814.467.999.489
*****
36.746.043.666.799.590.428.244.633.799.627.952.632.279.158.164.343.087.642.676.03
2.283.815.739.666.511.279.233.373.417.143.396.810.270.092.798.736.308.917

Im Jahr 2020 gelang dies mit einer 829-Bit langen Zahl. Mit jedem zusätzlichen Bit verdoppelt sich die benötigte Rechenzeit.
Da für die Verschlüsselung aber mindestens 1024-Bit oder besser 2056-Bit lange Zahlen verwendet werden, wird es noch eine Weile dauern, bis auch diese entschlüsselt werden können.

3 Diffie-Hellmann Schlüsselaustausch

Das Problem bei RSA sind die langen Schlüssel, die das Verfahren sehr rechenintensiv machen und es dadurch lange dauert, große Datenmengen zu chiffrieren.

Da die RSA-Verschlüsselung aber äußerst sicher ist, verwendet man sogenannte hybride Verschlüsselungsmethoden, bei denen die Verschlüsselung mit einem symmetrischen Algorithmus erfolgt, der Schlüssel selbst aber durch ein asymmetrisches Verfahren erzeugt wird. Die dabei verwendeten Schlüssel haben eine Länge von 56, 128 oder 256 Bits und müssen nur jeweils ein Mal berechnet werden.

Diffie und Hellmann entwickelten ein Verfahren zum Austausch der Schlüssel, bei dem es nahezu unmöglich ist, Rückschlüsse auf die ursprünglichen Werte zu ziehen. Dabei findet wieder der diskrete Logarithmus Anwendung, der die Potenz a aus einer Modulo-Gleichung sucht. $4^a \bmod 11 = 3$ Berechne a. Dies ist bei großen Zahlen nicht lösbar.

Funktionsweise: Lena und Ben wollen sich auf einen gemeinsamen Schlüssel k verständigen.

Beide einigen sich auf eine große Primzahl p und eine kleinere ganze Zahl q, den sog. **Generator**.
Beide Zahlen können öffentlich sein.
Jeder wählt einen persönlichen privaten Schlüssel a, und b, die beide kleiner als p sind.

Daraus berechnen beide die beiden öffentlichen Schlüssel A und B mit den Formeln:

Lena: $A = q^a \bmod p$ Ben: $B = q^b \bmod p$

Anhand von A und B kann aufgrund des Problems des diskreten Logarithmus nicht auf die privaten Schlüssel geschlossen werden.

Beide tauschen A und B aus und berechnen daraus den gemeinsamen Schlüssel k

Lena: $k = B^a \bmod p$ Ben: $k = A^b \bmod p$

Diese beiden Werte müssen identisch sein und können von außerhalb nicht berechnet werden, da der private Schlüssel nicht bekannt ist.

Beispiel:
public: p = 17 q = 5

Lena wählt a = 3 und berechnet $A = 5^3 \bmod 17 = 6$

Ben wählt b = 7 und berechnet $B = 5^7 \bmod 17 = 10$

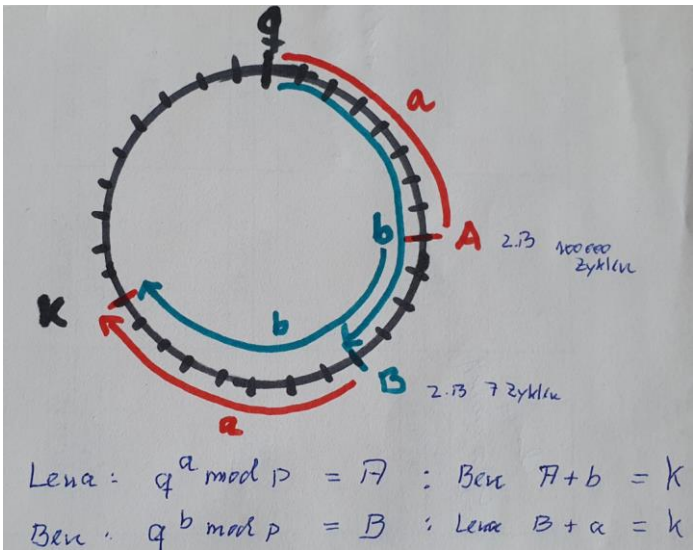
Beide tauschen ihre Nachrichten aus

Lena berechnet k mit $k = 10^3 \bmod 17 = 14$

Ben berechnet k mit $k = 6^7 \bmod 17 = 14$

<u>Warum müssen beide auf denselben Schlüssel kommen?</u>

Da es sich bei der Berechnung des Modulo mit dem Exponenten a oder b um eine zyklische Gruppe handelt, die nach a oder p Schritten wieder bei demselben Divisionsrest landen müssen, kann der Rechenweg anschaulich auf dem Zifferblatt einer Uhr dargestellt werden.



Lena berechnet den öffentlichen Schlüssel A anhand ihres privaten keys **a** und dem Generator **q**. Ben tut das gleiche und erhält B. Obwohl A und B öffentlich sind, weiß niemand, wie viele Runden auf dem Zifferblatt gedreht wurden, da **a und b** nicht bekannt sind. Nach dem Austausch von A und B addiert Lena ihren key a zum Schlüssel B und Ben seinen key b zu A. Beide kommen an der gleichen Stelle der Uhr heraus. D. h. die beiden Werte für k sind identisch.

Anhand von A oder B kann niemand auf a oder b schließen, da ja z.B. 2^5 modulo 11 dasselbe ist wie 2^{15} mod 11 aufgrund der Rechenregeln mit dem Modulus. (In zyklischen Gruppen wird bei einem Generator q jeder Divisionsrest bezüglich eines Modulus n in unregelmäßiger Reihenfolge durchlaufen. Allerdings gelangt man nach n-1 Schritten immer wieder auf den Ausgangspunkt). Niemand kann herausbekommen, wie viele Runden (auf der Uhr) durchlaufen wurden. Und damit kann auch niemand herausfinden, welche Werte a oder b haben.

<u>Auch dieses nochmal erklärt:</u>

Du kennst A, B, den Modulus p und den Generator q. Sei A = 8, B = 12 und p= 11 und q = 2. Du stehst nun vor der Gleichung $8 = 3^a \bmod 11$. Der gesuchte key ist a, denn damit kannst du den gemeinsamen key k berechnen.

Durch Probieren findest du a = 3. Denn $2^3 = 8 \bmod 11$. Nun bist du happy, merkst aber schnell, dass dieser key nicht stimmen kann. Du findest weiterhin, dass für a = 13 die Gleichung auch stimmt, allerdings auch für 23, 33 usw. Welches a ist nun richtig? Du kannst auch hier nicht alle Möglichkeiten durchprobieren, da a eine Zahl von mindestens 512 Bit-Länge ist.

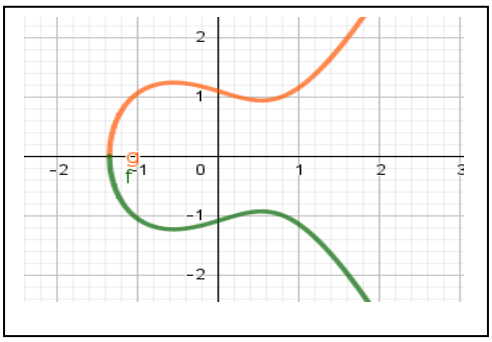
Das größte Problem bei Diffie-Hellman ist der Mangel an Authentifizierung. Verbindungen, die nur mit Diffie-Hellman gesichert werden, sind deswegen anfällig. Diffie-Hellman sollte aus diesem Grund nur zusammen mit einer allgemein akzeptierten Authentifizierungsmethode in Form einer digitalen Signatur eingesetzt werden, um die Identität der Kommunikationspartner über öffentliche Netze überprüfen zu können. Diffie-Hellman eignet sich nichtsdestotrotz gut zur Absicherung von Datenverbindungen, wird aber nur selten für Daten genutzt, die lange Zeit gespeichert und archiviert werden sollen.

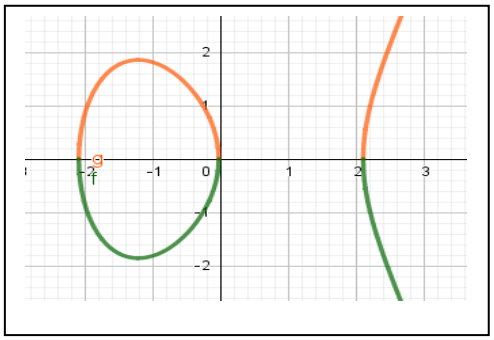
4 Elliptische Kurven - ECC

Elliptische Kurven sind die Grundlage für modernste Verschlüsselungstechniken, wie man sie z. B. bei Kryptowährungen und Blockchains findet. Im Gegensatz zu den anderen Verfahren, bei denen mit Zahlen gerechnet wird, wird bei ECC mit Punkten gerechnet. Das Rechnen mit Punkten unterliegt anderen Rechenregeln. Dadurch ist die Verschlüsselung mit EC sicherer bzw. es genügt eine wesentlich kürzere Schlüssellänge. Die Verschlüsselung mit ECC ist ein **hybrides** Verfahren, bei dem die Nachrichten mit einem symmetrischen Verfahren verschlüsselt werden, die Schlüssel aber asymmetrisch erzeugt werden.

Grundlage ist die Formel für die Ellipse mit $r^2 = ax^2 + by^2$, wobei a und b die Form der Ellipse bestimmen. Die Formel für eine elliptische Kurve lautet : $\mathbf{y^2 = x^3 + ax + b}$

Beispiele für elliptische Kurven:



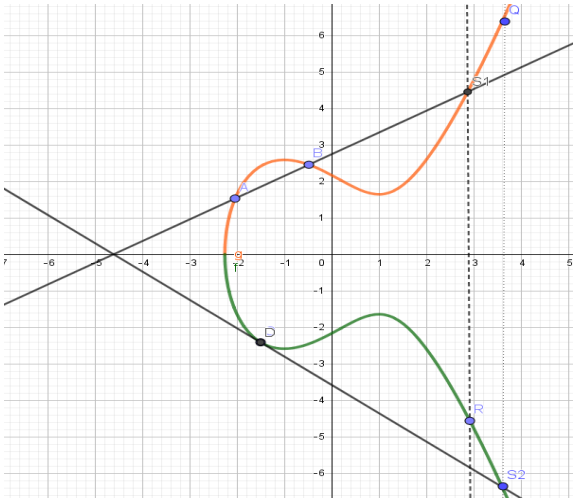


Es gibt eine Reihe von elliptischen Kurven, die sich für ECC eignen.
Für Bitcoin ist a = 0 und b = 7 und die Kurvengleichung lautet: $Y^2 = x^3 + 7$

4.1 Grundlagen

Punkte auf einer elliptischen Kurve werden als eine Gruppe betrachtet, für die eine eigene Arithmetik mit speziellen Regeln gilt. Um zwei Punkte A und B zu addieren, verfährt man wie folgt: Man zieht eine Gerade durch A und B und sucht den 3. Schnittpunkt dieser Gerade mit der Kurve. Dieser liegt in S1. Wenn dieser Punkt an der x-Achse gespiegelt wird, erhält man die Inverse in R. R ist der gesuchte Punkt, der durch die Addition von A und B auf einer elliptischen Kurve entsteht.
A + B = R.

Häufig rechnet man mit nur einem Punkt, den man erhält indem man zwei PunktE D1 und D2 soweit zusammenlegt, dass nur noch ein Punkt D übrigbleibt. In diesem Fall legt man die Tangente an Punkt D und sucht deren Schnittpunkt S2 mit der Kurve. Die Inverse dazu ist wieder das Ergebnis der Punktaddition von D + D. D + D = Q.



Für die Berechnung der **Summe** zweier Punkte ergeben sich die folgenden Formeln:

$A(x_1/y_1) + B(x_2/y_2) = R(x_3/y_3)$

$s = \frac{y2-y1}{x2-x1}$, $x_3 = s^2 - x_1 - x_2$ und $y_3 = s(x_1 - x_3) - y_1$

im Fall der **Punktverdoppelung** gilt für s: $s = \frac{3x1^2+a}{2y1}$ a ist der Faktor a aus der Formel der elliptischen Kurve. Diese Formel ergibt sich aus der ersten Ableitung der allg. Funktionsgleichung der elliptischen Kurve $y^2 = x^3 + ax + b$ oder $y = \sqrt{x^3 + ax + b}$

Anwendung der Kettenregel $y' = \frac{3x^2+a}{2\sqrt{x^3+ax+b}} = \frac{3x^2+a}{2y}$

Mit Kenntnissen der Mittelstufe und etwas Analysis kann man diese Berechnung nachvollziehen: Man berechnet die Steigung der Geraden durch A und B und anschließend die Funktionsgleichung der Geraden. Der dritte Schnittpunkt ergibt sich durch Gleichsetzen: $(sx + c)^2 = x^3 + ax + b$ Man erhält eine Gleichung 3. Grades, die man entweder mit einem geeigneten TR lösen kann oder durch Ausklammern der beiden Linearfaktoren x_1 und x_2.

Weitere Punkte erhält man dadurch, dass man eine Gerade durch R und P legt und wieder den dritten Punkt durch Addition von R und P ermittelt. Damit fährt man fort, wobei immer der zu Beginn gewählte Punkt P beibehalten wird.

Die Punkte, die sich ergeben, bilden eine zyklische Gruppe mit einer bestimmten Anzahl von Elementen (Ordnung). Die Ordnung ergibt sich aus der Anzahl der Rechnungen, die möglich sind, bis sich das neutrale Element ergibt.

4.2 Beispiel

Um die zur Verschlüsselung benötigten Punkte auf einer elliptischen Punkte zu bestimmen, wählt man zunächst einen Punkt P. Ausgehend von der Tangente an den Punkt (Punktverdopplung) erhält man den nächsten Punkt durch Addition (R = 2P). Nun werden R und P addiert und man erhält den 3. Punkt

3P (Dieser ergibt sich als 3. Schnittpunkt der Sekante durch R und P). Es wird wieder eine Sekante durch den neuen Punkt 3P und P gelegt, um 4P zu erhalten usw.

Berechnung: Gegeben ist die elliptische Kurve mit **$y^2 = x^3 + 2x + 2 \mod 17$** und der Punkt **P(5/1).**

Es sollen die ersten 3 Punkte berechnet werden, die sich durch die Addition zweier Punkte ergeben.

1. Im ersten Fall handelt es sich um eine Punktverdoppelung, da nur der Startpunkt P bekannt ist.

$s = \frac{3 \cdot 5^2 + 2}{2 \cdot 1} = \frac{77}{2}$ Für die Division auf elliptischen Kurven wird mit der modularen Inversen mulitpliziert.

$77 \cdot 2^{-1} \mod 17 = 9 \cdot 9 = 81 \equiv 13 \mod 17$ 2^{-1} ist die Lösung der Gleichung $2 \cdot x = 1 \mod 17$

Damit ist s = 13.

$x_3 = 13^2 - 5 - 5 \mod 17 = 159 \mod 17 \equiv 6 \mod 17$ $x_3 = 6$

$y_3 = 13(5 - 6) - 1 \mod 17 = -14 \mod 17 \equiv 3 \mod 17$ $y_3 = 3$ -> **R(6/3)**

2. Nächster Punkt durch Addition von P(5/1) + R(6/3)

$S = \frac{3-1}{6-5} = 2$

$x_3 = 2^2 - 5 - 6 \bmod 17 = -7 \bmod 17 \equiv 10 \bmod 17$ $x_3 = 10$

$y_3 = 2(5 - 10) - 1 \bmod 17 = -11 \bmod 17 \equiv 6 \bmod 17$ $y_3 = 6$ -> **R(10/6)**

3. Nächster Punkt durch Addition von P(5/1) + R(10/6)

$S = \frac{6-1}{10-5} = 5 \cdot 5^{-1} = 5 \cdot 7 \bmod 17 \equiv 1 \bmod 17$

$x_3 = 1^2 - 5 - 10 \bmod 17 = -14 \bmod 17 \equiv 3 \bmod 17$ $x_3 = 3$

$y_3 = 1(5 - 3) - 1 \bmod 17 = 1 \bmod 17 \equiv 1 \bmod 17$ $y_3 = 1$ -> **R(3/1)**

4.3 Verschlüsselung

Zunächst wird ein Schlüssel mit der asymmetrischen Diffie-Hellmann-Methode erzeugt, mit dem dann die Nachricht anschließend symmetrisch codiert wird.

1. Wahl einer elliptischen Kurve, eines Punktes P als Generator und einer Primzahl p.
   Die Länge der Primzahl p ergibt die Sicherheit der Kurve. Bei 256 Bit beträgt die Sicherheit 128 Bit, welches mit 2^{256} eine ausreichende Sicherheit gewährleistet. Bei RSA wäre dafür eine wesentlich längere Zahl erforderlich.

   Bsp.: $y^2 = x^3 + 2x + 2 \bmod p$ P(5/1)

Alice	**public: P($x_{p/y}$p), p**	**Ben**
Private: a <p		Private: b <p
Berechnet A: a · P = (x_A/$y_{A)}$	Austausch von A und B Dieser Punkt ist public	Berechnet B: b · P = (x_B /y_B)
Berechnet a · B = (X_{AB}/$Y_{AB)}$)	Identischer Punkt	Berechnet b · A = (X_{AB}/Y_{AB})

Der öffentliche key ist ein Punkt, also ein Gruppenelement der elliptischen Kurve. Die private keys a und b sind ganze Zahlen (256 Bit), die die Anzahl der Sprünge, d. h. Additionen als Gruppenoperationen angibt.

2. Die zu verschlüsselnde Nachricht m wird dann mit der x-Koordinate X_{AB} mit der AES-Methode verschlüsselt.
   Alice: C = AES(m) Bob: m = AES^{-1}(c)
   Dabei werden alle Eingabewerte in Werte bestimmter Länge umgewandelt.

Alice	**public: P(5/1), p = 19**	**Ben**
Private: a =3		Private: b=10
Berechnet A: 3 · P = (10/6)	Austausch von A und B Dieser Punkt ist public	Berechnet B: 10 · P = (7/11)
Berechnet a · B = 3(7/11) = (13/10)	Identischer Punkt	Berechnet b · A = 10(10/6) = (13/10)

Dabei wird entweder x = 13 oder y = 10 als Sitzungsschlüssel verwendet.

Da sowohl die privaten Schlüssel a als auch b i.d.R. 256 Bit lange Zahlen sind, stellt sich die Frage, wie Alice oder Bob Rechnungen wie a · P oder a · B, die ja eine a-malige Punktaddition bedeuten, in akzeptabler Zeit durchführen können. Dies kann mit der Double-and-add-Methode erreicht werden. Als Beispiel soll die Zahl 227 P berechnet werden. Dies kann durch 227 Additionen erreicht werden. Allerdings gilt ja auch: 2P + 2P = 4 P. D. h., um auf 4 P zu kommen muss man nur 2 Additionen durchführen: P + P = 2P und 2P + 2P = 4 P. Das bedeutet, dass man eine Zahl lediglich in ihre 2er Potenzen zerlegen muss, um die Anzahl der notwendigen Additionen zu ermitteln.
Für 227 gilt: $227 = 128 + 64 + 32 + 2 + 1 = 2^7 + 2^6 + 2^5 + 2^1 + 2^0$. Damit kann man durch die Berechnung von nur 8 2er Potenzen jede beliebige Zahl von 0 bis 255 erzeugen.

5 Digitale Signatur

Ein weiteres Problem der Verschlüsselung ist der Nachweis, dass eine empfangene Nachricht tatsächlich von einem bestimmten Absender stammt. Denn wenn es tatsächlich gelungen wäre, den Schlüssel zu knacken, könnte der böse Hacker die Nachricht zu seinen Gunsten verfälschen, ohne dass dies dem Empfänger auffallen würde. Deshalb ist das Erstellen und das Verifizieren einer Signatur ein weiteres Problem der Kryptographie. Das Prinzip funktioniert ähnlich, wie die Verschlüsselung der Nachricht. Der Absender wendet seinen privaten Schlüssel auf die Nachricht an und der Empfänger prüft, ob die Nachricht, die Signatur und der öffentliche Schlüssel zusammenpassen.
Wie das im Einzelnen funktioniert, soll aber nicht mehr Gegenstand dieser Aufzeichnungen sein.