Bachelorarbeit

Public-Key-Kryptographie und Komplexitätstheorie

Enno Teßmer
Matrikelnummer: 3224370

25. November 2019

Leibniz Universität Hannover
Fakultät fur Elektrotechnik und Informatik
Institut für Theoretische Informatik

Erstprüfer: Prof. Dr. Heribert Vollmer
Zweitprüfer: Dr. Arne Meier
Betreuer: M. Sc. Fabian Müller

Erklärung der eigenständigen Arbeit

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 25. November 2019

Enno Teßmer

Inhaltsverzeichnis

1 Einleitung

Schon seit der Antike haben Menschen versucht „geheime Nachrichten“ auszutauschen. Einer der Ersten uns bekannten Einsätze von Kryptographie fand 1500 v. Chr. statt [1]. Ich halte es jedoch für wahrscheinlich, dass auch schon früher Menschen durch Verschlüsselung versucht haben, Informationen zu schützen. In der Vergangenheit gab es dafür viele berühmte Kryptosysteme, wie die „Caesar-Chiffre“ oder die „Enigma“ Maschine, welche alle jedoch mit der Zeit gebrochen wurden. Im Jahr 1883 formulierte Auguste Kerckhoff das sogenannte Kerckhoffs'sche Prinzip, welches besagt, dass man die Sicherheit eines Kryptosystems nicht auf die Geheimhaltung des Verfahrens, sondern alleinig auf die Geheimhaltung der Schlüssel basieren sollte. Dieses Prinzip ist einer der Grundsätze für die moderne Kryptographie, welche in den 1970er Jahren entstanden ist. Man hat angefangen, mit Hilfe der Komplexitätstheorie, über die Komplexität von Kryptosystemen zu diskutieren, anstatt die Verfahren geheim zu halten. Dank der Komplexitätstheorie kann man in der Kryptographie mit präzisen Annahmen und Definitionen arbeiten, um zu argumentieren wie „schwer“ ein Kryptosystem zu brechen ist. Ein weiterer Durchbruch war der Beginn der Erforschung von Public-Key-Kryptosystemen (auch asymmetrische Kryptosysteme genannt). Vorher gab es lediglich Private-Key-Kryptosysteme (symmetrische Kryptosysteme), wobei der sichere Schlüsselaustausch eines der fundamentalen Probleme für praktische Anwendungen ist. Dieses Problem gibt es bei Public-Key-Kryptosystemen nicht mehr.

In dieser Arbeit möchte ich den Zusammenhang von Kryptographie und Komplexitätstheorie verdeutlichen und die Signifikanz von Public-Key-Kryptographie für praktische Anwendungen hervorheben. Dabei werde ich zuerst die Grundlagen der Kryptographie vorstellen, bei denen ich Sanjeev Arora, Boaz Barak [2] und John M. Talbot, Dominic J. A. Welsh [3] folge. Ich werde dabei nicht genauer auf die Grundlagen der Komplexitätstheorie in dieser Arbeit eingehen, sie können diesbezüglich bei Bedarf in [2] nachlesen. Danach möchte ich Public-Key-Kryptographie etwas genauer bezüglich den zwei Familien, in welche die meisten momentan erforschten Public-Key-Kryptosysteme fallen, untersuchen. Anschließend werde ich mit Hilfe von Impagliazzo's Welten mögliche Auswirkungen auf unsere Welt beleuchten, falls gewisse unbewiesene Annahmen, auf denen die moderne Kryptographie beruht, wahr oder falsch sind. In Anbetracht der Tatsache, dass alle aktuellen praktischen Anwendungen von Kryptosystemen immernoch auf

unbewiesenen Annahmen beruhen, besteht diese Arbeit in gewisser Hinsicht mehr aus einer Betrachtung von Fragen, als von Antworten. Aufgrund der Natur einer solchen Diskussion von Fragen, wird diese Arbeit auch einige rein subjektive Kommentare beinhalten.

2 Moderne Kryptographie

Doch was ist überhaupt ein Kryptosystem? Wir definieren Kryptosysteme als Folgendes:

Definition 2.1 (Kryptosystem)**.** Ein Kryptosystem besteht aus den drei Komponenten Schlüsselerzeugung, Verschlüsselung und Entschlüsselung.

Schlüsselerzeugung: Hier werden die geheimen (und wenn nötig öffentlichen) Schlüssel generiert.

Verschlüsselung: Bei der Verschlüsselung wird der Klartext, welcher übermittelt werden soll, mit Hilfe des Schlüssels zum Geheimtext (auch Ciphertext genannt) umgeformt, welcher dann zum Empfänger versendet wird.

Entschlüsselung: Beim Entschlüsseln wird aus dem empfangenen Geheimtext mit Hilfe des Schlüssels wieder der Klartext ermittelt.

Moderne Kryptographie behandelt aber neben Kryptosystemen die kryptographischen Primitive im Allgemeinen. Diese Primitive sind Bausteine, welche nicht nur zur Erzeugung von Kryptosystemen benutzt werden können, sondern auch für andere kryptographischen Systeme, wie zum Beispiel digitale Signaturen oder sogenannte „zero knowledge proofs“, welche ich aber nicht weiter in dieser Arbeit behandeln werde. Zusammen mit der Kryptoanalyse, bei der es um das Brechen von verschlüsselten Informationen geht, ist die Kryptographie ein Teilgebiet der Wissenschaft der Kryptologie.

2.1 Effiziente Sicherheit statt perfekte Sicherheit

Das absolute Ziel eines Kryptosystems wäre natürlich, dass es für einen Angreifer mit unbegrenztem Rechenaufwand nicht nur unmöglich ist den Klartext aus dem Geheimtext herauszufinden, sondern dass er nicht einmal irgendwelche Teilinformationen herausbekommen kann. Dies nennt man perfekte Sicherheit. Perfekte Sicherheit ist in der Tat sogar möglich, wie das „One-Time-Pad“, ein Private-Key-Kryptosystem, zeigt.

Kryptosystem 1 (One-Time-Pad)**.**
Schlüsselerzeugung: Der Schlüssel k ist ein zufälliges $k \in \{0,1\}^n$ wobei n die Länge des Klartextes ist.

Verschlüsselung: Der Geheimtext c ergibt sich bitweise aus einer XOR-Verknüpfung von Klartext m und Schlüssel k. Bei einem Klartext $m \in \{0,1\}^n$ ist $c = m \oplus k$.

Entschlüsselung: Den Klartext erhält man beim Entschlüsseln wieder aus einer bitweisen XOR-Verknüpfung von Geheimtext c und Schlüssel k, $m = c \oplus k$.

Solange man den Schlüssel nur einmal verwendet, woher auch der Name „One-Time“ kommt, kann ein Angreifer außer der Länge des Klartextes keinerlei Information aus dem Geheimtext ziehen, da jeder mögliche Klartext derselben Länge genau gleich wahrscheinlich der versendete Klartext ist. Dieses Verfahren ist allerdings sehr aufwändig in der Praxis umzusetzen, da der Schlüssel die selbe Länge wie der Klartext besitzen muss und nur einmal verwendet werden darf. Es wurde zudem gezeigt, dass kein Kryptosystem perfekte Sicherheit bieten kann, ohne eine Schlüssellänge von mindestens der Klartextlänge. Ein sicherer Austausch solcher Schlüssel ist ohne weiteres nicht möglich. Das One-Time-Pad wurde jedoch schon z. B. in Regierungsangelegenheiten benutzt [1].

Da also perfekte Sicherheit für die Praxis ungeeignet ist, brauchen wir eine abgeschwächte Definition von Sicherheit, die für uns „sicher genug“ ist. Damit kommen wir zu effizienter Sicherheit (auch komplexitätstheoretische Sicherheit genannt). Angreifer verfügen in der Praxis nur über begrenzte Rechenkapazität, wobei wir von polynomieller Rechenkapazität ausgehen. Deswegen bezeichnen wir ein Kryptosystem als effizient sicher, wenn es sicher gegen einen Angreifer ist, der in Polynomialzeit arbeitet. Auch bei effizienter Sicherheit ist es wichtig, dass Angreifer nicht einmal Teilinformationen herausfinden können. Nun kann ein Angreifer allerdings nicht nur durch gezielte Angriffe versuchen den Klartext zu berechnen, sondern kann auch einfach eine polynomielle Anzahl an zufällig geratenen Werten testen. Mit einem solchen Verfahren ist die Wahrscheinlichkeit, Informationen durch Raten herauszubekommen, bei den meisten Kryptosystemen größer als 0. Wir sagen also desweiteren, dass ein Kryptosystem effizient sicher ist, wenn ein probabilistischer Algorithmus, der einen auf Raten basierenden Ansatz repräsentiert, eine vernachlässigbar kleine Chance auf Erfolg hat.

Definition 2.2 (Probabilistische Turingmaschine)**.**

Eine probabilistische Turingmaschine M ist eine nichtdeterministische Turnigmaschine, dabei besitzt jeder nichtdeterministische Schritt genau zwei mögliche Übergänge und durch einen idealen Münzwurf (also 50/50 Chance) wird entschieden, welcher genommen wird. Dieser Schritt wird auch Coin-flip step genannt. Jeder Zweig b besitzt bei Eingabe w die Wahrscheinlichkeit

$$Pr[b] = 2^{-k}$$

wobei k die Anzahl an coin-flip steps in b ist. Die Wahrscheinlichkeit, dass M bei Eingabe w akzeptiert, ist die Summe der Wahrscheinlichkeiten aller Zweige, die akzeptiert werden. Geschrieben wird die Wahrscheinlichkeit, dass M bei Eingabe w auf einem Zustand mit Ausgabe x hält, als

$$Pr[M(w) = x].$$

Ein probabilistischer Algorithmus kann durch eine probabilistische Turingmaschine repräsentiert werden.

Definition 2.3 (Vernachlässigbarkeit)**.** Eine Funktion $r : \mathbb{N} \to \mathbb{R}$ ist vernachlässigbar, wenn es für jedes positive Polynom $p(k)$ eine Ganzzahl k_0 gibt, so dass $r(k) < \frac{1}{p(k)}$ für jedes $k > k_0$ gilt.

Für einen Angreifer auf ein effizient sicheres Kryptosystem muss die Chance auf ein erfolgreiches Raten also kleiner sein, als der Kehrwert jedes Polynoms. Wäre dies nicht der Fall, könnte ein Angreifer nämlich mit einer polynomiellen Anzahl an Versuchen eine gute Chance darauf haben, einmal richtig zu raten.

2.2 „average case“ Schwere

Eine weitere wichtige Grundlage für moderne Kryptosysteme ist, dass diese nicht auf Problemen beruhen dürfen, die nur im „worst case“ schwer sind, sondern schon im „average case“. Man kann nämlich in der Praxis nicht davon ausgehen, dass beim Verschlüsseln immer ein „worst case“ Fall vorliegt. Stattdessen will man leicht Fälle generieren können, die schwer sind.

Ansonsten könnte man (angenommen $\mathrm{P} \neq \mathrm{NP}$) einfach ein NP-Schweres Problem nehmen und daraus ein Kryptosystem konzipieren, da es für ein

solches Problem keinen effizienten Algorithmus gibt, der es löst. Da diese jedoch häufig nur im „worst case“ schwer sind, geht das nicht so einfach. Wir benötigen in der Praxis Probleme, die in den meisten Fällen, also im „average case“, schwer sind.

3 Private-Key-Kryptographie

Bevor wir zu der Public-Key-Kryptographie kommen, möchte ich erst einmal über Private-Key-Kryptographie und deren Funktionsweise sprechen.

In der Private-Key-Kryptographie wird der geheime Schlüssel sowohl zum Verschlüsseln, als auch zum Entschlüsseln benutzt. Es gibt auch Private-Key-Kryptosysteme, die nicht exakt den selben Schlüssel benutzen. Kennt man bei einem solchen allerdings einen der Schlüssel, kann man einfach den anderen daraus berechnen, weshalb diese Kryptosysteme auch zur Kategorie der Private-Key-Kryptographie gehören.

Die Funktionsweise von Private-Key-Kryptographie beruht fundamental auf dem theoretischen Objekt der sogenannten Einwegfunktion.

Definition 3.1 (Einwegfunktion)**.** Eine in Polynomialzeit berechenbare Funktion $f : \{0,1\}^n \to \{0,1\}^n$ ist eine Einwegfunktion, wenn es für jeden probabilistischen Polynomialzeit Algorithmus A eine vernachlässigbare Funktion r gibt, sodass für jedes n

$$Pr[A(y) = x] \leq r(n),$$

wobei $y = f(x)$ ist.

Eine Einwegfunktion ist demnach im komplexitätstheoretischen Sinn „einfach “ (in Polynomialzeit) berechenbar, aber schwer zu invertieren. Wenn es also eine probabilistische Turingmaschine gibt, die es schafft, die Eingabe aus der Ausgabe zu berechnen, dann nur mit vernachlässigbarer Wahrscheinlichkeit.

Es ist jedoch noch nicht bewiesen ob es Einwegfunktionen tatsächlich gibt und in der Tat würde die Existenz von Einwegfunktionen unter anderem eines der größten Probleme in der Komplexitätstheorie lösen. Es können nämlich Einwegfunktionen nur existieren, wenn $\mathrm{P} \neq \mathrm{NP}$ gilt [2]. Zudem würden Einwegfunktionen die Existenz von vielen kryptographischen Primitiven wie z. B. Pseudozufallszahlengeneratoren (englisch: pseudo random generator) oder digitalen Signaturen implizieren [4]. Einen offentsichtlichen Bezug von Einwegfunktionen auf Private-Key-Kryptosysteme gibt es zwar nicht, es kommt schließlich auch noch kein Schlüssel in irgendeiner Weise vor, jedoch würde die Existenz von Einwegfunktionen unter anderem auch die Existenz vieler Schemata von Private-Key-Kryptosystemen implizieren. Dieser Zusammenhang, der mit Hilfe einer Vielzahl an Werken

gezeigt wurde [4], ist nicht so einfach zu beweisen, weshalb ich nicht genauer darauf eingehen werde. Der Grundgedanke dabei ist, dass mit Hilfe von Pseudozufallszahlengeneratoren die Schlüssellänge wesentlich kleiner als die Nachrichtenlänge sein kann und die Verschlüsselung dennoch sicher bleibt. Siehe [5], wenn Sie Genaueres über Reduktionen von Einwegfunktionen lesen wollen.

Die ganze moderne Kryptographie basiert also auf der Annahme, dass Einwegfunktionen existieren. Mittlerweile gibt es schon eine überwältigende Anzahl an Kandidaten für Einwegfunktionen, was einer der Hauptgründe ist, warum man annehmen könnte, dass Einwegfunktionen existieren. Ein anderer Grund wäre, dass die Existenz sich „natürlich“ anfühlt. Guckt man allein auf simple Dinge in der Welt um sich herum, kann man schon viele Funktionen bzw. Vorgänge sehen, die schwer zu invertieren sind, sei es der Abriss eines Gebäudes, während der Aufbau wieder wesentlich aufwändiger ist, oder simple mathematische Funktionen wie die Multiplikation.

Doch kommen wir zurück zu Private-Key-Kryptosystemen. Da sowohl beim Verschlüsseln als auch beim Entschlüsseln der selbe Schlüssel benutzt wird, muss der Schlüssel vor dem verschlüsselten Informationsaustausch erst einmal sicher zwischen den Parteien ausgetauscht werden. Man braucht also schon einen sicheren Informationskanal zum Schlüsselaustausch, um danach eine Information sicher übermitteln zu können. Dieser Schlüsselaustausch ist eines der großen Probleme für praktische Anwendungen von Private-Key-Kryptosystemen und einer der Gründe der Erforschung von Public-Key-Kryptographie. Bei dieser ist nämlich kein sicherer Schlüsselaustausch mehr nötig.

4 Public-Key-Kryptosysteme

Wie der Name schon suggeriert, gibt es bei den Public-Key-Kryptosystemen nicht nur einen privaten Schlüssel, sondern auch einen öffentlichen Schlüssel. Mit diesem öffentlichen Schlüssel wird die Information verschlüsselt, während mit dem privaten Schlüssel entschlüsselt wird. Da öffentliche Schlüssel für jeden zugänglich sind, kann jeder eine verschlüsselte Nachricht an den Besitzer des privaten Schlüssels senden. Hierfür wird kein sicherer Schlüsselaustausch benötigt. Dies ist der große Vorteil von Public-Key-Kryptosystemen gegenüber Private-Key-Kryptosystemen.

Bei der Funktionsweise benötigen Public-Key-Kryptosysteme eine spezielle Form von Einwegfunktionen, die sogenannten Falltürfunktionen (englisch: trapdoor functions).

Definition 4.1 (Falltürfunktion)**.** Eine in Polynomialzeit berechenbare Funktion $f : \{0,1\}^n \to \{0,1\}^n$ ist eine Falltürfunktion, wenn es für jeden probabilistischen Polynomialzeit Algorithmus A eine vernachlässigbare Funktion r gibt, sodass für jedes n

$$Pr[A(y) = x] \leq r(n),$$

wobei $y = f(x)$ ist.

Zusätzlich gibt es mit Kenntniss einer Falltür t einen Polynomialzeit Algorithmus B, mit $B(y,t) = x$.

Eine Falltürfunktion ist also eine Einwegfunktion, die aber mit Hilfe eines Geheimnisses t (der Falltür) leicht invertierbar ist. Anders als der Zusammenhang zwischen Einwegfunktionen und Private-Key-Kryptographie, ist die Notwendigkeit von Falltürfunktionen für Public-Key-Kryptosysteme wesentlich klarer. Man kann ein Public-Key-Kryptosystem einfach als Falltürfunktion sehen, wobei das Geheimnis t der private Schlüssel ist. Der private Schlüssel darf dabei natürlich auch nicht effizient aus dem öffentlichen Schlüssel berechnet werden können.

4.1 Algebraische Familie (Abelian group)

Kommen wir nun zur ersten und bisher am meisten erforschten Familie von Public-Key-Kryptosystemen. Bei der Familie der algebraischen Public-Key-Kryptosysteme handelt es sich um Systeme, die auf der Schwierigkeit algebraischer Probleme basieren [4]. Zu diesen gehören unter anderem Kryptosysteme basierend auf dem Faktorisierungsproblem, oder dem Diskretem Logarithmus, welche zu den meist erforschten und benutzten Systemen zählen.

Definition 4.2 (Das Faktorisierungsproblem)**.** Gegeben ist eine zusammengesetzte Zahl n, die das Produkt zweier unterschiedlicher k-bit Primzahlen p und q ist.

Das Problem ist es, n zu faktorisieren, also p und q zu bestimmen.

Bei dem Faktorisierungsproblem handelt es sich also um den Vorgang, eine zusammengesetzte Zahl in ihre beiden Primfaktoren zu zerlegen. Bis heute ist kein effizientes Verfahren bekannt, um dies zu bewerkstelligen. Das Faktorisierungsproblem wird deshalb auch als Kandidat für Einwegfunktionen angesehen. Ein auf dem Faktorisierungsproblem beruhendes Kryptosystem ist das „Rabin-Kryptosystem“, welches im Jahr 1979 von dem israelischen Informatiker Michael O. Rabin veröffentlicht wurde [6] und das erste Public-Key Kryptosystem war, das beweisbar mindestens genauso schwer zu lösen ist, wie das Faktorisierungsproblem. Dieses werde ich nun etwas genauer erläutern, um die Funktionsweise asymmetrischer Kryptosysteme aus der algebraischen Familie zu verdeutlichen.

Definition 4.3 (Die Faktorisierungsannahme)**.** Für jeden probabilistischen Polynomialzeit Algorithmus A ist die Wahrscheinlichkeit, dass dieser das Faktorisierungsproblem löst, also n faktorisiert, vernachlässigbar. Es gibt also eine vernachlässigbare Funktion r, sodass bei genügend großem k

$$Pr[A(n) = (p, q)] \leq r(k)$$

wobei p und q zufällige k-bit Primzahlen sind und $n = pq$.

4.1.1 Das Rabin-Kryptosystem

Kryptosystem 2 (Rabin-Kryptosystem)**.**

Schlüsselerzeugung: Privater Schlüssel p und q, mit $p \neq q$, sind Primzahlen mit der Kongruenzbedingung $p \equiv q \equiv 3 \ (mod\ 4)$.

Öffentlicher Schlüssel ist $n = p \cdot q$.

Verschlüsselung: Geheimtext c ergibt sich aus Klartext m mit $c = m^2 \ (mod\, n)$. Dabei muss $m \in \{0, 1, ..., n-1\}$ sein.

Entschlüsselung: Berechne y_p und y_q mit $y_p \cdot p + y_q \cdot q = 1$
und m_p, m_q mit

$$m_p^2 = c \ (mod\ p)$$
$$m_q^2 = c \ (mod\ q).$$

Berechne damit die Quadratwurzeln

$$\begin{aligned} r &= (y_p \cdot p \cdot m_q + y_q \cdot q \cdot m_p) \ (mod\ n) \\ -r &= n - r \\ s &= (y_p \cdot p \cdot m_q - y_q \cdot q \cdot m_p) \ (mod\ n) \\ -s &= n - s \end{aligned}$$

von denen eine der ursprüngliche Klartext m ist.

Hierbei ist die Kongruenzbedingung der Primzahlen notwendig, um

$$m_p^2 = c \ (mod\ p)$$
zu
$$m_p \equiv c^{\frac{p+1}{4}} \ (mod\ p)$$

umformen zu können.

y_p und y_q kann man mit Hilfe des erweiterten euklidschen Algorithmus berechnen.

Für die Entschlüsselung ist kein effizientes Verfahren bekannt ohne die Kenntnis der beiden Primzahlen p und q, dessen Produkt n den öffentlichen Schlüssel bildet. Könnte man das Faktorisierungsproblem effizient lösen, könnte man also auch das Rabin-Kryptosystem effizient knacken. Zudem

wäre es auch umgekehrt möglich, wenn es ein effizientes Verfahren um das Rabin-Kryptosystem zu knacken gäbe, also eine Quadratwurzel modulo n zu berechnen, mit Hilfe von diesem Verfahren eine zusammengesetzte Zahl effizient zu faktorisieren. Denn kennt man ein Paar (a, b), wobei a und b Quadratwurzeln einer Zahl x $(mod\ n)$ sind und $a \neq b$ und $a \neq -b$, dann ist $a = b$ $(mod\ p)$ und $a = -b$ $(mod\ q)$ oder anders herum (tausche p und q). Damit kann man ein $c = b - a$ $(mod\ n)$ berechnen, das ein Vielfaches von entweder p oder q ist. Dann berechnet man den größten gemeinsamen Teiler von c und n und erhält p bzw. q. Man könnte also dem Algorithmus, der das Rabin-Kryptosystem effizient entschlüsseln kann, ein selbst gewähltes a^2 als Geheimtext geben und würde zu 50% Wahrscheinlichkeit, da eine Quadratzahl modulo n (einer zusammengesetzten Zahl aus 2 Primzahlen) vier Quadratwurzeln hat, ein b als Ergebnis bekommen, das weder a noch $-a$ ist. Durch wiederholtes Versuchen kann diese Wahrscheinlichkeit beliebig nah an 100% kommen, ohne die 100% jedoch erreichen zu können. Damit hat man ein Paar und kann mit dem gezeigten Verfahren einen Primfaktor von n herausfinden. Den größten gemeinsamen Teiler zu berechnen ist auch in Polynomialzeit möglich, man könnte also die Zahl n effizient faktorisieren. Solange die Faktorisierungsannahme stimmt, kann man den Geheimtext also nur effizient entschlüsseln, wenn man die Faktorisierung von n kennt. Mehr über diesen Beweis können Sie in [6] nachlesen.

In der Praxis werden sehr große Primzahlen (in der Größenordnung von 10^{200} und größer) verwendet, um den Aufwand der Entschlüsselung für Angreifer erheblich zu vergrößern. Zudem wird der Klartext in verschiedene Blöcke aufgeteilt, die einzeln verschlüsselt versendet werden, falls der komplette Klartext nicht die Anforderung $m \in \{0, 1, ..., n-1\}$ erfüllt.

Ein Problem des Rabin-Kryptosystems ist jedoch, dass man vier potentielle Ergebnisse beim Entschlüsseln bekommt; die vier berechneten Quadratwurzeln. Eine Aussage darüber treffen, welches dieser Ergebnisse der ursprüngliche Klartext war, kann man ohne weiteres nicht. Es ist allerdings relativ einfach möglich das richtige Ergebnis herauszufinden, falls der Klartext eine gewisse Struktur aufgewiesen hat; hierbei reicht zum Beispiel schon, dass er in deutscher Sprache verfasst wurde. Solche Strukturen können allerdings die Sicherheit der verschlüsselten Nachricht erheblich schwächen. Nachfolgend ist ein selbstgewähltes Beispiel, um die Funktionsweise des Rabin-Kryptosystemes zu veranschaulichen.

Beispiel (Rabin-Kryptosystem mit Zahlen)**.** Wir wählen als privaten Schlüssel $p = 7$ und $q = 19$, die Kongruenzbedingung der beiden Primzahlen ist erfüllt: $7 \equiv 19 \equiv 3 \ (mod\ 4)$. Nun errechnet sich daraus der öffentliche Schlüssel

$$n = p \cdot q = 7 \cdot 19 = 133.$$

Als Klartext wählen wir $m = 12$, woraus sich der Geheimtext

$$c \equiv m^2 \ (mod\ n) \equiv 144 \ (mod\ 133) \equiv 11 \ (mod\ 133)$$

berechnet.

Zur Entschlüsselung berechnen wir zunächst m_p und m_q:

$$m_p \equiv c^{\frac{p+1}{4}} \ (mod\ p) \equiv 11^{\frac{7+1}{4}} \ (mod\ 7) \equiv 121 \ (mod\ 7) \equiv 2 \ (mod\ 7)$$
$$m_q \equiv c^{\frac{q+1}{4}} \ (mod\ q) \equiv 11^{\frac{19+1}{4}} \ (mod\ 19) \equiv 161051 \ (mod\ 19) \equiv 7 \ (mod\ 19).$$

Als nächstes berechnen wir

$$1 = y_p \cdot p + y_q \cdot q = y_p \cdot 7 + y_q \cdot 19 = -8 \cdot 7 + 3 \cdot 19 = -56 + 57 = 1$$

und erhalten somit y_p= -8 und y_q= 3.

Nun können wir die vier Quadratwurzeln errechnen:

$$\begin{aligned} r &= (y_p \cdot p \cdot m_q + y_q \cdot q \cdot m_p) \ (mod\ n) \\ &= (-8 \cdot 7 \cdot 7 + 3 \cdot 19 \cdot 2) \ (mod\ 133) \\ &= (-392 + 114) \ (mod\ 133) \\ &= 121 \ (mod\ 133) \end{aligned}$$

$$-r = n - r = 133 - 121 = 12$$

$$\begin{aligned} s &= (y_p \cdot p \cdot m_q - y_q \cdot q \cdot m_p) \ (mod\ n) \\ &= (-8 \cdot 7 \cdot 7 - 3 \cdot 19 \cdot 2) \ (mod\ 133) \\ &= (-392 - 114) \ (mod\ 133) \\ &= 26 \ (mod\ 133) \end{aligned}$$

$$-s = n - s = 133 - 26 = 107.$$

Wir sehen, dass auch unser ursprünglicher Klartext $m = 12$ eine dieser vier Quadratwurzeln ist.

Sowohl das Faktorisierungsproblem als auch der diskrete Logarithmus sind in der Komplexitätsklasse TFNP (Total Function Nondeterministic Polynomial). In dieser Klasse sind Funktionsprobleme, deren Lösungen in Polynomialzeit auf Korrektheit überprüft werden können und welche auch garantiert mindestens eine korrekte Lösung besitzen. Es ist noch nicht geschafft und womöglich nicht möglich, Problemen dieser Klasse NP-Schwere nachzuweisen. Bei üblichen Nachweisen von NP-Schwere, zum Beispiel mit Hilfe einer Cook Reduktion von einem NP-Schweren Problem A auf Problem B, versucht man für Instanzen x eine Übergangsfunktion f zu finden, so dass $x \in \mathrm{A} \Leftrightarrow f(x) \in \mathrm{B}$. Bei den Problemen in TFNP gibt es allerdings immer eine korrekte Lösung weshalb, eine solche Reduktion als unmachbar erscheint. Es ist zudem bewiesen [7], dass, wenn ein TFNP Problem NP schwer ist, ist NP = coNP, wovon generell angenommen wird, dass es nicht stimmt. Dies ist allerdings nur eine Vermutung für eins der großen offenen Probleme in der Komplexitätstheorie. Sie können in [7], [8] mehr über TFNP lesen.

Wir können also noch nicht die Schwere der Probleme, auf denen die algebraischen Kryptosysteme beruhen, beweisen. Es gibt allerdings mit dem Shor-Algorithmus [9] schon einen polynomialzeit Quantenalgorithmus, der sowohl das Faktorisierungsproblem, als auch den Diskreten Logarithmus löst. Die Public-Key Kryptosysteme in der algebraischen Familie wären also definitiv nicht sicher, falls genügend große Quantencomputer in Zukunft Realität werden.

4.2 Geometrische Familie

In der zweiten großen Familie von Kryptosystemen, der Familie der geometrischen Kryptosysteme, sind vor allem auf Gitter (englisch: lattices) oder auf Coding basierende Kryptosysteme [4]. Auch Kryptosysteme, die auf Problemen wie Knappsack basieren, gehören zu dieser Familie. Auf diese werde ich jedoch wenig genauer eingehen.

Public-Key Kryptosysteme, basierend auf diesen Problemen, sind im Moment unsere beste Hoffnung auf sichere Kryptosysteme, falls die Probleme in der algebraischen Familie aus einem quantentechnischen (oder vielleicht auch anderem) Grund nicht mehr sicher sind. In [10] können Sie einen Überblick über Kryptographie, die auf Gittern basieren, finden.

Ein Problem basierend auf Gittern ist das „Learning with errors“, auch LWE genanntes Problem, welches von dem Wissenschaftler Oded Regev vor-

gestellt wurde [11]. Dieses basiert darauf, dass das Gaußsche Eliminationsverfahren über endliche Körper nicht mehr funktioniert, wenn eine gewisse zufällige Störung in den Gleichungen vorhanden ist. Es ist zwar möglich mit dem „least squares minimization algorithm“ lineare Gleichungen, mit zufälliger Störung, über den reellen Zahlen zu lösen, doch gibt es noch kein Verfahren analog zu diesem, wenn man in einer diskreten Umgebung arbeitet [4] (z.B. wenn man mit ganzen Zahlen arbeitet, oder alle Gleichungen in einem $\mathbb{Z}_p$ berechnet). Im nachfolgenden Unterkapitel werde ich ein auf der LWE Annahme basiehrendes Kryptosystem vorstellen. Dabei werden [11] und [12] als Quellen benutzt, sofern nicht anders angegeben.

Definition 4.4 ($\mathbb{Z}_q^n$)**.** $\mathbb{Z}_q$ ist die Menge aller ganzer Zahlen modulo eines q, also $\mathbb{Z}_q^n = \{0, 1, 2, ..., q-1\}$.

$\mathbb{Z}_q^n$ bezeichnet die Menge aller Vektoren, die ein n-Tupel ganzer Zahlen modulo q sind, also beispielsweise $\mathbb{Z}_q \times \mathbb{Z}_q \times \mathbb{Z}_q$ für $n = 3$.

4.2.1 Das LWE-Kryptosystem

Im Folgenden bezeichnet $\langle a, s \rangle$ das Skalarprodukt von a und s und „ $\gg$ “ bezeichnet „viel größer als“. Die Störungsverteilung über $\mathbb{Z}_q$ ist eine Normalverteilung mit einer Standardabweichung von $\alpha \cdot q$, mit $0 < \alpha < \frac{1}{\sqrt{m}}$. Die Werte der Störungsverteilung werden auf die nächste ganze Zahl gerundet.

Definition 4.5 (LWE Problem)**.** Sei s ein festes Element aus $\mathbb{Z}$. Gegeben ist ein $n \geq 1$, eine Primzahl q, sowie eine polynomielle Anzahl an Beispielpaaren (a, b) wobei a ein zufällig gewähltes Element aus $\mathbb{Z}_q^n$ ist und

$$b = \langle a, s \rangle + e \ (mod\ q)$$

mit e aus einer Störungsverteilung über $\mathbb{Z}_q$.

Das Problem ist es s zu bestimmen.

Definition 4.6 (LWE Annahme)**.** Für jeden probabilistischen polynomialzeit Algorithmus A ist die Wahrscheinlichkeit, dass dieser das LWE Problem löst, vernachlässigbar. Es gibt also eine vernachlässigbare Funktion r, sodass bei genügend großem n

$$Pr[A(a,b) = s] \leq r(n)$$

wobei s, a, b den gleichnamigen Variabeln aus dem LWE-Problem entsprechen.

Kryptosystem 3 (LWE Public-Key Kryptosystem)**.**

Schlüsselerzeugung: Seien $n \in \mathbb{Z}$ und q eine Primzahl.

Der private Schlüssel ist ein zufällig gewähltes $s \in \mathbb{Z}_q^n$.

Der öffentliche Schlüssel besteht aus n, q, und M, wobei M eine Anzahl an Paaren $(a_1, b_1), ..., (a_m, b_m)$ist, mit einem $m \gg n \cdot \log q$. Dabei ist a_i ein zufällig gewähltes Element aus $\mathbb{Z}_q^n$ und

$$b = \langle a, s \rangle + e \ (mod\ q)$$

mit e aus einer Störungsverteilung über $\mathbb{Z}_q$.

Verschlüsselung: Zur Verschlüsselung eines Nachrichtenbits x wähle eine zufällige Teilmenge T von M, wobei (a_i', b_i') das i-te Element in T ist.

Errechne nun u und v, welche die verschlüsselte Nachricht bilden mit

$$u = (\sum_{i=1}^{|T|} a_i') \ (mod\ q)$$
$$v = (\sum_{i=1}^{|T|} b_i') + \left\lfloor \frac{q}{2} \right\rfloor \cdot x \ (mod\ q).$$

Entschlüsselung: Berechne d mit $d = v - s \cdot u \ (mod\ q)$. Falls d näher an 0 als an $\lfloor \frac{q}{2} \rfloor$ ist, war das Nachrichtenbit eine 0, ansonsten eine 1.

Die Störungen werden gering gehalten, damit ein korrektes Entschlüsseln möglich ist, dazu muss die Summe aller addierten Störungen kleiner sein als $\frac{p}{4}$, was mit diesen Einschränkungen zu sehr großer Wahrscheinlichkeit gegeben ist. Nun folgt zur Veranschaulichung ein selbstgewähltes Beispiel zum LWE Kryptosystem.

Beispiel (LWE Kryptosystem mit Zahlen)**.** Wir wählen $n = 1$, um es einfach zu halten, $p = 89$ und den privaten Schlüssel $s = 6$. Für den Teil M des öffentlichen Schlüssels wählen wir $m = 20$ und

$$a = [24, 52, 69, 77, 14, 4, 85, 34, 70, 15, 42, 48, 17, 29, 53, 1, 16, 36, 62, 30]$$

und eine Störungsverteilung

$$e = [1, 4, 2, 3, 2, 2, 4, 1, 1, 3, 2, 4, 4, 3, 2, 2, 3, 2, 2, 1].$$

Daraus berechnen sich

$$b_1 = a_1 \cdot s + e_1 \ (mod\ p) = 24 \cdot 6 + 1 \ (mod\ 89) = 145 \ (mod\ 89) = 56$$
$$b_2 = a_2 \cdot s + e_2 \ (mod\ p) = 52 \cdot 6 + 4 \ (mod\ 89) = 316 \ (mod\ 89) = 49$$

und so weiter für die restlichen b und wir erhalten insgesamt

$$b = [56, 49, 60, 20, 86, 26, 69, 27, 65, 4, 76, 25, 17, 88, 53, 8, 10, 40, 18, 3].$$

Der Teil M des öffentlichen Schlüssels ist also

$$M = [(24, 56), (52, 49), (69, 60), (77, 20), (14, 86), (4, 26), (85, 69), (34, 27), (70, 65), (15, 4), (42, 76), (48, 25), (17, 17), (29, 88), (53, 53), (1, 8), (16, 10), (36, 40), (62, 18), (30, 3)].$$

Nun besteht die zu übertragende Nachricht aus 2 Bits, einer 1 und einer 0. Zum Verschlüsseln der 1 wählen wir als Teilmenge T_1 die Paare 4, 9, 2, 16, 13 und zum Verschlüsseln der 0 für T_0 die Paare 18, 11, 4, 7, 6. Wir erhalten also

$$T_1 = [(77, 20), (70, 65), (52, 49), (1, 8), (17, 17)]$$
$$T_0 = [(36, 40), (42, 76), (77, 20), (85, 69), (4, 26)].$$

Wir berechnen den Geheimtext u_1, v_1 und u_0, v_0:

$$\begin{aligned}
u_1 &= (\sum_{i=1}^{|T|} a'_i) \ (mod\ q) = 77 + 70 + 52 + 1 + 17 \ (mod\ 89) \\
&= 217 \ (mod\ 89) = 39 \\
v_1 &= (\sum_{i=1}^{|T|} b'_i) + \frac{q}{2} \cdot x \ (mod\ q) \\
&= 20 + 65 + 49 + 8 + 17 + \left\lfloor \frac{89}{2} \right\rfloor \cdot 1 \ (mod\ 89) \\
&= 159 + 44 \ (mod\ 89) = 25 \\
u_0 &= (\sum_{i=1}^{|T|} a'_i) \ (mod\ q) = 36 + 42 + 77 + 85 + 4 \ (mod\ 89) \\
&= 244 \ (mod\ 89) = 66 \\
v_0 &= (\sum_{i=1}^{|T|} b'_i) + \frac{q}{2} \cdot x \ (mod\ q) \\
&= 40 + 76 + 20 + 69 + 26 + \left\lfloor \frac{89}{2} \right\rfloor \cdot 0 \ (mod\ 89) \\
&= 231 + 0 \ (mod\ 89) = 53.
\end{aligned}$$

Bei der Entschlüsselung wird nun d_1, d_0 berechnet mit

$$\begin{aligned}
d_1 &= v_1 - s \cdot u_1 \ (mod\ q) = 25 - 6 \cdot 39 \ (mod\ 89) = 58 \\
d_0 &= v_0 - s \cdot u_0 \ (mod\ q) = 53 - 6 \cdot 66 \ (mod\ 89) = 13.
\end{aligned}$$

Wir sehen, dass d_1 näher an $\lfloor \frac{89}{2} \rfloor$ und d_0 näher an 0 ist. Die Entschlüsselung ergibt also, dass bei d_1 eine 1 und bei d_0 eine 0 der Klartext war, was auch unserer gewählten Nachricht entspricht.

Im Gegensatz zu den algebraischen Problemen, wie dem Faktorisierungsproblem, sind allerdings auf Gittern basierende Probleme, wie das LWE Problem, noch wesentlich neuer und deswegen auch noch nicht so genau untersucht, weshalb man bei Annahmen der Schwere solcher Probleme noch etwas skeptischer sein kann. Es wurde jedoch schon gezeigt, dass LWE schwer zu lösen ist, wenn andere „worst case“ Gitter Probleme wie zum Beispiel „GapSVP“ (ein Entscheidungsproblem, bei dem es darum geht in einem Gitter den kürzesten Vektor zu finden) schwer sind [13].

Das LWE Problem ist eine sehr flexible Basis für Kryptosysteme, weshalb heutzutage viele Ideen neuer Kryptosysteme auf der LWE Annahme

basieren. Auch praktischere Versionen des LWE Kryptosystems, wie „Ring LWE“ [14], sind ein großes Forschungsthema. Es wird sogar vermutet, dass die LWE Annahme gegen Quantencomputer hält [15].

Das LWE Kryptosystem und alle anderen bekannten, auf Gittern basierenden Kryptosysteme sind allerdings effizient lösbar, falls $\mathrm{NP} \cap \mathrm{coNP} \subseteq \mathrm{P}$ gilt. Auch wenn die meisten Experten nicht glauben, dass $\mathrm{NP} \cap \mathrm{coNP}$ in P enthalten ist, zeigt dieser Zusammenhang zumindest, dass solche Kryptosysteme zumindest gewisse rechnerische Strukturen aufweisen, die viele andere Kandidaten für Einwegfunktionen nicht haben [4].

4.3 Probleme und Sicherheit von Public-Key-Kryptosystemen

Public-Key-Kryptosysteme haben natürlich auch Nachteile oder Schwachstellen. Wir wissen, dass die Kryptosysteme aus der algebraischen Familie nicht gegen Quantencomputer sicher sind, doch gibt es noch andere, grundlegendere Probleme von Public-Key-Kryptosystemen.

Gegenüber den Private-Key-Kryptosystemen, bei denen perfekte Sicherheit möglich, wenn auch bisher unpraktisch ist, gibt es noch kein Public-Key-Kryptosystem, das perfekte Sicherheit bietet und es ist vermutlich sogar gar nicht möglich. Ich möchte die Wahrscheinlichkeit jedoch nicht ausschließen. Ein Angreifer der über unbegrenzten Rechenaufwand verfügt, kann jede von einem bisherigen Public-Key-Kryptosystem verschlüsselte Information brechen. Die meisten, eventuell sogar alle Probleme, die hinter den Public-Key-Kryptosystemen stehen, können mit unbegrenztem Rechenaufwand gebrochen werden. Selbst wenn dies aus irgendwelchen Gründen nicht möglich ist, kann ein solcher Angreifer auch den Geheimtext eines deterministischen Kryptosystems knacken. Da deterministische Public-Key-Kryptosysteme bei dem Verschlüsseln eines Klartextes mit dem selben öffentlichen Schlüssel immer den selben Geheimtext erzeugen, kann ein Angreifer mit unbegrenztem Rechenaufwand einfach alle Möglichkeiten beim Verschlüsseln durchprobieren und wird irgendwann den Geheimtext herausbekommen. Selbst bei Public-Key-Kryptosystemen, bei denen mehr als ein Klartext beim Verschlüsseln den selben Geheimtext erzeugen, wie z. B. das Rabin-Kryptosystem, würde der Angreifer alle Möglichkeiten, bei dem Rabin-Kryptosystem die vier Quadratwurzeln, herausfinden. Er hätte den Geheimtext also effektiv entschlüsselt, da der Empfänger, wie wir bei dem Rabin-Kryptosystem gesehen haben, auch nur die vier verschiedenen Möglichkeiten herausbe-

kommt. Ein weiterer Ansatz zum Brechen eines Public-Key-Kryptosystems wäre das durchprobieren möglicher privater Schlüssel, bis man einen erhält, der zum öffentlichen Schlüssel passt. Selbst wenn es mehrere mögliche private Schlüssel gibt, die zu dem öffentlichen Schlüssel passen würden, müssen diese jedoch alle die Vorgabe

$$\mathrm{Dec}(\mathrm{Enc}(m, pk), sk) = m$$

erfüllen. Hierbei steht m für den Klartext, $\mathrm{Enc}(m, pk)$ für die Verschlüsselung mit dem öffentlichen Schlüssel pk und $\mathrm{Dec}(\mathrm{Enc}(m, pk), sk)$ für die Entschlüsselung mit dem privaten Schlüssel pk. Ein Public-Key-Kryptosystem muss stets bei einer Entschlüsselung eines Geheimtextes den zuvor verschlüsselten Klartext wieder erzeugen, solange jeweils zusammengehörende private und öffentliche Schlüssel benutzt wird.

Public-Key-Kryptosysteme können also vermutlich niemals perfekte Sicherheit bieten, doch dies ist für praktische Anwendungen im Moment sowieso nicht unbedingt nötig. In der Praxis sind Angreifer immer im Rechenaufwand begrenzt, weshalb uns effiziente Sicherheit ausreicht. Es gibt jedoch auch bei Angriffen mit polynomiellem Aufwand Probleme bei Public-Key-Kryptosystemen. Hierbei muss man zwischen verschiedenen Angriffszenarien unterscheiden, da Angreifer über verschiedene Informationen verfügen können.

Ciphertext Only: Bei einem Ciphertext Only Angriff kennt der Angreifer nur Geheimtexte (einen oder mehrere).

Kown Plaintext: Hierbei kennt der Angreifer Paare von Klartexten und den dazu gehörigen Geheimtexten.

Chosen Plaintext: Der Angreifer kann hier selbst gewählte Klartexte verschlüsseln lassen und erhält die dazugehörenden Geheimtexte. Es gibt dabei auch eine adaptive Variante. Bei dieser kann der Angreifer, nachdem er die erhaltenen Paare von Klar- und Geheimtext analysiert hat, weitere Klartexte verschlüsseln lassen.

Chosen Ciphertext: Bei einem Chosen Ciphertext Angriff kann der Angreifer selbst gewählte Geheimtexte entschlüsseln lassen und erhält die dazugehörigen Klartexte. Hier gibt es auch eine adaptive Variante, wobei der Angreifer über längere Zeit die Möglichkeit hat, Geheimtexte entschlüsseln zu lassen.

Chosen Text: Eine Kombination von Chosen Plaintext und Chosen Ciphertext. Ein Angreifer kann also gewählte Klartexte verschlüsseln und Geheimtexte entschlüsseln lassen. Auch hier gibt es wieder eine adaptive Variante.

Die Stärke der Angriffe ist hierbei aufsteigend sortiert. Der Chosen Text Angriff ist also der Angriff, der dem Angreifer die besten Möglichkeiten zugesteht. Angriffe auf Public-Key-Kryptosysteme sind normalerweise mindestens so stark wie Chosen Plaintext Angriffe, da die Verschlüsselungsmethode und der dazu benötigte Schlüssel öffentlich sind. Ein Chosen Ciphertext Angriff auf ein Public-Key-Kryptosystem ist deswegen auch identisch mit einem Chosen Text Angriff, die Differenzierung spielt nur für Angriffe auf Private-Key-Kryptosysteme eine Rolle. In der Praxis sehen solche Angriffe (bei denen nicht nur die öffentlich vorhandenen Mittel, der öffentliche Schlüssel zum Verschlüsseln bzw. der abgelauschte Geheimtext benutzt werden) so aus, dass der Angreifer z. B. bei jemandem eingebrochen ist, sich Zugang zu dessen Laptop verschafft hat, um selbst gewählte Geheimtexte entschlüsseln lassen zu können. Mit den adaptiven Varianten wird die Situation beschrieben, in der der Angreifer über längere Zeit Zugang zu diesem System (der Laptop in dem Beispiel) hat.

Manche Kryptosysteme weisen gewisse Strukturen auf, sodass der Schlüssel mit solchen Angriffen ermittelt werden kann. Bei dem zuvor vorgestellten Rabin-Kryptosystem kann z. B. mit einem Chosen Ciphertext Angriff der geheime Schlüssel ermittelt werden [16]. Doch, auch wenn mit solchen Angriffen nicht unbedingt der Klartext oder geheime Schlüssel ermittelt werden kann, ist schon die Möglichkeit Teilinformationen über die Nachricht herauszubekommen bei einem Public-Key-Kryptosystem für praktische Anwendungen sehr nachteilhaft. Zudem sind Nachrichten meistens nicht zufällig. Das heißt ein Angreifer, der den Kontext der Nachricht kennt, kann womöglich damit wichtige Teilinformation aus der Nachricht herausbekommen. Falls die Nachricht z. B. „Ich möchte X Äpfel kaufen“ ist und der Angreifer dies weiß, kann er sehr einfach den Wert für X herausbekommen. Auch sollte nicht mehr als einmal dieselbe Nachricht versendet werden, wenn das Public-Key-Kryptosystem deterministisch verschlüsselt (das Verschlüsseln eines bestimmten Klartextes mit demselben öffentlichen Schlüssel führt immer zu demselben Geheimtext). Schon die Information, dass eine Nachricht zweimal gesendet wurde, was bei deterministischen Kryptosystem am versenden des gleichen Geheimtextes bemerkt werden kann, selbst ohne diesen zu entschlüsseln, kann für Angreifer wichtig sein. In der Tat werden für die Praxistauglichkeit eines Public-Key-Kryptosystems schon die kleinsten Teilinformationen, die ein Angreifer herausbekommen kann, als fatale Sicherheitslücken angesehen. Sichere Kryptosysteme dürfen also nicht deterministisch verschlüsseln, sondern benötigen Zufall in irgendeiner Art [3].

Wegen dieser Probleme und der Tatsache, dass Private-Key-Kryptosysteme wesentlich weniger Rechenaufwand brauchen, werden in der Praxis häufig

Hybride Kryptosysteme eingesetzt. Bei hybriden Kryptosystemen wird mit Hilfe eines Public-Key-Kryptosystems der geheime Schlüssel für ein Private-Key-Kryptosystem ausgetauscht, mit dem dann die eigentliche Nachricht verschlüsselt wird.

Eines der zentralen Probleme – nicht nur von Public-Key-Kryptosystemen, sondern Kryptographie im Allgemeinen – ist zudem, dass immer noch alle kryptographischen Primitive auf unbewiesenen Annahmen basieren. Wir können uns also nicht sicher sein, ob die Kryptosysteme, die wir als sicher annehmen, dies überhaupt sind. Im folgenden Kapitel will ich mich ein wenig mit diesem Problem befassen.

5 Impagliazzo's Welten

Der US-amerikanische Informatiker Russell Impagliazzo veröffentlichte 1995 eine Arbeit [17], in der er fünf verschiedene mögliche Welten vorstellt und deren Auswirkungen auf Kryptographie und andere Bereiche der Informatik untersucht. In diesen Welten wird davon ausgegangen, dass bestimmte komplexitätstheoretische Annahmen (wie z. B. $\mathrm{P} = \mathrm{NP}$) stimmen und deren Auswirkungen für die Informatik diskutiert. Wir gucken uns also mögliche Auswirkungen auf unser Leben an, für verschiedene Fälle die eintreten können, wenn wir die Kryptographie nicht mehr nur auf Annahmen stützen müssen. Mittlerweile wurde noch eine sechste mögliche Welt, die Komplexitätstheoretiker vorgeschlagen haben, ergänzt, die ich auch kurz beschreiben möchte. Hierbei ist [17] die Quelle für die ersten 5 Welten.

5.1 Algorithmica

Algorithmica ist die erste von Impagliazzo's Welten. In dieser ist $\mathrm{P} = \mathrm{NP}$. Eine solche Welt wäre ein algorithmisches Paradies. Man könnte alle NP-Schweren Probleme in Polynomialzeit lösen. Probleme, an denen schon viele Menschen gescheitert sind, wären leicht lösbar. Es wären revolutionäre Fortschritte in Bereichen der Mathematik, Programmieren oder auch künstliche Intelligenz möglich. Man könnte einem Computer mit Hilfe eines induktiven Lernalgorithmus so gut wie alles beibringen, was auch Menschen können. Man müsste dem Algorithmus nur eine genügend große Anzahl an Trainingsets von möglichen Ein- und Ausgaben des Problems geben und er würde den einfachsten Algorithmus finden, um dies zu reproduzieren.

In der Welt Algorithmica wäre allerdings Kryptographie, wie sie im Moment benutzt wird, im Grunde unmöglich. Jeder Code, der zum Verschlüsseln einer Information entwickelt werden würde, könnte genauso einfach auch gebrochen werden. Wir können es jedoch nicht ausschließen, dass Kryptosysteme wie das One-Time-Pad oder ganz grundlegend andere Kryptosysteme, in einer solchen Welt praktikabel wären. Es könnte auch keinerlei Identifizierung über Computer stattfinden, da diese leicht das Verhalten anderer kopieren und sich als diese ausgeben könnten.

Um zu zeigen, dass unsere Welt Algorithmica ist, könnte man z. B. einen effizienten Algorithmus für ein NP-Vollständiges Problem aufzeigen.

5.2 Heuristica

Heuristica ist die Welt in der NP-Probleme im „worst case“ schwer, im „average case“ jedoch leicht zu lösen sind. In gewisser Weise ist diese Welt paradox, da schwere Instanzen von NP-Problemen zwar existieren, diese zu finden, ist allerdings selbst ein schweres Problem. Diese Welt ist in algorithmischer Hinsicht fast identisch zu Algorithmica, jedoch mit kleinen Unterschieden. Falls $P = NP$ ist, dann ist auch die polynomielle Hierarchie (PH) in P. Doch wenn alle Probleme in NP im „average case“ leicht zu lösen sind, heißt das noch nicht, dass das selbe für alle Probleme in der polynomiellen Hierarchie gilt. Es ist durchaus denkbar, dass dies nicht für alle Probleme in PH gilt, doch eine klare Antwort auf diese Frage gibt es noch nicht.

Im Bezug auf Kryptographie gibt es auch wenig Unterschiede gegenüber Algorithmica. In Heuristica ist Kryptographie, wie wir sie kennen, ebenfalls im Grunde unmöglich. Kommunikationspartner müssten hohen Aufwand aufbringen, um schwere Probleme zu finden, welche ein Angreifer dann in ähnlich hohem Aufwand knacken könnte und man sollte für ein sicheres System immer davon ausgehen, dass ein Angreifer mehr Ressourcen aufbringt, als der Nutzer.

Um zu zeigen, dass unsere Welt Heuristica ist, könnte man eine Methode finden, mit der fast alle Instanzen eines NP-vollständigen Problems leicht gelöst werden. In dem Fall müsste zusätzlich gezeigt werden, dass eine untere Schranke für die „worst case“ Komplexität eines NP-Schweren Problems existiert.

5.3 Pessiland

Pessiland ist aus Impagliazzo's Sicht die schlechteste Welt für uns aus den möglichen Welten. In dieser Welt gibt es NP-Probleme, die im „average case“ schwer zu lösen sind, es existieren jedoch keine Einwegfunktionen. Es wäre also leicht, schwere Instanzen eines Problems zu finden, allerdings wäre es auch schwer eine Lösung für ein solches Problem zu finden. Es gäbe keine Möglichkeit eine Funktion, welche leicht zu berechnen und schwer zu invertieren ist, daraus zu erzeugen. In algorithmischer Hinsicht würde Pessiland unserer Welt nicht viele bekannte Vorteile bieten. Die meisten Probleme wären immer noch schwer lösbar und Fortschritt würde langsam durch vollständigeres Verständnis der Welt (Pessiland) geschehen, so wie es im Moment auch in unserer Welt geschieht. Durch die Nichtexistenz von Ein-

wegfunktionen wären allerdings kleinere Fortschritte in Gebieten wie dem unüberwachten Lernen (englisch: unsupervised learning) oder der Datenkompression möglich. Es ist durchaus denkbar, dass noch wesentlich mehr Fortschritt in Pessiland möglich ist, da es eine noch relativ unerforschte Welt ist.

Kryptographie, wie wir sie kennen, wäre auch in dieser Welt nicht möglich. Die Kenntnis eines schweren Problems, ohne Kenntnis dessen Lösung, würde einem in diesem Bereich nicht weiterhelfen. Doch auch im Bereich der Kryptographie ist es nicht auszuschließen (sogar wahrscheinlicher als bei Algorithmica oder Heuristica, da diese stärkere Einschränkungen haben), dass neue Möglichkeiten gefunden werden können.

Um zu zeigen, dass unsere Welt Pessiland ist, könnte man zeigen, dass eine untere Schranke für die „average case “ Komplexität eines NP-Problems existiert. Zudem müsste man zeigen, dass keine Einwegfunktionen existieren, z. B. indem man einen effizienten Algorithmus findet, der Levin's universelle Einwegfunktion invertiert. Diese universelle Einwegfunktion ist eine Funktion, die eine Einwegfunktion ist, genau dann, wenn Einwegfunktionen existieren (für Genaueres über diese siehe [18]).

5.4 Minicrypt

In der Welt Minicrypt existieren Einwegfunktionen, allerdings ist Public-Key-Kryptographie unmöglich. Hierbei wird mit Public-Key-Kryptographie nicht nur das Benutzen eines Public-Key-Kryptosystems zum Versenden einer Nachricht gemeint, sondern auch jeglicher geheime Schlüsselaustausch über einen öffentlichen Kanal. Es ist also der Vorgang, sich mit einem Fremden über einen öffentlichen, nicht sicheren Kanal auf ein gemeinsames Geheimnis zu einigen, nicht möglich.

In Minicrypt hätten wir also all die Möglichkeiten, die Einwegfunktionen mit sich bringen, wie z. B. Pseudozufallsgeneratoren oder digitale Signaturen. Ansonsten gäbe es aus algorithmischer Hinsicht jedoch nur minimale bekannte Vorteile, wie die Möglichkeit das Faktorisierungsproblem effizient zu lösen.

Für die Kryptographie würde es jedoch bedeuten, dass Private-Key-Kryptographie möglich ist. Es würde sogar bedeuten, dass es Private-Key-Kryptosysteme gibt, die gegen Chosen Chipertext und Chosen Plaintext Angriffe sicher sind. Wir wären jedoch wieder bei dem grundsätzlichen Problem, dass der Schlüsselaustausch über einen sicheren Kanal geschehen muss, was praktische Anwendungen stark einschränken würde.

Um zu zeigen, dass unsere Welt Minicrypt ist, müsste man beweisen, dass Einwegfunktionen existieren, z. B. indem man für eine Funktion zeigt, dass es keinen effizienten Algorithmus zum invertieren von dieser Funktion gibt. Außerdem müsste man zeigen, dass jedes Public-Key-Kryptosystem und sicherer Schlüsselaustausch über einen öffentlichen Kanal gebrochen werden kann. Für diesen zweiten Beweis ist allerdings noch nicht genau klar, wie man diesen überhaupt bewerkstelligen kann.

5.5 Cryptomania

Cryptomania ist die Welt, in der Public-Key-Kryptographie, wie wir sie kennen, existiert. In dieser Welt ist auch die LWE Annahme wahr. Auch in dieser Welt existieren Einwegfunktionen. Die Vorteile dieser gibt es hier also ebenfalls. Zusätzlich wären Vorgänge, wie z. B. sichere online Wahlen in Cryptomania möglich. Es gäbe aber aus algorithmischer Hinsicht noch weniger Vorteile als in Minicrypt.

Für Kryptographie wäre diese Welt jedoch fast ein Paradies. Es würde nicht nur Private-Key-Kryptosysteme, sondern auch Public-Key-Kryptosysteme geben, die gegen Chosen Ciphertext und Chosen Plaintext Angriffe sicher sind. Man könnte also einfach und sicher mit fremden Menschen über das Internet kommunizieren. Diese sichere Kommunikation wäre nicht mehr von der Lösung technischer Probleme abhängig, wie in den vorherigen Welten.

Um zu zeigen, dass unsere Welt Crptomania ist, könnte man beweisen, dass eine Annahme für Public-Key-Kryptographie, wie z. B. die Faktorisierungsannahme oder die LWE Annahme, stimmt. Man könnte ebenfalls zeigen, dass es eine sichere Methode (die auf keinen unbewiesenen Annahmen beruht) für einen Schlüsselaustausch über einen öffentlichen Kanal gibt.

5.6 Obfustopia

Obfustopia ist die von Komplexitätstheoretikern vorgeschlagene sechste Welt [4]. Diese ist ähnlich Cryptomania, auch hier existiert Public-Key-Kryptographie. Es ist jedoch zusätzlich „Indistinguishability Obfuscation“, auch IO genannt, möglich. Indistinguishability Obfuscation ist ein hochinteressantes, abstraktes kryptographisches Primitiv, worüber gerade ähnlich wie bei LWE viel geforscht wird. Der Grundgedanke dabei ist, dass das Geheimnis (der geheime Schlüssel bei Public-Key-Kryptosystemen) durch die

Unverständlichkeit des Programmcodes nicht herausfindbar ist. Dies ist ein grundsätzlich verschiedener Ansatz gegenüber aktuell verwendeten kryptographischen Primitiven und würde auch dem Kerckhoffs'schen Prinzip wiedersprechen, weil man versucht das Geheimnis zu verstecken. Falls Sie mehr über IO lesen wollen, siehe [19], [20], [21]. Die meisten aktuellen Kandidaten für IO könnten gebrochen werden, wenn LWE gebrochen werden kann [4], weshalb Obsfustopia als eine Art Unterwelt von Cryptomania angesehen wird. Es ist allerdings noch längst nicht klar, ob IO überhaupt einen algorithmischen Ansatz benötigt, oder ob es nicht auf einem aktuell unbekannten Weg auch anders möglich sein könnte, oder ob es überhaupt möglich ist.

In Obfustopia, wären zahlreiche weitere kryptographische Fortschritte durch IO möglich, zusätzlich zu den Möglichkeiten in Cryptomania. Um zu zeigen, dass unsere Welt Obfustopia ist, müsste man eine sichere Methode für Indistinguishability Obfuscation aufzeigen.

5.7 Was ist unsere Welt?

In was für einer Welt wir uns befinden, hat Impagliazzo als eine wichtige Aufgabe für Komplexitätstheoretiker definiert. Es könnte uns sowohl in algorithmischer, wie auch kryptographischer Hinsicht Fortschritte bringen und würde Klarheit über einige der größten bekannten Probleme in diesen Bereichen verschaffen. Allerdings müssten diese Probleme vermutlich erst einmal gelöst werden, um herauszufinden, in welcher Welt wir uns befinden. Von den vorgestellten Welten, wäre Pessiland wahrscheinlich die für uns ungünstigste Welt. In dieser wären weder große algorithmische Fortschritte möglich, wie in Algorithmica oder Heuristica, noch könnte man Kryptographie wie wir sie kennen betreiben. Auch, wenn in Minicrypt vermutlich keine Public-Key-Kryptographie durchführbar ist, wäre die Existenz von Private-Key-Kryptographie schon ein enormer Vorteil gegenüber Pessiland.

Unsere Welt ist im Moment Cryptomania am nähesten [4]. Wir nehmen aktuell an, dass Einwegfunktionen existieren, was schon die ersten drei Welten ausschließen würde. Im Gegensatz zu der Annahme, dass Einwegfunktionen existieren, könnte man an der Annahme, dass sichere Public-Key-Krypographie existiert, noch etwas mehr Zweifel haben. Eine subjektive Aussage darüber zu machen ist hier nicht so leicht wie bei Einwegfunktionen, die einfach in unser aktuelles Weltbild passen würden und deren Idee schon wesentlich mehr erforscht wurde. Die Public-Key-Kryptographie ist noch neuer und unerforschter und der überzeugendste Grund zur Annahme ihrer Existenz ist, dass es immer noch ungebrochene Public-Key-Kryptosysteme gibt. Aber auch von älteren Kryptosystemen, die mit der Zeit gebrochen

wurden, hat man früher einmal angenommen, dass sie sicher sind. Auch die Tatsache, dass es mit einem Quantencomputer möglich ist, viel erforschte Probleme wie das Faktorisierungsproblem oder den Diskreten Algorithmus und damit die meisten aktuell benutzten Kryptosysteme zu brechen, lässt einen etwas skeptisch werden. Doch solange es noch über längere Zeit ungebrochene Public-Key-Kryptosysteme gibt, wäre es „natürlich“ anzunehmen, dass auch Public-Key-Kryptographie existiert. Wir wären mit unseren aktuellen komplexitätstheoretischen Überzeugungen also in der Welt Cryptomania, eventuell sogar Obfustopia, in der sowohl Private-Key-Kryptographie als auch Public-Key-Kryptographie möglich sind.

6 Schluss

Der Zusammenhang zwischen Komplexitätstheorie und moderner Kryptographie ist nicht zu übersehen. Kryptographie basiert nicht nur auf komplexitätstheoretischen Konstrukten wie Einwegfunktionen, sondern es werden auch komplexitätstheoretische Annahmen und Reduktionen benutzt, um über die Sicherheit von Kryptosystemen zu diskutieren.

Public-Key-Kryptographie spielt dabei eine zentrale Rolle in der modernen Kryptographie, weil sie die praktische Zugänglichkeit von Kryptosystemen, im Gegensatz zu Private-Key-Kryptosystemen, revolutioniert hat. Hierbei sind der Großteil der aktuell benutzten Public-Key-Kryptosysteme jedoch aus der algebraischen Familie und diese sind zumindest gegen Quantencomputer nicht sicher. Die Kryptosysteme in der geometrischen Familie sind hier eine unserer besten aktuellen Hoffnungen, sichere Kryptosysteme zu finden, falls genügend große Quantencomputer in der Zukunft erreichbar sind und deshalb ein großes Forschungsthema in der Kryptographie. Man könnte deswegen umgangssprachlich auch von alten (algebraische Familie) und neuen (geometrische Familie) Kryptosystemen sprechen, da solche Quantencomputer schon in einer nicht allzu entfernten Zukunft realisiert sein könnten.

Wir können jedoch noch nicht mit Gewissheit sagen, dass sichere Public-Key-Kryptographie und auch Private-Key-Kryptographie wie wir sie kennen überhaupt möglich sind. Alle aktuellen praktischen Anwendungen von Kryptosystemen basieren auf bisher unbewiesenen Annahmen. Es wäre sehr wünschenswert, herauszufinden, in was für einer Welt wir uns befinden und Klarheit über die Existenz von Kryptosystemen zu bekommen. Auch wenn schon viele Wissenschaftler an Beweisen gescheitert sind, besteht immernoch die Hoffnung, dass wir irgendwann doch Gewissheit über diese Annahmen haben. Gerade wenn man sich den rapiden technischen Fortschritt in den letzten Jahren anguckt, ist es gar nicht so abwägig, dass auch im Bereich der Kryptographie revolutionäre Fortschritte realistisch sind. Das ganze Thema der Kryptographie ist zwar in dem Sinne wie eine offene Frage, auf die wir die Antwort einfach nicht wissen, doch startet jede Frage über etwas Unbekanntes erst einmal mit der Antwort „ich weiß es nicht“. Wir müssen nur genug suchen und werden hoffentlich eine Antwort finden.

Literatur

[1] D. Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet.* Scribner, 1996.

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach.* Cambridge University Press, 2009.

[3] John M. Talbot and Dominic J. A. Welsh. *Complexity and cryptography - an introduction.* Cambridge University Press, 2006.

[4] Boaz Barak. The complexity of public-key cryptography. *IACR Cryptology ePrint Archive*, 2017:365, 2017.

[5] Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques.* Cambridge University Press, 2001.

[6] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Cambridge, MA, USA, 1979.

[7] Nimrod Megiddo and Christos H Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.

[8] Paul W. Goldberg and Christos H. Papadimitriou. Towards a unified complexity theory of total functions. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:56, 2017.

[9] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[10] Chris Peikert. A decade of lattice cryptography. *IACR Cryptology ePrint Archive*, 2015:939, 2015.

[11] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

[12] Oded Regev. The learning with errors problem (invited survey). In *IEEE Conference on Computational Complexity*, pages 191–204. IEEE Computer Society, 2010.

[13] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342. ACM, 2009.

[14] Chris Peikert. How (not) to instantiate ring-lwe. *IACR Cryptology ePrint Archive*, 2016:351, 2016.

[15] Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. *IACR Cryptology ePrint Archive*, 2014:599, 2014.

[16] Douglas R. Stinson. *Cryptography - theory and practice*. Discrete mathematics and its applications series. CRC Press, 1995.

[17] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 134–147. IEEE Computer Society, 1995.

[18] Leonid A. Levin. The tale of one-way functions. *CoRR*, cs.CR/0012023, 2000.

[19] Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the sub-exponential barrier in obfustopia. In *EUROCRYPT (3)*, volume 10212 of *Lecture Notes in Computer Science*, pages 156–181, 2017.

[20] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *IACR Cryptology ePrint Archive*, 2013:451, 2013.

[21] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001.