UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Fakultät für Elektrotechnik, Informatik und Mathematik

Bachelorarbeit

Untersuchungen zur Sicherheit einzelner RSA-Bits

Andreas Cord-Landwehr
E-Mail: phoenixx@uni-paderborn.de

Paderborn, den 25. September 2007

vorgelegt bei
Prof. Dr. Johannes Blömer

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Paderborn, den 25. September 2007

Andreas Cord-Landwehr

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Einleitung . . . 1
1.2 Aufbau . . . 2

2 Einfache Techniken **3**
2.1 Das RSA-Verfahren . . . 3
2.1.1 Idee eines Public-Key-Kryptosystemes . . . 3
2.1.2 Der RSA-Algorithmus . . . 4
2.1.3 Sicherheit von RSA . . . 6
2.2 Werkzeuge . . . 6
2.2.1 Begriffe . . . 6
2.2.2 Wahrscheinlichkeitsanalysen . . . 7
2.2.3 Sicher . . . 9
2.3 CMHNP - ein schweres Problem? . . . 9
2.4 Sicherheit der Randbits . . . 10
2.4.1 Das Verfahren von Alexi et al. . . . 10
2.4.2 Grenzen dieses Ansatzes . . . 12
2.4.3 Ausblick . . . 13

3 Sicherheit der inneren Bits **15**
3.1 Test-Techniken . . . 16
3.1.1 Erzeugung von Zufallswerten . . . 16
3.1.2 Unterscheider über Intervallen . . . 19
3.2 Verfahren von Fischlin und Schnorr . . . 21
3.3 Der Näslund-Ansatz . . . 24
3.4 Kombination beider Ansätze . . . 25
3.4.1 Struktur der Näslund-Intervalle . . . 27
3.4.2 Der Invertierungsalgorithmus . . . 29
3.5 Existenz guter Rechtecke . . . 31
3.5.1 Der Fall v gerade . . . 33
3.5.2 Der Fall v ungerade . . . 35
3.6 Gesamtergebnis . . . 43

4 Untersuchungen zu zuverlässigen Orakeln **45**
4.1 Zuverlässige Bestimmung des untersten Bits . . . 45
4.2 Vereinfachungen beim Håstad- und Näslund-Ansatz . . . 47
4.3 Fazit . . . 48

1 Einleitung

1.1 Einleitung

Kryptographie ist heutzutage aus dem täglichen Leben nicht mehr wegzudenken. Allein wer beim Interneteinkauf eine Bestellung abschickt, wird dafür mehrfach kryptographische Hilfsmittel verwenden – wissentlich oder auch unwissentlich.

Bei einer solchen Kommunikation werden teils hochsensible Daten zwischen zwei Computern hin- und hergeschickt. Meistens werden dafür aber sehr unsichere Medien, wie z.B. das Internet verwendet. So wäre es für einen Angreifer ein Leichtes, im Internet alle übertragenen Daten abzuhorchen. Aus diesem Grund müssen Daten so übertragen werden, dass sie allein vom Empfänger verstanden werden können. Daher sind diese Daten im besten Falle verschlüsselt und nur der Empfänger besitzt den richtigen Schlüssel um sie lesen zu können. Ein Angreifer sollte natürlich auch nicht in der Lage sein, diesen Schlüssel oder den Klartext zu berechnen, egal wie lange er den Datenverkehr abhört.

Aber stimmt dieses tatsächlich? Kann ein Angreifer wirklich nichts mit den abgefangenen Daten anfangen? – In dieser Arbeit wird dazu das RSA-Verfahren untersucht. Das RSA Verfahren, benannt nach seinen Entwicklern Rivest, Shamir und Adleman, wurde 1978 in [RSA78] publiziert. Seitdem hat es sich zu einem der am häufigsten verwendeten Kryptosysteme weltweit entwickelt. Seit nunmehr 30 Jahren widersteht es unzähligen Angriffen, die ständig nach Schwachstellen in diesem Verfahren suchen. Bis heute beruhen aber die einzigen bekannten Angriffstechniken lediglich auf einer schlechten Wahl bestimmter Parameter.

Allerdings konnte auch noch kein Beweis zur Sicherheit dieses Verfahrens angegeben werden. Zahlreiche Argumente sprechen zwar für diese Sicherheit[1], es fehlt jedoch der Beweis. Dennoch lautet die vorherrschende Fachmeinung, dass es sich bei RSA um ein sicheres Verfahren handelt.

Wenn man nun annimmt, dass RSA sicher ist, so kann dennoch eine ähnliche Fragestellung untersucht werden. Die Frage ist nun, ob Teile des Klartextes bestimmt werden können, ohne den Aufwand einer Bestimmung des vollständigen Klartextes zu verursachen. Eine Bestimmung solcher ausgewählter Teile, würde dem Angreifer beim Beispiel der Onlinebestellung bereits genügen. Wenn dieser nur Kontonummer und Adresse des Käufers herausfinden könnte, hätte der Angreifer bereits Erfolg gehabt.

Die ersten Untersuchungen zu dieser Thematik wurden bereits Anfang der 80er in [GMT82] vorgenommen. Dort konnte aber nur gezeigt werden, dass das niederwertigste Bit nicht zuverlässig bestimmt werden kann. Nach einigen weiteren Einzelergebnissen gelang es schließlich Alexi, Chor, Goldreich und Schnorr in [ACGS88] einen Sicherheitsbeweis aller niederwertigsten und höchstwertigsten Bits anzugeben. Veröffentlicht wurde dieser Beweis 1988. Zu einem vollständigen Beweis, der die Sicherheit aller RSA-Bits zeigt, dauerte es aber bis Håstad und Näslund 1998 in [HN98] ihr Ergebnisse veröffentlichten.

Den Kern dieser Bachelorarbeit bildet nun der Beweis von Håstad und Näslund. In den folgenden Kapiteln werden die dazu notwendigen Techniken erläutert. Dabei folgt eine ausführliche

[1] Sicher ist in dem Sinne gemeint, dass die Berechnung des Klartextes oder des geheimen Schlüssels nicht in annehmbarer Zeit erfolgen kann.

und genaue Darstellung des Beweises.

Einen weiteren Untersuchungsgegenstand wird die Frage bilden, ob und in welchem Maße bei der Verwendung eines zuverlässigen Orakels dieser Beweis von Håstad und Näslund vereinfacht werden kann. Bekannt ist, dass sich bei der Untersuchung des untersten Klartextbits eine drastische Vereinfachung des Beweises ergibt, wenn ein zuverlässiges Orakel vorausgesetzt wird. Ergebnis wird bei den inneren Bits jedoch sein, dass eine wesentliche Vereinfachung beim betrachteten Beweis von Håstad und Näslund nicht möglich ist.

1.2 Aufbau

Der Ablauf dieser Untersuchungen wird dabei folgendermaßen sein. Zuerst werden in Kapitel 2 die Grundlagen für die folgenden Analysen gelegt. Das umfasst eine Erklärung des RSA-Verfahrens, Techniken für Wahrscheinlichkeitsanalysen und die formale Definition eines Orakels für die einzelnen Bits. Zudem wird das Verfahren von Alexi, Chor, Goldreich und Schnorr aus [ACGS88] vorgestellt, welches die Sicherheit der niederwertigsten und höchstwertigsten Randbits zeigt.

In Kapitel 3 erfolgt dann die Analyse der inneren Bits. Dazu werden Arbeiten von Fischlin und Schnorr ([FS97]) sowie von Näslund ([Nä96]) vorgestellt. Daran schließt die Analyse eines kombinierten Verfahrens aus diesen beiden Techniken an. Dieses kombinierte Verfahren wurde von Håstad und Näslund in [HN04] vorgestellt und liefert einen Sicherheitsbeweis für alle inneren Bits.

Zum Schluss wird in Kapitel 4 die Fragestellung untersucht, ob sich der in [HN04] vorgestellte Beweis wesentlich vereinfachen ließe, wenn andere Annahmen genommen würden. Als stärkere Forderung wird dabei das Problem der zuverlässigen Bestimmung eines Bits betrachtet. Bei den Untersuchungen zur Sicherheit des niederwertigsten Bits konnten erhebliche Vereinfachungen des Beweises festgestellt werden. In dem hier betrachteten Fall gestaltet sich dieses aber als deutlich aufwändiger. Es wird sich dabei herausstellen, dass bei diesem Verfahren kaum eine Vereinfachung möglich ist.

2 Einfache Techniken

In diesem Kapitel werden die Grundlagen für die folgenden Untersuchungen gegeben. Dieses umfasst Begriffe und Werkzeuge für die folgenden Analysen, aber auch eine erste einfache Analysetechnik zur Sicherheit der äußeren Bits.

Begonnen wird mit einer Vorstellung des RSA-Verfahrens. Dabei wird auf die Idee eines Public-Key-Kryptosystemes eingegangen, eine Vorstellung des RSA-Algorithmus vorgenommen sowie die Sicherheit dieses Verfahrens diskutiert.

Es folgt eine Klärung der später verwendeten Begriffe sowie eine Einführung der Wahrscheinlichkeitstechniken, die für die Untersuchungen verwendet werden.

Den Abschluss dieses Kapitels bildet die Vorstellung eines Verfahrens von Alexi, Chor, Goldreich und Schnorr [ACGS88], mit dem die Sicherheit der niederwertigsten und höchstwertigsten Bits gezeigt werden kann.

2.1 Das RSA-Verfahren

In dieser Arbeit wird das RSA-Verfahren untersucht. Bei RSA handelt es sich um eines der heutzutage am häufigsten verwendeten *Public-Key-Kryptosysteme*. Nach seiner Entwicklung in den 70ern konnte es weltweit eine große Verbreitung finden und besitzt auch heute noch erhebliche praktische Bedeutung.

Bevor eine genaue Definition des Verfahrens gegeben wird, soll zuvor die Idee eines Public-Key-Kryptosystemes vermittelt werden, denn durch seine spezielle Struktur unterstützt RSA diese Verfahren in einer natürlichen Weise. Dieses führte auch zu der großen Verbreitung von RSA. Auf der anderen Seite ist diese zugrunde liegende Struktur von RSA gerade einer der Hauptangriffspunkte.

2.1.1 Idee eines Public-Key-Kryptosystemes

Betrachten wir noch einmal das Problem des Interneteinkaufes. Bei einem solchen Einkauf sollen sämtliche Daten sicher übertragen werden. Da das Internet aber keine sichere Verbindung darstellt, müssen kryptographische Verfahren verwendet werden, um diese Sicherheit zu gewährleisten. Damit eine verschlüsselte Kommunikation zwischen zwei Partnern gestartet werden kann, ist es notwendig, dass sich diese Partner über die verwendeten Schlüssel einig werden. Jeder, der den richtigen Schlüssel zum Entschlüsseln kennt, wird die Datenübertragung mitlesen können. Ein sicherer Austausch dieser Schlüssel ist daher von größter Bedeutung.

In einer Laborsituation könnten sich beide Kommunikationspartner natürlich vor der Kommunikation persönlich treffen und gemeinsame Schlüssel vereinbaren. Bei einem Einkauf in einem beliebigen Internetversandhaus ist dieses aber wesentlich komplizierter, denn ein Kunde möchte nicht erst persönlich die Geschäftsstelle aufsuchen, bevor er den Einkauf beginnt.

Dieses Dilemma wurde erst in den 70er Jahren gelöst. Eines der ersten dazu verwendeten Verfahren war RSA. Der *einfache* Trick bei diesem Verfahren ist es, dass unterschiedliche Schlüssel zur Verschlüsselung und zur Entschlüsselung verwendet werden. Dabei ist es natürlich auch nicht möglich, aus dem Verschlüsselungsschlüssel den Entschlüsselungsschlüssel zu berechnen.

Wegen dieses Schlüsselpaares aus zwei unterschiedlichen Schlüsseln wird häufig auch von *asymmetrischer Verschlüsselung* gesprochen.

Bei solch einem Verfahren kann daher der Verschlüsselungsschlüssel öffentlich bekannt gegeben werden. Der Schlüssel zur Entschlüsselung bleibt aber geheim beim Empfänger. Deshalb wird auch der Begriff *Public-Key-Kryptosystem* verwendet.

Bei der Definition eines asymmetrischen Verfahrens wird immer ein mathematisches Problem benötigt, das in einer Richtung einfach, in der anderen Richtung aber sehr ineffizient zu berechnen ist. Bei RSA ist dieses gerade die Multiplikation zweier großer Primzahlen. Diese ist schnell zu berechnen. Eine Faktorisierung dieses Produktes benötigt aber gleich exponentielle Zeit und kann damit nicht in annehmbarer Zeit berechnet werden.

Durch diese zugrunde liegenden mathematischen Strukturen bieten sich aber immer wieder Angriffspunkte auf diese Verfahren. Einige dieser Eigenschaften, wie z.B. die Multiplikativität von RSA, werden später noch verwendet, um Angriffe zu konstruieren.

2.1.2 Der RSA-Algorithmus

Bevor mit dem RSA-Verfahren ein Klartext verschlüsselt werden kann, muss ein Schlüsselaustausch durchgeführt werden. Ziel dieses Vorganges ist es, zwischen zwei Kommunikationspartnern zwei Schlüssel e und d so zu verteilen, dass einer zum Verschlüsseln und der andere zum Entschlüsseln verwendet werden kann. Hierbei ist wichtig, dass sich der geheime Schlüssel e zum Entschlüsseln nicht effizient aus dem öffentlichen Schlüssel d errechnen lässt. Dadurch wird sichergestellt, dass nur der Besitzer des geheimen Schlüssels, die mit dem zugehörigen Schlüssel verschlüsselten Nachrichten lesen kann.

Wir betrachten dazu folgendes Szenario, in dem zwei Personen Alice und Bob miteinander kommunizieren möchten. In diesem Fall möchte Bob eine Nachricht zu Alice schicken. Die Wahl der Schlüssel und der Vorgang des Ver- und Entschlüsselns geschieht folgendermaßen:

1. Alice wählt zwei große, verschiedene[a] Primzahlen p und q.
2. Alice berechnet $N := p \cdot q$ und $\varphi(N) := (p-1) \cdot (q-1)$.
3. Alice wählt $e \in \{2, \ldots, \varphi(N) - 1\}$ mit $\mathrm{ggT}(e, \varphi(N)) = 1$.
4. Alice bestimmt d mit $d \cdot e \equiv 1 \pmod{\varphi(N)}$.
5. Alice gibt den öffentlichen Schlüssel (N, e) bekannt und behält den geheimen Schlüssel (N, d).
6. Alice löscht die Werte $\varphi(N)$, p und q.

[a] Diese Primzahlen werden mit gleicher Bitanzahl gewählt.

Bob möchte nun eine verschlüsselte Nachricht $x \in \{0, \ldots, N-1\}$ zu Alice schicken. Bob und Alice müssen dazu folgende Schritte durchführen.

7. Bob berechnet $y := E_N(x) = x^e \bmod N$ und sendet y zu Alice.
8. Alice entschlüsselt die Nachricht durch Berechnen von $x = D_N(y) = y^d \bmod N$.

Nach Definition besteht der RSA-Algorithmus aus den zwei Funktionen E_N und D_N. Dabei sind die Funktionen als $E_N : \mathbb{Z}_N \longrightarrow \mathbb{Z}_N, x \longmapsto x^e \bmod N$ und $D_N : \mathbb{Z}_N \longrightarrow \mathbb{Z}_N, y \longmapsto$

$y^d \bmod N$ definiert. Damit hierbei von einem kryptographischen System gesprochen kann, muss E_N eine bijektive Funktion sein. Dadurch wird sichergestellt, dass die Entschlüsselung eines Chiffretextes tatsächlich die ursprüngliche Nachricht ergibt.

Satz 1. *Sei die Funktion $E_N : \mathbb{Z}_N \longrightarrow \mathbb{Z}_N, x \longmapsto x^e \bmod N$ gegeben mit $N := p \cdot q$, p und q prim. e sei passend gewählt, so dass für ein $d \in \{2, \ldots, \varphi(N) - 1\}$ gilt $e \cdot d \equiv 1 \pmod{\varphi(N)}$ mit $\mathrm{ggT}(e, d) = 1$. Dann gilt Folgendes:*

1. *E_N ist bijektiv mit Umkehrfunktion D_N.*
2. *$E_N(1) = 1 = D_N(1)$.*
3. *$E_N(x) \cdot E_N(y) = E_N(x \cdot y)$, mit $x, y \in \mathbb{Z}_N$.*

Beweis. 1. Sei $x \in \mathbb{Z}_N$. Aus $e \cdot d \equiv 1 \pmod{\varphi(N)}$ folgt, dass eine ganze Zahl k existiert mit $k \cdot \varphi(N) = e \cdot d - 1$. Es gilt für $D_N(E_N(x)) = D_N(x^e \bmod N) = x^{e \cdot d} \bmod N = x \cdot x^{e \cdot d - 1} \bmod N$. Dieses lässt sich umformen zu

$$x \cdot x^{e \cdot d - 1} \equiv x \cdot x^{k \cdot \varphi(N)} \equiv x \cdot x^{k \cdot (p-1) \cdot (q-1)} \pmod{N}$$

Mit $N = p \cdot q$ und $\mathrm{ggT}(p, q) = 1$ ergibt sich das folgende System linearer Kongruenzen:

$$\begin{aligned} x^{e \cdot d} &\equiv x \cdot x^{k(p-1)(q-1)} \pmod{p} \\ x^{e \cdot d} &\equiv x \cdot x^{k(p-1)(q-1)} \pmod{q} \end{aligned}$$

Eine Betrachtung der Kongruenzen in den jeweiligen Faktorgruppen $\mathbb{Z}_p$ und $\mathbb{Z}_q$ und Anwendung des kleinen Satzes von Fermat liefert ein vereinfachtes System.

$$\begin{aligned} x^{e \cdot d} &\equiv x \pmod{p} \\ x^{e \cdot d} &\equiv x \pmod{q} \end{aligned}$$

Mit dem Chinesischen Restsatz folgt schließlich $x^{e \cdot d} \equiv 1 \pmod{N}$.

2. Unmittelbar aus den Potenzgesetzen folgt $E_N(1) = 1^e = 1 = 1^d = E_N(1)$.
3. $E_N(x) \cdot E_N(y) = x^e \cdot y^e \bmod N = (x \cdot y)^e \bmod N = E_N(x \cdot y)$.

□

Eine Angriffsmöglichkeit auf das RSA-Verfahren ist das Finden eines Faktors von N bzw. die Primfaktorisierung von N. Sobald ein $x \in \mathbb{Z}_N \backslash (\mathbb{Z}_N^* \cup \{0\})$ verschlüsselt wird, ist es möglich einen solchen Faktor zu finden. Die Wahrscheinlichkeit für das Finden eines solchen Elementes ist jedoch verschwindend gering. Das folgende Lemma wird es nun erlauben, in sämtlichen weiteren Untersuchungen nur Elemente aus $\mathbb{Z}_N^*$ zu betrachten.

Lemma 2. *Sei $x \in \mathbb{Z}_N \backslash (\mathbb{Z}_N^* \cup \{0\})$, N, e, d seien geeignet gewählt. Es sei der öffentliche Schlüssel (N, e) sowie der Chiffretext $y = E_N(x)$ bekannt. Dann ist es möglich, zwei Zahlen p und q effizient zu berechnen, so dass gilt $N = p \cdot q$.*

Beweis. $\mathbb{Z}_N^*$ ist eine multiplikative Gruppe. Mit den Ergebnissen des vorherigen Satzes ist die Einschränkung von E_N auf $\mathbb{Z}_N^*$ ein Automorphismus von $\mathbb{Z}_N^*$ nach $\mathbb{Z}_N^*$. Damit ist E_N durch Einschränkung auf $\mathbb{Z}_N \backslash (\mathbb{Z}_N^* \cup \{0\})$ ebenfalls eine Bijektion und es gilt somit $E_N(x) \notin \mathbb{Z}_N^*$. Der größte gemeinsame Teiler von y und N ist damit ein nicht trivialer Teiler von N. Durch Division kann der zweite Teiler gefunden werden und N ist faktorisiert. □

2.1.3 Sicherheit von RSA

Falls die Primfaktorzerlegung des RSA-Modulus N bekannt sein sollte, kann durch den euklidischen Algorithmus der geheime Schlüssel berechnet werden. Eine Kernfrage zur Sicherheit des RSA-Algorithmus ist damit die Schwierigkeit der Primfaktorzerlegung des Modulus N.

Bis heute besitzt der beste bekannte Algorithmus zur Primfaktorzerlegung aber immer noch eine exponentielle Laufzeit von

$$e^{\mathcal{O}(\log^{1/3}(N)\cdot\log\log^{2/3}(N))}$$

und ist damit im allgemeinen nicht praktikabel zu verwenden. Es handelt sich dabei um eine Anwendung des Zahlkörpersiebes von Lenstra et al. [LJMP90].

Auf der anderen Seite existiert aber auch die Möglichkeit, dass die Lösung des RSA-Problems einfacher ist als das Problem der Primfaktorzerlegung. Immerhin konnten Boneh und Venkatesan in [BV98] zeigen, dass keine algebraische Reduktion von der Lösung des low-exponent-RSA[1] Problems auf das Primfaktorisierungsproblem existieren kann. Dieses ist ein Indiz in die Richtung, dass möglichweise das RSA-Problem einfacher als das Primfaktorisierungsproblem sein könnte. Ob dieses aber überhaupt der Fall ist und wenn, wie viel leichter das RSA-Problem dann zu lösen wäre, ist jedoch nicht bekannt.

Angriffe auf das RSA-Verfahren beschränken sich daher bis heute auf das Ausnutzen von schlecht gewählten Exponenten oder ungleich groß gewählten Primzahlen. Ein Brechen dieses Verfahrens bei normgerechter Implementierung, ist damit nicht möglich. Jedoch können ungeschickte Implementierungen hervorragende Angriffsziele auf das RSA-Verfahren darstellen. Daher ist gerade dort genau auf eine eine sorgfältige Implementierung zu achten.

2.2 Werkzeuge

In diesem Abschnitt sollen Werkzeuge und Begriffe bereitgestellt werden, die zur Analyse der folgenden Techniken notwendig sind. Dieses umfasst spezielle in dieser Arbeit verwendet Begriffe, ebenso wie Techniken aus der Wahrscheinlichkeitsanalyse.

2.2.1 Begriffe

Sei im Folgenden der RSA-Modulus N definiert als $N := p \cdot q$ mit p und q prim. Es bezeichne n die Anzahl der Bits der Binärdarstellung von N, also $n = \lceil \log_2(N) \rceil$. Für die folgenden Untersuchungen wird angenommen, dass diese Werte eingangs gewählt werden und im Weiteren konstant sind.

Mit der Bezeichnung $[x]_N$ für ein $x \in \mathbb{Z}$ ist die Restklasse $x + N \cdot \mathbb{Z}$ gemeint. Als Konvention wird diese immer durch ihren positiven ganzzahligen Repräsentanten $x \in \{0, \ldots, N-1\}$ angegeben.

Der Absolutwert einer Zahl $[a]_l$ ist definiert als $\mathrm{abs}_l(a) = \min\{[a]_l, l - [a]_l\}$. Wenn für ein $\delta \in [0, 1]$ der Absolutwert $\mathrm{abs}_l(a) \leq \delta l$ ist, heißt dieses *a δ-klein modulo l*.

Eine Zahl x heißt δ-bestimmt modulo N, wenn sie in der Form $a + b$ geschrieben werden kann, wobei a aber δ-klein und der Wert von b bekannt ist.

Um die abstrakten Begriffe ein wenig anschaulicher zu erläutern, werden diese Definitionen im folgenden Beispiel auf konkrete Zahlen angewendet.

[1] Der öffentliche Exponent e ist kleiner einer festen Konstante.

Beispiel 3. *Sei ein RSA-Modulus* $N = 17 \cdot 13$ *gegeben. Damit ist die Länge der binären Darstellung von* N *gleich* $\lceil \log_2 221 \rceil = 8$.

Das RSA-Verfahren arbeitet mit diesem Modulus auf der Zahlenmenge $[0]_{221}, \ldots, [220]_{221}$. *Hierbei stellt jedes* $[a]_{221}$ *eine Restklasse modulo* 221 *da. So bezeichnet z.B.* $[3]_{221}$ *auch die Elemente* $224 = 3 + 1 \cdot 221$ *und* $445 = 3 + 2 \cdot 221$, *aber auch* $-218 = 3 + (-1) \cdot 221$.

Für einen Wert 15 *ist der Absolutwert modulo* 221 *gleich* 15, *da* $15 = \min\{15, 221 - 15\}$. *Ebenso ist auch der Absolutwert von* 206 *modulo* 221 *vom Wert* $15 = \min\{205, 221 - 206\}$. *Der Absolutwert von* 15 *modulo* 17 *ist aber* 2.

Betrachten wir den Wert 2. *Dieser Wert ist* $0,1$*-klein modulo* 221, *wegen* $221 \cdot 0,1 > 2$. *Ebenso ist die Zahl* 219 $0,1$*-klein modulo* 221, *da hierbei der Absolutwert der Zahl betrachtet wird.*

Sei nun bekannt, dass c *eine Zahl gleich* $c = a + b$ *ist. Dabei sei außerdem bekannt, dass* $b = 118$ *gilt und* a *vom Wert* $0,1$*-klein ist. Folglich sind alle Zahlen* c *gleich* $117, 118$ *oder* 119 $0,1$*-bestimmt modulo* 221. *Somit ist nicht sicher, welchen dieser Werte* c *trägt. Bis auf einen kleinen Fehler kann* c *aber genau bestimmt werden.*

Ähnliche Untersuchungen werden in den folgenden Analysen oft durchgeführt. Dabei wird es sich aber um erheblich kompliziertere Sachverhalte handeln.

Da in dieser Arbeit vor allem die Sicherheit einzelner Bitpositionen in RSA-Texten betrachtet wird, muss zuerst geklärt werden, wie diese Bitzahlen dargestellt sind. Die hier verwendete Darstellung verwendet die Konvention, dass das niederwertigste Bit am weitesten rechts und das höchstwertigste Bit am weitesten links steht. Synonym für diese Begriffe werden auch *unterstes Bit* für das niederwertigste Bit sowie *oberstes Bit* für das höchstwertigste Bit verwendet. *Innere* Bits werden die Bits genannt, die nicht am Rand liegen. Das sind alle Bitpositionen, die bei einer n-Bit langen Zahl mindestens $\log(n)$ vom obersten und untersten Bit entfernt sind.

Als Bezeichnung für das i-te niederwertigste Bit in der Binärdarstellung von x, wird $\mathrm{bit}_i(x)$ verwendet. Falls dieses Bit nicht existiert, ist der Wert 0. Verallgemeinert bezeichnet $B_i^j(x)$ das Intervall der Bits von x von i bis j der Binärdarstellung von x. Also das Tupel $B_i^j(x) = (\mathrm{bit}_j(x), \mathrm{bit}_{j-1}(x), \ldots \mathrm{bit}_i(x))$. Das unterste Bit wird auch als $\mathrm{lsb}(x) = \mathrm{bit}_0(x)$ bezeichnet. Diese besondere Bezeichnung ist durch die Sonderrolle begründet, die dieses Bit besitzt. So wird durch das unterste Bit angegeben, ob eine Zahl gerade oder ungerade ist.

2.2.2 Wahrscheinlichkeitsanalysen

Für die folgenden Wahrscheinlichkeitsanalysen werden Orakel verwendet. Orakel sind probabilistische Algorithmen, die für die hier vorgestellten Untersuchung in der Laufzeit polynomiell beschränkt sind. Ein Orakel liefert mit einer Erfolgswahrscheinlichkeit größer $1/2$ ein korrektes Ergebnis. Ebenso existiert eine Gegenwahrscheinlichkeit, dass das Orakel falsch antwortet. Bildlich kann man sich ein Orakel als eine Person vorstellen, die Fragen beantwortet und dieses öfters richtig als falsch macht. Um dieses *öfters richtig als falsch* genauer zu fassen, wird der Begriff der Wahrscheinlichkeit und der Wahrscheinlichkeitsfunktion benötigt.

Als Konvention sei zuvor noch angegeben, dass $x \in_D \{0,1\}^k$ ($k \in \mathbb{N}$) das zufällige Wählen eines Elementes aus $\{0,1\}^k$ mit der Verteilung D bezeichnet. Dabei ist D eine Verteilung über einer Zahlenmenge $\{0,1\}^k$. Die Gleichverteilung wird dabei mit U bezeichnet.

Damit wird nun für die Wahrscheinlichkeitsfunktion die folgende Definition verwendet.

Definition 4 (Wahrscheinlichkeitsfunktion)**.** Seien $A_1, \ldots, A_k$ Mengen und $D_1, \ldots, D_k$ Wahrscheinlichkeitsverteilungen auf diesen Mengen. Für $i = 1 \ldots k$ seien die Werte x_i mit den Wahrscheinlichkeitsexperimenten $x_i \in_{D_i} A_i$ gewählt.

Dann bezeichnet für ein Ereignis $E = E(x_1, \ldots, x_k)$ die Funktion

$$Pr(x_1 \in_{D_1} A_1, \ldots, x_k \in_{D_k} A_k : E) \in [0, 1],$$

die Wahrscheinlichkeit für das Eintreten von E bei Durchführung der Wahrscheinlichkeitsexperimente $x_1 \in_{D_1} A_1, \ldots, x_k \in_{D_k} A_k$.

Jedes Orakel besitzt eine Erfolgswahrscheinlichkeit ϵ bzw. $\epsilon(k)$, wenn der Wert von einer Eingabegröße k abhängt. Diese Erfolgswahrscheinlichkeit gibt an, wie gut dieses Orakel bei einer zufälligen Eingabe, den Funktionswert einer anderen Funktion bestimmen kann. Wenn diese Erfolgswahrscheinlichkeit 1 beträgt, spricht man von einem zuverlässigen Orakel. Ein zuverlässiges Orakel ist damit eine deterministische Turingmaschine.

Da sämtliche Untersuchungen auf Zahlenbereichen $\{0, 1\}^k$ mit $k \in \mathbb{N}$ stattfinden sind die folgenden Definitionen spezieller formuliert, als sie in der Standardliteratur zu finden sind. Dieses vermeidet eine unnötige zusätzliche Notation und soll das Lesen erleichtern.

Definition 5 (Orakel)**.** Eine polynomiell beschränkte, nichtdeterministische Turingmaschine $\mathcal{P}$ heißt $\epsilon(k)$-Orakel[2] für eine polynomiell beschränkte Funktion $f : \{0, 1\}^k \longrightarrow \{0, 1\}^k$, wenn gilt

$$Pr\left(x \in_U \{0, 1\}^k : \mathcal{P}(x) = f(x)\right) \geq \frac{1 + \epsilon(k)}{2}.$$

Eine speziellere Form eines Orakels ist ein Unterscheider. Ziel eines Unterscheiders ist es, zwei Mengen bzw. zwei Verteilungen voneinander zu unterscheiden. Als Unterscheider wird hier ein deterministischer polynomieller Algorithmus bezeichnet, der nur 0 oder 1 ausgibt und dieses bei verschiedenen Verteilungen unterschiedlich oft. Der Unterscheider darf dabei aber auf ein nichtdeterministisches Orakel zugreifen.

Da das Verhalten eines Unterscheiders gerade durch die Häufigkeit der Antwort 1 beschrieben werden kann, wird im Folgenden vom Anteil der 1-Antworten gesprochen. Dieses ist gleichbedeutend mit der Wahrscheinlichkeit der Antwort 1 auf einer Verteilung.

Definition 6 (Unterscheider)**.** Seien D, D' Verteilungen über $\{0, 1\}^n$. $\mathcal{P}$ sei ein Orakel, welches befragt werden darf. Dann heißt ein deterministischer Algorithmus $A^{\mathcal{P}} : \{0, 1\}^n \longrightarrow \{0, 1\}$ ein $\zeta(n)$-Unterscheider, wenn gilt:

$$\left|Pr(x \in_D \{0, 1\}^n : A^{\mathcal{P}}(x) = 1) - Pr(x \in_{D'} \{0, 1\}^n : A^{\mathcal{P}}(x) = 1)\right| \geq \zeta(n).$$

Die Qualität eines Unterscheiders ergibt sich also dadurch, dass sich der Wert der 1-Antworten auf beiden Verteilungen deutlich unterscheidet.

Bei großen Zahlen, die vom Wert nahe bei N liegen, tritt der Effekt auf, dass die höheren Bits nahezu alle gleich sind. Betrachtet man beispielsweise eine n Bit lange Zahl. So ist das n-te Bit die ersten 2^{n-1} Zahlen gleich 0, die nächsten 2^{n-1} Zahlen gleich 1. Dieses kann die Erfolgswahrscheinlichkeit eines Unterscheiders erheblich beeinträchtigen, da der Unterscheider seinen Vorteil gerade aus der Bestimmung des i-ten Bits bezieht. Wenn aber bei beiden zu unterscheidenden Verteilungen nahezu gleich oft Werte mit gleichem i-te Bit vorhanden sind, wird die Unterscheidung wesentlich erschwert. Um diese Problematik präziser zu fassen, wird der Begriff der Verzerrung eingeführt.

Definition 7 (Verzerrung)**.** Mit der *Verzerrung* $\beta_i(N)$ (engl. Bias) eines i-ten Bits wird folgende Wahrscheinlichkeit beschrieben

$$\Pr\left(z \in_U \mathbb{Z}_N : \mathrm{bit}_i(z) = 0\right) = \frac{1 + \beta_i(N)}{2}.$$

[2] Dieser Begriff kann allgemeiner formuliert werden als in diesem Fall. Für die geforderten Zwecke ist diese Definition jedoch hinreichend.

2.2.3 Sicher

Die Ergebnisse dieser Arbeit geben Sicherheitsaussagen über einzelne RSA-Bits. Intuitiv ist der Begriff *sicher* klar. Beispielsweise geht man intuitiv davon aus, dass ein Zahlenschloss sicher ist, wenn man dieses nur mit einer sehr geringen Wahrscheinlichkeit durch „herumprobieren“ öffnen kann. Zu Erklären bleibt aber das Verständnis von *gering*. Für diese Größe wird der formale Begriff *vernachlässigbar* eingeführt.

Definition 8 (vernachlässigbar und signifikant)**.** Eine Funktion $g(n)$ heißt *vernachlässigbar*, wenn für alle $c > 0$ und für alle $n \in \mathbb{N}$ (genügend groß) gilt, $g(n) < n^{-c}$. Wenn eine Funktion nicht vernachlässigbar ist, heißt sie *signifikant*.

In den beiden folgenden Definitionen wird nun das intuitive Verständnis von Sicherheit formalisiert.

Definition 9 (sicher)**.** Eine polynomielle Funktion heißt *sicher*, wenn kein $\epsilon(n)$-Orakel mit signifikantem $\epsilon(n)$ für diese Funktion existiert. Eine nicht sichere Funktion heißt *unsicher*.

Definition 10 (ununterscheidbar und unterscheidbar)**.** Zwei Verteilungen D und D' heißen *ununterscheidbar*, wenn kein $\epsilon(n)$-Unterscheider mit signifikantem $\epsilon(n)$ für diese Verteilungen existiert. Ansonsten heißen sie *unterscheidbar*.

2.3 CMHNP - ein schweres Problem?

Das *Chosen Multiplier Hidden Number Problem* (CMHNP) ist eine Abwandlung des *Hidden Number Problems* (HNP), welches von Venkatesan und Boneh [BV96] eingeführt wurde. In seiner Urform beschreibt das HNP folgende Frage: Gegeben sei ein Orakel $\mathcal{P}$, welches bei Aufruf ein zufälliges $a \in \mathbb{Z}_N$ wählt und partielle Informationen über $[ax]_N$ sowie den Wert a zurück gibt. Aus diesen partiellen Informationen ist dann der Wert von x zu bestimmen.

Die Abwandlung im CMHNP ist, dass als zusätzliche Eingabe bereits der Multiplikator a gewählt werden kann. Damit gibt hier das Orakel partielle Informationen über $[ax]_N$ zurück, wobei die Wahl von a beim Angreifer liegt.

Problem 11 (CMHNP)**.** Formal stellt sich dieses Problem wie folgt dar:

Gegeben: Ein Orakel $\mathcal{P}$ sowie ein $a \in \mathbb{Z}_N$. Bei Eingabe a gibt $\mathcal{P}$ partielle Informationen über $[ax]_N$ zurück.

Gesucht: Es ist der Wert von x zu bestimmen.

Zu erklären bleibt das Orakel $\mathcal{P}$, welches die partiellen Informationen von $[ax]_N$ zurückgibt. Auch wenn dieses Orakel bereits als Eingabe den Multiplikator a erhält und diesen nicht intern bestimmt, ergibt sich dessen Wahrscheinlichkeit über alle Eingabewerte.

Definition 12 (CMHNP-Orakel)**.** Eine polynomielle, nichtdeterministische Turingmaschine $\mathcal{P}$ heißt $(N, i, \epsilon(n))$-CMHNP-Orakel, wenn für $n = \lceil \log N \rceil$ und $x \in \mathbb{Z}_N$ gilt

$$Pr\left(a \in_U \mathbb{Z}_N : \mathcal{P}(a) = \mathrm{bit}_i([ax]_N)\right) \geq \frac{1 + \epsilon(n)}{2}.$$

Zur einfacheren Notation wird im Folgenden von einem $\epsilon(n)$-CMHNP-Orakel oder einfach einem CMHNP-Orakel $\mathcal{P}$ gesprochen, da N, i und x für die Dauer der Untersuchung als fest gewählt angenommen werden und immer aus dem Kontext ersichtlich sind. Falls die Erfolgswahrscheinlichkeit des Orakels direkt aus dem Kontext ersichtlich ist, wird auch diese Angabe weggelassen.

Solche CMHNP-Orakel werden die Grundlage für die folgenden Untersuchungen bilden. Die Möglichkeit ein einzelnes RSA-Bit zu bestimmen, wird dabei durch die Erfolgswahrscheinlichkeit eines CMHNP-Orakel für dieses Bit beschrieben. Dass es sich beim CMHNP tatsächlich um eine realistische Basis für diese Untersuchungen handelt, kann durch eine einfache Umformung gezeigt werden.

Angenommen, es existiert ein $\epsilon(n)/2$-CMHNP-Orakel für das i-te RSA-Bit. Durch die Umformung $[a^e \cdot E_N(x)]_N = E_N(ax)$, mit einem beliebigen a, ist dieses Orakel zugleich ein CMHNP-Orakel für dieses Bit. Analog ist ein CMHNP-Orakel für ein i-tes Bit durch die Wahl des Multiplikators 1 zugleich ein Orakel für das i-te Bit des Klartextes x. Daher sind beide Formulierungen äquivalent.

Ziel des folgenden Kapitels wird es daher sein, aus einem CMHNP-Orakel für ein Bit i eine Lösung des CMHNP zu konstruieren, also den Klartext x zu bestimmen.

Da aber angenommen wird, dass das CMHNP, wie auch das RSA-Problem, schwer ist, kann solch diese Lösung nicht existieren. Folge wird ein Widerspruch zur Annahme sein, dass das verwendete CMHNP-Orakel existiert.

2.4 Sicherheit der Randbits

Es hat sich gezeigt, dass die Sicherheit der untersten und obersten Bits eines RSA-Klartextes einfacher zu zeigen ist, als die Sicherheit der inneren Bits. Dieser Unterschied ist vor allem durch die engere Beziehung eines Randbits zum niederwertigsten bzw. höchstwertigsten Bit zu rechtfertigen. Dadurch können bei den äußeren Bits Methoden verwendet werden, deren Funktionsweisen auf Eigenschaften des untersten oder obersten Bits basieren.

Untersuchungen zur Sicherheit der äußeren Bits haben eine lange Geschichte. Als erste zeigten Goldwasser, Micali und Tong in [GMT82] die Sicherheit des untersten Bits vor einer zuverlässigen Bestimmung. Weitere Untersuchungen verschiedener Autoren folgten. Über Zwischenschritte, wie einer $0.732 + \epsilon(n)$-Sicherheit von Vazirani und Vazirani [VV84] oder einer $1/2 + 1/\operatorname{poly}(\log(n))$-Sicherheit von Chor und Goldreich [CG85], gelang es schließlich Alexi, Chor, Goldreich und Schnorr in [ACGS88] eine $1/2 + 1/\operatorname{poly}(n)$-Sicherheit zu zeigen. Dieses Ergebnis konnten sie sogar auf die $\mathcal{O}(\log(n))$ obersten und $\mathcal{O}(\log(n))$ untersten Bits ausdehnen. Später zeigten in [FS97] auch Fischlin und Schnorr dieses Ergebnis. Sie verwendeten jedoch einen effizienteren Algorithmus zur Berechnung des Klartextes.

Beide Verfahren werden im Laufe dieser Arbeit vorgestellt. Zuerst erfolgt die Beschreibung des Verfahrens von Alexi et al. Anschließend wird im folgenden Kapitel das Verfahren von Fischlin und Schnorr zusammen mit einem weiteren Verfahren verwendet, um die Sicherheit aller inneren Bits zu zeigen. Diese Kombination zweier Verfahren geht auf [HN04] zurück und wird den Kern der Arbeit bilden.

2.4.1 Das Verfahren von Alexi et al.

In diesem Abschnitt wird der Sicherheitsbeweis für die äußeren RSA-Bits von Alexi, Chor, Goldreich und Schnorr vorgestellt. Ziel wird es dabei nicht sein, eine vollständige Analyse durchzu-

führen, sondern lediglich die zugrunde liegenden Ideen zu vermitteln. Der vollständige Beweis ist in [ACGS88] zu finden.

Bei dieser Untersuchung wird vereinfacht untersucht, ob das unterste Klartextbit zuverlässig bestimmt werden kann. Dazu sei ein zuverlässiges CMHNP-Orakel $\mathcal{P}$ für das unterste Bit gegeben. Mit Verwendung dieses Orakels wird ein Algorithmus konstruiert, der in polynomieller Zeit den gesamten Klartext bestimmen kann. Unter der Annahme, dass das CMHNP schwer zu lösen ist, kann die ursprüngliche Annahme der Existenz des Orakels aber nicht stimmen. Es folgt somit die Sicherheit, dass dieses Bit nicht zuverlässig bestimmt werden kann.

Die nun bestehende Aufgabe ist demnach die Konstruktion des Invertierungsalgorithmus. Von Alexi et al. wurde ein Verfahren zum Bestimmen des größten gemeinsamen Teilers zweier Zahlen verwendet. Es wird zuerst angenommen, dass zwei Zahlen a und b gegeben wären, für die die Eigenschaft $\mathrm{ggT}([ax]_N, [bx]_N) = 1$ gilt. Zudem gelte für x, dass x invertierbar ist und eine multiplikativ inverse Zahl l besitzt, so dass $[xl]_N = 1$ gilt.

Natürlich besitzen nahezu alle Zahlen in $\mathbb{Z}_N$ ein multiplikativ inverses Element. Warum dieses bei x und l so besonders hervorgehoben wird, hat aber einen einfachen Grund. Die Bestimmung von $\mathrm{ggT}([ax]_N, [bx]_N)$ kann nicht mit einem Standardverfahren, wie z.B. dem erweiterten euklidischen Algorithmus vorgenommen werden. Als Problem stellt sich nämlich dar, dass der Wert von x unbekannt ist. Lediglich über das CMHNP-Orakel $\mathcal{P}$ kann der Wert des untersten Bits von $[ax]_N$ für beliebiges $a \in \mathbb{Z}_N$ bestimmt werden. Durch diesen Wert kann aber zugleich auch bestimmt werden, ob es sich um eine gerade oder eine ungerade Zahl handelt.

Es kann nun gezeigt werden, dass die Möglichkeit, die Paritäten zweier Zahlen bestimmen zu können, genügt, um den größten gemeinsamen Teiler dieser Zahlen zu bestimmen. Dieses liefert der ggT-Algorithmus von Brent und Kung [BK83], welcher im Folgenden auch BRENTKUNGGGT genannt wird.

Der BRENTKUNGGGT erwartet, für die Bestimmung des größten gemeinsamen Teilers von $[ax]_N$ und $[bx]_N$, als Eingabe zwei Werte a und b sowie Zugriff auf ein CMHNP-Orakel $\mathcal{P}$ für das unterste Bit. In einem iterativen Verfahren wird damit ein Wert l bestimmt, welcher der Gleichung $\mathrm{ggT}([ax]_N, [bx]_N) = [lx]_N$ genügt.

Das Vorgehen des BRENTKUNGGGT ist iterativ. In jedem Schritt wird der Wert einer der beiden Zahlen a und b verringert, gleichzeitig bleibt der Wert des größten gemeinsamen Teilers aber unverändert. Für dieses Verfahren ist invariant, dass zu Beginn jeder Iteration einer der Werte von $[ax]_N$ und $[bx]_N$ ungerade und der andere ungleich 0 ist. Ohne Einschränkung sei hier $[ax]_N$ ungerade und $[bx] \neq 0$. Es soll nun ein neuer kleinerer Wert für $[bx]_N$ bestimmt werden, ohne den Wert des größten gemeinsamen Teilers zu ändern. Dazu wird eine der folgenden Operationen ausgeführt:

Falls $\mathcal{P}([bx]_N) = 0$ **(gerade):** Dann gilt $\mathrm{ggT}([ax]_N, [bx]_N) = \mathrm{ggT}([ax]_N, [bx \cdot 2^{-1}]_N)$

Falls $\mathcal{P}([bx]_N) = 1$ **(ungerade):**

Falls $\mathcal{P}([(ax + bx)/2]) = 0$**:** $\mathrm{ggT}([ax]_N, [bx]_N) = \mathrm{ggT}(a, \frac{a+b}{2})$

Falls $\mathcal{P}([(ax + bx)/2]) = 1$**:** $\mathrm{ggT}([ax]_N, [bx]_N) = \mathrm{ggT}(a, \frac{a-b}{2})$

Da durch dieses Vorgehen der Wert von a und b kontinuierlich verringert wird, ergibt sich nach endlich vielen Iterationen der Wert von l.

Damit kann nun ein Invertierungsalgorithmus angegeben werden. Die hier vorgestellte Form ist jedoch eine erhebliche Vereinfachung zum ursprünglichen Verfahren. Beim ursprünglichen Verfahren von Alexi et al. konnte auch ein CMHNP-Orakel mit einer Erfolgswahrscheinlichkeit

von nur $1/2 + \epsilon(n)$ verwendet werden. Dort wurde daraus ein neues Orakel mit einer erheblich höheren Erfolgswahrscheinlichkeit konstruiert. Das neue Orakel konnte dann direkt in einer komplizierteren Version des folgenden Algorithmus eingesetzt werden.

Die einfache Form des Algorithmus lautet nun wie folgt:

Input : $N, e, E_N(x)$
Output : x
repeat
  $a, b \in_U \mathbb{Z}_N$;
  $l \leftarrow$BRENTKUNGGGT($[ax]_N, [bx]_N$);
until $E_N(lx) = 1$;
return $x = l^{-1}$;

Algorithmus 1 : Vereinfachter Invertierungsalgorithmus nach Alexi et al.

Die Terminierungsbedingung ist eine erfolgreiche Überprüfung von $E_N(lx) = 1$. Im positiven Falle liefert diese Überprüfung $[lx]_N = 1$. Damit kann dann der Klartext als $x = l^{-1}$ bestimmt werden. Dabei wird die RSA-Eigenschaft $E_N(1) = 1$ verwendet.

Es bleibt die Frage zu klären, wie viele Durchläufe erwartet durchzuführen sind, bis der Algorithmus ein gutes Paar a, b gefunden hat. Dass dieses in polynomieller Zeit zu erwarten ist, wird von einem berühmten Satz von Dirichlet (siehe [Knu98], Seite 342) garantiert. Dieser besagt, dass die Wahrscheinlichkeit, für zwei zufällig gewählte ganze Zahlen aus dem Intervall $[-K, K]$ (K genügend groß) teilerfremd zu sein, gegen $6/\pi^2$ konvergiert. Somit müssen erwartet $\lceil \pi^2/6 \rceil$ Zahlenpaare zufällig gleichverteilt gewählt werden, damit eines dieser Paare die Bedingung $[ax]_N$, $[bx]_N$ teilerfremd erfüllt.

Die hier vorgestellte Variante des Verfahrens von Alexi et al. erlaubt es, bei Verwendung eines zuverlässigen CMHNP-Orakel für das unterste Klartextbit, den vollständigen Klartext in Polynomialzeit zu berechnen. Eine Erweiterung des Verfahrens für unzuverlässige Verfahren, ebenso wie für alle äußeren Bits ist wie in [ACGS88] beschrieben möglich.

Da aus der Existenz des Invertierungsalgorithmus ein Widerspruch zur Schwere des CMHNPs folgt, können die hier verwendeten Orakel nicht existieren. Demnach sind diese Bits sicher.

2.4.2 Grenzen dieses Ansatzes

Mit diesem Ansatz gelingt es, die Sicherheit der oberen und unteren Bits zu zeigen. Eine Anwendung auf die inneren Bits liefert jedoch keinen Erfolg. Dieses hat mehrere Gründe. Das Kernargument ist dabei, dass die Korrelationen zwischen dem untersten (oder auch obersten) Bit und dem vom Orakel bestimmten Bit sehr gering ist. Je größer der Abstand zwischen beiden Positionen wird, desto schwieriger sind Rückschlüsse auf das unterste Bit zu ziehen. Eine Bestimmung des untersten Bits ist jedoch Funktionsvoraussetzung für das Verfahren von Alexi et al.

Als Erweiterung der hier vorgestellten Technik kann das Verfahren von Fischlin und Schnorr [FS97] gesehen werden. Statt dem größten gemeinsamen Teiler, wird dort direkt der Wert des Klartextes approximiert. Dieses ergibt zudem eine verbesserte Laufzeit. Ein Sicherheitsbeweis für innere Bits wird aber auch dadurch nicht erreicht.

2.4.3 Ausblick

Die in diesem Kapitel eingeführten Begriffe sind grundlegend für die weiteren Analysen. Das betrifft im speziellen die Begrifflichkeiten der Orakel, Unterscheider und Wahrscheinlichkeitsverteilungen, aber auch die Begriffe des Absolutwertes und der Bestimmtheiten von Bits. Bei den komplexeren Methoden des nächsten Kapitels werden diese als bekannt vorausgesetzt.

Die Kernfrage des folgenden Kapitels wird die Sicherheit der inneren Bits sein. Für diese Analysen ist eine komplett neue Technik notwendig. Dieses wird das Verfahren von Näslund sein. Eine erfolgreiche Anwendung auf das gegebene Problem liefert aber erst eine Kombination mit der Technik von Fischlin und Schnorr und die weiteren Untersuchungen von Håstad und Näslund.

3 Sicherheit der inneren Bits

In diesem Kapitel wird die Sicherheit der inneren Bits eines RSA-Klartextes untersucht. Dazu werden im wesentlichen zwei Techniken verwendet, die von Fischlin und Schnorr [FS97] sowie von Näslund [Nä96]. Beide Techniken können für sich alleine jedoch nicht die Sicherheit aller Bits zeigen. Erst eine von Håstad und Näslund vorgenommene Kombination [HN04] zeigt die Sicherheit aller inneren Bits.

Die Technik von Fischlin und Schnorr kann die Sicherheit der äußeren Bits zeigen. Dazu werden zwei große Zahlenintervalle in $\mathbb{Z}_N$ sowie ein Unterscheider für diese Intervalle benötigt. Aus einem gegebenen CMHNP-Orakel für ein äußeres Bit können diese Voraussetzungen hergeleitet werden. Eine Anwendung des Verfahrens von Fischlin und Schnorr ergibt dann einen Invertierungsalgorithmus für den RSA-Klartext und zeigt so die Sicherheit des betrachteten Bits.

Näslund verfolgt mit seiner Technik einen anderen Ansatz. Statt den Wert des Klartextes durch algebraische Operationen zu bestimmen, wird dieser bitweise berechnet. Dieses geschieht über eine Art 2-Bit-Fenster, welches über alle Bits des Klartextes geschoben werden kann und dabei die Werte der einzelnen Bits bestimmt. Allerdings ist dieses Verfahren allgemein für Funktionen des Typs $x \mapsto ax + b \bmod p$, mit einer Primzahl p definiert. Damit kann es nicht ohne Weiteres auf das RSA-Verfahren angewendet werden.

Das im Folgenden vorgestellte Ergebnis ist das Verfahren von Håstad und Näslund. Dieses Verfahren stellt eine Kombination der Techniken von Näslund und von Fischlin und Schnorr dar. Dadurch ergibt sich eine Technik, welche für fast alle Bits erfolgreich ist.

Ähnlich wie zuvor bei Fischlin und Schnorr, müssen auch hierbei Unterscheider und Intervalle angegeben werden. Der wesentliche Unterschied ist der, dass diese Intervalle eine andere Form haben. Technisch setzten sie sich dabei aus vielen einzelnen Intervallstücken mit bestimmten Eigenschaften zusammen. Wesentlicher Vorteil dieser neuen Intervalle ist, dass sie nahezu immer existieren. Mit Eingabe dieser Intervalle und des Unterscheiders kann dann ein Algorithmus definiert werden, der zwei Bits des Klartextes zugleich bestimmen kann. Ergebnis ist wieder ein 2-Bit-Fenster. Durch Schieben dieses Fensters über sämtliche Bits folgt schließlich die Bestimmung des Klartextes. Die Konstruktion der Intervalle und des Unterscheiders wird dabei auf Grundlage des angenommenen CMHNP-Orakels vorgenommen.

Erfreulicherweise ergibt eine weitere Analyse, dass in genau den Fällen, in denen die kombinierte Methode versagt, alle Voraussetzungen für das ursprüngliche Verfahren von Fischlin und Schnorr geschaffen werden können.

Zusammengefasst werden in diesem Kapitel folgende Schritte absolviert:

1. Erarbeiten einiger Hilfsmittel.
2. Vorstellung der Methode von Fischlin und Schnorr.
3. Vorstellung der Methode von Näslund.
4. Kombination beider Methoden.
5. Untersuchung, in welchen Fällen die kombinierte Methode angewandt werden kann.

6. Zeigen, dass in den übrigen Fällen die ursprüngliche Methode von Fischlin und Schnorr funktioniert.

Auch in diesem Kapitel gilt weiterhin die Konvention, dass i stets das Bit bezeichnet, für welches ein CMHNP-Orakel gegeben ist. Ebenso gibt $\epsilon(n)$ immer die Erfolgswahrscheinlichkeit dieses CMHNP-Orakels an.

Die in diesem Kapitel vorgestellten Verfahren und Definitionen basieren auf [HN04].

3.1 Test-Techniken

Bevor mit der Vorstellung dieser Verfahren begonnen wird, werden noch einige Hilfmittel erarbeitet. Dieses betrifft zum einen die Möglichkeit der Generierung einer großen Anzahl von Zufallszahlen mit bestimmten Eigenschaften, zum anderen die Möglichkeit der Bestimmung des Anteils der 1-Antworten bei Eingabe dieser neuen Zufallszahlen. Die Bestimmung des Anteils der 1-Antworten wird dadurch ermöglicht, dass die generierten Zufallswerte, obwohl sie von x abhängig sind, aufgrund ihrer Konstruktion bis auf einen kleinen Fehler genau bestimmt werden können.

Im Speziellen sollen Zahlen r_1 bis $r_{m(n)}$ erzeugt werden, die bei Multiplikation mit x gleichverteilte und paarweise unabhängige Werte ergeben. Eine weitere wesentliche Eigenschaft ist, dass diese Werte $[r_j x]_N$ bis auf kleine Fehler genau modulo N, wie auch modulo 2^{i+1}, bestimmt werden können.

Bei den weiteren Untersuchungen wird die Bestimmung der Anteile der 1-Antworten das gängige Werkzeug sein, zwei Mengen voneinander zu unterscheiden. Bei Eingabe der hier generierten Punkte ist diese Bestimmung der Antworten jedoch sehr genau möglich.

3.1.1 Erzeugung von Zufallswerten

Ziel dieses Abschnittes ist es, Zahlen r_1 bis $r_{m(n)}$ mit den zuvor skizzierten Eigenschaften zu erzeugen. Bevor aber das technische Lemma angegeben wird, sollen diese Eigenschaften genauer vorgestellt werden.

gleichverteilt und paarweise unabhängig Die generierten Punkte $[r_j x]_N$ sollen gleichverteilt und paarweise unabhängig sein. Durch diese Eigenschaft können die hier erzeugten Zahlen in mehreren Iterationen eines Algorithmus verwendet werden. Die Erfolgswahrscheinlichkeiten jeder einzelnen Iteration ergibt sich aufgrund der paarweisen Unabhängigkeit aber unabhängig von den anderen Iterationen. Dieses ist eine häufig verwendete Technik, um die Erfolgswahrscheinlichkeit eines Verfahrens durch iteriertes Berechnen zu erhöhen.

gut approximierbar Ebenso sollen die Punkte $[r_j x]_N$ bis auf einen sehr geringen Fehler approximiert werden können. Diese Approximation soll jeweils modulo N, als auch modulo 2^{i+1} möglich sein. Ergebnis dieser Approximationseigenschaft wird die sehr genaue Bestimmung des Unterscheiderverhaltens bei Eingabe der hier definierten Punkte sein.

Die Konstruktion der Zahlen $\{r_j\}$ geht auf Alexi et al. [ACGS88] zurück. Das folgende Lemma beschreibt dieses Konstruktionsverfahren und gibt die zuvor informell beschriebenen Eigenschaften präzise an.

Dabei seien die Werte $d_I(n)$ und $d_Y(n)$ ganze Zahlen in $\mathcal{O}(\log n)$. Der Wert von $m(n)$ soll polynomiell in n sein. Auch im weiteren Verlauf der Untersuchungen wird stets angenommen, dass für diese Werte gilt $d_I(n), d_Y(n), \log(m(n)) \in \mathcal{O}(\log n)$.

Lemma 13. *Sei $m(n) \in \mathrm{poly}(n)$. Seien für $d_I(n), d_Y(n) \in \mathcal{O}(\log n)$ zufällig gleichverteilte $r, s \in_U \mathbb{Z}_N$ gegeben, zusammen mit den Werten für*

$$B^i_{i-d_I(N)}([rx]_N), B^i_{i-(d_I(n)+\log m(n))}([sx]_N)$$

und

$$\left\lfloor \frac{2^{1+d_Y(n)}[rx]_N}{N} \right\rfloor, \left\lfloor \frac{2^{(1+d_Y(n)+\log m(n))}[sx]_N}{N} \right\rfloor.$$

Dann gilt:

1. *In polynomieller Zeit kann eine Liste von $m(n)$ Werten $\{r_j\}$ erzeugt werden, so dass jedes $[r_jx]_N$ gleichverteilt ist und alle $\{[r_jx]_N\}$ paarweise unabhängig sind.*

2. *Es können zwei Mengen $\{z_j^I\}$ und $\{z_j^Y\}$ gefunden werden, so dass für z_j mit $[z_j]_N = [r_jx]_N$ gilt*

$$|z_j - z_j^Y| \leq \frac{N}{2^{d_Y(n)}} \tag{3.1}$$

$$\mathrm{abs}_{2^{i+1}}(z_j - z_j^I) \leq 2^{i+1-d_I(n)} \tag{3.2}$$

Beweis. **zu 1:** Definiere für $j = 0 \ldots m(n) - 1$ die Werte $r_j := r + js$. Da r und s zufällig gleichverteilt gewählt wurden, sind auch alle erzeugten Werte $[r_jx]_N = [(r + js)x]_N$ zufällig gleichverteilt.

Es bleibt noch die paarweise Unabhängigkeit zu zeigen. Seien $a := [rx]_N$ und $b := [sx]_N$. Mit $0 \leq i, j < m(n)$ existieren für zufällig gewählte $c_1, c_2 \in \mathbb{Z}_N$ für die Gleichungen $a + ib = c_1$ und $a + jb = c_2$ eindeutige Lösungen für a und b. Dieses gilt, da wegen $i, j < m(n)$ die Werte i und j keine Teiler von N sein können[1]. Somit existiert ein multiplikativ Inverses zu $i - j$ modulo N und es gilt $z = [(i - j)^{-1} \cdot (c_1 - c_2)]_N$ und $y = [c_1 - iz]_N$. Damit gilt für zwei beliebige Zahlen r_i und r_j aus der generierten Menge:

$$\begin{aligned}
&\Pr(c_1, c_2 \in_U \mathbb{Z}_N : [r_ix]_N = c_1 \wedge [r_jx] = c_2) \\
&= \frac{1}{N^2} \\
&= \Pr(c_1 \in_U \mathbb{Z}_N : [r_ix]_N = c_1) \cdot \Pr(c_2 \in_U \mathbb{Z}_N : [r_jx] = c_2)
\end{aligned}$$

Dieses zeigt gerade die paarweise Unabhängigkeit.

zu 2: Sei $z_j = [r_jx]_N$. Definiere z_j^Y wie folgt:

$$z_j^Y := \frac{N}{2^{d_Y(n)+1}} \left\lfloor \frac{2^{1+d_Y(n)}[rx]_N}{N} \right\rfloor + j \cdot \frac{N}{2^{1+d_Y(n)+\log m(n)}} \left\lfloor \frac{2^{(1+d_Y(n)+\log m(n))}[sx]_N}{N} \right\rfloor$$

[1] Einzige echte Teiler von N sind p und q. Diese besitzen jeweils $\lceil n/2 \rceil$ Bitstellen und sind damit vom Wert nicht polynomiell in n.

Zu beachten ist, dass $2^{d_Y(n)}/N$ stets kleiner 1 gilt. Es gelte o.E. $z_j > z_j^Y$. Damit kann die folgende Abschätzung durchgeführt werden.

$$\begin{aligned}
|z_j - z_j^Y| &= z_j - z_j^Y \\
&= [r_j x]_N - z_j^Y \\
&= [rx + j \cdot sx]_N - \frac{N}{2^{d_Y(n)+1}} \left\lfloor \frac{2^{1+d_Y(n)}[rx]_N}{N} \right\rfloor - \\
&\quad j \cdot \frac{N}{2^{(d_Y(n)+1+\log m(n))}} \left\lfloor \frac{2^{(1+d_Y(n)+\log m(n))}[sx]_N}{N} \right\rfloor \\
&= \frac{N}{2^{d_Y(n)}} \left(\frac{2^{d_Y(n)}}{N}[rx + j \cdot sx]_N - \frac{1}{2} \left\lfloor \frac{2^{1+d_Y(n)}[rx]_N}{N} \right\rfloor - \right. \\
&\quad \left. \frac{1}{2} \cdot \frac{j}{m(n)} \cdot \left\lfloor \frac{2^{(1+d_Y(n)+\log m(n))}[sx]_N}{N} \right\rfloor \right) \\
&= \frac{N}{2^{d_Y(n)}} \left(\frac{1}{2} \cdot \frac{2^{1+d_Y(n)}}{N}[rx]_N - \frac{1}{2} \cdot \left\lfloor \frac{2^{1+d_Y(n)}[rx]_N}{N} \right\rfloor + \right. \\
&\quad \left. \frac{1}{2} \cdot \frac{2^{1+d_Y(n)}}{N}[j \cdot sx]_N - \frac{j}{2 \cdot m(n)} \cdot \left\lfloor \frac{2^{(1+d_Y(n)+\log m(n))}[sx]_N}{N} \right\rfloor \right) \\
&\leq \frac{N}{2^{d_Y(n)}} \cdot \left(\frac{1}{2} + \frac{1}{2} \right) = \frac{N}{2^{d_Y(n)}}
\end{aligned}$$

Ähnliche Argumente werden auch für die zweite Bedingung verwendet. Die Idee ist hierbei, die Bits $B^i_{i-d_I(n)+1}(z_j)$ zu eliminieren. Die Konstruktion von z_j^I erfolgt durch

$$z_j^I := B^i_{i-d_I(n)}([rx]_N)(d - d_I(n)) + B^i_{i-(d_I(n)+\log m(n))}([sx]_N) \cdot (i - d_I(n)) \cdot j.$$

Betrachtet man den Fall $\mathrm{abs}_{2^{i+1}}(z_j - z_j^I) = [z_j - z_j^I]_{2^{i+1}}$. Dann sind nach dieser Konstruktion die Bits $B^i_{i-d_I(n)}(z_j - z_j^I)$ gleich Null. Somit ist der Absolutwert modulo 2^{i+1} kleinergleich $2^{i+1-d_I(n)}$.

Im Falle $\mathrm{abs}_{2^{i+1}}(z_j - z_j^I) = 2^{i+1} - [z_j - z_j^I]_{2^{i+1}}$ ist der Wert offenbar kleiner als $2^{i+1-d_I(n)}$, denn nach Definition des Absolutwertes gilt $\mathrm{abs}_{2^{i+1}}(z_j - z_j^I) = \min\{[z_j - z_j^I]_{2^{i+1}}, 2^{i+1} - [z_j - z_j^I]_{2^{i+1}}\}$. Daraus folgt die zweite Behauptung. □

An dieser Stelle soll noch einmal die Bedeutung der Formeln (3.1) und (3.2) unterstrichen werden. Durch (3.1) wird garantiert, dass die Werte $[r_j x]_N$ modulo N mindestens $2^{-d_Y(n)}$-bestimmt sind. Eine ähnliche Aussage liefert (3.2). Dadurch wird garantiert, dass die Punkte $[r_j x]_N$ modulo $2^{i+1-d_I(n)}$ jeweils $2^{-d_I(n)}$-bestimmt sind.

Zum Schluss ist noch auf eine wesentliche Konventionen hinzuweisen. Die Werte r und s werden, nach erstmaliger zufälliger Wahl, im weiteren Verlauf als fest angenommen. Das bedeutet, dass diese Werte für die vollständige Untersuchung eines Bits fest bleiben.

In den folgenden Untersuchungen kann nun angenommen werden, dass eine polynomielle Anzahl an Werten mit den oben beschriebenen Eigenschaften zur Verfügung steht.

3.1.2 Unterscheider über Intervallen

Im weiteren Verlauf der Arbeit werden vor allem Unterscheider für Intervalle sowie deren Erfolgswahrscheinlichkeiten untersucht. Eine Definition dieser Bezeichnungen findet im Folgenden statt.

Definition 14. Mit einem *Intervall* $J := \{[u]_N, [u+1]_N, \ldots, [v]_N\}$ ist eine Menge von aufeinander folgenden Zahlen aus $\mathbb{Z}_N$ gemeint. Die *Länge* von J ist $|J|$, also die Anzahl der Elemente von J. Das *Maß* von J ist $\lambda(J) := \frac{|J|}{N}$.

Für ein Intervall J und ein $z \in \mathbb{Z}_N$ bezeichnet $J + z := \{[y+z]_N | y \in J\}$ die *Verschiebung* dieses Intervalls um z.

Mit $A^{\mathcal{P}}(J)$ ist die Wahrscheinlichkeit gemeint, dass der Unterscheider A, unter Verwendung des Orakels $\mathcal{P}$ bei Eingabe von $ax \in_U J$, die Antwort 1 ausgibt, also

$$A^{\mathcal{P}}(J) = \Pr(ax \in_U J : A(a) = 1).$$

Analog wird der Erfolg eines Unterscheiders $\Delta^{\mathcal{P}}(J_1, J_2)$ für zwei Intervalle J_1, J_2 definiert als

$$\Delta^{\mathcal{P}}(J_1, J_2) = |A^{\mathcal{P}}(J_1) - A^{\mathcal{P}}(J_2)|.$$

Um zwei Intervalle voneinander unterscheiden zu können, wird ein Unterscheider benötigt, der auf diesen Intervallen jeweils unterschiedliche viele Antworten 1 liefert.

Im weiteren Verlauf wird der Hauptverwendungszweck von Unterscheidern der sein, zu entscheiden in welchem von zwei gegebenen Intervallen ein Wert $[r_j x]_N$ liegt. Dieses kann mit einer gewissen Erfolgswahrscheinlichkeit aus der Antwort des Unterscheiders geschlossen werden. Um jedoch entscheiden zu können, in welchem von zwei Intervalle ein Wert liegt, ist es notwendig, die Wahrscheinlichkeit für eine 1-Antwort in beiden Intervallen zu kennen. Dieser Anteil der 1-Antworten in einem Intervall kann durch das folgende Lemma bestimmt werden.

Lemma 15. *Sei $\{r_j\}$ eine Menge von $m(n)$ Werten mit $[r_j x]_N$ gleichverteilt und $\{[r_j x]_N\}$ paarweise unabhängig. Sei $J \subset \mathbb{Z}_N$ eine Teilmenge[2] mit einem signifikantem Maß $\lambda(J)$. Zudem sei für jedes $[r_j x]_N$ mit einer maximalen Fehlerwahrscheinlichkeit von δ zu entscheiden, ob $[r_j x]_N$ in J liegt.*

Dann existiert eine Konstante c, und es gilt für jedes signifikante $\zeta(n)$ und $K(n) \in \mathrm{poly}(n)$, falls $m(n) = c\lambda(J)^{-1}\zeta(n)^{-2}K(n)$ und $\delta \leq c^{-1}\lambda(J)\zeta(n)/K(n)$, dass in polynomieller Zeit ein Wert $\tilde{p}$ berechnet werden kann, für den gilt

$$\Pr\left(|A^{\mathcal{P}}(J) - \tilde{p}| \geq \zeta(n)\right) \leq \frac{1}{K(n)} \tag{3.3}$$

Beweis. Durch die Bestimmung des Anteils der 1-Antworten über alle, als in J gelegen klassifizierten Punkte, kann der Wert von $\tilde{p}$ bestimmt werden. Die Klassifikation dieser Punkte $[r_j x]_N$ erfolgt durch deren Bestimmtheit.

Der hierbei entstehende Fehler setzt sich aus zwei Ereignissen zusammen:

1. E_1 sei das Ereignis, dass Punkte falsch klassifiziert werden.

2. E_2 sei das Ereignis, dass der erwartete Anteil der 1-Antworten falsch ist. D.h. der bestimmte Wert liegt mindestens $\zeta(n)/2$ vom tatsächlichen Wert entfernt.

[2] An dieser Stelle ist explizit kein zusammenhängendes Intervall gefordert. Vielmehr reicht bereits eine genügend große Anzahl von Punkten.

Das Ereignis E bezeichne nun den Fall, dass $|A^{\mathcal{P}}(J) - \tilde{p}| \leq \zeta(n)$ gilt. Die gesuchte Wahrscheinlichkeit ist daher $\Pr(\neg E)$. Dieses kann aber umgeformt werden zu:

$$\begin{aligned}
\Pr(\neg E) &= \Pr(\neg E \wedge E_1) + \Pr(\neg E \wedge E_1) \\
&= \Pr(\neg E | E_1) \Pr(E_1) + \Pr(\neg E \wedge \neg E_1) \\
&\leq \Pr(E_1) + \Pr(\neg E | \neg E_1) \Pr(\neg E_1) \\
&\leq \Pr(E_1) + \Pr(\neg E | \neg E_1)
\end{aligned}$$

Diese Wahrscheinlichkeiten gilt es zu bestimmen.

Es bezeichnet $\Pr(E_1)$ die Wahrscheinlichkeit, dass Punkte falsch klassifiziert wurden. Die damit falsch klassifizierten Punkte sind maximal $\delta m(n) \leq \zeta(n)^{-1}$ Punkte, was direkt aus den geforderten Bedingungen für δ folgt. Da diese Punkte gleichverteilt sind, kann dieser Wert mit einer Wahrscheinlichkeit von $\frac{1}{2K(n)}$ auf den Wert von $2K(n)\zeta(n)^{-1}$ begrenzt werden.

Als nächstes muss der zweite Fehlerterm abgeschätzt werden. Dieser bezeichnet die Wahrscheinlichkeit, dass bei einer zuverlässigen Klassifikation der Punkte ein falscher Anteil der 1-Antworten bestimmt wird.

Dabei sei p die Wahrscheinlichkeit für einen beliebigen in J gelegenen Wert, dass das Orakel bei Eingabe dieses Punktes 1 ausgibt. Angenommen, im schlechtesten Falle würde für alle der höchstens $2K(n)\zeta(n)^{-1}$ falsch klassifizierten Punkte die gleiche Antwort gegeben. Dann liegt der Erwartungswert von p mal die in J klassifizierten Punkte maximal $2K(n)\zeta(n)^{-1}/m(n)$ vom tatsächlichen Wert entfernt. Der Erwartungswert sei nun $\tilde{p}$.

Die Varianz der Orakelantwort für einen einzelnen Punkt aus J ist $p - p^2 = p \cdot (1-p)$. Sei L die Anzahl der in J klassifizierten Punkte. Aufgrund der paarweisen Unabhängigkeit dieser Punkte ergibt sich für die Varianz:

$$\begin{aligned}
\mathrm{Var}(X) &= L \cdot p \cdot (1-p) \\
&\leq \lambda(J)^{-1} m(n) \cdot p(1-p) \\
&\leq \lambda(J)^{-1} m(n)/2
\end{aligned}$$

Mit einer Anwendung der Tschebyscheff'schen Ungleichung folgt als Ergebnis für die Wahrscheinlichkeit $\Pr(\neg E | E_1)$:

$$\begin{aligned}
&\Pr\left(|X - \hat{p}| \geq \zeta(n)\right) \leq \frac{\lambda(J)^{-1}}{\zeta(n)} \frac{1}{2} = m(n)^2 \cdot \frac{1}{2K(n)} \\
\Rightarrow &\Pr\left(|A^{\mathcal{P}}(J) - \tilde{p}| \geq \zeta(n)/m(n)\right) \leq m(n)^2 \frac{1}{2K(n)} \\
\Rightarrow &\Pr\left(|A^{\mathcal{P}}(J) - \tilde{p}| \geq \zeta(n)\right) \leq \frac{1}{2K(n)}
\end{aligned}$$

Insgesamt ergibt dieses eine Wahrscheinlichkeit von

$$\Pr\left(|A^{\mathcal{P}}(J) - \tilde{p}| \geq \zeta(n)\right) \leq \frac{1}{2K(n)} + \frac{1}{2K(n)} = \frac{1}{K(n)}$$

□

Dieses Lemma hat entscheidende Bedeutung für die Funktionsweise der von Håstad und Näslund vorgestellten Methode. Durch den Appoximationswert $\tilde{p}$ kann eine Aussagen darüber getroffen werden, ob es sich bei getesteten Punkten um Punkte aus dem Intervall J handelt oder nicht. Diese Positionsbestimmung eines Punktes wird eine wesentliche Technik im weiteren Verlauf sein.

3.2 Verfahren von Fischlin und Schnorr

Dieser Abschnitt beschreibt eine Invertierungsmethode, die auf dem Algorithmus von Fischlin und Schnorr [FS97] basiert. Das hier vorgestellte Verfahren hat große Ähnlichkeit zum Verfahren von Alexi et al. Es wird jedoch, statt der Berechnung eines größten gemeinsamen Teilers, der gesuchte Klartextwert durch ein Approximationsverfahren direkt bestimmt.

Ein weiterer Unterschied liegt in der hier vorgenommenen Beschreibung. Beim Verfahren von Alexi et al. wurde unmittelbar ein CMHNP-Orakel im Invertierungsalgorithmus verwendet. Für das nun vorgestellte Verfahren wird dagegen ein Unterscheider verlangt. Ebenso wird auch die Existenz von zwei Intervallen benötigt, die signifikant unterschieden werden können. Die Konstruktion des Unterscheiders und das Finden der Intervalle wird in den späteren Abschnitten vorgenommen. Teilweise ergeben sich diese sogar als „Abfallprodukte“ während der Analysen.

Die Idee dieses Verfahrens ist es nun, von einer guten Vermutung für x aus, nacheinander die Zahlen $[x/2]_N, [x/4]_N, \ldots$ zu bestimmen. Dabei soll der Wert aber jeweils genauer werden. Dieses erfolgt durch die Benutzung eines Orakel für das unterste Bit. Dieses Orakel kann aus dem gegebenen Intervallunterscheider konstruiert werden und besitzt eine sehr hohe Genauigkeit. Der überraschende Effekt bei diesem Verfahren wird sein, dass sich der Fehler bei jeder Berechnung einer neuen Zahl halbiert. Sobald ein genügend kleiner Fehler erreicht wurde, ist daher $[x/2^k]_N$ bereits eindeutig bestimmt, damit also auch der Wert von $[x]_N$.

Aus diesem Kurzabriss ergeben sich aber einige fundamentale Fragen:

1. Wie wird das Orakel für das unterste Bit konstruiert?

2. Wie kann ein guter Initialwert gefunden werden?

3. Wie kann der Fehler bei der Bestimmung des nächsten Wertes verringert werden?

4. Wie können der Unterscheider und die Intervalle gefunden werden?

Diese Fragen sollen nun nach und nach beantwortet werden.

Zuerst ist die Frage nach der Orakelkonstruktion zu beantworten. Die hier vorgenommene Konstruktion folgt dem Vorgehen in [HN04]. Zu beachten ist, dass das zu konstruierende Orakel neben dem Chiffretext auch einen Multiplikator a als Eingabe erhält. Es ist damit das unterste Bit von $[ax]_N$ zu bestimmen.

Die grundsätzliche Idee bei der Konstruktion ist es, den Wert des untersten Bits dadurch zu bestimmen, in welchem Intervall sich ein bestimmtes $[ax]_N$ befindet. Angenommen wir hätten ein $a := [r_j + \frac{1}{2}]_N$, welches einige technische Details erfüllt[3] und es gelte $r_j x + \frac{x - \mathrm{lsb}(x)}{2} \in J$ für ein passend gewähltes Intervall J. Dann gilt offenbar

$$[ax]_N = r_j x + \frac{x - \mathrm{lsb}(x)}{2} + \mathrm{lsb}(x)\frac{N+1}{2} \in J + \mathrm{lsb}(x)\frac{N+1}{2}.$$

Somit kann durch einen Intervallunterscheider, der angibt in welchem dieser beiden Intervalle sich $[ax]_N$ befindet, das unterste Bit bestimmt werden. Mit den Ergebnissen des vorherigen Abschnittes kann zudem eine hohe Erfolgswahrscheinlichkeit für dieses Vorgehen garantiert werden.

Die genaue Konstruktion des Orakels liefert das folgende Lemma.

[3] Diese werden in Lemma 16 aufgeführt.

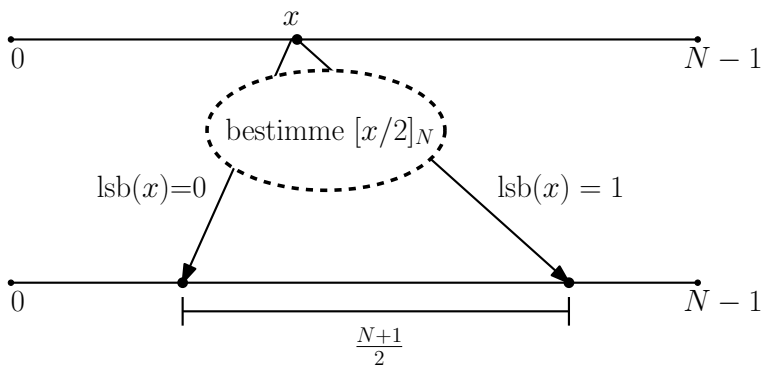

Abbildung 3.1: Bestimmung von $\mathrm{lsb}(x)$ aus der Position von $[x/2]_N$.

Lemma 16. *Sei $J \subset \mathbb{Z}_N$ ein Intervall mit signifikantem $\lambda(J)$. Sei für dieses Intervall ein Unterscheider $\Delta^{\mathcal{P}}(J, J + \frac{N+1}{2}) > \zeta'(n)$ mit signifikantem $\zeta'(n)$ gegeben.*

Dann kann in zufälliger polynomieller Zeit ein Orakel $\mathcal{P}'$ konstruiert werden, so dass für alle $\frac{\lambda(J)\zeta'(n)}{96cn}$-bestimmte Punkte $[ax]_N$ das Orakel $\mathcal{P}'$ das unterste Bit von $[ax]_N$ mit Wahrscheinlichkeit mindestens $1 - \frac{1}{2n}$ bestimmen kann. c ist konstant und trägt den Wert wie in Lemma 15 angegeben.

Beweis. Sei $\{r_j\}$ eine Menge von $m(n)$ Werten. Dabei sei $m(n)$ polynomiell und die Werte seien mit Lemma 13 generiert. Durch Verwendung dieser Werte im vorliegenden Intervallunterscheider, können nach Lemma 15 Approximationen $\tilde{p}_1$ von $A^{\mathcal{P}}(J)$ und $\tilde{p}_2$ von $A^{\mathcal{P}}(J + \frac{N+1}{2})$ bestimmt werden.

Für die Bestimmung der Approximationen sei in Lemma 15 die Wahrscheinlichkeit $\zeta(n) := \zeta'(n)/6$ gewählt. Der Wert $K(n)$ kann als $K(n) := 4n$ bestimmt werden, da $m(n)$ mit dieser Wahl weiterhin polynomiell in n bleibt. Ebenso wird mit der Wahl von $d_Y(n) = \log \frac{48nc}{\lambda(J)\zeta'(n)} \in \mathcal{O}(\log n)$ die in Lemma 15 geforderte Bedingung für δ eingehalten, denn

$$\delta = 2^{-d_Y(n)} = \left(\frac{48nc}{\lambda(J)\zeta'(n)}\right)^{-1} = \frac{\lambda(J)\zeta'(n)}{48nc} \leq \frac{\lambda(J)\zeta'(n)}{24nc}.$$

Damit approximieren also $\tilde{p}_1$ und $\tilde{p}_2$ mit einer Wahrscheinlichkeit von mindestens $1 - \frac{1}{4n}$, bis auf einen maximalen Fehler $\zeta'(n)/6$, das Verhalten von $\mathcal{P}'$ auf beiden Intervallen.

Durch eine weitere Anwendung von Lemma 15 auf die Werte $[2^{-1}a + r_j]_N$, mit r_j wie in Lemma 13 generiert, wird nun der Wert des untersten Bits wie folgt bestimmt.

Angenommen es gelte $\mathrm{lsb}(ax) = 0$. Wegen der $\frac{\lambda(J)\zeta'(n)}{96cn}$-Bestimmtheit von $[ax]_N$ kann eine neue Approximation von $A^{\mathcal{P}}(J)$ mit Ergebnis nahe $\tilde{p}_1$ durchgeführt werden. Lemma 15 garantiert dafür einen sehr kleinen Fehler und der Wert liegt nahe $\tilde{p}_1$. Andernfalls ergibt sich ein Wert nahe $\tilde{p}_2$ und es gilt $\mathrm{lsb}(ax) = 1$.

Seien ohne Einschränkung $\tilde{p}_1 < \tilde{p}_2$. Damit kann $\mathcal{P}'$ nach folgendem Verfahren das unterste Bit entscheiden:

$$\mathcal{P}'(ax) = \begin{cases} 0, & \text{falls neuer Wert} \leq (\tilde{p}_1 + \tilde{p}_2)/2 \\ 1, & \text{sonst} \end{cases} \tag{3.4}$$

Aufgrund der anfangs gewählten Wahrscheinlichkeiten für $K(n)$ ergibt sich ein Fehler von höchstens $\frac{1}{2n}$. □

Als nächstes ist die Frage nach dem Initialwert zu klären. Ein guter Initialwert, also eine gute Vermutung, ist nämlich Voraussetzung für das Funktionieren dieses Verfahrens. Falls kein guter Initialwert gefunden wird, ist es nicht möglich in einer polynomiellen Anzahl an Schritten den Wert von x zu bestimmen. Die Lösung des Problems ist simpel. Anstelle eines einzelnen Initialwertes, wird zu Anfang bereits eine polynomiell große Anzahl an Kandidaten gewählt. Diese Menge enthält dann, mit genügend hoher Wahrscheinlichkeit, mindestens einen guten Kandidaten. Die Laufzeit des Verfahrens bleibt damit auch weiterhin polynomiell.

Die letzte Frage, die noch mit den aktuell vorhandenen Hilfsmitteln geklärt werden kann, ist die Konstruktion des Invertierungsalgorithmus, also die Bestimmung des Klartextes. Gegeben sei ein guter Initialwert y'. Aus diesem soll der Wert $y/2$ berechnet werden. Es gelte jedoch die Einschränkung, dass $y/2$ nur noch einen halb so großen Fehler wie y' besitzt soll.

Dafür kann nun das Orakel $\mathcal{P}'$ für das unterste Bit verwendet werden, wie es in Formel (3.4) konstruiert wurde. $\mathcal{P}'(a, y)$ erhält dabei als Eingabe einen Multiplikator a sowie die Approximation y von $[ax]_N$.

Die genaue Berechnung eines neuen Wertes ist wie folgt. Sei y der bereits berechnete Wert aus einem Schritt j, also für $[x/2^j]_N$. Ein neuer Wert y' berechnet sich daraus durch die Formel

$$y' = \frac{y - \mathcal{P}'(2^{-j}, y)}{2} + \mathcal{P}(2^{-j}, y)\frac{N+1}{2}. \tag{3.5}$$

Das Ergebnis y' dieser Berechnung ist damit eine Approximation von $[x/2^{j+1}]_N$. Für die erste Iteration ist dieses gerade $y' = (y - \mathrm{lsb}(x))/2 + \mathrm{lsb}(x) \cdot (N+1)/2$ mit der Initialschätzung y. Der anfänglich große Approximationsfehler ergibt sich aus dem Fehler der Initialschätzung.

Die Halbierung des Approximationsfehlers folgt nun aus der Befragung von $\mathcal{P}'$. Dadurch wird in einem j-ten Schritt nicht direkt $[y/2]_N$, sondern ein etwas genauerer Wert für $[x/2^j]_N$ bestimmt. Genauer gilt damit, wenn der Approximationsfehler von y im j-ten Schritt noch durch $|y - [x2^{-j}]_N| \leq \frac{N}{2^j n^k}$ beschränkt war, ist dieses nach der Berechnung nur noch $|y' - [x2^{-(j+1)}]_N| \leq \frac{N}{2^{j+1} n^k}$. Dieses Entspricht der Verringerung des Fehlers um den Faktor $1/2$. Induktiv erhalten wir nach n Schritten eine sehr genaue Approximation von $[x2^{-n}]_N$, mit einem Fehler kleiner $\frac{N}{2^n n^k}$.

Die Umsetzung dieses Verfahrens folgt in Algorithmus 2.

Input : $N, e, E_N(x)$
Output : x
„Rate“ $y \in \mathbb{Z}_N$ mit $\mathrm{abs}_N(x - y) \leq N\lambda(J)\zeta(n)/(6n^2)$;
for $j \leftarrow 0$ **to** $n - 1$ **do**
  $b \leftarrow \mathcal{P}'(2^{-j}, y)$;
  $y \leftarrow \frac{y-b}{2} + b\frac{N+1}{2}$ `// Approximation des nächsten Wertes`
end
return $x = y \cdot 2^n$ `// Berechnung von` x

Algorithmus 2 : Invertierungsalgorithmus nach Fischlin und Schnorr

Lemma 17. *Sei $\mathcal{P}$ ein Orakel, so dass für ein Intervall J ein Unterscheider $\Delta^{\mathcal{P}}(J, J + \frac{N+1}{2}) \geq \zeta(n)$ mit signifikantem $\zeta(n)$ und signifikantem $\lambda(J)$ existiert. Dann kann in zufälliger polynomieller Zeit mit einer Wahrscheinlichkeit von mindestens $1/2$ der Klartext x bestimmt werden.*

Beweis. Mit Lemma 16 kann mit den genannten Voraussetzungen ein Orakel $\mathcal{P}'(a, y)$ für das unterste Bit von $[ay]_N$ bei unbekanntem y und bekannten a konstruiert werden. Dieses Orakel besitzt eine Erfolgswahrscheinlichkeit von mindestens $1 - \frac{1}{2n}$.

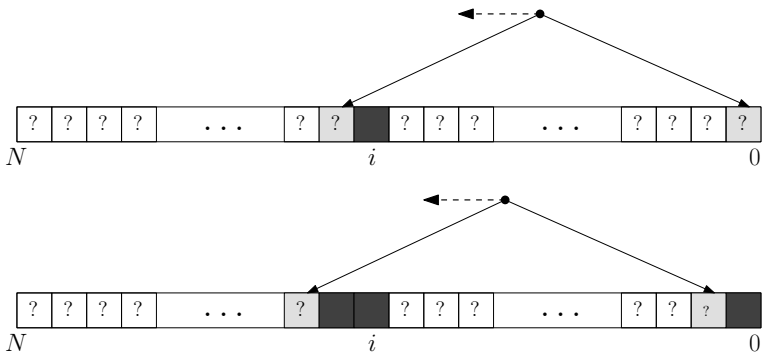

Abbildung 3.2: Das Näslund-Verfahren mit 2-Bit Fenster.

Grundlage für das Invertierungsverfahren ist der oben beschriebene Algorithmus 2 von Fischlin und Schnorr. Durch iteratives Testen einer polynomiellen Anzahl gleichverteilter Punkte kann das „Raten“ ersetzt werden. Dabei erhöht sich die Laufzeit lediglich um einen polynomiellen Faktor.

Durch vollständige Induktion kann nun gezeigt werden, dass in einem j-ten Schritt für den Absolutwert gilt

$$\mathrm{abs}_N(x-y) \leq 2^{-j} N \lambda(J) \zeta(n)/(6n^2).$$

Damit sind wieder die Voraussetzungen für eine erneute Verwendung des Orakels im $j+1$-ten Schritt gegeben.

Als Erfolgswahrscheinlichkeit ergibt sich bei n Iterationen $1 - n \cdot \frac{1}{2n} = 1 - \frac{1}{2} = \frac{1}{2}$. Damit folgt die Behauptung. □

Ergebnis dieses Abschnittes ist das Verfahren von Fischlin und Schnorr. Sobald die Voraussetzungen aus Lemma 17 vorliegen, kann damit direkt auf einen Invertierungsalgorithmus geschlossen werden. Die schwierige Voraussetzung für dieses Verfahren, ist aber die Existenz und das Finden der benötigten Intervalle. Durch die Einschränkung auf ein signifikantes $\lambda(J)$ kann diese Forderung unmittelbar nur für eine sehr geringen Anzahl an Unterscheidern gezeigt werden. Für die gewünschten und umfassenderen Ergebnisse für alle Bits, sind genauere Methoden notwendig, wie sie im Folgenden u.a. von Näslund bereit gestellt werden.

3.3 Der Näslund-Ansatz

Einen grundlegend anderen Ansatz verwendet nun das Verfahren von Näslund [Nä96]. Statt den Wert des Klartextes durch Approximation oder algebraische Operationen zu bestimmen, wird dieser bitweise berechnet. Dazu wird ein 2-Bit Fenster konstruiert, welches über sämtliche Klartextbits von rechts nach links „geschoben“ werden kann. Ergebnis ist die Bestimmung sämtlicher Klartextbits und somit des Klartextes.

Ursprünglich verwendete Näslund dieses Verfahren, um in [Nä96] die Sicherheit aller Bits für Funktionen des Typs $x \mapsto [ax+b]_p$ mit $a, b \in \mathbb{Z}_N$ und p prim zu zeigen. Er konstruierte dazu aus einem gegebenen CMHNP-Orakel für ein i-tes Bit ein Orakel für die Bits 0 und $i+1$. Wie in Abbildung 3.2 graphisch dargestellt, handelt es sich bei diesem neuen Orakel um eine Art Fenster. Das Schieben dieses Fensters über sämtliche Bits wird durch Manipulation der Werte a und b ermöglicht.

Die Form der RSA-Verschlüsselungsfunktion E_N ist jedoch nur in gewissem Maße mit der Form der ursprünglich untersuchten Funktionen vergleichbar. Den Unterschied bilden dabei zwei wesentliche Punkte. Zum einen ist der RSA-Modulus aufgrund seiner Definition keine Primzahl. Zum anderen, was die erheblich größere Einschränkung bedeutet, wird x auf $[x^e]_N$ abgebildet.

Dennoch werden die folgenden Untersuchungen ergeben, dass die Idee dieses Verfahrens auch für den aktuellen Fall angewendet werden kann.

3.4 Kombination beider Ansätze

In diesem Abschnitt werden die Verfahren von Näslund sowie von Fischlin und Schnorr in der Weise kombiniert, dass die Sicherheit aller inneren Bits gezeigt werden kann. Damit werden hier die Ergebnisse der Arbeit von Håstad und Näslund [HN04] dargestellt. Die äußeren Bits können von dieser Untersuchung ausgeschlossen werden, da ihre Sicherheit bereits von Alexi et al. [ACGS88] gezeigt wurde.

Es sollen nun die Techniken von Näslund, Fischlin und Schnorr so miteinander verknüpft werden, dass sie zusammen einen Invertierungsalgorithmus auf Grundlage eines CMHNP-Orakels für ein i-tes Bit ergeben. Zuerst wird das Verfahren von Fischlin und Schnorr, bzw. eine modifizierte Form davon, die Voraussetzungen für ein 2-Bit Orakel schaffen. Mit der Näslund-Technik wird dann aus diesem 2-Bit-Orakel ein Invertierungsalgorithmus konstruiert. Während dieser Untersuchungen wird sich aber herausstellen, dass dieses Vorgehen nur dann gelingt, wenn bestimmte Beziehungen zwischen N und 2^{i+1} gelten. Das Vorhandensein dieser Beziehungen ist abhängig von i und nicht immer gegeben. Weitere Untersuchungen werden jedoch ergeben, dass ohne Vorliegen dieser Beziehungen, die Voraussetzungen für das Verfahren von Fischlin und Schnorr geschaffen werden können. Somit werden mit diesem Verfahren alle Fälle behandelt.

Der erste Schritt wird nun die Konstruktion eines Invertierungsalgorithmus auf Grundlage des 2-Bit Orakels sein. Dieser Schritt ist relativ einfach. Aufwändiger gestaltet sich die Konstruktion des 2-Bit Orakels. Dessen Konstruktion wird durch ein ähnliches Verfahren erreicht, wie es zuvor beim Verfahren von Fischlin und Schnorr zur Bestimmung des untersten Bits verwendet wurde.

Idee für das 2-Bit Orakel

Bei Fischlin und Schnorr wurde der Wert $[x/2]_N$ untersucht, da die Position dieses Wertes Aufschluss über das unterste Bit von x geben konnte (vgl. Seite 21). Zur Konstruktion des 2-Bit Orakels wir ähnlich vorgegangen. Als erstes ist jedoch folgender technischer Parameter zu definieren

$$\tau(n) := 34 + \log \epsilon(n)^{-1} + \log n.$$

Dieser Wert hat technische Gründe, die aber erst bei den Analysen zum Finden von Eingabeintervallen für dieses Verfahren vollständig klar werden. Ein bereits ersichtlicher Grund ist aber, dass im Folgenden nur Bitpositionen $i \leq n - \tau(n) - 1$ untersucht werden. Durch diese Beschränkung können Reduktionen modulo N besser kontrolliert werden, ebenso bleibt damit der Wert der Verzerrung $\beta_i(N)$ niedrig (siehe Seite 8). Dieser Wert wächst bei den oberen Randbits exponentiell schnell an und erschwert deren Analyse erheblich.

Die Konstruktion des 2-Bit Orakels erfolgt nun mit einer nahezu identischen Idee, wie zuvor die Konstruktion des Orakels für das unterste Bit bei Fischlin und Schnorr. Dort wurde der Wert von $[x/2]_N$ untersucht. Bei genauerem Hinsehen ergab sich, dass aus der Position dieses Wertes auf die Parität des untersten Bits geschlossen werden konnte.

Statt mit $1/2$ wird hier der Wert von x aber mit $2^{-\tau(n)}$ multipliziert. Abgesehen von den komplizierter aussehenden Termen die dabei entstehen, können einige interessante Eigenschaften gefunden werden. Diese Eigenschaften werden dabei aber im Wesentlichen modulo 2^{i+1} untersucht. Der Grund dafür ist, dass sich die Analyse der gesuchten Bits 0 und $i+1$ bei Betrachtung modulo 2^{i+1} vereinfacht. Wenn die gegebene Zahl nämlich in $i+1$ Bit lange Bitintervalle geteilt wird, sind die beiden gesuchten Bits gerade die untersten Bits zweier aufeinander folgender Intervalle.

Als Annahme seien für diese Untersuchung d Bits rechts von i bekannt. Hierbei sei d klein genug, so dass der Wert eingangs einfach „geraten“ werden kann. Dieser Wert stellt damit eine Art Initialwert für die folgenden Untersuchungen dar.

Bei dieser Analyse werden die Bitintervalle $B_l^k(x)$ als $k-l+1$ Bit lange Binärzahlen aufgefasst, mit denen nach Umwandlung in das Dezimalsystem normal weiter gerechnet werden kann. Die Umwandlung in das Dezimalsystem geschieht dabei implizit.

Durch Umformen und Auseinanderziehen ergibt sich für $[x2^{-\tau(n)}]_N$ folgender Wert:

$$[x2^{-\tau(n)}]_N = \frac{x - B_{i+1}^{i+\tau(n)}(x)2^{i+1} - B_{i-d+1}^{i}(x)2^{i-d+1} - B_0^{\tau(n)-1}(x)}{2^{\tau(n)}} \tag{3.6}$$

$$+ B_{i-d+1}^{i}(x)2^{i-d+1-\tau(n)} \tag{3.7}$$

$$+ B_{i+1}^{i+\tau(n)}(x)2^{i+1-\tau(n)} + B_0^{\tau(n)-1}(x)[2^{-\tau(n)}]_N \tag{3.8}$$

Die somit entstandenen Terme werden nun analysiert. Der Zähler des Bruches (3.6) ist aufgrund seiner Konstruktion durch $2^{\tau(n)}$ teilbar. Zudem sind die d Bits rechts von Bit i gleich Null. Bei der vorgenommenen Division durch $2^{\tau(n)}$ tritt damit keine modulare Reduktion auf. Wenn der sich ergebende Wert nun modulo 2^{i+1} betrachtet wird, ist der sich ergebende Wert sehr klein.

Der zweite Summand (3.7) ist nach Voraussetzung bereits bekannt und muss nicht bestimmt werden. Somit ergibt sich als einziger bedeutender, unbekannter Term die Summe (3.8).

Die Bestimmung dieser Bitintervalle wird die wesentliche Aufgabe im weiteren Ablauf sein. Tatsächlich ist das Ziel aber einfacher. Bei genauer Betrachtung sind nämlich die gesuchten Bits $i+1$ und 0 gerade die untersten Bits dieser Intervalle. Allein diese gilt es korrekt zu bestimmen.

Skizze des 2-Bit Orakels

Aufgrund der Länge der unbekannten Intervalle existieren genau $2^{2\tau(n)}$ Möglichkeiten, wie dieses Paar aus zwei $\tau(n)$-langen Bitintervallen zusammengesetzt sein kann. Hierbei handelt es sich, aufgrund der Wahl von $\tau(n)$, um eine polynomiell große Menge. Das beabsichtigte Ziel wird es nun sein, diese Liste so lange zu verkleinern, bis alle darin vorhandenen Intervalle gleichen Werte für die untersten Bits besitzen. Damit wären die gesuchten Bits 0 und $i+1$ gefunden.

Das Vorgehen ist es nun, zuerst eine Kandidatenliste für die Paare $(B_0^{\tau(n)-1}(x), B_{i+1}^{i+\tau(n)}(x))$ zu erstellen. In dieser Liste sind sämtliche Möglichkeiten (u_j, v_j) mit $0 \leq u_j, v_j \leq 2^{\tau(n)}$ enthalten. Die Frage ist nun, wie alle falschen Paare aus dieser Liste entfernt werden können.

Seien dazu (u_j, v_j) und (u_k, v_k) Elemente dieser Liste mit $\mathrm{lsb}(u_j) \neq \mathrm{lsb}(u_k)$ oder $\mathrm{lsb}(v_j) \neq \mathrm{lsb}(v_k)$. Mit einem passend gewähltem z' kann das Problem, welches dieser beiden Bits korrekt ist (und es kann nur eines korrekt sein), in eine äquivalente Fragestellung umgewandelt werden. Die neue Frage ist, ob $[2^{-\tau(n)}x]_N$ gleich $z' + u_1 2^{i+1-\tau(n)} + v_1[2^{-\tau(n)}]_N$ oder gleich $z' + u_2 2^{i+1-\tau(n)} + v_2[2^{-\tau(n)}]_N$ ist. Definiert man die Variablen $u := u_2 - u_1$ und $v := v_2 - v_1$ ergibt sich daraus ein Unterscheidungsproblem zwischen einem

$$z \text{ oder } z + u2^{i+1-\tau(n)} + v[2^{-\tau(n)}]_N.$$

Zur Vereinfachung wird o.E. angenommen, dass v positiv ist.

Bei genauer Betrachtung befinden wir uns nun in etwa der Ausgangslage, in der wir uns schon beim Verfahren von Fischlin und Schnorr befanden. Wieder ist das Problem gegeben, zu entscheiden, in welcher von zwei Zahlenmengen eine Zahl $[ax]_N$ liegt. In diesem Fall bedeutet das für eine Zahlenmenge $S \subset \mathbb{Z}_N$: Liegt der Wert von $[2^{-\tau(n)}x]_N$ in S oder in der Verschiebung $S + u2^{i+1-\tau(n)} + v[2^{-\tau(n)}]_N$. Falls dieser Wert in S liegen sollte, kann direkt geschlossen werden, dass (u_j, v_j) korrekt ist. Andernfalls muss (u_k, v_k) der gesuchte Wert sein.

Zur einfacheren Notation wird eine Variable $\alpha(u, v)$ für $u2^{i+1-\tau(n)} + v[2^{-\tau(n)}]_N$ eingeführt.

Definition 18. Sei N der Modulus und $\tau(n) = 34 + \log \epsilon(n)^{-1} + \log n$, sowie $0 \leq v \leq 2^{\tau(n)} - 1$ und $|u| \leq 2^{\tau(n)} - 1$. Dann ist $\alpha(u, v)$ wie folgt definiert:

$$\alpha(u, v) := u2^{i+1-\tau(n)} + v[2^{-\tau(n)}]_N$$

Es ist zu beachten, dass nur Werte von $\alpha(u, v)$ mit ungeradem u oder ungeradem v interessant sind. Andernfalls sind die unteren Bits zweier zu unterscheidender Paare nämlich gleich und es ist nicht sinnvoll ein Paar aus der Menge zu entfernt.

Ein wesentlicher Punkt, der noch nicht diskutiert wurde, ist das Aussehen der Mengen S. Die Analyse und das Finden dieser Mengen wird im weiteren Vorlauf die größte Herausforderung darstellen.

3.4.1 Struktur der Näslund-Intervalle

Damit der Vorteil des CMHNP-Orakels signifikant gemessen werden kann, sind die Mengen S so zu wählen, dass für alle Zahlen in einer Menge S das i-te Bit gleich ist. Die bisher vorgestellten Techniken funktionieren jedoch nur auf zusammenhängenden Intervallen. Dieses ist hier aber nicht mehr möglich. Der Grund ist, dass zusammenhängende Intervalle von Zahlen mit gleichem i-ten Bit durch die Länge 2^i beschränkt sind. Da 2^{i+1} vernachlässigbar klein gegenüber N ist, können Intervalle dieser Art nicht verwendet werden.

Gelöst wird dieses Problem, indem eine große Menge dieser kleinen Intervallen so gewählt wird, dass die gewählten Intervalle in einer bestimmten Weise doch noch verbunden sind. Die Größe dieser Menge sei nach oben dadurch beschränkt, dass sie nur so viele kleine Intervalle enthalten soll, bis der Gesamtwert aller Intervalle nicht mehr vernachlässigbar klein gegenüber N ist. Da die Werte der i-ten Bits aller Zahlen in dieser Menge gleich sein müssen, sind diese im Abstand von 2^{i+1} zu wählen.

Diese abstrakte Struktur kann graphisch sehr einfach ausgedrückt werden. Da immer nur Intervalle im Abstand von 2^{i+1} gewählt werden, können die Zahlen von 0 bis $N - 1$ in Zahlenintervalle der Länge je 2^{i+1} aufgeteilt werden. Eine Anordnung dieser Intervalle vertikal übereinander ergibt die Ebene aus Abbildung 3.3. Die Koordinaten für eine Zahl b sind damit auf der horizontalen Achse (I-Achse) durch $b \bmod 2^{i+1}$ gegeben, die Koordination auf der vertikalen Achse (Y-Achse) durch $b \operatorname{div} 2^{i+1}$. Dieses folgt gerade daraus, dass b geschrieben werden kann als $b = 2^{i+1} \cdot (b \operatorname{div} 2^{i+1}) + b \bmod 2^{i+1}$.

Damit stellt diese Ebene die Menge aller Zahlen 0 bis $N - 1$, zuzüglich eines kleinen Restes dar. Die graphische Interpretation für eine Menge S ist damit gerade ein kleines Rechteck auf der Ebene. Obwohl die Intervallmengen keine zusammenhängenden Intervalle bilden, ist nun durch diese Konstruktion ein örtlicher Zusammenhang gegeben.

Damit diese Ebene in den folgenden Analysen verwendet werden kann, folgt ihre formale Definition. Zuvor sind noch einige Begrifflichkeiten zu definieren.

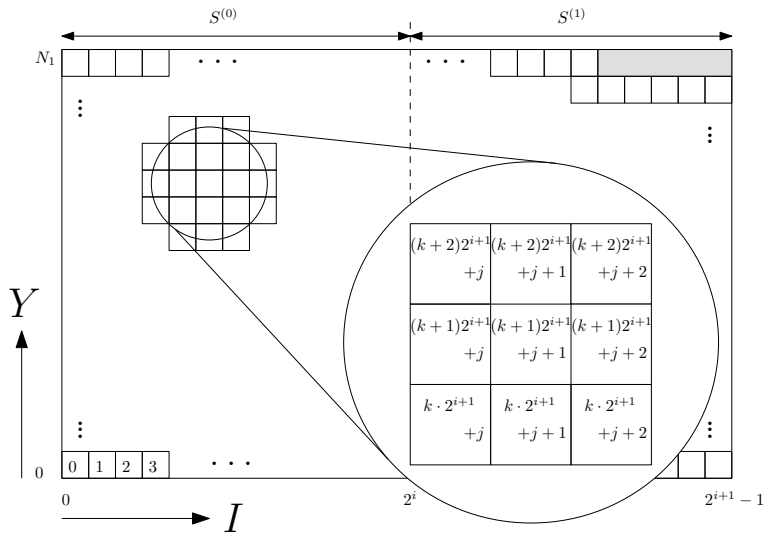

Abbildung 3.3: Darstellung der Zahlen 0 bis $N-1$ in einem Rechteck.

Da die Werte aus Z_N im Folgenden modulo 2^{i+1} betrachtet werden, genügt es nicht mehr für Abschätzungen den Wert von N zu verwenden. Es ist eine feinere Aufteilung von N in verschieden große Teile notwendig.

Definition 19. N wird geschrieben als $N = N_1 2^{i+1} + N_0$, wobei $N_0 < 2^{i+1}$ gilt. N_1 wird geschrieben als $N_1 = N_3 2^{\tau(n)} + N_2$ mit $N_2 < 2^{\tau(n)}$.

Die Abbildung von Werten aus $\mathbb{Z}_N$ auf Punkte der Ebene erfolgt durch eine Projektion π.

Definition 20. Seien $I := \mathbb{Z}_{2^{i+1}}, Y := \mathbb{Z}_{N_1+1}$. Dann ist die natürliche Projektion π wie folgt definiert:

$$\left.\begin{array}{ll} \pi: & \mathbb{Z}_N \longrightarrow I \times Y \\ & \pi(z) \longmapsto (z \bmod 2^{i+1}, \lfloor \frac{z}{2^{i+1}} \rfloor) \end{array}\right\}$$

Diese Abbildung ist injektiv und *fast* subjektiv. Das bedeutet, dass bis auf die Punkte (j, N_1) mit $j \geq N_0$ alle Punkte der Ebene ein Urbild besitzen, damit ist eine obere Schranke für die Punkte ohne Urbild gegeben.

Als nächstes ist die Ebene zu definieren, ebenso Rechtecke auf dieser Ebene, die im weiteren Verlauf die Rolle der Intervalle aus der Fischlin und Schnorr Methode übernehmen.

Definition 21. Die *Ebene* $\Pi(N, i)$ ist definiert als $\Pi(N, i) = (I \times Y) \cap \pi(\mathbb{Z}_N)$. Diese Ebene wird nach der Parität des i-ten Bits in zwei Hälften aufgeteilt. Es gelte für $b \in \{0, 1\}$

$$S^{(b)} := \{z \in \mathbb{Z}_N | \operatorname{bit}_i(z) = b\}$$

Auf dieser Ebene können sich Rechtecke befinden. Ein *Rechteck* mit Breite w und Höhe h ist dabei definiert als

$$S := \left\{\pi(z + 2^{i+1}y) | z_0 \leq z < z_0 + w, y_0 \leq y < y_0 + h\right\}.$$

Das *Maß* eines Rechtecks ist dabei gegeben durch

$$\lambda(S) = \frac{|S|}{N} = \frac{w \cdot h}{N}.$$

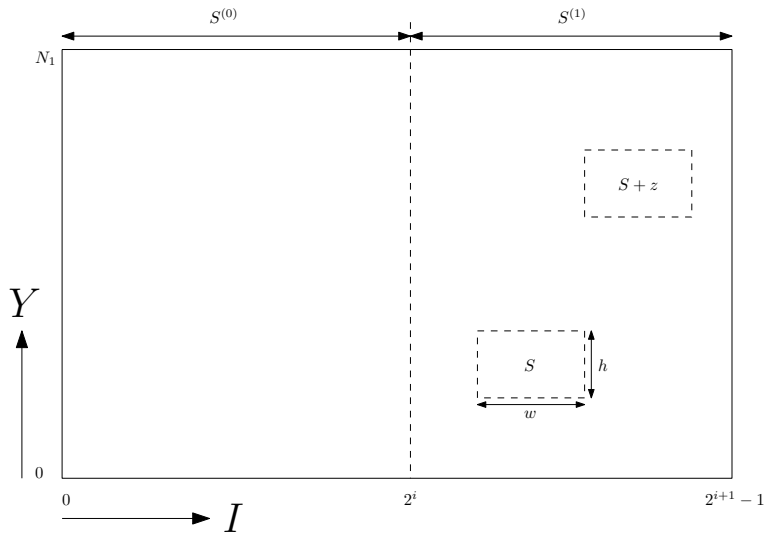

Abbildung 3.4: $\Pi(N,i)$-Ebene des Näslund-Verfahrens, ein Rechteck S, Verschiebung $S+z$.

Die *Verschiebung* eines Rechtecks S heißt z-Verschiebung und ist definiert als

$$S+z = S+(\pi_I(z),\pi_Y(z)) := \left\{(\pi_I(z'+z),\pi_Y(y'+z))|(z',y')\in S\right\}.$$

Analog zu den bereits definierten Intervallunterscheidern werden nun im Folgenden Unterscheider für die hier definierten Rechtecke verwendet. Sämtliche Begriffe wie Orakel oder Unterscheider sind sinngemäß zu übertragen.

3.4.2 Der Invertierungsalgorithmus

Nach diesen technischen Vorbereitungen ist es nun möglich, einen Invertierungsalgorithmus anzugeben. Vorausgesetzt wird lediglich die Existenz der beschriebenen Rechtecke sowie eines signifikanten Unterscheiders für diese Rechtecke.

Um noch einmal die Idee aufzugreifen: Es soll aus einem CMHNP-Orakel für ein einzelnes Bit, ein Orakel für zwei Bits konstruiert werden. Dieses neue Orakel erzeugt ein 2-Bit-Fenster, welches über sämtliche Bits geschoben wird. Damit werden die Bits des Klartextes bestimmt und der Klartext ist gefunden.

Als erster Schritt ist das 2-Bit-Orakel zu konstruieren.

Lemma 22. *Seien $0 \le v \le 2^{\tau(n)}-1$ und $|u| \le 2^{\tau(n)}-1$, wobei u oder v ungerade sind. Gegeben sei ein Rechteck $S_{u,v}$ mit signifikanter Breite $w(n)2^{i+1}$ und signifikanter Höhe $h(n)N_1$ sowie ein Unterscheider $\Delta^{\mathcal{P}}(S_{u,v}, S_{u,v}+\alpha(u,v)) \ge \zeta(n)$ mit signifikantem $\zeta(n)$.*

Definiere damit

$$d(n) := \min\left\{i, \log\zeta(n)^{-1} + \log(w(n)h(n))^{-1} + 9 + 2\tau(n) + 2\log n\right\}.$$

Dann kann ein Orakel $\mathcal{P}'$ konstruiert werden, welches bei Eingabe $j, B^{i+j}_{i-d(n)}(x), B^{j-1}_0(x)$ und y mit $\mathrm{abs}_N(x-y) \le 2^{-d(n)}N$ sowie $0 \le j \le \max\{n-i-2, i\}$, die Bits $\mathrm{bit}_{i+j+1}(x)$ und $\mathrm{bit}_j(x)$ mit einer Wahrscheinlichkeit von mindestens $1-\frac{1}{2n}$ bestimmt.

Beweis. Die Idee dieser Konstruktion ist im Wesentlichen das bereits skizzierte Vorgehen. Zuerst wird eine polynomiell große Liste mit sämtlichen Kandidaten für die Paare (u_k, v_k) erstellt. Aus dieser sollen so lange Elemente gelöscht werden, bis die Werte für die untersten Bits bei allen Elementen gleich sind. Damit ist bei jeder Iteration die Frage zu beantworten, ob ein Paar (u_1, v_1) oder ein Paar (u_2, v_2) korrekt ist.

Es wird in diese Untersuchung o.E. angenommen, dass für die Werte von i und j gilt, dass $i > d(n)$ und $j \leq \min\{i - d(n), n - (i + 1)\}$. Diese Einschränkung bedeutet, dass die beide Bits $i + j + 1$ und j unbekannt sind. Dadurch wird hier der allgemeine Fall betrachtet.

Das im 2-Bit Orakel (Algorithmus 3) umgesetzte Verfahren verfolgt dabei die dargestellte Strategie um zwischen den gegebenen beiden Paaren zu unterscheiden. Dafür ist es jedoch notwendig, nicht nur unterscheiden zu können, sondern präzise zu sagen, in welchem Rechteck sich die um $\alpha(u, v)$ verschobenen Werte befinden. Dieses kann über die Anteile der 1-Antworten gemacht werden. Da der Unterscheider einen signifikanten Vorteil besitzt, müssen sich die Anteile der 1-Antworten auf beiden Rechtecken unterscheiden.

Mit Lemma 15 können für den Entscheideralgorithmus $A^{\mathcal{P}}$ auf $S_{u,v}$ und auf $S_{u,v} + \alpha(u, v)$ Näherungswerte für die Anteile der 1-Antworten bestimmt werden. Diese Näherungen $\tilde{p}_{u,v}$ für $A^{\mathcal{P}}(S_{u,v})$ und $\tilde{p}'_{u,v}$ für $A^{\mathcal{P}}(S_{u,v} + \alpha(u, v))$ können ebenso mit Lemma 15 auf die Genauigkeit $\zeta(n)/8$ gebracht werden. Somit gilt jeweils mit einer Wahrscheinlichkeit von mindestens $1 - \frac{1}{8n}$ für die Approximationen $|\tilde{p}_{u,v} - A^{\mathcal{P}}(S_{u,v})| \leq \zeta(n)/8$ und $|\tilde{p}'_{u,v} - A^{\mathcal{P}}(S_{u,v} + \alpha(u, v))| \leq \zeta(n)/8$. O.E. gelte für diese Werte $\tilde{p}'_{u,v} > \tilde{p}_{u,v}$.

Input : $j, B^{i+j}_{i-d(n)}(x), B^{j-1}_0(x), y$
Output : $\mathrm{bit}_{i+j+1}(x), \mathrm{bit}_j(x)$
$T \leftarrow \{0,1\}^{\tau(n)} \times \{0,1\}^{\tau(n)}$ `// Erzeuge Kandidatenliste`
while $\exists (u_1, v_1), (u_2, v_2) \in T$ mit $\mathrm{lsb}(u_1) \neq \mathrm{lsb}(u_2) \wedge \mathrm{lsb}(v_1) \neq \mathrm{lsb}(v_2)$ **do**
  sortiere $(u_1, v_1), (u_2, v_2)$, so dass $v_2 \geq v_1$;
  $R \leftarrow \{\}$;
  for $k \leftarrow 1$ **to** $m(n)$ **do**
    `/* berechne` π' `unter der Annahme, dass` (u_1, v_1) `korrekt */`
    $\pi' \leftarrow$ Approximation von $\pi([r_k + 2^{-(j+\tau(n))}x]_N)$;
    if $\pi' \in S_{u,v}$ **then**
      $R \leftarrow R \cup \{r_k + 2^{-(j+\tau(n))}\}$
    end
  end
  $p \leftarrow$ Anzahl von $1-$Antworten von $\mathcal{P}$ auf R;
  if $p \leq \lambda(S_{u,v})m(n)(\tilde{p}_{u,v} + \tilde{p}'_{u,v})$ **then**
    $T \leftarrow T \backslash \{(u_2, v_2)\}$ `//` (u_2, v_2) `ist nicht korrekt`
  end
  else
    $T \leftarrow T \backslash \{(u_1, v_1)\}$ `//` (u_1, v_1) `ist nicht korrekt`
  end
end
Wähle ein $(u, v) \in T$, **return**$(\mathrm{lsb}(u), \mathrm{lsb}(v))$;

Algorithmus 3 : Orakelalgorithmus für zwei Bits

Um zu entscheiden, welches der Paare (u_1, v_1) oder (u_2, v_2) korrekt ist, wird eine weitere Messung p von 1-Antworten erstellt. Diese Messung beruht auf der Annahme, dass (u_1, v_1)

der korrekte Wert ist. Falls dieser Wert in der Nähe von $\tilde{p}$ liegt, kann geschlossen werden, dass (u_1, v_1) korrekt ist. Andernfalls ist das Paar (u_2, v_2) richtig und das andere Paar (u_1, v_1) muss falsch sein.

Um die Messung von p durchführen zu können wird abermals eine Menge R von Punkten mit Lemma 13 konstruiert. Die dort erzeugten $m(n)$ Werte $\{r_j\}$ sind zugleich mit Approximationen z_j^I und z_j^Y der Zahlen $[r_j x]_N$ sowie des Absolutwertes modulo 2^{i+1} gegeben. Es wird nun angenommen, dass das Paar (u_1, v_1) korrekt ist. Auf dieser Grundlage kann eine Approximation des Projektionswertes erstellt werden. Nur wenn dieser in dem Rechteck $S_{u,v}$ gelegen ist, wird der Punkt weiter verwendet. Die so projizierten Werte sind weiterhin paarweise unabhängig. Ebenso sind diese Punkte gleichverteilt über $S_{u,v}$, falls (u_1, v_1) korrekt ist. Andernfalls sind diese gleichverteilt über $S_{u,v} + \alpha(u, v)$.

Eine Anwendung von Lemma 15 auf die resultierende Punktemenge liefert den Anteil der 1-Antworten. Grund ist, dass durch die Approximationen z_j^I und z_j^Y der Fehler δ zur Bestimmung dieser Bits auf $\leq 1/2^{d(n)}$ beschränkt wird. Durch Anwendung von Lemma 15 kann daher eine Bestimmung von p mit Genauigkeit kleiner $\zeta(n)/4$ und Fehlerwahrscheinlichkeit kleiner $1/(4n)$ durchgeführt werden.

Mit dem gerade gezeigten Vorgehen werden nun iteriert immer wieder zwei Punkte mit unterschiedlichen Werten für die untersten Bits gewählt. Die Entfernung eines falschen Paares erfolgt dabei mit einer Fehlerwahrscheinlichkeit von maximal $\frac{1}{2n}$.

Insgesamt arbeitet der Algorithmus also mit einer Erfolgswahrscheinlichkeit von mindestens $1 - \frac{1}{2n}$. □

Der nächste Schritt ist analog zur Methode von Näslund. Dort wird ein solches 2-Bit Orakel dazu verwendet, einen Invertierungsalgorithmus zu konstruieren. Dazu wird das Orakel durch Manipulation des j-Wertes über sämtliche Bits des Klartextes geschoben.

Lemma 23. *Es seien gleiche Voraussetzungen, wie in Lemma 22 gegeben. Dann ist es möglich, in zufälliger polynomieller Zeit den Wert von x mit einer Erfolgswahrscheinlichkeit von mindestens $1/2$ zu bestimmen.*

Beweis. Mit Lemma 22 kann ein Orakel $\mathcal{P}'$ zum Entscheiden zweier Bits konstruiert werden. Darauf aufbauend wird mit Algorithmus 4 ein Invertierungsalgorithmus angegeben.

Ein kritischer Punkt bei diesem Algorithmus ist die Wahl von y sowie von z'. Bei einer polynomiellen Anzahl an Versuchen ist aber mindestens eine gute Wahl zu erwarten.

Der eigentliche Invertierungsalgorithmus beschreibt damit nur noch das Verfahren zur Umwandlung einer Binärzahl in eine Dezimalzahl. Durch Veränderung des Parameters j wird ein Bit nach dem anderen bestimmt. Je nach Wert des betrachteten Bits wird dann die zugehörige 2er-Potenz zur Summe z hinzuaddiert.

Mit n Orakelaufrufen ergibt sich ein maximaler Fehler von $n \cdot \frac{1}{2n} = \frac{1}{2}$. □

Die Schlüsselfrage für das Funktionieren dieses Verfahrens ist die Existenz der verwendeten Rechtecke, spezieller noch das Finden dieser Rechtecke und das Konstruieren von Unterscheidern auf diesen Intervallen. Die Untersuchung dieser Fragestellung wird den letzten Teil der Analyse bilden.

3.5 Existenz guter Rechtecke

Der vorherige Abschnitt hat einen Invertierungsalgorithmus nach der Idee von Näslund geliefert. Als Eingabe benötigt dieser Algorithmus lediglich einen Unterscheider sowie unterscheidbare

Input : $j, B^{i+j}_{i-d(n)}(x), B^{j-1}_0(x), y$
Output : x
„Rate“ $y \in \mathbb{Z}_N$ mit $\mathrm{abs}_N(y-x) \leq N/2^{d(n)}$;
„Rate“ $z' = B^i_{i-d(n)+1}(x)$;
$z \leftarrow 0$;
for $j \leftarrow 0$ **to** $\max\{n-(i+1), i-d(n)\}$ **do**
  $(b', b) \leftarrow \mathcal{P}'(j, z', z, y)$ `// verwende 2-Bit Orakel`
  if $i+j < n$ **then**
    $z' \leftarrow 2^{j+d(n)}b' + z'$
  end
  if $j + d(n) < i$ **then**
    $z \leftarrow 2^j b + z$
  end
  $y \leftarrow b(N+1)/2 + (y-b)/2$;
end
return $x = z'2^{i+1-d(n)} + z$

Algorithmus 4 : Invertierungsalgorithmus mit Näslund-Technik

Rechtecke.

Die Existenz unterscheidbarer Rechtecke und die Suche nach diesen bilden nun den Hauptteil des Verfahrens. In Abhängigkeit der Werte von u und v wird sich diese Suche unterschiedlich schwer gestalten. Zu beachten ist, dass durch die Werte von u und v die Verschiebung $\alpha(u, v)$ in jeder Iteration bereits eindeutig festgelegt wurde.

Da das beabsichtigte Ziel bei diesem Verfahren die Entscheidung zwischen einem falschen und einem richtigen Paar ist, wird im Folgenden immer mindestens einer der Werte von u oder v ungerade sein. Andernfalls wären die untersten Bits beider Paare gleich.

Es werden hier die folgenden beiden Fälle betrachtet, die aber sämtliche Fallunterscheidungen abdecken:

1. v gerade und u ungerade
2. v ungerade

Der erste Fall ist relativ einfach zu behandeln. Er besagt nämlich, dass zwei Alternativen mit gleichem untersten Bit vorhanden sind. Diese Alternativen unterscheiden sich jedoch im $i+1$-ten Bit.

Wie auf Seite 25 skizziert, wird eine Verschiebung um $2^{\tau(n)-1}\alpha(u, v)$ untersucht. Dieser Verschiebungswert ist bei geradem v und bei Betrachtung modulo N gerade

$$2^{\tau(n)-1}\alpha(u, v) \equiv u2^i + v' \pmod{N}.$$

Durch u ungerade wird damit bei einer um diesen Wert verschobenen Zahl das i-te Bit von 0 nach 1 oder anders herum von 1 nach 0 abgebildet. Somit benötigt der Unterscheider ein Rechteck, in dem sich der Anteil der Zahlen mit i-tem Bit gleich 1 von denen mit i-tem Bit gleich 0 signifikant unterscheidet. In Lemma 25 wird ein solches Rechteck gefunden.

Der Fall v ungerade ist jedoch aufwändiger zu behandeln. Hier liegen nun zwei Alternativen vor, die sich im untersten Bit unterscheiden. Ziel ist wieder das Finden zweier Rechtecke S und $S + k \cdot \alpha(u, v)$, auf denen sich ein Unterscheider verschieden verhält. Als Ansatz wird hierbei ein

Rechteck S genommen und sämtliche Verschiebungen dieses Rechtecks um $k \cdot \alpha(u, v)$ betrachtet. Dabei soll k aber auf eine polynomiell große Zahl beschränkt sein. Falls in dieser Menge zwei unterscheidbare Rechtecke gefunden werden sollten, wäre das Ziel bereits erreicht. Was passiert aber, wenn diese Rechtecke nicht unmittelbar gegeben sind?

Betrachten wir also den Fall, dass sich das Orakel auf allen Rechtecken $S + k\alpha(u, v)$ nahezu gleich verhält. Da das CMHNP-Orakel nach Annahme einen signifikanten Vorteil besitzt, muss eine Menge von Rechtecken $S + k\alpha(u, v)$ existieren, in der das i-te Bit mit signifikanter Erfolgswahrscheinlichkeit bestimmt werden kann.

Bei aufeinander folgenden Zahlen sind die Werte des i-ten Bits oft gleich. Da das Orakel einen signifikanten Vorteil zur Bestimmung dieses Bits besitzt, sich aber auf allen Rechtecken der Menge gleich verhält, können diese Rechtecke nicht gleichmäßig über diese Zahlen verteilt sein. Durch eine gleichmäßige Verteilung gäbe es nämlich Rechtecke, mit einer unterschiedlich großen Anzahl von Zahlen, deren i-tes Bit 1 ist. Es wird später möglich sein, dieses als eine Eigenschaft des Verhältnisses zwischen der Größe der Verschiebung und der Zahl 2^{i+1} auszudrücken. Durch Analyse dieses Verhältnisses kann geschlossen werden, ob die gesuchten Rechtecke vorliegen.

Es bleibt aber der Fall übrig, in dem die gesuchten Rechtecke nicht existieren. In diesem Fall wird es möglich sein, ein neues Orakel zu konstruieren. Mit diesem neuen Orakel können dann zwei signifikante, unterscheidbare Intervalle im Abstand $(N + 1)/2$ gefunden werden. Damit sind alle Voraussetzungen für das Verfahren von Fischlin und Schnorr erfüllt und es folgt die Invertierbarkeit.

Bevor nun mit der technischen Analyse begonnen wird, ist noch die symmetrische Differenz zu definieren. Diese gibt an, wie viele Punkte nicht im Durchschnitt zweier Mengen liegen. Im weiteren Verlauf werden solche Punkte meistens die Erfolgswahrscheinlichkeiten von Unterscheidern beeinträchtigen.

Definition 24. Für zwei Mengen S und T bezeichnet $S \triangledown T$ die symmetrische Differenz

$$S \triangledown T := (S \backslash T) \cup (T \backslash S).$$

3.5.1 Der Fall v gerade

Wie bereits oben erläutert, handelt es sich bei „v gerade“ um den einfachen Fall. Da bei einem Paar (u, v) stets davon ausgegangen wird, dass mindestens u oder v ungerade ist, gilt u ungerade. Die Existenz der geforderten Rechtecke wird nun über das eingangs skizzierte Verfahren gezeigt. Wesentliches Argument ist hierbei, dass durch die Verschiebung um $2^{\tau(n)-1}\alpha(u, v)$ die Werte des i-ten Bits von 0 auf 1 bzw. von 1 auf 0 abgebildet werden.

Lemma 25. *Falls v gerade und u ungerade ist, dann existiert ein $k \leq 2^{\tau(n)} - 1$, so dass gilt*

$$\Delta^{\mathcal{P}}\left(S^{(0)} + k\alpha(u, v), S^{(0)} + (k + 1)\alpha(u, v)\right) \geq \epsilon(n)2^{-\tau(n)}.$$

Beweis. v ist nach Voraussetzung eine gerade Zahl, somit gibt es eine ganze Zahl v' mit $v = 2v'$. Wenn man die Verschiebung $2^{\tau(n)-1}\alpha(u, v)$ modulo N betrachtet, kann diese in einer einfacheren Form geschrieben werden, nämlich

$$2^{\tau(n)-1}\alpha(u, v) \equiv u2^i + v' \pmod{N}.$$

Der zu konstruierende Unterscheider gewinnt nun seinen Vorteil daraus, dass das i-te Bit in beiden Rechtecken unterschiedlich oft den Wert 1 trägt. Dieses wird dadurch erhalten, dass die

Verschiebung um $u2^i + v'$ im Wesentlichen Zahlen aus der Menge $S^{(0)}$ nach $S^{(1)}$ abbildet und anders herum.

Negativ wird die Erfolgswahrscheinlichkeit somit gerade durch die Zahlen beeinflusst, die bei der Verschiebung eine modulare Reduktion erfahren haben. Durch die modularen Reduktionen werden diese Punkte wieder auf den gleichen Bit-Wert abgebildet. Das Maß dieser Wertemenge lässt sich durch das Maß der folgenden symmetrischen Differenz beschreiben

$$\lambda\left((S^{(0)} + 2^{\tau(n)-1}\alpha(u,v)) \triangledown S^{(1)}\right).$$

Offenbar ist dieser Wert durch die Anzahl der modularen Reduktionen beschränkt, die von $u2^i$ und v' verursacht werden. Durch die Verschiebung um mindestens 2^i wird nämlich immer ein Wert von $S^{(0)}$ nach $S^{(1)}$ abgebildet, falls keine modulare Reduktion vorliegt. Zu beachten ist, dass die Werte u und v bzw. v' kleiner $2^{\tau(n)}$ sind. Damit ist für $u2^i \bmod N$ die Wahrscheinlichkeit, dass eine modulare Reduktion durchgeführt wird, gerade $2^{\tau(n)}\frac{2^i}{N}$. Für v' beträgt diese Wahrscheinlichkeit $2^{\tau(n)}\frac{1}{2^i}$. Dieses sind die einzigen Fälle, in denen auf der I-Achse eine Reduktion modulo 2^{i+1} oder auf der Y-Achse modulo N durchgeführt wird; dadurch ergibt sich eine obere Schranke für die symmetrische Differenz.

Somit kann nun unter Verwendung der Annahmen von $\tau(n) = 34 + 5\log\epsilon(n)^{-1} + \log(n)$ sowie $i \leq n - 2\tau(n) - 1$ der Wert der symmetrische Differenz abgeschätzt werden.

$$\lambda((S^{(0)} + 2^{\tau(n)-1}\alpha(u,v)) \triangledown S^{(1)}) \leq 2^{\tau(n)} \cdot \frac{2^i}{N} + \frac{2^{\tau(n)}}{2^i} \leq \frac{\epsilon(n)}{3}.$$

Es soll nun der Erfolg beschrieben werden, mit dem ein Unterscheider bei Verwendung des gegebenen Orakels die beiden Mengen $S^{(0)}$ und $S^{(1)}$ unterscheiden kann. Bei großem i tritt dabei der Effekt der Verzerrung auf (siehe Seite 8), der den Vorteil des Orakels verkleinern kann.

$$\Delta^{\mathcal{P}}(S^{(0)}, S^{(1)}) \geq \epsilon(n) - \beta_i(N).$$

Der Wert von $\beta_i(N)$ ist aufgrund seiner Definition beschränkt durch $2^i/N$. Durch Einsetzen der Beschränkung von i ergibt sich $\beta_i(N) \leq \epsilon(n)/6$.

Eine Kombination dieser Ergebnisse liefert als untere Schranke für den Unterscheider

$$\Delta^{\mathcal{P}}((S^{(0)} + 2^{\tau(n)-1}\alpha(u,v)), S^{(0)}) \geq \frac{\epsilon(n)}{2}.$$

Durch Addition einer Teleskopsumme und Anwendung der Dreiecksungleichung ergibt sich:

$$\begin{aligned}
\frac{\epsilon(n)}{2} &\leq \Delta^{\mathcal{P}}((S^{(0)} + 2^{\tau(n)-1}\alpha(u,v)), S^{(0)}) \\
&= \left|A^{\mathcal{P}}(S^{(0)} + 2^{\tau(n)-1}\alpha(u,v)) - A^{\mathcal{P}}(S^{(0)}\right| \\
&= \left|A^{\mathcal{P}}(S^{(0)} + 2^{\tau(n)-1}\alpha(u,v)) - A^{\mathcal{P}}(S^{(0)} + 2^{\tau(n)-2}\alpha(u,v))\right| \\
&\quad + \left|A^{\mathcal{P}}(S^{(0)} + 2^{\tau(n)-2}\alpha(u,v)) - A^{\mathcal{P}}(S^{(0)} + 2^{\tau(n)-3}\alpha(u,v))\right| \\
&\quad \vdots \\
&\quad + \left|A^{\mathcal{P}}(S^{(0)} + \alpha(u,v)) - A^{\mathcal{P}}(S^{(0)})\right|
\end{aligned}$$

Die hier konstruierte Menge von Unterscheidern besitzt insgesamt eine Erfolgswahrscheinlichkeit von $\epsilon(n)/2$. Da es sich hierbei um genau $2^{\tau(n)}$ Unterscheider handelt, muss mindestens einer davon eine Erfolgswahrscheinlichkeit von größergleich $\epsilon(n)2^{-\tau(n)}$ besitzen. Daraus folgt das Lemma. □

Die hier durchgeführte Konstruktion wird im weiteren Verlauf mehrfach angewendet. Speziell die Anwendung der Teleskopsumme und der Dreiecksungleichung wird bei weiteren Anwendungen jedoch kompakter dargestellt werden.

Als ein erstes Ergebnis konnte im Fall v gerade die Existenz zweier unterscheidbarer Rechtecke nachgewiesen werden. Dieses sind gerade, für ein zu bestimmendes k, die Rechtecke $S^{(0)} + k\alpha(u, v)$ und $S^{(0)} + (k+1)\alpha(u, v)$. Für k gibt es hierbei nur polynomiell viele Möglichkeiten.

3.5.2 Der Fall v ungerade

In diesem Unterabschnitt wird der kompliziertere Fall betrachtet. Dieser erfordert einige zusätzlich Werkzeuge. Als erstes wird eine neue Verschiebung namens $\tilde{\alpha}(u, v)$ eingeführt. Ergebnis dieser Verschiebungen um $\tilde{\alpha}(u, v), 2\tilde{\alpha}(u, v), 3\tilde{\alpha}(u, v), \ldots$ ist im Wesentlichen eine Permutation der ehemals vorgenommenen Verschiebungen um Vielfache von $\alpha(u, v)$. Jedoch haben die neuen Verschiebungen um $\tilde{\alpha}(u, v)$ den wesentlichen Vorteil, dass in der Folge $k\tilde{\alpha}(u, v)$ keine modularen Reduktionen auftreten.

Anschließend wird eine Charakterisierung des Verhältnisses $\tilde{\alpha}(u, v)/2^{i+1}$ gegeben, welche die Möglichkeit einer Approximation beschreibt. Als Ergebnis wird sich damit in etwa die folgende Aussage ergeben: *Wenn es keine gute Approximation r/s von $\tilde{\alpha}(u, v)/2^{i+1}$ gibt, dann existieren zwei Rechtecke mit Abstand von $\tilde{\alpha}(u, v)$ auf denen sich das Orakel unterschiedlich verhält.*

Für den Fall, dass diese Approximation nicht existiert, ist weiter zu zeigen, dass gute Rechtecke gefunden werden können. Ebenso wird zu zeigen sein, dass bei Vorliegen dieser Approximation die Voraussetzungen für das Verfahren von Fischlin und Schnorr erfüllt werden können.

Vorbereitungen

Die bisher betrachtete Verschiebung um $\alpha(u, v)$ wurde im Wesentlichen durch die v-Komponente bestimmt. Der Wert der Verschiebung ist damit oft sehr groß, was in etwa der Größenordnung von N entspricht. Folglich kann es zu modularen Reduktionen kommen. Um diese Reduktionen zu minimieren, wird der Wert $\tilde{\alpha}(u, v)$ als Ersatz für $\alpha(u, v)$ eingeführt.

Definition 26. Für $0 < v \leq 2^{\tau(n)} - 1$, v ungerade und $|u| \leq 2^{\tau(n)-1}$ definiere ein $u' = [-uv^{-1}N]_{2^{\tau(n)}}$. Es ergibt sich die Definition

$$\tilde{\alpha}(u, v) := u' 2^{i+1-\tau(n)} + \left\lceil \frac{N}{2^{\tau(n)}} \right\rceil .$$

Damit dieser Wert an Stelle von $\alpha(u, v)$ verwendet werden kann, werden im folgenden Lemma die Eigenschaften dieses Wertes untersucht. Dabei wird gezeigt, dass es sich bei der Verschiebungen um $\tilde{\alpha}(u, v)$ im Wesentlichen um Verschiebungen um $k\alpha(u, v)$ handelt. Weiterhin wird sich herausstellen, dass die Reihenfolge der neuen Verschiebungen um $\tilde{\alpha}(u, v)$ im Grunde genommen eine Permutation der Verschiebungen um $\alpha(u, v)$ darstellt.

Lemma 27. *Sei v ungerade. Falls ein Rechteck S' mit Höhe h und Breite w existiert, so dass $\Delta^{\mathcal{P}}(S', S' + \tilde{\alpha}(u, v)) \geq \zeta(n)$ gilt, dann existiert auch ein Rechteck S gleicher Größe mit*

$$\Delta^{\mathcal{P}}\left(S, S + \alpha(u, v)\right) \geq \frac{\zeta(n)}{2^{\tau(n)}} - \frac{2}{h(n)} - \frac{2}{w(n)}.$$

Beweis. Betrachtet man $k \cdot \alpha(u,v)$ mit einer Wahl von k als $k := [-v^{-1}N]_{2^{\tau(n)}}$, so bestimmt dieses im Wesentlichen den Wert von $\tilde{\alpha}(u,v)$. Wenn man den Wert $k \cdot \alpha(u,v)$ modulo N betrachtet, kann dieser umgeschrieben werden zu:

$$\begin{aligned} k\alpha(u,v) &\equiv [-v^{-1}N]_{2^{\tau(n)}}(u2^{i+1-\tau(n)} + v[2^{-\tau(n)}]_N) \\ &\equiv u'2^{i+1-\tau(n)} + c_1 2^{\tau(n)} - N[2^{-\tau(n)}]_N + c_2 2^{\tau(n)}[2^{-\tau(n)}]_N \\ &\equiv u'2^{i+1-\tau(n)} + c_1 2^{\tau(n)} + c_2 \pmod N \end{aligned}$$

Hierbei bezeichnen c_1 und c_2 modulare Reste, die bei der Berechnung modulo $2^{\tau(n)}$ bzw. modulo N entstehen. Durch diese Modulen sind die Werte c_1 und c_2 in der Größe beschränkt. Es gilt nämlich $0 \leq c_1 \leq 2^{\tau(n)}$ und $0 \leq -N + c_2 2^{\tau(n)} \leq 2^{2\tau(n)}$.

Berechnet man nun die Differenz von $k\alpha(u,v)$ und $\tilde{\alpha}(u,v)$ modulo N erhält man

$$\begin{aligned} k\alpha(u,v) - \tilde{\alpha}(u,v) &= c_1 2^{\tau(n)} + c_2 - \left\lceil N/2^{\tau(n)} \right\rceil \bmod N \\ &= c_1 2^{\tau(n)} + c_2' \bmod N \end{aligned}$$

wobei für c_2' gelte $c_2' := c_2 - \lceil N/2^{\tau(n)} \rceil$. Durch Einsetzen in die Abschätzung für c_2, ergibt sich $0 \leq c_2' < 2^{\tau(n)}$.

Die Erfolgswahrscheinlichkeit eines Unterscheiders über S' und $S' + \alpha(u,v)$, gegenüber einem Unterscheider über S' und $S' + \tilde{\alpha}(u,v)$, variiert gerade durch die Punkte, die entweder nicht in $S' + \alpha(u,v)$ oder nicht in $S' + \tilde{\alpha}(u,v)$ liegen. Dieses ist gleichbedeutend mit dem Wert der symmetrischen Differenz. Diese wird durch die maximalen Werte von c_1 und c_2 beschränkt. Denn es gilt gerade, dass der Wert von c_1 die Verschiebung auf der Y-Achse und der Wert von c_2 die Verschiebung auf der I-Achse darstellt. Damit ergibt sich für die Anzahl der Werte in der symmetrischen Differenz

$$\left|((S' + \tilde{\alpha}(u,v)) \triangledown (S' + k\alpha(u,v))\right| \leq 2c_2 w + 2c_2 h.$$

Bei einer Menge von $w \cdot h$ Punkten pro Rechteck ergibt sich eine Erfolgswahrscheinlichkeit des Unterscheiders bei Verschiebung um $k\alpha(u,v)$ von

$$\Delta^{\mathcal{P}}\left(S', S' + k\alpha(u,v)\right) \geq \zeta(n) - \frac{2c_1}{h} - \frac{2c_2'}{w}.$$

Eine Anwendung der Dreiecksungleichung in Kombination mit einer Teleskopsumme, wie im vorherigen Lemma 25, liefert das Ergebnis. □

Es ist an dieser Stelle noch einmal der Vorteil der neuen Verschiebungen um $j\tilde{\alpha}(u,v)$ zu unterstreichen. Durch das gerade bewiesene Lemma kann von nun an die Sequenz $\{j\tilde{\alpha}(u,v)\}_{j\geq 0}$ betrachtet werden. Die Werte dieser Sequenz sind bzgl. der Y-Achse, also im Wert von π_Y, monoton steigend und es treten keine modularen Reduktionen in der Folge $\{j\tilde{\alpha}(u,v)\}_{j\geq 0}$ auf.

Bevor mit diesen neuen Verschiebungen untersucht werden kann, ob unterscheidbare Rechtecke gegeben sind, sind noch einige Definitionen notwendig. Dieses umfasst vor allem die Approximationseigenschaft, die als Kriterium verwendet wird, ob solche Rechtecke gefunden werden können.

Wenn keine gute Approximation des Verhältnisses $\tilde{\alpha}(u,v)/2^{i+1}$ vorliegt, wird dieses Verhältnis $(Q(n), \psi(n))$-Typ genannt. Hierbei sind $Q(n)$ und $\psi(n)$ speziell gewählte Parameter für die Güte dieser Approximation.

Definition 28. Eine Zahl $\chi \in \mathbb{Q}$ wird als (Q, ψ)-Typ bezeichnet, wenn für alle $r, s \in \mathbb{Z}$ mit $0 \leq s \leq Q$ und $\mathrm{ggT}(r, s) = 1$ gilt

$$\left|\chi - \frac{r}{s}\right| > \frac{1}{s^2\psi}.$$

Für die hier durchgeführten Untersuchungen sind die Werte von Q und ψ wie folgt definiert.

Definition 29. Es seien definiert:

$$Q(n) := 2^{20}\epsilon(n)^{-1} \text{ und } \psi(n) := \frac{\epsilon(n)2^{\tau(n)}}{2^{12}\log^2(Q(n))}.$$

Das Vorliegen eines $(Q(n), \psi(n))$-Typs bedeutet, dass ein χ nicht gut approximiert werden kann. Für den Wert χ wird daher im folgenden immer das Verhältnis $\tilde{\alpha}(u, v)/2^{i+1}$ betrachtet und aus dem Vorliegen oder nicht Vorliegen des Typs auf die Existenz guter Rechtecke geschlossen.

Bei den nun folgenden Analysen werden nicht mehr einzelne Rechtecke, sondern Mengen von Rechtecken analysiert. Genauer beinhalten diese Mengen sämtliche Verschiebungen eines Rechtecks um $k\tilde{\alpha}(u, v)$ auf der Ebene. Diese Mengen werden *Bahnen* genannt. Eine Einschränkung an diese Bahnen soll jedoch sein, dass sie nur Rechtecke beinhalten, für die jeweils gilt, dass alle Punkte eines Rechtecks gleiche Werte für das i-te Bit besitzen.

Dieses ergibt die folgende Definition.

Definition 30 (Bahnen)**.** Ein Rechteck heißt *geteilt*, wenn es die beiden Rechtecke $S^{(0)}$ und $S^{(1)}$ schneidet.

Es seien die Breite $w(n)$ als $w(n) := 2^{i+1}\epsilon(n)/8$ und die Höhe $h(n)$ als $h(n) := \pi(\tilde{\alpha}(u, v))$ definiert. Damit sei $S_{0,0}$ das Rechteck $[0 \ldots w(n) - 1] \times [0 \ldots h(n) - 2]$. Daraus ergeben sich mit $m(n) := \left\lfloor \frac{2^{i+1}}{w(n)} \right\rfloor$ weitere Rechtecke als

$$S_{j,k} := S_{0,0} + jw(n) + k\tilde{\alpha}(u, v),$$

für $0 \leq j \leq m(n) - 1$ und $0 \leq k \leq 2^{\tau(n)} - 2$.

Eine *Bahn* ist die Vereinigung eines Rechtecks mit der Menge aller seiner möglichen Verschiebungen um $k\tilde{\alpha}(u, v)$, die keine geteilten Rechtecke erzeugen. Genauer ist eine Bahn o_j definiert als:

$$o_j := \bigcup_k S_{j,k},$$

wobei diese Vereinigung nur über nicht geteilte Rechtecke genommen wird.

Eine graphische Darstellung dieser Bahnen ist in Abbildung 3.5 zu finden.

Approximation liegt nicht vor

Dieser Unterabschnitt beschäftigt sich mit dem Fall, dass keine gute Approximation vom Verhältnis $\tilde{\alpha}(u, v)/2^{i+1}$ vorliegt. Die formelle Bezeichnung ist dafür, dass dieses ein $(Q(n), \psi(n))$-Typ ist. Bevor mit den technischen Lemmas begonnen wird, soll der Ablauf der Untersuchung skizziert werden.

Wegen dem gegebenen Orakel für das i-te Bit muss eine Bahn auf der Ebene existieren, auf der dieses Orakel einen Vorteil besitzt. Falls sich auf dieser Bahn zwei Rechtecke befinden, die unterscheidbar sind, wäre das Ziel bereits erreicht. Deshalb wird angenommen, dass sich das Orakel auf allen Rechtecken gleich verhält.

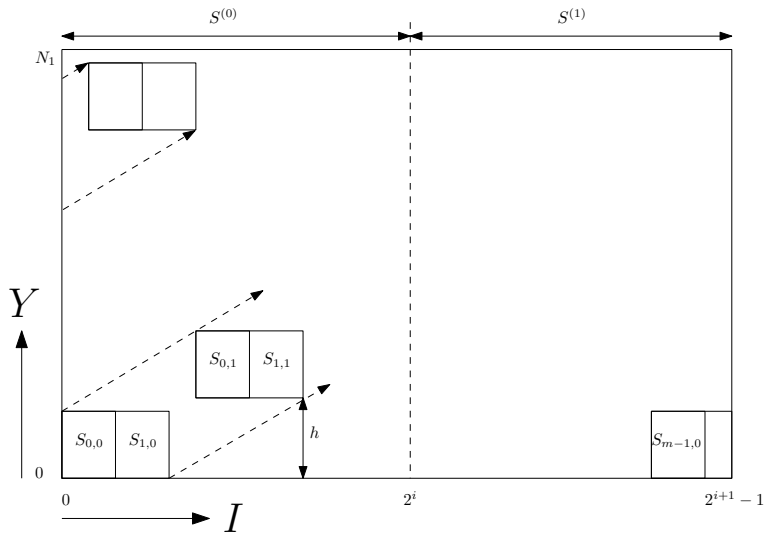

Abbildung 3.5: Rechtecke und Bahnen.

Der einzige Fall in dem dieses zutreffen kann, ist wenn die betrachtete Bahn nicht über Folgen von Zahlen mit i-tem Bit gleich 0 und i-tem Bit gleich 1 gleich geteilt wird. Weiter kann gezeigt werden, dass dann eine gute Approximation des Wertes von $\tilde{\alpha}(u, v)$ gefunden werden kann. Dieses widerspricht aber der Annahme, dass keine gute Approximation vorliegt. Somit müssen mindestens zwei Rechtecke existieren, auf denen sich das Orakel unterschiedlich verhält.

Als erstes ist nun zu zeigen, dass nahezu die gesamte Ebenen von den zuvor definierten Bahnen bedeckt wird.

Lemma 31. *Die Rechtecke $\{S_{j,k}\}$ sind paarweise disjunkt und bedecken die Ebene $\Pi(N, i)$ bis auf einen Anteil von höchstens $\epsilon(n)/2$. Das Maß aller geteilter Rechtecke ist höchstens $\epsilon(n)/4$.*

Beweis. Grund für die Einführung der Verschiebungen um $\tilde{\alpha}(u, v)$ war, dass dadurch nun keine modularen Reduktionen beachtet werden müssen. Dieses erleichtert wesentlich die hier vorgenommene Untersuchung.

Wir betrachten den maximalen Punkt auf der Ebene $\Pi(N, i)$, der von einem dieser Rechtecke eingeschlossen wird. Für die Position dieses Punktes gilt:

$$\begin{aligned}
&w(n) - 1 + (h(n) - 2)2^{i+1} + (\left\lfloor \frac{2^{i+1}}{w(n)} \right\rfloor - 1)w(n) + (2^{\tau(n)} - 2)\tilde{\alpha}(u, v) \\
&\leq (h(n) - 1)2^{i+1} + (2^{\tau(n)} - 2)(2^i + \frac{N}{2^{\tau(n)}}) \\
&< N
\end{aligned}$$

Dieser Wert ist aber kleiner als N und somit treten keine modularen Reduktionen auf.

Als nächstes ist zu untersuchen, ob die so definierten Rechtecke paarweise disjunkt sind. Seien dazu $S_{j,k}$ und $S_{j',k'}$ zwei Rechtecke, bei denen $j \neq j'$ oder $k \neq k'$ gilt.

Falls $k \neq k'$ gilt, sind diese Rechtecke disjunkt, da die Y-Werte Abbildungen der um $k\tilde{\alpha}(u, v)$ bzw. um $k'\tilde{\alpha}(u, v)$ verschobenen Rechtecke nicht übereinstimmen. Dieses folgt aus der Höhe der Rechtecke, die als $\pi_Y(\tilde{\alpha}(u, v))$ definiert wurde.

Falls $j \neq j'$ ist, können die Rechtecke ebenso keine gemeinsamen Punkte haben. Dieses liegt an der Abbildung der verschobenen I-Werte. Durch deren Definition entspricht der Unterschied zwischen $S_{j,k}$ und $S_{j+1,k}$ gerade der Verschiebung auf der I-Achse um die Breite eines Rechtecks.

Somit sind die Rechtecke $\{S_{j,k}\}$ paarweise disjunkt. Als nächstes ist das Maß der von diesen Rechtecken eingeschlossen Punkte zu bestimmen.

Die Gesamtanzahl aller durch diese Rechtecke bedeckten Punkte ist:

$$\begin{aligned}(h(n)-1)w(n)m(n)(2^{\tau(n)}-1) &\geq (h(n)w(n)-w(n))(1-2^{i-\tau(n)})N_1 \\ &\geq (2^{i+1}-w(n))(1-2^{i-\tau(n)})N_1 \\ &\geq \left(\frac{1-\epsilon(n))}{4}\right)N\end{aligned}$$

Damit wird maximal ein Anteil von $\epsilon(n)/4$ der Ebene nicht bedeckt.

Bei festem k können für die Rechtecke $S_{j,k}$ maximal zwei Rechtecke geteilt sein. Dieses folgt aus geometrischen Beobachtungen. Es gibt nämlich lediglich zwei Möglichkeiten, sowohl in $S^{(0)}$ als auch in $S^{(1)}$ zu liegen. Insgesamt ergibt sich dadurch eine maximale Anzahl von $2^{\tau(n)} \cdot 2 = 2^{1+\tau(n)}$ geteilten Rechtecken.

Die maximale Fläche dieser Rechtecke beträgt damit

$$2^{1+\tau(n)}(h(n)-1)w(n) = 2^{1+\tau(n)}(h(n)-1)\frac{2^{i+1}\epsilon(n)}{4} \leq \frac{\epsilon(n)N}{4}.$$

□

Ein wesentlicher Schritt ist es nun, eine Beziehung zwischen dem Vorliegen einer guten Approximation und dem Wert von $\tilde{\alpha}(u,v)/2^{i+1}$ herzustellen. Die Idee dahinter ist es, das Weyl'sche Gleichverteilungstheorem zu verwenden (u.a. zu finden in [KN74]). Dieses Theorem gibt an, dass für eine komplexe Zahl ξ die Dezimalstellen der Sequenz $\{j\xi\}_{j=0}^{K-1}$ für $k \to \infty$ in $[0,1]$ gleichverteilt sind. Anders formuliert erhält jedes Intervall $[a,b] \subset [0,1]$ die erwartete Anzahl an Elementen aus dieser Sequenz. Die Konvergenzrate dieser Folge bzgl. der Gleichverteilung kann nun dadurch beschrieben werden, wie gut sich der Wert von ξ mit rationalen Zahlen approximieren lässt.

Offenbar ähnelt dieses sehr der hier untersuchten Fragestellung. Darauf angepasst, wird nun die Folge $\{j\tilde{\alpha}(u,v)\}_{j=0}^{2^{\tau(n)}-1}$ betrachtet. Die Konsequenz wird sein, dass aus einer guten Approximation von $\tilde{\alpha}(u,v)/2^{i+1}$ auf eine gute Verteilung dieser Sequenz modulo 2^{i+1} geschlossen werden kann.

Ohne Beweis wird zuerst die folgende technische Abschätzung angegeben. Ein Beweis dazu kann in [HN04] gefunden werden.

Satz 32. *Sei* $0 \leq v \leq 2^{\tau(n)}-1$, *$v$ ungerade,* $|u| \leq 2^{\tau(n)}-1$. *Wenn* $\tilde{\alpha}(u,v)/2^{i+1} \in \mathbb{Q}$ *vom Typ* $(Q(n), \psi(n))$ *ist, dann gilt für alle* $0 \leq a < b < 2^{i+1}$

$$\begin{aligned}&\left|\Pr\left(j \in_U \{0,1,\ldots,2^{\tau(n)}-2\} : a \leq [j\tilde{\alpha}(u,v)]_{2^{i+1}} \leq b\right) - \frac{b-a}{2^{i+1}}\right| \\ &\qquad\qquad \leq 14\left(\frac{1}{Q(n)} + \frac{4\psi(n)\log^2 Q(n)}{2^{\tau(n)}}\right)\end{aligned}$$

Beweis. Für einen Beweis siehe [HN04] (Satz 6.20). □

Damit sind die notwendigen Vorbereitungen getroffen, um eine Beziehung zwischen dem $(Q(n), \psi(n))$-Typ und der Existenz unterscheidbarer Rechtecke herzustellen.

Lemma 33. *Sei v ungerade. Wenn die rationale Zahl $\tilde{\alpha}(u,v)/2^{i+1}$ vom Typ $(Q(n), \psi(n))$ ist, dann existiert ein Rechteck S mit Breite $2^{i+1}\epsilon(n)/8$ und Höhe mindestens $N_3 - 1$ sowie ein Unterscheider mit $\Delta^{\mathcal{P}}(S, S + \tilde{\alpha}(u,v)) \geq \frac{\epsilon(n)}{2^{\tau(n)+3}}$.*

Beweis. Es existiert ein j, so dass auf einer Bahn o_j mit einer Wahrscheinlichkeit von mindestens $\epsilon(n)/4$ das i-te Bit bestimmt werden kann. Dass eine solche Bahn existieren muss, folgt aus dem Vorteil des CMHNP-Orakels von mindestens $\epsilon(n)$. Mit Lemma 31 gilt weiter, dass für einen zufällig gewählten Punkt der Ebene die Wahrscheinlichkeit in einem geteilten Rechteck zu liegen, maximal $\epsilon(n)/4$ beträgt.

Nach Definition sind alle Rechtecke einer Bahn, also auch von o_j, nicht geteilt. Die Rechtecke aus o_j werden nun in zwei Mengen aufgeteilt, entsprechend den Werten des i-ten Bits.

$$o_{j,0} := o_j \cap S^{(0)}$$
$$o_{j,1} := o_j \cap S^{(1)}$$

Die Anzahl der Rechtecke in diesen Mengen seien beschrieben durch n_0 als die Anzahl der Rechtecke in $o_{j,0}$, und n_1 als die Anzahl der Rechtecke in $o_{j,1}$.

Ziel ist es, die Differenz von n_0 und n_1 zu bestimmen. Um Satz 32 anwenden zu können, müssen zuvor Intervalle $[a,b]$ bestimmt werden, über die dieser Satz Aussagen macht. Wenn nun ein Rechteck in $S^{(0)}$ liegt, gilt für für die linke untere Ecke dieses Rechtecks, dass dieser Punkt zwischen 0 und $2^i - w(n)$ liegen muss. Gleiches gilt für ein Rechteck in $S^{(1)}$, wobei dort dieses Intervall um 2^i verschoben ist. Dieses kann in der folgenden Abschätzung verwendet werden.

$$\begin{aligned}
|n_1 - n_0| &= (2^{\tau(n)} - 1) \Big| \Pr\left(j \in_U \{0, \ldots, 2^{\tau(n)} - 2\} : 0 \leq [j\tilde{\alpha}(u,v)]_{2^{i+1}} \leq 2^i - w(n)\right) \\
&\quad - \Pr\left(j \in_U \{0, \ldots, 2^{\tau(n)} - 2\} : 2^i \leq [j\tilde{\alpha}(u,v)]_{2^{i+1}} \leq 2^{i+1} - w(n)\right)\Big| \\
&\leq (2^{\tau(n)} - 1) \left(\left| \Pr\left(j : 0 \leq [j\tilde{\alpha}(u,v)]_{2^{i+1}} \leq 2^i - w(n)\right) - \frac{i - w(n)}{2^{i+1}} \right| \right. \\
&\quad \left. - \left| \frac{i - w(n)}{2^{i+1}} - \Pr\left(j : 2^i \leq [j\tilde{\alpha}(u,v)]_{2^{i+1}} \leq 2^{i+1} - w(n)\right)\right) \right| \Bigg) \\
&\overset{\text{Satz 32}}{\leq} 28(2^{\tau(n)} - 1) \left(\frac{1}{Q(n)} - \frac{4\psi(n) \log^2 Q(n)}{2^{\tau(n)}} \right) \\
&= 28(2^{\tau(n)} - 1) \left(\frac{\epsilon(n)}{2^9} \right) \\
&\leq \frac{2^{\tau(n)}\epsilon(n)}{16}
\end{aligned}$$

Als nächstes ist eine Abschätzung von $n_0 + n_1$ durchzuführen. Nach Konstruktion bilden $o_{j,0}$ und $o_{j,1}$ gerade eine disjunkte Zerlegung von o_j. Damit gilt

$$n_1 + n_0 = |o_{j,0}| + |o_{j,1}| = |o_j| \geq 2^{\tau(n)-1}.$$

O.E. gelte $n_1 \geq n_0$. Konstruiere eine injektive Abbildung von $o_{j,0} \longrightarrow o_{j,1}$, die jeweils zwei Rechtecke miteinander paart. Wegen $|n_1 - n_0| \leq \frac{2^{\tau(n)}\epsilon(n)}{16}$ ist maximal ein Anteil von $\epsilon(n)/16$ der Rechtecke in $o_{j,1}$ ohne Urbild.

Die Grundannahme war, dass $\mathcal{P}$ auf o_j einen Vorteil von $\epsilon(n)/4$ hat. Damit müssen nun zwei Indizes k_0, k_1 existieren, mit $S_{j,k_0} \in o_{j,0}$ und $S_{j,k_1} \in o_{j,1}$, für die $\mathcal{P}$ über $S_{j,k_0} \cup S_{j,k_1}$ einen Vorteil von $\epsilon(n)/8$ hat. Grund ist, dass mit einer maximalen Wahrscheinlichkeit von $\epsilon(n)/16$ ein Rechteck ohne Partner gewählt wird.

Da nun $S_{j,k_0} \subset S^{(0)}$ und $S_{j,k_1} \subset S^{(1)}$ gilt, existiert ein Unterscheider mit

$$\Delta^{\mathcal{P}}(S_{j,k_0}, S_{j,k_1}) \geq \frac{\epsilon(n)}{8}.$$

Eine Anwendung der Teleskopsumme und Dreiecksungleichung, wie in Lemma 25, liefert den gewünschten Unterscheider. □

Approximation liegt vor – guter Fall

Bei dieser Untersuchung sind unter der Voraussetzung Rechtecke zu finden, dass eine gute Approximation existiert. Dabei soll *gut* sowohl den Wert der Approximationsgüte unterstreichen, als auch eine Bedingung an diesen Approximationswert stellen. Diese Bedingung ist nämlich, dass die Approximation durch den Bruch r/s gegeben ist, mit $r, s \in \mathbb{Z}$ sowie s gerade und r ungerade. Spezieller noch sollen r und s teilerfremd sein.

Auch in diesem Fall können wieder gute Rechtecke gefunden werden.

Lemma 34. *Sei v ungerade. Es seien zwei teilerfremde Zahlen r und s gegeben mit $0 < s \leq Q(n)$ und s gerade, und es gelte*

$$\left| \frac{\tilde{\alpha}(u,v)}{2^{i+1}} - \frac{r}{s} \right| \leq \frac{1}{s^2 \psi(n)}.$$

Dann gibt es ein k mit $k \leq s$, so dass ein Unterscheider existiert mit

$$\Delta^{\mathcal{P}}\left(S^{(0)} + k\tilde{\alpha}(u,v), S^{(0)} + (k+1)\tilde{\alpha}(u,v)\right) \geq \frac{\epsilon(n)}{2s}.$$

Beweis. Ähnlich dem Beweis von Lemma 25 wird nun ein Unterscheider konstruiert, der die Existenz des gesuchten Unterscheiders impliziert.

Da s gerade ist, kann es geschrieben werden als $s = 2s'$. Es wird nun $s'\tilde{\alpha}(u,v)$ betrachtet. Aus $\mathrm{ggT}(r,s) = 1$ und s gerade folgt r ungerade. Damit liefert eine Umformung der Voraussetzung die Abschätzung:

$$\begin{aligned}
\frac{2^{i+1}}{s\psi(n)} &\geq \left| s\tilde{\alpha}(u,v) - r2^{i+1} \right| \\
&\geq \left| \pi_I(s'\tilde{\alpha}(u,v)) - \pi_I(r2^i) \right| \\
&= \left| \pi_I(s'\tilde{\alpha}(u,v)) - 2^i \right|
\end{aligned}$$

Ebenso kann direkt aus den Definitionen von $\tilde{\alpha}(u,v)$ und von s die Abschätzung erfolgen

$$\left| s'\tilde{\alpha}(u,v) \right| \leq \frac{Q(n)}{2}\tilde{\alpha}(u,v) \leq Q(n)N/2^{\tau(n)}.$$

Wieder interessiert das Maß der symmetrischen Differenz von $S^{(0)} + s'\tilde{\alpha}(u,v)$ und $S^{(1)}$ als wesentliche Beschränkung für den zu konstruierenden Unterscheider. Die beiden vorherigen Abschätzung sind gerade die Fehlerterme der Verschiebungen auf der I- und Y-Achse.

$$\lambda\left(S^{(0)} + s'\tilde{\alpha}(u,v) \nabla S^{(1)}\right) \leq \frac{2Q(n)}{2^{\tau(n)}} + 2\frac{1}{s\psi(n)}$$

Durch Einsetzen der Werte von $Q(n)$, $\psi(n)$ und $\tau(n)$, ergibt sich dafür eine obere Schranke von $\epsilon(n)/3$. Diese Beschränkung gilt aufgrund der Wahl von $\epsilon(n)$ und $\tau(n)$.

Die Verzerrung von $S^{(0)}$ und $S^{(1)}$ ist, wie schon in Lemma 25 erläutert wurde, beschränkt durch

$$\Delta^{\mathcal{P}}\left(S^{(0)}, S^{(1)}\right) \geq \epsilon(n) - \beta_i(N) \geq \epsilon(n) - \epsilon(n)/6.$$

Insgesamt kann daher für den konstruierten Unterscheider geschlossen werden

$$\Delta^{\mathcal{P}}\left((S^{(0)} + s'\tilde{\alpha}(u,v))\nabla S^{(0)}\right) \geq \frac{\epsilon(n)}{3}.$$

Das Umschreiben dieses Ergebnisses, sowie die Anwendung der Dreiecksungleichung liefern nun das Ergebnis. □

Approximation liegt vor – schlechter Fall

Nachdem in sämtlichen anderen Fällen die Existenz guter Rechtecke gezeigt werden konnte, schlagen diese Techniken nun fehl. Der Grund ist, dass sich das Orakel bei jeder Bahn auf sämtlichen darin enthaltenen Rechtecken gleich verhält. Durch ein anderes Verfahren kann aber zumindest gezeigt werden, dass aus dem gegebenen CMHNP-Orakel für das i-te Bit ein neues Orakel konstruiert werden kann, für welches zwei unterscheidbare Intervalle in Abstand $(N+1)/2$ existieren. Diese Intervalle und ein neu konstruierter Unterscheider erfüllen damit sämtliche Voraussetzungen für das Verfahren von Fischlin und Schnorr.

Der Beweis zu dem folgenden Lemma wird hier nur skizziert. Für die konkrete Beweisführung wird auf [HN04] verwiesen.

Lemma 35. *Seien ganze Zahlen u, v, r, s mit $0 < v \leq 2^{\tau(n)} - 1$, v ungerade, $|u| \leq 2^{\tau(n)} - 1$, $0 < s \leq Q(n)$, $\mathrm{ggT}(r,s) = 1$ und s ungerade gegeben. Und es gelte*

$$\left|\frac{\tilde{\alpha}(u,v)}{2^{i+1}} - \frac{r}{s}\right| \leq \frac{1}{s^2\psi(n)}$$

Zudem gelte für alle Rechtecke S mit einer Höhe von mindestens $sN_1 2^{-\tau(n)}$ und Breite von mindestens $2^{i+1}\epsilon(n)/(30s)$, dass ein Unterscheider existiert mit

$$\Delta^{\mathcal{P}}(S, S + \tilde{\alpha}(u,v)) \leq \epsilon(n) 2^{-(\tau(n)+3)}.$$

Sofern kein echter Teiler von N durch Verwendung von $\mathcal{P}$ gefunden werden kann, kann in polynomieller Zeit ein Orakel $\mathcal{P}'$ konstruiert und ein Intervall J der Länge mindestens $N\epsilon(n)/32$ gefunden werden, so dass $\Delta^{\mathcal{P}'}(J, J + (N+1)/2) \geq \epsilon(n)/8$ gilt.

Beweis. Der Beweis dieses Lemmas ist in [HN04] (Lemma 6.22) zu finden. □

Die für diesen Beweis verwendete Konstruktion ähnelt dem Vorgehen von Ben-Or, Chor und Shamir in [BOCS83]. In dieser Arbeit konstruierten Ben-Or et al. aus einem CMHNP-Orakel $\mathcal{P}$ für ein i-tes Bit durch $\mathcal{P}'(a) := \mathcal{P}([N_1^{-1}a]_N)$ ein neues Orakel. Mit dieser Konstruktion werden gerade Intervalle im Abstand von 2^i auf Intervalle im Abstand von $\frac{N+1}{2}$ abgebildet.

Eine ähnliche Konstruktion verfolgten nun auch Håstad und Näslund in [HN04]. Sie konnten zuerst zeigen, dass es eine Bahn gibt, auf der sich das Orakel signifikant anders verhält, als auf der Verschiebung dieser Bahn um 2^i. Durch die Verwendung des neu konstruierten Orakels ergab sich damit aber sofort ein Unterscheider für zwei Intervalle im Abstand von $(N+1)/2$. Dieses

erfüllt die Voraussetzungen für das Verfahren von Fischlin und Schnorr [FS97] und liefert damit den Invertierungsalgorithmus.

Die hier vorgenommene Konstruktion kann jedoch nicht allgemein für die anderen Fälle angewendet werden. Voraussetzung für das Gelingen ist nämlich, dass sich das Orakel auf allen Rechtecken einer Bahn gleich verhält. Erst dadurch kann gezeigt werden, dass eine Bahn existiert, bei der sich das Orakelverhalten bei einer Verschiebung um 2^i signifikant ändert.

3.6 Gesamtergebnis

Auf Grundlage der vorgestellten Techniken kann nun die Sicherheit aller RSA-Bits gezeigt werden. Der Beweis dazu fällt sehr kurz aus, da bereits sämtliche Fälle zuvor behandelt wurden. Vor Angabe des Beweises soll aber dieses Ineinandergreifen der gezeigten Ergebnisse verdeutlicht werden.

Gegeben ist zu Anfang ein CMHNP-Orakel für das i-te Klartextbit eines RSA-verschlüsselten Textes. Bekannt sind der öffentlichen Schlüssel (e, N) sowie der Chiffretext. Alle Klartexte befinden sich damit im Intervall $0, \ldots, N-1$. Abstrakter wird dieses Intervall als eine Ebene aufgefasst, auf der Rechtecke und Verschiebungen dieser Rechtecke untersucht werden können.

Mit den Techniken von Fischlin, Schnorr und Näslund kann dann gezeigt werden, dass bereits ein Unterscheider sowie zwei unterscheidbare Rechtecke genügen, um einen Invertierungsalgorithmus zu konstruieren. Der konstruierte Algorithmus arbeitet in polynomieller Zeit und kann mit einer Erfolgswahrscheinlichkeit von $1/2$ den Klartext bestimmt.

Die damit auftretende Frage ist die nach der Existenz dieser Rechtecke. Diese Existenzfrage wird mit einer vollständigen Fallunterscheidung untersucht. In genau einem Fall kann dabei jedoch nicht die Existenz der gesuchten Rechtecke gezeigt werden. Allerdings gelingt es dort, die Voraussetzungen für das Verfahren von Fischlin und Schnorr zu schaffen.

Der folgende Satz fasst diese Erkenntnisse zusammen.

Satz 36. *Es existieren Konstanten c_1 und c_2, so dass für ein $\epsilon(n)$-CMHNP-Orakel mit signifikantem $\epsilon(n)$ für das i-te Klartextbit gilt, $0 \leq i \leq n - c_1 \log \epsilon(n)^{-1} - c_2 \log n$.*

Dann kann in polynomieller Zeit mit einer Wahrscheinlichkeit von mindestens $1/2$ eine polynomielle Liste von Elementen aus $\mathbb{Z}_N$ erstellte werden, die den Klartext x enthält, oder ein echter Teiler von N konnte gefunden werden.

Beweis. Falls die Annahmen aus Lemma 35 gelten, kann bei Verwendung von Lemma 17 mit einer Wahrscheinlichkeit von $1/2$ der Klartext berechnet werden.

Gilt diese Annahme nicht, so können die benötigten Rechtecke für Lemma 23 gefunden werden. Dieses wird einmal durch Lemma 25 bei Vorliegen der dort benötigten Voraussetzungen garantiert, in allen anderen Fällen durch Lemma 27 sowie die darauf aufbauenden Lemmas 33 und 34. Bei Verwendung der dort gefundenen Rechtecke liefert 23 den Klartext mit einer Erfolgswahrscheinlichkeit von $1/2$. □

Die Aussage dieses Satzes ist es also, dass, bei Verwendung eines CMHNP-Orakels für ein i-tes Bit mit signifikanter Erfolgswahrscheinlichkeit, der vollständige Klartext mit einer Erfolgswahrscheinlichkeit von $1/2$ berechnet werden kann. Dieses geschieht in polynomieller Zeit.

Ein analoges Ergebnis für äußere Bits liefern Alexi et al. in [ACGS88]. Dort wird gezeigt, dass bei Vorliegen eines CMHNP-Orakels für ein i-tes Bit, mit $i \in \mathcal{O}(\log n)$ oder $i \in \Omega(n - \log n)$, ein polynomieller Invertierungsalgorithmus konstruiert werden kann

Unter der Annahme, dass das RSA-Verfahren sicher ist, kann ein solcher Invertierungsalgorithmus nicht existieren. Es folgt daher, dass kein CMHNP-Orakel mit signifikanter Erfolgwahrscheinlichkeit für irgendein Bit existieren kann.

Anders ausgedrückt sind alle einzelnen Bits des Klartextes ebenso sicher wie der gesamte Klartext.

4 Untersuchungen zu zuverlässigen Orakeln

Das Verfahren von Alexi et al. kann die Sicherheit des untersten RSA-Bits zeigen. Wenn die Sicherheit dieses Bits aber unter dem Aspekt untersucht wird, ob es zuverlässig bestimmt werden kann, vereinfacht sich diese Untersuchung um ein Vielfaches. Die logische Frage ist: Gilt dieser Effekt auch für weitere Bits?

Das Ziel dieses Kapitels wird es nun sein, kurz das Verfahren darzustellen, wie mit einem zuverlässigen Orakel die Sicherheit des untersten RSA-Bits sehr einfach gezeigt werden kann. Speziell soll erläutert werden, warum diese Vereinfachungen an diesen Stellen möglich sind.

Als nächstes soll die Frage der Vereinfachung im Verfahren von Håstad und Näslund untersucht werden. Ergebnis wird dabei sein, dass aufgrund der zahlreichen Fallunterscheidungen, die bei diesem Beweis notwendig sind, kaum Möglichkeiten einer Vereinfachung vorliegen.

4.1 Zuverlässige Bestimmung des untersten Bits

Mit einem sehr einfachen Beweis kann gezeigt werden, dass es nicht möglich ist, das unterste Klartextbit in polynomieller Zeit zu bestimmen. Das hier vorgestellte Verfahren geht auf Goldwasser, Micali und Tong in [GMT82] zurück.

Gegeben sei ein CMHNP-Orakel für das unterste Bit, welches mit einer Erfolgswahrscheinlichkeit von 1 arbeitet. Mit diesem Orakel kann nun der Algorithmus half konstruiert werden, welcher die Grundlage für den im Folgenden vorgestellten Invertierungsalgorithmus bildet. Als erstes werden die gewünschten Eigenschaften von half angegeben.

Definition 37. Sei $y = E_N(x)$ der gegebene Chiffretext. Ebenso sei (N, e) der bekannte öffentliche Schlüssel. Dann ist $\mathrm{half}(y)$ wie folgt definiert:

$$\mathrm{half}(y) = \begin{cases} 0, & \text{falls gilt } 0 \leq x < N/2 \\ 1, & \text{falls gilt } N/2 < x \leq N-1 \end{cases}$$

Da N nach Wahl seiner Faktoren ungerade und x eine ganze Zahl ist, tritt der Fall $x = N/2$ nicht auf.

Es kann nun gezeigt werden, dass beide Orakel half sowie das CMHNP-Orakel polynomiell äquivalent sind und durch einfache algebraische Umformungen ineinander umgeformt werden können. Da aber das CMHNP-Orakel gegeben ist, wird es genügen zu zeigen, wie das CMHNP-Orakel verwendet werden kann, um den Wert von half zu bestimmen.

Lemma 38. *Sei $\mathcal{P}$ ein CMHNP-Orakel für das unterste Bit des Klartextes x. Es bezeichne $y := E_N(x)$ den Chiffretext. Dann kann $\mathrm{half}(y)$ berechnet werden als*

$$\mathrm{half}(y) = \mathcal{P}\left(y \cdot E_N(2) \bmod N\right).$$

Beweis. Zuerst wird eine einfache Umformulierung von $y \cdot E_N(2) \bmod N$ betrachtet. An dieser Stelle ist noch einmal darauf hinzuweisen, dass das RSA-Verfahren multiplikativ ist. Es kann

also aus den Chiffretexten zweier Zahlen, der Chriffretext des Produktes dieser beiden Zahlen berechnet werden. Dieses wird hier nun benötigt.

$$\begin{aligned}\mathcal{P}\left(y \cdot E_N(2) \bmod N\right) &= \mathcal{P}\left(x^e 2^e \bmod N\right)\\ &= \mathcal{P}\left((2x)^e \bmod N\right)\\ &= \mathcal{P}\left(E_N(2x)\right)\end{aligned}$$

Es ergeben sich zwei Fälle, die betrachtet werden müssen.

$\mathrm{half}(y) = 0$: In diesem Fall befindet sich x im Intervall $0 \leq x < N/2$. Daher ist $2x \bmod N$ eine gerade Zahl und $\mathcal{P}$ liefert dafür den Wert $\mathcal{P}(yE_N(2) \bmod N) = \mathcal{P}(E_N(2x)) = 0$ zurück.

$\mathrm{half}(y) = 1$: Nun befindet sich x im Intervall $N/2 < x \leq N - 1$. Die Zahl $2x$ ist gerade und befindet sich im Intervall $N < 2x < 2N$. Mit der Wahl von p und q muss der Wert von N ungerade sein. Daher liefert die Bestimmung von $2x \bmod N$ eine ungerade Zahl. Anders ausgedrückt liefert $\mathcal{P}$ bei dieser Eingabe $\mathcal{P}(yE_N(2) \bmod N) = \mathcal{P}(E_N(2x)) = 1$.

Dieses liefert die Behauptung. □

Somit kann half synonym für das CMHNP-Orakel verwendet werden. Durch eine binäre Suche ergibt sich damit direkt der Invertierungsalgorithmus.

```
Input : N, e, y, n
Output : x
for i ← 0 to n do
    h_i ← half(y) ;                 // Bestimmung aller Werte half(E_N(x2^i))
    y ← y · 2^e mod N;
end
low ← 0; high ← N;
for i ← 0 to n do
    middle ← (low + high)/2 ;       // Durchführung der binären Suche
    if h_i = 1 then low ← middle;
    else high ← middle
end
return(⌊high⌋)
```

Algorithmus 5 : Invertierungsalgorithmus mit zuverlässigem Orakel.

Lemma 39. *Mit Algorithmus 5 kann unter Verwendung eines zuverlässigen CMHNP-Orakels für das unterste Klartextbit, in polynomieller Zeit der Klartext bestimmt werden.*

Beweis. In der ersten For-Schleife werden die Werte für alle h_i, $i = 0 \ldots n-1$ gesetzt als $h_i := \mathrm{half}(y \cdot E_N(2)^i \bmod N)$. Dieses ist gleichbedeutend mit $h_i = \mathrm{half}((x \cdot 2^i)^e \bmod N) = \mathrm{half}(E_N(x2^i))$.

Die zweite For-Schleife ist lediglich eine binäre Suche. Durch die bereits berechneten Werte für $\mathrm{half}(E_N(x2^i))$ kann dort der genaue Wert bestimmt werden. Die Anzahl der Iterationen ist zudem auf $n+1$ beschränkt, da die binäre Suche nach $\lceil \log N \rceil$ den richtigen Wert aus N Werten gefunden haben muss. □

Verglichen mit dem Verfahren von Alexi et al. stellt dieses Verfahren eine erhebliche Vereinfachung dar. Diese Vereinfachung entsteht dabei allein aus der stärkeren Voraussetzungen eines zuverlässigen CMHNP-Orakels. Die sich ergebende Frage ist nun die, ob dieser Effekt nur beim untersten Bit oder auch bei weiteren Bits auftritt.

4.2 Vereinfachungen beim Håstad- und Näslund-Ansatz

Bevor nun mögliche Vereinfachungen beim Verfahren von Håstad und Näslund untersucht werden, soll noch einmal das Verfahren von Goldwasser, Micali und Tong gegenüber dem Verfahren von Alexi et al. betrachtet werden. Beim Vergleich dieser beiden Verfahren in Hinblick auf ihre Vereinfachungen fallen zwei wesentliche Punkte auf.

1. Beim Verfahren von Alexi et al. wird aus einem gegebenen CMHNP-Orakel ein neues Orakel für das unterste Klartextbit konstruiert. Diese Konstruktion ist notwendig, da das neue Orakel eine erheblich höhere Erfolgswahrscheinlichkeit liefert, durch die die Anzahl der Iterationen des Algorithmus auf eine polynomielle Anzahl beschränkt wird.

2. Für das Verfahren von Alexi et al. werden paarweise unabhängige, gleichverteilte Zufallswerte benötigt, die auch die Eigenschaften aus Lemma 13 erfüllen. Erst durch sie ergibt sich bei iteriertem Ausführen des Algorithmus die hohe Erfolgswahrscheinlichkeit, den Klartext zu bestimmen.

Diese Vereinfachungen können aber nicht ohne Weiteres beim Verfahren von Håstad und Näslund durchgeführt werden. Dieses ergibt sich aus der Konsequenz, dass dort das gegebene Orakel zur Konstruktion eines Unterscheiders verwendet wird. Sämtliche weiteren Analysen bei diesem Verfahren laufen damit über Unterscheider und das ursprüngliche Orakel tritt nur noch selten in Erscheinung.

Aus diesem Grund ist beim Verfahren von Håstad und Näslund die Qualität des Unterscheiders wesentlich für die Laufzeit der Verfahren und ebenso für mögliche Vereinfachungen. Natürlich ergibt sich bei sämtlichen in diesem Beweis konstruierten Unterscheidern eine Verbesserung des Erfolges um den Faktor $\epsilon(n)^{-1}$. Dieses stellt gegenüber den vorherigen Werten eine beeindruckende Verbesserung der Qualität dar. Dennoch gibt es in diesem Beweis keinen Unterscheider, der allein durch Verwendung eines zuverlässigen Orakels selber zuverlässig arbeiten würde. Das Problem ist nämlich die Stuktur der Rechtecke und ebenso die Größe der Verschiebungen.

Aber einmal angenommen, es wäre ein zuverlässiger Unterscheider gegeben. So wären es im 2-Bit Algorithmus (siehe Seite 30) möglich, immense Vereinfachungen durchzuführen. Dieses umfasst vor allem die dann unnötige Erzeugung einer großen Menge an Zufallszahlen sowie den Wegfall der Iterationen in diesem Algorithmus. Die Ausführungszeit würde sich damit auf eine konstante Zeit reduzieren. Dennoch würden weiterhin noch die Fallunterscheidungen für die Paare (u, v) erhalten bleiben. Trotz des Wegfalls der Iterationen ist diese Entscheidung zwischen zwei Paaren weiter durchzuführen. Damit würde der wesentliche Teil des Beweises erhalten bleiben.

Es kann nun aber leider nicht aus der Verwendung eines zuverlässigen Orakels geschlossen werden, dass auch ein zuverlässiger Unterscheider für alle Fälle existiert. Deshalb kann nur von einer verbesserten Erfolgswahrscheinlichkeit der Unterscheider ausgegangen werden. Immerhin verursachen die verbesserten Wahrscheinlichkeiten geringer Laufzeiten der Algorithmen. Diese Laufzeitverbesserung ergibt sich direkt aus der Definition von $\tau(n)$ als $\tau(n) = 34 + \log n + \log \epsilon(n)^{-1}$. Der neue Wert von $\tau(n)$ ist damit $\tau(n) = 34 + \log n$. Dadurch verkürzt sich nun im

2-Bit Algorithmus die Kandidatenliste, die als $2^2\tau(n)$-lang definiert wurde. Auswirkungen für den Beweis ergeben sich aus dieser Laufzeitverbesserung aber keine.

4.3 Fazit

In dieser Arbeit wurde eine ausführliche Darstellung des Sicherheitsbeweises für alle inneren RSA-Bits von Håstad und Näslund gegeben. Unter Einbeziehung der in Kapitel 2 skizzierten Methode von Alexi et al. ergibt dieses die Sicherheit aller einzelnen RSA-Bits.

Unter dem Gesichtspunkt der Vereinfachung des Verfahrens, bei Verwendung eines zuverlässigen Orakels, ergab sich eine negative Antwort. Diese Antwort ist aber nur im Zusammenhang mit dem hier untersuchten Beweis zu sehen. Eine offene Frage bleibt weiterhin, ob möglicherweise ein einfacherer Beweis für die Sicherheit der inneren RSA-Bits als der von Håstad und Näslund existiert. Ebenso würde sich dann auch für dieses Verfahren die Frage stellen, ob eine Vereinfachung möglich wäre.

Literaturverzeichnis

[ACGS88] ALEXI, W. B., BENNY CHOR, ODED GOLDREICH und CLAUS-PETER SCHNORR: *RSA and Rabin functions: certain parts are as hard as the whole*. SIAM Journal on Computing, 17(2):194–209, 1988.

[BK83] BRENT, RICHARD P. und H. KUNG: *Systolic VLSI arrays for linear time gcd computation*. In: F. ANCEAU, E. J. AAS (Herausgeber): *Proceedings of the International Conference on Very Large Scale Integration*, Seiten 145–154, Amsterdam, 1983. IFIP, Elsevier Science Publishers.

[BOCS83] BEN-OR, MICHAEL, BENNY CHOR und ADI SHAMIR: *On the cryptographic security of single RSA bits*. In: *STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing*, Seiten 421–430, New York, NY, USA, 1983. ACM Press.

[BV96] BONEH, DAN und RAMARATHNAM VENKATESAN: *Hardness of computing the most significant bits of secret keys in Diffie-Hellmann and related schemes*. In: KOBLITZ, N. (Herausgeber): *Advances in Cryptology - CRYPTO '96*, Band 1109 der Reihe *Lecture Notes in Computer Science*, Seiten 129–142, Santa Barbara, Calif., August 1996. Springer-Verlag.

[BV98] BONEH, DAN und RAMARATHNAM VENKATESAN: *Breaking RSA May Be Easier Than Factoring*. In: *Eurocrypt '98*, Band 1233 der Reihe *Lecture Notes in Computer Science*, Seiten 59–71. Springer-Verlag, 1998.

[CG85] CHOR, BENNY und ODED GOLDREICH: *RSA/ Rabin least significant bits are* $1/2 + 1/\operatorname{poly}(\log N)$ *secure*. In: BLAKLEY, G.R. und DAVID CHAUM (Herausgeber): *Advances in Cryptology: Proceedings of CRYPTO '84*, Nummer 196 in *Lecture Notes in Computer Science*, Seiten 303–313, University of California, Santa Barbara, 1985. Springer-Verlag.

[FS97] FISCHLIN, ROGER und CLAUS-PETER SCHNORR: *Stronger Security Proofs for RSA and Rabin Bits*. In: *Advances in Cryptology - Eurocrypt '97*, Band 1233 der Reihe *Lecture Notes in Computer Science*, Seiten 267–279. Springer-Verlag, 1997.

[GMT82] GOLDWASSER, SHAFI, SILVIO MICALI und PO TONG: *Why and How to Establish a Private Code on a Public Network*. In: *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS'82)*, Seiten 134–144, Chicago, Illinois, Oktober 1982. IEEE.

[HN98] HÅSTAD, JOHAN und MATS NÄSLUND: *The Security of Individual RSA Bits*. In: *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, Seiten 510–519, Palo Alto, CA, November 1998. IEEE.

[HN04] HÅSTAD, JOHAN und MATS NÄSLUND: *The Security of All RSA and Discrete Log Bits*. Journal of the ACM, 51(2):187–230, 2004.

[KN74] KUIPERS, LAUWERENS und HARALD NIEDERREITER: *Uniform Distributions on Sequences*. Pure & Applied Mathematics. Wiley, New York, 1. Auflage, 1974.

[Knu98] KNUTH, DONALD: *The Art of Computer Programming*, Band 2. Addison-Wesley, 3. Auflage, 1998. Seite 342.

[LJMP90] LENSTRA, ARJEN K., HENDRIK W. LENSTRA JR., MARK S. MANASSE und J. M. POLLARD: *The Number Field Sieve*. In: *ACM Symposium on Theory of Computing*, Seiten 564–572, 1990.

[Nä96] NÄSLUND, MATS: *All Bits in $ax + b$ mod p are Hard*. In: KOBLITZ, N. (Herausgeber): *Advances in Cryptology - CRYPTO '96*, Band 1109 der Reihe *Lecture Notes in Computer Science*, Seiten 114–128, Santa Barbara, Calif., August 1996. Springer-Verlag.

[RSA78] RIVEST, RONALD L., ADI SHAMIR und LEONARD ADLEMAN: *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Communications of the ACM, 21(2):120–126, 1978.

[VV84] VAZIRANI, UMESH V. und VIJAY V. VAZIRANI: *RSA Bits are .732 + ϵ Secure*. In: CHAUM, DAVID (Herausgeber): *Proceedings of CRYPTO '83*, Advances in Cryptology, Seiten 369–375, University of California, Santa Barbara, 1984. Plenum Press.