fhwedel
UNIVERSITY OF APPLIED SCIENCES

Seminar IT-Sicherheit

Angriffe auf existierende Algorithmen durch Quanten-Computer II

Björn Philip Ridder | ite102659

27. Juni 2017

Betreuer: Prof. Dr. Gerd Beuster
Fachhochschule Wedel
Feldstraße 143
22880 Wedel

I think I can safely say that nobody understands quantum mechanics.

Richard Feynman

Zusammenfassung

Im Sommersemester 2017 fand an der Fachhochschule Wedel das Seminar im Fachbereich IT-Sicherheit mit dem Themengebiet „Quanten-Computer und (Post-)Quanten-Kryptographie“ statt. Der leitende Dozent dieses Seminars war Prof. Dr. Gerd Beuster. Interessierte Studenten sollten sich im Zuge dieses Seminars mit einem ausgewählten Thema beschäftigen und abschließend die Ergebnisse ihrer Recherche im Rahmen eines Vortrages mit dazugehöriger Ausarbeitung präsentieren.

Die vorliegende Ausarbeitung behandelt das Thema „Angriffe auf existierende Algorithmen durch Quanten-Computer“, insbesondere den Grover-Algorithmus und dessen Auswirkungen auf die Sicherheit aktueller symmetrischer Verschlüsselungsverfahren wie beispielsweise AES.

Inhaltsverzeichnis

Abbildungsverzeichnis **IV**

Tabellenverzeichnis **V**

1 Einleitung **1**

1.1 Berechnungsprobleme in der Informatik 1

1.2 Die unstrukturierte Suche 1

2 Übersicht über den Grover-Algorithmus **3**

2.1 Allgemeiner Ansatz für Quanten-Algorithmen 3

2.2 Grafische Veranschaulichung des Grover-Algorithmus 4

3 Der Grover-Algorithmus im Detail **7**

3.1 Die Hadamard-Transformation 7

3.2 Das Quanten-Orakel 8

3.3 Laufzeitanalyse des Grover-Algorithmus 8

4 Anwendungen des Grover-Algorithmus **10**

4.1 Schlüsselgenerierung bei aktuellen Verschlüsselungsverfahren 10

4.2 Auswirkungen des Grover-Algorithmus 11

5 Fazit und Ausblick **12**

Literatur **13**

Abbildungsverzeichnis

2.1 Gleichverteilte Superposition 4
2.2 Amplitudeninversion der Lösung 5
2.3 Lage des Mittelwertes aller Amplituden 5
2.4 Spiegelung aller Amplituden am Mittelwert 6

3.1 Quanten-Schaltkreis des Grover-Algorithmus 7
3.2 Schaltsymbol der Hadamard-Transformation 8

Tabellenverzeichnis

1.1 Beispiele von Suchproblemen 1

4.1 Symmetrische Verschlüsselungsverfahren und ihre Schlüssellängen 11
4.2 Auswirkungen auf symmetrische Verschlüsselungsverfahren 11

1 Einleitung

Bevor in den nachfolgenden Kapiteln insbesondere auf den Grover-Algorithmus eingegangen wird, ist es ratsam, sich zunächst eine grundlegende Fragestellung der theoretischen Informatik klarzumachen. Hierzu sollen die folgenden beiden Abschnitte dienen.

1.1 Berechnungsprobleme in der Informatik

Es ist allgemein bekannt, dass klassische Rechner einen Großteil ihrer Rechenzeit für Suchprobleme aufwenden. Meist ist eine Datenmenge gegeben, in der ein bestimmtes Element mit einer bestimmten Eigenschaft gefunden werden soll. Auch viele Berechnungsprobleme können als Suchproblem angesehen werden [Hom15]. In Tabelle 1.1 sind einige Beispiele von Berechnungsproblemen aufgeführt, die ebenso als Suchproblem formuliert werden können.

Berechnungsproblem	**Formulierung als Suchproblem**
Problem des Handlungsreisenden	Suche die kürzeste Route!
Optimierung	Suche die optimale Lösung!
Kryptoanalyse	Suche den Schlüssel!

Tabelle 1.1: Beispiele von Suchproblemen

1.2 Die unstrukturierte Suche

Das nachfolgende Beispiel zeigt ein weiteres Suchproblem, welches im Anschluss genauer betrachtet werden soll.

Beispiel 1.2.1 *Gegeben sei eine Telefonnummer und ein übliches, nach Namen sortiertes Telefonbuch. Wem gehört die Telefonnummer?*

Offensichtlich ist die Datenbank, dargestellt durch das Telefonbuch, unstrukturiert bezüglich des Suchkriteriums. Es kann also keine Aussage darüber getroffen werden, an

welcher Stelle sich das gesuchte Element, nämlich der zur Telefonnummer gehörige Name, in dieser Datenbank befindet. Um das gesuchte Element zu finden, gibt es keine andere Möglichkeit, als den gesamten Suchraum zu durchlaufen, bis letztendlich das gesuchte Element gefunden wurde.

Im Folgenden soll das oben geschilderte Problem etwas formaler betrachtet werden. Die Datenbank enthält N Elemente $\{0, \ldots, N-1\}$, für deren Anzahl $N = 2^n$ gilt. Jedes Element stammt aus der Definitionsmenge $\{0,1\}^n$. Das gesuchte Element wird als $\hat{x}$ bezeichnet. Die Datenbank kann nun als Funktion

$$f : \{0,1\}^n \to \{0,1\}$$

modelliert werden. Eine Datenbank-Anfrage, um zu erfahren, ob ein Element das Gesuchte ist, entspricht der Auswertung dieser Funktion, welche der Regel

$$f(x) = \begin{cases} 1, \text{ für } x = \hat{x} \in \{0,1\}^n \\ 0, \text{ für } x \in \{0,1\}^n \text{ mit } x \neq \hat{x} \end{cases}$$

folgt. Das Ergebnis der Funktion ist also 1, wenn das übergebene Element x auch das gesuchte Element $\hat{x}$ ist, sonst 0. Wie oben schon beschrieben, müssen die Elemente der Datenbank sukzessive durchlaufen werden, bis das gesuchte Element gefunden wird. Der Rechenaufwand und damit auch die benötigte Zeit steigen somit linear mit der Anzahl der Elemente N. Die unstrukturierte Suche ist also der Komplexitätsklasse $\mathcal{O}(N)$ zuzuordnen.

2 Übersicht über den Grover-Algorithmus

Für klassische Computer mögen die im vorangegangenen Kapitel gezogenen Schlussfolgerungen bezüglich der unstrukturierten Suche stimmen. Betrachtet man jedoch Quanten-Computer, ist die Einstufung der unstrukturierten Suche in die Komplexitätsklasse $\mathcal{O}(N)$ nicht mehr gültig. Der im Jahre 1960 geborene Informatiker Lov K. Grover entwickelte im Jahre 1996 bei den *Bell Laboratories* einen Quanten-Suchalgorithmus, der den Rechenaufwand für die unstrukturierte Suche deutlich verringert und seitdem als Grover-Algorithmus bekannt ist [Gro96]. Bevor dieser Algorithmus vorgestellt und analysiert wird, erfolgt im nächsten Abschnitt zunächst die Vorstellung eines allgemeinen Ansatzes für Quanten-Algorithmen.

2.1 Allgemeiner Ansatz für Quanten-Algorithmen

Der hier vorgestellte allgemeine Ansatz für Quanten-Algorithmen besteht aus den folgenden drei Schritten:

1. Beginne mit gleichverteilter Superposition aller möglichen Zustände.
2. Erhöhe sukzessive die Amplitude des Lösungszustands und verringere gleichzeitig die Amplituden aller anderen Zustände.
3. Miss den Quantenzustand, um die Lösung zu erhalten.

Aufgrund dieser Abfolge wird der Ansatz auch *Amplitudenverstärkung* genannt [Hom15].

2.2 Grafische Veranschaulichung des Grover-Algorithmus

Der Grover-Algorithmus implementiert den nun bekannten Ansatz der Amplitudenverstärkung beispielhaft [Hom15]. Der Ablauf des Algorithmus findet wie folgt statt:

Gleichverteilte Superposition

Entsprechend der Formel

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

werden die Qubits in gleichverteilte Superposition versetzt. Dies dient als Initialzustand für die nachfolgenden Schritte. Abbildung 2.1 zeigt das Ergebnis dieser Operation.

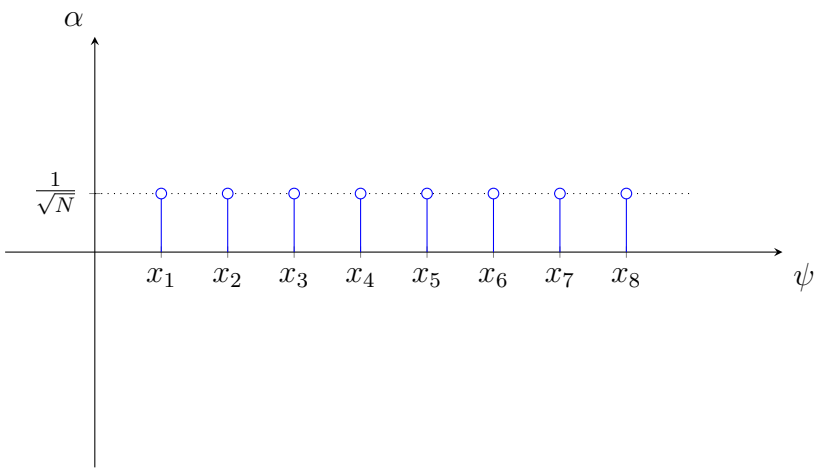

Abbildung 2.1: Gleichverteilte Superposition

Grover-Iteration

Nachdem nun der Initialzustand hergestellt wurde, wird solange die Grover-Iteration ausgeführt, bis die Lösung eine ausreichend hohe Wahrscheinlichkeit erreicht hat. Die Grover-Iteration wird in zwei Schritte unterteilt. Im ersten dieser Schritte wird gezielt die Amplitude der Lösung invertiert. Das Resultat dieser Operation ist der Abbildung 2.2 zu entnehmen. Hierbei sei x_3 die Lösung $\hat{x}$.

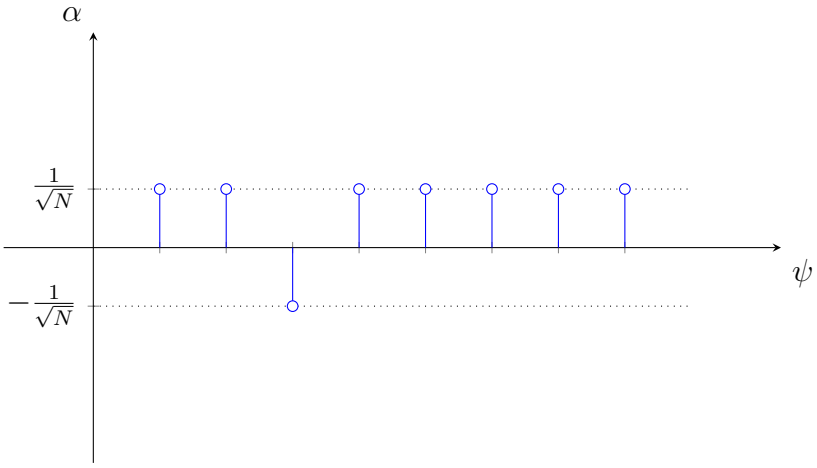

Abbildung 2.2: Amplitudeninversion der Lösung

Der zweite Schritt der GROVER-Iteration entspricht der Spiegelung aller Amplituden am Mittelwert. Dies hat zur Folge, dass die Amplitude der Lösung $\hat{x}$ erhöht wird, während alle anderen Amplituden verringert werden. In Abbildung 2.3 wird zunächst die Lage des Mittelwertes verdeutlicht, bevor in Abbildung 2.4 die Spiegelung der Amplituden an diesem Mittelwert dargestellt wird.

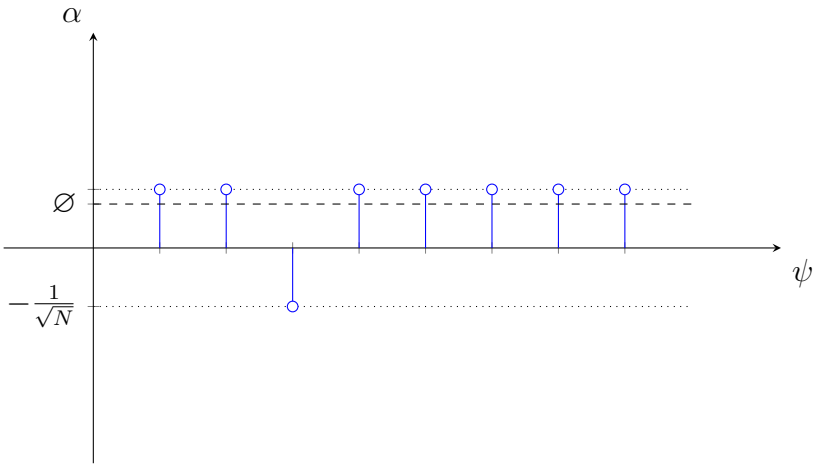

Abbildung 2.3: Lage des Mittelwertes aller Amplituden

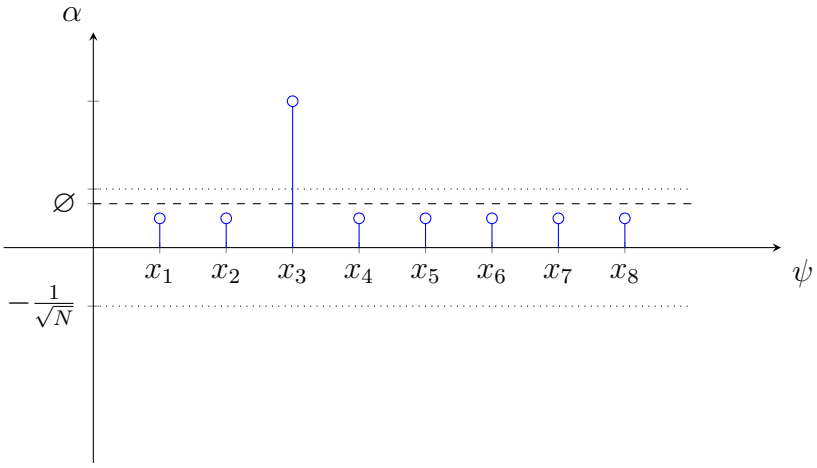

Abbildung 2.4: Spiegelung aller Amplituden am Mittelwert

Die Grover-Iteration wird solange wiederholt, bis das Ergebnis mit ausreichend hoher Wahrscheinlichkeit gemessen werden kann. Nun stellt sich jedoch die Frage, was *ausreichend hohe* Wahrscheinlichkeit bedeutet. Dieser und anderer Fragen wird im anschließenden Kapitel nachgegangen.

3 Der Grover-Algorithmus im Detail

Das vorige Kapitel hat den Grover-Algorithmus anschaulich, aber nur oberflächlich vorgestellt. In diesem Kapitel sollen die einzelnen Bestandteile des Algorithmus etwas genauer untersucht werden. Des Weiteren wird die Laufzeit des Algorithmus analysiert, um sie mit der in Kapitel 1 hergeleiteten klassischen Laufzeit von $\mathcal{O}(N)$ zu vergleichen.

Bevor nun jedoch die einzelnen Bestandteile des Grover-Algorithmus näher betrachtet werden, lohnt sich ein Blick auf den entsprechenden Quanten-Schaltkreis. Dieser ist für zwei Qubits in Abbildung 3.1 dargestellt [NC10].

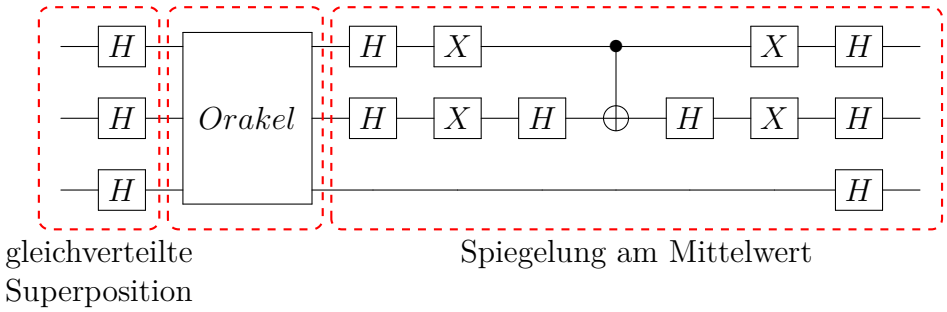

Abbildung 3.1: Quanten-Schaltkreis des Grover-Algorithmus

3.1 Die Hadamard-Transformation

Der erste Schritt des Grover-Algorithmus, das Herstellen der gleichverteilten Superposition, wird durch die Anwendung der Hadamard-Transformation auf jedes einzelne Qubit erreicht. Auf ein Qubit angewandt lässt sich die Hadamard-Transformation mit Hilfe der Matrix

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

beschreiben. Die Anwendung der Hadamard-Transformation auf ein Qubit-Register entspricht der Anwendung der Hadamard-Transformation auf jedes einzelne Qubit [Hom15].

In Quanten-Schaltkreisen besitzt die Hadamard-Transformation ein eigenes Symbol, welches in Abbildung 3.2 dargestellt ist.

Abbildung 3.2: Schaltsymbol der Hadamard-Transformation

Auf die quantenmechanischen Hintergründe und andere Anwendungsfälle der Hadamard-Transformation wird hier nicht weiter eingegangen, da dies für das hier angestrebte Maß des Verständnisses des Grover-Algorithmus nicht notwendig ist.

3.2 Das Quanten-Orakel

In Kapitel 1 wurde die Funktion

$$f(x) = \begin{cases} 1, \text{ für } x = \hat{x} \in \{0,1\}^n \\ 0, \text{ für } x \in \{0,1\}^n \text{ mit } x \neq \hat{x} \end{cases}$$

eingeführt, welche im klassischen Fall der unstrukturierten Suche dazu genutzt wurde, das Finden der gesuchten Lösung zu signalisieren. Auch der Grover-Algorithmus bedient sich dieser Funktion, welche in Form des Quanten-Orakels implementiert ist. Das Quanten-Orakel dient der Ausführung des ersten Schrittes der Grover-Iteration, also der Invertierung der Amplitude der Lösung. Es mag zunächst paradox klingen, dass explizit und allein die Amplitude der Lösung invertiert werden kann, da hierzu scheinbar die Lösung schon bekannt sein muss. Dem ist jedoch nicht so. Es muss hier deutlich zwischen zwei Fähigkeiten unterschieden werden. Das Quanten-Orakel ist in der Lage, die Lösung zu *erkennen*, nicht jedoch zu *bestimmen* [NC10].

Da das Quanten-Orakel für jedes Suchproblem speziell implementiert werden muss, also nicht universell einsetzbar ist, wird es in dieser Ausarbeitung als Black Box angenommen, deren Eigenschaft bekannt ist, jedoch nicht ihre genaue Implementierung.

3.3 Laufzeitanalyse des Grover-Algorithmus

Die Laufzeit der unstrukturierten Suche mit Hilfe von klassischen Computern liegt in der Komplexitätsklasse $\mathcal{O}(N)$. Dies wurde bereits in Kapitel 1 erläutert. Mit dem Aufkommen von Quanten-Computern kann für dieses Problem nun der Grover-Algorithmus verwendet werden, welcher eine deutliche Steigerung der Geschwindigkeit ermöglicht [Gro96]. In

diesem Abschnitt soll die Laufzeit des GROVER-Algorithmus analysiert werden, um ihn einer Komplexitätsklasse zuzuordnen.

Der GROVER-Algorithmus ist ein probabilistischer Algorithmus. Er findet die Lösung also nur mit einer bestimmten Wahrscheinlichkeit. Die in den vorangegangenen Abschnitten erläuterte GROVER-Iteration dient der Manipulation dieser Wahrscheinlichkeit zugunsten der richtigen Lösung. Dabei entspricht die Wahrscheinlichkeit, dass die Messung der Qubits die richtige Lösung ergibt, dem Quadrat der Amplitude des jeweiligen Zustandes. Für den GROVER-Algorithmus wird gefordert, dass die Wahrscheinlichkeit der Lösung vor der Messung mindestens $1/2 = 50\,\%$ beträgt [Gro96]. Die Amplitude der Lösung muss also mindestens $1/\sqrt{2}$ betragen. Des Weiteren kann gezeigt werden, dass sich die Amplitude der Lösung mit jeder Ausführung der GROVER-Iteration um mindestens $2/\sqrt{2N}$ erhöht [Gro96]. Die Anzahl der GROVER-Iterationen, die benötigt werden, um die geforderte Wahrscheinlichkeit zu erreichen, wird nun wie folgt berechnet:

$$\frac{\text{geforderte Wahrscheinlichkeit}}{\text{Verbesserung pro Iteration}} = \frac{\frac{1}{\sqrt{2}}}{\frac{2}{\sqrt{2N}}} = \frac{\sqrt{N}}{2} \Rightarrow \mathcal{O}(\sqrt{N})$$

Außerdem wurde bewiesen, dass die GROVER-Iteration keinesfalls öfter als $\sqrt{N}$-mal ausgeführt werden darf, da in diesem Fall die Amplitude der Lösung, und somit auch deren Wahrscheinlichkeit, wieder kleiner wird [Hom15].

Abschließend zu den hier angestellten Überlegungen stellt sich die Frage, ob es nicht einen Quanten-Algorithmus geben kann, der das Problem der unstrukturierten Suche in noch kürzerer Zeit zu lösen vermag. Dies ist jedoch nicht der Fall. Im Jahre 1996 haben die Wissenschaftler BENNETT, BERNSTEIN, BRASSARD und VAZIRANI mathematisch bewiesen, dass kein Quanten-Algorithmus für die unstrukturierte Suche existieren kann, der weniger als $\Omega(\sqrt{N})$ Schritte benötigt, um das gesuchte Element zu finden [Ben+97]. Der GROVER-Algorithmus gehört also zu den schnellstmöglichen Quanten-Suchalgorithmen und ist mit $\mathcal{O}(\sqrt{N})$ optimal bezüglich seiner Laufzeit.

4 Anwendungen des Grover-Algorithmus

Bereits in Kapitel 1 wurden einige Beispiele für Suchprobleme genannt. Da diese Ausarbeitung im Seminar „IT-Sicherheit“ entstanden ist, wird in diesem Kapitel untersucht, welche Auswirkungen der Grover-Algorithmus im Zusammenhang mit Quanten-Computern auf die Sicherheit aktueller Verschlüsselungsverfahren hat. Insbesondere werden hier symmetrische Verschlüsselungsverfahren betrachtet. Asymmetrische Verfahren wie beispielsweise RSA werden ausführlich in der Seminar-Ausarbeitung von Eduard Zeller behandelt.

4.1 Schlüsselgenerierung bei aktuellen Verschlüsselungsverfahren

Um die Auswirkungen des Grover-Algorithmus auf aktuelle symmetrische Verschlüsselungsverfahren einschätzen zu können, wird zunächst untersucht, auf welche Weise die Schlüsselgenerierung stattfindet. Anstelle von zahlentheoretischen Überlegungen wie bei den asymmetrischen Verfahren werden die Schlüssel bei symmetrischen Verfahren durch Zufallsgeneratoren erzeugt. Aufgrund dieser Tatsache sind für symmetrische Verfahren vergleichsweise kurze Schlüssel ausreichend sicher. In Tabelle 4.1 sind einige aktuelle symmetrische Verschlüsselungsverfahren mitsamt der jeweiligen Schlüssellängen aufgezählt.

Abkürzung	Bezeichnung	Schlüssellänge
DES	Data Encryption Standard	56 bit
3DES	Triple-DES	168 bit
AES	Advanced Encryption Standard	128 bit, 192 bit und 256 bit
?	Blowfish	32 bit bis 448 bit
?	Twofish	128 bit, 192 bit und 256 bit

Tabelle 4.1: Symmetrische Verschlüsselungsverfahren und ihre Schlüssellängen

4.2 Auswirkungen des Grover-Algorithmus

Im vorangegangenen Kapitel wurde gezeigt, dass der Grover-Algorithmus eine Laufzeit von $\mathcal{O}(\sqrt{N})$ besitzt. Wendet man diesen Algorithmus also im Zuge der Kryptoanalyse an, um Schlüssel von symmetrischen Verschlüsselungsverfahren zu knacken, so halbiert sich die effektive Schlüssellänge [Fum17]. Die Tabelle 4.2 verdeutlicht die Auswirkungen des Grover-Algorithmus auf die wohl bekanntesten symmetrischen Verschlüsselungsverfahren.

Verfahren	Schlüssellänge	Effektive Schlüssellänge
DES	56 bit	28 bit
AES-128	128 bit	64 bit
AES-192	192 bit	96 bit
AES-256	256 bit	128 bit

Tabelle 4.2: Auswirkungen auf symmetrische Verschlüsselungsverfahren

Aus den Daten ist ersichtlich, dass lediglich AES-256 nach dem Einsatz des Grover-Algorithmus noch über eine effektive Schlüssellänge von über 100 bit aufweist. Alle anderen Verfahren sind nicht mehr als ausreichend sicher anzusehen. So negativ dies nun klingen mag, bedeutet dies jedoch auch, dass es ausreichend ist, die Schlüssellänge zu verdoppeln, um dasselbe Sicherheitslevel zu erhalten wie vor dem Einsatz des Grover-Algorithmus.

5 Fazit und Ausblick

Die vorliegende Ausarbeitung hat einen Überblick über den Grover-Algorithmus gegeben, ohne ein allzu großes Vorwissen der Quantenmechanik vorauszusetzen. Zusammenfassend lässt sich der Grover-Algorithmus als probabilistischer Quanten-Algorithmus ansehen, welcher das Problem der unstrukturierten Suche mit einem Zeitaufwand von $\mathcal{O}(\sqrt{N})$ und einer Wahrscheinlichkeit von mindestens 50 % zu lösen vermag.

Allerdings gibt es noch einige offene Probleme, die im Folgenden angesprochen werden.

Die Messung des Quantenregisters wird im Großteil der Betrachtungen zu Quanten-Algorithmen nicht weiter behandelt und quasi als trivial angenommen. Dies ist jedoch nicht der Fall. Die Messung von Qubits ist ein sehr komplexes Unterfangen. Um die Messung so zuverlässig wie möglich durchzuführen, kann sogar mehr Zeit notwendig sein, als die eigentliche Berechnung benötigt.

Die Datenbank mit allen möglichen Zuständen, in welcher der Grover-Algorithmus suchen soll, muss entweder direkt als Quantensystem implementiert oder zumindest über ein solches adressierbar sein. Auch dies bringt einige Hürden mit sich, welche die Anwendung des Grover-Algorithmus weiter erschweren.

Das Quanten-Orakel muss ohne genaue Kenntnis der Lösung implementiert werden. Mit akademischen Betrachtungen, die das Quanten-Orakel aus der bereits bekannten Lösung ableiten, kann zwar gezeigt werden, dass der Grover-Algorithmus in der Theorie funktioniert. In praktischen Anwendungen wird die Lösung jedoch nicht bekannt sein, sondern kann nur anhand von bestimmten Kriterien erkannt werden.

Aufgrund dieser Probleme wurde der Grover-Algorithmus bisher *nicht* praxisrelevant eingesetzt.

Literatur

[Ben+97] Charles H. Bennett u. a. „Strengths and weaknesses of quantum computing“. In: *SIAM journal on Computing* 26.5 (1997), S. 1510–1523.

[Fum17] Walter Fumy. „Quantencomputer und die Zukunft der Kryptographie“. In: *Datenschutz und Datensicherheit-DuD* 41.1 (2017), S. 13–16.

[Gro01] Lov K. Grover. „From Schrödinger's equation to the quantum search algorithm“. In: *American Journal of Physics* 69.7 (2001), S. 769–777.

[Gro96] Lov K. Grover. „A fast quantum mechanical algorithm for database search“. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM. 1996, S. 212–219.

[Gro97] Lov K. Grover. „Quantum mechanics helps in searching for a needle in a haystack“. In: *Physical review letters* 79.2 (1997), S. 325.

[Hom15] Matthias Homeister. *Quantum Computing verstehen: Grundlagen-Anwendungen-Perspektiven*. 4. Aufl. Springer-Verlag, 2015.

[Mos09] Tim Moses. „Quantum Computing and Cryptography“. In: *Entrust Inc. January* (2009).

[NC10] Michael A. Nielsen und Isaac L. Chuang. *Quantum Computation and Quantum Information*. 10. Aufl. Cambridge University Press, 2010.