INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Quantum computational attack on two diophantine cryptosystems

by

Michael Salwen

A dissertation submitted to the Graduate Faculty in Mathematics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy, The City University of New York

2001

This manuscript has been read and accepted for the Graduate Faculty in Mathematics in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

3/30/01
Date

Michael Anshel
Chair of Examining Committee

3/30/01
Date

Józef Dodziuk
Executive Officer

Michael Anshel

Burton Randol

Alphonse Vasquez

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

Quantum computational attack on two diophantine cryptosystems

by

Michael Salwen

Advisor: Professor Michael Anshel

A computing paradigm based on the principles of quantum physics was first suggested in the early 1980s by Benioff and Feynman and formalized thereafter by Deustch. Since that time, Shor introduced specific polynomial-time quantum algorithms to solve the factoring and discrete logarithm problems and Grover presented a quantum database search algorithm with quadratic improvement over classical search. Shor's method has been generalized to the problem of finding a hidden subgroup within an abelian group G and is intimately connected to the Fourier transform on G. Grover's method is fundamentally different and has a decidedly geometric flavor.

Quantum algorithms offer an inherently parallel structure, which derives from the quantum physical notion of superposition, whereby particles are thought to exist simultaneously in various states, subject to some probability amplitude distribution. In terms of the computing paradigm, we can think of an n-bit quantum register as simultaneously holding all possible 2^n values, again subject to some probability distribution on the values. The

power of quantum parallelism, in essence, is that a function can be evaluated simultaneously rather than sequentially on all points in a large space.

The interest in quantum computing and algorithms is stimulated largely by the promise of tractable solutions to problems believed to be intractable within the classical computing model. Among these are those problems underlying all modern encryption schemes, including the two examined here: the NTRU and Chor–Rivest cryptosystems. The most widely used classical technique to attack these systems is lattice basis reduction. The approach taken here is quite different. These problems are treated as inherently generic: find one of some special elements within an unsorted database. Grover's algorithm is then applied in this context.

The basic mathematical notions for the quantum computing paradigm are introduced, followed by a detailed description of Grover's algorithm. The NTRU and Chor–Rivest cryptosystems are then presented, with discussions of conventional attacks and how Grover's algorithm may be applied to these problems.

Acknowledgments

As anyone who has ever endeavored to survive graduate school knows, the choice of thesis advisor is perhaps the greatest test of judgment and most important decision to be made throughout the entire enterprise. I can say unequivocally that in this, at least, my performance has been exemplary. Far beyond the serious responsibility of intellectual guidance and mentoring, a great advisor has to know how and (more important) when to push, cajole, encourage, discourage, bully, coddle, persevere and relent, all in an effort to help the unwieldy object known as his graduate student get over the final hurdle. I certainly tested Mike Anshel's skills in all these ways, probably much more, at times, than he thought he had bargained for. I cannot thank him enough with these words for the support he provided throughout many years. During some awfully tough times, when finishing this degree seemed impossible, he said or did what needed to be said or done to keep me on track and moving forward (albeit sometimes very slowly).

Many thanks go as well to Professors Burt Randol and Al Vasquez for serving on my committee and for the enjoyment of their company and intellectual guidance over the years. My most pleasant memories of life in the CUNY Mathematics department will often revolve around time spent with them. In particular, some of my most enjoyable and stimulating experiences in a classroom anywhere were when Al Vasquez was standing up in front.

Finally, in the broader picture of my life, I must thank those who know perhaps the least about mathematics, but the most about me. Elisa

Bienenstock returned the favor of support, encouragement and, I admit, the occasional strong remark of not only a good friend but of one who had been there—literally locking herself in a closet to finish her dissertation! Since childhood, the two best friends one could wish for in this world, Bruce Woolf and Jon Frankel, have stood with me in the best and worst of times. I have shared everything with them as brothers, except parents.

But, the greatest source of strength and support has always been my wonderful family. My *tantes*, Beatrice and Leah Kimelman, are the most exuberantly loving fans a kid could ever hope for (though I think they are unduly biased in their judgment), and they keep me well fed. For all their lives, my sisters Betsy, Judy and Nancy have shown me that, quantum *schmantum*, to give unconditional love and support is the true reason for us being in this chaotic universe. What's more, they brought home three terrific guys to play with: Jon Stahl, Dan Lehman and Kevin Ruhl are the best brothers any brotherless boy could ever have. And, of course, the most unbearably cute kids in the world, my nephews and niece Max, Dylan, Charlotte and Sam, bring instant radiant joy into my life as soon as I get my hands on them. Finally, Yaffa Lerea is always there with her love, understanding, encouragement and the well timed kick in the butt. She kept telling me I could finish this thing, and it turns out she was right.

Thank you all.

To the memory of Mom and Dad,
who gave me life,
liberty (often more than prudently advisable)
and the pursuit of knowledge.

MTYEK

Contents

1 Introduction

A computing paradigm based on the principles of quantum physics was first broached in the early 1980s by Benioff [1] and Feynman [8] and formalized thereafter by Deustch [5]. Since that time, Shor introduced specific polynomial-time quantum algorithms to solve the factoring and discrete logarithm problems [19] and Grover presented a quantum database search algorithm with quadratic improvement over classical search [10]. Shor's method has been generalized to the problem of finding a hidden subgroup within an abelian group G [12, 14, 16] (e.g. the problem of finding the subgroup generated by a specific element of $\mathbb{Z}/n\mathbb{Z}$ and, hence, its order) and is intimately connected to the Fourier transform on G. Grover's method is fundamentally different and, as we will see in detail, has a decidedly geometric flavor.

The interest in quantum computing and algorithms is stimulated largely by the promise of tractable solutions to problems believed to be intractable within the classical computing model. While no one knows whether problems such as factoring and the computation of discrete logarithms are truly hard within classical constraints, there is probably no one alive who believes not. Quantum algorithms offer an inherently parallel structure, which derives from the quantum physical notion of superposition, whereby particles are thought to exist simultaneously in various states, subject to some probability amplitude distribution. In terms of the computing paradigm, we can think of an n-bit quantum register as simultaneously holding all possible 2^n

values, again subject to some probability distribution on the values. The power of quantum parallelism, in essence, is that a function can be evaluated simultaneously rather than sequentially on all points in a large space.

Of course, there are constraints within the quantum computational model, not the least of which is that once a quantum register is examined, superposition is lost and one and only one state (corresponding to one of the 2^n possible values) is recovered. The aim of quantum algorithms is to manipulate quantum registers without "peeking," waiting until the right moment to destroy superposition and recover one of the possible outcomes. The right moment, of course, is when the algorithm has created an amplitude distribution which heavily favors a desirable outcome.

Another limitation, which will not be treated here, but which obviously bears directly on the feasibility of quantum computing, is that the hardware for such devices does not exist. More important, it remains to be seen when and whether such hardware will exist, though most observers are hopeful. The hardest problem seems to be the ability to manipulate quantum states while isolating them from unwanted environmental interference.

This paper applies Grover's algorithm to attacking the NTRU and Chor–Rivest cryptosystems. Of the two, NTRU remains the most resistant to conventional attacks, though the example provided by Chor–Rivest remains theoretically interesting. The most widely used conventional technique to solve these problems is lattice basis reduction. Initially, an attacker must create a lattice in which a short vector exists which will yield a solution to

the specific problem. If such a lattice can be constructed, the lattice basis reduction algorithm due to Lenstra, Lenstra and Lovász [36] (the so-called LLL algorithm) and successive improvements thereupon [33, 34, 35, 37, 38] can be used to search for short vectors. The original LLL algorithm guarantees finding within a d-dimensional lattice a vector of length no greater than $2^{d/2}$ times the length of the shortest non-zero vector, though in practice the algorithm often does much better. As part of the algorithm, the lengths of linear combinations of two basis vectors are minimized. Changes to the basic algorithm include increasing the number of basis vectors in these linear combinations (block size) beyond two, which yields, at a time cost exponential in the size of the block, shorter vectors overall, and methods for optimizing the balance between the speed of the arithmetic computations required by the algorithm and excessive rounding errors which can lead to complete instability and failure to converge.

The approach taken here is quite different. We view these problems as that of finding one of some special elements within an unsorted database and apply Grover's algorithm to them. This methodology is inherently generic; the algebraic structures underlying the systems examined here play very little role in the operation of the search algorithm. Indeed, the only interaction with the search space required by the algorithm is a membership function that can faithfully determine which elements are in the special subset.

The sequel is divided into five sections. In section 2 the basic mathematical notions for the quantum computing paradigm are introduced. Sec-

tion 3 describes Grover's algorithm in detail. Sections 4 and 5 introduce the NTRU and Chor–Rivest cryptosystems, respectively, with discussions of conventional attacks and how Grover's algorithm may be applied to these problems. Finally, section 6 mentions directions for future work.

2 Mathematical framework of quantum computation

2.1 Introductory quantum concepts

The context for quantum systems is a Hilbert space, $\mathcal{H}$, equipped with hermitian inner product $(\cdot,\cdot)$ such that for any $x, y, z \in \mathcal{H}$ and $\alpha \in \mathbb{C}$:

1. $(x,x) \geq 0$ and $(x,x) = 0$ iff $x = 0$,
2. $(x,y) = \overline{(y,x)}$,
3. $(x, \alpha y + z) = \alpha(x,y) + (x,z)$.

$\mathcal{H}$ is variously referred to as the *state space*, *quantum probability space* (QPS) or, when there is no room for confusion, just *space*. For our purposes, $\mathcal{H}$ will be finite dimensional. *States* or *state vectors* will be normalized elements of $\mathcal{H}$ and will be denoted $|\ell\rangle$, where ℓ is some label.

In quantum systems, an *event* is a set of initial and final states. By $\langle\phi|$ we mean the element $(|\phi\rangle, \cdot) \in \mathcal{H}^*$. With this convention, $\langle\phi|\iota\rangle$ shall mean $(|\phi\rangle, |\iota\rangle)$ and will be called the *probability amplitude* (or more simply *amplitude*) of an event with initial state $|\iota\rangle$ and final state $|\phi\rangle$. In the context of quantum mechanics, the nomenclature suggests the interpretation of the amplitude; in fact, $|\langle\phi|\iota\rangle|^2$ is the probability of occurrence of the event with initial state $|\iota\rangle$ and final state $|\phi\rangle$.

The amplitude has analogous properties to probability measures. If an event can be split into two sequential events, then the amplitude of the event

is equal to the product of the amplitudes of the constituent events. Therefore, if $|x\rangle$ is an intermediate state between states $|\iota\rangle$ and $|\phi\rangle$, then

$$\langle\phi|\iota\rangle = \langle\phi|x\rangle\langle x|\iota\rangle.$$

If an event can occur in several independent ways, then the amplitude of the event is the sum of the amplitudes of the several alternative events. For example, if the event can be represented as either of two independent events through the intermediate states $|x\rangle$ and $|y\rangle$, then

$$\langle\phi|\iota\rangle = \langle\phi|x\rangle\langle x|\iota\rangle + \langle\phi|y\rangle\langle y|\iota\rangle.$$

Remark. An important distinction not to be lost between amplitudes and probabilities is that since amplitudes carry phase as well as magnitude, linear operations on states can create destructive as well as constructive interference. Grover's algorithm exploits this aspect to augment certain desirable event probabilities while suppressing others.

By a basis for $\mathcal{H}$ we shall mean a set of state labels $\mathcal{B}$ such that for all $i, j \in \mathcal{B}$,

$$\langle i|j\rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

and for any initial state $|\iota\rangle$ and final state $|\phi\rangle$,

$$\langle\phi|\iota\rangle = \sum_{i\in\mathcal{B}}\langle\phi|i\rangle\langle i|\iota\rangle.$$

Any state $|\varphi\rangle$ in a QPS can be represented as a linear combination of basis states:

$$|\varphi\rangle = \sum_{i\in\mathcal{B}} a_i|i\rangle, \text{ where } a_i = \langle i|\varphi\rangle \text{ for all } i \in \mathcal{B}.$$

If $\{|i\rangle\}_{i\in\mathcal{B}}$ forms a basis for $\mathcal{H}$ then $\{\langle i|\}_{i\in\mathcal{B}}$ forms a basis for $\mathcal{H}^*$ and any element of $\mathcal{H}^*$ can be represented as a linear combination of elements from the dual basis:

$$\langle\psi| = \sum_{i\in\mathcal{B}} a_i\langle i|, \text{ where } a_i = \langle\psi|i\rangle \text{ for all } i \in \mathcal{B}.$$

From the definition of the inner product, it follows that if $|\varphi\rangle = \sum a_i|i\rangle$, then $\langle\varphi| = \sum \overline{a_i}\langle i|$.

For a given basis $\mathcal{B}$ of $\mathcal{H}$ (and, therefore, a given dual basis of $\mathcal{H}^*$), the foregoing implies that if $|\varphi\rangle = \sum_{i\in\mathcal{B}} a_i|i\rangle$ and $|\psi\rangle = \sum_{i\in\mathcal{B}} b_i|i\rangle$, then $\langle\psi|\varphi\rangle = \sum_{i\in\mathcal{B}} a_i\overline{b_i}$. It is also useful in this context to think of elements of $\mathcal{H}^*$ as vectors under the mapping $\mathcal{H} \to \mathcal{H}^* : \alpha \mapsto \overline{\alpha}$. The inner product then becomes the standard dot product, with $(\alpha, \beta) = \overline{\alpha} \cdot \beta$.

2.2 Qubits and quantum registers

The fundamental unit of data in the quantum computational model is the "qubit" (for *quantum* bit). Its mathematical representation is a state in a two-dimensional Hilbert space, $\mathbb{H}_2$. The canonical basis for $\mathbb{H}_2$ is suggestively denoted as $\{|0\rangle, |1\rangle\}$. A qubit is, therefore, of the form $|\varphi\rangle = a|0\rangle + b|1\rangle$ where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$. If both a and b are nonzero, we say that $|\varphi\rangle$ is

in a *superposition* of $|0\rangle$ and $|1\rangle$, to be distinguished from *either* $|0\rangle$ *or* $|1\rangle$ as in a classical computational model.

Quantum qubit registers are superficially analogous to classical bit registers, although the same distinction as in the single bit/qubit case can be made: whereas an n-bit classical register can hold *one* of 2^n values, an n-qubit register can be realized as a normalized element in the n-fold tensor product of $\mathbb{H}_2$. (The 2^n-dimensional Hilbert space $\bigotimes_n \mathbb{H}_2$ will be denoted henceforth as $\mathbb{H}_N$, where $N = 2^n$.) Extending the notation introduced for the basis of $\mathbb{H}_2$, a 2-qubit register is represented as a state in the space spanned by $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, where $|ij\rangle$ means $|i\rangle \otimes |j\rangle$. Canonical basis states of $\mathbb{H}_N$ will be denoted $|i\rangle$, for $i \in \{0, 1, \ldots, N-1\}$, where i is understood to represent the n-bit binary expansion of the integer. In general, the tensor of any $|x\rangle \in \mathbb{H}_N$ and $|y\rangle \in \mathbb{H}_M$ may be written as any of $|x\rangle \otimes |y\rangle$, $|xy\rangle$, $|x\rangle|y\rangle$ or $|x, y\rangle$, always in the hope of clarity. Note that, as in the case of a single qubit, an arbitrary n-qubit register $|\varphi\rangle$ can be represented as a linear combination of the basis states of $\mathbb{H}_N$:

$$|\varphi\rangle = \sum_{i=0}^{N-1} a_i |i\rangle, \text{ where } a_i = \langle i|\varphi\rangle \text{ for all } i \text{ and } \sum_{i=0}^{N-1} |a_i|^2 = 1.$$

2.3 Evolution of quantum systems; entanglement

A state $|\varphi\rangle \in \mathbb{H}_N$ is called *pure* if there exist $|\varphi_1\rangle, |\varphi_2\rangle, \ldots, |\varphi_n\rangle \in \mathbb{H}_2$ such that

$$|\varphi\rangle = |\varphi_1\rangle \otimes |\varphi_2\rangle \otimes \cdots \otimes |\varphi_n\rangle.$$

If no such representation of $|\varphi\rangle$ is possible, then $|\varphi\rangle$ is said to be *entangled.* As an example, compare the states

$$|\varphi\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle), \text{ and}$$

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Since $|\varphi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes |0\rangle$, we see that it is a pure state. No such decomposition is possible for $|\psi\rangle$ and it is, therefore, entangled.

If $\mathcal{O}_1 : \mathcal{H}_1 \to \mathcal{G}_1$ and $\mathcal{O}_2 : \mathcal{H}_2 \to \mathcal{G}_2$ are operators, we define their tensor product by:

$$(\mathcal{O}_1 \otimes \mathcal{O}_2)\left(e_i^1 \otimes e_j^2\right) = \mathcal{O}_1 e_i^1 \otimes \mathcal{O}_2 e_j^2,$$

where e_i^1 and e_j^2 are basis elements of $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. We then extend this definition linearly to define the operator $(\mathcal{O}_1 \otimes \mathcal{O}_2) : \mathcal{H}_1 \otimes \mathcal{H}_2 \to \mathcal{G}_1 \otimes \mathcal{G}_2$. The tensor product has the following properties:

1. $(\mathcal{O}_1 \otimes \mathcal{O}_2)^* = \mathcal{O}_1^* \otimes \mathcal{O}_2^*$, where $\mathcal{O}^*$ is the adjoint of $\mathcal{O}$;
2. $(\mathcal{O}_1 \otimes \mathcal{O}_2)^{-1} = \mathcal{O}_1^{-1} \otimes \mathcal{O}_2^{-1}$; and
3. If $\mathbf{A}$ and $\mathbf{B}$ are matrices relative to given bases of $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively, we have, from the definition of $\mathcal{O}_1 \otimes \mathcal{O}_2$, that $(\mathbf{A} \otimes \mathbf{B})(x \otimes y) = \mathbf{A}x \otimes \mathbf{B}y$ for all $x \in \mathcal{H}_1$ and $y \in \mathcal{H}_2$. If $\mathbf{A} = (a_{ij})$ is $r \times s$, then the matrix of $\mathcal{O}_1 \otimes \mathcal{O}_2$ is given by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1s}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2s}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1}\mathbf{B} & a_{r2}\mathbf{B} & \cdots & a_{rs}\mathbf{B} \end{pmatrix}.$$

Quantum systems evolve through unitary transformations of the state space. A unitary operator is one which maps one orthonormal basis on to another. Equivalently, if $\mathcal{U}$ is unitary, then $\mathcal{U}^{-1} = \mathcal{U}^*$, *i.e.*, its inverse and adjoint are identical. If $\mathcal{U}_1$ and $\mathcal{U}_2$ are unitary operators on Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, then $\mathcal{U}_1 \otimes \mathcal{U}_2$ is a unitary operator on $\mathcal{H}_1 \otimes \mathcal{H}_2$.

Not all useful unitary operators on $\mathbb{H}_N$ can be represented as the tensor of operators on $\mathbb{H}_2$; however, one that does come up frequently in the study of quantum algorithms is $\mathcal{S}_n = \otimes_n \mathcal{S}$, where $\mathcal{S} : \mathbb{H}_2 \rightarrow \mathbb{H}_2$ is the *Walsh-Hadamard* transformation defined by:

$$\mathcal{S}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$\mathcal{S}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

$\mathcal{S}_n$ has the useful property that for an arbitrary basis element $|x\rangle \in \mathbb{H}_N$:

$$\mathcal{S}_n|x\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{x \cdot i} |i\rangle,$$

where $x \cdot i$ is the (mod 2) dot product of the binary vectors.

If we take $x = 0$, then:

$$\mathcal{S}_n|0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle,$$

i.e., $\mathcal{S}_n$ provides a ready tool for creating a state that is in equiprobable superposition of the canonical basis states of $\mathbb{H}_N$.

The matrix of $\mathcal{S}_1$ relative to the canonical basis of $\mathbb{H}_2$ is

$$\mathbf{S}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

and, for $n > 1$, we have the nice recursive definition for the matrix of $\mathcal{S}_n$ relative to the canonical basis of $\mathbb{H}_N$:

$$\mathbf{S}_n = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{S}_{n-1} & \mathbf{S}_{n-1} \\ \mathbf{S}_{n-1} & -\mathbf{S}_{n-1} \end{pmatrix}.$$

The Walsh-Hadamard transformation and other operators which are tensors of operators on $\mathbb{H}_2$ are easy to understand and apply as the operation can be performed qubit by qubit. They also clearly carry pure states into pure states. Operators which cannot be represented as the tensor of operators on $\mathbb{H}_2$ will entangle pure states. For example, if

$$\mathbf{A} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix}, \text{ and}$$

$$|\varphi\rangle = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}$$

$$= \frac{1}{2} \left(|00\rangle + |01\rangle - |10\rangle - |11\rangle \right)$$

$$= \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle \right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle \right),$$

then $\mathbf{A}$ cannot be represented as a tensor of operators on $\mathbb{H}_2$ and $|\varphi\rangle$ is, as

demonstrated, a pure state. Now,

$$\mathbf{A}|\varphi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \left(|01\rangle + |10\rangle \right),$$

which is an entangled state.

3 Grover's algorithm for database search

3.1 Description of the basic algorithm

Grover's algorithm is a quantum search technique designed to retrieve one of some elements satisfying a given property from an unsorted database. The algorithm assumes a database of size S to be represented by the canonical basis states $|i\rangle$, for $i = 0, 1, \ldots, S-1$, and operates within the subspace $\mathbb{H}_S \subset \mathbb{H}_N$, where $N = 2^n$ and $n = \lceil \log_2 S \rceil$. We also assume the existence of a one-to-one correspondence identifying the problem-specific search space with the set $\{0, 1, \ldots, S-1\}$; a problem-specific membership function $f : \{0, 1, \ldots, S-1\} \rightarrow \{0, 1\}$, where $f(i) = 1$ if and only if $|i\rangle$ satisfies the given property; and a unitary transformation $\mathcal{U}_f$ that takes states $|i\rangle\,|y\rangle$ to $|i\rangle\,|f(i) \oplus y\rangle$, where $|i\rangle$ is a canonical basis element of $\mathbb{H}_S$, $|y\rangle \in \mathbb{H}_2$ and $\oplus$ signifies addition mod 2.

The state $|i\rangle$ is called *marked* if $f(i) = 1$. We will denote by M both the set of basis labels corresponding to the marked states and the subspace of $\mathbb{H}_S$ spanned by these states, with the expectation that this mild abuse will not cause confusion. The number of elements in M will be denoted by the integer t, where we will assume throughout that $t > 0$ and, for now, that t is a known parameter.

The algorithm consists of initialization, an iterative loop, measurement and verification.

Initialization. The algorithm is initialized by first preparing an equiprobable state

$$\frac{1}{\sqrt{S}} \sum_{i=0}^{S-1} |i\rangle |y\rangle$$

where $|y\rangle$ is set to $\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \in \mathbb{H}_2$. This choice for $|y\rangle$ is amplified in the following section. In the event that S is a power of 2, we can employ the Walsh-Hadamard transformation to prepare this state.

The iterative loop. The heart of the algorithm is a loop, which iterates $J = O\left(\sqrt{\frac{S}{t}}\right)$ times. The exact value of J will be discussed in the next section. The loop comprises two steps. The first step applies the specific transformation $\mathcal{U}_f$ to the input state. The second step then applies what Grover calls the *diffusion* transformation, $\mathcal{D}$, which is defined by

$$\sum_{i=0}^{S-1} a_i |i\rangle \mapsto \sum_{i=0}^{S-1} (2\mathfrak{a} - a_i) |i\rangle, \text{ where } \mathfrak{a} = \frac{1}{S} \sum_{i=0}^{S-1} a_i.$$

Measurement. The state output by the iterative loop is measured. This destroys superposition and returns one and only one basis state as a candidate solution for the problem.

Verification. This is a classical check of whether the algorithm has returned a correct answer. In the event the algorithm fails, the process repeats. In the discussion below, we show that the probability of failure is no greater than $\frac{t}{S}$, which is quite small for $t \ll S$.

3.2 Analysis of the algorithm

The key to Grover's algorithm is how the iterative loop transforms an equiprobable state to one which, upon measurement, returns some basis state $|i\rangle \in M$ with near certainty. It is this loop which we examine in this section.

Recall that for $i \in \{0, 1, \ldots, S-1\}$ and $|y\rangle \in \mathbb{H}_2$,

$$\mathcal{U}_f : |i\rangle|y\rangle \mapsto |i\rangle|f(i) \oplus y\rangle.$$

If we take $|y\rangle$ to be $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, then

$$\begin{aligned}\mathcal{U}_f : |i\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \mapsto |i\rangle\frac{1}{\sqrt{2}}(|f(i) \oplus 0\rangle - |f(i) \oplus 1\rangle) \\ = |i\rangle\frac{1}{\sqrt{2}}(|f(i)\rangle - |f(i) \oplus 1\rangle) \\ = |i\rangle(-1)^{f(i)}\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).\end{aligned}$$

If we suppress the notation of the final qubit, we have

$$\mathcal{U}_f : \sum_{i=0}^{S-1} a_i|i\rangle \mapsto \sum_{i=0}^{S-1}(-1)^{f(i)}a_i|i\rangle,$$

so that, with these manipulations, $\mathcal{U}_f$ is shown to encode f as a conditional phase change of the marked states.

We see here an explicit manifestation of the power of quantum algorithms. One application of $\mathcal{U}_f$ creates a state which encodes global properties of f. The key to exploiting this power hinges on maintaining superposition, while employing those unitary transformations of the state space which will

"promote" a correct problem solution into the overwhelmingly likely outcome of measurement.

The other component of the loop is the transformation $\mathcal{D}$. Note that $\mathcal{D} = -I + 2\mathcal{P}$, where $\mathcal{P}$ is defined by

$$\sum_{i=0}^{S-1} a_i|i\rangle \mapsto \mathfrak{a} \sum_{i=0}^{S-1} |i\rangle, \text{ where again } \mathfrak{a} = \frac{1}{S}\sum_{i=0}^{S-1} a_i.$$

We observe that $\mathcal{P}^2 = \mathcal{P}$, from which it follows immediately that $\mathcal{D}^2 = I$ and, hence, that $\mathcal{D}$ is unitary.

Grover describes the action of $\mathcal{D}$ as an "inversion about average," which is readily understood by making the trivial observation that $2\mathfrak{a} - a_i = \mathfrak{a} + (\mathfrak{a} - a_i)$. Coupled with the conditional phase change applied by $\mathcal{U}_f$ (*i.e.*, only to marked states) immediately prior to this inversion, the loop amplifies the amplitude of marked states and suppresses that of the others. Let's examine this process closely.

Suppose $k, l \in \mathbb{R}$ are such that $tk^2 + (S-t)l^2 = 1$. Put

$$|\Psi(k,l)\rangle = k\sum_{i\in M} |i\rangle + l\sum_{i\notin M} |i\rangle.$$

Then

$$\mathcal{U}_f|\Psi(k,l)\rangle = -k\sum_{i\in M} |i\rangle + l\sum_{i\notin M} |i\rangle.$$

Note that the average of the coefficients of $\mathcal{U}_f|\Psi(k,l)\rangle$, $\mathfrak{a}$, equals $\frac{(S-t)l-tk}{S}$. The coefficient for each of the marked basis states in the state $\mathcal{D}\mathcal{U}_f|\Psi(k,l)\rangle$

is, therefore, transformed as

$$k \rightarrow -k \rightarrow 2a - (-k) = \frac{2}{S}\left[(S-t)\,l - tk\right] + k$$
$$= \frac{S-2t}{S}k + \frac{2(S-t)}{S}l.$$

Similarly, the coefficient for each of the unmarked states is transformed as

$$l \rightarrow 2a - l = \frac{2}{S}\left[(S-t)\,l - tk\right] - l$$
$$= \frac{S-2t}{S}l - \frac{2t}{S}k.$$

The iteration of the central loop of Grover's algorithm yields, therefore, the following recurrence among the coefficients of the marked and unmarked states:

$$k_{j+1} = \frac{S-2t}{S}k_j + \frac{2(S-t)}{S}l_j \qquad \text{and} \qquad l_{j+1} = \frac{S-2t}{S}l_j - \frac{2t}{S}k_j,$$

with initial conditions $k_0 = l_0 = \frac{1}{\sqrt{S}}$. Significantly, this recurrence conserves total probability, since

$$tk_{j+1}^2 + (S-t)l_{j+1}^2 = tk_j^2 + (S-t)l_j^2.$$

Let the angle θ be such that $\sin\theta = \sqrt{\frac{t}{S}}$ and $0 < \theta \leq \frac{\pi}{2}$. Then the following closed form for the recurrence given above can be verified by a straightforward induction argument:

$$k_j = \frac{1}{\sqrt{t}}\sin(2j+1)\theta \qquad \text{and} \qquad l_j = \frac{1}{\sqrt{S-t}}\cos(2j+1)\theta.$$

The amplitude formula clearly shows that we can allow the algorithm to work too hard. Indeed, if $(2j+1)\theta$ exceeds $\frac{\pi}{2}$, the magnitude of k_j begins to decline while that of l_j increases. Optimally, we would iterate the loop $\tilde{J}$ times, where $(2\tilde{J}+1)\theta = \frac{\pi}{2}$ or $\tilde{J} = \frac{\pi}{4\theta} - \frac{1}{2}$. Of course, we are constrained to iterate an integer number of times. If we put $J = \left\lfloor \frac{\pi}{4\theta} \right\rfloor$, then $\left|\tilde{J} - J\right| \leq \frac{1}{2}$, from which it follows that $\left|(2\tilde{J}+1)\theta - (2J+1)\theta\right| \leq \theta$. But $\tilde{J}$ was chosen expressly so that $(2\tilde{J}+1)\theta = \frac{\pi}{2}$, which implies that $\left|\frac{\pi}{2} - (2J+1)\theta\right| \leq \theta$ and, therefore, that $|\cos(2J+1)\theta| \leq \sin\theta$.

Now, the probability of failure, *i.e.*, that upon measurement we obtain one of the $S-t$ unmarked states is $(S-t)l_J^2$ and, from the foregoing,

$$(S-t)l_J^2 = \cos^2(2J+1)\theta \leq \sin^2\theta = \frac{t}{S}.$$

In short, the iterative loop has transformed an input state that had probability $\frac{t}{S}$ of returning a marked basis state upon measurement to a superposition which will return some marked state with probability no less than $\frac{S-t}{S}$. Note that $J \leq \frac{\pi}{4\theta} \leq \frac{\pi}{4\sin\theta} = \frac{\pi}{4}\sqrt{\frac{S}{t}}$ so that $J = O\left(\sqrt{\frac{S}{t}}\right)$, as promised.

3.3 Remarks on the geometry of the algorithm

Let $W \subset \mathbb{H}_S$ be a subspace with basis $\{w\}$. Put $R_W = I - 2\sum_w |w\rangle\langle w|$. Then R_W is a reflection in the orthogonal complement, $W^\perp$, of W. Recall that for any state $|x\rangle = \sum_i a_i|i\rangle$,

$$\mathcal{U}_f : \sum_{i=0}^{S-1} a_i|i\rangle \mapsto \sum_{i=0}^{S-1}(-1)^{f(i)}a_i|i\rangle = \sum_{i\notin M} a_i|i\rangle - \sum_{i\in M} a_i|i\rangle,$$

where we assume that $f(i) = 1$ if and only if $i \in M$. Now

$$
\begin{aligned}
R_M|x\rangle &= |x\rangle - 2\sum_{i\in M}\langle i|x\rangle\,|i\rangle \\
&= \sum_{i=0}^{S-1} a_i|i\rangle - 2\sum_{i\in M}\left\langle i \,\middle|\, \sum_{j=0}^{S-1} a_j|j\rangle \right\rangle |i\rangle \\
&= \sum_{i=0}^{S-1} a_i|i\rangle - 2\sum_{i\in M} a_i|i\rangle \\
&= \sum_{i\notin M} a_i|i\rangle - \sum_{i\in M} a_i|i\rangle \\
&= \mathcal{U}_f|x\rangle,
\end{aligned}
$$

so that $\mathcal{U}_f = R_M$.

Induced by the notation introduced in the previous section, we denote by $|\Psi_j\rangle$ for $0 \leq j \leq J$ the superpositioned state under evolution by the algorithm. Then for $|\Psi_0\rangle$ and $|x\rangle = \sum_i a_i|i\rangle$ we have

$$
\begin{aligned}
-R_{|\Psi_0\rangle}|x\rangle &= 2\langle\Psi_0|x\rangle|\Psi_0\rangle - |x\rangle \\
&= 2\left\langle \frac{1}{\sqrt{S}}\sum_j\langle j| \,\middle|\, \sum_i a_i|i\rangle \right\rangle |\Psi_0\rangle - \sum_i a_i|i\rangle \\
&= \frac{2}{\sqrt{S}}\sum_j a_j \frac{1}{\sqrt{S}}\sum_i |i\rangle - \sum_i a_i|i\rangle \\
&= 2\frac{\sum_j a_j}{S}\sum_i |i\rangle - \sum_i a_i|i\rangle \\
&= \sum_i (2\mathfrak{a} - a_i)|i\rangle \\
&= \mathcal{D}|x\rangle,
\end{aligned}
$$

where $a = \sum_j a_j/S$, as before.

Suppose that the $(S-1)$-dimensional orthogonal complement to $|\Psi_0\rangle$, $|\Psi_0\rangle^\perp$, has orthonormal basis $\{|\gamma_i\rangle\}_{i=1}^{S-1}$. Then any state $|x\rangle \in \mathbb{H}_S$ may be written as $a_0|\Psi_0\rangle + \sum_i a_i|\gamma_i\rangle$ for suitable coefficient choices. From this decomposition we see that

$$-R_{|\Psi_0\rangle}|x\rangle = a_0|\Psi_0\rangle - \sum_{i=1}^{S-1} a_i|\gamma_i\rangle = R_{|\Psi_0\rangle^\perp}|x\rangle,$$

which gives us that $-R_{|\Psi_0\rangle} = R_{|\Psi_0\rangle^\perp}$. We conclude that $\mathcal{DU}_f = R_{|\Psi_0\rangle^\perp} R_M$.

Let $|\mu\rangle = \frac{1}{\sqrt{S-t}} \sum_{i \notin M} |i\rangle$ and $|m\rangle = \frac{1}{\sqrt{t}} \sum_{i \in M} |i\rangle$ and let V denote the *real* two-dimensional subspace of $\mathbb{H}_S$ with orthonormal basis $\{|\mu\rangle, |m\rangle\}$. Given $|x\rangle = a|\mu\rangle + b|m\rangle$ in V,

$$\begin{aligned}
R_M|x\rangle &= |x\rangle - 2\sum_{i\in M}\langle i|x\rangle|i\rangle \\
&= a|\mu\rangle + b|m\rangle - 2\sum_{i\in M}\langle i \mid a|\mu\rangle + b|m\rangle\rangle\,|i\rangle \\
&= \frac{a}{\sqrt{S-t}}\sum_{i\notin M}|i\rangle + \frac{b}{\sqrt{t}}\sum_{i\in M}|i\rangle \\
&\quad -2\sum_{i\in M}\left\langle i \,\middle|\, \frac{a}{\sqrt{S-t}}\sum_{j\notin M}|j\rangle + \frac{b}{\sqrt{t}}\sum_{j\in M}|j\rangle \right\rangle |i\rangle \\
&= \frac{a}{\sqrt{S-t}}\sum_{i\notin M}|i\rangle + \frac{b}{\sqrt{t}}\sum_{i\in M}|i\rangle - 2\frac{b}{\sqrt{t}}\sum_{i\in M}|i\rangle \\
&= \frac{a}{\sqrt{S-t}}\sum_{i\notin M}|i\rangle - \frac{b}{\sqrt{t}}\sum_{i\in M}|i\rangle \\
&= a|\mu\rangle - b|m\rangle
\end{aligned}$$

$$= R_{|m\rangle}|x\rangle,$$

so that, restricted to V, $R_M = R_{|m\rangle}$, which is reflection in the line $|\mu\rangle$. The matrix of $R_{|m\rangle}$ relative to the basis $\{|\mu\rangle, |m\rangle\}$ is

$$\mathtt{X} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Let $|\psi_0\rangle$ be the unit vector perpendicular to $|\Psi_0\rangle$ in V. Since

$$|\Psi_0\rangle = \frac{1}{\sqrt{S}} \sum_{i=0}^{S-1} |i\rangle = \frac{1}{\sqrt{S}} \sum_{i \notin M} |i\rangle + \frac{1}{\sqrt{S}} \sum_{i \in M} |i\rangle$$

$$= \sqrt{\frac{S-t}{S}}\, |\mu\rangle + \sqrt{\frac{t}{S}}\, |m\rangle,$$

$|\psi_0\rangle$ is given by

$$|\psi_0\rangle = \sqrt{\frac{t}{S}}\, |\mu\rangle - \sqrt{\frac{S-t}{S}}\, |m\rangle.$$

These relationships put the matrix taking $\{|\mu\rangle, |m\rangle\}$ into $\{|\Psi_0\rangle, |\psi_0\rangle\}$ as

$$\mathtt{Y} = \begin{pmatrix} \sqrt{\frac{S-t}{S}} & \sqrt{\frac{t}{S}} \\ \sqrt{\frac{t}{S}} & -\sqrt{\frac{S-t}{S}} \end{pmatrix}.$$

Note that $\mathtt{Y}^2 = I$.

The definition of reflection implies that, for arbitrary subspaces $U, W \subset \mathbb{H}_S$, $U + W$ is invariant under both R_U and R_W. From this and the foregoing discussion we see that V is invariant under the composition $-R_{|\Psi_0\rangle} R_{|m\rangle}$ and can conclude that, restricted to V, $\mathcal{DU}_f = R_{|\psi_0\rangle} R_{|m\rangle}$. In particular, $|\Psi_j\rangle \in V$ for all j. Grover's algorithm can, therefore, be viewed as the product of reflections in the lines $|\mu\rangle$ and $|\Psi_0\rangle$, operating in the real two-dimensional subspace V.

The matrix X is also the matrix of $R_{|\psi_0\rangle}$ relative to the basis $\{|\Psi_0\rangle, |\psi_0\rangle\}$. Using this observation and the change of basis matrix Y, we can calculate the matrix Z of $R_{|\psi_0\rangle}$ relative to the basis $\{|\mu\rangle, |m\rangle\}$:

$$\begin{aligned}
\mathtt{Z} &= \mathtt{YXY} \\
&= \begin{pmatrix} \sqrt{\frac{S-t}{S}} & \sqrt{\frac{t}{S}} \\ \sqrt{\frac{t}{S}} & -\sqrt{\frac{S-t}{S}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \sqrt{\frac{S-t}{S}} & \sqrt{\frac{t}{S}} \\ \sqrt{\frac{t}{S}} & -\sqrt{\frac{S-t}{S}} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{S}(S-2t) & \frac{2}{S}\sqrt{t(S-t)} \\ \frac{2}{S}\sqrt{t(S-t)} & -\frac{1}{S}(S-2t) \end{pmatrix}.
\end{aligned}$$

Finally, if G is the matrix of the Grover iteration operating on V relative to the basis $\{|\mu\rangle, |m\rangle\}$, we can obtain G as the product

$$\begin{aligned}
\mathtt{G} &= \mathtt{ZX} \\
&= \begin{pmatrix} \frac{1}{S}(S-2t) & \frac{2}{S}\sqrt{t(S-t)} \\ \frac{2}{S}\sqrt{t(S-t)} & -\frac{1}{S}(S-2t) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{S}(S-2t) & -\frac{2}{S}\sqrt{t(S-t)} \\ \frac{2}{S}\sqrt{t(S-t)} & \frac{1}{S}(S-2t) \end{pmatrix} \\
&= \begin{pmatrix} \cos 2\theta & -\sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{pmatrix},
\end{aligned}$$

where we recall that θ is defined by $\sin\theta = \sqrt{\frac{t}{S}}$.

The closed-form amplitude formula and foregoing analysis of the action on the subspace V yield a nice geometric interpretation of the algorithm's

central process. From

$$\begin{aligned} |\Psi_j\rangle &= k_j \sum_{i \in M} |i\rangle + l_j \sum_{i \notin M} |i\rangle \\ &= k_j \sqrt{t}\, |m\rangle + l_j \sqrt{S-t}\, |\mu\rangle \\ &= \sin(2j+1)\theta\, |m\rangle + \cos(2j+1)\theta\, |\mu\rangle, \end{aligned}$$

we obtain that

$$\langle \Psi_j | \mu \rangle = \cos(2j+1)\theta,$$

so that the angle between $|\Psi_j\rangle$ and $|\mu\rangle$ is $(2j+1)\theta$.

At initialization, $|\Psi_0\rangle$ is rotated from $|\mu\rangle$ by θ radians and is, therefore, nearly orthogonal to $|m\rangle$. (Recall that, for $t \ll S$, $\theta \approx \sqrt{\frac{t}{S}}$, which is very small indeed.) Each iteration of the central loop then rotates $|\Psi_j\rangle$ a further 2θ radians away from $|\mu\rangle$ until, after J iterations, $|\Psi_J\rangle$ is as colinear with $|m\rangle$ as an integer number of iterations will allow.

The matrix G shows that, consistent with the algebraic analysis, the composition of reflections in $|\mu\rangle$ and $|\Psi_0\rangle$ produces a rotation of V through an angle of 2θ for each application of Grover's loop.

3.4 An example

In this section a very small example will be worked in two fashions. First, we will iterate algorithmically, following the steps outlined in section 3.1. Following that, the equivalent matrix algebra will be examined to highlight the geometric analysis given in section 3.3.

Suppose we have a search space of size $S = 36$ with $t = 3$ marked entries. Then $\sin\theta = \sqrt{\frac{3}{36}} = \frac{\sqrt{3}}{6}$ and $J = \left\lfloor \frac{\pi}{4\theta} \right\rfloor = 2$. We identify the search space with $\mathbb{H}_{36} = \{|0\rangle, |1\rangle, \ldots, |35\rangle\}$ and, without loss of generality, assume the marked states to be $|0\rangle$, $|1\rangle$ and $|2\rangle$.

We recall that the loop first applies $\mathcal{U}_f$, which flips the sign of the coefficients of marked states, then applies $\mathcal{D}$, which transforms each coefficient a_i into $2a - a_i$.

Initialization.

$$|\Psi_0\rangle = \frac{1}{6}\sum_{i=0}^{35}|i\rangle$$

$$k_0 = \frac{1}{6} = l_0 \qquad \mathrm{P}_{\text{success}} = 3k_0^2 = \frac{1}{12} \approx 0.083$$

Iterative Loop.

Iteration 1.

$$\mathcal{U}_f|\Psi_0\rangle = -\frac{1}{6}(|0\rangle + |1\rangle + |2\rangle) + \frac{1}{6}\sum_{i=3}^{35}|i\rangle$$

$$a = \frac{1}{36}\left[3\left(-\frac{1}{6}\right) + 33\left(\frac{1}{6}\right)\right] = \frac{5}{36} \quad \Rightarrow \quad 2a = \frac{5}{18}$$

$$|\Psi_1\rangle = \mathcal{D}\mathcal{U}_f|\Psi_0\rangle$$

$$= \left(\frac{5}{18} + \frac{1}{6}\right)(|0\rangle + |1\rangle + |2\rangle) + \left(\frac{5}{18} - \frac{1}{6}\right)\sum_{i=3}^{35}|i\rangle$$

$$= \frac{4}{9}(|0\rangle + |1\rangle + |2\rangle) + \frac{1}{9}\sum_{i=3}^{35}|i\rangle$$

$$k_1 = \frac{4}{9}, \quad l_1 = \frac{1}{9} \qquad \mathrm{P}_{\text{success}} = 3k_1^2 = \frac{16}{27} \approx 0.593$$

Iteration 2.

$$\mathcal{U}_f|\Psi_1\rangle = -\frac{4}{9}(|0\rangle + |1\rangle + |2\rangle) + \frac{1}{9}\sum_{i=3}^{35}|i\rangle$$

$$a = \frac{1}{36}\left[3\left(-\frac{4}{9}\right) + 33\left(\frac{1}{9}\right)\right] = \frac{7}{108} \quad \Rightarrow \quad 2a = \frac{7}{54}$$

$$|\Psi_2\rangle = \mathcal{D}\mathcal{U}_f|\Psi_1\rangle$$

$$= \left(\frac{7}{54} + \frac{4}{9}\right)(|0\rangle + |1\rangle + |2\rangle) + \left(\frac{7}{54} - \frac{1}{9}\right)\sum_{i=3}^{35}|i\rangle$$

$$= \frac{31}{54}(|0\rangle + |1\rangle + |2\rangle) + \frac{1}{54}\sum_{i=3}^{35}|i\rangle$$

$$k_2 = \frac{31}{54}, \quad l_2 = \frac{1}{54} \qquad \mathrm{P}_{\text{success}} = 3k_2^2 = \frac{2883}{2916} \approx 0.989$$

Recall that in the analysis of section 3.2 we found that the probability of success would equal *at least* $\frac{S-t}{S}$. In this case that promise equals $\frac{11}{12} \approx 0.917$ and we see that the algorithm has done much better in this instance. The point was also made that if we allowed the algorithm to iterate beyond the optimum J, the probability of success would decline. To illustrate this we will iterate one more time.

Iteration 3.

$$\mathcal{U}_f|\Psi_2\rangle = -\frac{31}{54}(|0\rangle + |1\rangle + |2\rangle) + \frac{1}{54}\sum_{i=3}^{35}|i\rangle$$

$$\mathtt{a} = \frac{1}{36}\left[3\left(-\frac{31}{54}\right) + 33\left(\frac{1}{54}\right)\right] = -\frac{5}{162} \quad \Rightarrow \quad 2\mathtt{a} = -\frac{5}{81}$$

$$|\Psi_3\rangle = \mathcal{D}\mathcal{U}_f|\Psi_2\rangle$$

$$= \left(-\frac{5}{81} + \frac{31}{54}\right)(|0\rangle + |1\rangle + |2\rangle) + \left(-\frac{5}{81} - \frac{1}{54}\right)\sum_{i=3}^{35}|i\rangle$$

$$= \frac{83}{162}(|0\rangle + |1\rangle + |2\rangle) - \frac{13}{162}\sum_{i=3}^{35}|i\rangle$$

$$k_3 = \frac{83}{162}, \quad l_3 = -\frac{13}{162} \qquad \mathrm{P}_{\text{success}} = 3k_3^2 = \frac{20667}{26244} \approx 0.787$$

Matrix algebra on V

For the given parameters $S = 36$ and $t = 3$, let the basis $\{|\mu\rangle, |m\rangle\}$ of V be defined by

$$|\mu\rangle = \frac{1}{\sqrt{33}}\sum_{i=3}^{35}|i\rangle \qquad \text{and} \qquad |m\rangle = \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle)\,.$$

The matrix of the Grover iteration relative to this basis is given by

$$\mathtt{G} = \frac{1}{6}\begin{pmatrix} 5 & -\sqrt{11} \\ \sqrt{11} & 5 \end{pmatrix}.$$

Initialization.

$$|\Psi_0\rangle = \frac{1}{6}\sum_{i=0}^{35}|i\rangle = \frac{1}{6}\left(\sqrt{33}\,|\mu\rangle + \sqrt{3}\,|m\rangle\right) = \frac{1}{6}\begin{pmatrix}\sqrt{33}\\ \sqrt{3}\end{pmatrix}.$$

The probability of successfully retrieving a marked state at this stage is given by the square of the coefficient of $|m\rangle$, which is $\left(\frac{\sqrt{3}}{6}\right)^2 = \frac{1}{12} = 3k_0^2$.

Iterative Loop.

Iteration 1.

$$\frac{1}{6}\begin{pmatrix}5 & -\sqrt{11}\\ \sqrt{11} & 5\end{pmatrix}\frac{1}{6}\begin{pmatrix}\sqrt{33}\\ \sqrt{3}\end{pmatrix} = \frac{1}{9}\begin{pmatrix}\sqrt{33}\\ 4\sqrt{3}\end{pmatrix} = \frac{\sqrt{33}}{9}|\mu\rangle + \frac{4\sqrt{3}}{9}|m\rangle = |\Psi_1\rangle$$

The probability of success is $\left(\frac{4\sqrt{3}}{9}\right)^2 = \frac{16}{27} = 3k_1^2$.

Iteration 2.

$$\frac{1}{6}\begin{pmatrix}5 & -\sqrt{11}\\ \sqrt{11} & 5\end{pmatrix}\frac{1}{9}\begin{pmatrix}\sqrt{33}\\ 4\sqrt{3}\end{pmatrix} = \frac{1}{54}\begin{pmatrix}\sqrt{33}\\ 31\sqrt{3}\end{pmatrix} = \frac{\sqrt{33}}{54}|\mu\rangle + \frac{31\sqrt{3}}{54}|m\rangle = |\Psi_2\rangle$$

The probability of success reaches a maximum at $\left(\frac{31\sqrt{3}}{54}\right)^2 = \frac{2883}{2916} = 3k_2^2$.

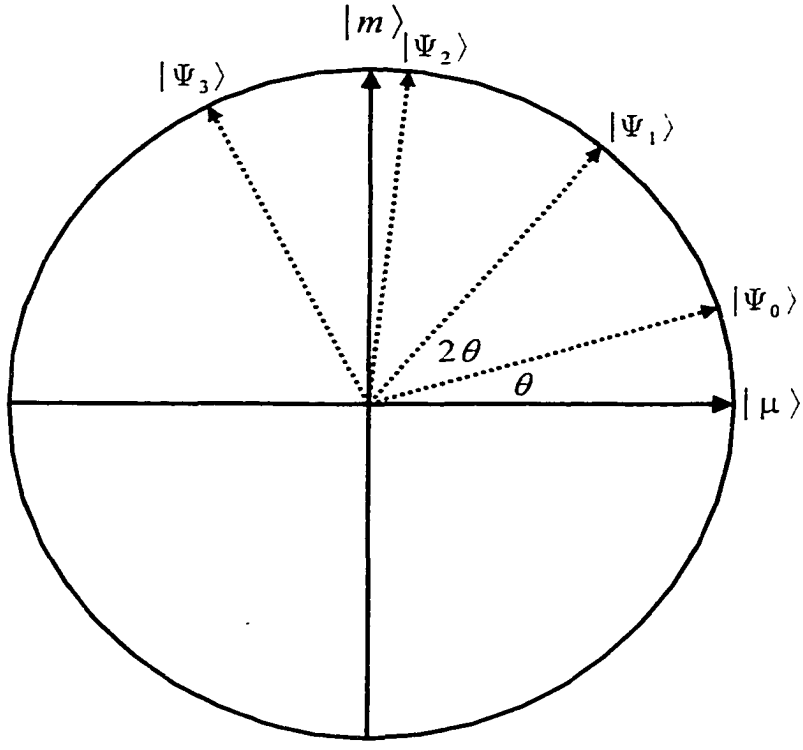
Iteration 3.

$$\frac{1}{6}\begin{pmatrix}5 & -\sqrt{11}\\ \sqrt{11} & 5\end{pmatrix}\frac{1}{54}\begin{pmatrix}\sqrt{33}\\ 31\sqrt{3}\end{pmatrix} = \frac{1}{162}\begin{pmatrix}-13\sqrt{33}\\ 83\sqrt{3}\end{pmatrix}$$

$$= \frac{-13\sqrt{33}}{162}|\mu\rangle + \frac{83\sqrt{3}}{162}|m\rangle = |\Psi_3\rangle$$

The probability of success declines to $\left(\frac{83\sqrt{3}}{162}\right)^2 = \frac{20667}{26244} = 3k_3^2$.

The following figure provides a visualization of the geometry of this example:



3.5 Grover's algorithm applied to an unknown number of marked states

In [4] the following quantum algorithm is given to solve the search problem when the database contains t marked states, where t is unknown and it is assumed that $1 \leq t \leq \frac{3}{4}S$:

1. Initialize $c = 1$ and choose λ such that $1 < \lambda < \frac{4}{3}$. For concreteness we can put $\lambda = \frac{8}{7}$.

2. Choose an integer j uniformly at random such that $0 \leq j < c$.

3. Apply j iterations of Grover's algorithm starting from initial state $|\Psi_0\rangle$.

4. Measure the register and let $|i\rangle$ be the outcome.

5. If $f(i) = 1$, the problem is solved and the procedure halts.

6. Otherwise, set c to $\min\{\lambda c, \sqrt{S}\}$ and return to step 2.

This algorithm is shown in [4] to find a marked state in $O\left(\sqrt{\frac{S}{t}}\right)$ expected time. For $t > \frac{3}{4}S$, conventional sampling can readily be used in expected constant time for any given probability of success.

4 The NTRU Cryptosystem

4.1 Description of the system

The NTRU public-key cryptosystem is specified by six integer parameters $(n, p, q, d_k, d_g, d_\phi)$ and operates within the ring $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$. The integers p and q are such that $\gcd(p, q) = 1$ and q is much greater than p. An element $r \in R$ may be specified by the vector of its coordinates: $r = (r_0, r_1, \ldots, r_{n-1})$. Reduction of an element of R modulo i will always mean reduction of its coefficients into the interval $\left(-\frac{i}{2}, \frac{i}{2}\right]$.

Messages are drawn from the set $\mathsf{S}_m \subset R$, where

$$\mathsf{S}_m = \{m \in R : m \text{ is reduced modulo } p\}.$$

Three additional subsets S_k, S_g and S_ϕ of R are required for the system. For positive integers d_1 and d_2 let

$$\mathsf{S}(d_1, d_2) = \{a \in R : a \text{ has } d_1 \text{ coefficients equal to } 1,\ d_2 \text{ coefficients equal to } -1 \text{ and } n - d_1 - d_2 \text{ coefficients equal to } 0\}.$$

Then we may specify the other subsets by $\mathsf{S}_k = \mathsf{S}(d_k, d_k - 1)$, $\mathsf{S}_g = \mathsf{S}(d_g, d_g)$ and $\mathsf{S}_\phi = \mathsf{S}(d_\phi, d_\phi)$.

To generate a key pair, a user uniformly selects $k \in \mathsf{S}_k$ and $g \in \mathsf{S}_g$ and computes $K_p, K_q \in R$ such that $kK_p = 1 \pmod p$ and $kK_q = 1 \pmod q$. The public key is $h = gK_q \pmod q$ and the private key are k and g.

Remark. The inverses K_p and K_q are shown in [22] to exist with probability very near 1 for suitably chosen system parameters. The parameters

suggested in [20], where different sets are given according to varying security requirements, allow for this probability to be *at least* $1 - 10^{-25}$ and often much higher.

An encrypted message has the form $e \equiv p\phi h + m \pmod{q}$, where ϕ is chosen uniformly from S_ϕ. To decrypt, a legitimate recipient computes

$$\begin{aligned} ke &\equiv kp\phi h + km \pmod{q} \\ &\equiv kp\phi g K_q + km \pmod{q} \\ &\equiv p\phi g + km \pmod{q} \end{aligned}$$

Denote by b the *non-modular* expression $p\phi g + km$. Assume that the coefficients of b lie in $\left(-\frac{q}{2}, \frac{q}{2}\right]$. Then reduction modulo q during the decryption phase will recover these coefficients faithfully as integers. The message can then be accurately decrypted by computing $m \equiv bK_p \pmod{p}$.

For any $a \in R$, define $|a|_\infty$ to be the length of the interval containing the coefficients of a. As the notation implies, this function is treated as a norm in [20], where it is noted that for decryption to occur correctly, $|b|_\infty < q$ must hold. The event $|b|_\infty \geq q$ is called a *gap failure* and the event that the largest or smallest coefficient of b falls outside of $(-\frac{q}{2}, \frac{q}{2}]$ (in the absence of gap failure) is called a *wrapping failure*. Gap failure is unrecoverable, whereas wrapping failure may be corrected by successive computations of

$$(b \pmod{q} + (i, i, \ldots, i)) \pmod{q} - (i, i, \ldots, i) \qquad \text{for } i = \pm 1, \pm 2, \ldots,$$

where we assume the existence of a verification scheme to determine the

correct adjustment.

A second norm central to the design and analysis of NTRU is defined in [20, 24] as follows for any $a \in R$:

$$[\![a]\!] = \left[\sum_{i=0}^{n-1} (a_i - \mathfrak{a})^2\right]^{\frac{1}{2}}, \quad \text{where} \quad \mathfrak{a} = \frac{1}{n}\sum_{i=0}^{n-1} a_i.$$

Note that $[\![a]\!]/\sqrt{n}$ is the standard deviation of the coefficients of a.

This norm is used as an approximation for the "infinity" norm defined above in order to derive parameters for the system that will insure high reliability with regard to gap and wrapping failure. The particular detail of interest to us is the following approximation given in [24]:

$$[\![b]\!]^2 = [\![p\phi g + km]\!]^2 \approx p^2 [\![\phi]\!]^2 [\![g]\!]^2 + [\![k]\!]^2 [\![m]\!]^2.$$

We note that the quantities on the right-hand side of this approximation are almost all definitively computable. Indeed,

$$[\![\phi]\!]^2 = 2d_\phi,$$
$$[\![g]\!]^2 = 2d_g, \text{ and}$$
$$[\![k]\!]^2 = 2d_k - 1 - \frac{1}{n}.$$

As for $[\![m]\!]^2$, we use an average value. Specifically, for $p = 3$, as it is for all parameter sets currently contemplated by the system's designers, we assume that the average message will have one-third of its coefficients equal to each of 1, -1 and 0. This gives an average value for $[\![m]\!]^2$ of $\frac{2}{3}n$.

Heuristic arguments given in [20] and experimental results reported in [23] imply that parameter sets derived from the approximation given above will

produce NTRU systems protected against gap and wrapping failures with high probability. In this inquiry, these claims as well as the validity of the approximation will be assumed as given.

In [20, 24] the existence of spurious private keys is discussed. It is observed that any $k' \in \mathsf{S}_k$ will serve as a decoding key, as long as k' satisfies

$$|b'|_\infty = |p\phi g' + k'm|_\infty < q,$$

where $g' = k'h$. For such k', $k'e \equiv p\phi g' + k'm \pmod{q}$. If $p\phi g' + k'm$ has coefficients lying in $(-\frac{q}{2}, \frac{q}{2}]$, then, as above, reduction of $k'e$ modulo q allows us to recover faithfully $k'm \pmod{p}$. From knowledge of k', K'_p can be calculated and the decryption completed by computing $m \equiv K'_p k'm \pmod{p}$.

In [24] it is specifically noted that for $b' = p\phi g' + k'm$ such that $[\![b']\!]^2$ is comparable to $[\![b]\!]^2$, the pair (k', g') may be used as a decoding key as described above. From the approximation given, we require that

$$p^2\,[\![\phi]\!]^2\,[\![g']\!]^2 + [\![k']\!]^2\,[\![m]\!]^2 \leq p^2\,[\![\phi]\!]^2\,[\![g]\!]^2 + [\![k]\!]^2\,[\![m]\!]^2.$$

Since both k and k' are in S_k, $[\![k']\!]^2 = [\![k]\!]^2$ so that the preceding inequality reduces to $[\![g']\!]^2 \leq [\![g]\!]^2 = 2d_g$. It is this measure that will be of central importance to us.

Remark. In [24] it is observed that if $[\![b']\!]^2$ is even two or three times larger than $[\![b]\!]^2$, useful partial decoding keys can be obtained. This is especially useful if several such keys can be found. We will ignore this possibility, however, in light of this observation, merely note that the search criterion

used in Grover's algorithm could be loosened to find such partial keys.

4.2 Conventional attack

A lattice based attack on NTRU described in [20, 24] creates a lattice basis $\{e_0, e_1, \ldots, e_{2n-1}\}$, where e_i is the i^{th} row of the following $2n \times 2n$ matrix:

$$\left(\begin{array}{cccc|cccc} \alpha & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{n-1} \\ 0 & \alpha & \cdots & 0 & h_{n-1} & h_0 & \cdots & h_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha & h_1 & h_2 & \cdots & h_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right)$$

The "balancing" parameter α will be specified below.

Since $h = gK_q \pmod q$, the lattice contains the vector $\tau = (\alpha k, g) \pmod q$, by which is meant a vector with its first n coordinates equal to the coefficients of k scaled by α and its last n coordinates equal to the coefficients of g. (Note that $\tau = \sum_{i=0}^{n-1} k_i e_i + \sum_{i=n}^{2n-1} x_{i-n} e_i$, where $x = (x_0, x_1, \ldots, x_{n-1})$ is an arbitrary integer vector representing multiples of q.)

To derive a value for α which creates the most favorable conditions for an attacker, the authors of [20, 23] employ the Gaussian heuristic as a guide. This heuristic gives an expected size of the smallest vector in a random lattice of dimension d and determinant D between

$$D^{1/d}\sqrt{\frac{d}{2\pi e}} \quad \text{and} \quad D^{1/d}\sqrt{\frac{d}{\pi e}}.$$

In this case $d = 2n$ and $D = \alpha^n q^n$, so the expected smallest length is not much larger than

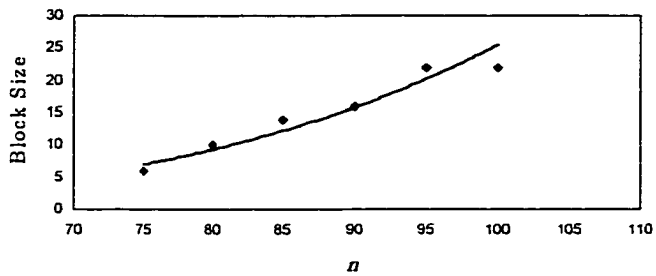
$$s = \sqrt{\frac{n\alpha q}{\pi e}}.$$

Using s as a guide, the balancing parameter is chosen to obtain a minimum for the ratio $\|\tau\|/s$. Squaring this ratio and observing that $\|\tau\|^2 = \alpha^2\|k\|^2 + \|g\|^2$, we must choose α so as to find the minimum of $\alpha\|k\|^2 + \frac{1}{\alpha}\|g\|^2$. This occurs when $\alpha = \|g\|/\|k\|$. Note that these quantities are public; indeed, $\|k\| = \sqrt{2d_k - 1}$ and $\|g\| = \sqrt{2d_g}$ for all $k \in \mathsf{S}_k$ and $g \in \mathsf{S}_g$.

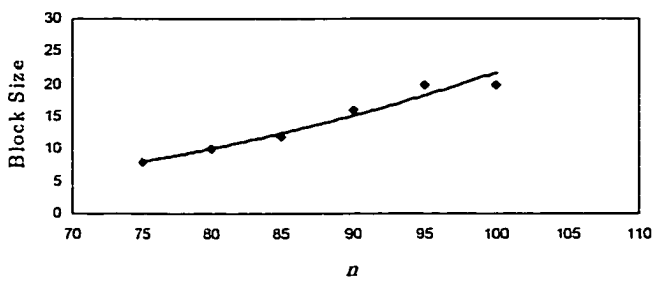
NTRU allows for practical implementation of system sizes which resist attack while remaining feasible. In [20], NTRU implementations are reported to compare favorably with comparable RSA instances with respect to key generation, encryption and decryption speeds. Reports in [20] of extensive testing of the lattice attack described here, using various improvements of the LLL algorithm, suggest that defeating the NTRU system as specified requires block sizes exponential in n. The experiments were run on three parameter sets ("moderate", "high" and "highest" security), with the size of n varied.

The following plots were constructed from data presented in [20], where the block size is that required to find the target vector successfully. In each case, the data appears to fit the superimposed exponential trend line fairly well.

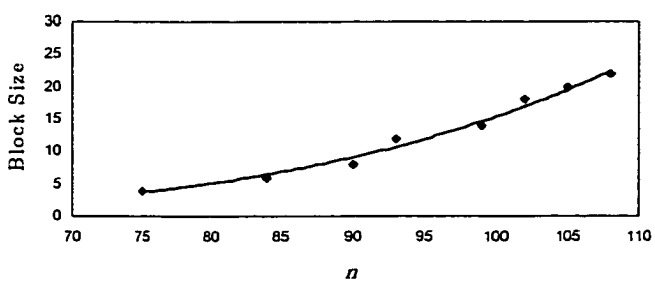
Moderate security parameters



High security parameters



Highest security parameters



The authors of [20] extrapolate from these trends and, based upon the actual time required to find the target vectors in their experiments, derive the following time estimates for breaking various parameter settings of NTRU using a modest desktop computer:

Security level		Time
Moderate	$(n, p, q, d_k, d_g, d_\phi) = (107, 3, 64, 15, 12, 5)$	9 days
High	$(n, p, q, d_k, d_g, d_\phi) = (167, 3, 128, 61, 20, 18)$	380 years
Highest	$(n, p, q, d_k, d_g, d_\phi) = (503, 3, 256, 216, 72, 55)$	$6.2 \cdot 10^{27}$ years

If we accept these estimates, the so-called highest settings would certainly allow for improvements in computing power of many orders of magnitude while still maintaining Cold War strength security. Of course, such computational improvements would facilitate scaling up the cryptosystem size as well.

The experiments also reportedly indicate that the algorithms must work just as hard to find vectors of comparable length to the specific private key, vectors which could then yield a spurious key. It is this experience which leads the creators of NTRU to disregard the existence of spurious keys as problematic for the system. We will see that this assumption may not hold under quantum attack.

4.3 Attack on NTRU with Grover's algorithm

The situation presented by NTRU is one where some unknown t spurious points exist within the sample space S_k, any of which may function as a decoding key. The central loop of the algorithm operates on basis states of $\mathbb{H}_S$, where

$$S = |\mathsf{S}_k| = \binom{n}{d_k}\binom{n-d_k}{d_k-1}.$$

We can apply the algorithm of section 3.5 to this problem, under the mild condition that $1 \leq t \leq \frac{3}{4}S$. We assume a one-to-one correspondence

$$\{0, 1, \ldots, S-1\} \rightarrow \mathsf{S}_k : i \mapsto k^{(i)}$$

For each candidate $k^{(i)}$ we compute $g^{(i)} = k^{(i)}h \pmod{q}$ and put $f(i) = 1$ if and only if $[\![g^{(i)}]\!]^2 \leq 2d_g$.

The expected run time for the algorithm is $O\left(\sqrt{\frac{S}{t}}\right)$, which, regardless of the value of t, is no worse than $O\left(n^{d_k}\right)$.

4.4 Discussion

As just noted above, the attack on NTRU using Grover's algorithm has an expected run time polynomial in the parameter n. The experimental evidence offered in [20] suggests that this is an improvement over currently feasible implementations of lattice reduction technology. The creators of NTRU have already proposed a non-commutative version of the cryptosystem [21], which they feel should greatly confound the efforts of lattice basis algorithms to converge to appropriate solutions. The system is essentially that presented above, however, the platform changes from $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$ to some non-commutative ring R containing a commutative subring R_0. The authors suggest as a concrete example the ring $R = \mathbb{Z}[D_{2n}]$, the integral group ring of the dihedral group of order $2n$. The system is so structurally similar to the original that it can be attacked by Grover's algorithm exactly as contemplated here. The strength of Grover's algorithm in this regard is that it depends only on the simple membership evaluation assigned by $\mathcal{U}_f$ to

basis elements of $\mathbb{H}_S$. The underlying algebraic structure of the cryptosystem matters not as long as there is a computable function which accurately marks appropriate basis states.

In addition, a weakness of both versions of NTRU is the existence of spurious keys. While it may be true that all lattice vectors useful to a potential attacker employing basis reduction lie below some size threshold, making finding one as difficult as finding any other, the expected time of Grover's algorithm clearly indicates a dependence on the number of solutions. Of particular interest is how the ratio S/t grows with S. This is not known, however, should it be true that this ratio grows much more slowly than S or, worse yet for the security of the cryptosystem, that it is bounded by some constant, this fact could thwart attempts at increasing security by scaling up the system size.

5 The Chor–Rivest System

5.1 Subset Sum Problems

In its most general form, the *knapsack* or *subset sum problem* (SSP) can be stated as follows [28]: Given sets of positive numbers $A = \{a_0, a_1, \ldots, a_{n-1}\}$ and $B = \{b_0, b_1, \ldots, b_{n-1}\}$, and a nonnegative number s, find a vector of nonnegative integers $X = (x_0, x_1, \ldots, x_{n-1})$ which

obtains a maximum for $\sum_{i=0}^{n-1} x_i b_i,$

subject to $\sum_{i=0}^{n-1} x_i a_i \leq s.$

In practice $A = B$ is often the case, which is the assumption made here.

The general problem poses no decision question; a solution clearly exists and the problem becomes one of finding it. Indeed, in many applications it is sufficient to find good approximations to a solution of the problem as posed above. The interest here, however, is in solutions to the following exact version of the problem: Given a nonnegative integer s and a set of positive integers $A = \{a_0, a_1, \ldots, a_{n-1}\}$, determine whether some and which binary vectors (entries restricted to 0 or 1) $X = (x_0, x_1, \ldots, x_{n-1})$ exist such that

$$\sum_{i=0}^{n-1} x_i a_i = s.$$

The term knapsack derives from the association of the integer s with the capacity of a knapsack and the integers a_i with the sizes of various objects. The question is then equivalent to asking whether some subset of the objects completely fills the knapsack.

If an efficient method were available simply to determine whether such a solution existed, finding a solution would follow quite readily. To test whether a solution existed with $x_0 = 1$, for example, we could determine whether a solution existed to

$$\sum_{i=1}^{n-1} x_i a_i = s - a_0.$$

If no such solution existed, we could deduce that $x_0 = 0$. A complete solution could then be found by repeating this process for $x_1, x_2, \ldots, x_{n-1}$. However, the decision question associated with this problem is known to be NP-complete [28].

5.2 Description of the system

The Chor–Rivest knapsack system [26] is designed to encrypt binary vectors of length n and fixed Hamming weight w, *i.e.*, with exactly w entries equal to one and $n - w$ entries equal to zero. This condition poses no serious restriction, as an efficient algorithm is presented in [26] to transform arbitrary binary streams into strings of fixed length and weight. The knapsack chosen for this system is described in the following version of a theorem of

Bose and Chowla [25]:

Theorem (Bose–Chowla): Let p be prime and $w \geq 2$. Then there exists a set of integers $A = \{a_0, a_1, \ldots, a_{p-1}\}$ such that:

1. $0 \leq a_i \leq p^w - 1$ $(i = 0, 1, \ldots, p-1)$.

2. If $(x_0, x_1, \ldots, x_{p-1})$ and $(y_0, y_1, \ldots, y_{p-1})$ are two distinct vectors with nonnegative integer entries such that $\sum x_i = v = \sum y_i$, where $v \leq w$, then $\sum x_i a_i \neq \sum y_i a_i$.

The construction takes place in GF(p^w). Enumerate the image of GF(p) in GF(p^w) as $\alpha_0, \alpha_1, \ldots, \alpha_{p-1}$. If z is any element of GF(p^w) which is algebraic of degree w over GF(p) and g is a generator of GF(p^w)*, then $a_i = \log_g(z + \alpha_i)$ $(i = 0, 1, \ldots, p-1)$.

Chor and Rivest's system can now be described.

System Generation:

1. Pick p and $w \leq p$.

2. Find a random irreducible monic polynomial $f(z)$ of degree w in GF(p)[z]. Arithmetic in GF(p^w) is represented by operations in GF(p)[z]/$\langle f(z) \rangle$.

3. Pick a random generator g of GF(p^w)*.

4. Compute $c_i = \log_g(z + \alpha_i)$, $\forall \alpha_i \in$ GF(p).

5. Choose a random $\wp \in S_p$. Put $b_i = c_{\wp(i)}$.

6. Pick d at random such that $0 < d < p^w - 1$. Set $a_i = b_i + d$.

Public Key: $A = \{a_0, a_1, \ldots, a_{p-1}\}$, p and w.

Private Key: $f(z)$, g, d and $\wp^{-1}$.

Encryption: To encrypt a binary vector $X = (x_0, x_1, \ldots, x_{p-1})$ of weight w, compute $s \equiv \sum x_i a_i \pmod{p^w - 1}$.

Decryption: After receiving s, the owner of the private key computes $q(z) \equiv g^r \pmod{f(z)}$, where $r = s - wd$. It can be shown in a straightforward fashion that from these manipulations the following equality holds:

$$q(z) + f(z) = \prod_{i=0}^{p-1} (z + \alpha_{\wp(i)})^{x_i}$$

By factoring the left-hand side of this equation (requiring at most p substitutions), we obtain which of the x_i are non-zero.

5.3 Conventional attack

The development of knapsack cryptosystems spawned much interest in efficient methods of finding solutions to the problem posed in the previous section. The most well developed technique relies upon lattice basis reduction, the success of which hinges significantly on the structure of the set A. In particular, the cardinality of this set relative to the size of its largest member [32, 29] bears directly on the ease of obtaining solutions to an SSP drawn from it.

The *density* of a set of positive integers $A = \{a_0, a_1, \ldots, a_{n-1}\}$ shall mean $\frac{n}{\log(\max A)}$, where the logarithm is to the base 2 [29]. While the understanding

of what constitutes a critical density as far as determining problem difficulty has changed over time [29, 32, 34], it is generally accepted that the greater the density of A, the more resistant instances of the SSP are to lattice reduction techniques.

Attacks on knapsack cryptosystems based on lattice basis techniques were introduced by Lagarias and Odlyzko [30] and are presented here in a modified version of the original [32]. When presented with the set A and an integer s, the attacker forms the lattice basis $\{e_0, e_1, \ldots, e_p\}$, where e_i is the i^{th} row of the matrix

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & Pa_0 \\ 0 & 1 & \cdots & 0 & Pa_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & Pa_{p-1} \\ \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & Ps \end{pmatrix}$$

and $P > \frac{1}{2}\sqrt{p}$. If $X = (x_0, x_1, \ldots, x_{p-1})$ is the encrypted vector to be determined, then $Y = (y_0, y_1, \ldots, y_{p-1}, 0)$, where $y_i = x_i - \frac{1}{2}$, is in the lattice spanned by $e_0, e_1, \ldots, e_p$ ($Y = \sum_{i=0}^{p-1} x_i e_i - e_p$). It is shown in [32] that if A has density less than 0.9408 ..., then with high probability Y is the shortest vector in this lattice. A call to the lattice oracle can then return Y with high probability and in polynomial time, from which the attacker can recover X. Note that this method of attack does not require that we recover the original set from which the published set A was derived.

In [34], Schnorr and Hörner report an improved algorithm which appears to render sets of density greater than one vulnerable to lattice basis attack, including successful attacks on a Chor–Rivest knapsack system of modest

size ($p = 103, w = 12$) and encouraging beginnings on an attack of a system of greater size ($p = 151, w = 16$). Schnorr and Hörner also report that an alternative algorithm of Ritter [33] has been used successfully to attack a Chor–Rivest system with $p = 103$ and $w = 12$.

5.4 Attack on Chor–Rivest with Grover's algorithm

The knapsack problem posed by the Chor–Rivest cryptosystem readily fits the algorithm described in section 3.1. Given system parameters p and w, let H_p^w denote the set of binary strings of length p and weight w. The algorithm is run on basis elements of $\mathbb{H}_S$, where $S = \left|\mathrm{H}_p^w\right| = \binom{p}{w}$. We assume a one-to-one correspondence

$$\{0, 1, \ldots, S-1\} \to \mathrm{H}_p^w : i \mapsto i'$$

If $|i\rangle$ is a basis element of $\mathbb{H}_S$ and s is the target sum, we set $f(i) = 1$ if and only if the elements of A corresponding to the positions of i' holding a 1 sum to s. Note that in this situation there is a unique i for which $f(i) = 1$. The run time for the algorithm is $O\left(\sqrt{S}\right)$, which is no worse than $O\left(p^{\frac{w}{2}}\right)$.

5.5 Discussion

The dimension of the derived lattice for knapsack problems is $n + 1$, where n is the size of the set A. Lattice attacks on Chor–Rivest will, therefore, have running times polynomial in $p + 1$. Note that this is comparable, at least in terms of fundamental complexity classes, to the runtime for Grover's algorithm.

The current state of knowledge and technology has favored lattice basis reduction attacks on Chor–Rivest. Unfortunately, this cryptosystem cannot be easily scaled to avoid lattice attacks since key generation depends upon solving the discrete logarithm problem in GF(p^w)*. Improvements to the basic LLL algorithm [33, 34] have shown the Chor–Rivest system to be vulnerable for most practical parameter choices under current operating conditions. However, the example serves an important purpose. It is not at all unreasonable to imagine either a system similar to Chor–Rivest which can be sized beyond conventional techniques or, more intriguing yet, a knapsack system based on a non-commutative platform which, as in the case of the NTRU system, may create fatal convergence problems for conventional lattice-based attacks. Grover's algorithm would still provide a tool to compromise such systems under these scenarios.

6 Future directions

This paper has presented a framework for the application of Grover's quantum search algorithm to diophantine problems such as breaking cryptographic systems based upon the subset sum problem of Chor–Rivest and the mixed-modulus polynomial algebra of NTRU. By the time quantum machines become available to run such algorithms, many details must be filled in. Among these are specific implementations of (1) the unitary analog, $\mathcal{U}_f$, of the membership function f, (2) equiprobable superpositions of the canonical basis states of arbitrarily sized subspaces of $\mathbb{H}_N$ and (3) correspondence between the actual search spaces and the basis states $|i\rangle$ $(i = 0, 1, \ldots, S-1)$.

In addition, the general expectation for quantum hardware is that it will require distributed computing for the manipulation of registers of any meaningful size. While it is not conceptually difficult to imagine how the quantum search presented here could be broken into subproblems and distributed among different processors, a careful working through of implementation details is required before such algorithms go online.

Profitable inquiry of a different focus could be in the design of non-commutative knapsack systems. The key ingredient needed here is an analog to the Bose–Chowla theorem for a non-commutative structure. In the context of NTRU, further investigation into the ratio S/t would have meaningful consequences for the security of this cryptosystem in a quantum environment.

Finally, turning the problem on its head, the construction of diophatine systems resistant to quantum attack would be immensely important.

References

[1] P. Benioff, Quantum mechanical hamiltonian models of Turing machines, *Jornal of Statistical Physics* **29** (1982), 515-546.

[2] A. Berthiaume, Quantum Computation, in *Complexity Theory Retrospective II*, L. Hemaspaandra and A.L. Selman, eds., Springer-Verlag, 1997.

[3] J. Blank, P. Exner and M. Havlíček, *Hilbert Space Operators in Quantum Physics*, American Institute of Physics Press, 1994.

[4] M. Boyer, G. Brassard, P. Hoyer and A. Tapp, Tight Bounds on Quantum Searching, *Proc.* 4^{th} *Workshop on Physics and Comput.*, T. Toffoli, M. Biafore and J. Leao, eds., New England Complex Systems Institute, 1996.

[5] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, *Proc. R. Soc. Lond. A* **400** (1985), 97-117.

[6] D. Deutsch and R. Josza, Rapid solution of problems by quantum computation, *Proc. R. Soc. Lond. A* **439** (1992), 553-558.

[7] C. Dürr and P. Høyer, A quantum algorithm for finding the minimum, LANL preprint quant-ph/9607014, available at xxx.lanl.gov.

[8] R. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21** (1982), 467-488.

[9] R. Feynman, R. Leighton and M. Sands, *The Feynman Lectures on Physics, Vol. III, Quantum Mechanics*, Addison-Wesley, 1965.

[10] L. Grover, A fast quantum mechanical algorithm for database search, *Proc.* 28^{th} *Ann. ACM Symp. on Theory of Comput.*, 1996, 212-219.

[11] R. Jozsa, Searching in Grover's Algorithm, LANL preprint quant-ph/9901021, available at xxx.lanl.gov.

[12] R. Jozsa, Quantum Algorithms and the Fourier Transform, LANL preprint quant-ph/9707033, available at xxx.lanl.gov.

[13] R. Jozsa, Entanglement and Quantum Computation, in *Geometric Issues in the Foundations of Science*, S. Huggett, L. Mason, K.P. Tod, S.T. Tsou and N.M.J. Woodhouse, eds., Oxford University Press, 1997.

[14] A. Y. Kitaev, Quantum measurements and the Abelian Stabilizer Problem, LANL preprint quant-ph/9511026, available at xxx.lanl.gov.

[15] S. Lomonaco, Jr., A Rosetta Stone for Quantum Mechanics: Lecture Notes for The AMS Short Course on Quantum Computation, Washington, D.C., January 2000.

[16] M. Mosca and A. Ekert, The Hidden Subgroup Problem and Eigenvalue Estimation on a Quantum Computer, LANL preprint quant-ph/9903071, available at xxx.lanl.gov.

[17] A. Pittenger, *An Introduction to Quantum Computing Algorithms*, Birkhäuser, 2000.

[18] J. Preskill, Lecture Notes for *Quantum Information and Computation*, a course given at the California Institute of Technology, 1998.

[19] P. Shor, Polynomial-time algorithms for prime factorisation and discrete logarithms on a quantum computer, *SIAM J. Comput.* **26** (1997), 1484-1510.

[20] J. Hoffstein, J. Piper and J.H. Silverman, NTRU: A new high speed public key cryptosystem, in *Algorithmic Number Theory (ANTS III)*, J.P Buhler, ed., *LNCS* **1423**, Springer-Verlag, 1998, 267-288.

[21] J. Hoffstein and J.H. Silverman, A non-commutative version of the NTRU public key cryptosystem, available at www.ntru.com.

[22] J. Silverman, NTRU Cryptosystems Technical Report No. 9: Invertibility in Truncated Polynomial Rings, available at www.ntru.com.

[23] J. Silverman, NTRU Cryptosystems Technical Report No. 11: Wraps, Gaps and Lattice Constants, available at www.ntru.com.

[24] D. Coppersmith and A. Shamir, Lattice Attacks on NTRU, in *Proceedings of Eurocrypt '97*, *LNCS* **1233**, Springer-Verlag, 1997, 52-61.

[25] R.C. Bose and S. Chowla, Theorems in the Additive Theory of Numbers, *Comment. Math. Helvet.* **37** (1962), 141-147.

[26] B. Chor and R. Rivest, A Knapsack-Type Public Key Cryptosystem Based on Arithmetic in Finite Fields, *IEEE Trans. Inform. Theory* **IT-34** (1988), 901-909.

[27] R.C. Merkle and M.E. Hellman, Hiding Information and Signatures in Trapdoor Knapsacks, *IEEE Trans. Inform. Theory* **IT-24** (1978), 525-530.

[28] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, Wiley, 1990.

[29] A.M. Odlyzko, The Rise and Fall of Knapsack Cryptosystems, in *Cryptology and Computational Number Theory*, C. Pomerance, ed., *Proc. Symp. Appl. Math.* **42**, Amer. Math. Soc., 1990, 75-88.

[30] J.C. Lagarias and A.M. Odlyzko, Solving Low-Density Subset Sum Problems, *J. Assoc. Comp. Mach.* **32** (1985), 229-246.

[31] A.M. Frieze, On the Lagarias-Odlyzko Algorithm for the Subset Sum Problem, *SIAM J. Comput.* **15** (1986), 536-539.

[32] M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.P. Schnorr and J. Stern, Improved Low-Density Subset Sum Algorithms, in *computational complexity 2*, Birkhauser, 1992, 111-128.

[33] H. Ritter, Breaking Knapsack Cryptosystems by ℓ_∞–norm Enumeration, in *Proc. 1st Intl. Conf. Theory and Appl. of Cryptology—Pragocrypt '96* (1996), 480-492.

[34] C.P. Schnorr and H.H. Hörner, Attacking the Chor–Rivest Cryptosystem by Improved Lattice Reduction, in *Advances in Cryptology—Eurocrypt '95*, L. Guillou and J.-J. Quisquater, eds., *LNCS* **921**, Springer-Verlag, 1995, 1-12.

[35] M. Kaib and H. Ritter, Block Reduction for Arbitrary Norms, *Technical Report*, Universität Frankfurt, 1994.

[36] A.K. Lenstra, H.W. Lenstra, Jr. and L. Lovász, Factoring Polynomials with Rational Coefficients, *Math. Ann.* **261** (1982), 515-534.

[37] C.P. Schnorr, A More Efficient Algorithm for Lattice Basis Reduction, *J. Algorithms* **9** (1988), 47-62.

[38] C.P. Schnorr and M. Euchner, Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems, *Math. Prog.* **66** (1994), 181-194.