A classical one-way function to confound quantum adversaries

Cristopher Moore
moore@cs.unm.edu
University of New Mexico
and the Santa Fe Institute

Alexander Russell
acr@cse.uconn.edu
University of Connecticut

Umesh Vazirani
vazirani@cs.berkeley.edu
U. C. Berkeley

January 18, 2007

Abstract

The promise of quantum computation and its consequences for complexity-theoretic cryptography motivates an immediate search for cryptosystems which can be implemented with current technology, but which remain secure even in the presence of quantum computers. Inspired by recent negative results pertaining to the nonabelian hidden subgroup problem, we present here a classical algebraic function $f_V(M)$ of a matrix M which we believe is a one-way function secure against quantum attacks. Specifically, inverting f_V reduces naturally to solving a hidden subgroup problem over the general linear group (which is at least as hard as the hidden subgroup problem over the symmetric group). We also demonstrate a reduction from Graph Isomorphism to the problem of inverting f_V; unlike Graph Isomorphism, however, the function f_V is random self-reducible and therefore uniformly hard.

These results suggest that, unlike Shor's algorithm for the discrete logarithm—which is, so far, the only successful quantum attack on a classical one-way function—quantum attacks based on the hidden subgroup problem are unlikely to work. We also show that reconstructing any entry of M, or the trace of M, with nonnegligible advantage is essentially as hard as inverting f_V. Finally, f_V can be efficiently computed and the number of output bits is less than $1 + \epsilon$ times the number of input bits for any $\epsilon > 0$.

1 Introduction

When a quantum computer is finally built, perhaps its most important practical impact will be on modern cryptography, thanks to Shor's celebrated quantum algorithms for factoring and discrete logs [Sho97] (and a sequence of followup results). Quantum cryptography provides a partial recourse, though its scope is limited by "no-go" theorems such as the impossibility of quantum bit commitment, as well as extravagant physical infrastructure requirements. A plausible route to a more acceptable antidote was suggested in a result contempraneous with Shor's paper, showing that quantum computers require exponential time to invert a random permutation in a black box model [BBBV97]. Since a random permutation is a standard abstraction for a one-way function, this result suggested the possibility of creating classical cryptography that is resistant to quantum cryptanalysis. The practical challenge is to design a function $f : \Sigma^n \to \Sigma^m$ that can be computed very efficiently by a classical computer, while providing credible evidence that inversion is difficult even with a quantum computer. It is also desirable that f be nonexpansive, i.e., that m not be much larger than n. This is the goal of this paper.

Our task is facilitated by new insights obtained over the last few years into the limits of quantum algorithms for the non-abelian hidden subgroup problem (HSP). A series of negative results [HRTS00, GSVV01, MRS05] culminating in Hallgren, et al. [HMR$^+$06] shows that for sufficiently non-abelian groups the HSP is hard for quantum computers in the following sense: any quantum algorithm using the coset state framework requires exponential time unless it makes highly entangled measurements of $\Omega(\log |G|)$ registers. Very few algorithmic models for highly-entangled measurements are known; one of the few proposals for carrying out such measurements efficiently is a "quantum sieve," developed by Kuperberg [K05] for the HSP on the dihedral group. However, a recent result of Moore, Russell, and Śniady [MRS06] shows that no such approach yields an efficient algorithm over the symmetric groups. In fact, for the cases relevant to Graph Isomorphism, algorithms of this form cannot even do much better than the best known classical algorithms. This forms the basis of our main assumption about the limitations of quantum algorithms.

Our function, which we denote f_V, is parametrized by a list of vectors $V = v_1, v_2, \ldots, v_m$; we will choose each v_i uniformly at random from $\mathbb{F}_q^n$, where q is some small prime. Then given $M \in \mathrm{GL}_n(\mathbb{F}_q)$, that is, an invertible $n \times n$ matrix over $\mathbb{F}_q$, we define $f_V(M)$ as the collection

$$MV = \{Mv \mid v \in V\} \ .$$

However, f_V returns this collection as an unordered set (say, sorted in lexicographic order). In other words, we know that each $w \in f_V(M)$ is Mv for some $v \in V$, but we do not know with what permutation the vs and ws correspond.

In Section 2, we show that f_V is one-to-one with high probability in V whenever m is slightly larger than n, say $m = n + O(\ln^2 n)$. Also, clearly f_V can be computed very efficiently, in time $M(n)$, the time to multiply two $n \times n$ matrices. As a function of the input length $k = n^2$, the time is essentially $\sqrt{M(k)}$.

In Section 3, we point out that the natural reduction of inverting f_V to a hidden subgroup (or hidden shift) problem results in hidden subgroup problems on the general linear group GL_n. This group contains the symmetric group S_n as a subgroup, and its HSP appears resistant to all known quantum techniques. Moreover, we reduce the Graph Isomorphism problem to the problem of inverting f_V. This implies that no quantum attack analogous to Shor's algorithm for the discrete logarithm can succeed, unless there is an efficient quantum algorithm for Graph Isomorphism.

We stress that, unlike Graph Isomorphism for which there is no known way to generate hard random instances, inverting f_V is uniformly hard because of the following simple observation: for any matrix A, we have $f_V(AM) = Af_V(M)$. By choosing A randomly, this allows us to map a fixed instance $f_V(M)$ to a random one with the same V. It follows that, for any fixed V, if f_V can be inverted on even a $1/poly(n)$ fraction of matrices M, then there is a probabilistic algorithm that inverts it on arbitrary inputs M. A similar though more complicated assertion can be made about uniform hardness with respect to choice of V (see refsec:hard-core).

Moreover, we show in Section 4 that reconstructing partial information about $f_V^{-1}(x)$ is almost as hard as inverting f_V. Specifically, assuming that f_V is a one-way function, we show that *any entry of M is hard to recover in any basis*, though this requires a quasipolynomial hardness assumption on f_V. We observe, also, that **tr** M, the trace of M, is hard to recover even under typical super-polynomial hardness assumptions.

It remains an open question whether we can embed a trapdoor in f_V or a suitable modification. We should point out that there are some classical cryptosystems that are not known to be breakable by a quantum computer—lattice-based cryptosystems such as the Ajtai-Dwork [AD97] cryptosystem and their subsequent improvements due to Regev [Reg04a], and the McEliece cryptosystem [McE78]. Indeed, Regev's improvement in the efficiency of lattice-based cryptosystems is based on a quantum reduction—thus the increased efficiency is predicated on resistance of the cryptosystem to quantum attacks! Evidence of quantum intractibility for this cryptosystem comes from the relationship between finding short vectors and the dihedral hidden subgroup problem [Reg04b]. In particular, even though single register Fourier sampling is information-theoretically sufficient to reconstruct the hidden subgroup, the classical reconstruction problem is as hard as Subset Sum. On the other hand, quantum reconstruction is not ruled out, and Kuperberg's quantum sieve [K05] provides what may be thought of as a mildly subexponential quantum reconstruction algorithm.

The evidence for quantum intractibility for the one-way function proposed here is stronger: single register Fourier sampling is provably insufficient, highly-entangled measurements on polynomially many registers is necessary, and no Kuperberg-like approach can yield an efficient algorithm. The design of efficient cryptographic primitives resistant to quantum attack is a pressing practical problem whose solution can have an enormous impact on the practice of cryptography long before a quantum computer is physically realized. A program to create such primitives must necessarily rely on insights into the limits of quantum algorithms, and this paper explores consequences of the strongest such insights we have about the limits of quantum algorithms.

Notation. As above, we let $\mathbb{F} = \mathbb{F}_q$ denote the finite field with q elements, q a fixed prime. We let $\mathrm{GL}_n(\mathbb{F}_q)$ (abbreviated GL_n when the context is clear) denote the collection of invertible $n \times n$ matrices over $\mathbb{F}_q$. Similarly $\mathrm{End}_n = \mathrm{End}_n(\mathbb{F}_q)$ denotes the set of all $n \times n$ matrices. If $M \in \mathrm{End}_n$ and $V \subset \mathbb{F}_q^n$, we let MV denote the collection $\{M\mathbf{v} \mid \mathbf{v} \in V\}$.

2 The function is one-to-one

Our first theorem shows that when m is slightly larger than n, then f_V is a one-to-one function with high probability. We have made only desultory attempts to optimize the rate at which $\delta = m - n$ must grow for the theorem to hold.

Theorem 1. *There is a constant A such that if $m = n + \delta$ where $\delta \geq A \ln^2 n$, then f_V is one-to-one with high probability in V.*

Proof. If there are two matrices M, M' such that $MV = M'V$, then $KV = V$ where $K = M^{-1}M'$. In other words, there is a permutation $\pi \in S_m$ such that $Kv_i = v_{\pi(i)}$ for all i. We will show that with high probability $K = \mathbb{1}$ is the only matrix with this property, and therefore that $M = M'$.

Let us call a particular permutation $\pi \in S_m$ *consistent* if there is a K such that $Kv_i = v_{\pi(i)}$ for all i, and let Cons_π be this event. We will show that

$$\Pr\left[\bigvee_{\pi \neq 1} \mathrm{Cons}_\pi\right] = o(1) \ .$$

i.e., with high probability the only consistent permutation is the identity $\pi = 1$.

Given a fixed π, we determine an order on V as follows. First, we sort the cycles of π in order of increasing length, starting with the fixed points. We break ties by assigning each cycle an index equal to the smallest i such that v_i appears in it and putting cycles with the smallest index first. Then, we rotate each cycle so that the v_i with smallest i in that cycle comes first. The details here are irrelevant; all that matters is that each π determines an order on V with the properties that the vectors corresponding to fixed points come first, and that groups of vectors corresponding to cycles of π are contiguous.

Now fix a constant C, and let L_π consist of the first $n + \delta - C \ln n$ vectors in V according to this order. Let Spans_π be the event that L_π spans the entire space $\mathbb{F}_q^n$. Then the union bound gives

$$\Pr\left[\bigvee_{\pi \neq 1} \text{Cons}_\pi\right] \leq \sum_{\pi \neq 1} \Pr\left[\text{Cons}_\pi \,|\, \text{Spans}_\pi\right] + \Pr\left[\bigvee_{\pi} \overline{\text{Spans}_\pi}\right]$$

To bound the conditional probability $\Pr[\text{Cons}_\pi \,|\, \text{Spans}_\pi]$, note that if L_π spans the entire space, then K is determined by the images of the vectors in L_π. Therefore, if all the vectors in L_π are fixed by K, then $K = \mathbb{1}$ and $\pi = 1$. On the other hand, we have sorted V so that the fixed vectors come first, so if $\pi \neq 1$ none of the the $C \ln n$ vectors outside L_π can be fixed. We expose these vectors in sorted order. For each $v_i \notin L_\pi$ which is not the first in its cycle, the probability that v_i is the image under K of its predecessor $v_{\pi^{-1}(i)}$ is q^{-n} since v_i is uniformly random. These events are independent and each of these cycles is of length at least 2, so the probability that $Kv_i = v_{\pi(i)}$ for all $v_i \notin L$ is at most $q^{-(C/2)n \ln n}$. Summing over all $(n + \delta)!$ permutations π and assuming for simplicity that $\delta \leq n$ (a condition which we can easily remove), the conditional probability that any $\pi \neq 1$ is consistent is at most

$$(2n)!\, q^{-(C/2)n \ln n} = n^{O(1)} (2/\mathrm{e})^{2n} n^{(2-(C/2)\ln q)n}$$

which is $o(1)$ if

$$C \geq 4/\ln q \ . \tag{1}$$

Now we bound the probability that Spans_π fails to hold for any π by proving that with high probability V contains no subsets L of size $n + \delta - C \ln n$ which do not span the entire space. By Markov's inequality, the probability that a given such L does not span the space is at most the expected number of nonzero vectors u which are perpendicular to all $v \in L$. Since the $v \in V$ are uniformly random, for any fixed u the inner product $u \cdot v$ is zero with probability $1/q$. Thus this expectation is

$$(q^n - 1)/q^{n+\delta-C\ln n} < q^{-\delta + C \ln n} = n^{O(1)} n^{-(A \ln q) \ln n}$$

where we used $\delta = A \ln^2 n$. The number of subsets of size $n + \delta - C \ln n$ is

$$\binom{n+\delta}{C \ln n} < (2n)^{C \ln n} = n^{O(1)} n^{C \ln n}$$

where we again assume for simplicity that $\delta \leq n$. So, by the union bound, the probability that a non-spanning subset of size $n + \delta - C \ln n$ is at most $n^{O(1)} n^{(C - A \ln q) \ln n}$ which is $o(1)$ if

$$A > C/\ln q \ . \tag{2}$$

In order to satisfy (1) and (2), we set, say, $C = 4/\ln q$ and $A = 5/\ln^2 q$. Then with high probability, the identity permutation 1 is the only consistent one. Finally, note that V spans the entire space with overwhelming probability; and in this case, if $Kv = v$ for all v in V, then K must be the identity. $\square$

3 Evidence for immunity against hidden subgroup attacks

In this section we relate the hardness of our function to several fundamental problems in the area of quantum computation. Our principal hardness result, suggesting that f_V can resist the quantum attacks which Shor applied so dramatically to factoring and discrete log, shows that Graph Isomorphism can be reduced to the problem of inverting f_V. Our current belief, based on a series of negative results, is that Graph Isomorphism, and more generally the HSP on groups like S_n and GL_n which have exponentially high-dimensional representations, is hard for quantum computers. If this belief is correct, then f_V cannot be efficiently inverted by such methods. We observe, also, that inverting f_V can be reduced to natural hidden shift and hidden subgroup problems on the group GL_n.

We begin by reducing the problem of inverting f_V to the Hidden Shift Problem on the group GL_n. Given a group G, an instance of a Hidden Shift problem consists of two functions $f_1, f_2 : G \to S$, with the promise that $f_2(g) = f_1(gs)$ for some shift $s \in G$. Now, given V and $f_V(M) = MV$, we can define two functions $f_1, f_2 : \mathrm{GL}_n \to S$ where S is the set of unordered lists of vectors in $\mathbb{F}_q^n$. Namely, we define

$$f_1(N) = NV \text{ and } f_2(N) = Nf_V(M) = NMV \ .$$

Then $f_1(N) = f_V(N)$ and $f_2(N) = f_V(NM) = f_1(NM)$, and M is the hidden shift.

Now, given a Hidden Shift Problem on a group G where the functions f_1, f_2 are one-to-one, we can reduce it to a Hidden Subgroup Problem on a larger group, namely the wreath product $G \wr \mathbb{Z}_2$. This group is the semidirect product $(G \times G) \rtimes \mathbb{Z}_2$, where we extend $G \times G$ with an involution which exchanges the two copies of G. We denote its elements (g_1, g_2, z), where those with $z = 0$ form the normal subgroup which fixes the two copies of G, and those with $z = 1$ form its nontrivial coset which exchanges them.

Recall that an instance of the Hidden Subgroup Problem consists of a function $f : G \to S$ with the promise that, for some subgroup H, $f(x) = f(y)$ if and only if $x = yh$ for some $h \in H$. Given a Hidden Shift Problem with functions $f_1, f_2 : G \to S$, define the following function $f : G \wr \mathbb{Z}_2 \to S^2$:

$$f(g_1, g_2, 0) = (f_1(g_1), f_2(g_2))$$
$$f(g_1, g_2, 1) = (f_2(g_2), f_1(g_1))$$

Now suppose that $f_2(g) = f_1(gs)$ and let α be the involution $(s^{-1}, s, 1)$. If multiplication in $G \wr \mathbb{Z}_2$ is defined so that $(g_1, g_2, 0) \cdot \alpha = (g_2 s, g_1 s^{-1}, 1)$, then f's hidden subgroup is the order-2 subgroup $H = \{1, \alpha\}$. (Indeed, the canonical reduction of Graph Isomorphism to the Hidden Subgroup Problem over $S_n \wr \mathbb{Z}_2$ is exactly of this type, where $\alpha = (\pi^{-1}, \pi, 1)$ exchanges the two graphs and π is the isomorphism between them.) Finally, we point out that GL_{2n} contains a copy of $\mathrm{GL}_n \wr \mathbb{Z}_2$: namely, the subgroup consisting of matrices of the form

$$\begin{pmatrix} g_1 & 0 \\ 0 & g_2 \end{pmatrix} \text{ or } \begin{pmatrix} 0 & g_1 \\ g_2 & 0 \end{pmatrix}$$

where $g_1, g_2 \in \mathrm{GL}_n$. Thus the problem of inverting f_V reduces to the Hidden Shift and Hidden Subgroup Problems in GL_n and GL_{2n} respectively.

Now, we give a reduction from Graph Isomorphism to the problem of inverting f_V. Specifically, we reduce the decision problem of telling whether two graphs G_1, G_2 are isomorphic to the decision problem of telling, given V and W, whether there is a matrix M such that $MV = W$, and hence whether W is in the image of f_V. The same construction reduces the promise problem of finding the isomorphism between two isomorphic graphs to the problem of finding $M = f_V^{-1}(W)$.

The reduction is quite simple. Given a graph G_1 with n vertices and m edges, V will consist of $n + m$ vectors in $\mathbb{F}_q^n$. We identify each vertex u with a basis vector, which we include in V, and for each edge (u, v) we include the vector $u + v$. We construct W from G_2 similarly.

Clearly $G_1 \cong G_2$ if and only if $MV = W$ for some permutation matrix M. First we show that, if $q \geq 3$, any M such that $MV = W$ is necessarily a permutation matrix. To see this, note that since each vertex of G_1 gets mapped to a vertex or an edge of G_2, each column of M is zero except for one or two

1s. But in $\mathbb{F}_q^n$ with $q \geq 3$, the sum of two such vectors has at least two nonzero components, so no edge of G_1 can be mapped to a vertex of G_2. It follows that every vertex of G_1 is mapped to a vertex of G_2, so M is a permutation matrix.

In the case $q = 2$, it is possible that M is not a permutation matrix, and that some vertices get mapped to edges and vice versa. However, M's existence still implies that G_1 and G_2 are isomorphic, and allows us to easily determine the isomorphism π between them. Let us call a vertex of G_1 "green" or "red" if it is mapped to a vertex or an edge respectively, and consider a vertex w of G_2. Since $M^{-1}w$ is either a vertex or an edge, either there is a green vertex u such that $Mu = w$, or there is a red vertex u with a unique green neighbor v such that $Mu = w + Mv$ and so $M(u + v) = w$. In either case, define $\pi(u) = w$; since π is one-to-one, it follows that *every* red vertex has a unique green neighbor.

It remains to check that π is an isomorphism. Denote the set of edges of G_1 and G_2 as E_1 and E_2 respectively, and suppose that $(u, v) \in E_1$. If u and v are green, then $M(u + v) = \pi(u) + \pi(v)$. If u is red and v is its unique green neighbor, then $Mu = \pi(u) + \pi(v)$. Finally, if u and v are both red, they must have the same green neighbor t since otherwise $M(u + v)$ would be the sum of four basis vectors; then $M(u + v) = \pi(u) + \pi(v) + 2\pi(t) = \pi(u) + \pi(v)$. In each case, since $\pi(u) + \pi(v) \in W$ we have $(\pi(u), \pi(v)) \in E_2$, and this completes the proof.

4 Uniformity of hardness, amplification, and hard-core predicates

Self-reducibility and uniform hardness. As we pointed out in the Introduction, our function has a simple symmetry which causes it to be self-reducible from the worst case to the random case: for any fixed V, we have $f_V(AM) = Af_V(M)$. It follows by standard amplification that, for any fixed V, if f_V can be inverted on even a $1/\text{poly}(n)$ fraction of matrices M then it can be inverted with probability $1 - e^{-\text{poly}(n)}$ on any particular M.

We can define uniform hardness with respect to V using another obvious symmetry,

$$f_{BV}(M) = f_V(MB) \ .$$

Let us say that $V \sim V'$ if there is a $B \in \text{GL}_n$ such that $V' = BV$. This is clearly an equivalence relation; we will call the equivalence class containing V its *orbit*, and denote it $[V]$. Then a similar argument shows that inverting f_V is uniformly hard within each orbit: namely, if f_V can be inverted on even a $1/\text{poly}(n)$ fraction of matrices M and vectors $V' \in [V]$ then it can be inverted with probability $1 - e^{-\text{poly}(n)}$ on any particular M and $V' \in [V]$.

A priori, even if it is hard to invert f_V, one might hope to recover partial information about M from its image $f_V(M)$, such as its trace or a single entry in some basis. In this section, we show that this is essentially as hard as recovering all of M. Therefore, under reasonable hardness assumptions regarding f_V, these goals are also impossible for quantum computers to carry out efficiently.

Hard-core predicates. A hard-core predicate is an efficient description of a bit of information that is concealed by a given one-way function. Specifically, if $\{f_n : D_n \to R_n\}$ is a family of one-way functions, then an $s(n)$-*hard-core predicate* is a polynomial time computable family of functions $\{b_n : D_n \to \{0, 1\}\}$ so that for any algorithm A running in time $s(n)$, for sufficiently large n,

$$\left| \Pr_{f_n, w} [A(f_n(w)) = b_n(w)] - \frac{1}{2} \right| \leq \frac{1}{s(n)} \ .$$

Our goal here is to show that every individual entry of M is a hard-core bit in any basis; in particular, *recovering any entry of M is as hard as inverting f_V*. We also point out that recovering the trace of M is as hard as inverting f_V.

We begin by formalizing the notions of hardness we require for the function f_V.

Assumption 1 ($t(n)$-hardness)**.** *For each $n \geq 1$, let $m = m(n) = (1+\epsilon)n$ for some constant $\epsilon > 0$, let M be a uniformly random element of $\mathrm{GL}_n(\mathbb{F})$, and let V be a collection of m independently and uniformly selected elements of $\mathbb{F}^n$. Then for all quantum algorithms A running in time $t(n)$,*

$$\Pr_{V,M}[A(M(V), V) = M] = \frac{1}{t(n)} \ .$$

We devote the remainder of this section to showing the following two theorems.

Theorem 2. *If f_V is quasipolynomially hard (that is, $t(n)$-hard for every $t(n) = 2^{\log^{O(1)} n}$) then every entry of M (in any basis) is a quasipolynomially hard-core predicate.*

Theorem 3. *If f_V is polynomially hard (that is, $t(n)$-hard for every $t(n) = n^{O(1)}$) then the trace $\boldsymbol{tr} : \mathrm{GL}_n(\mathbb{F}) \to \mathbb{F}$ is a polynomially hard-core predicate.*

4.1 The bilinear predicate: every matrix entry is hard

Given two basis vectors $\mathbf{a}$ and $\mathbf{b}$, the corresponding matrix element can be written as an inner product $\langle \mathbf{a}, M\mathbf{b} \rangle$. We will show that if f_V is quasipolynomially hard, then this function is a hard-core predicate for f_V for any fixed nonzero $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$. Specifically, given an algorithm P running in time $2^{\log^{O(1)} n}$ for which

$$\Pr_{V,M}\big[P(f_V(M), V) = \langle a, Mb \rangle\big] \geq 1/2 + \epsilon \ \text{ with } \ \epsilon = 2^{-\log^{O(1)} n} \ ,$$

we show how to invert f_V on a $2^{-\log^{O(1)} n}$ fraction of its inputs M, which would contradicting the assumption that f_V is quasipolynomially hard.

To simplify the exposition, we will fix q to be 2 in this section, and write $\mathbb{F} = \mathbb{F}_2$. We rely on the Goldreich-Levin theorem [GL89]; for larger prime q, we rely on its generalization to arbitrary finite fields by Goldreich, Rubinfeld, and Sudan [GRS95].

Initially, we wish to focus attention on certain "good" choices of V, where the algorithm P is a good predictor for $\langle \mathbf{a}, M\mathbf{b} \rangle$. Recall that $[V]$ denotes the orbit of V under multiplication by elements of GL_n. Define an element V to be "good" if

$$\Pr_{V' \in [V], M}\big[P(f_{V'}(M), V') = \langle a, Mb \rangle\big] \geq \frac{1}{2} + \frac{\epsilon}{2} \ . \tag{3}$$

It is easy to show that at least an $\epsilon/2$ fraction of V must be good in this sense; we fix a specific such V for the remainder of the proof, and show how to invert the function f_V in this case.

We first show how to use P to implement an algorithm for any fixed M, which takes as input $x, y \in \mathbb{F}^n$ (and $(f_V(M), V)$) and outputs $\langle x, My \rangle$ correctly on $1/2 + \epsilon/2$ fraction of x, y. First note that for two matrices $A, B \in \mathrm{GL}_n$, the pair $(f_{BV}(AMB^{-1}), BV) = (AMV, BV)$ can be computed efficiently from $(f_V(M), V) = (MV, V)$ by left-multiplying MV and V by A and B respectively. Defining $T(A, B) = P(f_{BV}(AMB^{-1}), BV)$, we may then rewrite (3) in terms of $T(\cdot, \cdot)$:

$$\Pr_{A,B \in \mathrm{GL}_n(\mathbb{F})}[T(A, B) = \langle a, AMB^{-1}b \rangle] \geq \frac{1}{2} + \frac{\epsilon}{2} \ . \tag{4}$$

Finally, for a pair of vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$, define $t(\mathbf{x}, \mathbf{y}) = T(A, B)$, where A and B are random elements of $\mathrm{GL}_n(\mathbb{F})$ for which $A^t\mathbf{a} = \mathbf{x}$ and $B^{-1}\mathbf{b} = \mathbf{y}$, so that $\langle \mathbf{a}, AMB^{-1}\mathbf{b} \rangle = \langle \mathbf{x}, M\mathbf{y} \rangle$. Rewriting (4), we conclude:

$$\Pr_{\mathbf{x},\mathbf{y} \in \mathbb{F}^n}[t(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, M\mathbf{y} \rangle] \geq \frac{1}{2} + \frac{\epsilon}{2} \ . \tag{5}$$

Let us call a vector $\mathbf{x} \in \mathbb{F}^n$ *ℓ-good* if $\Pr_{\mathbf{y} \in \mathbb{F}^n}[t(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, M\mathbf{y} \rangle] \geq 1/2 + \epsilon/4$. If follows that a uniformly selected $\mathbf{x}$ is ℓ-good with probability at least $\epsilon/4$. Note, furthermore, that if $\mathbf{x}$ is a fixed ℓ-good element of $\mathbb{F}^n$, then the Goldreich-Levin construction [GL89] can be used to determine $\langle \mathbf{x}, M\mathbf{y} \rangle$ for

all $\mathbf{y} \in \mathbb{F}^n$ (in time polynomial in n and ϵ^{-1}). In particular, this determines an entire row of M when expressed in a basis containing $\mathbf{x}$.

We consider now a family G, consisting of $2 \log m$ vectors selected independently and uniformly in $\mathbb{F}^n$. We say that G is ℓ-good if this is true of each of its elements, a favorable event that occurs with probability at least $(\epsilon/4)^{\log 2m}$. Furthermore, the probability that G contains a linearly dependent set of vectors is no more than $2 \log(m) \cdot 2^{-n+2 \log m} = 2^{-\Omega(n)}$. (This can be seen by selecting the elements of G in order and bounding the unlikely event that an element falls into the span of the previously chosen vectors.) Thus

$$\Pr[G \text{ is } \ell\text{-good} \wedge G \text{ is independent}] \geq (\epsilon/4)^{2 \log m} + e^{-\Omega(n)} \ .$$

Now, for each $\mathbf{g} \in G$, application of the Goldreich-Levin construction to each component of $\mathbf{g}$ (reconstructing $\langle \mathbf{g}, M\mathbf{y} \rangle$ for all $\mathbf{y}$) determines $\langle \mathbf{g}, M\mathbf{v} \rangle$ for each $\mathbf{v} \in V$ and $\mathbf{g} \in G$. Therefore, in this case we can reconstruct $2 \log m$ "generalized rows" of M.

Observe that if the elements of V (and hence $W = M(V)$) are considered to be selected independently and uniformly at random, *and* independently of G, then the probability that two elements w and w' of W have the property that $\langle \mathbf{g}, \mathbf{w} \rangle = \langle \mathbf{g}, \mathbf{w}' \rangle$ for all $\mathbf{g} \in G$ is $2^{-2 \log m}$. Let $\Pi_G : \mathbb{F}^n \to \mathbb{F}^{2 \log m}$ denote the projection onto the space spanned by the vectors in G. In particular, this information would appear to determine the bijection $b_M : V \to W$ effected by the action of M on V. This intuitive argument is misleading, as written, since the notion of ℓ-good depends on V (and so on W) via the arbitrary predicting algorithm P. Instead, our goal below will be to show that the total number of permutations of the set W under which Π_G is invariant is small enough that we can exhaustively search them to uncover the bijection b_M and hence the linear operator M.

Consider random (and independent) selection of G, V, and M (so that $W = M(V)$ is also determined) with no extra conditioning except that G be linearly independent. Let I_G denote the collection of permutations $\phi : M \to M$ with the property that $\Pi_G w = \Pi_G \phi(w)$, for all $w \in W$. We will show below that $\mathrm{E}_{V,M,G}[|I_G|] = O(\sqrt{m})$. Then Markov's inequality will allow us to bound the probability that $|I_G|$ exceeds $\epsilon^{O(\log n)}$. To round out the proof we will show that the chance that V is good and that G is l-good is much higher than this failure probability, thereby concluding that there is a significant chance that V is good, G is l-good and that $|I_G| = \epsilon^{O(\log n)}$.

As the elements of w are selected independently (and uniformly) in $\mathbb{F}^n$, each $\Pi_G w$ is an independent, uniform element of $\mathbb{F}^{|G|}$. Fixing a permutation ϕ, let $\lambda_1, \lambda_2, \ldots$ be the lengths of its cycles, arranged in nonincreasing order. The probability that the elements of M in each of these cycles are mapped to the same element under Π_G is no more than $\prod_i (2^{-|G|})^{\lambda_i - 1} = \prod_i (m^{-2})^{\tau}(\phi)$, where $\tau(\phi) = \sum_i (\lambda_i - 1)$ is also the minimum number of transpositions required to write ϕ.

This quantity is bounded by the lemma below. Its proof uses the machinery of exponential generating functions, and is relegated to Appendix A.

Lemma 4. *Let $0 < z < 1/k$; then*

$$q_k(z) = \sum_{\pi \in S_k} z^{t(\pi)} = O(\sqrt{k}) \frac{e^{-k}}{(1 - zk)^{1/z}} \ . \tag{6}$$

In light of this bound, the expectation of $|I_G|$, the number of ϕ under which π_G is invariant, is no more than

$$\sum_{\phi \in S_m} \left(\frac{1}{m^2} \right)^{\tau(\phi)} = O(\sqrt{m}) \frac{e^{-m}}{(1 - 1/m)^{m^2}} \ .$$

As $-\ln(1 - x) = x + x^2/2 + x^3/3 + \ldots$, we have

$$e^{-m} \cdot (1 - 1/m)^{-m^2} = \exp(-m + m^2[1/m + (1/m)^2/2 + O(1/m^3)]) = O(1) \ .$$

Thus $\mathrm{E}[|I_G|] = O(\sqrt{m})$.

Putting the pieces together, with M, V, and G selected as above,

$$\Pr_{V,M,G}[(V \text{ is good}) \wedge (G \text{ is both } \ell\text{-good and linearly independent})] \geq \frac{\epsilon}{2} \cdot \left(\frac{\epsilon}{4}\right)^{2\log m} \geq \left(\frac{\epsilon}{4}\right)^{1+2\log m} \quad .$$

As $\mathrm{E}_{V,M,G}[|I_G|] = O(\sqrt{m})$, by Markov's inequality there is a constant c so that

$$\Pr_{V,M,G}\left[|I_G| \geq c\sqrt{m}(4/\epsilon)^{2\log m}\right] \leq \frac{1}{2} \cdot \left[\left(\frac{\epsilon}{4}\right)^{1+2\log m}\right] \quad .$$

Thus, with probability at least $(1/2)(\epsilon/4)^{(1+2\log m)}$, V is good, G is ℓ-good, and there are $(4/\epsilon)^{O(\log n)}$ permutations of W that fix Π_G. These permutations determine a set of no more than $(4/\epsilon)^{O(\log n)}$ mappings between V and W consistent with M; these can be exhaustively searched in time $\mathrm{poly}(n) \cdot (\epsilon/4)^{O(\log n)}$, which is quasipolynomial when ϵ^{-1} is.

We conclude this section with a proof that, even if ϵ^{-1} is only polynomial in n, hardness with respect to quasipolynomial time is the most we can hope for in the case of the bilinear predicate (in absence of further information about the preimage). First, choose a subspace S of $\mathbb{F}^n$ with dimension $\dim S = \log_2 n$. Now consider an oracle $P(a, b)$ defined as follows. If either a or b is orthogonal to S, then $P(a, b) = \langle a, Mb\rangle$, but if neither of them is orthogonal to S, then $P(a, b)$ is uniform in $\mathbb{F}$. Since a uniform vector in $\mathbb{F}^n$ is orthogonal to S with probability $1/n$, it follows that $P(a, b)$ is correct with probability $1/2 + \epsilon$ where $\epsilon > 1/n$.

Now choose a basis for $\mathbb{F}^n$, and let S be the subspace generated by the first $\dim S$ basis vectors. It is clear that this oracle gives us no information whatever regarding the matrix elements in the $\dim S \times \dim S$ minor at the upper left-hand corner of M. Therefore, we are forced to try all possible values for the elements of this minor by exhaustive search, and this takes $2^{(\dim S)^2} = 2^{\log^2 n}$ time.

4.2 The trace predicate

The proof that the trace predicate is hard is a direct consequence of the Goldreich-Levin theorem [GL89] and its generalization to arbitrary finite fields by Goldreich, Rubinfeld, and Sudan [GRS95]. Specifically, consider the trace $\mathbf{tr} : \mathrm{GL}_n(\mathbb{F}) \to \mathbb{F}$. Suppose now that there is a polynomial-time quantum algorithm P so that for M selected uniformly at random in GL_n and V a collection of m independent and uniform vectors of $\mathbb{F}^n$,

$$\Pr_{V,M}\left[P(f_V(M), V) = \mathbf{tr}(M)\right] \geq \frac{1}{2} + \epsilon \ ,$$

where $\epsilon = n^{-O(1)}$. It follows that for at least an $\epsilon/2$ fraction of the V, when selected as above, we have

$$\Pr_{M}\left[P(f_V(M), V) = \mathbf{tr}(M)\right] \geq \frac{1}{2} + \frac{\epsilon}{2} \ .$$

We show how to invert f_V for such "good" V; as these occur with probability $\epsilon/2$, this would contradict the assumption that f_V is polynomially hard. For the remainder of the proof we fix a specific V satisfying the the equation above.

Again note that for any matrix $N \in \mathrm{GL}_n$, the collection $f_V(NM) = NMV$ can be computed in polynomial time from $f_V(M)$, simply by left-multiplying the collection $f_V(M) = MV$ by N. In particular, given $f_V(M)$, the function $T : \mathrm{GL}_n(\mathbb{F}) \to \mathbb{F}$ given by $T(N) = P(f_V(NM), V)$ can be computed in polynomial time and has the property that

$$\Pr_{N}\left[T(N) = \mathbf{tr}(NM)\right] \geq \frac{1}{2} + \epsilon \ . \tag{7}$$

Now, for a fixed matrix C, the function $\ell_C : M \mapsto \mathbf{tr}(CM)$ is a linear function and, moreover, all linear combinations of the entries of M can be written in this way. In light of this, note that if the guarantee (7)

could be arranged with the matrix C being selected uniformly at random from the collection of *all* matrices (rather than the invertible ones), we could immediately apply the Goldreich-Levin [GL89] construction at this point to recover M. This "oracle" T can, however, be extended to an oracle $\tilde{T}$ defined on the family of all matrices C by simply assigning random values to the singular matrices $C \notin \mathrm{GL}_n$, in which case with constant probability (over the selection of random values for this oracle),

$$\Pr_N\big[T(N) = \mathbf{tr}(NM)\big] \geq \frac{1}{2} + \alpha_p(n)\epsilon \ , \tag{8}$$

where

$$\alpha_p(n) = \prod_{i=0}^{n-1}\left(1 - \frac{1}{p^{n-i}}\right) \geq \prod_{i=0}^{\infty}\left(1 - \frac{1}{2^i}\right) \approx .2711$$

is the probability that a random $n \times n$ matrix over $\mathbb{F}_p$ is invertible. In this case, when $p = 2$ the Goldreich-Levin theorem can be applied directly:

Theorem 5 ([GL89])**.** *Let $g : \mathbb{F}_2^n \to \mathbb{F}_2$ be a function so that for some $h \in \mathbb{F}_2^n$, $\Pr_{x\in\mathbb{F}_2^n}[g(x) = \langle x, h\rangle] \geq \frac{1}{2} + \epsilon$ and let $c \geq 0$. Then there is a randomized algorithm running in time poly(n, ϵ^{-1}) (and making no more than poly(n, ϵ^{-1}) black-box queries to g) that determines h with probability $1 - 1/n^c$.*

When $q > 2$, one has to apply the generalization of [GL89] to arbitrary finite fields by Goldreich, Rubinfeld, and Sudan [GRS95].

References

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In ACM, editor, *Proceedings of the 29th annual ACM Symposium on the Theory of Computing*, pages 284–293, New York, NY, USA, 1997. ACM Press.

[BBBV97] Charles Bennett, Ethan Bernstein, Gilles Brassard and Umesh Vazirani. Strengths and Weaknesses of Quantum Computation. *SIAM Journal on Computing*, 26(5):1510–1523, October 1997.

[GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In ACM, editor, *Proceedings of the twenty-first annual ACM Symposium on Theory of Computing, Seattle, Washington, May 15–17, 1989*, pages 25–32, New York, NY, USA, 1989. ACM Press. ACM order no. 508890.

[GRS95] O. Goldreich, R. Rubinfeld, M. Sudan. Learning polynomials with queries: the highly noisy case. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 294–303, Milwaukee, WI, October, 1995.

[GSVV01] Michelangelo Grigni, Leonard Schulman, Monica Vazirani, and Umesh Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. In ACM, editor, *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 68–74, New York, NY, USA, 2001. ACM Press.

[HMR+06] Sean Hallgren, Cristopher Moore, Martin Rötteler, Alexander Russell, and Pranab Sen. Limitations of quantum coset states for graph isomorphism. In ACM, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 604–617, New York, NY, USA, 2006. ACM Press.

[HRTS00] Sean Hallgren, Alexander Russell, and Amnon Ta-Shma. Normal subgroup reconstruction and quantum computation using group representations. In ACM, editor, *Proceedings of the 32nd annual ACM Symposium on Theory of Computing*, pages 627–635, New York, NY, USA, 2000. ACM Press.

[KNV02] E. Kashefi, H. Nishimura and V. Vedral. On quantum one-way permutations. In Quantum Information and Computation, 5, 379, 2002.

[K05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005.

[McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. Technical Report 42-44, Jet Propulsion Lab, Pasadena, CA, 1978.

[MR06] Cristopher Moore and Alexander Russell. On the impossibility of a quantum sieve algorithm for graph isomorphism. Technical Report quant-ph/0609138, arXiv.org e-Print archive, 2006.

[MRS06] Cristopher Moore, Alexander Russell, and Piotr Śniady. On the impossibility of a quantum sieve algorithm for graph isomorphism: unconditional results. Technical Report quant-ph/0612089, arXiv.org e-Print archive, 2006.

[MRS05] Cristopher Moore, Alexander Russell, and Leonard Schulman. The symmetric group defies Fourier sampling. In *Proceedings of the 46th Symposium on Foundations of Computer Science*, pages 479–488, 2005.

[Reg04a] Oded Regev. New lattice-based cryptographic constructions. *Journal of the ACM*, 51(6):899–942, November 2004.

[Reg04b] Oded Regev. Quantum Computation and Lattice Problems. *SIAM Journal on Computing*, 33(3):738-760, 2004.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997.

[Sta97] Richard P. Stanley. *Enumerative Combinatorics*, volume I. Cambridge, 1997.

[Wil94] Hebert Wilf. *Generatingfunctionology*. Academic Press, 1994.

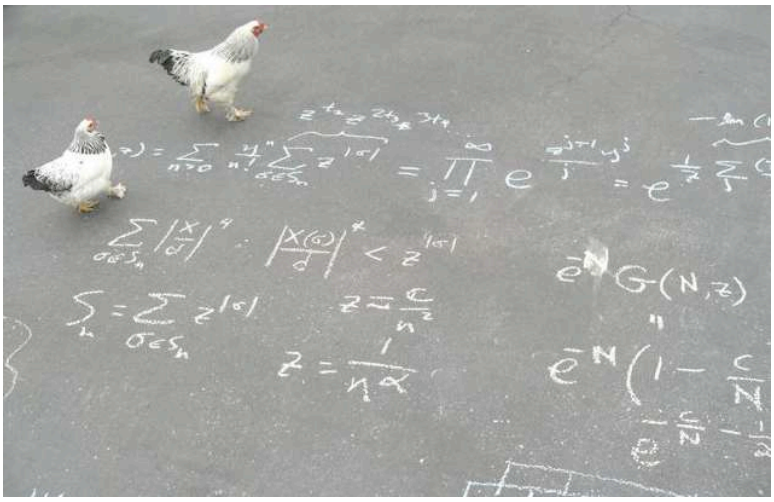

Figure 1: Two of the authors hard at work chalking up the proof of Lemma 4 on an asphalt driveway.

A Proof of Lemma 4

Recall that Lemma 4 asserts that if $0 < z < 1/k$; then

$$q_k(z) = \sum_{\pi \in S_k} z^{t(\pi)} = O(\sqrt{k})\, \frac{e^{-k}}{(1-zk)^{1/z}} \quad . \tag{9}$$

Proof of Lemma 4. Consider the exponential generating function

$$g(y,z) = \sum_{m=0}^{\infty} \frac{y^m}{m!}\, q_m(z) \quad .$$

Using the techniques of [Wil94, Chapter 3], we can write this as a product over all k of contributions from the $(k-1)!$ possible k-cycles, including fixed points. Since each such cycle contributes k to m and $k-1$ to $t(\pi)$, and since there are $(k-1)!$ k-cycles on a given set of k objects, it follows (cf. Figure A) that

$$g(y,z) = \prod_{k=1}^{\infty} \exp\left(\frac{y^k z^{k-1}}{k}\right) = \exp\left(\sum_{k=1}^{\infty} \frac{y^k z^{k-1}}{k}\right) = \exp\left(-\frac{1}{z}\ln(1-yz)\right) = \frac{1}{(1-yz)^{1/z}} \quad .$$

Now note that $e^{-k} g(k,z)$ is the expectation of $q_m(z)$, where m is Poisson-distributed with mean k. Since $q_m(z) > 0$, this expectation is at least $q_k(z)$ times the probability that $m = k$, which is $e^{-k}k^k/k! = (1-o(1))/\sqrt{2\pi k}$. Thus we have

$$q_k(z) \le (1+o(1))\sqrt{2\pi k} \cdot e^{-k} g(k,z)$$

which concludes the proof. □

This figure "proof-chickens.png" is available in "png" format from:

http://lanl.arXiv.org/ps/quant-ph/0701115