SIAM J. COMPUT.
Vol. 36, No. 3, pp. 763–778

QUANTUM ALGORITHMS FOR SOME HIDDEN SHIFT PROBLEMS*

WIM VAN DAM†, SEAN HALLGREN‡, AND LAWRENCE IP§

Abstract. Almost all of the most successful quantum algorithms discovered to date exploit the ability of the Fourier transform to recover subgroup structures of functions, especially periodicity. The fact that Fourier transforms can also be used to capture shift structure has received far less attention in the context of quantum computation. In this paper, we present three examples of "unknown shift" problems that can be solved efficiently on a quantum computer using the quantum Fourier transform. For one of these problems, *the shifted Legendre symbol problem*, we give evidence that the problem is hard to solve classically, by showing a reduction from breaking algebraically homomorphic cryptosystems. We also define the *hidden coset problem,* which generalizes the hidden shift problem and the hidden subgroup problem. This framework provides a unified way of viewing the ability of the Fourier transform to capture subgroup and shift structure.

Key words. quantum computing, efficient algorithms, Legendre symbol

AMS subject classifications. 81P68, 68W40, 11Y16

DOI. 10.1137/S009753970343141X

1. Introduction.

The first problem to demonstrate a superpolynomial separation between random and quantum polynomial time was the recursive Fourier sampling problem [6]. Exponential separations were subsequently discovered by Simon [35], who defined a problem with respect to an oracle, and by Shor [34], who found polynomial-time quantum algorithms for factoring and discrete logarithms. We now understand that the natural generalization of Simon's problem and the factoring and discrete log problems is the hidden subgroup problem (HSP), and that when the underlying group is abelian and finitely generated, we can solve the HSP efficiently on a quantum computer. While recent results have continued to study important generalizations of the HSP (for example, [19, 21, 24, 25, 27, 37]), only the recursive Fourier sampling problem remains outside the abelian HSP framework.

In this paper, we give quantum algorithms for several hidden shift problems where we are given two functions f, g such that there is a shift s for which $f(x) = g(x + s)$ for all x. The problem is then to find s. We show how to solve this problem for several classes of functions, but perhaps the most interesting example is the shifted Legendre symbol problem, where g is the *Legendre symbol* with respect to a prime

*Received by the editors July 15, 2003; accepted for publication (in revised form) February 28, 2006; published electronically October 24, 2006. A preliminary version appeared as [15].
http://www.siam.org/journals/sicomp/36-3/43141.html

†Departments of Computer Science and Physics, University of California, Santa Barbara, Santa Barbara, CA 93106-5110 (vandam@cs.ucsb.edu). This author's work was supported in part by the NSA and ARDA under ARO contract W911NF-04-R-0009. Part of this work was done while the author was at MIT, University of California, Berkeley, MSRI, and HP Labs.

‡NEC Laboratories America, Inc., Princeton, NJ 08540 (hallgren@nec-labs.com). This author's work was supported at Caltech in part by an NSF Mathematical Sciences Postdoctoral Fellowship and in part by the NSF through the Institute for Quantum Information at the California Institute of Technology. Most of this work was done while the author was at the Mathematical Sciences Research Institute and the University of California, Berkeley, with partial support from DARPA QUIST grant F30602-01-2-0524.

§Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043 (lip@google.com). This author's work was supported by NSF grant CCR-0049092, DARPA grant F30602-00-2-0601, and DARPA QUIST grant F30602-01-2-2054. This work was done while the author was at the University of California, Berkeley, and the Institute for Quantum Information at the California Institute of Technology.

size finite field, and the problem is the following: "Given the function $f(x) = \left(\frac{x+s}{p}\right)$ as an oracle, find s."

The oracle problem that our algorithms solve can be viewed as the problem of predicting a pseudorandom function f. Such tasks play an important role in cryptography and have been studied extensively under various assumptions about how one is allowed to query the function (nonadaptive versus adaptive, deterministic versus randomized, etc.) [7, 31]. In this paper we consider the case where the function is queried in a quantum mechanical superposition of different values x. We show that if $f(x)$ is an s-shifted multiplicative character $\chi(x+s)$—like the Legendre symbol—then a polynomial-time quantum algorithm making such queries can determine the hidden shift s, breaking the pseudorandomness of f.

We conjecture that classically the shifted Legendre symbol is a pseudorandom function; that is, it is impossible to efficiently predict the value of the function after a polynomial number of queries if one is allowed only a classical algorithm with oracle access to f. Damgård gave partial evidence for this conjecture, proposing the related task: "Given a part of the Legendre sequence $\left(\frac{s}{p}\right), \left(\frac{s+1}{p}\right), \ldots, \left(\frac{s+\ell}{p}\right)$, where ℓ is $O(\log p)$, predict the next value $\left(\frac{s+\ell+1}{p}\right)$" as a hard problem with applications in cryptography [17].

As further evidence of our conjecture, we show that breaking certain algebraically homomorphic cryptosystems can be reduced to the shifted Legendre symbol problem. The reduction, together with our quantum algorithm for the shifted Legendre symbol problem, yields a polynomial-time quantum algorithm for breaking such cryptosystems. The best known classical algorithm [9] for breaking these cryptosystems is subexponential and is based on a smoothness assumption. Thus the shifted Legendre symbol problem is a problem for which there is an exponential separation between a quantum algorithm and the fastest known classical algorithm. These cryptosystems can also be broken by Shor's algorithm for period finding, but the two attacks on the cryptosystems appear to use completely different ideas.

While current quantum algorithms solve problems based on an underlying group and the Fourier transform over that group, we initiate the study of problems where there is an underlying ring or field. The Fourier transform over the additive group of the ring is defined using the characters of the additive group, the additive characters of the ring. Similarly, the multiplicative group of units induces multiplicative characters of the ring. The interplay between additive and multiplicative characters is well understood [30, 36], and we show that this connection can be exploited in quantum algorithms. In particular, we put a multiplicative character into the phase of the registers and compute the Fourier transform over the additive group. The resulting phases are the inner products between the multiplicative character and each of the additive characters, a Gauss sum. We hope the new tools presented here will lead to other quantum algorithms.

We give algorithms for three types of hidden shift problems. In the first problem, g is a multiplicative character of a finite field. Given f, a shifted version of g, the shift is uniquely determined from f and g. An example of a multiplicative character of $\mathbb{Z}/p\mathbb{Z}$ is the Legendre symbol. Our algorithm uses the Fourier transform over the additive group of a finite field. In the second problem, g is a multiplicative character of the ring $\mathbb{Z}/n\mathbb{Z}$. This problem has the feature that the shift is not uniquely determined by f and g, and our algorithm identifies all possible shifts. An example of a multiplicative character of $\mathbb{Z}/n\mathbb{Z}$ is the *Jacobi symbol*. In the third problem we have the same setup as in the second problem with the additional twist that n is unknown.

We also define the *hidden coset problem*, which is a generalization of the hidden shift problem and the hidden subgroup problem. This definition provides a unified way of viewing the quantum Fourier transform's ability to capture subgroup and shift structure.

Some of our hidden shift problems can be reduced to the nonabelian HSP, although efficient algorithms for these HSP instances are not known. The shifted Legendre symbol problem over $\mathbb{Z}/p\mathbb{Z}$ can be reduced to an instance of the HSP over the dihedral group $D_p = \mathbb{Z}/p\mathbb{Z} \rtimes \mathbb{Z}/2\mathbb{Z}$ if we assume a conjecture about subsequences of the Legendre symbol. Let $f(x,0) = \left(\left(\frac{x}{p}\right), \left(\frac{x+1}{p}\right), \ldots, \left(\frac{x+\ell}{p}\right)\right)$ and $f(x,1) = \left(\left(\frac{x+s}{p}\right), \left(\frac{x+s+1}{p}\right), \ldots, \left(\frac{x+s+\ell}{p}\right)\right)$, where s is unknown and $\ell = \text{polylog}(p)$. Then the hidden subgroup is $H = \{(0,0),(s,1)\}$. The conjecture that is necessary to ensure that f will be distinct on distinct cosets of H is thus the statement that the subsequence $\left(\left(\frac{x+s}{p}\right), \left(\frac{x+s+1}{p}\right), \ldots, \left(\frac{x+s+\ell}{p}\right)\right)$ is unique for every x (cf. Conjecture 2.1 in [9]). For the general shifted multiplicative character problem, the analogous reduction to the HSP may fail because f may not be distinct on distinct cosets. However, we can efficiently generate random coset states, that is, superpositions of the form $|x,0\rangle + |x+s,1\rangle$, although it is unknown how to use these to efficiently find s [18]. The issue of nondistinctness on cosets in the HSP has been studied for some groups [8, 20, 23, 22].

The existence of a time-efficient quantum algorithm for the shifted Legendre symbol problem was posed as an open question in [13]. The Fourier transform over the additive group of a finite field was independently proposed for the solution of a different problem in [4]. The current paper subsumes [14, 15, 26]. Building on the ideas in this paper, a quantum algorithm for estimating Gauss sums is described in [16].

This paper is organized as follows. Section 2 contains some definitions and facts. In section 3, we give some intuition for the ideas behind the algorithms. In section 4, we present an algorithm for the shifted multiplicative problem over finite fields, of which the shifted Legendre symbol problem is a special case, and show how we can use this algorithm to break certain algebraically homomorphic cryptosystems. In section 5, we extend our algorithm to the shifted multiplicative problem over rings $\mathbb{Z}/n\mathbb{Z}$. This has the feature that, unlike in the case of the finite field, the possible shifts may not be unique. We then show that this algorithm can be extended to the situation where n is unknown. In section 6, we show that all these problems lie within the general framework of the hidden coset problem. We give an efficient algorithm for the hidden coset problem provided g satisfies certain conditions. We also show how our algorithm can be interpreted as solving a deconvolution problem using Fourier transforms.

2. Background.

2.1. Notation and conventions. We use the following notation: ω_n is the nth root of unity $\exp(2\pi\mathrm{i}/n)$, and $\hat{f}$ denotes the Fourier transform of the function f. An algorithm computing in $\mathbb{F}_q$, $\mathbb{Z}/n\mathbb{Z}$, or G runs in polynomial time if it runs in time polynomial in $\log q$, $\log n$, or $\log |G|$.

In a ring $\mathbb{Z}/n\mathbb{Z}$ or a field $\mathbb{F}_q$, additive characters $\psi : \mathbb{Z}/n\mathbb{Z} \to \mathbb{C}^*$ or $\psi : \mathbb{F}_q \to \mathbb{C}^*$ are characters of the additive group, that is, $\psi(x+y) = \psi(x)\psi(y)$, and multiplicative characters $\chi : (\mathbb{Z}/n\mathbb{Z})^* \to \mathbb{C}^*$ or $\chi : \mathbb{F}_q^* \to \mathbb{C}^*$ are characters of the multiplicative group of units, that is, $\chi(xy) = \chi(x)\chi(y)$ for all x and y. We extend the definition of a multiplicative character to the entire ring or field by assigning the value zero to elements outside the unit group. All nonzero $\chi(x)$ values have unit norm and thus $\chi(x^{-1}) = \overline{\chi(x)}$.

We ignore the normalization term in front of a superposition unless we need to explicitly calculate the probability of measuring a particular value.

2.2. Computing superpositions. We will need to be able to compute the superposition $\sum_x f(x)|x\rangle$, where the function $f : G \to \mathbb{C}$ describes the *amplitudes* of the state in an efficient way. The specific functions f that we deal with in this article have the property that for each x either $f(x) = 0$ or $f(x)$ is an mth root of unity. Hence there is a function $f' : G \to \mathbb{Z}$ such that if $f(x) \neq 0$, then $f(x) = \exp(2\pi \mathrm{i} f'(x)/m)$. This additional function helps us in the following lemma, which describes how to construct the superpositions in an efficient way.

LEMMA 2.1 (computing superpositions). *Let $f : G \to \mathbb{C}$ be a complex-valued function with a finite domain G that is characterized by a function $f' : G \to \mathbb{Z}/m\mathbb{Z} \cup \{\mathrm{i}\infty\}$ such that $f(x) = \omega_m^{f'(x)}$ for all x with $f'(x) \in \mathbb{Z}/m\mathbb{Z}$ and $f(x) = 0$ if $f'(x) = \mathrm{i}\infty$. Then there is an efficient algorithm for creating the superposition $\sum_x f(x)|x\rangle$ with success probability equal to the fraction of $x \in G$ with $f(x)$ nonzero and that uses two queries to the function f'.*

Proof. Start with the superposition over all $x \in G$: $\sum_x |x, 0\rangle$. Compute $f'(x)$ into the second register and measure to see whether $f'(x) \neq \mathrm{i}\infty$. This succeeds with probability equal to the fraction of x such that $f(x)$ is nonzero. If successful, we are left with a superposition over all x such that $f(x)$ is nonzero. Next, compute the phase shift $\omega_m^{f'(x)}$ by adding $\bmod m$ the value $f'(x)$ to the superposition $\sum_j \omega_m^{-j}|j\rangle$ (which by itself is the Fourier transform over $\mathbb{Z}/m\mathbb{Z}$ of the state $|-1\rangle$), such that

$$\begin{aligned}\sum_{x\in G, f(x)\neq 0} |x\rangle \otimes \frac{1}{\sqrt{m}} \sum_{j\in\mathbb{Z}/m\mathbb{Z}} \omega_m^{-j}|j\rangle &\mapsto \sum_{x\in G, f(x)\neq 0} |x\rangle \otimes \frac{1}{\sqrt{m}} \sum_{j\in\mathbb{Z}/m\mathbb{Z}} \omega_m^{-j}|j + f'(x)\rangle \\ &= \sum_{x\in G, f(x)\neq 0} \omega_m^{f'(x)}|x\rangle \otimes \frac{1}{\sqrt{m}} \sum_{j\in\mathbb{Z}/m\mathbb{Z}} \omega_m^{-j}|j\rangle \\ &= \sum_{x\in G} f(x)|x\rangle \otimes \frac{1}{\sqrt{m}} \sum_{j\in\mathbb{Z}/m\mathbb{Z}} \omega_m^{-j}|j\rangle.\end{aligned}$$

If necessary, the state $\sum_j \omega_m^{-j}|j\rangle = \mathcal{F}|-1\rangle$ can be approximated arbitrarily closely (see [33, section 5.1]). □

2.3. The Fourier transform and approximate Fourier sampling. Although it is not known how to efficiently compute the quantum Fourier transform over $\mathbb{Z}/n\mathbb{Z}$ exactly, it is known how to efficiently approximate such transformations [12, 23, 28, 29]. The current section deals with the problem of approximating the right probability distribution induced by the Fourier transform if the size of the group is not known. This result will be used in section 5.2.

Fourier sampling a quantum state is the process of computing the Fourier transform and measuring the resulting state. The best-known example is Shor's factoring algorithm which finds the period of a function f defined on $\mathbb{Z}$. In that case the function f is periodic with period r and is injective in $\{0, \ldots, r-1\}$. By evaluating the function in superposition up to some chosen value q and measuring the function value, the state $|\phi\rangle = \sqrt{\frac{r}{q}} \sum_{i=0}^{q/r-1} |k + ir\rangle$ is created, where k depends on which function value was measured. If q were a multiple of r, then Fourier sampling $|\phi\rangle$ would result in a random integer multiple of q/r.

When a multiple of r is not known, we must understand the distribution induced by Fourier sampling $|\phi\rangle$ for values of q that we choose. This understanding was at the

heart of Shor's factoring algorithm when he showed that it is possible to still compute multiples of q/r using continued fractions. This principle has been generalized for arbitrary states $|\phi\rangle$ and is known as *approximate* Fourier sampling [23]. This process, described below, allows one to sample from a distribution which is close to the one that could be generated if a multiple of r were known. This will be required in section 5.2 for finding the period of the shifted character problem. In that case the shifted character problem will have a property very different from that of Shor's periodic function. In particular, the periodic function f in Shor's case takes r different values, whereas there are nontrivial cases of the shifted character problem when the function only takes *two* values (ignoring zero amplitudes, which appear in an exponentially small fraction of the amplitudes and thus are insignificant).

Approximate Fourier sampling works as follows: Let $|\phi\rangle = \sum_{x=0}^{n-1} \phi_x |x\rangle$ be an arbitrary superposition, and let $\hat{\mathcal{D}}_{|\phi\rangle}$ be the distribution induced by Fourier sampling $|\phi\rangle$ over $\mathbb{Z}/n\mathbb{Z}$. Let the superposition $|\tilde{\phi}\rangle = \sum_{x=0}^{q'-1} \phi_{x \bmod n} |x\rangle$ be $|\phi\rangle$ repeated until some arbitrary integer q', not necessarily a multiple of n. Let $\hat{\mathcal{D}}_{|\tilde{\phi}\rangle}$ be the distribution induced by Fourier sampling $|\tilde{\phi}\rangle$ over $\mathbb{Z}/q\mathbb{Z}$, where $q > q'$ and $\phi_x = 0$ if $x \geq q'$.

Since $\hat{\mathcal{D}}_{|\phi\rangle}$ is a distribution on $\mathbb{Z}/n\mathbb{Z}$ and $\hat{\mathcal{D}}_{|\tilde{\phi}\rangle}$ is a distribution on $\mathbb{Z}/q\mathbb{Z}$, we define new distributions over fractions which can be compared. Define $\hat{\mathcal{D}}^{\mathrm{RF}}_{|\phi\rangle}(j,k) = \hat{\mathcal{D}}_{|\phi\rangle}(jm)$ if $mk = n$. The distribution $\hat{\mathcal{D}}^{\mathrm{RF}}_{|\phi\rangle}$ is the distribution on the reduced fractions of $\hat{\mathcal{D}}_{|\phi\rangle}$ since it describes the process of sampling x from $\hat{\mathcal{D}}_{|\phi\rangle}$ and returning the fraction x/n in lowest terms.

Let $\hat{\mathcal{D}}^{\mathrm{CF}}_{|\tilde{\phi}\rangle}$ be the distribution induced on fractions from sampling $\hat{\mathcal{D}}_{|\tilde{\phi}\rangle}$ to obtain x, and then using continued fractions to compute the closest approximation to x/q with denominator at most n. It is a theorem that if $q' = \Omega(\frac{n^2}{\epsilon^2})$ and $q = \Omega(\frac{q'}{\epsilon})$, then $|\hat{\mathcal{D}}^{\mathrm{RF}}_{|\phi\rangle} - \hat{\mathcal{D}}^{\mathrm{CF}}_{|\tilde{\phi}\rangle}|_1 < \epsilon$ [23].

It is easy to apply this to the periodic function with period r that is injective on $\{0, \ldots, r-1\}$. Let $n = r$ and consider the state $|\phi\rangle = \sum_{i=1}^{r} |k\rangle$. The distribution $\hat{\mathcal{D}}^{\mathrm{RF}}_{|\phi\rangle}$ is uniform over $\{0, 1/r, 2/r, \ldots, (r-1)/r\}$. By the theorem, if a large enough value q is chosen, then $\hat{\mathcal{D}}^{\mathrm{CF}}_{|\tilde{\phi}\rangle}$ will be ϵ-close to this and efficiently computable.

2.4. Legendre symbol and Jacobi symbol. The Legendre symbol $\left(\frac{\cdot}{p}\right) : \mathbb{F}_p \to \{0, \pm 1\}$ is a quadratic multiplicative character of $\mathbb{F}_p$ defined by

$$\left(\frac{x}{p}\right) = \begin{cases} 0 & \text{if } x = 0, \\ +1 & \text{if } x \text{ is a nonzero square in } \mathbb{F}_p, \\ -1 & \text{if } x \text{ is not a square in } \mathbb{F}_p. \end{cases}$$

The Legendre symbol satisfies $\left(\frac{x}{p}\right) = x^{(p-1)/2} \bmod p$, which shows that we can efficiently compute the Legendre symbol using repeated squaring mod$p$.

The Jacobi symbol $\left(\frac{\cdot}{n}\right) : \mathbb{Z}/n\mathbb{Z} \to \{0, \pm 1\}$ is a quadratic multiplicative character of $\mathbb{Z}/n\mathbb{Z}$ with n an odd integer. It is defined so that it satisfies the relation $\left(\frac{a}{bc}\right) = \left(\frac{a}{b}\right)\left(\frac{a}{c}\right)$ and reduces to the Legendre symbol when the lower parameter is prime. With $n = p_1^{r_1} \cdots p_k^{r_k}$ and all p_i odd primes, this gives the definition $\left(\frac{x}{n}\right) = \left(\frac{x}{p_1}\right)^{r_1} \cdots \left(\frac{x}{p_k}\right)^{r_k}$ such that $\left(\frac{x}{n}\right) \neq 0$ if and only if $x \in \mathbb{Z}/n\mathbb{Z}^*$. The value of the Jacobi symbol can be calculated efficiently without factoring n using the *quadratic reciprocity theorem,* which states that $\left(\frac{m}{n}\right) = (-1)^{(m-1)(n-1)/4}\left(\frac{n}{m}\right)$ in combination with the rule that $\left(\frac{m}{n}\right) = \left(\frac{m'}{n}\right)$ if $m = m' \bmod n$.

2.5. Finite fields. The elements of a finite field $\mathbb{F}_q$ (where $q = p^r$ for some prime p) can be represented as polynomials in $\mathbb{F}_p[X]$ modulo a degree r irreducible polynomial in $\mathbb{F}_p[X]$. In this representation, addition, subtraction, multiplication, and division can all be performed in $O((\log q)^2)$ time [2].

We will need to compute the Fourier transform over the additive group of a finite field, which is isomorphic to $(\mathbb{Z}/p\mathbb{Z})^r$. The additive characters are of the form $\psi_y(x) = \omega_p^{\mathrm{Tr}(xy)}$, where $\mathrm{Tr} : \mathbb{F}_q \to \mathbb{F}_p$ is the trace of the finite field $\mathrm{Tr}(x) = \sum_{j=0}^{r-1} x^{p^j}$ and $y \in \mathbb{F}_q$ [30]. We can efficiently compute the Fourier transform over the additive group of a finite field. (The efficiency of this transform was independently shown in [4].)

An operation U approximates U' to within ϵ if for any unit vector $|\psi\rangle$, $||U|\psi\rangle - U'|\psi\rangle||_2 \leq \epsilon$.

LEMMA 2.2 (Fourier transform over $\mathbb{F}_q$). *The Fourier transform*

$$|x\rangle \mapsto \frac{1}{\sqrt{q}} \sum_{y \in \mathbb{F}_q} \omega_p^{\mathrm{Tr}(xy)} |y\rangle$$

for all $x \in \mathbb{F}_q$ can be approximated to within error ϵ in time polynomial in $\log q$ and $\log 1/\epsilon$.

Proof. Let $q = p^r$, where p is the prime number that denotes the base field: $\mathbb{F}_q = \mathbb{F}_p[X]/f(X)$, with $f(X)$ an irreducible polynomial of degree r. Assume that the mapping $|x\rangle \mapsto \bigotimes_{j=0}^{r-1} |\mathrm{Tr}(xX^j)\rangle$ can be computed in polynomial time. First apply this map and then compute the Fourier transform over $(\mathbb{Z}/p\mathbb{Z})^r$. This gives us the final state

$$\bigotimes_{j=0}^{r-1} \frac{1}{\sqrt{p}} \sum_{y_j \in \mathbb{F}_p} \omega_p^{\mathrm{Tr}(xX^j)y_j} |y_j\rangle = \frac{1}{\sqrt{q}} \sum_{y \in \mathbb{F}_q} \omega_p^{\mathrm{Tr}(xy)} |y\rangle.$$

We first show that the map $|x\rangle \mapsto |\mathrm{Tr}(x), \mathrm{Tr}(xX), \ldots, \mathrm{Tr}(xX^{r-1})\rangle$ is reversible and then that it can be computed in polynomial time. Let $T(x) = (\mathrm{Tr}(x), \mathrm{Tr}(xX), \ldots, \mathrm{Tr}(xX^{r-1}))$. T is additive since Tr is; thus if $T(a) = T(b)$, then $T(a-b)$ is the zero vector. If T is not one-to-one, there is a nonzero x with $T(x)$ equal to the zero vector. Since Tr is not the zero map, choose $a \in \mathbb{F}_q$ such that $\mathrm{Tr}(a) \neq 0$. Choose elements of the base field $z_0, \ldots, z_{r-1}$ such that $x \cdot \sum_j z_j X^j = a$ (these must exist because x is nonzero). Then $\mathrm{Tr}(a) = \mathrm{Tr}(\sum_j z_j x X^j) = \sum_j z_j \mathrm{Tr}(xX^j) = 0$, since $\mathrm{Tr}(xX^j) = 0$ for all j. But this contradicts $\mathrm{Tr}(a) \neq 0$. Thus T is one-to-one.

We now show that the map is computable in polynomial time. Write $x = \sum_{j=0}^{r-1} x_j X^j$, where the x_j are from the base field of $\mathbb{F}_q$. Then for the trace $\mathrm{Tr}(xX^k) = \sum_{j=0}^{r-1} x_j \mathrm{Tr}(X^{j+k})$; hence the components of $T(x)$ are linear combinations of the x_js and thus can be computed in polynomial time.

So far we have assumed that all operations are performed exactly. As we observed earlier in Lemma 2.1 we can approximate the powers of ω_p to within ϵ in time polynomial in $\log p$ and $\log 1/\epsilon$. The Fourier transform over $(\mathbb{Z}/p\mathbb{Z})^r$ can also be approximated to within ϵ in time polynomial in $\log p^r$ and $\log 1/\epsilon$. ∎

For clarity of exposition we assume throughout the rest of the paper that the Fourier transform over $\mathbb{F}_q$ can be performed exactly, as we can make the errors due to the approximation exponentially small with only polynomial overhead.

2.6. Multiplicative characters and their Fourier transforms. The multiplicative group $\mathbb{F}_q^*$ of a finite field $\mathbb{F}_q$ is cyclic. Let g be a generator of $\mathbb{F}_q^*$. Then the

multiplicative characters of $\mathbb{F}_q$ are of the form $\chi(g^\ell) = \omega_{q-1}^{k\ell}$ for all $\ell \in \{0, \ldots, q-2\}$, where the $q-1$ different multiplicative characters are indexed by $k \in \{0, \ldots, q-2\}$. The trivial character is the character with $k = 0$. We can extend the definition of χ to $\mathbb{F}_q$ by defining $\chi(0) = 0$. On a quantum computer we can efficiently compute $\chi(x)$ because the value is determined by the discrete logarithm $\log_g(x)$, which can be computed efficiently using Shor's algorithm [34]. The Fourier transform of a multiplicative character χ of the finite field $\mathbb{F}_q$ is given by $\hat{\chi}(0) = 0$, and $\hat{\chi}(y) = \overline{\chi(y)}\hat{\chi}(1) = \omega_{q-1}^{-k\ell} \sum_j \omega_{q-1}^{kj} \omega_p^{\mathrm{Tr}(g^j)}$, where $y = g^\ell$ [30, 36].

Let $n = p_1^{m_1} \cdots p_k^{m_k}$ be the prime factorization of n. Then by the Chinese remainder theorem, $(\mathbb{Z}/n\mathbb{Z})^* \cong (\mathbb{Z}/p_1^{m_1}\mathbb{Z})^* \times \cdots \times (\mathbb{Z}/p_k^{m_k}\mathbb{Z})^*$. Every multiplicative character χ of $\mathbb{Z}/n\mathbb{Z}$ can be written as the product $\chi(x) = \chi_1(x_1) \cdots \chi_k(x_k)$, where χ_i is a multiplicative character of $\mathbb{Z}/p_i^{m_i}\mathbb{Z}$ and $x_i \equiv x \bmod p_i^{m_i}$. We say χ is *completely nontrivial* if each of the χ_i is nontrivial. We extend the definition of χ to all of $\mathbb{Z}/n\mathbb{Z}$ by defining $\chi(y) = 0$ if $\gcd(y, n) \neq 1$. The character χ is aperiodic on $\{0, \ldots, n-1\}$ if and only if all its χ_i factors are aperiodic over their respective domains $\{0, \ldots, p_i^{m_i} - 1\}$. We call χ a *primitive character* if it is completely nontrivial and aperiodic. Hence, χ is primitive if and only if all its χ_i terms are primitive.

If χ is primitive, the Fourier transform of χ is the product of the Fourier transform of its components and has an expression analogous to the Fourier transform of a multiplicative transform of a finite field. That is,

$$\begin{aligned}\hat{\chi}(y) &= \hat{\chi}_1(y_1) \cdots \hat{\chi}_k(y_k) \\ &= \overline{\chi_1(y_1)}\hat{\chi}_1(1) \cdots \overline{\chi_k(y_k)}\hat{\chi}_k(1) \\ &= \overline{\chi_1(y_1) \cdots \chi_k(y_k)}\hat{\chi}_1(1) \cdots \hat{\chi}_k(1) \\ &= \overline{\chi}(y)\hat{\chi}(1).\end{aligned}$$

If χ is completely nontrivial but periodic with period ℓ, let χ' be the primitive character of $\mathbb{Z}/\ell\mathbb{Z}$ given by $\chi'(x) = \chi(x)$ for $x \in \{0, \ldots, \ell - 1\}$. The Fourier transform of χ is then given by

$$\hat{\chi}(y) = \begin{cases} 0 & \text{if } n/\ell \nmid y, \\ \hat{\chi}'(y\ell/n) = \overline{\chi'(y\ell/n)}\hat{\chi}'(1) & \text{if } n/\ell \mid y. \end{cases}$$

See the book by Tolimieri, An, and Lu [36] for details.

3. The intuition behind the algorithms for the hidden shift problem. We give some intuition for the ideas behind our algorithms for the hidden shift problem. We use the shifted Legendre symbol problem as our running example, but the approach works more generally. In the shifted Legendre symbol problem we are given a function $f_s : \mathbb{Z}/p\mathbb{Z} \to \{0, \pm 1\}$ such that $f_s(x) = \left(\frac{x+s}{p}\right)$, and are asked to find s.

The algorithm starts by putting the function value in the phase to get $|f_s\rangle = \sum_x f_s(x)|x\rangle = \sum_x \left(\frac{x+s}{p}\right)|x\rangle$. For random f we can expect the functions f_z to be mutually (near) orthogonal, so that the inner product squared $|\langle f_z|f_s\rangle|^2$ approximates the delta function $\delta_s(z)$. The Legendre sequence $\left(\frac{0}{p}\right), \left(\frac{1}{p}\right), \ldots, \left(\frac{p-1}{p}\right)$ has many pseudo-random properties, and for its autocorrelation we have, in fact, $|\langle f_z|f_s\rangle|^2 = \delta_s(z) - \frac{1}{p}$. With this, we define the (near) unitary matrix C, where the zth row is $|f_{-z}\rangle$. The state $|f_s\rangle$ is one of the rows; hence $C|f_s\rangle = |-s\rangle$. The problem then reduces to the following: How do we efficiently implement C? By definition, C is a circulant

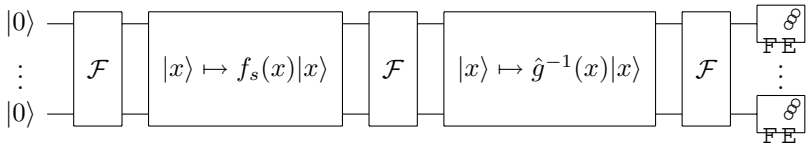

FIG. 3.1. *Circuit for hidden shift problem for a known function g and an unknown shift s of the black-box $f_s(x) = g(x+s)$. Notice that the function values of f_s and $\hat{g}^{-1}$ are computed into the phase.*

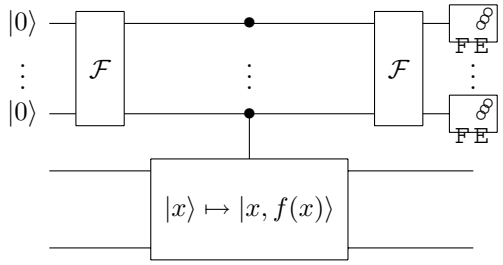

FIG. 3.2. *Circuit for hidden subgroup problem. Here f is computed into a register.*

matrix ($c_{x,y} = f_{-x}(y) = f_0(y-x) = f_{-(x+1)}(y+1) = c_{x+1,y+1}$). Since the Fourier transform matrix diagonalizes a circulant matrix, we can write $C = \mathcal{F}(\mathcal{F}^{-1}C\mathcal{F})\mathcal{F}^{-1} = \mathcal{F}D\mathcal{F}^{-1}$, where D is diagonal. Thus we can implement C if we can implement D. The vector on the diagonal of D is the vector $\mathcal{F}^{-1}|f_0\rangle = \mathcal{F}^{-1}\sum_x \left(\frac{x}{p}\right)|x\rangle$, the inverse Fourier transform of the Legendre symbol. The Legendre symbol is an eigenvector of the Fourier transform, so the diagonal matrix contains the values of the Legendre symbol times a global constant that can be ignored. Because the Legendre symbol can be computed efficiently classically, it can be computed into the phase, so C can be implemented efficiently.

In summary, to implement C for the hidden shift problem for the Legendre symbol, compute the Fourier transform, compute $\left(\frac{x}{p}\right)$ into the phase at $|x\rangle$, and then compute the Fourier transform again (it is not important whether we use $\mathcal{F}$ or $\mathcal{F}^{-1}$). Figure 3.1 shows a circuit diagram outlining the algorithm for the hidden shift problem for a general function g. Contrast this with the circuit for the hidden subgroup problem shown in Figure 3.2.

4. Shifted multiplicative characters of finite fields. In this section we show how to solve the hidden shift problem for any nontrivial multiplicative character of a finite field. The Fourier transform we use is the Fourier transform over the additive group of the finite field.

DEFINITION 4.1 (shifted multiplicative character problem over $\mathbb{F}_q$). *Given a nontrivial multiplicative character $\chi : \mathbb{F}_q \to \mathbb{C}$ (where $q = p^r$ for some prime p) and a black-box function f for which there is an s such that $f(x) = \chi(x+s)$ for all x, find s.*

ALGORITHM 1 (shifted multiplicative character problem over finite field $\mathbb{F}_q$).

1. Create $\sum_{x\in\mathbb{F}_q} \chi(x+s)|x\rangle$, using Lemma 2.1.

2. Compute the Fourier transform to obtain the state $\sum_{y\in\mathbb{F}_q}\omega_p^{\mathrm{Tr}(-sy)}\hat{\chi}(y)|y\rangle$, using Lemma 2.2.
3. For all $y\neq 0$, compute $\chi(y)$ into the phase to obtain $\hat{\chi}(1)\sum_{y\in\mathbb{F}_q^*}\omega_p^{\mathrm{Tr}(-sy)}|y\rangle$.
4. Compute the inverse Fourier transform and measure the outcome $-s$.

THEOREM 4.2. *For any finite field $\mathbb{F}_q$ and any nontrivial multiplicative character, Algorithm* 1 *solves the shifted multiplicative character problem over finite fields with probability* $(1-1/q)^2$.

Proof.

1. Since $\chi(x)=0$ only at $x=0$, by Lemma 2.1 we can create the superposition with probability $1-1/q$.
2. By Lemma 2.2 we can compute the Fourier transform efficiently. The Fourier transform moves the shift s into the phase as described.
3. Because $\hat{\chi}(y)=\overline{\chi(y)}\hat{\chi}(1)$ for every nonzero y, the phase change $|y\rangle\mapsto\chi(y)|y\rangle$ establishes the required transformation.
4. The amplitude of $|-s\rangle$ is

$$\frac{1}{\sqrt{q}}\frac{1}{\sqrt{q-1}}\sum_{y\in\mathbb{F}_q^*}\omega_p^{\mathrm{Tr}(-sy)}\omega_p^{\mathrm{Tr}(sy)}=\frac{1}{\sqrt{q}}\frac{1}{\sqrt{q-1}}\sum_{y\in\mathbb{F}_q^*}1$$
$$=\sqrt{\frac{q-1}{q}},$$

and thus the probability of measuring $-s$ is $1-1/q$. ∎

4.1. The Legendre symbol and homomorphic encryption. The quantum algorithm of the previous section showed us how we can determine the shift $s\in\mathbb{F}_p$ given the function $f_s(x)=\left(\frac{x+s}{p}\right)$. We now show how this algorithm enables us to break schemes for "algebraically homomorphic encryption."

A cryptosystem is *algebraically homomorphic* [9] if given the encryption of two plaintexts $E(x)$, $E(y)$ with $x,y\in\mathbb{F}_p$, an untrusted party can construct the encryption of the plaintexts $E(x+y)$ and $E(xy)$ in polynomial time. More formally, we have the secret encryption and decryption functions $E:\mathbb{F}_p\to S$ and $D:S\to\mathbb{F}_p$, in combination with the public add and multiplication transformations $A:S^2\to S$ and $M:S^2\to S$ such that $D(A(E(x),E(y)))=x+y$ and $D(M(E(x),E(y)))=xy$ for all $x,y\in\mathbb{F}_p$. We assume that the functions E, D, A, and M are deterministic. This definition is slightly more general than the definition in [9, Definition 4.1] because we require equality between texts after unencryption rather than equality between encrypted texts. In other words, the decryption function may be many-to-one. As a result the encryption of a given number can vary depending on how the number is constructed. For example, $A(E(4),E(2))$ may not be equal to $M(E(2),E(3))$. In addition to the public A and M functions, we also assume the existence of a public zero tester $Z:S\to\{0,1\}$, with $Z(E(x))=0$ if $x=0$, and $Z(E(x))=1$ otherwise. In [9] the existence of a zero tester is trivial because the decryption function is injective.

An algebraically homomorphic cryptosystem is a cryptographic primitive that enables two players to perform noninteractive secure function evaluation. It is an open problem whether or not such a cryptosystem can be constructed. We say we can break such a cryptosystem if, given $E(s)$, we can recover s in time polylog(p) with the help of the public functions A, M, and Z. The best known classical attack, due to Boneh and Lipton [9], has expected running time $O\left(\exp\left(c\sqrt{\log p\log\log p}\right)\right)$ for the field $\mathbb{F}_p$ and is based on a smoothness assumption.

Suppose we are given the ciphertext $E(s)$. Test $E(s)$ using the Z function. If s is not zero, create the encryption $E(1)$ via the identity $x^{p-1} \equiv 1 \bmod p$, which holds for all nonzero x. In particular, using $E(s)$ and the M function, we can use repeated squaring and compute $E(s)^{p-1} = E(1)$ in $\log p$ steps.

Clearly, from $E(1)$ and the A function we can construct $E(x)$ for every $x \in \mathbb{F}_p$. Then, given such an $E(x)$, we can compute $f(x) = \left(\frac{x+s}{p}\right)$ in the following way. Add $E(s)$ and $E(x)$, yielding $E(x+s)$, and then compute the encrypted $(p-1)/2$th power of $x+s$, giving $E(\left(\frac{x+s}{p}\right))$. Next, add $E(0)$, $E(-1)$, or $E(1)$ and test if it is an encryption of zero, and return 0, 1, or -1 accordingly. Applying this method on a superposition of $|x\rangle$ states, we can create (after reversibly uncomputing the garbage of the algorithm) the state $\frac{1}{\sqrt{p-1}} \sum_x f_s(x)|x\rangle$. We can then recover s by using Algorithm 1.

COROLLARY 4.3. *Given an efficient test to decide if a value is an encryption of zero, Algorithm* 1 *can be used to break any algebraically homomorphic encryption system.*

We can also break algebraically homomorphic cryptosystems using Shor's discrete log algorithm as follows. Suppose g is a generator for $\mathbb{F}_p^*$ and that we are given the unknown ciphertext $E(g^s)$. Create the superposition $\sum_{i,j} |i, j, E(g^{si+j})\rangle$ and then append the state $|\psi_{si+j}\rangle = \sum_t \left(\frac{g^{si+j}+t}{p}\right)|t\rangle$ to the superposition in i, j by the procedure described above. Next, uncompute the value $E(g^{si+j})$, which gives $\sum_{i,j} |i, j\rangle|\psi_{si+j}\rangle$. Rewriting this as $\sum_{i,r} |i, r - si\rangle|\psi_r\rangle$ and observing that the ψ_r are almost orthogonal, we see that we can apply the methods used in Shor's discrete log algorithm to recover s and thus g^s.

5. Shifted multiplicative characters of finite rings. In this section we show how to solve the shifted multiplicative character problem for $\mathbb{Z}/n\mathbb{Z}$ for any completely nontrivial multiplicative character of the ring $\mathbb{Z}/n\mathbb{Z}$ and extend this to the case when n is unknown. Unlike in the case for finite fields, the characters may be periodic. Thus the shift may not be unique. The Fourier transform is now the familiar Fourier transform over the additive group $\mathbb{Z}/n\mathbb{Z}$.

5.1. Shifted multiplicative characters of $\mathbb{Z}/n\mathbb{Z}$ for known n. We start with the following definition.

DEFINITION 5.1 (shifted multiplicative character problem over $\mathbb{Z}/n\mathbb{Z}$). *Given χ, a completely nontrivial multiplicative character of $\mathbb{Z}/n\mathbb{Z}$, and a function f for which there is an s such that $f(x) = \chi(x + s)$ for all x, find all t satisfying $f(x) = \chi(x + t)$ for all x.*

Multiplicative characters of $\mathbb{Z}/n\mathbb{Z}$ may be periodic, so to solve the shifted multiplicative character problem we first find the period and then we find the shift. If the period is ℓ, then the possible shifts will be $\{s, s + \ell, s + 2\ell, \dots\}$. Note that step 1 of Algorithm 2, which computes the period of χ, uses different properties of a periodic function than Shor's algorithm. In particular, when χ is the Legendre symbol, the function takes only three values, whereas Shor's algorithm assumes functions are injective in $\{0, \dots, r - 1\}$ when the period is r.

ALGORITHM 2 (shifted multiplicative character problem over $\mathbb{Z}/n\mathbb{Z}$).

1. Find the period ℓ of χ. Let χ' be χ restricted to $\{0, \dots, \ell - 1\}$.
   (a) Create $\sum_{x=0}^{n-1} \chi(x + s)|x\rangle$ using f.
   (b) Compute the Fourier transform over $\mathbb{Z}/n\mathbb{Z}$ to obtain the superposition $\sum_{y=0}^{\ell-1} \omega_\ell^{-sy} \hat{\chi}'(y)|yn/\ell\rangle$.
   (c) Measure $|yn/\ell\rangle$. Compute $n/\ell = \gcd(n, yn/\ell)$.

2. Find s using the period ℓ and χ'.
   (a) Create $\sum_{x=0}^{\ell-1} \chi'(x+s)|x\rangle$.
   (b) Compute the Fourier transform over $\mathbb{Z}/\ell\mathbb{Z}$ to obtain $\sum_y \omega_\ell^{-sy}\hat{\chi}'(y)|y\rangle$.
   (c) For all y coprime to ℓ, compute $\hat{\chi}'(y)^{-1}$ into the phase to obtain $\sum_{y:\hat{\chi}'(y)\neq 0} \omega_\ell^{-sy}|y\rangle$.
   (d) Compute the inverse Fourier transform and measure.

THEOREM 5.2. *Algorithm 2 solves the shifted multiplicative character problem over $\mathbb{Z}/n\mathbb{Z}$ for completely nontrivial multiplicative characters of $\mathbb{Z}/n\mathbb{Z}$ in polynomial time with probability at least $(\phi(n)/n)^3 = \Omega((\frac{1}{\log\log n})^3)$, where ϕ is Euler's totient function; $\phi(n)$ is the number of positive integers less than n that are coprime to n.*

Proof. Note that because χ is completely nontrivial, χ' is a primitive character of $\mathbb{Z}/\ell\mathbb{Z}$.

1. (a) $\chi(x+s)$ is nonzero exactly when $\gcd(x+s,n)=1$; thus by Lemma 2.1 we can create the superposition with probability $\phi(n)/n$.
   (b) Since χ has period ℓ, the Fourier transform is nonzero only on multiples of n/ℓ.
   (c) Since $\hat{\chi}'(y) = \overline{\chi'(y)}\hat{\chi}'(1)$, and $\chi'(y)$ is nonzero precisely when $\gcd(y,n) = 1$, when we measure yn/ℓ we have $n/\ell = \gcd(n, yn/\ell)$.
2. (a) Similar to the argument above, we can create the superposition with probability $\phi(\ell)/\ell$.
   (b) The Fourier transform moves the shift s into the phase.
   (c) As in the case for the finite field, this can be done by computing the phase of $\chi'(y)$ into the phase of $|y\rangle$.
   (d) Let $A = \{y \in \mathbb{Z}/\ell\mathbb{Z} : \hat{\chi}'(y) \neq 0\}$. $A = (\mathbb{Z}/\ell\mathbb{Z})^*$ and thus $|A| = \phi(\ell)$. Then the amplitude of $|-s\rangle$ after the Fourier transform is

$$\frac{1}{\sqrt{\phi(\ell)}}\frac{1}{\sqrt{\ell}}\left(\sum_{y\in A}\omega_\ell^{-ys}\,\omega_\ell^{ys}\right) = \frac{1}{\sqrt{\phi(\ell)}}\frac{1}{\sqrt{\ell}}\left(\sum_{y\in A}1\right) = \sqrt{\frac{\phi(\ell)}{\ell}}.$$

   Hence the probability of measuring $|-s\rangle$ is $\phi(\ell)/\ell$.

Thus the algorithm succeeds with probability $(\phi(n)/n)(\phi(\ell)/\ell)^2$, which is lower bounded by $\Omega((\frac{1}{\log\log n})^3)$ (because of the bound $\phi(n) = \Omega(n/\log\log n)$). ∎

5.2. Shifted multiplicative characters of $\mathbb{Z}/n\mathbb{Z}$ for unknown n. We now consider the case when n is unknown.

DEFINITION 5.3 (shifted multiplicative character problem over $\mathbb{Z}/n\mathbb{Z}$ with unknown n). *Given a completely nontrivial multiplicative character $\chi : \mathbb{Z}/n\mathbb{Z} \to \mathbb{C}$ for some unknown n, and a function f for which there is an s such that $f(x) = \chi(x+s)$ for all x, find all t satisfying $f(x) = \chi(x+t)$ for all x.*

THEOREM 5.4. *Given an upper bound on the size of the period of f, we can efficiently solve the shifted multiplicative character problem over $\mathbb{Z}/n\mathbb{Z}$ for unknown n on a quantum computer.*

Proof. Let ℓ be the period of f and χ' be χ restricted to $\mathbb{Z}/\ell\mathbb{Z}$. Using the Fourier sampling algorithm described in section 2.3, we can approximately Fourier sample f over $\mathbb{Z}/\ell\mathbb{Z}$. Because $\chi'(y)$ is nonzero precisely when $\gcd(y,\ell) = 1$, this Fourier sampling algorithm returns y/ℓ with high probability, where y is coprime to ℓ. Thus we can find ℓ with high probability. Next, apply Algorithm 2 to find $s \bmod \ell$. ∎

6. The hidden coset problem. In this section we define the hidden coset problem and give an algorithm for solving the problem for abelian groups under certain conditions on the functions. As we will show, this problem abstracts out two properties that appeared in the hidden shift problems in earlier sections. We will also show how finding the coset representative can be interpreted as solving a deconvolution problem.

DEFINITION 6.1 (hidden coset problem). *Given functions f and g defined on a group G such that for some $s \in G$, $f(x) = g(x+s)$ for all x in G, find the set of all t satisfying $f(x) = g(x+t)$ for all x in G. The function f is given as an oracle, and g is known but not necessarily efficiently computable.*

LEMMA 6.2. *The answer to the hidden coset problem is a coset of some subgroup H of G, and g is constant on cosets of H.*

Note that g is not necessarily a hidden subgroup problem instance because while it is constant on cosets, it does not have to be distinct on different cosets.

Proof. Let $S = \{t \in G : f(x) = g(x+t) \text{ for all } x \in G\}$ be the set of all solutions and let H be the largest subgroup of G such that g is constant on cosets of H. Clearly this is well defined (note that H may be the trivial subgroup as in the shifted Legendre symbol problem). Suppose t_1, t_2 are in S. Then we have $g(x+(-t_2+t_1)) = g((x-t_2)+t_1) = f(x-t_2) = g((x-t_2)+t_2) = g(x)$ for all x in G; thus $-t_2+t_1$ is in H. This shows that S is contained in a coset of H. Since s is in S we must have that S is contained in $s+H$. Conversely, suppose $s+h$ is in $s+H$ (where h is in H). Then $g(x+s+h) = g(x+s) = f(x)$ for all x in G; hence $s+h$ is in S. It follows that $S = s+H$. While this proof was written with additive notation, it carries through if the group is nonabelian. ∎

A familiar example of a nonabelian case of the hidden coset problem is given by the graph isomorphism problem. For each n vertex graph X we let $M(X) \in \{0,1\}^{n\times n}$ denote the adjacency matrix of X, and given X we define the function $g : S_n \to \{0,1\}^{n\times n}$ by $g : \sigma \mapsto M(\sigma(X))$ for all permutations $\sigma \in S_n$ in the symmetric group. Now, the shifted function $f : S_n \to \{0,1\}^{n\times n}$ coincides with a description of the permuted adjacency matrix $M(\pi(X))$ and has $f(\sigma) = M(\sigma \cdot \pi(X))$ for all σ. This shows how the search for an element $\sigma' \in S_n$ such that $f(\sigma) = g(\sigma \cdot \sigma')$ is identical to the search for a permutation that transforms X to $\pi(X)$: the GRAPH ISOMORPHISM problem. Moreover, the determination of the constant subgroup of g of all the permutations σ that have $M(X) = M(\sigma(X))$ solves the GRAPH AUTOMORPHISM problem of the graph X.

Unlike for the hidden subgroup problem, we can prove that there are cases of the hidden coset problem that cannot be solved efficiently on a quantum computer. Let $g : \mathbb{Z}/n\mathbb{Z} \to \{0,1\}$ be the delta function δ_0 with $g(0) = 1$, and $g(x) = 0$ otherwise. Consequently, f_s will be the unknown delta function δ_{-s}, which determines the hidden coset $\{s\}$. However, given f_s and g, finding s amounts to searching a list of n items, which requires $\Omega(\sqrt{n})$ queries to f_s [5].

The algorithm for the hidden coset problem instances that we can solve consists of two parts: identifying the subgroup on which g is constant and finding a coset representative, where computing a coset representative corresponds to computing one hidden shift. The algorithms in this article that compute the subgroup and a coset representative exploit different facets of the power of the quantum Fourier transform. After computing a Fourier transform, the subgroup structure is captured in the magnitude, whereas the shift structure is captured in the phase. In the hidden subgroup problem we measure after computing the Fourier transform and so discard information about shifts. Our algorithms for hidden shift problems do additional processing

to take advantage of the information encoded in the phase. Thus the solution to the hidden coset problem requires fully utilizing the abilities of the Fourier transform.

6.1. Identifying the unknown subgroup. The first step of the hidden coset problem algorithm is to compute the unknown subgroup of g. As the examples in the previous section show, computing the subgroup may be difficult for at two least reasons. First, g may be an HSP instance over a nonabelian group, for which no efficient algorithm is known. Second, while g is constant on cosets, it may not be distinct on different cosets; that is, it may not be an HSP instance.

We start by finding the subgroup H. We need two different algorithms for determining H: the "standard" algorithm for the hidden subgroup problem and the algorithm we used in section 5.

In the standard algorithm for the hidden subgroup problem we form a superposition over all inputs, compute $g(x)$ into a register, measure the function value, compute the Fourier transform, and then sample. The standard algorithm may fail when g is not distinct on different cosets of H. In such cases, we need other restrictions on g to be able to find the hidden subgroup H using the standard algorithm. Boneh and Lipton [8], Mosca and Ekert [32], and Hales and Hallgren [23] have all given criteria under which the standard hidden subgroup algorithm outputs H even when g is not distinct on different cosets of H.

In section 5 we used a different algorithm to determine H because the function we were considering did not satisfy the conditions mentioned above. In this algorithm we compute the value of g into the amplitude, Fourier transform, and then sample, whereas in the standard hidden subgroup algorithm we compute the value of g into a register. In general, this algorithm works when the fraction of values for which $\hat{g}$ is zero is sufficiently small and the nonzero values of $\hat{g}$ have constant magnitude.

6.2. Finding a coset representative as a deconvolution problem. Once we have identified H, we can find a coset representative by solving the associated hidden coset problem for f' and g', where f' and g' are defined on the quotient group G/H and are consistent in the natural way with f and g. For notational convenience we assume that f and g are defined on G and that H is trivial, that is, the shift is uniquely defined.

The hidden shift problem may be interpreted as a *deconvolution* problem. In a deconvolution problem, we are given functions g and $f = g \star h$ (the convolution of g with some unknown function h) and asked to find this h. Let $\delta_y(x) = \delta(x - y)$ be the delta function centered at y. In the hidden shift problem, f is the convolution of δ_{-s} and g, that is, $f = g \star \delta_{-s}$. Finding s or, equivalently, finding δ_{-s}, given f and g, is therefore a deconvolution problem.

Recall that under the Fourier transform convolution becomes pointwise multiplication. Thus, taking Fourier transforms, we have $\hat{f} = \hat{g} \cdot \hat{\delta}_{-s}$ and hence $\hat{\delta}_{-s} = \hat{g}^{-1} \cdot \hat{f}$, provided that $\hat{g}$ is everywhere nonzero. For the multiplication by $\hat{g}^{-1}$ to be performed efficiently on a quantum computer would require $\hat{g}$ to have constant magnitude and be everywhere nonzero. However, even if only a fraction of the values of $\hat{g}$ are zero we can still approximate division of $\hat{g}$ by only dividing when $\hat{g}$ is nonzero and doing nothing otherwise. The zeros of $\hat{g}$ correspond to loss of information about δ_{-s}.

ALGORITHM 3.

1. Create $\sum_{x \in G} g(x + s)|x\rangle$.
2. Compute the Fourier transform to obtain $\sum_{y \in G} \overline{\psi_y(s)} \hat{g}(\psi_y)|y\rangle$, where ψ_y are the characters of the group G.

3. For all y for which $\hat{g}(\psi_y)$ is nonzero compute $\hat{g}(\psi_y)^{-1}$ into the phase of $|y\rangle$ to obtain $\sum_{y,\hat{g}(\psi_y)\neq 0} \overline{\psi_y(s)}|y\rangle$.
4. Compute the inverse Fourier transform and measure to obtain $-s$.

THEOREM 6.3. *Suppose f and $\hat{g}$ are efficiently computable, the magnitude of $f(x)$ is constant for all values of x in G for which $f(x)$ is nonzero, and the magnitude of $\hat{g}(\psi_y)$ is constant for all values of ψ_y in $\hat{G}$ for which $\hat{g}(\psi_y)$ is nonzero. Let α be the fraction of x in G for which $f(x)$ is nonzero and let β be the fraction of ψ_y in $\hat{G}$ for which $\hat{g}(\psi_y)$ is nonzero. Then Algorithm 3 outputs $-s$ with probability $\alpha\beta$.*

Proof.

1. By Lemma 2.1 we can create the superposition with probability α.
2. The Fourier transform moves the shift s into the phase.
3. Because $\hat{g}$ has constant magnitude, for values where $\hat{g}$ is nonzero, $\hat{g}(\psi_y)^{-1} = C\overline{\hat{g}(\psi_y)}$ for some constant C. So we can perform this step by computing the phase of $\overline{\hat{g}}$ into the phase. For the values where $\hat{g}$ is zero we can just leave the phase unchanged as those terms are not present in the superposition.
4. Let $A = \{y \in G : \hat{g}(\psi_y) \neq 0\}$. Then the amplitude of $|-s\rangle$ is

$$\begin{aligned}\frac{1}{\sqrt{|A|}}\frac{1}{\sqrt{|G|}}\left(\sum_{y\in A}\overline{\psi_y(s)}\,\overline{\psi_y(-s)}\right) &= \frac{1}{\sqrt{|A|}}\frac{1}{\sqrt{|G|}}\left(\sum_{y\in A}1\right)\\ &= \sqrt{\frac{|A|}{|G|}}\\ &= \sqrt{\beta}.\end{aligned}$$

Hence we measure $|-s\rangle$ with probability β.

Thus the algorithm succeeds in identifying s with probability $\alpha\beta$ and requires only one query of f and one query of $\hat{g}$. ∎

6.3. Examples. We show how the hidden shift problems we considered earlier fit into the framework of the hidden coset problem. In the shifted multiplicative character problem over finite fields, G is the additive group of $\mathbb{F}_q$, $g = \chi$, and H is trivial since the shift is unique for nontrivial χ. In the shifted multiplicative character problem over $\mathbb{Z}/n\mathbb{Z}$, G is the additive group of $\mathbb{Z}/n\mathbb{Z}$, $g = \chi$, and H is the subgroup $\{0, \ell, \ldots, n/\ell\}$, where ℓ (which is a factor of n) is the period of χ. In the shifted period multiplicative character problem over $\mathbb{Z}/n\mathbb{Z}$ for unknown n, G is the additive group of $\mathbb{Z}$, $g = \chi$, and H is the infinite subgroup $\ell\mathbb{Z}$.

Acknowledgments. We would like to thank the anonymous referee who pointed out the application of the shifted Legendre symbol problem to algebraically homomorphic cryptosystems, and Umesh Vazirani, whose many suggestions greatly improved this paper. We also thank Dylan Thurston and an anonymous referee for pointing out that algebraically homomorphic cryptosystems can be broken using Shor's algorithm for discrete log. Thanks to Lisa Hales for helpful suggestions.

REFERENCES

[1] M. ABADI AND J. FEIGENBAUM, *Secure circuit evaluation. A protocol based on hiding information from an oracle*, J. Cryptology, 2 (1990), pp. 1–12.

[2] E. BACH AND J. SHALLIT, *Algorithmic Number Theory—Efficient Algorithms*, Vol. I, MIT Press, Cambridge, MA, 1996.

[3] R. Beals, *Quantum computation of Fourier transforms over symmetric groups*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, 1997, pp. 48–53.
[4] J. Niel de Beaudrap, R. Cleve, and J. Watrous, *Sharp quantum versus classical query complexity separations*, Algorithmica, 34 (2002), pp. 449–461.
[5] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, *Strengths and weaknesses of quantum computing*, SIAM J. Comput., 26 (1997), pp. 1510–1523.
[6] E. Bernstein and U. Vazirani, *Quantum complexity theory*, SIAM J. Comput., 26 (1997), pp. 1411–1473.
[7] M. Blum and S. Micali, *How to generate cryptographically strong sequences of pseudo-random bits*, SIAM J. Comput., 13 (1984), pp. 850–864.
[8] D. Boneh and R. J. Lipton, *Quantum cryptanalysis of hidden linear functions*, in Advances in Cryptology—CRYPTO '95, Lecture Notes in Comput. Sci. 963, Springer-Verlag, Berlin, 1995, pp. 424–437.
[9] D. Boneh and R. J. Lipton, *Algorithms for black-box fields and their application to cryptography*, in Advances in Cryptology—CRYPTO '96, Lecture Notes in Comput. Sci. 1109, Springer-Verlag, Berlin, 1996, pp. 283–297.
[10] R. Cleve, *The query complexity of order-finding*, Inform. and Comput., 192 (2004), pp. 162–171.
[11] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, *Quantum algorithms revisited*, R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci., 454 (1998), pp. 339–354.
[12] R. Cleve and J. Watrous, *Fast parallel circuits for the quantum Fourier transform*, in Proceedings of the 41st Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 2000, pp. 526–536.
[13] W. van Dam, *Quantum algorithms for weighing matrices and quadratic residues*, Algorithmica, 34 (2002), pp. 413–428.
[14] W. van Dam and S. Hallgren, *Efficient Quantum Algorithms for Shifted Quadratic Character Problems*, http://www.arxiv.org/abs/quant-ph/0011067 (2000).
[15] W. van Dam, S. Hallgren, and L. Ip, *Quantum algorithms for some hidden shift problems*, in Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 489–498.
[16] W. van Dam and G. Seroussi, *Efficient Quantum Algorithms for Estimating Gauss Sums*, http://www.arxiv.org/abs/quant-ph/0207131 (2002).
[17] I. B. Damgård, *On the randomness of Legendre and Jacobi sequences*, in Advances in Cryptology—CRYPTO'88, Lecture Notes in Comput. Sci. 403, Springer-Verlag, Berlin, 1990, pp. 163–172.
[18] M. Ettinger and P. Høyer, *On quantum algorithms for noncommutative hidden subgroups*, Adv. in Appl. Math., 25 (2000), pp. 239–251.
[19] M. Ettinger, P. Høyer, and E. Knill, *Hidden Subgroup States Are Almost Orthogonal*, http://www.arxiv.org/abs/quant-ph/9901034 (1999).
[20] K. Friedl, F. Magniez, M. Santha, and P. Sen, *Quantum testers for hidden group properties*, in Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Comput. Sci. 2747, Springer-Verlag, Berlin, 2003, pp. 419–428.
[21] M. Grigni, L. Schulman, M. Vazirani, and U. Vazirani, *Quantum mechanical algorithms for the nonabelian hidden subgroup problem*, Combinatorica, 24 (2004), pp. 137–154.
[22] L. Hales, *The Quantum Fourier Transform and Extensions of the Abelian Hidden Subgroup Problem*, Ph.D. thesis, University of California-Berkeley, Berkeley, CA, 2002.
[23] L. Hales and S. Hallgren, *An improved quantum Fourier transform algorithms and applications*, in Proceedings of the 41st Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 2000, pp. 515–525.
[24] S. Hallgren, *Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 2002, pp. 653–658.
[25] S. Hallgren, A. Russell, and A. Ta-Shma, *The hidden subgroup problem and quantum computation using group representations*, SIAM J. Comput., 32 (2003), pp. 916–934.
[26] L. Ip, *Solving Shift Problems and the Hidden Coset Problem Using the Fourier Transform*, http://www.arxiv.org/abs/quant-ph/0205034 (2002).
[27] G. Ivanyos, F. Magniez, and M. Santha, *Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem*, Internat. J. Found. Comput. Sci., 14 (2003), pp. 723–739.
[28] A. Yu. Kitaev, *Quantum Measurements and the Abelian Stabilizer Problem*, http://www.arxiv.org/abs/quant-ph/9511026 (1995).

[29] A. Yu. Kitaev, *Quantum computations: Algorithms and error correction*, Russian Math. Surveys, 52 (1997), pp. 1191–1249.
[30] R. Lidl and H. Niederreiter, *Finite Fields*, 2nd ed., Encyclopedia Math. Appl. 20, Cambridge University Press, Cambridge, UK, 1997.
[31] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptology, CRC Press, Boca Raton, FL, 1997.
[32] M. Mosca and A. Ekert, *The hidden subgroup problem and eigenvalue estimation on a quantum computer*, in Proceedings of the 1st NASA International Conference on Quantum Computing and Quantum Communication, Lecture Notes in Comput. Sci. 1509, Springer-Verlag, Berlin, 1999, pp. 174–188.
[33] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2000.
[34] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput., 26 (1997), pp. 1484–1509.
[35] D. R. Simon, *On the power of quantum computation*, SIAM J. Comput., 26 (1997), pp. 1474–1483.
[36] R. Tolimieri, M. An, and C. Lu, *Algorithms for Discrete Fourier Transform and Convolution*, Springer-Verlag, New York, 1989.
[37] J. Watrous, *Quantum algorithms for solvable groups*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, 2001, pp. 60–67.