Efficient quantum processing of ideals in finite rings

Pawel M. Wocjan[1], Stephen P. Jordan[2], Hamed Ahmadi[1,3], and Joseph P. Brennan[3,4]

[1]*School of Electrical Engineering and Computer Science, University of Central Florida, Orlando*
[2]*Institute for Quantum Information, Caltech, Pasadena*
[3]*Department of Mathematics, University of Central Florida, Orlando*
[4]*Nanoscience Technology Center, University of Central Florida, Orlando*

Suppose we are given black-box access to a finite ring R, and a list of generators for an ideal I in R. We show how to find an additive basis representation for I in poly$(\log |R|)$ time. This generalizes a recent quantum algorithm of Arvind *et al.* which finds a basis representation for R itself. We then show that our algorithm is a useful primitive allowing quantum computers to rapidly solve a wide variety of problems regarding finite rings. In particular we show how to test whether two ideals are identical, find their intersection, find their quotient, prove whether a given ring element belongs to a given ideal, prove whether a given element is a unit, and if so find its inverse, find the additive and multiplicative identities, compute the order of an ideal, solve linear equations over rings, decide whether an ideal is maximal, find annihilators, and test the injectivity and surjectivity of ring homomorphisms. These problems appear to be hard classically.

Here we present quantum algorithms for several problems regarding finite rings. All of the algorithms run in time scaling polylogarithmically in the size of the ring. A ring is normally specified by a set of elements that generate the ring via linear combination and multiplication, and an ideal is normally specified by a set of elements that generate the ideal via linear combination and multiplication by arbitrary ring elements. To apply the known quantum techniques for Abelian groups we find sets that generate rings and ideals *as Abelian groups*, that is, by linear combination only. The problem of finding such a generating set for rings has been already solved by Arvind *et al.*[2]. Our solution for ideals generalizes their result.

As shown in [6], both integer factorization and graph isomorphism reduce to the problem of counting automorphisms of rings. This counting problem is contained in AM∩coAM. Therefore it is unlikely to be NP-hard. Integer factorization also reduces to the problem of finding nontrivial automorphisms of rings and to the problem of finding isomorphisms between two rings. Furthermore, graph ismorphism reduces to ring isomorphism for commutative rings. Thus these ring automorphism and isomorphism problems are attractive targets for quantum computation. Perhaps the quantum algorithms given in this paper can serve as steps toward efficient quantum algorithms for some of these problems.

Let R be a finite ring with identity, which need not be commutative. Let $\tilde{R} = \{r_1, \ldots, r_n\}$ be a subset of R such that each element of R can be obtained by some sequence of additions and multiplications of elements of $\tilde{R}$. We say that $\tilde{R}$ is a generating set for R. Let I be the left ideal in R generated by $\tilde{I}$. That is, I is the smallest subset of R containing $\tilde{I}$ that is closed under addition and closed under left multiplication by elements of R. Throughout this paper we mainly discuss left ideals. One can similarly define right ideals and two-sided ideals, and the generalization of our algorithms to these cases is a straightforward generalization. Note that R is itself an ideal in R.

A left ideal I in a finite ring R forms an Abelian group $(I, +)$ under addition. Any generating set $\{a_1, \ldots, a_\ell\}$ for an Abelian group A yields a homomorphism from $\mathbb{Z}_{s_1} \times \ldots \times \mathbb{Z}_{s_\ell}$ to A where $s_1, \ldots, s_l$ are the orders of $a_1, \ldots, a_\ell$. In additive notation, this homomorphism takes the integers $z_1, \ldots, z_\ell$ to $\sum_{j=1}^{\ell} z_j a_j$. The structure theorem for finite Abelian groups states that there exists a generating set for A such that this homomorphism is an isomorphism. We call this a *generating set of the invariant factors*, or i. f. generating set for short. The main tool in this paper is an efficient quantum algorithm to find an i. f. generating set for $(I, +)$. No polynomial time classical algorithm for this problem is known.

The computational difficulty of problems on rings may depend on how the algorithm is allowed to access the ring. We assume only blackbox access to the ring. That is, the ring elements are assigned arbitrary bit strings by some injective map η and we have access to blackboxes implementing $f_+(\eta(a), \eta(b)) = \eta(a + b)$ and $f_\times(\eta(a), \eta(b)) = \eta(a \times b)$. The ideal I is specified by a list of generators $\tilde{I} = \{i_1, \ldots, i_m\}$ with $m = O(\log |R|)$. Given these inputs, our method for finding an i. f. generating set for $(I, +)$ proceeds in two steps. First we find a generating set for $(I, +)$. Although the elements of $\tilde{I}$ generate I as an ideal, they do not generate I as an Abelian group, that is, by addition only with no left-multiplication by R elements. After finding a generating set for $(I, +)$ we then convert it to an i. f. generating set for $(I, +)$ using the quantum algorithms of [3, 11].

To find a generating set for $(I, +)$, let $\tilde{B}_1 = \tilde{I}$ and apply the following iteration. Let B_k be the Abelian group additively generated by $\tilde{B}_k$. At the kth step we search for an element $i \in I$ not contained in B_k. If we find one, we let $\tilde{B}_{k+1} = \tilde{B}_k \cup \{i\}$. For some sufficiently large k, $B_k = I$, at which point the search for i fails and the process terminates. We now show in detail how this works and that we need at most $\log_2 |I|$ iterations.

Suppose we know $\tilde{B}_k$. To find an element of I not contained in B_k, we choose any generator $r \in \tilde{R}$ of R.

Let $rB_k = \{rx | x \in B_k\}$. We create the superpositions

$$|B_k\rangle = \frac{1}{\sqrt{|B_k|}} \sum_{x \in B_k} |x\rangle$$

and

$$|rB_k\rangle = \frac{1}{\sqrt{|B_k|}} \sum_{x \in B_k} |rx\rangle.$$

Because B_k and rB_k are Abelian groups whose generators we know, these states can be created efficiently to polynomial precision using the results of [3, 11].

To determine the intersection of B_k and rB_k we use the swap test to estimate the inner product $\langle B_k | rB_k \rangle$. Polynomially many applications of the swap test yield $\langle B_k | rB_k \rangle$ to 1/poly precision. $B_k \cap rB_k$ is a subgroup of B_k. Thus by Lagrange's theorem, either $\frac{|B_k \cap rB_k|}{|B_k|} = 1$ or $\frac{|B_k \cap rB_k|}{|B_k|} \leq \frac{1}{2}$. These two cases can be distinguished with high reliability by swap tests, because

$$\langle B_k | rB_k \rangle = \frac{|B_k \cap rB_k|}{|B_k|}.$$

If we find that $\frac{|B_k \cap rB_k|}{|B_k|} \leq \frac{1}{2}$ then we choose an element $i \in rB_k$ uniformly at random. We can do this using the techniques of [3, 11] to find an i. f. generating set for B_k and then sampling uniformly from the product of cyclic groups to which B_k is isomorphic. Thus, along with i we get an expression for i as r times some linear combination of the elements of $\tilde{B}_k$. i is definitely contained in I, and with probability at least $1/2$, i is not contained in B_k. If $i \in B_k$ then $\langle B_k | i + B_k \rangle = 1$, otherwise $\langle B_k | i + B_k \rangle = 0$. Thus, to determine whether $i \in B_k$ we create the states $|B_k\rangle$ and $|i + B_k\rangle$ and use the swap test. If $i \in B_k$ we choose a different random element of rB_k and try again. With probability $1 - \epsilon$, this process terminates in $O(\log(1/\epsilon))$ time. Once it does, we let $\tilde{B}_{k+1} = \tilde{B}_k \cup \{i\}$.

If we instead find that $\frac{|B_k \cap rB_k|}{|B_k|} = 1$, we choose a different $r \in \tilde{R}$ and swap test again. We keep repeating this process until we find some $r \in \tilde{R}$ such that $\frac{|B_k \cap rB_k|}{|B_k|} \neq 1$ or we exhaust $\tilde{R}$. If $\frac{|B_k \cap rB_k|}{|B_k|} = 1$ for all $r \in \tilde{R}$ we are done, because $B_k = I$. We can prove this with the following lemma.

Lemma 1 *Let I be a left ideal generated by $\{i_1, \ldots, i_m\}$ in a finite ring R. Let $\tilde{B}_k$ be a subset of I containing $\{i_1, \ldots, i_m\}$. The set of ring elements B_k additively generated by $\tilde{B}_k$ is equal to I if and only if $rB_k \subseteq B_k \ \forall r \in \tilde{R}$.*

Proof: If $rB_k \subseteq B_k$ for all $r \in \tilde{R}$ then, because $\tilde{R}$ is a generating set for R, $rB_k \subseteq B_k$ for all $r \in R$. Thus, B_k is a left ideal in R. By construction, B_k contains $i_1, \ldots, i_m$. By the definition of generators, I is the smallest left ideal in R containing $i_1, \ldots, i_m$. B_k is also contained in I. Thus $B_k = I$. The converse follows immediately from the fact that I is a left ideal. $\square$

In the above procedure, the time needed to obtain each additive generator is $\text{poly}(\log |R|)$. Furthermore, every time we add another generator, we increase the size of the generated group by at least a factor of two. Thus, we need to perform the above iteration at most $\log_2 |I|$ times. We can also in polynomial time obtain expressions for the elements of this set in terms of the original generators for I by recursively composing the expressions we obtained at each step for i in terms of the preceding generators B_k.

Once we have a set B_k of elements that generate I as an Abelian group, we can efficiently find an i. f. generating set for $(I, +)$, as well as expressions for the i. f. generators as linear combinations of B_k using the techniques of [3, 11]. These techniques also efficiently yield the additive orders of the i. f. generators.

After finding an i. f. generating set for $(I, +)$, one would like to have a procedure to take a given element $i \in I$ and decompose it as a linear combination of these generators. Note that i is given as an arbitrary bit string from the encoding η, so initially we know nothing about i. We can efficiently perform this decomposition as described below.

Let $G = \mathbb{Z}_{s_1} \times \mathbb{Z}_{s_2} \times \ldots \times \mathbb{Z}_{s_\ell} \times \mathbb{Z}_s$, where $s_1, \ldots, s_\ell$ are the orders of the i. f. generators $h_1, \ldots, h_\ell$ and s is the order of i. Let

$$f(n_1, n_2, \ldots, n_\ell, m) = \eta \left(\sum_{j=1}^{\ell} n_j h_j + mi \right).$$

This function hides the cyclic subgroup of G generated by

$$(n_1(i), n_2(i), \ldots, n_\ell(i), -1),$$

where $n_1(i), \ldots, n_\ell(i)$ is the decomposition of i in terms of the i. f. generators:

$$i = \sum_{j=1}^{\ell} n_j(i) h_j.$$

Using the polynomial time quantum algorithm for the Abelian hidden subgroup problem [9], we thus recover this decomposition.

Let $\{h_1, \ldots, h_\ell\}$ be an i. f. generating set for I. The multiplication in I can be fully specified by the tensor M_{ij}^k defined by

$$h_1 h_j = \sum_{k=1}^{\ell} M_{ij}^k h_k.$$

We can compute all l^3 of the entries of M_{ij}^k by taking each pair h_i, h_j, using the multiplication oracle to find the bit string encoding their product, and then using the Abelian hidden subgroup algorithm to decompose the element represented by the resulting bit string, as described above. Together, the i. f. generators for I, their

orders, and the multiplication tensor are called a basis representation for I. The previous work of Arvind *et al.* shows how to efficiently quantum compute a basis representation in the special case that I is the entire ring R [2]. The best existing classical algorithm for this problem requires order $|R|$ queries[12].

Given a basis representation for an ideal I it is straightforward to construct a uniform superposition $|I\rangle$ over all elements of I. By constructing the superpositions $|I\rangle$ and $|J\rangle$ for two ideals I and J we can determine whether $I = J$ using the swap test. By Lagrange's theorem, if $I \neq J$ then $\langle I|J\rangle \leq 1/2$. Thus we need only use $O(\log(\epsilon))$ swap tests to ensure that the chance of falsely concluding $I = J$ is at most ϵ. After constructing $|I\rangle$ and being given a ring element r, we can use the addition blackbox to construct the coset state $|r + I\rangle$. If $r \in I$ then the inner product of these states is one, and otherwise it is zero. Thus, the swap test on $|I\rangle$ and $|r + I\rangle$ tells us whether $r \in I$. Given $r \in R$, let Rr be the left ideal in R generated by r. $Rr = R$ if and only if r is a unit. If $Rr \neq R$ then Rr contains at most half the elements of R. Thus one can determine whether a given $r \in R$ is a unit by constructing $|Rr\rangle$ and $|R\rangle$ and comparing them using the swap test. If r is a unit, then we can find its inverse using the quantum order finding algorithm[10]. If $r^c = \mathbb{1}$ then $r^{-1} = r^{c-1}$.

Suppose r is contained in the ideal I. To obtain an explicit construction for r in terms of the generators of I, we can first obtain a basis representation for I. We can obtain an expression for r as a linear combination of the basis for I by solving the Abelian hidden subgroup problem. From the algorithm for obtaining a basis representation for I we also obtain expressions for the basis elements in terms of the original generators of I. Thus one can efficiently convert the expression for r as a linear combination of the basis representation for I into an expression for r in terms of the original generators for I.

Suppose we are given generating sets for two ideals I and J. We wish to find a basis for $I \cap J$. By techniques described above, we can create the superposition $|J\rangle$ over all elements of J, and we can find a basis representation for I. A reversible circuit for addition performs the unitary transformation $U_+|a\rangle|b\rangle = |a\rangle|a + b\rangle$. Thus, $U_+|a\rangle|J\rangle = U_+|a\rangle|a + J\rangle$, where $|a + J\rangle$ is a superposition over the coset $a + J$. If $a \in J$ then $\langle a + J|J\rangle = 1$. Otherwise $\langle a + J|J\rangle = 0$. Hence applying addition to the state J is an operation that "hides" the subgroup $(I \cap J, +)$ of the group $(I, +)$ of inputs. Thus, one can use the quantum algorithms for the Abelian hidden subgroup problem[9] to find a set of generators for $(I \cap J, +)$. From this we easily extract a basis representation. (Typically in a hidden subgroup problem one is given a blackbox that maps group elements to classical bit strings. This map is constant and distinct on cosets of the hidden subgroup. However, examining the algorithm of [9], one sees that it works just the same if the blackbox maps the different cosets to any set of orthogonal states, the classical bit string states being just a special case.)

If I and J are two ideals in R, one defines $(I : J) = \{x \in R | xJ \subseteq I\}$. $(I : J)$ is an ideal, and is called an ideal quotient or a colon ideal. $(I : J)$ is a subgroup of $(R, +)$. Let U be the unitary transformation defined by $U|x\rangle|y_1\rangle \ldots |y_m\rangle = |x\rangle|xj_1 + y_1\rangle \ldots |xj_m + y_m\rangle$ for all $x, y_1, \ldots, y_m \in R$. Given quantum black boxes for arithmetic on R, U can be efficiently implemented by a quantum circuit. The states $|xj_1 + I\rangle \ldots |xj_m + I\rangle$ and $|yj_1 + I\rangle \ldots |yj_m + I\rangle$ are identical if x and y belong to the same coset of $(I : J)$ in $(R, +)$ and are orthogonal if x and y come from different cosets. Thus, we can efficiently find an additive generating set for $(I : J)$ by solving the Abelian hidden subgroup problem using U to hide $(I : J)$.

The left annihilator A_S of $S = \{s_1, \ldots, s_n\} \subseteq R$ is defined as $A_S = \{x \in R | xs_1 = 0, \ldots, xs_n = 0\}$. A_S forms a subgroup of $(R, +)$. The function on R given by $f_S(x) = (xs_1, \ldots, xs_n)$ hides this subgroup. Thus, after finding an i. f. generating set for R one can use the quantum algorithm for the Abelian hidden subgroup problem to find generators for any annihilator provided S is at most polynomially large. The same method will work if S is given by a polynomially large set of additive generators.

Given generators for an ideal I in a finite ring, we can find the order of I, by finding an i. f. generating set for it and taking the product of the orders of the generators. Finding the order of a ring is a special case, as any ring is an ideal in itself.

Suppose we are given a black-box implementing a homomorphism $\rho : R \to R'$ between two rings. Determining whether ρ is injective is an Abelian hidden subgroup problem, where the kernel of ρ is the hidden subgroup in $(R, +)$. ρ is injective if and only if its kernel is $\{0\}$. We can efficiently find generators for the kernel of ρ by finding an i. f. generating set for R, and then solving the Abelian hidden subgroup problem. To determine whether ρ is surjective, we first compute the order of R'. Similarly, the image of ρ is a ring. If R is generated by $\{r_1, \ldots, r_n\}$ then R' is generated by $\{\rho(r_1), \ldots, \rho(r_n)\}$. After querying the homomorphism black-box to obtain the generators $\{\rho(r_1), \ldots, \rho(r_n)\}$ we can compute the order of the ring they generate (R') as described in the preceding paragraph. ρ is surjective if and only if the order of the image of ρ equals the order of R'.

Suppose we wish to solve a linear equation $ax = b$ over R. To do this we find an i. f. generating set $\{h_1, \ldots, h_\ell\}$ for R, and decompose a and b in terms of these generators

$$a = \sum_{i=1}^{\ell} a_i h_i \qquad b = \sum_{i=1}^{\ell} b_i h_i.$$

Let

$$A_{ij} = \sum_k a_k M_{kj}^i$$

where M_{kj}^i is the multiplication tensor from the basis representation. Parametrize x as $x = \sum_{i=1}^{\ell} x_i h_i$ for integers $x_1, \ldots, x_\ell$. Then, in an i. f. generating set, $ax = b$ if and

only if

$$\sum_{j=1}^{\ell} A_{ij} x_j \equiv b_i \quad \mathrm{mod}\ s_i, \qquad (1)$$

for each $i = 1, 2, \ldots, \ell$. (Here s_i is the additive order of h_i.) We can introduce additional integer unknowns $k_1, \ldots, k_\ell$ and rewrite this as a system of linear diophantine equations:

$$\sum_{j=1}^{\ell} A_{ij} x_j + k_i s_i = b_i, \quad i = 1, 2, \ldots, \ell. \qquad (2)$$

A solution to a system of m diophantine equations in n variables can be found in $\mathrm{poly}(n, m)$ time using the classical algorithms of [4]. Thus we can classically find an integer solution to equation 2, which has ℓ equations and 2ℓ unknowns, in $\mathrm{poly}(\ell)$ time. Equation 2 is undetermined because the original system of equations 1 is modular.

By a similar technique, we can find the identity in R. Again suppose we have computed a basis representation for R. Since the basis representation has the following property,

$$n_1 h_1 + \ldots + n_\ell h_\ell = h_\alpha \Rightarrow n_\beta = \delta_{\alpha\beta} \quad 1 \le \beta \le \ell$$

where $n_i \in \mathbb{Z}_{s_i}$, an element $r = \sum_{i=1}^{\ell} r_i h_i$ is the identity if and only if

$$\sum_{i=1}^{\ell} r_i M_{ij}^k \equiv \delta_{jk} \quad \mathrm{mod}\ s_k$$

for all $j, k = 1, 2, \ldots, \ell$. This is again a system of linear modular equations, which we can convert to a system of linear diophantine equations that we solve in polynomial time using[4]. Note that the quantum algorithm of [1] solves a very different problem although the authors refer to it as identity testing.

In a black box ring, finding the additive identity is also nontrivial. Because all ring elements have additive inverses, we can choose any $r \in R$, find its order c using the quantum order finding algorithm [10], find the additive inverse of r by computing $(c-1)r$, and find the additive identity by computing cr. The computation of cr and $(c-1)r$ requires $O(\log_2 c)$ queries to f_+.

We now show how to efficiently determine whether a given two-sided ideal I is prime. Recall that an ideal I is prime if $ab \in I$ implies that $a \in I$ or $b \in I$ for all $a, b \in R$, which is equivalent to the fact that the quotient ring $S = R/I$ does not have any zero-divisors. This already implies that S is a division ring (i.e., each non-zero element has a multiplicative inverse) since S is finite. Wedderburn's theorem shows that all finite division rings are finite fields [7]. R/I a field implies I is maximal, thus I is prime implies I is maximal. The converse is also true.

Let S^* denote the group of units of the quotient ring S. We choose an element r uniformly at random in R. With probability at least $1/2$ we have $r \notin I$. Once we obtain such r we determine the size of the (additively generated) cyclic subgroup $\langle \bar{r} \rangle$ of S, where $\bar{r}$ denotes the image of r in S under the canonical projection. This can be done by applying Shor's period finding algorithm to the state $(1/\sqrt{q}) \sum_{x=0}^{q} |x\rangle |xr + I\rangle)$ where q is a power of 2 with $|S|^2 < q \le 2|S|^2$. This state can be prepared efficiently.

If S is a field, then with probability at least $\varphi(|S| - 1)/|S| \ge \Omega(1/\log|S|)$ we have $\langle \bar{r} \rangle = S^*$ where φ denotes Euler's totient function. This follows from the fact that the group of units $\mathbb{F}_d^*$ of an arbitrary finite field $\mathbb{F}_d$ with d element is cyclic of order $d-1$ and $\varphi(m)/m = \Omega(1/\log m)$ for integers m [5]. If S is not a field, then S^* cannot have order $|S| - 1$ (otherwise every non-zero element would have a multiplicative inverse, implying that S is a field). If we find that S is a field then we know I is prime, otherwise I is not prime. The above procedure for determining whether the quotient ring S is a field can be applied to any finite blackbox ring, offering a simpler alternative to the algorithm in [2].

Our quantum algorithms for rings R also extend to R-modules. Beyond this, we conjecture that our quantum algorithms apply to any category posessing a faithful functor to the category of Abelian groups.

It would be interesting to find efficient quantum algorithms for deciding whether a given ideal I is principal and computing the group of units R^* of R. The quantum algorithms in [3, 11] make it possible to determine the structure of any finite abelian black-box group according to the structure theorem. So, the question arises naturally whether a similar quantum algorithm exists for decomposing finite black-box rings. More precisely, is it possible to efficiently learn the structure of a finite black-box ring according to a structure theorem in ring theory such as the Wedderburn-Artin theorem [8]?

It would be worthwhile to investigate whether the above algorithms extend to the case of infinite rings. It is not obvious that we can consider arbitrary infinite rings. However, it seems likely that the above algorithms could be extended to a black-box ring R which is endowed with a grading by Abelian groups $R_0, R_1, R_2, \ldots$ and each component R_g is finite. Additionally, we would need a promise, making it possible to do all the computations in a component R_g for some g. For example, such a situation occurs for polynomial rings over a finite field when the number of indeterminates is fixed. The complexity of the algorithms would then depend on the growth of the Hilbert function, which measures the dimension of the graded components R_g as R_0-modules.

[1] V. Arvin and Partha Mukhopadhyay. Quantum query complexity of multilinear identity testing. In *Symposium on Theoretical Aspects of Computer Science*, pages 87–98, Freiburg, 2009.

[2] V. Arvind, Bireswar Das, and Partha Mukhopadhyay. The complexity of black-box ring problems. In D.Z. Chen and D.T. Lee, editors, *Proceedings of COCOON 2006*, volume 4112 of *Lecture Notes in Computer Science*, pages 126–145. Springer-Verlag, 2006.

[3] Kevin K. H. Cheung and Michele Mosca. Decomposing finite Abelian groups. *Quantum Information and Computation*, 1(2):26–32, 2001. arXiv:cs/0101004.

[4] Tsu-Wu Chou and George E. Collins. Algorithms for the solution of systems of linear diophantine equations. *SIAM Journal on Computing*, 11(4):687–708, 1982.

[5] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Oxford, 1979.

[6] Neeraj Kayal and Nitin Saxena. On the ring isomorphism and automorphism problems. In *Proceedings of the Twentieth Annual IEEE Conference on Computational Complexity*, pages 2–12, 2005.

[7] Rudolf Lidl and Harald Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, UK, 1997.

[8] Bernard R. McDonald. *Finite Rings with Identity*. Marcel Dekker Inc., 1974.

[9] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.

[10] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 2005. arXiv:quant-ph/9508027.

[11] John Watrous. Quantum algorithms for solvable groups. In *Proceedings of the 33rd annual ACM Symposium on the Theory of Computing*, pages 60–67, 2001. arXiv:quant-ph/0011023.

[12] J. Zumbragel, G. Maze, and J. Rosenthal. Efficient recovering of operation tables of black box groups and rings. In *IEEE International Symposium on Information Theory*, pages 639–643, 2009.