Symmetric functions of qubits in an unknown basis

Ashley Montanaro
*Department of Computer Science, University of Bristol, Woodland Road, Bristol, BS8 1UB, U.K.**
(Dated: June 18, 2009)

Consider an n qubit computational basis state corresponding to a bit string x, which has had an unknown local unitary applied to each qubit, and whose qubits have been reordered by an unknown permutation. We show that, given such a state with Hamming weight $|x| \leq \lfloor n/2 \rfloor$, it is possible to reconstruct $|x|$ with success probability $1 - |x|/(n - |x| + 1)$, and thus to compute any symmetric function of x. We give explicit algorithms for computing whether or not $|x| \geq t$ for some t, and for computing the parity of x, and show that these are essentially optimal. These results can be seen as generalisations of the swap test for comparing quantum states.

I. INTRODUCTION

Consider the following scenario. Alice is a physicist who has just completed a long quantum computing experiment. Her quantum computer has produced an n qubit state $|x\rangle$, corresponding to the bit string x. However, before she can measure the state to determine x, she is called away from the lab. In her absence, Eve sneaks in and sabotages the experiment. First, she applies an arbitrary local rotation to the qubits (the same rotation on each qubit); she then rearranges all the qubits in an arbitrary order.

It is clearly now hopeless for Alice to determine x exactly. Indeed, she cannot even determine an individual bit of x with any probability better than guessing. But what if she only needs to calculate $f(x)$, for some function f? Because of the arbitrary rearrangement of the qubits, she only has a chance of being able to compute symmetric functions, i.e. functions f where $f(x)$ depends only on $|x|$, the Hamming weight of x. Also, because of the arbitrary local rotation, she can only compute functions f where $f(x) = f(\bar{x})$, with $\bar{x}$ denoting bitwise negation. This is equivalent to imposing the constraint that $|x| \leq \lfloor n/2 \rfloor$.

The purpose of this note is to show that Alice can in fact compute *any* f that satisfies these constraints (with some probability, which may be low in the worst case). Indeed, we have the following result.

Theorem 1. *Let x be an n-bit string with $|x| \leq \lfloor n/2 \rfloor$. Let U be an unknown and arbitrary single qubit unitary operator, and σ be an unknown and arbitrary permutation of n qubits. Then there is a procedure which, given $\sigma(U^{\otimes n}|x\rangle)$, outputs $|x|$ correctly with probability*

$$1 - \frac{|x|}{n - |x| + 1}.$$

Assuming that $|x|$ is distributed uniformly at random, for large n this corresponds to an average probability of success of approximately $2(1 - \ln 2) \approx 0.614$.

*Electronic address: montanar@cs.bris.ac.uk

Previous work has studied the closely related question of communicating without a shared reference frame [4]. In this setting, Alice wishes to communicate some (classical or quantum) information to Bob by sending him n qubits, but Bob does not know Alice's basis for each of the qubits. The results of [4] show that, by encoding across multiple qubits, Alice can send Bob a number of bits that approaches n, in the large n limit. By contrast, in the present work we do not allow prior encoding of Alice's information. Also note related previous work on the problem of computation in a hidden basis [12].

Theorem 1 can be used to obtain procedures for computing any symmetric function of x. These results can be seen as generalisations of the problem of determining whether n qubits are all in the same state [13, 14], which is in turn a generalisation of the question of determining equality of two qubits, which can be solved using the well-known *swap test* [3, 7]. For example, we have the following results for the threshold function TH_t ($\mathrm{TH}_t(x) = 1 \Leftrightarrow |x| \geq t$) and the parity function ($\textsc{Parity}(x) = \bigoplus_i x_i$).

Corollary 2. *Let x be an n-bit string with $|x| \leq \lfloor n/2 \rfloor$. Let U be an unknown and arbitrary single qubit unitary operator, and σ be an unknown and arbitrary permutation of n qubits. Then there is a procedure which, given $\sigma(U^{\otimes n}|x\rangle)$, can compute $\mathrm{TH}_t(x)$ with success probability at least $1 - t/(n + 1)$, and can compute $\textsc{Parity}(x)$ with probability at least $1/2 + 1/(2(n + 1))$.*

These success probabilities are essentially optimal, as we will show with the following theorem.

Theorem 3. *Let x be an n-bit string with $|x| \leq \lfloor n/2 \rfloor$. Let U be an unknown and arbitrary single qubit unitary operator, and σ be an unknown and arbitrary permutation of n qubits. Let $f(x)$ be some function such that $f(k + 1) \neq f(k)$ for some $0 \leq k < \lfloor n/2 \rfloor$. Then any procedure that computes $f(x)$ given access to $\sigma(U^{\otimes n}|x\rangle)$ succeeds with probability at most $1 - (k + 1)/(2(n - k))$ in the worst case.*

This implies that, for example, one can determine whether or not $|x| = 0$ very effectively, but distinguishing $|x| = n/2$ from $|x| = n/2 - 1$ is hard. Interestingly, this phenomenon also occurs in the study of quantum

and classical query complexity [6], and more generally in approximation theory [16, Section 3.4].

II. THE SYMMETRIC GROUP AND WEAK SCHUR SAMPLING

The results in this note will be proven using some basic representation theory of the symmetric group S_n, which we now outline. Conjugacy classes of S_n are labelled by partitions $\lambda \vdash n$. Each partition λ containing k parts can be written as a non-increasing sequence of positive integers $(\lambda_1, \ldots, \lambda_k)$, and expressed as a *Young diagram*. The diagram corresponding to the partition λ is a collection of boxes arranged in left-justified rows, where the number of boxes in row i is given by λ_i. Irreducible representations (irreps) of the symmetric group are thus in one-to-one correspondence with Young diagrams. We let λ denote both a partition and its corresponding diagram, and V_λ denote the corresponding irrep.

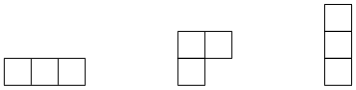

FIG. 1: The irreducible representations (3), (2,1), (1,1,1) of the group S_3.

We think of the input to our problem as an n-qubit state $|x\rangle$ in a known basis, to which an unknown, and arbitrary, tensor product unitary $U^{\otimes n}$ has been applied, followed by an unknown permutation of the qubits σ. In order to take advantage of these symmetries, we will use *Schur-Weyl duality*. This states that the space of n qudits decomposes into a direct sum of tensor products of subspaces corresponding to irreps of the symmetric and unitary groups, as follows:

$$(\mathbb{C}^d)^{\otimes n} \cong \bigoplus_{\lambda \vdash n} \mathcal{P}_\lambda \otimes \mathcal{Q}_\lambda^d, \tag{1}$$

where $\mathcal{P}_\lambda$ and $\mathcal{Q}_\lambda^d$ correspond to irreps of S_n and U_d, respectively. For good introductions to Schur-Weyl duality in the context of quantum information theory, see the theses [11] and [9].

The Schur transform [1, 2, 11] performs an implementation of this decomposition, mapping a state in the computational basis to one of the form $|\lambda\rangle|p\rangle|q\rangle$. In this case, as we are indifferent to permutations of the subsystems and local unitaries on each subsystem, we will only measure the $|\lambda\rangle$ register. This is known as *weak Schur sampling* [8]. The projector onto a given value of λ is given by (see [8] or [17, Theorem 8])

$$P_\lambda = \frac{d_\lambda}{n!} \sum_{\pi \in S_n} \chi_\lambda(\pi) D(\pi), \tag{2}$$

where d_λ is the dimension of the irrep V_λ, χ_λ is the character $\operatorname{tr} V_\lambda$, and D is the defining representation of S_n that acts by permuting the n subsystems,

$$D(\pi)|i_1\rangle \cdots |i_n\rangle = |i_{\pi^{-1}(1)}\rangle \cdots |i_{\pi^{-1}(n)}\rangle.$$

It is unnecessary to perform the full Schur transform to measure λ; it suffices to use the quantum Fourier transform over the symmetric group S_n, in a procedure known as generalised phase estimation [8, 11], which can be seen as a generalisation of the swap test [3, 7]. To perform generalised phase estimation on an n qubit state ρ, one first prepares an ancilla register in the state $\frac{1}{n!}\sum_{\pi \in S_n} |\pi\rangle$. This register is used to control a conditional permutation of the subsystems of ρ, which is followed by an inverse quantum Fourier transform (over S_n) on the ancilla register. Measuring the ancilla gives a value of λ with probability $\operatorname{tr}(P_\lambda \rho)$. This whole procedure can be performed efficiently, i.e. in time polynomial in n [5].

Let $|x\rangle$ be the input state and assume that x has Hamming weight $k \leq \lfloor n/2 \rfloor$. Letting σ be an arbitrary permutation of n qubits and U be an arbitrary local unitary, we now compute $\operatorname{tr}(P_\lambda\, D(\sigma) U^{\otimes n} |x\rangle\langle x| (U^\dagger)^{\otimes n} D(\sigma)^\dagger)$. As P_λ commutes with local unitaries and permutations [8], this is equal to $\operatorname{tr}(P_\lambda |x\rangle\langle x|)$. Invariance under permutation also implies that the probability of obtaining a given outcome λ depends only on the Hamming weight of x. The following crucial lemma allows us to write down exactly what these probabilities are.

Lemma 4. *Let $\Pr[\ell|k]$ denote the probability of getting the measurement outcome corresponding to the partition $(n-\ell, \ell)$ when performing weak Schur sampling on an n-bit string with Hamming weight k. Then, if $\ell > k$, $\Pr[\ell|k] = 0$. Otherwise,*

$$\Pr[\ell|k] = \frac{\binom{n}{\ell} - \binom{n}{\ell-1}}{\binom{n}{k}}.$$

In particular, $\sum_{\ell=0}^{k} \Pr[\ell|k] = 1$.

Before we prove this lemma, we show that it implies the results stated in Section I. In the case of Theorem 1, we give an explicit algorithm that achieves the success probability required by the theorem:

1. Perform weak Schur sampling, obtaining outcome $\lambda = (n-\ell, \ell)$.
2. Output the guess that $k = \ell$.

It is clear that this procedure, which we will term the *standard algorithm*, will output the correct answer with probability $\Pr[k|k] = 1 - k/(n-k+1)$.

One might consider more complicated strategies for inferring k from weak Schur sampling. A general inference strategy can be expressed as a matrix O, where $O_{k\ell} = \Pr[\text{output } k | \text{get outcome } (n-\ell, \ell)]$, and $\sum_k O_{k\ell} = 1$ for all ℓ. If one wishes to maximise the worst-case probability of outputting the correct value of k, for example, it is required to find an O that maximises

$$\min_k \left(\sum_{\ell=0}^{k} O_{k\ell} \left(\frac{\binom{n}{\ell} - \binom{n}{\ell-1}}{\binom{n}{k}} \right) \right).$$

This is a linear programming problem and can be solved exactly for small n, although we do not know a closed form for the solution for general n.

An alternative setting for the inference problem is the Bayesian scenario where one maximises the probability of success assuming an a priori probability distribution on k (see [15] for a comprehensive introduction to Bayesian inference). Letting $\{p_k\}$ denote this probability distribution, the problem is to maximise

$$\begin{aligned} P_{succ} &= \sum_{k=0}^{\lfloor n/2 \rfloor} p_k \left(\sum_{\ell=0}^{k} O_{k\ell} \left(\frac{\binom{n}{\ell} - \binom{n}{\ell-1}}{\binom{n}{k}} \right) \right) \\ &= \sum_{\ell=0}^{\lfloor n/2 \rfloor} \left(\binom{n}{\ell} - \binom{n}{\ell-1} \right) \left(\sum_{k=\ell}^{\lfloor n/2 \rfloor} O_{k\ell} \frac{p_k}{\binom{n}{k}} \right). \end{aligned}$$

This is clearly maximised by taking $O_{k\ell} = 1$ for $k = \max_{k'} p_{k'}/\binom{n}{k'}$, and $O_{k\ell} = 0$ otherwise. In the particularly natural case where we assume that the a priori distribution on k is uniform, this maximisation in fact shows that the standard algorithm is optimal, and gives an average probability of success of

$$\frac{1}{\lfloor n/2 \rfloor + 1} \sum_{k=0}^{\lfloor n/2 \rfloor} 1 - k/(n-k+1).$$

For large n, this can be estimated as

$$\begin{aligned} & 1 - \frac{2}{n} \int_0^{n/2} k/(n-k+1)\, dk \\ &= 2 \left(1 - \frac{(n+1)}{n} \ln \left(\frac{n+1}{n/2+1} \right) \right) \\ &\approx 2(1 - \ln 2) \approx 0.614. \end{aligned}$$

What about the scenario of Corollary 2, where we only want to compute some function $f(k)$, rather than to output k? It is natural to try to produce an algorithm with high success probability for all $0 \leq k \leq \lfloor n/2 \rfloor$. Again, if one attempts to maximise this worst-case probability over all strategies that consist of performing weak Schur sampling and attempting to infer $f(k)$ from the result, one is led to a linear programming problem for which we do not know a closed form solution. A more straightforward approach is to guess k using the standard algorithm (call this guess $\widetilde{k}$), and then to output $f(\widetilde{k})$.

The probability that this gives the right answer can easily be calculated for threshold functions $\textsc{Th}_t$. Assuming $k \geq t$,

$$\begin{aligned} \Pr[f(\widetilde{k}) \neq f(k)] &= \sum_{\ell, f(\ell) \neq f(k)} \Pr[\ell|k] \\ &= \frac{1}{\binom{n}{k}} \sum_{\ell=0}^{t-1} \left(\binom{n}{\ell} - \binom{n}{\ell-1} \right) = \frac{\binom{n}{t-1}}{\binom{n}{k}}. \end{aligned}$$

This is clearly maximised by $k = t$, giving a failure probability of at most $t/(n-t+1)$. On the other hand, if $k < t$, note that this algorithm succeeds with certainty, as $\Pr[\ell|k] = 0$ for $\ell > k$. We thus have a probabilistic algorithm that computes the threshold function $\textsc{Th}_t$ with one-sided error. This can be modified to give an algorithm with small worst-case probability of error, as follows.

Consider any procedure that attempts to compute an arbitrary boolean function $f(k)$ from $f(\widetilde{k})$, where k is picked to minimise the probability that $f(k) = f(\widetilde{k})$. Such a procedure can be parametrised by two probabilities q_0, q_1, where q_0 is the probability that the procedure outputs 0, given that $f(\widetilde{k}) = 0$, and q_1 is the probability of outputting 0, given that $f(\widetilde{k}) = 1$. Let $p_i = \Pr[f(\widetilde{k}) = 0 | f(k) = i]$ for $i \in \{0, 1\}$, and assume that $p_0 \geq p_1$. Then the probability of success of such a procedure (in the worst case) is at least

$$\min\{q_0\, p_0 + q_1(1-p_0), (1-q_0)p_1 + (1-q_1)(1-p_1)\}.$$

We pick q_1 such that these two values are equal, which gives

$$q_1 = \frac{1 - q_0(p_0 + p_1)}{2 - p_0 - p_1}.$$

Our goal is to maximise the corresponding expression for the probability of success,

$$q_0\, p_0 + q_1(1-p_0) = \frac{1 + q_0(p_0 - p_1) - p_0}{2 - p_0 - p_1},$$

over q_0, while still obeying the constraints $0 \leq q_0, q_1 \leq 1$. This is straightforward and gives the answer

$$q_0 = \min\{\frac{1}{p_0 + p_1}, 1\}, q_1 = \max\{0, \frac{1 - p_0 - p_1}{2 - p_0 - p_1}\},$$

which corresponds to a maximum worst-case probability of success of $p_0/(p_0 + p_1)$ when $p_0 + p_1 \geq 1$, and $(1 - p_1)/(2 - p_0 - p_1)$ when $p_0 + p_1 \leq 1$.

Applying this result to the threshold function $\textsc{Th}_t$, where $p_0 = 1$ and $p_1 = t/(n-t+1)$, it can easily be seen that we obtain a two-sided error algorithm that always succeeds with probability at least $1 - t/(n+1)$, as stated in Corollary 2.

In the case of the Parity function, we can calculate the probability that $f(\widetilde{k}) \neq f(k)$ from

$$\begin{aligned} & \Pr[\widetilde{k} \text{ is even}] - \Pr[\widetilde{k} \text{ is odd}] \\ &= \frac{1}{\binom{n}{k}} \sum_{\ell=0}^{k} (-1)^\ell \left(\binom{n}{\ell} - \binom{n}{\ell-1} \right) \\ &= (-1)^k \left(1 - \frac{2k}{n} \right). \end{aligned}$$

It is then immediate that

$$\Pr[\widetilde{k} \text{ is even}] = \begin{cases} 1 - k/n & (k \text{ even}) \\ k/n & (k \text{ odd}), \end{cases}$$

and of course $\Pr[\widetilde{k} \text{ is odd}] = 1 - \Pr[\widetilde{k} \text{ is even}]$. Assume that n is even and $n/2$ is also even (the cases where n or $n/2$ is odd are analogous). This implies that, for k even, we have $\Pr[f(\widetilde{k}) = f(k)] \geq 1/2$, and for k odd, $\Pr[f(\widetilde{k}) = f(k)] \geq 1/2 + 1/n$. We can use the same technique as before to get an algorithm that succeeds with probability at least $1/2 + 1/(2(n+1))$ in the worst case.

III. LIMITS ON SUCCESS PROBABILITY

To prove Theorem 3, and hence to show that these algorithms are almost optimal, consider the restricted problem of distinguishing a bit-string x with weight k from one with weight $k+1$, as is required to compute a symmetric function f where $f(k) \neq f(k+1)$.

We first argue that any procedure for computing f might as well simply consist of weak Schur sampling and post-processing the results. Imagine that an adversary, as in the scenario of Section I, has performed a random permutation and a random local rotation of each qubit on the initial state $|x\rangle$. Then, from Alice's perspective, the resulting state looks like

$$\rho = \frac{1}{n!} \int_U dU \, U^{\otimes n} \sum_{\sigma \in S_n} D(\sigma)|x\rangle\langle x|D(\sigma)^\dagger (U^\dagger)^{\otimes n},$$

which is equal to $\sum_{\lambda \vdash n} k_\lambda P_\lambda$ for some coefficients $\{k_\lambda\}$. (This follows from the decomposition of eqn. (1) and Schur's Lemma, using the fact that ρ commutes with all permutations and local unitaries.) This implies that, without loss of generality, a measurement strategy can be taken as consisting of measuring λ and performing some classical post-processing.

Let $p_k(\ell)$ denote the probability distribution over partitions $(n-\ell, \ell)$ obtained by performing weak Schur sampling on an input with weight k. We calculate the ℓ_1 distance between the distributions p_k, p_{k+1} for arbitrary $0 \leq k < \lfloor n/2 \rfloor$:

$$\begin{aligned} \|p_k - p_{k+1}\|_1 &= \frac{\binom{n}{k+1} - \binom{n}{k}}{\binom{n}{k+1}} \\ &\quad + \sum_{\ell=0}^{k} \left(\binom{n}{\ell} - \binom{n}{\ell-1} \right) \left(\frac{1}{\binom{n}{k}} - \frac{1}{\binom{n}{k+1}} \right) \\ &= 2 \left(1 - \frac{\binom{n}{k}}{\binom{n}{k+1}} \right) = 2 \left(\frac{n-2k-1}{n-k} \right). \end{aligned}$$

Using standard results on distinguishing probability distributions, this distance puts an upper bound on the probability of success of any algorithm attempting to distinguish between weights k and $k+1$, and implies that the above algorithms are asymptotically optimal. We finally turn to the proof of Lemma 4.

IV. PROOF OF LEMMA 4

Let λ be the partition $(n-\ell, \ell)$ and let x be a bit-string with Hamming weight $k \leq \lfloor n/2 \rfloor$, assuming without loss of generality that $|x\rangle = |1 \cdots 1 0 \cdots 0\rangle$, where the first k bits of x are 1 and the last $n-k$ are 0. We now calculate $\Pr[\ell|k] = \mathrm{tr}(P_\lambda |x\rangle\langle x|)$ using (2). It is easy to see that $\mathrm{tr}(D(\pi)|x\rangle\langle x|) = 0$ unless π leaves the bit-string x unchanged, in which case $\mathrm{tr}(D(\pi)|x\rangle\langle x|) = 1$. All such permutations π can be decomposed as a direct product of a permutation of the first k bits, and a permutation of the last $n-k$ bits. This implies that

$$\mathrm{tr}(P_\lambda |x\rangle\langle x|) = \frac{d_\lambda}{n!} \, \mathrm{tr} \left(\sum_{\pi \in S_k \times S_{n-k}} V_\lambda(\pi) \right).$$

We first calculate the sum over the group $S_k \times S_{n-k}$. Note that the representation V_λ, while irreducible over S_n, is not necessarily irreducible over $S_k \times S_{n-k}$, but may split into a direct sum of irreps. The following simple lemma, which can be proven using Schur's Lemma, shows that only the trivial irrep is of interest.

Lemma 5. *Let V_λ be an irreducible representation of a finite group G. Then*

$$\sum_{g \in G} V_\lambda(g) = \begin{cases} |G| & \text{if } V_\lambda \text{ is the trivial irrep} \\ 0 & \text{otherwise.} \end{cases}$$

Each occurence of the trivial irrep in the decomposition of the representation V_λ over $S_k \times S_{n-k}$ will thus give a contribution of $k!(n-k)!$ to the sum; all other irreps will contribute nothing. This number of occurrences can be calculated using a special case of the *Littlewood-Richardson rule* known as Pieri's formula [10].

Let μ be the diagram corresponding to the trivial irrep of S_k, for some k. Then, for any λ and ν, we define the Littlewood-Richardson number $N_{\lambda\mu\nu}$ as the number of ways that λ can be expanded to ν by adding k boxes to λ, under the constraint that at most one new box is added to each column. See Figure 2 for an illustration of this process, and note that $N_{\lambda\mu\nu}$ is always either 0 or 1 (though this does not remain true in the more general setting where μ can be arbitrary; then the rule is more complicated).

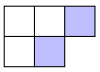

FIG. 2: Expanding the diagram $(2,1)$ to $(3,2)$ using the diagram (2).

Littlewood-Richardson numbers are relevant because of the following theorem [10].

Theorem 6. *Let μ be the partition $(n-k)$. The multiplicity of the irrep $V_\lambda \otimes V_\mu$ in the restriction of the irrep V_ν from S_n to $S_k \times S_{n-k}$ is equal to $N_{\lambda\mu\nu}$.*

As we are only interested in expansions of the trivial irrep (k) by another trivial irrep $(n-k)$, this multiplicity is particularly simple to calculate. Let ν be a partition of n, λ be the partition (k), and μ be the partition $(n-k)$. Then, if ν has more than two parts, $N_{\lambda\mu\nu} = 0$. (This was expected anyway because each of the n subsystems we are dealing with has dimension 2.) If ν has two parts, express it as $(n-\ell, \ell)$. Then, if $k < \ell$, $N_{\lambda\mu\nu} = 0$. Otherwise, $N_{\lambda\mu\nu} = 1$.

This deals with the sum; to finish the calculation, we need to find the dimension d_λ. This can be evaluated using the famous *hook-length formula* [10]. Let x be a box in a Young diagram. Then the hook-length $h(x)$ is defined as the total number of boxes in the same row and to the right of x, plus the total number in the same column and below x, plus 1 (for x itself). See Figure 3 for an illustration.

5	4	2	1
2	1		

FIG. 3: Hook-lengths of the cells in the diagram (4,2).

The hook-length formula states that

$$d_\lambda = \frac{n!}{\prod_{x\in\lambda} h(x)}.$$

As we only need calculate d_λ for partitions $\lambda = (n-\ell, \ell)$, this formula is particularly simple, and gives

$$d_\lambda = \frac{n!(n-2\ell+1)}{\ell!(n-\ell+1)!} = \binom{n}{\ell}\left(\frac{n-2\ell+1}{n-\ell+1}\right).$$

To sum up, we have, for $\ell \le k$,

$$\begin{aligned}\Pr[\ell|k] &= \frac{d_\lambda}{n!}\,\mathrm{tr}\left(\sum_{\pi\in S_k\times S_{n-k}} V_\lambda(\pi)\right)\\ &= \frac{1}{n!}\binom{n}{\ell}\left(\frac{n-2\ell+1}{n-\ell+1}\right)k!(n-k)!\\ &= \frac{\binom{n}{\ell}}{\binom{n}{k}}\left(\frac{n-2\ell+1}{n-\ell+1}\right) = \frac{\binom{n}{\ell}-\binom{n}{\ell-1}}{\binom{n}{k}},\end{aligned}$$

where the last step is a binomial coefficient identity that can be verified directly. For $\ell > k$, by Pieri's formula the sum over $S_k \times S_{n-k}$ is zero. This implies that $\Pr[\ell|k] = 0$ in this case, and completes the proof.

V. CONCLUSION

We conclude that it is possible to compute symmetric functions of an n qubit state $|x\rangle$, even if a malicious adversary has applied an arbitrary local rotation and an arbitrary permutation to the state, even without any prior encoding of x. This is in (perhaps surprising) contrast to the fact that any individual bit of x cannot be retrieved.

Acknowledgements

This work was supported by the EC-FP6-STREP network QICS, and was partly carried out during a visit to the Perimeter Institute for Theoretical Physics. I would like to thank Aram Harrow, Richard Low and Tobias Osborne for helpful discussions on the subject of this work, and an anonymous referee for helpful comments which improved the paper.

[1] D. Bacon, I. Chuang, and A. Harrow. Efficient quantum circuits for Schur and Clebsch-Gordan transforms. *Phys. Rev. Lett.*, 97(17):170502, 2006. `quant-ph/0407082`.

[2] D. Bacon, I. Chuang, and A. Harrow. The quantum Schur transform: I. Efficient qudit circuits. In *Proc. 18th ACM-SIAM Symposium on Discrete Algorithms*, 2007. `quant-ph/0601001`.

[3] A. Barenco, A. Berthiaume, D. Deutsch, A. Ekert, R. Jozsa, and C. Macchiavello. Stabilisation of quantum computations by symmetrisation. *SIAM J. Comput.*, 26(5):1541–1557, 1997. `quant-ph/9604028`.

[4] S. Bartlett, T. Rudolph, and R. Spekkens. Classical and quantum communication without a shared reference frame. *Phys. Rev. Lett.*, 91(2):027901, 2003. `quant-ph/0302111`.

[5] R. Beals. Quantum computation of Fourier transforms over symmetric groups. In *Proc. 29th Annual ACM Symp. Theory of Computing*, pages 48–53, 1997.

[6] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. `quant-ph/9802049`.

[7] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Phys. Rev. Lett.*, 87(16):167902, 2001. `quant-ph/0102001`.

[8] A. Childs, A. Harrow, and P. Wocjan. Weak Fourier-Schur sampling, the hidden subgroup problem, and the quantum collision problem. In *Proc. 24th Symposium on Theoretical Aspects of Computer Science (STACS 2007)*, pages 598–609, 2007. `quant-ph/0609110`.

[9] M. Christandl. *The Structure of Bipartite Quantum States – Insights from Group Theory and Cryptography.* PhD thesis, University of Cambridge, 2006. `quant-ph/0604183`.

[10] W. Fulton and J. Harris. *Representation theory: a first course.* Springer, 1991.

[11] A. Harrow. *Applications of coherent classical communication and the Schur transform to quantum information theory.* PhD thesis, Massachusetts Institute of Technology, 2005. `quant-ph/0512255`.

[12] L. Ioannou and M. Mosca. Universal quantum computation in a hidden basis, 2008. `arXiv:0810.2780`.

[13] I. Jex, E. Andersson, and A. Chefles. Comparing the

states of many quantum systems. *Journal of Modern Optics*, 51(4):505–523, 2004. quant-ph/0305102.

[14] M. Kada, H. Nishimura, and T. Yamakami. The efficiency of quantum identity testing of multiple states. *J. Phys. A: Math. Gen.*, 41:395309, 2008. arXiv:0809.2037.

[15] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[16] P. P. Petrushev and V. A. Popov. *Rational approximation of real functions*. Cambridge University Press, 1987.

[17] J.-P. Serre. *Linear representations of finite groups*. Springer-Verlag, 1977.