Fingerprints from Quantum Mechanics

Ronald de Wolf
rdewolf@cs.berkeley.edu

1 Introduction

Two dead bodies have been found on the same night in different parts of town, strangled to death. The police are anxious to know whether the killer in both cases was the same person. Fortunately, the murderers were stupid enough to leave their fingerprints on the necks of the victims, so the police only need to compare fingerprints from both necks to see whether they match or not. This will tell them whether both murderers are actually the same person.

This is an interesting phenomenon. The reason it works is that every human being has his own unique fingerprint. Such fingerprints do not give much information about their respective owners (assuming we do not have a complete table matching all possible fingerprints with all possible humans), but they *do* allow us to test for identity: if we want to know whether John and Jack are the same person, it suffices to compare their fingerprints — no need for John or Jack to be present themselves! However, human fingerprints are clearly restricted to human beings, and most other objects (houses, computers, cans of tomatoes) do not have fingerprints in a similar way. Here we will describe a general method that allows us to take short fingerprints of anything that can be described in bits. That is, we will associate with each n-bit string an exponentially smaller fingerprint, such that identity between two strings can be detected by comparing their fingerprints. The caveat is that our fingerprints will need to be *quantum mechanical*: they will be *superpositions* of classical states. This quantum fingerprinting method allows us to do certain things that are provably impossible in the world of classical physics and classical computing.

This article describes joint work with Harry Buhrman (CWI and UvA), Richard Cleve (Calgary), and John Watrous (Calgary), published recently in [3]. Before describing our quantum fingerprinting scheme, we will first give a brief introduction to quantum states and their use in computation.

2 Quantum computing

2.1 States and operations

In a classical computer, the unit of information is a *bit*, which can take on the values 0 or 1. In a quantum computer, the unit is a *quantum* bit, which is a linear combination of those two values. That is, a qubit is a superposition of the two "basis states" $|0\rangle$ and $|1\rangle$:

$$\alpha_0|0\rangle + \alpha_1|1\rangle,$$

where complex number α_0 is called the *amplitude* of the basis state $|0\rangle$, and α_1 is the amplitude of $|1\rangle$. We require $|\alpha_0|^2 + |\alpha_1|^2 = 1$. Viewing $|0\rangle$ and $|1\rangle$ as the vectors $\left(\begin{array}{c} 1 \\ 0 \end{array}\right)$ and $\left(\begin{array}{c} 0 \\ 1 \end{array}\right)$ respectively, the qubit corresponds to $\left(\begin{array}{c} \alpha_0 \\ \alpha_1 \end{array}\right)$. In a way, such a qubit is in both classical states simultaneously.

More generally, the state $|\phi\rangle$ of an m-qubit quantum computer can be described by a superposition of all 2^m classical m-bit states:

$$|\phi\rangle = \sum_{i\in\{0,1\}^m} \alpha_i|i\rangle,$$

with the condition that the squared amplitudes sum to 1: $\sum_i |\alpha_i|^2 = 1$. We can also view this state as the 2^m-dimensional complex unit vector that has the α_i as amplitudes.

There are basically two ways in which a quantum computer can manipulate such a state: it can make a measurement or apply a unitary transformation. Suppose we measure state $|\phi\rangle$. We cannot "see" a superposition itself, but only classical states. Accordingly, if we measure $|\phi\rangle$ we will see one and only one classical m-bit state $|i\rangle$. Which specific $|i\rangle$ will we see? This is not determined in advance; the only thing we can say is that we will see state $|i\rangle$ with probability $|\alpha_i|^2$. Because $|\phi\rangle$ is a unit vector, these probabilities nicely sum to 1. If we measure $|\phi\rangle$ and see classical state $|i\rangle$ as a result, then $|\phi\rangle$ itself has "disappeared", and all that is left is $|i\rangle$. In other words, observing $|\phi\rangle$ "collapses" the quantum superposition $|\phi\rangle$ to the classical state $|i\rangle$ that we saw, and all information that might have been contained in the other amplitudes is gone.

Instead of measuring $|\phi\rangle$, we can also apply some operation to it, i.e., change the state to some

$$|\psi\rangle = \sum_{i\in\{0,1\}^m} \beta_i|i\rangle.$$

Quantum mechanics only allows *linear* operations to be applied to quantum states, so the operation must correspond to multiplying the vector $|\phi\rangle$ with some matrix U:

$$U\left(\begin{array}{c} \alpha_0 \\ \vdots \\ \alpha_{2^m-1} \end{array}\right) = \left(\begin{array}{c} \beta_0 \\ \vdots \\ \beta_{2^m-1} \end{array}\right).$$

Because $|\psi\rangle$ should also be a unit vector, we have the constraint that U preserves norm, and hence is unitary (that is, U^{-1} equals the conjugate transpose U^*). Just like Boolean circuits, a well-chosen unitary matrix U followed by an appropriate measurement can compute any computable function. Every U can be built up from a small number of "elementary gates". These are unitary transformations that each act on only one or two qubits, just as classical

Boolean AND, OR, and NOT gates act on only one or two bits. A quantum computation is deemed efficient to the extent that U can be built up from a small number of such elementary gates.

As a simple example, consider the 1-qubit *Hadamard* gate, specified by the following unitary matrix:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

If we apply this to the classical state $|0\rangle$, we obtain the equal superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. If we measure this, we will see either $|0\rangle$ or $|1\rangle$, each with probability 50%. If we apply H to $|1\rangle$, then we obtain $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, which induces the same probability distribution when measured. However, if we apply H to a superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, then we get the classical state $|0\rangle$ back, because the positive and negative contributions to the amplitude of $|1\rangle$ add up to 0. This effect is known as *interference.*

2.2 What is it good for?

Why should we consider quantum computing? On a fundamental level, the answer is that computers are physical systems and physical systems are quantum mechanical. Accordingly, if we want to study the ultimate power and limits of computers, we should consider the full power and limits of quantum mechanics.

On a moderately more practical level, the main reason to consider quantum computers is that they can solve certain computational problems much faster than classical computers. For most computational problems, a quantum computer is not significantly more efficient than a classical computer (most problems are hard by any standard — classical as well as quantum), but for some it is.

The most important example of this is the problem of finding prime factors of large numbers. Peter Shor's quantum algorithm from 1994 [8] finds a factor of an n-bit number in roughly n^2 steps (elementary gates). In contrast, the best classical algorithms that we know, need about $2^{n^{1/3}}$ steps to find a factor. Even with massive parallelism, todays computers need several months to factor 512-bit numbers — and rightly so, because much of modern cryptography would become completely insecure if computers could quickly factor numbers of 512 or 1024 bits. In principle quantum computers could do this, but practice lags far behind theory in this young field. The largest number factored by a quantum computer to date is 15(=3*5), on a 7-qubit quantum computer [9].

A second example where quantum computers are much faster than classical ones is the problem of searching an unordered set of N elements for some target element. For example, searching for the person with phone number 5260248 in a phone directory that is ordered by name but not by phone number. Grover's quantum search algorithm from 1996 [5] finds the target element in about $\sqrt{N}$ steps, while a classical algorithm can do no better than just go through all records sequentially, which takes N steps. For example, Grover's algorithm can find a satisfying assignment for an n-bit Boolean formula in roughly $\sqrt{2^n}$ steps, while classical exhaustive search would have to go over all 2^n possible truth assignments separately.

3 How to construct short quantum fingerprints and test them

We will now use quantum states to construct a fingerprinting scheme. Recall the main idea behind fingerprinting: we want to map large objects (n-bit strings) to short objects (their fingerprints), such that we can decide whether two such large objects are equal by comparing only their fingerprints. A good quantum fingerprinting scheme thus requires two things: (1) a mapping from n-bit strings x to their short quantum fingerprints $|\phi_x\rangle$ and (2) a test to decide whether $x = y$, given only fingerprints $|\phi_x\rangle$ and $|\phi_y\rangle$.

It is not hard to show that non-orthogonal states (= states with non-zero inner product) cannot be distinguished with probability 1. Thus, if we want our test to work perfectly, the fingerprints $|\phi_x\rangle$ and $|\phi_y\rangle$ would need to be exactly orthogonal for all pairs of distinct n-bit strings x and y. Unfortunately, this constraint makes the quantum fingerprints way too long: an orthonormal set of 2^n states requires 2^n dimensions, which corresponds to n qubits — not much savings over n classical bits! Instead we will settle for *near*-orthogonality, where the required number of dimensions can be made much smaller. Giving up exact orthogonality implies that our test will have a certain error probability, but we can make this error probability as small as we want.

There are many ways to obtain a set of 2^n near-orthogonal states in a small number of dimensions. Below we will use a simple application of the probabilistic method for this, but more constructive methods based on sophisticated error-correcting codes exist as well.

Suppose we pick a set S of 2^n d-bit strings at random, for some d to be determined later. Then the expected Hamming distance $H(s,t)$ between two such strings s and t is $d/2$, and the Chernoff bound tells us that the actual distance is probably close to its expectation:

$$\Pr\left(H(s,t) \notin [0.49d, 0.51d]\right) \leq 2^{-cd}$$

for some positive constant c. Now suppose we choose $d = 2n/c$, then the above probability is at most 2^{-2n} and using the union bound we have

$$\begin{aligned} \Pr\left(\exists s,t \in S \text{ with } H(s,t) \notin [0.49d, 0.51d]\right) &\leq \sum_{s,t\in S} \Pr\left(H(s,t) \notin [0.49d, 0.51d]\right) \\ &\leq \binom{2^n}{2} 2^{-2n} < 1. \end{aligned}$$

In particular, there exists at least one set S where $H(s,t) \in [0.49d, 0.51d]$ *for all* distinct $s,t \in S$. Let us consider such a set. We can index the elements in $S = \{s^x \mid x \in \{0,1\}^n\}$ by the n-bit strings, and derive quantum states from them by using the bits s_i^x in s^x for signs of amplitudes in $|\phi_x\rangle$:

$$|\phi_x\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} (-1)^{s_i^x} |i\rangle.$$

Since these states live in $d = 2n/c$ dimensions, we only need $\log d = \log n + O(1)$ qubits to represent them, so our quantum fingerprints are indeed short compared to the underlying n-bit strings. Two fingerprints are almost orthogonal, because the inner product between $|\phi_x\rangle$ and $|\phi_y\rangle$ is

$$\frac{1}{d} \sum_{i=1}^{d} (-1)^{s_i^x + s_i^y}.$$

Because the Hamming distance between s^x and s^y is close to $d/2$, $s_i^x + s_i^y$ will be even for about half of the is and odd for the other half. Therefore the above sum contains about as many $+1$s as -1s and hence will be small.

We now have our mapping from strings to short quantum fingerprints. It remains to show how we can test whether $x = y$, when given only fingerprints $|\phi_x\rangle$ and $|\phi_y\rangle$. Our test is pictured in Figure 1, where time progresses from left to right: we add on an auxiliary $|0\rangle$-qubit, apply a Hadamard transform to it to get $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, apply a controlled swap to the two registers containing the fingerprints (this swaps the two registers if the auxiliary qubit is $|1\rangle$ and does nothing if it is $|0\rangle$), then apply another Hadamard transform, and finally measure the auxiliary qubit.

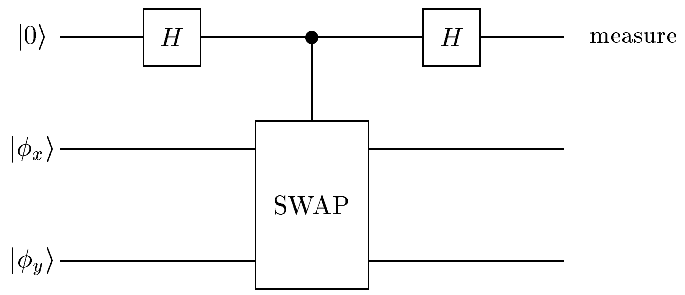

Figure 1: Test whether 2 fingerprints $|\phi_x\rangle$ and $|\phi_y\rangle$ are equal

Let us analyze what happens here. First, if $x = y$ then $|\phi_x\rangle = |\phi_x\rangle$ and the controlled swap has no effect, since swapping two identical things doesn't do anything. In this case, the second Hadamard transform will just set the auxiliary qubit back to the $|0\rangle$-state, so our measurement will give outcome 0 with certainty. On the other hand, if $x \neq y$ then $|\phi_x\rangle$ and $|\phi_y\rangle$ are almost orthogonal. In this case, by calculating the final state one can show that the measurement will give outcome $|1\rangle$ with probability close to 1/2 (where the "closeness" depends on the inner product between $|\phi_x\rangle$ and $|\phi_y\rangle$). Thus one such test allows us to distinguish the two cases $x = y$ and $x \neq y$ with one-sided error probability about $1/2$. If we have a few copies of both fingerprints available, then we can repeat the above test and reduce the error probability to a small constant.

4 Application: Saving communication

We now describe an application of our quantum fingerprinting scheme. We will consider a simple communication scenario. There are three parties: Alice, Bob, and a referee. Alice receives n-bit input x and Bob receives n-bit input y. The referee receives no input, but he wants to find out whether $x = y$ or not (the *equality* problem). Alice and Bob each can send information to the referee, but cannot receive messages from the referee, nor can they communicate with each other. We want a scheme that uses only little communication, but that allows the referee to determine whether $x = y$ with high probability, for all inputs x, y.

Clearly, Alice can send the whole x and Bob can send the whole y, allowing the referee to solve the problem at a cost of $2n$ bits of communication. However, smarter things with less communication are possible. The classical communication complexity of this problem has been studied by various researchers in the last decade, and it turns out that about $\sqrt{n}$ bits of communication are sufficient [1] as well as necessary [6, 2] to solve this equality problem.[1] In contrast, the construction of short quantum fingerprints together with the equality test outlined above, immediately suggest a much more efficient *quantum* solution to the equality problem: Alice sends the fingerprint $|\phi_x\rangle$ to the referee (or a few copies thereof), Bob sends the fingerprint $|\phi_y\rangle$, and the referee just tests whether the two fingerprints he received are equal or almost orthogonal. This gives us a solution to the equality problem that works with high success probability and requires only $O(\log n)$ qubits to be sent, which is exponentially better than the $\sqrt{n}$ bits of communication that are required classically (this also implies that there is no efficient *classical* fingerprinting scheme that achieves the same as our quantum scheme).

For example, suppose Alice and Bob are flying through space, each in their own spaceship. They can only send messages to the command center on earth. They have each gathered a large chunk of data, of 2^{40} bits say, and for some reason the command center needs to know whether they have *the same* chunk of data. Classically, Alice and Bob would each need to send about $\sqrt{2^{40}} \approx 1,000,000$ bits to the referee. In the quantum case, only about 50 qubits of communication would already suffice — a significant savings.

[1] Only $O(1)$ classical bits of communication would suffice if Alice and Bob had access to some shared source of randomness, but we're not allowing that here.

5 Conclusion

We described the quantum fingerprinting technique from [3]. To each n-bit string x we can associate a $\log n$-qubit state $|\phi_x\rangle$, such that we can decide whether $x = y$ by deciding whether $|\phi_x\rangle = |\phi_y\rangle$. In other words, for the purposes of identification the long object x can be replaced by its short fingerprint $|\phi_x\rangle$. This gives rise to an exponential reduction in the communication complexity of the equality problem when we allow quantum communication.

What about other applications of quantum fingerprinting? Note that the fingerprint $|\phi_x\rangle$ gives only little information about x, because a $\log n$-qubit state can contain only $\log n$ bits of classical information (Holevo's theorem). In some sense the quantum fingerprint "contains" x completely without revealing it. Yet we can clearly test or verify whether the hidden x equals some string y of our choice, by testing $|\phi_x\rangle$ against $|\phi_y\rangle$. This information-hiding property of quantum fingerprints smacks of cryptography, and indeed there has recently been some work on "quantum signatures" that uses quantum fingerprints as a building block [4]. Further applications of quantum fingerprinting in communication complexity or cryptography may lie ahead.

References

[1] A. Ambainis. Communication complexity in a 3-computer model. *Algorithmica*, 16(3):298–301, 1996.

[2] L. Babai and P. G. Kimmel. Randomized simultaneous messages: Solution of a problem of Yao in communication complexity. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity*, pages 239–246, 1997.

[3] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16), September 26, 2001. http://xxx.lanl.gov/abs/quant-ph/0102001.

[4] D. Gottesman and I. Chuang. Quantum signatures. quant-ph/0105032, 8 May 2001.

[5] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM STOC*, pages 212–219, 1996. quant-ph/9605043.

[6] I. Newman and M. Szegedy. Public vs. private coin flips in one round communication games. In *Proceedings of 28th ACM STOC*, pages 561–570, 1996.

[7] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[8] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94. quant-ph/9508027.

[9] L. Vandersypen, M. Steffen, G. Breyta, C. Yannoni, M. Sherwood, and I. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(25):883–887, 2001. quant-ph/0112176.