

A Quantum Goldreich-Levin Theorem with Cryptographic Applications

Mark Adcock and Richard Cleve*

Dept. of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4
mark.adcock@cdcg.y.com, cleve@cpsc.ucalgary.ca

Abstract. We investigate the Goldreich-Levin Theorem in the context of quantum information. This result is a reduction from the problem of inverting a one-way function to the problem of predicting a particular bit associated with that function. We show that the quantum version of the reduction is quantitatively more efficient than the known classical version. If the one-way function acts on n -bit strings then the overhead in the reduction is by a factor of $O(n/\varepsilon^2)$ in the classical case but only by a factor of $O(1/\varepsilon)$ in the quantum case, where $\frac{1}{2} + \varepsilon$ is the probability of predicting the hard-predicate. We also show that, using the Goldreich-Levin Theorem, a quantum bit (or qubit) commitment scheme that is perfectly binding and computationally concealing can be obtained from any quantum one-way permutation.

1 Introduction

Fast quantum algorithms are potentially useful in that, if quantum computers that can run them are built, they can then be used to solve computational problems quickly. Algorithms can also be the basis of *reductions* between computational problems in instances where the underlying goals are different from fast computations. For example, reductions are often used as indicators that certain problems are computationally hard, as in the theory of NP-completeness (see [12] and references therein). Another domain where reductions play an important role is in complexity-based cryptography, where a reduction can show that breaking a particular cryptosystem is as difficult (or almost as difficult) as solving a computational problem that is presumed to be hard.

We investigate such a cryptographic setting where quantum algorithms yield different reductions than are possible in the classical case: the so-called Goldreich-Levin Theorem [13]. This result is a reduction from the computational problem of inverting a one-way function to the problem of predicting a particular hard-predicate associated with that function. Roughly speaking, a *one-way function* is a function that can be efficiently computed in the forward direction but is hard to compute in the reverse direction, and a *hard-predicate* of a function is a bit that can be efficiently computed from the input to the function but is hard

* Partially supported by Canada's NSERC and the Killam Trust.

to estimate from the output of the function. We show that the quantum version of the reduction is quantitatively more efficient than the classical version.

Goldreich and Levin essentially showed that, for a problem instance of size n bits, if their hard-predicate can be predicted with probability $\frac{1}{2} + \varepsilon$ with computational cost T then the one-way function can be inverted with computational cost $O(TD(n, \varepsilon))$, where $D(n, \varepsilon)$ is polynomial in n/ε . Taken in its contrapositive form, this means that, if inverting the one-way function requires a computational cost of $\Omega(T)$, then predicting the hard-predicate with probability $\frac{1}{2} + \varepsilon$ requires a computational cost of $\Omega(T/D(n, \varepsilon))$. Note that if we start with a specific lower bound of $\Omega(T)$ for inverting the function then we end up with a weaker lower bound—by a *dilution factor* of $D(n, \varepsilon)$ —for breaking the hard-predicate. In [14], it is shown that the dilution factor can be as small as $O(n/\varepsilon^2)$.

We show that there is a quantum implementation of the reduction where the dilution factor is only $O(1/\varepsilon)$. We also show that $\Omega(n/\varepsilon^2)$ is a lower bound on the dilution factor for classical versions of the reduction in a suitable black-box framework. In the standard parameterization of interest in cryptography, T is assumed to be superpolynomial in n and $\varepsilon \in 1/n^{O(1)}$. In this case, although $1/\varepsilon$ is smaller than n/ε^2 , the diluted computational cost, $T/D(n, \varepsilon)$, remains superpolynomial in both cases. However, there are other parameterizations where the difference between the achievable quantum reduction and best possible classical reduction is more pronounced. One example is the case where $T = n^3$ and $\varepsilon = 1/n$. If we start with a classical one-way function that requires a computational cost of $\Omega(n^3)$ to invert and apply the Goldreich-Levin Theorem to construct a classical hard-predicate then the reduction implies only that the computational cost of predicting the predicate with probability $\frac{1}{2} + \frac{1}{n}$ is lower bounded only by a *constant*. However, if we start with a *quantum* one-way function that requires a computational cost of $\Omega(n^3)$ to invert and apply our quantum version of the Goldreich-Levin Theorem then the computational cost of predicting the predicate with probability $\frac{1}{2} + \frac{1}{n}$ is lower bounded by $\Omega(n^2)$.

A particular application of hard-predicates is for bit commitment. It is well-known that an information theoretically secure bit commitment scheme cannot be based on the information-theoretic properties of quantum devices alone [16, 17]. Of course, this is also the case with *classical* devices, though *computationally secure* bit commitment schemes have been widely proposed, investigated, and applied. Such schemes can be based on the existence of one-way permutations. Most of these proposed one-way permutations are hard to invert only if problems such as factoring or the discrete logarithm are hard, and are insecure against quantum computers, which can efficiently solve such problems [18]. Recently, Dumais, Mayers and Salvail considered the possibility of *quantum* one-way permutations [11], and showed how to base quantum bit commitment on them (see also [10]). Their scheme is perfectly concealing and *computationally binding*, in the sense that changing a commitment is computationally hard if inverting the permutation is hard. We exhibit a complementary quantum bit commitment scheme that is perfectly binding and computationally concealing. As with hard-predicates, the dilution factor in the measure of computational

security is lower than possible with the corresponding classical construction. Furthermore, a possible advantage of our protocol is that the information that must be communicated and stored between the parties consists of $O(n)$ classical bits for bit commitment (and $O(n)$ classical bits plus one qubit for qubit commitment), whereas the scheme in [11] employs $O(n)$ qubits.

The organization of this paper is as follows. In Section 2, we investigate a simple black-box problem associated to the Goldreich-Levin Theorem. In Section 3, we give definitions pertaining to one-way permutations and hard-predicates (classical and quantum versions) and investigate the complexity of reductions from the former to the latter. In Section 4, we show how to use the Goldreich-Levin Theorem to construct a perfectly binding and computationally concealing quantum bit commitment scheme from a quantum one-way permutation.

2 A black-box problem

Our results about the Goldreich-Levin Theorem are based on the query complexity of the following black-box problem, which we refer to as the *GL problem* (see, e.g., [2]). Let n be a positive integer and $\varepsilon > 0$. Let $a \in \{0, 1\}^n$ and let information about a be available only from inner product and equivalence queries.

Definition 1. A *classical inner product (IP)* query (with bias ε) has input $x \in \{0, 1\}^n$ and outputs a bit that is correlated with $a \cdot x$ (the inner product of a and x modulo two) in the sense that, with respect to the uniform distribution,

$$\Pr_x[IP(x) = a \cdot x] \geq \frac{1}{2} + \varepsilon. \quad (1)$$

Definition 2. An *classical equivalence (EQ)* query has input $x \in \{0, 1\}^n$, and the output is 1 if $x = a$ and 0 otherwise.

The goal is to determine a with a minimum number of *IP* and *EQ* queries. A secondary resource under consideration is the number of auxiliary bit/qubit operations. When $\varepsilon = \frac{1}{2}$, this is essentially a problem that Bernstein and Vazirani [3] considered, where *IP* queries return $a \cdot x$ on input x . For this problem, n *IP* queries are necessary and sufficient to solve it classically; however, it can be solved with a single (appropriately defined) quantum *IP* query. (See also [19].) When ε is small—say, $\varepsilon \in 1/n^{O(1)}$ —an efficient classical solution to this problem is nontrivial. The correctness probability of an *IP* query for a particular x cannot readily be amplified by simple techniques such as repeating queries; for some x , $IP(x)$ may always be wrong. Goldreich and Levin [13] were the first to (implicitly) solve this problem with a number of queries and auxiliary operations that is polynomial in n/ε —and this is the basis of their cryptographic reduction in Theorem 11.

We show that any classical algorithm solving the GL problem with constant probability must make $\Omega(n/\varepsilon^2)$ queries (for a reasonable range of values of ε), whereas there is a quantum algorithm that solves the GL problem with $O(1/\varepsilon)$ queries.

Theorem 3. *Any classical probabilistic algorithm solving the GL problem with success probability $\delta > 0$ requires either more than $2^{n/2}$ EQ queries or $\Omega(\delta n/\varepsilon^2)$ IP queries when $\varepsilon \geq \sqrt{n}2^{-n/3}$.*

Proof. The proof uses classical information theory, bounding the conditional mutual information about an unknown string that is revealed by each IP query, in conjunction with an analysis of the effect of EQ queries. Consider an algorithm to be *successful* on a particular input if and only if it performs an EQ query whose output is 1.

We begin by showing that it is sufficient to consider algorithms (formally, decision trees) that are in a convenient simple form. First, by a basic game-theoretic argument [20], it suffices to consider deterministic algorithms, where their input data—embodied in the black-boxes for IP and EQ queries—may be generated in a probabilistic manner. Second, it can be assumed that all EQ queries occur only after all IP queries have been completed. To see why this is so, start with an algorithm that interleaves IP and EQ queries, and modify it as follows. Whenever an EQ query occurs before the end of the IP queries, the modified algorithm stores the value of the input to the query and proceeds as if the result were 0. Then, at the end of the IP queries, each such deferred EQ query is applied. The modified algorithm will behave consistently whenever the actual output of a deferred EQ query is 0, and also it will perform (albeit later) any EQ query where the output is 1. Henceforth, we consider only algorithms with the above simplifications.

Now we describe a probabilistic procedure for constructing the black boxes that perform IP and EQ queries. First, $a \in \{0, 1\}^n$ is chosen randomly, uniformly. Then a set $S \subseteq \{0, 1\}^n$ is chosen randomly, uniformly subject to the condition that $|S| = (\frac{1}{2} + \varepsilon)2^n$ (assuming that $\varepsilon 2^n$ is an integer). Then

$$IP(x) = \begin{cases} a \cdot x & \text{if } x \in S \\ \bar{a} \cdot x & \text{if } x \notin S \end{cases} \quad (2)$$

$$EQ(x) = \begin{cases} 1 & \text{if } x = a \\ 0 & \text{if } x \neq a. \end{cases} \quad (3)$$

Consider an algorithm that makes m IP queries. If $m \geq \delta n/\varepsilon^2$ then the theorem is proven. Otherwise, since $\varepsilon \geq \sqrt{n}2^{-n/3}$, we have

$$m < \frac{\delta n}{\varepsilon^2} \leq \delta 2^{2n/3}. \quad (4)$$

We proceed by determining the amount of information about a that is conveyed by the application of m IP queries. Let A be the $\{0, 1\}^n$ -valued random variable corresponding to the probabilistic choice of $a \in \{0, 1\}^n$, and let Y_1, Y_2, \dots, Y_m be the $\{0, 1\}$ -valued random variables corresponding to the respective outputs of the m IP queries. Let H be the Shannon entropy function (see, e.g., [9]). Then, for each $i \in \{1, 2, \dots, m\}$,

$$H(A|Y_1, \dots, Y_i) = H(A|Y_1, \dots, Y_{i-1}) - H(Y_i|Y_1, \dots, Y_{i-1}) + H(Y_i|A, Y_1, \dots, Y_{i-1}).$$

Combining the above equations yields

$$H(A|Y_1, \dots, Y_m) = H(A) + \sum_{i=1}^m (H(Y_i|A, Y_1, \dots, Y_{i-1}) - H(Y_i|Y_1, \dots, Y_{i-1})). \quad (6)$$

We shall bound each term on the right side of Eq. 6. Since the *a priori* distribution of A is uniform, $H(A) = n$. Since Y_i is $\{0, 1\}$ -valued, $H(Y_i|Y_1, \dots, Y_{i-1}) \leq 1$ for all $i \in \{1, 2, \dots, m\}$. Next, we show that, for all $i \in \{1, 2, \dots, m\}$,

$$H(Y_i|A, Y_1, \dots, Y_{i-1}) \geq 1 - (16/\ln 2)\varepsilon^2. \quad (7)$$

It is useful to view the set S as being generated during the execution of the IP queries as follows. Initially S is empty, and when the first IP query is performed on some input x , x is placed in S with probability $\frac{1}{2} + \varepsilon$ and in \bar{S} with probability $\frac{1}{2} - \varepsilon$. The inputs to subsequent IP queries are placed in S or \bar{S} with appropriate probabilities: the input to the i^{th} query is placed in S with probability

$$\frac{(\frac{1}{2} + \varepsilon)2^n - j}{2^n - (i-1)}, \quad (8)$$

where $j \in \{0, 1, \dots, i-1\}$ is the number of previous inputs to queries that have been placed in S . Using Eq. 4, the above probability can be shown to lie between $\frac{1}{2} - 2\varepsilon$ and $\frac{1}{2} + 2\varepsilon$. It follows that

$$\begin{aligned} H(Y_i|A, Y_1, \dots, Y_{i-1}) &\geq H(\frac{1}{2} + 2\varepsilon, \frac{1}{2} - 2\varepsilon) \\ &= -(\frac{1}{2} + 2\varepsilon) \log(\frac{1}{2} + 2\varepsilon) - (\frac{1}{2} - 2\varepsilon) \log(\frac{1}{2} - 2\varepsilon) \\ &\geq 1 - (16/\ln 2)\varepsilon^2, \end{aligned} \quad (9)$$

establishing Eq. 7. Substituting the preceding inequalities into Eq. 6, we obtain

$$H(A|Y_1, \dots, Y_m) \geq n - (16/\ln 2)m\varepsilon^2. \quad (10)$$

Intuitively, the IP queries yield information about the value of A in terms of their effect on the probability distribution of A conditioned on the values of Y_1, \dots, Y_m . Eq. 10 lower bounds the decrease in entropy possible.

From the conditions of the theorem, it can be assumed that, after the IP queries, $2^{n/2}$ EQ are performed. The algorithm succeeds with probability at least δ only if there exist $2^{n/2}$ elements of $\{0, 1\}^n$ whose total probability is at least δ . The maximum entropy that a distribution with this property can have is for a bi-level distribution, where $2^{n/2}$ elements of $\{0, 1\}^n$ each have probability $\delta/2^{n/2}$ and $2^n - 2^{n/2}$ elements each have probability $(1-\delta)/(2^n - 2^{n/2})$. Therefore,

$$\begin{aligned} H(A|Y_1, \dots, Y_m) &\leq H\left(\underbrace{\frac{\delta}{2^{n/2}}, \dots, \frac{\delta}{2^{n/2}}}_{2^{n/2}}, \underbrace{\frac{1-\delta}{2^n - 2^{n/2}}, \dots, \frac{1-\delta}{2^n - 2^{n/2}}}_{2^n - 2^{n/2}}\right) \\ &= H(\delta, 1-\delta) + \delta \log(2^{n/2}) + (1-\delta) \log(2^n - 2^{n/2}) \\ &< 1 + \delta n/2 + (1-\delta)n \\ &= n - \delta n/2 + 1. \end{aligned} \quad (11)$$

Combining Eq. 10 with Eq. 11, yields $m > (\ln 2)(\delta n - 2)/(32\varepsilon^2) \in \Omega(\delta n/\varepsilon^2)$. \square

Definition 4. A *quantum inner product query* (with bias ε) is a unitary transformation U_{IP} on $n + m$ qubits, or its inverse U_{IP}^\dagger , such that U_{IP} satisfies the following two properties:

1. If $x \in \{0, 1\}^n$ is chosen uniformly at random and the last qubit of $U_{IP}|x\rangle|0^m\rangle$ is measured, yielding $w \in \{0, 1\}$, then $\Pr[w = a \cdot x] \geq \frac{1}{2} + \varepsilon$.
2. For any $x \in \{0, 1\}^n$ and $y \in \{0, 1\}^m$, the state of the first n qubits of $U_{IP}|x\rangle|y\rangle$ is $|x\rangle$.

The first property captures the fact that, taking a query to be a suitable application of U_{IP} followed by a measurement of the last qubit, Eq. 1 is satisfied. Any implementation of a quantum circuit that produces an output that is $a \cdot x$ with probability on average $\frac{1}{2} + \varepsilon$ can be modified to consist of a unitary stage U_{IP} followed by a measurement of one qubit. The second property is for technical convenience and does not restrict generality. Moreover, given a circuit implementing U_{IP} , it is easy to construct a circuit implementing U_{IP}^\dagger .

Definition 5. A *quantum equivalence query* is the unitary operation U_{EQ} such that, for all $x \in \{0, 1\}^n$ and $b \in \{0, 1\}$,

$$U_{EQ}|x\rangle|b\rangle = \begin{cases} |x\rangle|\bar{b}\rangle & \text{if } x = a \\ |x\rangle|b\rangle & \text{if } x \neq a. \end{cases} \quad (12)$$

For the quantum GL problem, $a \in \{0, 1\}^n$ and information about a is available only from quantum *IP* and *EQ* queries and the goal is to determine a . We can now state and prove the result about quantum algorithms for the GL problem (which is similar to a result in [8] in a different context).

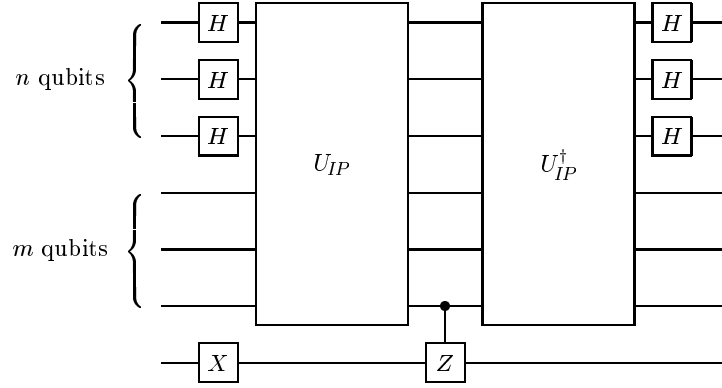
Theorem 6. *There exists a quantum algorithm solving the GL problem with constant probability using $O(1/\varepsilon)$ U_{IP} , U_{IP}^\dagger and U_{EQ} queries in total. Also, the number of auxiliary qubit operations used by the procedure is $O(n/\varepsilon)$.*

Proof. The proof is by a combination of two techniques: the algorithm in [3] for the exact case (i.e., when $\varepsilon = \frac{1}{2}$), which is shown to be adaptable to “noisy” data in [8] (with a slightly different noise model than the one that arises here); and amplitude amplification [5, 15, 6].

Since U_{IP} applied to $|x\rangle|y\rangle$ has no net effect on its first n input qubits, for each $x \in \{0, 1\}^n$, $U_{IP}|x\rangle|0^m\rangle = |x\rangle(\alpha_x|v_x\rangle|a \cdot x\rangle + \beta_x|w_x\rangle|\overline{a \cdot x}\rangle)$, where α_x and β_x are nonnegative real numbers, and $|v_x\rangle$ and $|w_x\rangle$ are $m - 1$ qubit quantum states. If the last qubit of $U_{IP}|x\rangle|0^m\rangle$ is measured then the result is: $a \cdot x$ with probability α_x^2 , and $\overline{a \cdot x}$ with probability β_x^2 . Therefore, since, for a random uniformly distributed $x \in \{0, 1\}^n$, measuring the last qubit of $U_{IP}|x\rangle|0^m\rangle$ yields $a \cdot x$ with probability at least $\frac{1}{2} + \varepsilon$, it follows that

$$\frac{1}{2^n} \sum_{x \in \{0, 1\}^n} \alpha_x^2 \geq \frac{1}{2} + \varepsilon \quad \text{and} \quad \frac{1}{2^n} \sum_{x \in \{0, 1\}^n} \beta_x^2 \leq \frac{1}{2} - \varepsilon. \quad (13)$$

Now, consider the quantum circuit C in Figure 1.

Fig. 1. Quantum circuit C .

We will begin by showing that $\langle a, 0^m, 1 | C | 0^n, 0^m, 0 \rangle$ is real-valued and

$$\langle a, 0^m, 1 | C | 0^n, 0^m, 0 \rangle \geq 2\varepsilon. \quad (14)$$

Note that C can be decomposed into the following five operations:

C_1 : Apply H to each of the first n qubits, and NOT to the last qubit.

C_2 : Apply U_{IP} to the first $n + m$ qubits.

C_3 : Apply a controlled- Z to the last two qubits.

C_4 : Apply U_{IP}^\dagger to the first $n + m$ qubits.

C_5 : Apply H to each of the first n qubits.

The value $\langle a, 0^m, 1 | C | 0^n, 0^m, 0 \rangle$ is the inner product of $C_3 C_2 C_1 | 0^n \rangle | 0^m \rangle | 0 \rangle$ and $C_4^\dagger C_5^\dagger | a \rangle | 0^m \rangle | 1 \rangle$. These states are (omitting normalization factors of $\frac{1}{\sqrt{2^n}}$)

$$\begin{aligned} C_3 C_2 C_1 | 0^n \rangle | 0^m \rangle | 0 \rangle &= C_3 C_2 \sum_{x \in \{0,1\}^n} |x\rangle | 0^m \rangle | 1 \rangle \\ &= C_3 \sum_{x \in \{0,1\}^n} |x\rangle (\alpha_x |v_x\rangle |a \cdot x\rangle + \beta_x |w_x\rangle |\overline{a \cdot x}\rangle) | 1 \rangle \\ &= \sum_{x \in \{0,1\}^n} |x\rangle (\alpha_x (-1)^{a \cdot x} |v_x\rangle |a \cdot x\rangle + \beta_x (-1)^{\overline{a \cdot x}} |w_x\rangle |\overline{a \cdot x}\rangle) | 1 \rangle \\ &= \sum_{x \in \{0,1\}^n} (-1)^{a \cdot x} |x\rangle (\alpha_x |v_x\rangle |a \cdot x\rangle - \beta_x |w_x\rangle |\overline{a \cdot x}\rangle) | 1 \rangle \end{aligned}$$

and

$$\begin{aligned} C_4^\dagger C_5^\dagger | a \rangle | 0^m \rangle | 1 \rangle &= C_4^\dagger \sum_{x \in \{0,1\}^n} (-1)^{a \cdot x} |x\rangle | 0^m \rangle | 1 \rangle \\ &= \sum_{x \in \{0,1\}^n} (-1)^{a \cdot x} |x\rangle (\alpha_x |v_x\rangle |a \cdot x\rangle + \beta_x |w_x\rangle |\overline{a \cdot x}\rangle) | 1 \rangle. \end{aligned}$$

From these equations and the fact that $\langle x|y\rangle = 0$ whenever $x \neq y$, we obtain

$$\langle a, 0^m, 1|C|0^n, 0^m, 0\rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (\alpha_x^2 - \beta_x^2) \geq (\frac{1}{2} + \varepsilon) - (\frac{1}{2} - \varepsilon) = 2\varepsilon,$$

which establishes Eq. 14.

Note that Eq. 14 implies that, if C is executed on input $|0^n, 0^m, 0\rangle$ and the result is measured in the classical basis, then the first n bits of the result will be a with probability at least $4\varepsilon^2$. Therefore, if this process is repeated $O(1/\varepsilon^2)$ times, checking each result with an EQ query, then a will be found with constant probability. A more efficient way of finding the value of a is to use *amplitude amplification* [5, 15, 6] with C and C^\dagger in combination with EQ queries. As shown in [6], an expected number of $O(1/\varepsilon)$ executions of C , C^\dagger , and U_{EQ} are sufficient to obtain a . This implies that $O(1/\varepsilon)$ U_{IP} , U_{IP}^\dagger , and U_{EQ} are sufficient to succeed with constant probability. \square

3 Hard-predicates from one-way permutations

In this section, we give definitions pertaining to one-way permutations and hard-predicates (classical and quantum versions) and investigate the complexity of the reduction of Goldreich and Levin [13] from the former to the latter.

In the definitions below, the *size* of a classical [quantum] circuit, is understood to be relative to a suitable set of gates on one and two bits [qubits]. For a quantum circuit C acting on m qubits, and $x \in \{0, 1\}^n$ (for $n \leq m$), let $C_k(x)$ ($k \in \{1, \dots, m\}$), denote the result of measuring the first k qubits of $C|x\rangle|0^{m-n}$ in the classical basis. The subscript k may be omitted when the value of k is clear from the context.

Intuitively, a quantum one-way permutation f on n bits is easy to compute in the forward direction but is hard to invert¹. For the former property, the standard requirement is that f be computable by a uniform circuit of size $n^{O(1)}$ (though it is also possible to impose other upper bounds on the uniform circuit size).

Definition 7. A permutation $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is *classically [quantumly] (δ, T) -hard to invert* if there is no classical [quantum] circuit C of size T such that $\Pr_a[C(f(a)) = a] \geq \delta$.

Now the standard requirement for the hard-to-invert condition is that f is (δ, T) -hard to invert for all $\delta \in 1/n^{O(1)}$ and $T \in n^{O(1)}$ (again, other bounds can be imposed). The idea behind a hard-predicate [4] is to concentrate the information that a one-way function “hides” about its input into a single bit. Intuitively, $h : \{0, 1\}^n \rightarrow \{0, 1\}$ is a hard-predicate of f if, given $a \in \{0, 1\}^n$, it is easy to compute $h(a)$; whereas, given $f(a)$ for randomly chosen $a \in \{0, 1\}^n$, it is hard to predict the value of the bit $h(a)$ with probability significantly better than $\frac{1}{2}$.

¹ The reversibility of quantum computations does not exclude this possibility [7].

The hard-predicate defined in [13] is

$$h(y, x) = y \cdot x, \tag{16}$$

(the inner product modulo two of x and y), for $(y, x) \in \{0, 1\}^n \times \{0, 1\}^n$. This is not a hard-predicate of f , but for a slightly modified version of f , as follows.

Definition 8. For a permutation $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$, let \tilde{f} denote the permutation $\tilde{f} : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n \times \{0, 1\}^n$ defined as $\tilde{f}(y, x) = (f(y), x)$.

Goldreich and Levin showed that if f is one-way then h is hard to predict from \tilde{f} . Instead of quantifying how well a circuit predicts h from \tilde{f} as the amount by which $\Pr_{y,x}[C(\tilde{f}(y, x)) = h(y, x)]$ exceeds $\frac{1}{2}$, we adopt a related but slightly more complicated definition.

Definition 9. A circuit C (δ, ε) -predicts h from \tilde{f} if

$$\Pr_y[\Pr_x[C(\tilde{f}(y, x)) = h(y, x)] \geq \frac{1}{2} + \varepsilon] \geq \delta. \tag{17}$$

To explain Eq. 17 in words, call $y \in \{0, 1\}^n$ ε -good if $\Pr_x[C(\tilde{f}(y, x)) = h(y, x)] \geq \frac{1}{2} + \varepsilon$ for that value of y . Then then Eq. 17 means $\Pr_y[y \text{ is } \varepsilon\text{-good}] \geq \delta$.

The following lemma, which relates the two measures of prediction, is straightforward to prove by an averaging argument.

Lemma 10. If $\Pr_{y,x}[G(\tilde{f}(y, x)) = h(y, x)] \geq \frac{1}{2} + \varepsilon$ then G $(\varepsilon/(1 - \varepsilon), \varepsilon/2)$ -predicts h from \tilde{f} .

Proof. Let $p = \Pr[y \text{ is } \frac{\varepsilon}{2}\text{-good}]$. Then

$$\begin{aligned} \frac{1}{2} + \varepsilon &\leq \Pr[G(\tilde{f}(y, x)) = h(y, x) | y \text{ is } \frac{\varepsilon}{2}\text{-good}] \Pr[y \text{ is } \frac{\varepsilon}{2}\text{-good}] \\ &\quad + \Pr[G(\tilde{f}(y, x)) = h(y, x) | y \text{ is not } \frac{\varepsilon}{2}\text{-good}] \Pr[y \text{ is not } \frac{\varepsilon}{2}\text{-good}] \\ &< p + (\frac{1}{2} + \frac{\varepsilon}{2})(1 - p), \end{aligned}$$

which implies $p > \varepsilon/(1 - \varepsilon)$. □

Note that, if $\Pr_{y,x}[G(\tilde{f}(y, x)) = h(y, x)] \geq \frac{1}{2} + 1/n^{O(1)}$ then $G(1/n^{O(1)}, 1/n^{O(1)})$ -predicts h from \tilde{f} .

Theorem 11 ([13, 14]). If $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is classically $(\delta/2, T)$ -hard to invert then any classical circuit that (δ, ε) -predicts h from \tilde{f} must have size $\Omega(T\varepsilon^2/n)$.

The proof of this theorem is essentially a reduction from the problem of inverting f to the problem of (δ, ε) -predicting h . One begins by assuming that a circuit G of size $o(T\varepsilon^2/n)$ (δ, ε) -predicts h from \tilde{f} and then shows that, by making $O(n/\varepsilon^2)$ calls to both G and f (plus some additional computations), f can be inverted with probability $\delta/2$ [14]. The total running time of the inversion procedure is $o((n/\varepsilon^2)(T\varepsilon^2/n)) = o(T)$, contradicting the fact that f is $(\delta/2, T)$ -hard to invert.

Our quantum version of the Goldreich-Levin Theorem is the following.

Theorem 12. *If $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is quantumly $(\delta/2, T)$ -hard to invert then any quantum circuit that (δ, ε) -predicts h from \tilde{f} must have size $\Omega(T\varepsilon)$.*

Proof. As in the classical case, the proof is essentially a reduction from the problem of inverting f to the problem of (δ, ε) -predicting h . Let $b = f(a)$ be an input instance—the goal is to determine a from b . We will show how to simulate EQ and IP queries in this setting and then apply the bounds in Theorem 6. It is easy to simulate an EQ query (relative to a) by making one call to f and checking if the result is b . Suppose that there exists a circuit G of size $o(T\varepsilon)$ that (δ, ε) -predicts h from \tilde{f} . Thus, $\Pr_y[\Pr_x[G(\tilde{f}(y, x)) = h(y, x)] \geq \frac{1}{2} + \varepsilon] \geq \delta$. Note that, with probability at least δ , a is ε -good, in the sense that $\Pr_x[G(\tilde{f}(a, x)) = h(a, x)] \geq \frac{1}{2} + \varepsilon$. When a is ε -good, computing $G(\tilde{f}(a, x)) = G(b, x)$ is simulating an IP query for x (relative to a). It follows from Theorem 6 that a can be computed with circuit-size $o((1/\varepsilon)(T\varepsilon)) = o(T)$ with success probability at least $\delta/2$ (where $1/2$ is the success probability of the algorithm that finds a when a is ε -good and δ is the probability that a is ε -good to begin with). This contradicts the $(\delta/2, T)$ -hardness of inverting f , thus such a G cannot exist. \square

To conclude this section, we note that it can be shown that the classical reduction in Theorem 11 cannot be improved asymptotically. Using Theorem 3, it can be shown that a lower bound for the size of \tilde{f} that is larger than $\Omega(T\varepsilon^2/n)$ cannot be obtained classically by a black-box reduction from the problem of inverting f . The details of this are omitted in this version of the paper.

4 Bit commitment from quantum one-way permutations

In this section, we show how to use the Goldreich-Levin Theorem to construct a quantum bit commitment scheme from a quantum one-way permutation.

Definition 13. A permutation $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a *quantum one-way permutation* if:

1. There is a uniform quantum circuit of size $n^{O(1)}$ computing $f(x)$ from x .
2. f is quantumly (δ, T) -hard to invert for any $\delta \in 1/n^{O(1)}$ and $T \in n^{O(1)}$.

Theorem 14. *If there exists a quantum one-way permutation $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ then there exists a bit [or qubit] commitment scheme that is perfectly binding and computationally concealing, in the sense that the committed bit cannot be predicted with probability $\frac{1}{2} + 1/n^{O(1)}$ by a circuit of size $n^{O(1)}$.*

Proof. From Theorem 12, it is straightforward to construct a quantum bit commitment scheme from Alice to Bob based on a one-way permutation f as follows (where $h(y, x) = y \cdot x$).

Bit-commit Let z be the bit to commit to. Alice chooses $a, x \in \{0, 1\}^n$ randomly, sets $c = z \oplus h(a, x)$, computes $b = f(a)$, and sends (b, x, c) to Bob.

Bit-decommit Alice sends a to Bob. Bob checks if $f(a) = b$ and rejects if this is not the case. Otherwise, Bob accepts and computes $c \oplus h(a, x)$ as the bit.

Since f is a permutation there is at most one *classical* value of a that is an acceptable decommitment of Alice's bit. This implies that the scheme is perfectly binding to Alice.

Theorem 12 implies that the scheme is also computationally concealing, since any $n^{O(1)}$ -size circuit that enables Bob to guess z from (b, x, c) with probability $\frac{1}{2} + 1/n^{O(1)}$ can be converted to a $n^{O(1)}$ -size circuit that inverts f with probability $1/n^{O(1)}$, violating the fact that f is one-way.

Finally, we explain how a *qubit* commitment scheme can be constructed using some of the ideas discussed in [1].

Qubit-commit Let $|\psi\rangle$ be the qubit to commit to. Alice chooses $a_1, a_2, x_1, x_2 \in \{0, 1\}^n$ randomly, constructs the state $|\psi'\rangle = X^{h(a_1, x_1)} Z^{h(a_2, x_2)} |\psi\rangle$, computes $b_1 = f(a_1)$ and $b_2 = f(a_2)$, and sends $(|\psi'\rangle, b_1, b_2, x_1, x_2)$ to Bob.

Qubit-decommit Alice sends a_1, a_2 to Bob. Bob checks if $f(a_1) = b_1$ and $f(a_2) = b_2$, rejecting if this is not the case. Otherwise, Bob accepts and computes $Z^{h(a_2, x_2)} X^{h(a_1, x_1)} |\psi'\rangle$ as the qubit.

Clearly, the scheme is perfectly binding. Intuitively, the scheme is computationally concealing, because $h(a_1, x_1)$ and $h(a_2, x_2)$ “look random” to Bob. If Bob can use his information to efficiently significantly distinguish between the qubit that he receives from Alice in the commitment stage and a totally mixed state (density matrix $\frac{1}{2}I$) then this procedure can be adapted to distinguish between the pair of bits $r_1 = h(a_1, x_1)$ and $r_2 = h(a_2, x_2)$ and a pair of truly random bits, which would violate the result proven in Theorem 12. \square

Acknowledgments

We would like to thank Lynn Burroughs, Paul Dumais, Peter Høyer, Dominic Mayers, and Louis Salvail for helpful discussions.

References

1. A. Ambainis, M. Mosca, A. Tapp, R. de Wolf, “Private quantum channels”, *Proc. 41st Ann. IEEE Symp. on Foundations of Computer Science (FOCS '00)*, pp. 547–553, 2000.
2. M. Bellare, “The Goldreich-Levin Theorem”, Manuscript, 1999. (Available at <http://www-cse.ucsd.edu/users/mihir/>.)
3. E. Bernstein and U. V. Vazirani, “Quantum complexity theory”, *SIAM J. on Comput.*, Vol. 26, No. 5, pp. 1411–1473, 1997.
4. M. Blum and S. Micali, “How to generate cryptographically strong sequences of pseudo-random bits”, *SIAM J. on Comput.*, Vol. 13, No. 4, pp. 850–864, 1984.
5. G. Brassard and P. Høyer, “An exact quantum polynomial-time algorithm for Simon's problem”, *Proc. Fifth Israeli Symp. on Theory of Computing and Systems*, pp. 12–23, 1997.
6. G. Brassard, P. Høyer, M. Mosca, A. Tapp, “Quantum amplitude amplification and estimation”, To appear in *Quantum Computation and Quantum Information: A Millennium Volume*, AMS Contemporary Mathematics Volume. Available on the LANL preprint archive as quant-ph/0005055, 2000.

7. H. F. Chau and H.-K. Lo, "One way functions in reversible computations", *Cryptologia*, Vol. 21, No. 2, pp. 139–148, 1997.
8. R. Cleve, W. van Dam, M. Nielsen, and A. Tapp, "Quantum entanglement and the communication complexity of the inner product function", *Proc. of the First NASA International Conf. on Quantum Computing and Quantum Communications*, C. P. Williams (Ed.), Lecture Notes in Computer Science 1509, Springer-Verlag, pp. 61-74, 1999.
9. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley and Sons, 1991.
10. C. Crépeau, F. L egar e and L. Salvail, "How to convert the flavor of a quantum bit commitment", to appear in *Advances in Cryptology — EUROCRYPT 2001*, B. Pfitzmann (Ed.), Lecture Notes in Computer Science 2045, Springer-Verlag, pp. 60-77, 2001.
11. P. Dumais, D. Mayers, and L. Salvail, "Perfectly concealing quantum bit commitment from any one-way permutation", *Advances in Cryptology — EUROCRYPT 2000*, B. Preneel (Ed.), Lecture Notes in Computer Science 1807, Springer-Verlag, pp. 300–315, 2000.
12. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., 1979.
13. O. Goldreich and L. Levin, "Hard-core predicates for any one-way function", *Proc. 21th Ann. ACM Symp. on Theory of Computing (STOC '89)*, pp. 25–32, 1989.
14. O. Goldreich, *Modern Cryptography, Probabilistic Proofs and Pseudo-Randomness*, Springer, 1999.
15. L. K. Grover, "A fast quantum mechanical algorithm for database search", *Proc. 28th Ann. ACM Symp. on Theory of Computing (STOC '96)*, pp. 212–219, 1996.
16. H.-K. Lo and H. F. Chau, "Is quantum bit commitment really possible?", *Phys. Rev. Lett.*, Vol. 78, No. 17, pp. 3410–3413, 1997.
17. D. Mayers, "Unconditionally secure bit commitment is impossible", *Phys. Rev. Lett.*, Vol. 78, No. 17, pp. 3414–3417, 1997.
18. P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", *SIAM J. on Computing*, Vol. 26, No. 5, pp. 1484–1509, 1997.
19. B. M. Terhal and J. A. Smolin, "Single quantum querying of a database", *Phys. Rev. A*, Vol. 58, No. 3, pp. 1822–1826, 1998.
20. A. C.-C. Yao, "Lower bounds by probabilistic arguments", *Proc. 24th Ann. IEEE Symp. on Foundations of Computer Science (FOCS '83)*, pp. 420–428, 1983.