

## A search for quantum coin-flipping protocols using optimization techniques

Ashwin Nayak · Jamie Sikora · Levent Tunçel

Received: date / Accepted: date

**Abstract** Coin-flipping is a cryptographic task in which two physically separated, mistrustful parties wish to generate a fair coin-flip by communicating with each other. Chailloux and Kerenidis (2009) designed quantum protocols that guarantee coin-flips with near optimal bias away from uniform, even when one party deviates arbitrarily from the protocol. The probability of any outcome in these protocols is provably at most  $\frac{1}{\sqrt{2}} + \delta$  for any given  $\delta > 0$ . However, no explicit description of these protocols is known; in fact, the smallest bias achieved by known explicit protocols is  $1/4$  (Ambainis, 2001).

We take a *computational optimization* approach, based mostly on convex optimization, to the search for simple and explicit quantum strong coin-flipping protocols. We present a search algorithm to identify protocols with low bias within a natural class, protocols based on *bit-commitment* (Nayak and Shor, 2003). The techniques we develop enable a computational search for protocols given by a mesh over the corresponding parameter space. We conduct searches for four-round and six-round protocols with bias below 0.2499 each of varying dimension which include the best known explicit protocol (with bias  $1/4$ ). After checking over  $10^{16}$  protocols, a task which would be infeasible using

---

A. Nayak

Department of Combinatorics and Optimization, and Institute for Quantum Computing, University of Waterloo. Address: 200 University Ave. W., Waterloo, ON, N2L 3G1, Canada. Tel.: +1 519 888-4567 extension 33601  
E-mail: ashwin.nayak@uwaterloo.ca

J. Sikora

Centre for Quantum Technologies, National University of Singapore. Address: Block S15, 3 Science Drive 2, Singapore 117543.  
E-mail: cqjwjs@nus.edu.sg

L. Tunçel

Department of Combinatorics and Optimization, University of Waterloo. Address: 200 University Ave. W., Waterloo, ON, N2L 3G1, Canada.  
Tel.: +1 519 888-4567 ext. 35598  
E-mail: ltuncel@uwaterloo.ca

semidefinite programming alone, we conjecture that the smallest achievable bias within the family of protocols we consider is  $1/4$ .

**Keywords** Semidefinite programming · Quantum coin-flipping · Computational optimization

## 1 Introduction

*Quantum coin-flipping* is a fundamental task in Quantum Cryptography which can potentially be used as a building block for other, more sophisticated tasks in quantum computing. Many fundamental problems about quantum coin-flipping (e.g., determination of the bias of a given protocol) allow formulations in the language of convex optimization in the space of Hermitian matrices over the complex numbers, in particular, in the language of semidefinite optimization. However, the problem of finding a good quantum coin-flipping protocol using such SDP (Semidefinite Programming) formulations becomes a very hard, nonconvex optimization problem. In our approach, we design an algorithm to approximately solve the nonconvex problem. We take a computational optimization approach. We treat the space of quantum coin-flipping protocols as a data space for our SDPs and numerically search for good protocols in this data space by solving a huge number of SDPs. To speed up the search, we derive new bounds on the optimal objective function values of the SDPs by special feasible solutions. These feasible solutions are obtained by analytically solving restrictions to the feasible region, which in turn are derived from good quantum coin-flipping strategies. The resulting bounds are attractive since they have closed-form expressions which can be computed very efficiently. If these solutions are found to have large objective function value (large with respect to the value of previously analyzed good protocols), then the need to invoke the general purpose SDP solver is eliminated, thereby saving time.

We now discuss *quantum coin-flipping* and introduce our approach.

**Quantum coin-flipping.** Coin-flipping is a classic cryptographic task introduced by Blum [6]. In this task, two remotely situated parties, Alice and Bob, would like to agree on a uniformly random bit by communicating with each other. The complication is that neither party trusts the other. If Alice were to toss a coin and send the outcome to Bob, Bob would have no means to verify whether this was a uniformly random outcome. In particular, if Alice wishes to cheat, she could send the outcome of her choice without any possibility of being caught cheating. We are interested in a communication protocol that is *designed to protect* an honest party from being cheated.

More precisely, a “strong coin-flipping protocol” with bias  $\epsilon$  is a two-party communication protocol in the style of Yao [29,30]. In the protocol, the two players, Alice and Bob, start with no inputs and compute a value  $c_A, c_B \in \{0, 1\}$ , respectively, or declare that the other player is cheating. If both players are honest, i.e., follow the protocol, then they agree on the outcome of the protocol ( $c_A = c_B$ ), and the coin toss is fair ( $\Pr(c_A = c_B = b) = 1/2$ ,

for any  $b \in \{0, 1\}$ ). Moreover, if one of the players deviates arbitrarily from the protocol in his or her local computation, i.e., is “dishonest” (and the other party is honest), then the probability of either outcome 0 or 1 is at most  $1/2 + \epsilon$ . Other variants of coin-flipping have also been studied in the literature. However, in the rest of the article, by “coin-flipping” (without any modifiers) we mean *strong* coin flipping.

A straightforward game-theoretic argument proves that if the two parties in a coin-flipping protocol communicate classically and are computationally unbounded, at least one party can cheat perfectly (with bias  $1/2$ ). In other words, there is at least one party, say Bob, and at least one outcome  $b \in \{0, 1\}$  such that Bob can ensure outcome  $b$  with probability 1 by choosing his messages in the protocol appropriately. Consequently, classical coin-flipping protocols with bias  $\epsilon < 1/2$  are only possible under complexity-theoretic assumptions, and when Alice and Bob have limited computational resources.

Quantum communication offers the possibility of “unconditionally secure” cryptography, wherein the security of a protocol rests solely on the validity of quantum mechanics as a faithful description of nature. The first few proposals for quantum information processing, namely the Wiesner quantum money scheme [27] and the Bennett-Brassard quantum key expansion protocol [4] were motivated by precisely this idea. These schemes were indeed eventually shown to be unconditionally secure in principle [15, 13, 21, 18]. In light of these results, several researchers have studied the possibility of *quantum* coin-flipping protocols, as a step towards studying more general secure multi-party computations.

Lo and Chau [12] and Mayers [14] were the first to consider quantum protocols for coin-flipping without any computational assumptions. They proved that no protocol with a finite number of rounds could achieve 0 bias. Nonetheless, Aharonov, Ta-Shma, Vazirani, and Yao [2] designed a simple, three-round quantum protocol that achieved bias  $\approx 0.4143 < 1/2$ . This is impossible classically, even with an unbounded number of rounds. Ambainis [3] designed a protocol with bias  $1/4$  *à la* Aharonov *et al.*, and proved that it is optimal within a class (see also Refs. [23, 10] for a simpler version of the protocol and a complete proof of security). Shortly thereafter, Kitaev [11] proved that any strong coin-flipping protocol with a finite number of rounds of communication has bias at least  $(\sqrt{2} - 1)/2 \approx 0.207$  (see Ref. [9] for an alternative proof). Kitaev’s seminal work uses semidefinite optimization in a central way. This argument extends to protocols with an unbounded number of rounds. This remained the state of the art for several years, with inconclusive evidence in either direction as to whether  $1/4 = 0.25$  or  $(\sqrt{2} - 1)/2$  is optimal. In 2009, Chailloux and Kerenidis [7] settled this question through an elegant protocol scheme that has bias at most  $(\sqrt{2} - 1)/2 + \delta$  for any  $\delta > 0$  of our choice (building on [17], see below). We refer to this as the CK protocol.

The CK protocol uses breakthrough work by Mochon [17], which itself builds upon the “point game” framework proposed by Kitaev. Mochon shows there are *weak* coin-flipping protocols with arbitrarily small bias. (This work has appeared only in the form of an unpublished manuscript, but has been

verified by experts on the topic; see e.g. [1].) A weak coin-flipping protocol is a variant of coin-flipping in which each party favours a distinct outcome, say Alice favours 0 and Bob favours 1. The requirement when they are honest is the same as before. We say it has bias  $\epsilon$  if the following condition holds. When Alice is dishonest and Bob honest, we only require that Bob’s outcome is 0 (Alice’s favoured outcome) with probability at most  $1/2+\epsilon$ . A similar condition to protect Alice holds, when she is honest and Bob is dishonest. The weaker requirement of security against a dishonest player allows us to circumvent the Kitaev lower bound. While Mochon’s work pins down the optimal bias for weak coin-flipping, it does this in a non-constructive fashion: we only know of the *existence* of protocols with arbitrarily small bias, not of its *explicit description*. Moreover, the number of rounds tends to infinity as the bias decreases to 0. As a consequence, the CK protocol for strong coin-flipping is also existential, and the number of rounds tends to infinity as the bias decreases to  $(\sqrt{2}-1)/2$ . It is perhaps very surprising that no progress on finding better explicit protocols has been made in over a decade.

**Search for explicit protocols.** This work is driven by the quest to find *explicit* and *simple* strong coin-flipping protocols with bias smaller than  $1/4$ . There are two main challenges in this quest. First, there seems to be little insight into the structure (if any) that protocols with small bias have; knowledge of such structure might help narrow our search for an optimal protocol. Second, the analysis of protocols, even those of a restricted form, with more than three rounds of communication is technically quite difficult. As the first step in deriving the  $(\sqrt{2}-1)/2$  lower bound, Kitaev [11] proved that the optimal cheating probability of any dishonest party in a protocol with an explicit description is characterized by a semidefinite program (SDP). While this does not entirely address the second challenge, it reduces the analysis of a protocol to that of a well-studied optimization problem. In fact this formulation as an SDP enabled Mochon to analyze an important class of weak coin-flipping protocols [16], and later discover the optimal weak coin flipping protocol [17]. SDPs resulting from strong coin-flipping protocols, however, do not appear to be amenable to similar analysis.

We take a *computational optimization* approach to the search for explicit strong coin-flipping protocols. We focus on a class of protocols studied by Nayak and Shor [19] that are based on “bit-commitment”. This is a natural class of protocols that generalizes those due to Aharonov *et al.* and Ambainis, and provides a rich test bed for our search. (See Section 3 for a description of such protocols.) Early proposals of multi-round protocols in this class were all shown to have bias at least  $1/4$ , without eliminating the possibility of smaller bias (see, e.g., Ref. [19]). A characterization of the smallest bias achievable in this class would be significant progress on the problem: it would either lead to simple, explicit protocols with bias smaller than  $1/4$ , or we would learn that protocols with smaller bias take some other, yet to be discovered form.

Chailloux and Kerenidis [8] have studied a version of bit-commitment that may have implications for coin-flipping. They proved that in any quantum bit-commitment protocol with computationally unbounded players, at least

one party can cheat with bias at least  $\approx 0.239$ . Since the protocols we study involve two interleaved commitments to independently chosen bits, this lower bound does not apply to the class. Chailloux and Kerenidis also give a protocol scheme for bit-commitment that guarantees bias arbitrarily close to 0.239. The protocol scheme is non-constructive as it uses the Mochon weak coin-flipping protocol. It is possible that any explicit protocols we discover for coin-flipping could also lead to explicit bit-commitment with bias smaller than  $1/4$ .

We present an algorithm for finding protocols with low bias. Each bit-commitment based coin-flipping protocol is specified by a 4-tuple of quantum states. At a high level, the algorithm iterates through a suitably fine mesh of such 4-tuples, and computes the bias of the resulting protocols. The size of the mesh scales faster than  $1/\nu^{\kappa D}$ , where  $\nu$  is a precision parameter,  $\kappa$  is a universal constant, and  $D$  is the dimension of the states. The dimension itself scales as  $2^n$ , where  $n$  is the number of quantum bits involved. In order to minimize the doubly exponential size of the set of 4-tuples we examine, we further restrict our attention to states of the form introduced by Mochon for weak coin-flipping [16]. The additional advantage of this kind of state is that the SDPs in the analysis of the protocols simplify drastically. In fact, all but a few constraints reduce to linear equalities so that the SDPs may be solved more efficiently.

Next, we employ two techniques to prune the search space of 4-tuples. First, we use a sequence of strategies for dishonest players whose bias is given by a closed form expression determined by the four states. The idea is that if the bias for any of these strategies is higher than  $1/4$  for any 4-tuple of states, we may safely rule it out as a candidate optimal protocol. This also has the advantage of avoiding a call to the SDP solver, the computationally most intensive step in the search algorithm. The second technique is to invoke symmetries in the search space as well as in the problem to identify protocols with the same bias. The idea here is to compute the bias for as few members of an equivalence class of protocols as possible.

These techniques enable a computational search for protocols with up to six rounds of communication, with messages of varying dimension. The Ambainis protocol with bias  $1/4$  has three rounds, and it is entirely possible that a strong coin-flipping protocol with a small number of rounds be optimal. Thus, the search non-trivially extends our understanding of this cryptographic primitive. We elaborate on this next.

**The results.** We performed searches that sought protocols within the mesh with bias at most  $1/4$  minus a small constant. We chose the constant to be 0.001. The rationale here was that if the mesh contains protocols with bias close to the lower bound of  $\approx 0.207$ , we would find protocols that have bias closer to 0.25 (but smaller than it) relatively quickly. We searched for four-round protocols in which each message is of dimension ranging from 2 to 9, each with varying fineness for the mesh. We found that our heuristics, i.e., the filtering by fixed cheating strategies, performed so well that they eliminated every protocol: all of the protocols given by the mesh were found to have bias larger than 0.2499 without the need to solve any SDP.

The initial search for four-round protocols helped us fine-tune the filter by a careful selection of the order in which the cheating strategies were tried. The idea was to eliminate most protocols with the least amount of computation. This made it feasible for us to search for protocols in finer meshes, with messages of higher dimension, and with a larger number of rounds. In particular, we were able to check six-round protocols with messages of dimension 2 and 3. Our heuristics again performed very well, eliminating almost every protocol before any SDP needed to be solved. Even during this search, not a single protocol with bias less than 0.2499 was found.

It may not immediately be evident that the above searches involved a computational examination of extremely large sets of protocols and that the techniques described above in Section 5, were crucial in enabling this search. The symmetry arguments pruned the searches drastically, and in some cases only 1 in every 1,000,000 protocols needed to be checked. In most cases, the cheating strategies (developed in Subsection 5.1) filtered out the rest of the protocols entirely. To give an example of the efficiency of our search, we were able to check  $2.74 \times 10^{16}$  protocols in a matter of days. Without the symmetry arguments and the use of cheating strategies as a filter, this same search would have taken well over 69 million years, even using the very simplified forms of the SDPs. Further refinement of these ideas may make a more thorough search of protocols with four or more rounds feasible.

Finally, based on our computational findings, we make the following conjecture: Any strong coin-flipping protocol based on bit-commitment as defined formally in Section 3 has bias at least  $1/4$ . This conjecture, if true, would imply that we need to investigate new kinds of protocols to find ones with bias less than  $1/4$ . Regardless of the truth of the above conjecture, we hope that the new techniques developed for analyzing protocols via modern optimization methods and for simplifying semidefinite optimization problems with special structure will be helpful in future work in the areas of quantum computing and semidefinite programming.

**Organization of the paper.** We begin with an introduction to the ideas contained in this paper in Section 2 including an introduction to quantum computing and semidefinite programming. Section 3 defines strong coin-flipping protocols and the measure of their security (namely, their bias). We define the notion of protocols based on bit-commitment and model optimal cheating strategies for such protocols using semidefinite programming in Section 4. Section 5 introduces several heuristics to speed up our search including the use of a *protocol filter* (Subsection 5.1) and *symmetry* (Subsection 5.2). Our search algorithm is presented in Section 6 and our numerical results in Section 7. We conclude with some final remarks in Section 8.

The background material on quantum computation and optimization is aimed at making this work accessible to researchers in both communities. Readers conversant with either topic need only skim the corresponding sections to familiarize themselves with the notation used.

## 2 Background and notation

In this section, we establish the notation and the necessary background for this paper.

**Linear algebra.** For a finite set  $A$ , we denote by  $\mathbb{R}^A$ ,  $\mathbb{R}_+^A$ ,  $\text{Prob}^A$ , and  $\mathbb{C}^A$  the set of real vectors, nonnegative real vectors, probability vectors, and complex vectors, respectively, each indexed by  $A$ . We use  $\mathbb{R}^n$ ,  $\mathbb{R}_+^n$ ,  $\text{Prob}^n$ , and  $\mathbb{C}^n$  for the special case when  $A = \{1, \dots, n\}$ . For  $x \in A$ , the vectors  $e_x$  denote the standard basis vectors of  $\mathbb{R}^A$ . The vector  $e_A \in \mathbb{R}^A$  denotes the all 1 vector  $\sum_{x \in A} e_x$ .

We denote by  $\mathbb{S}^A$  and  $\mathbb{S}_+^A$  the set of Hermitian matrices and positive semidefinite matrices, respectively, each over the reals with columns and rows indexed by  $A$ .

It is convenient to define  $\sqrt{x}$  to be the element-wise square root of a nonnegative vector  $x$ . The element-wise square root of a probability vector yields a unit vector (in the Euclidean norm). This operation maps a probability vector to a quantum state (defined later in this section).

For vectors  $x$  and  $y$ , the notation  $x \geq y$  denotes that  $x - y$  has nonnegative entries,  $x > y$  denotes that  $x - y$  has positive entries, and for matrices  $X$  and  $Y$ , the notation  $X \succeq Y$  denotes that  $X - Y$  is positive semidefinite, and  $X \succ Y$  denotes  $X - Y$  is positive definite when the underlying spaces are clear from context. When we say that a matrix is positive semidefinite or positive definite, it is assumed to be Hermitian which implies that  $\mathbb{S}_+^A \subset \mathbb{S}^A$ .

The Kronecker product of matrices  $X$  and  $Y$ , denoted  $X \otimes Y$ , is defined such that the  $i, j$ 'th block is equal to  $X_{i,j} \cdot Y$ . Note that  $X \otimes Y \in \mathbb{S}_+^{A \times B}$  when  $X \in \mathbb{S}_+^A$  and  $Y \in \mathbb{S}_+^B$  and  $\text{Tr}(X \otimes Y) = \text{Tr}(X) \cdot \text{Tr}(Y)$  when  $X$  and  $Y$  are square.

The *Schatten 1-norm*, or *nuclear norm*, of a matrix  $X$  is defined as

$$\|X\|_* := \text{Tr}(\sqrt{X^*X}),$$

where  $X^*$  is the adjoint of  $X$  and  $\sqrt{X}$  denotes the square root of a positive semidefinite matrix  $X$ , i.e., the positive semidefinite matrix  $Y$  such that  $Y^2 = X$ . Note that the 1-norm of a matrix is the sum of its singular values. The 1-norm of a vector  $p \in \mathbb{C}^A$  is denoted as

$$\|x\|_1 := \sum_{x \in A} |p_x|.$$

We use the notation  $\bar{a}$  to denote the complement of a bit  $a$  with respect to 0 and 1 and  $a \oplus b$  to denote the XOR of the bits  $a$  and  $b$ . We use  $\mathbb{Z}_2^n$  to denote the set of  $n$ -bit binary strings.

For a vector  $p \in \mathbb{R}^A$ , we denote by  $\text{Diag}(p) \in \mathbb{S}^A$  the diagonal matrix with  $p$  on the diagonal. For a matrix  $X \in \mathbb{S}^A$ , we denote by  $\text{diag}(X) \in \mathbb{R}^A$  the vector on the diagonal of  $X$ .

For a vector  $x \in \mathbb{C}^A$ , we denote by  $\text{supp}(x)$  the set of indices of  $A$  where  $x$  is nonzero. We denote by  $x^{-1}$  the element-wise inverse of  $x$  (mapping the 0 entries to 0).

For a matrix  $X$ , we denote by  $\text{Null}(X)$  the nullspace of  $X$ , by  $\det(X)$  the determinant of  $X$ , and by  $\lambda_{\max}(X)$  the largest eigenvalue of  $X$ . We denote by  $\langle X, Y \rangle$  the standard inner product  $\text{Tr}(X^*Y)$  of matrices  $X, Y$  of the same dimension.

**Convex analysis.** A *convex combination* of finitely many vectors  $x_1, \dots, x_n$  is any vector of the form  $\sum_{i=1}^n \lambda_i x_i$ , when  $\lambda_1, \dots, \lambda_n \in [0, 1]$  satisfy  $\sum_{i=1}^n \lambda_i = 1$ . The *convex hull* of a set  $C$  is the set of convex combinations of elements of  $C$ , denoted  $\text{conv}(C)$ . A set  $C$  is *convex* if  $C = \text{conv}(C)$ .

A *convex function*  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$  is one that satisfies

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \text{ for all } x, y \in \mathbb{R}^n, \lambda \in [0, 1].$$

A convex function is *strictly convex* if

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y), \text{ for all } x \neq y, x, y \in \mathbb{R}^n, \lambda \in (0, 1).$$

We say that a convex function is *proper* if  $f(x) < +\infty$  for some  $x \in \mathbb{R}^n$ . The *epigraph* of a function  $f$  is the set

$$\text{epi}(f) := \{(x, t) : f(x) \leq t\}$$

which are the points above the graph of the function  $f$ . A function is convex if and only if its epigraph is a convex set.

A function  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty\}$  is *(strictly) concave* if  $-f$  is (strictly) convex, and *proper* when  $f(x) > -\infty$  for some  $x \in \mathbb{R}^n$ . The *hypograph* of a function  $f$  is the set

$$\text{hypo}(f) := \{(x, t) : f(x) \geq t\}$$

which are the points below the graph of the function  $f$ . A function is concave if and only if its hypograph is a convex set.

Let  $f_1, \dots, f_n : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{\infty\}$  be proper, convex functions. We denote the *convex hull* of the functions  $\{f_1, \dots, f_n\}$  by  $\text{conv}\{f_1, \dots, f_n\}$  which is the greatest convex function  $f$  such that  $f(x) \leq f_1(x), \dots, f_n(x)$  for every  $x \in \mathbb{R}^m$ . The convex hull can be written in terms of the epigraphs

$$\text{conv}\{f_1, \dots, f_n\}(x) := \inf \{t : (x, t) \in \text{conv}(\cup_{i=1}^n \text{epi}(f_i))\}.$$

We denote the *concave hull* of  $\{f_1, \dots, f_n\}$  by  $\text{conc}\{f_1, \dots, f_n\}$  which can be written as

$$\text{conc}\{f_1, \dots, f_n\} := -\text{conv}\{-f_1, \dots, -f_n\}$$

when  $f_1, \dots, f_n : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{-\infty\}$  are proper, concave functions. The concave hull is the least concave function  $f$  such that  $f(x) \geq f_1(x), \dots, f_n(x)$  for every  $x \in \mathbb{R}^m$  and can be written as

$$\text{conc}\{f_1, \dots, f_n\}(x) := \sup \{t : (x, t) \in \text{conv}(\cup_{i=1}^n \text{hypo}(f_i))\}.$$

A *convex optimization problem* or *convex program* is one of the form

$$\inf_{x \in C} f(x),$$



where  $f$  is a convex function and  $C$  is a convex set. Alternatively, one could maximize a concave function over a convex set.

**Semidefinite programming.** A natural model of optimization when studying quantum information is semidefinite programming. A *semidefinite program*, abbreviated as SDP, is an optimization problem of the form

$$(P) \quad \sup\{\langle C, X \rangle : \mathcal{A}(X) = b, X \in \mathbb{S}_+^n\},$$

where  $\mathcal{A} : \mathbb{S}^n \rightarrow \mathbb{R}^m$  is linear,  $C \in \mathbb{S}^n$ , and  $b \in \mathbb{R}^m$ . The SDPs that arise in quantum computation may involve optimization over complex matrices. However, they may be transformed to the above standard form in a straightforward manner, by observing that Hermitian matrices form a real subspace of the vector space of  $n \times n$  complex matrices. However, in this paper we only have need to study SDPs defined over real variables.

Similar to linear programs, every SDP has a dual. We can write the dual of (P) as

$$(D) \quad \inf\{\langle b, y \rangle : \mathcal{A}^*(y) - S = C, S \in \mathbb{S}_+^n\},$$

where  $\mathcal{A}^*$  is the adjoint of  $\mathcal{A}$ . We refer to (P) as the primal problem and to (D) as its dual. We say  $X$  is *feasible* for (P) if it satisfies the constraints  $\mathcal{A}(X) = b$  and  $X \in \mathbb{S}_+^n$ , and  $(y, S)$  is feasible for (D) if  $\mathcal{A}^*(y) - S = C, S \in \mathbb{S}_+^n$ . The usefulness of defining the dual in the above manner is apparent in the following lemmas.

**Lemma 1 (Weak duality)** *For every  $X$  feasible for (P) and  $(y, S)$  feasible for (D) we have  $\langle C, X \rangle \leq \langle b, y \rangle$ .*

Using weak duality, we can prove bounds on the optimal objective value of (P) and (D), i.e., the objective function value of any primal feasible solution yields a lower bound on (D) and the objective function value of any dual feasible solution yields an upper bound on (P).

Under mild conditions, we have that the optimal objective values of (P) and (D) coincide.

**Lemma 2 (Strong duality)** *If the objective function of (P) is bounded from above on the set of feasible solutions of (P) and there exists a strictly feasible solution, i.e., there exists  $\bar{X} \succ 0$  such that  $\mathcal{A}(\bar{X}) = b$ , then (D) has an optimal solution and the optimal objective values of (P) and (D) coincide.*

A strictly feasible solution as in the above lemma is also called a *Slater point*.

Semidefinite programming has a powerful and rich duality theory and the interested reader is referred to [28], [26] and the references therein.

**Quantum information.** We now give a brief introduction to quantum information. For a more thorough treatment of the subject, we refer the reader to [20].

**Quantum states.** Quantum states are a description of the state of a physical system, such as the spin of an electron. In the simplest case, such a

state is a *unit vector* in a finite-dimensional Hilbert space (which is a complex Euclidean space). For example, the following vectors are quantum states in  $\mathbb{C}^2$

$$e_0 := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad e_1 := \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad e_+ := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad e_- := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

The first two are standard basis vectors and can be thought of as the logical states of a standard computer. In general, a qubit can be written as

$$\psi := \alpha_0 e_0 + \alpha_1 e_1,$$

where  $\alpha_0, \alpha_1 \in \mathbb{C}$  satisfy  $|\alpha_0|^2 + |\alpha_1|^2 = 1$ . This condition ensures that  $\psi$  has norm equal to 1. Up to factor of modulus 1, the set of pairs  $(\alpha_0, \alpha_1)$  defining a two-dimensional quantum state is in one-to-one correspondence with the unit sphere in  $\mathbb{R}^3$ .

Systems with a two dimensional state space are called *quantum bits* or *qubits*. The state space of a sequence of  $n$  qubits is given by the  $n$ -fold tensor product  $(\mathbb{C}^2)^{\otimes n} \cong \mathbb{C}^{2^n}$ . Higher dimensional systems, say, of dimension  $d \leq 2^n$ , may be viewed as being composed of a sequence of  $n$  qubits via a canonical isometry  $\mathbb{C}^d \rightarrow \mathbb{C}^{2^n}$ .

Notice that  $e_+ = \frac{1}{\sqrt{2}}e_0 + \frac{1}{\sqrt{2}}e_1$  and  $e_- = \frac{1}{\sqrt{2}}e_0 - \frac{1}{\sqrt{2}}e_1$ . These states are said to be in a *superposition* of the states  $e_0$  and  $e_1$  and exhibit properties of being in both states at the same time. This is in part what gives quantum computers the power to efficiently tackle hard problems such as factoring [22].

In general, a system may be in a random superposition according to some probability distribution. Suppose a quantum system is in such a state drawn from the ensemble of states  $(\psi_0, \psi_1, \dots, \psi_n)$  with probabilities  $(p_0, p_1, \dots, p_n)$ , respectively. This quantum state may be described more succinctly as a *density matrix*, defined as

$$\sum_{i=0}^n p_i \psi_i \psi_i^*.$$

Notice that this matrix is positive semidefinite and has unit trace. Moreover, any positive semidefinite matrix with unit trace can be written in the above form using its spectral decomposition.

Two different probability distributions over superpositions may have the same density matrix. For example, density matrices do not record “phase information”, i.e., the density matrix of state  $\psi$  is the same as that of  $-\psi$ . However, two ensembles with the same density matrix behave identically under all allowed physical operations. Therefore, there is no loss in working with density matrices, and we identify an ensemble with its density matrix.

A quantum superposition given by the vector  $\psi$  corresponds to the rank 1 density matrix  $\psi\psi^*$  and we call it a *pure state*. States with a density matrix of rank 2 or more are said to be *mixed*.

**Quantum operations.** The most basic quantum operation is specified by a unitary transformation. Suppose  $U$  is a unitary operator acting on  $\mathbb{C}^A$  and  $\psi \in \mathbb{C}^A$  is a quantum state. If we apply  $U$  to  $\psi$  then the resulting quantum

state is  $U\psi \in \mathbb{C}^A$ . Note this is a well-defined quantum state since unitary operators preserve Euclidean norm.

Suppose we are given a state drawn from the ensemble  $(\psi_0, \psi_1, \dots, \psi_n)$  with probabilities  $(p_0, p_1, \dots, p_n)$ . Then if we apply a unitary matrix  $U$  to the state, the resulting state is given by the ensemble  $(U\psi_0, U\psi_1, \dots, U\psi_n)$  with the same probabilities. The new density matrix is thus

$$\sum_{i=0}^n p_i U\psi_i\psi_i^*U^* = U \left( \sum_{i=0}^n p_i \psi_i\psi_i^* \right) U^*,$$

where  $U^*$  is the adjoint of  $U$ . Thus, if we apply the unitary  $U$  to a state (with density matrix)  $\rho$ , then the resulting quantum state is  $U\rho U^*$ . Note that this matrix is still positive semidefinite with unit trace.

We assume that parties capable of quantum information processing have access to qubits initialized to a fixed quantum state, say  $e_0$ , can apply arbitrary unitary operations, and can physically transport (“send”) qubits without disturbing their state. We use the phrase “prepare a quantum state  $\psi \in \mathbb{C}^A$ ” to mean that we start with sufficiently many qubits (say  $n$  such that  $\mathbb{C}^A \subseteq \mathbb{C}^{2^n}$ ) in state  $e_0^{\otimes n}$  and apply any unitary transformation that maps  $e_0^{\otimes n}$  to  $\psi$ .

**Quantum measurement.** Measurement is a means of extracting classical information from a quantum state. A *quantum measurement* on space  $\mathbb{C}^A$  is a sequence of positive semidefinite operators  $(\Pi_1, \dots, \Pi_n)$ , with  $\Pi_i \in \mathbb{S}_+^A$  for each  $i \in \{1, \dots, n\}$ , satisfying  $\sum_{i=1}^n \Pi_i = I$ . This sequence of operators is also called a *positive operator valued measure* or a POVM in the literature. If we have some qubits in state  $\rho$  and we apply the measurement  $(\Pi_1, \dots, \Pi_n)$  (or “observe the qubits according to the measurement”), we obtain *outcome* “ $i$ ” with probability  $\langle \Pi_i, \rho \rangle$ . The definitions of density matrices and measurements establish  $(\langle \Pi_i, \rho \rangle)$  as a well-defined probability distribution over the indices. The alteration of state resulting from a measurement is referred to as a *collapse*. Due to this restricted kind of access, in general only a limited amount of classical information may be extracted from a given quantum state.

For example, if we apply the measurement  $\{\Pi_0 := e_0e_0^*, \Pi_1 := e_1e_1^*\}$  to the state  $e_+e_+^*$ , we obtain the outcomes:

$$\begin{cases} \text{“0”} & \text{with probability } \langle \Pi_0, e_+e_+^* \rangle = 1/2, \\ \text{“1”} & \text{with probability } \langle \Pi_1, e_+e_+^* \rangle = 1/2. \end{cases}$$

**Multiple quantum systems.** For convenience, we refer to a quantum system with state space  $\mathbb{C}^A$  by the index set  $A$ . Suppose we have two quantum systems  $A_1, A_2$  that are independently in pure states  $\psi_1 \in \mathbb{C}^{A_1}$  and  $\psi_2 \in \mathbb{C}^{A_2}$ . Their combined state is  $\psi_1 \otimes \psi_2 \in \mathbb{C}^{A_1} \otimes \mathbb{C}^{A_2} \cong \mathbb{C}^{A_1 \times A_2}$  where  $\otimes$  denotes the Kronecker (or tensor) product. Note that the Kronecker product has the property that  $\|x \otimes y\|_2 = \|x\|_2 \|y\|_2$  so unit norm is preserved. It is not always possible to decompose a vector in  $\mathbb{C}^{A_1} \otimes \mathbb{C}^{A_2}$  as a Kronecker product of vectors in  $\mathbb{C}^{A_1}$  and  $\mathbb{C}^{A_2}$ ; a state with this property is said to be *entangled*. For example, the state  $\Phi^+ = [1/\sqrt{2}, 0, 0, 1/\sqrt{2}]^T$  is entangled; it cannot be expressed as  $\psi_1 \otimes \psi_2$  for any choice of  $\psi_1, \psi_2 \in \mathbb{C}^2$ .

These concepts extend to mixed states as well. If two disjoint quantum systems are independently in states  $\rho_1 \in \mathbb{S}_+^{A_1}$  and  $\rho_2 \in \mathbb{S}_+^{A_2}$ , then the joint state of the combined system is the density matrix  $\rho_1 \otimes \rho_2 \in \mathbb{S}_+^{A_1 \times A_2}$ . We make use of the properties that Kronecker products preserve positive semidefiniteness and that  $\text{Tr}(A \otimes B) = \text{Tr}(A) \text{Tr}(B)$ . It is not always possible to write a density matrix  $\rho \in \mathbb{S}_+^{A_1 \times A_2}$  as  $\rho_1 \otimes \rho_2$  where  $\rho_1 \in \mathbb{S}_+^{A_1}$  and  $\rho_2 \in \mathbb{S}_+^{A_2}$ , or more generally, as a convex combination of such Kronecker products. In the latter case, the state is said to be *entangled*, and otherwise, it is said to be *unentangled*.

We typically consider systems consisting of two-dimensional particles (i.e., qubits), but it is sometimes convenient to work with higher dimensional particles. Since higher dimensional spaces may be viewed as subspaces of suitable tensor powers of  $\mathbb{C}^2$ , we continue to describe such systems in terms of qubits.

**Partial trace.** The *partial trace over  $A_1$*  is the unique linear transformation  $\text{Tr}_{A_1} : \mathbb{S}^{A_1 \times A_2} \rightarrow \mathbb{S}^{A_2}$ , which satisfies

$$\text{Tr}_{A_1}(\rho_1 \otimes \rho_2) = \text{Tr}(\rho_1) \cdot \rho_2,$$

for all  $\rho_1 \in \mathbb{S}^{A_1}$  and  $\rho_2 \in \mathbb{S}^{A_2}$ . More explicitly, given any matrix  $X \in \mathbb{S}_+^{A_1 \times A_2}$  we define  $\text{Tr}_{A_1}$  as

$$\text{Tr}_{A_1}(X) := \sum_{x_1 \in A_1} (e_{x_1}^* \otimes \mathbf{I}_{A_2}) X (e_{x_1} \otimes \mathbf{I}_{A_2}),$$

where  $\{e_{x_1} : x_1 \in A_1\}$  is the standard basis for  $\mathbb{C}^{A_1}$ . In fact, the definition is independent of the choice of basis, so long as it is orthonormal. Note that the partial trace is positive, i.e.,  $\text{Tr}_{A_1}(X) \in \mathbb{S}_+^{A_2}$  when  $X \in \mathbb{S}_+^{A_1 \times A_2}$ , and also trace-preserving. (In fact, it is a *completely positive* operation.) This ensures that the image of any density matrix under this operation, called its *reduced state*, is a well-defined density matrix.

Consider the scenario where two parties, Alice and Bob, hold parts of a quantum system which are jointly in some state  $\rho$ , i.e., they “share” a quantum state  $\rho$  over the space  $\mathbb{C}^A \otimes \mathbb{C}^B$ . Then the partial trace of  $\rho$  over one space characterizes the quantum state over the remaining space (if we are interested only in operations on the latter space). For example,  $\text{Tr}_A(\rho)$  is the density matrix representing Bob’s half of the state and  $\text{Tr}_B(\rho)$  represents Alice’s half. Note that  $\rho$  may not equal  $\text{Tr}_B(\rho) \otimes \text{Tr}_A(\rho)$  in general.

Suppose we are given the density matrix  $\rho \in \mathbb{S}_+^A$ . We call the pure state  $\psi \in \mathbb{C}^A \otimes \mathbb{C}^B$  a *purification* of  $\rho$  if  $\text{Tr}_B(\psi\psi^*) = \rho$ . A purification always exists if  $|B| \geq |A|$ , and is in general not unique. An important property of purifications of the same state is that they are related to each other by a unitary operator: if  $\text{Tr}_B(\psi\psi^*) = \text{Tr}_B(\phi\phi^*)$ , then there exists a unitary  $U$  acting on  $\mathbb{C}^B$  alone such that  $\psi = (\mathbf{I}_A \otimes U)\phi$ .

The partial trace operation is the quantum analogue of calculating marginal probability distributions. Consider the linear operator  $\text{Tr}_A : \mathbb{R}^{A \times B} \rightarrow \mathbb{R}^B$  defined by

$$[\text{Tr}_A(v)]_y = \sum_{x \in A} v_{x,y} ,$$

for  $y \in B$ . This is called the partial trace over  $A$ . Note that  $\text{Tr}_A(p)$  gives the marginal distribution over  $B$  of the probability distribution  $p \in \text{Prob}^{A \times B}$ . One may view probability distributions as diagonal positive semidefinite matrices with unit trace. Then, taking the partial trace (as defined for quantum states) corresponds exactly to the computation of marginal distributions.

**Distance measures.** Notions of distance between quantum states and probability distributions are very important in quantum cryptography. Here, we discuss two measures used in this paper and how they are related.

We define the *fidelity* of two nonnegative vectors  $p, q \in \mathbb{R}_+^A$  as

$$F(p, q) := \left( \sum_{x \in A} \sqrt{p_x} \sqrt{q_x} \right)^2$$

and the fidelity of two positive semidefinite matrices  $\rho_1$  and  $\rho_2$  as

$$F(\rho_1, \rho_2) := \|\sqrt{\rho_1} \sqrt{\rho_2}\|_*^2.$$

Notice,  $F(\rho_1, \rho_2) \geq 0$  with equality if and only if  $\langle \rho_1, \rho_2 \rangle = 0$  and, if  $\rho_1$  and  $\rho_2$  are quantum states,  $F(\rho_1, \rho_2) \leq 1$  with equality if and only if  $\rho_1 = \rho_2$ . An analogous statement can be made for fidelity over probability vectors.

Fidelity has several useful properties, which we later use in this paper. We have occasion to consider fidelity only of probability distributions, and state the properties in terms of these. However, the following properties hold for quantum states as well. Fidelity is symmetric, positively homogeneous in both arguments, i.e.,  $\lambda F(p, q) = F(\lambda p, q) = F(p, \lambda q)$  for all  $\lambda > 0$ , and is concave, i.e.,  $F(\sum_{i=1}^n \lambda_i p_i, q) \geq \sum_{i=1}^n \lambda_i F(p_i, q)$ , for all  $\lambda \in \text{Prob}^n$ .

Another distance measure is the *trace distance*. We define the trace distance between two probability vectors  $p$  and  $q$ , denoted  $\Delta(p, q)$ , as

$$\Delta(p, q) := \frac{1}{2} \|p - q\|_1.$$

This is also commonly known as the total variation distance. We similarly define the trace distance between two quantum states  $\rho_1$  and  $\rho_2$  as

$$\Delta(\rho_1, \rho_2) := \frac{1}{2} \|\rho_1 - \rho_2\|_*.$$

Notice  $\Delta(\rho_1, \rho_2) \geq 0$  with equality if and only if  $\rho_1 = \rho_2$ , and  $\Delta(\rho_1, \rho_2) \leq 1$  with equality if and only if  $\langle \rho_1, \rho_2 \rangle = 0$ . An analogous statement can be made for the trace distance between probability vectors.

We now discuss two important notions in quantum cryptography. The first is how easily two states can be distinguished from each other. For example, if Alice gives to Bob one of two states  $\rho_1$  or  $\rho_2$  chosen uniformly at random, then Bob can measure to learn whether he has been given  $\rho_1$  or  $\rho_2$  with maximum probability

$$\frac{1}{2} + \frac{1}{4} \|\rho_1 - \rho_2\|_* = \frac{1}{2} + \frac{1}{2} \Delta(\rho_1, \rho_2).$$

The second notion is *quantum steering*. Suppose Alice has given to Bob the  $A_1$  part (i.e., the subsystem  $A_1$  of qubits) of  $\phi \in \mathbb{C}^{A_1 \times A_2}$ . Now suppose she wants to modify and send the  $A_2$  part in a way so as to convince Bob that a different state was sent, say  $\psi \in \mathbb{C}^{A_1 \times A_2}$ . Her most general strategy is to apply a quantum operation on  $A_2$  (i.e., a sequence of unitary operations and measurements) before sending it to Bob. If Bob measures according to the POVM  $(\psi\psi^*, I - \psi\psi^*)$ , Alice can convince him that the state is  $\psi$  with maximum probability

$$F(\text{Tr}_{A_2}(\psi\psi^*), \text{Tr}_{A_2}(\phi\phi^*)) .$$

### 3 Coin-flipping protocols

A strong coin-flipping protocol is a two-party quantum *communication protocol* in the style of Yao [30]. We concentrate on a class of communication protocols relevant to coin-flipping. Informally, in such protocols, two parties Alice and Bob hold some number of qubits; the qubits with each party are initialized to a fixed pure state. The initial joint state is therefore unentangled across Alice and Bob. The two parties then “play” in turns. Suppose it is Alice’s turn to play. Alice applies a unitary transformation on her qubits and then sends one or more qubits to Bob. Sending qubits does not change the overall superposition, but rather changes the ownership of the qubits. This allows Bob to apply his next unitary transformation on the newly received qubits. At the end of the protocol, each player makes a measurement of their qubits and announces the outcome as the result of the protocol.

Formally, the players Alice and Bob, hold some number of qubits, which initially factor into a tensor product  $\mathbb{C}^{A_0} \otimes \mathbb{C}^{B_0}$  of Hilbert spaces. The qubits corresponding to  $\mathbb{C}^{A_0}$  are in Alice’s possession, and those in  $\mathbb{C}^{B_0}$  are in Bob’s possession. When the protocol starts, the qubits in  $\mathbb{C}^{A_0}$  are initialized to some superposition  $\psi_{A,0}$  and those in  $\mathbb{C}^{B_0}$  to  $\psi_{B,0}$ , both of which specified by the protocol. The communication consists of  $t \geq 1$  alternations of message exchange (“rounds”), in which the two players “play”. Either party may play first. The protocol specifies a factorization of the joint state space just before each round, corresponding to the ownership of the qubits. In the  $i$ th round,  $i \geq 1$ , suppose it is Alice’s turn to play. Suppose the factorization of the state space just before the  $i$ th round is  $\mathbb{C}^{A_{i-1}} \otimes \mathbb{C}^{B_{i-1}}$ . Alice applies a unitary operator  $U_{A,i}$  to the qubits in  $\mathbb{C}^{A_{i-1}}$ . Then, Alice sends some of her qubits to Bob. Formally, the space  $\mathbb{C}^{A_{i-1}}$  is expressed as  $\mathbb{C}^{A_i} \otimes \mathbb{C}^{M_i}$ , where  $\mathbb{C}^{A_i}$  is Alice’s state space after the  $i$ th message is sent and  $\mathbb{C}^{M_i}$  is the state space for the  $i$ th message. Consequently, Bob’s state space after receiving the  $i$ th message is  $\mathbb{C}^{B_i} = \mathbb{C}^{M_i} \otimes \mathbb{C}^{B_{i-1}}$ . In the next round, Bob may thus apply a unitary operation to the qubits previously in Alice’s control.

At the end of the  $t$  rounds of play, Alice and Bob observe the qubits in their possession according to some measurement. The outcomes of these measurements represent their outputs. We emphasize that there are no measurements until all rounds of communication are completed. A protocol with intermedi-

ate measurements may be transformed into this form by appealing to standard techniques [5].

**Definition 1 (Strong coin-flipping)** A *strong coin-flipping protocol* is a two-party communication protocol as described above, in which the measurements of Alice and Bob are given by respective POVMs  $(\Pi_{A,0}, \Pi_{A,1}, \Pi_{A,\text{abort}})$  and  $(\Pi_{B,0}, \Pi_{B,1}, \Pi_{B,\text{abort}})$ . When both parties follow the protocol, they do not abort, i.e., only get outcomes in  $\{0, 1\}$ . Further, each party outputs the same bit  $c \in \{0, 1\}$  and each binary outcome occurs with probability  $1/2$ .

We are interested in the probabilities of the different outcomes in a coin-flipping protocol, when either party “cheats”. Suppose Alice and Bob have agreed upon a protocol, i.e., a set of rules for the state initialization, communication, quantum operations, and measurements. What if Alice or Bob do not follow protocol? Suppose Alice is dishonest and would like to force an outcome of 0. She may use a different number of qubits for her private operations, so that her space  $\mathbb{C}^{A'_i}$  may be much larger than  $\mathbb{C}^{A_i}$ . She may create any initial state she wants. During the communication, the only restriction is that she send a state of the correct dimension, e.g., if the protocol requires a message with 3 qubits in the first message, then Alice sends 3 qubits. Between messages, she may apply any quantum operation she wants on the qubits in her possession. At the end of the protocol, she may use a different measurement of her choice. For example, she may simply output “0” as this is her desired outcome (which corresponds to a trivial measurement). The rules that Alice chooses to follow instead of the protocol constitute a *cheating strategy*.

We would like to quantify the extent to which a cheating player can convince an honest one of a desired outcome, so we focus on runs of the protocol in which at most one party is dishonest. We analyze in this paper the maximum probability with which Alice (or Bob) can force a desired outcome in terms of the “bias”, i.e., the advantage over  $1/2$  that a cheating party can achieve.

**Definition 2 (Bias)** For a strong coin-flipping protocol, for each  $c \in \{0, 1\}$ , define

- $P_{A,c}^* := \sup \{\Pr[\text{honest Bob outputs } c \text{ when Alice may cheat}]\}$ ,
- $P_{B,c}^* := \sup \{\Pr[\text{honest Alice outputs } c \text{ when Bob may cheat}]\}$ ,

where the suprema are taken over all cheating strategies of the dishonest player. The bias  $\epsilon$  of the protocol is defined as

$$\epsilon := \max\{P_{A,0}^*, P_{A,1}^*, P_{B,0}^*, P_{B,1}^*\} - 1/2 .$$

In the Supplemental Material we work out the cheating probabilities of an example protocol.

**A family of protocols.** We now consider a family of protocols which generalizes the above idea. Alice and Bob each flip a coin and commit to their respective bits by exchanging quantum states. Then they reveal their bits and send the remaining part of the commitment state. Each party checks

the received state against the one they expect, and abort the protocol if they detect an inconsistency. They output the XOR of the two bits otherwise. We see that this is uniformly random, when  $a$  and  $b$  are uniformly random.

The difficulty in designing a good protocol is in deciding how Alice and Bob commit to their bits. If Alice or Bob leaks too much information early, then the other party has more freedom to cheat. Thus, we try to maintain a balance between the two parties so as to minimize the bias they can achieve by cheating.

Consider the following Cartesian product of finite sets  $A = A_1 \times \cdots \times A_n$ . These are used for Alice's first  $n$  messages to Bob. Suppose we are given two probability distributions  $\alpha_0, \alpha_1 \in \text{Prob}^A$ . Define the following two quantum states

$$\psi_a = \sum_{x \in A} \sqrt{\alpha_{a,x}} e_x \otimes e_x \in \mathbb{C}^A \otimes \mathbb{C}^{A'} \quad \text{for } a \in \{0, 1\},$$

where  $A' = A$ . The reason we define the state over  $\mathbb{C}^A$  and a copy is because in the protocol, Alice sends states in  $\mathbb{C}^A$  while retaining copies in  $\mathbb{C}^{A'}$  for herself. We may simulate Alice's choice of uniformly random  $a$  and the corresponding messages by preparing the initial state

$$\psi := \sum_{a \in \{0,1\}} \frac{1}{\sqrt{2}} e_a \otimes e_a \otimes \psi_a \in \mathbb{C}^{A_0} \otimes \mathbb{C}^{A'_0} \otimes \mathbb{C}^A \otimes \mathbb{C}^{A'},$$

where  $A_0 = A'_0 = \{0, 1\}$  are used for two copies of Alice's bit  $a$ , one for Bob and a copy for herself.

We now describe the setting for Bob's messages. Consider the following Cartesian product of finite sets  $B = B_1 \times \cdots \times B_n$  used for Bob's first  $n$  messages to Alice. Suppose we are given probability distributions  $\beta_0, \beta_1 \in \text{Prob}^B$ . Define the following two quantum states

$$\phi_b = \sum_{y \in B} \sqrt{\beta_{b,y}} e_y \otimes e_y \in \mathbb{C}^B \otimes \mathbb{C}^{B'} \quad \text{for } b \in \{0, 1\},$$

where  $B' = B$ . Bob's choice of uniformly random  $b$ , and the corresponding messages may be simulated by preparing the initial state

$$\phi := \sum_{b \in \{0,1\}} \frac{1}{\sqrt{2}} e_b \otimes e_b \otimes \phi_b \in \mathbb{C}^{B_0} \otimes \mathbb{C}^{B'_0} \otimes \mathbb{C}^B \otimes \mathbb{C}^{B'},$$

where  $B_0 = B'_0 = \{0, 1\}$  are used for two copies of Bob's bit  $b$ , one for Alice and a copy for himself.

We now describe the communication and cheat detection in the protocol.

**Definition 3 (Coin-flipping protocol based on bit-commitment)** A *coin-flipping protocol based on bit-commitment* is specified by a 4-tuple of probability distributions  $(\alpha_0, \alpha_1, \beta_0, \beta_1)$  that define states  $\psi, \phi$  as above.

- Alice prepares the state  $\psi$  and Bob prepares the state  $\phi$  as defined above.
- For  $i$  from 1 to  $n$ : Alice sends  $\mathbb{C}^{A_i}$  to Bob who replies with  $\mathbb{C}^{B_i}$ .



- Alice fully reveals her bit by sending  $\mathbb{C}^{A'_0}$ . She also sends  $\mathbb{C}^{A'}$  which Bob uses later to check if she was honest. Bob then reveals his bit by sending  $\mathbb{C}^{B'_0}$ . He also sends  $\mathbb{C}^{B'}$  which Alice uses later to check if he was honest.
- Alice observes the qubits in her possession according to the measurement  $(\Pi_{A,0}, \Pi_{A,1}, \Pi_{A,\text{abort}})$  defined on the space  $\mathbb{S}_+^{A_0 \times B'_0 \times B \times B'}$ , where

$$\Pi_{A,0} := \sum_{b \in \{0,1\}} e_b e_b^* \otimes e_b e_b^* \otimes \phi_b \phi_b^*, \quad \Pi_{A,1} := \sum_{b \in \{0,1\}} e_{\bar{b}} e_{\bar{b}}^* \otimes e_b e_b^* \otimes \phi_b \phi_b^*,$$

$$\text{and } \Pi_{A,\text{abort}} := \mathbf{I} - \Pi_{A,0} - \Pi_{A,1}.$$

- Bob observes the qubits in his possession according to the measurement  $(\Pi_{B,0}, \Pi_{B,1}, \Pi_{B,\text{abort}})$  defined on the space  $\mathbb{S}_+^{B_0 \times A'_0 \times A \times A'}$ , where

$$\Pi_{B,0} := \sum_{a \in \{0,1\}} e_a e_a^* \otimes e_a e_a^* \otimes \psi_a \psi_a^*, \quad \Pi_{B,1} := \sum_{a \in \{0,1\}} e_{\bar{a}} e_{\bar{a}}^* \otimes e_a e_a^* \otimes \psi_a \psi_a^*,$$

$$\text{and } \Pi_{B,\text{abort}} := \mathbf{I} - \Pi_{B,0} - \Pi_{B,1}. \text{ (These last two steps can be interchanged.)}$$

Note that the measurements check two things. First, they check whether the outcome is 0 or 1. The first two terms determine this, i.e., whether  $a = b$  or if  $a \neq b$ . Second, they check whether the other party was honest. For example, if Alice's measurement projects onto a subspace where  $b = 0$  and Bob's messages are not in state  $\phi_0$ , then Alice knows Bob has cheated and aborts.

Notice that our protocol is parameterized by the four probability distributions  $\alpha_0$ ,  $\alpha_1$ ,  $\beta_0$ , and  $\beta_1$ . It seems to be a very difficult problem to solve for the choice of these parameters that gives us the least bias. Indeed, we do not even have an upper bound on the dimension of these parameters in an optimal protocol. However, we can solve for the bias of a protocol once these parameters are fixed using the optimization techniques in Section 4. Once we have a means for computing the bias given some choice of fixed parameters, we then turn our attention to solving for the best choice of parameters. We use the heuristics in Subsections 5.1 and 5.2 to design an algorithm in Section 6 to search for these.

#### 4 Cheating strategies as semidefinite programs

In this section, we show that the optimal cheating strategy of a player in a coin-flipping protocol is characterized by highly structured semidefinite programs.

**Definition 4** We define *Bob's cheating polytope*, denoted as  $\mathcal{P}_B$ , as the set of all vectors  $(p_1, p_2, \dots, p_n)$  such that

$$\begin{aligned} \text{Tr}_{B_1}(p_1) &= e_{A_1}, \\ \text{Tr}_{B_2}(p_2) &= p_1 \otimes e_{A_2}, \\ &\vdots \\ \text{Tr}_{B_n}(p_n) &= p_{n-1} \otimes e_{A_n}, \\ p_j &\in \mathbb{R}_+^{A_1 \times B_1 \times \dots \times A_j \times B_j}, \text{ for all } j \in \{1, \dots, n\}, \end{aligned}$$

where  $e_{A_j}$  denotes the vector of all ones on the corresponding space  $\mathbb{C}^{A_j}$ .

In the Supplemental Material, we work out the cheating probabilities in the manner of Kitaev [11]. We now define simpler “reduced” problems that capture Bob’s optimal cheating probabilities.

**Theorem 1 (Bob’s Reduced Problems)** *The maximum probability with which cheating Bob can force honest Alice to accept outcome  $c \in \{0, 1\}$  is given by the optimal objective function value of the following convex optimization problem*

$$P_{B,c}^* = \max \left\{ \frac{1}{2} \sum_{a \in \{0,1\}} F((\alpha_a \otimes \mathbf{I}_B)^T p_n, \beta_{a \oplus c}) : (p_1, \dots, p_n) \in \mathcal{P}_B \right\},$$

where the arguments of the fidelity functions are probability vectors over  $B$ .

A proof of the above theorem is presented in the Supplemental Material. The connection between the fidelity function and semidefinite programming is detailed later in this section.

We can also define Alice’s cheating polytope.

**Definition 5** We define *Alice’s cheating polytope*, denoted as  $\mathcal{P}_A$ , as the set of all vectors  $(s_1, s_2, \dots, s_n, s)$  satisfying

$$\begin{aligned} \text{Tr}_{A_1}(s_1) &= 1, \\ \text{Tr}_{A_2}(s_2) &= s_1 \otimes e_{B_1}, \\ &\vdots \\ \text{Tr}_{A_n}(s_n) &= s_{n-1} \otimes e_{B_{n-1}}, \\ \text{Tr}_{A'_0}(s) &= s_n \otimes e_{B_n}, \\ s_1 &\in \mathbb{R}_+^{A_1}, \\ s_j &\in \mathbb{R}_+^{A_1 \times B_1 \times \dots \times B_{j-1} \times A_j}, \text{ for all } j \in \{2, \dots, n\}, \\ s &\in \mathbb{R}_+^{A'_0 \times A \times B}, \end{aligned}$$

where  $e_{B_j}$  denotes the vector of all ones on the corresponding space  $\mathbb{C}^{B_j}$ .

Now we can define Alice’s reduced problems.

**Theorem 2 (Alice’s Reduced Problems)** *The maximum probability with which cheating Alice can force honest Bob to accept outcome  $c \in \{0, 1\}$  is given by the optimal objective function value of the following convex optimization problem*

$$P_{A,c}^* = \max \left\{ \frac{1}{2} \sum_{a \in \{0,1\}} \sum_{y \in B} \beta_{a \oplus c, y} F(s^{(a,y)}, \alpha_a) : (s_1, \dots, s_n, s) \in \mathcal{P}_A \right\},$$

where  $s^{(a,y)} \in \mathbb{R}_+^A$  is the restriction of  $s$  with the indices  $(a, y)$  fixed, i.e.,  $[s^{(a,y)}]_x := s_{a,x,y}$ .

We present a proof of the above theorem in the Supplemental Material.

We point out that the reduced problems are also semidefinite programs. The containment of the variables in a polytope is captured by linear constraints, so it suffices to express the objective function as a linear functional of an appropriately defined positive semidefinite matrix variable.

**Lemma 3** For any  $p, q \in \mathbb{R}_+^A$ , we have

$$F(p, q) = \max \left\{ \left\langle X, \sqrt{p}\sqrt{p}^T \right\rangle : \text{diag}(X) = q, X \in \mathbb{S}_+^A \right\}.$$

*Proof* Notice that  $\bar{X} := \sqrt{q}\sqrt{q}^T$  is a feasible solution to the SDP with objective function value  $F(p, q)$ . All that remains to show is that it is an optimal solution. If  $p = 0$ , then we are done, so assume  $p \neq 0$ . The dual can be written as

$$\inf \{ \langle y, q \rangle : \text{Diag}(y) \succeq \sqrt{p}\sqrt{p}^T, y \in \mathbb{R}^A \}.$$

Define  $y$ , as a function of  $\varepsilon > 0$ , entry-wise for each  $x \in A$  as

$$y_x(\varepsilon) := \begin{cases} (\sqrt{F(p, q)} + \varepsilon) \frac{\sqrt{p_x}}{\sqrt{q_x}} & \text{if } p_x, q_x > 0, \\ \frac{(\sqrt{F(p, q)} + \varepsilon) \|p\|_1}{\varepsilon} & \text{if } q_x = 0, \\ \varepsilon & \text{if } p_x = 0, q_x > 0. \end{cases}$$

We can check that  $\langle y(\varepsilon), q \rangle \rightarrow F(p, q)$  as  $\varepsilon \rightarrow 0$ , so it suffices to show that  $y(\varepsilon)$  is dual feasible for all  $\varepsilon > 0$ . For any  $y > 0$ ,

$$\begin{aligned} \text{Diag}(y) \succeq \sqrt{p}\sqrt{p}^T &\iff I_A \succeq \text{Diag}(y)^{-1/2} \sqrt{p}\sqrt{p}^T \text{Diag}(y)^{-1/2} \\ &\iff 1 \geq \sqrt{p}^T \text{Diag}(y)^{-1} \sqrt{p} \\ &\iff 1 \geq \sum_{x \in A} \frac{p_x}{y_x}, \end{aligned}$$

noting  $\text{Diag}(y)^{-1/2} \sqrt{p}\sqrt{p}^T \text{Diag}(y)^{-1/2}$  is rank 1 so the largest eigenvalue is equal to its trace. From this, we can check that  $y(\varepsilon)$  is feasible for all  $\varepsilon > 0$ .  $\square$

The optimization problem in Lemma 3 remains an SDP if we replace  $q$  with a variable constrained to be in a polytope. Therefore, the reduced problems in Theorems 1 and 2 can be modelled as semidefinite programs.

In the Supplemental Material, we discuss how the reduced problems can be modelled as *second-order cone programs* and present numerical tests comparing them to the SDP formulations. The numerical experiments suggest that the current software is more robust in solving the SDP formulations for typical data in our search algorithm, so we use the SDP formulations. However, the second-order cone program formulations may be of independent interest so we include them in the Supplemental Material.

## 5 Speeding up the search

In this section, we introduce heuristics which dramatically speed up our search algorithm.

### 5.1 Protocol filter

In this subsection, we describe ways to bound the optimal cheating probabilities from below by finding feasible solutions to Alice and Bob's reduced cheating problems. In the search for parameters that lead to the lowest bias, our algorithm tests many protocols. The idea is to devise simple tests to check whether a protocol is a good candidate for being optimal. For example, suppose we can quickly compute the success probability of a certain cheating strategy for Bob. If this strategy succeeds with too high a probability for a given set of parameters, then we can rule out these parameters as being good choices. This saves the time it would have taken to solve the SDPs (or SOCPs).

We illustrate this idea using the Kitaev lower bound below.

**Theorem 3** ([11], [9]) *For any coin-flipping protocol, we have*

$$P_{A,0}^* P_{B,0}^* \geq \frac{1}{2} \quad \text{and} \quad P_{A,1}^* P_{B,1}^* \geq \frac{1}{2}.$$

Suppose that we find that  $P_{A,0}^* \approx 1/2$ , that is, the protocol is very secure against dishonest Alice cheating towards 0. Then, from the Kitaev bound, we infer that  $P_{B,0} \approx 1$  and the protocol is highly insecure against cheating Bob. Therefore, we can avoid solving for  $P_{B,0}^*$ .

The remainder of this section is divided according to the party that is dishonest. For each party, we discuss general cheating strategies and then also for the special cases of four and six-round protocols.

We now present a theorem which captures some of Alice's cheating strategies. The proof of this theorem and the similar theorem for cheating Bob can be found in the Supplemental Material.

**Theorem 4** *For a protocol parameterized by  $\alpha_0, \alpha_1 \in \text{Prob}^A$ ,  $\beta_0, \beta_1 \in \text{Prob}^B$ , we can bound Alice's optimal cheating probability as follows:*

$$P_{A,0}^* \geq \frac{1}{2} \sum_{y \in B} \text{conc} \{ \beta_{a,y} F(\cdot, \alpha_a) : a \in \{0, 1\} \} (v) \quad (1)$$

$$\geq \frac{1}{2} \lambda_{\max} \left( \eta \sqrt{\alpha_0} \sqrt{\alpha_0}^T + \tau \sqrt{\alpha_1} \sqrt{\alpha_1}^T \right) \quad (2)$$

$$\geq \left( \frac{1}{2} + \frac{1}{2} \sqrt{F(\alpha_0, \alpha_1)} \right) \left( \frac{1}{2} + \frac{1}{2} \Delta(\beta_0, \beta_1) \right), \quad (3)$$

where

$$\eta := \sum_{\substack{y \in B: \\ \beta_{0,y} \geq \beta_{1,y}}} \beta_{0,y} \quad \text{and} \quad \tau := \sum_{\substack{y \in B: \\ \beta_{0,y} < \beta_{1,y}}} \beta_{1,y},$$

and  $\sqrt{v}$  is the normalized principal eigenvector of  $\eta \sqrt{\alpha_0} \sqrt{\alpha_0}^T + \tau \sqrt{\alpha_1} \sqrt{\alpha_1}^T$ .

Furthermore, in a six-round protocol, we have

$$P_{A,0}^* \geq \frac{1}{2} \lambda_{\max} \left( \eta' \sqrt{\text{Tr}_{A_2}(\alpha_0)} \sqrt{\text{Tr}_{A_2}(\alpha_0)}^T + \tau' \sqrt{\text{Tr}_{A_2}(\alpha_1)} \sqrt{\text{Tr}_{A_2}(\alpha_1)}^T \right) \quad (4)$$

$$\geq \left( \frac{1}{2} + \frac{1}{2} \sqrt{\text{F}(\text{Tr}_{A_2}(\alpha_0), \text{Tr}_{A_2}(\alpha_1))} \right) \left( \frac{1}{2} + \frac{1}{2} \Delta(\text{Tr}_{B_2}(\beta_0), \text{Tr}_{B_2}(\beta_1)) \right) \quad (5)$$

where

$$\eta' := \sum_{\substack{y_1 \in B_1: \\ [\text{Tr}_{B_2}(\beta_0)]_{y_1} \geq [\text{Tr}_{B_2}(\beta_1)]_{y_1}}} [\text{Tr}_{B_2}(\beta_0)]_{y_1} \quad \text{and} \quad \tau' := \sum_{\substack{y_1 \in B_1: \\ [\text{Tr}_{B_2}(\beta_0)]_{y_1} < [\text{Tr}_{B_2}(\beta_1)]_{y_1}}} [\text{Tr}_{B_2}(\beta_1)]_{y_1}.$$

We have analogous bounds for  $P_{A,1}^*$ , which are obtained by switching the roles of  $\beta_0$  and  $\beta_1$  in the above expressions.

We call (1) Alice's *improved eigenstrategy*, (2) her *eigenstrategy*, and (3) her *three-round strategy*. For six-round protocols, we call (4) Alice's *eigenstrategy* and (5) her *measuring strategy*.

Note that only the improved eigenstrategy is affected by switching  $\beta_0$  and  $\beta_1$  (as long as we are willing to accept a slight modification to how we break ties in the definitions of  $\eta, \eta', \tau$ , and  $\tau'$ ).

We turn to strategies for a dishonest Bob.

**Theorem 5** For a protocol parameterized by  $\alpha_0, \alpha_1 \in \text{Prob}^A$ ,  $\beta_0, \beta_1 \in \text{Prob}^B$ , we can bound Bob's optimal cheating probability as follows:

$$P_{B,0}^* \geq \frac{1}{2} + \frac{1}{2} \sqrt{\text{F}(\beta_0, \beta_1)}, \quad (6)$$

and

$$P_{B,0}^* \geq \frac{1}{2} + \frac{1}{2} \Delta(\text{Tr}_{A_2 \times \dots \times A_n}(\alpha_0), \text{Tr}_{A_2 \times \dots \times A_n}(\alpha_1)). \quad (7)$$

In a four-round protocol, we have

$$P_{B,0}^* \geq \frac{1}{2} \sum_{a \in \{0,1\}} \text{F} \left( \sum_{x \in A} \alpha_{a,x} v_x, \beta_a \right) \quad (8)$$

$$\begin{aligned} &\geq \frac{1}{2} \sum_{x \in A} \lambda_{\max} \left( \sum_{a \in \{0,1\}} \alpha_{a,x} \sqrt{\beta_a} \sqrt{\beta_a}^T \right) \quad (9) \\ &\geq \max \left\{ \frac{1}{2} + \frac{1}{2} \Delta(\alpha_0, \alpha_1), \frac{1}{2} + \frac{1}{2} \sqrt{\text{F}(\beta_0, \beta_1)} \right\}, \end{aligned}$$

where  $\sqrt{v_x}$  is the normalized principal eigenvector of  $\sum_{a \in \{0,1\}} \alpha_{a,x} \sqrt{\beta_a} \sqrt{\beta_a}^T$ .

In a six-round protocol, we have

$$P_{B,0}^* \geq \frac{1}{2} \sum_{a \in A'_0} F \left( \sum_{x \in A} \alpha_{a,x} \tilde{p}_2^{(x)}, \beta_a \right) \quad (10)$$

$$\geq \frac{1}{2} \lambda_{\max} \left( \kappa \sqrt{\text{Tr}_{B_2}(\beta_0)} \sqrt{\text{Tr}_{B_2}(\beta_0)}^T + \zeta \sqrt{\text{Tr}_{B_2}(\beta_1)} \sqrt{\text{Tr}_{B_2}(\beta_1)}^T \right) \quad (11)$$

$$\geq \left( \frac{1}{2} + \frac{1}{2} \sqrt{F(\text{Tr}_{B_2}(\beta_0), \text{Tr}_{B_2}(\beta_1))} \right) \left( \frac{1}{2} + \frac{1}{2} \Delta(\alpha_0, \alpha_1) \right), \quad (12)$$

where

$$[\tilde{p}_2^{(x)}]_{y_1, y_2} := \begin{cases} c_{y_1} \frac{\beta_{g(x), y_1, y_2}}{[\text{Tr}_{B_2}(\beta_{g(x)})]_{y_1}} & \text{if } [\text{Tr}_{B_2}(\beta_{g(x)})]_{y_1} > 0, \\ c_{y_1} \frac{1}{|B_2|} & \text{if } [\text{Tr}_{B_2}(\beta_{g(x)})]_{y_1} = 0, \end{cases}$$

$$\kappa = \sum_{\substack{x \in A: \\ \alpha_{0,x} \geq \alpha_{1,x}}} \alpha_{0,x}, \quad \zeta = \sum_{\substack{x \in A: \\ \alpha_{0,x} < \alpha_{1,x}}} \alpha_{1,x}, \quad g(x) = \arg \max_a \{\alpha_{a,x}\},$$

and  $\sqrt{c}$  is the normalized principal eigenvector of

$$\frac{1}{2} \lambda_{\max} \left( \kappa \sqrt{\text{Tr}_{B_2}(\beta_0)} \sqrt{\text{Tr}_{B_2}(\beta_0)}^T + \zeta \sqrt{\text{Tr}_{B_2}(\beta_1)} \sqrt{\text{Tr}_{B_2}(\beta_1)}^T \right).$$

Furthermore, if  $|A_i| = |B_i|$  for all  $i \in \{1, \dots, n\}$ , then

$$P_{B,0}^* \geq \frac{1}{2} \sum_{a \in \{0,1\}} F(\alpha_a, \beta_a). \quad (13)$$

We get analogous lower bounds for  $P_{B,1}^*$  by switching the roles of  $\beta_0$  and  $\beta_1$  in the above expressions.

We call (6) Bob's *ignoring strategy* and (7) his *measuring strategy*. For four-round protocols, we call (8) Bob's *eigenstrategy* and (9) his *eigenstrategy lower bound*. For six-round protocols, we call (10) Bob's *six-round eigenstrategy*, (11) his *eigenstrategy lower bound*, and (12) his *three-round strategy*. We call (13) Bob's *returning strategy*.

Note that the only strategies that are affected by switching  $\beta_0$  and  $\beta_1$  are the eigenstrategy and the returning strategy.

Descriptions of the cheating strategies corresponding to the values in Theorems 4 and 5 can be found in the Supplemental Material.

## 5.2 Protocol symmetry

In this subsection, we discuss equivalence between protocols due to symmetry in the states used in them. Namely, we identify transformations on states under which the bias remains unchanged. This allows us to prune the search space of parameters needed to specify a protocol in the family under scrutiny. As a result, we significantly reduce the time required for our searches.

We first discuss *index symmetry*, that if we permute the elements of  $A_i$  or  $B_i$ , for any  $i \in \{1, \dots, n\}$ , then this does not change the bias of the protocol. It is easy to see from the reduced problems of Alice and Bob that any feasible solution can be permuted accordingly to accommodate for any (local) permutation in the indices of  $A_i$  or  $B_i$  while retaining the same objective function value. Thus, such a permutation does not decrease the bias.

We now identify a different kind of symmetry in the protocols, namely *symmetry between probability distributions*. If we simultaneously switch  $\alpha_0$  with  $\alpha_1$  and  $\beta_0$  with  $\beta_1$ , then it is obvious from the objective functions in Bob's reduced problems that his cheating probabilities do not change. We can claim the same thing for Alice noting that in Alice's reduced problems, the only constraints involving  $s^{(a,y)}$  can be written as

$$\sum_{a \in A'_0} s^{(a,y)} = s_n^{(y_1, \dots, y_{n-1})},$$

for each  $y = (y_1, \dots, y_{n-1}, y_n) \in B$ , which are symmetric about  $a$ . Also, switching  $\beta_0$  with  $\beta_1$  switches  $P_{B,0}^*$  with  $P_{B,1}^*$  and  $P_{A,0}^*$  with  $P_{A,1}^*$ . Thus, the bias does not decrease by independently switching  $\alpha_0$  with  $\alpha_1$  or  $\beta_0$  with  $\beta_1$ .

**The use of symmetry in the search algorithm.** Since we are able to switch the roles of  $\alpha_0$  and  $\alpha_1$ , we assume  $\alpha_0$  has the largest entry out of  $\alpha_0$  and  $\alpha_1$  and similarly that  $\beta_0$  has the largest entry out of  $\beta_0$  and  $\beta_1$ .

In four-round protocols, since we can permute the elements of  $A = A_1$ , we also assume  $\alpha_0$  has entries that are non-decreasing. This allows us to upper bound all the entries of  $\alpha_0$  and  $\alpha_1$  by the last entry in  $\alpha_0$ . We do this simultaneously for  $\beta_0$  and  $\beta_1$ .

In the six-round version, we need to be careful when applying the index symmetry, we cannot permute all of the entries in  $\alpha_0$ . The index symmetry only applies to local permutations so we only partially order them. We order  $A_2$  such that the entries  $\alpha_{0,\tilde{x}_1\tilde{x}_2}$  do not decrease for *one particular index*  $\tilde{x}_1 \in A_1$ . It is convenient to choose the index corresponding to the largest entry. Then we order the last block of entries in  $\alpha_0$  such that they do not decrease. Note that the last entry in  $\alpha_0$  is now the largest among all the entries in  $\alpha_0$  and  $\alpha_1$ . We do this simultaneously for  $\beta_0$  and  $\beta_1$ . Note that the search algorithm does not stop all symmetry; for example if  $\alpha_0$  and  $\alpha_1$  both have an entry of largest magnitude, we do not compare the second largest entries. But, as will be shown in the computational tests, we have a dramatic reduction in the number of protocols to be tested using the symmetry in the way described above.

## 6 Search algorithm

In this section, we develop an algorithm for finding coin-flipping protocols with small bias within our parametrized family.

To search for protocols, we first fix a local dimension  $d$  for the parameters

$$\alpha_0, \alpha_1, \beta_0, \beta_1 \in \text{Prob}^D,$$

where  $D := d$  for four-round protocols and  $D := d^2$  for six-round protocols. We then create a finite mesh over these parameters by creating a mesh over the entries in the probability vectors  $\alpha_0$ ,  $\alpha_1$ ,  $\beta_0$ , and  $\beta_1$ . We do so by increments of a precision parameter  $\nu \in (0, 1)$ . For example, we range over the values

$$\{0, \nu, 2\nu, \dots, 1 - \nu, 1\}$$

for  $[\alpha_0]_0$ , the first entry of  $\alpha_0$ . For the second entry of  $\alpha_0$ , we range over

$$\{0, \nu, 2\nu, \dots, 1 - [\alpha_0]_0\}$$

and so forth. Note that we only consider  $\nu = 1/N$  for some positive integer  $N$  so that we use the endpoints of the intervals.

This choice in creating the mesh makes it very easy to exploit the symmetry discussed in Subsection 5.2. We show computationally (in Section 7) that this symmetry helps by dramatically reducing the number of protocols to be tested. This is important since there are  $\binom{D+N-1}{N}^4$  protocols to test (before applying symmetry considerations).

Each point in this mesh is a set of candidate parameters for an optimal protocol. As described in Subsection 5.1, the protocol filter can be used to expedite the process of checking whether the protocol has high bias or is a good candidate for an optimal protocol. There are two things to be considered at this point which we now address.

First, we have to determine the order in which the cheating strategies in the protocol filter are applied. It is roughly the case that the computationally cheaper tasks give a looser lower bound to the optimal cheating probabilities. Therefore, we start with these easily computable probabilities, i.e. the probabilities involving norms and fidelities, then check the more computationally expensive tasks such as largest eigenvalues and calculating principal eigenvectors. We lastly solve the semidefinite programs. Another heuristic that we use is alternating between Alice and Bob's strategies. Many protocols with high bias seem to prefer either cheating Alice or cheating Bob. Having cheating strategies for both Alice and Bob early in the filter removes the possibility of checking many of Bob's strategies when it is clearly insecure concerning cheating Alice and vice versa. Starting with these heuristics, we then ran preliminary tests to see which order seemed to perform the best. The order (as well as the running times for the filter strategies) is shown in Tables 1 and 4 for the four-round version and for the six-round version, respectively.

Second, we need to determine a threshold for what constitutes a "high bias". If a filter strategy has success probability 0.9, do we eliminate this candidate protocol? The lower the threshold, the more quickly the filter eliminates protocols. However, if the threshold is too low, we may be too ambitious and not find any protocols. To determine a good threshold, consider the following protocol parameters

$$\alpha_0 = \frac{1}{2} [1, 0, 1]^T, \quad \alpha_1 = \frac{1}{2} [0, 1, 1]^T, \quad \beta_0 = [1, 0]^T, \quad \beta_1 = [0, 1]^T.$$



This is the four-round version of the optimal three-round protocol in [10]. Numerically solving for the cheating probabilities for this protocol shows that

$$P_{A,0}^* = P_{A,1}^* = P_{B,0}^* = P_{B,1}^* = 3/4.$$

Thus, there exists a protocol with the same bias as the best-known explicit coin-flipping protocol constructions. This suggests that we use a threshold around 0.75. Preliminary tests show that using a threshold of 0.75 or larger is much slower than a value of 0.7499. This is because using the larger threshold allows protocols with optimal cheating probabilities (or filter cheating probabilities) of 0.75 to slip through the filter and these protocols are no better than the one mentioned above (and many are just higher dimensional embeddings of it). Therefore, we use a threshold of 0.7499. (Tests using a threshold of slightly larger than 0.75 are considered in the Supplemental Material.)

Using these ideas, we now state the search algorithm.

#### Search Algorithm

- Fix a local dimension  $d$  and mesh precision  $\nu$ .
- For each protocol in the mesh (modulo the symmetry):
  - Use the Protocol Filter to eliminate protocols with bias above 0.2499.
  - Calculate the optimal cheating probabilities by solving the SDPs.
    - If any are larger than 0.7499, move on to the next protocol.
    - Else, output the protocol parameters with bias  $\epsilon < 1/4$ .

We test the algorithm on the cases of four and six-round protocols and for certain dimensions and precisions for the mesh. These are presented in detail next.

## 7 Numerical results

**Computational Platform.** We ran our search programs on Matlab, Version 7.12.0.635, on an SGI XE C1103 with 2x 3.2 GHz 4-core Intel X5672 x86 CPUs processor, and 10 GB memory, running Linux.

We solved the semidefinite programs using SeDuMi 1.3, a program for solving semidefinite programs in Matlab [24], [25].

Sample programs can be found at the following link:

<http://www.math.uwaterloo.ca/~anayak/coin-search/>

**Four-round search.** We list the filter cheating strategies in Table 1 which also give an estimate of how long it takes the program to compute the success probability for each strategy based on the average over 1000 random instances (i.e. four randomly chosen probability vectors  $\alpha_0$ ,  $\alpha_1$ ,  $\beta_0$ , and  $\beta_1$ .)

Notice the two strategies with codes F1 and F2 are special because they only involve two of the four probability distributions. Preliminary tests show that first generating  $\beta_0$  and  $\beta_1$  and checking with F1 is much faster than first generating  $\alpha_0$  and  $\alpha_1$  and checking with F2, even though F2 is much faster to compute.

Description or Equation	Comp. Time (s)	Code
(6)	0.000034429	F1
(7)	0.000004640	F2
(3)	0.000025980	F3
(13)	0.000023767	F4
(13) with $\beta_0, \beta_1$ switched	0.000018019	F5
(2)	0.000036613	F6
(9)	0.000073010	F7
(8)	0.000697611	F8
(8) with $\beta_0, \beta_1$ switched	0.000532954	F9
(1)	0.122971205	F10
(1) with $\beta_0, \beta_1$ switched	0.123375678	F11
$P_{A,0}^*$	0.149814373	SDPA0
$\frac{1}{2P_{A,0}^*}$	0.000000947	F12
$P_{B,0}^*$	0.070846378	SDPB0
$P_{A,1}^*$	0.149176117	SDPA1
$\frac{1}{2P_{A,1}^*}$	0.000000760	F13
$P_{B,1}^*$	0.070479449	SDPB1

**Table 1** Average running times for filter strategies for a 4-round protocol when  $d = 5$  over random protocol states. Equation references are to Theorems 4 and 5 and the codes (on the right) are used for reference in the numerical search experiments.

We can similarly justify the placement of  $P_{A,0}^*$  before  $P_{B,0}^*$  or  $P_{B,1}^*$ . The strategies F8 and F9 perform very well and the cheating probabilities are empirically very close to  $P_{B,0}^*$  and  $P_{B,1}^*$ . Thus, if a protocol gets through the F8 and F9 filter strategies, then it is likely that  $P_{B,0}^*$  and  $P_{B,1}^*$  are also less than 0.7499. This is why we place  $P_{A,0}^*$  first (although it will be shown that the order of solving the SDPs does not matter much).

We then give tables detailing how well the filter performs for four-round protocols, by counting the number of protocols that are *not* determined to have bias greater than 0.2499 by each prefix of cheating strategies. We test four-round protocols with message dimension  $d \in \{2, \dots, 9\}$  and precision  $\nu$  ranging up to  $1/2000$  (depending on  $d$ ). The summary of results for  $d \in \{2, 3, 4, 5\}$  can be found in Table 2 and the summary of results for  $d \in \{6, 7, 8, 9\}$  can be found in Table 3. Tables giving the exact numbers for the above searches and also for lower precision searches can be found in the Supplemental Material.

4-Rounds	$d = 2, \nu = \frac{1}{2000}$	$d = 3, \nu = \frac{1}{50}$	$d = 4, \nu = \frac{1}{30}$	$d = 5, \nu = \frac{1}{12}$
No. Protocols	1.60 e + 13	3.09 e + 12	8.86 e + 14	1.09 e + 13
Symmetry	1.00 e + 12	2.54 e + 10	7.36 e + 11	2.48 e + 09
F1	2.36 e + 10	1.02 e + 09	4.97 e + 10	5.67 e + 08
F2	1.76 e + 10	6.62 e + 08	2.77 e + 10	2.03 e + 08
F3	1.24 e + 02	4.41 e + 06	7.38 e + 08	1.77 e + 07
F4	0	2.02 e + 06	4.06 e + 08	1.36 e + 07
F5	0	2.00 e + 06	4.06 e + 08	1.36 e + 07
F6	0	1.76 e + 06	3.67 e + 08	1.31 e + 07
F7	0	1.15 e + 03	1.90 e + 05	4.34 e + 04
F8	0	0	0	0

**Table 2** A summary of the number of 4-round protocols that get past symmetry reductions and each strategy in the filter for  $d \in \{2, 3, 4, 5\}$  and the highest precision tested for each  $d$ .

4-Rounds	$d = 6, \nu = \frac{1}{12}$	$d = 7, \nu = \frac{1}{10}$	$d = 8, \nu = \frac{1}{9}$	$d = 9, \nu = \frac{1}{8}$
No. Protocols	1.46 e + 15	4.11 e + 15	1.71 e + 16	2.74 e + 16
Symmetry	4.61 e + 10	3.04 e + 10	4.23 e + 10	3.98 e + 10
F1	1.33 e + 10	1.09 e + 10	2.30 e + 10	2.48 e + 10
F2	3.84 e + 09	2.41 e + 09	4.41 e + 09	2.21 e + 09
F3	4.68 e + 08	4.49 e + 08	1.27 e + 09	4.32 e + 08
F4	3.90 e + 08	4.09 e + 08	1.20 e + 09	4.21 e + 08
F5	3.90 e + 08	4.09 e + 08	1.20 e + 09	4.21 e + 08
F6	3.77 e + 08	3.83 e + 08	1.15 e + 09	4.16 e + 08
F7	1.15 e + 06	1.80 e + 06	3.19 e + 06	1.80 e + 06
F8	0	0	0	0

**Table 3** A summary of the number of 4-round protocols that get past symmetry reductions and each strategy in the filter for  $d \in \{6, 7, 8, 9\}$  and the highest precision tested for each  $d$ .

**Observations on the four-round search.** We were able to search larger spaces than feasible with the SDP formulations alone. For example, suppose we took the  $2.74 \times 10^{16}$  protocols from the  $d = 9, \nu = 1/8$  search and checked to see if any of these had bias less than 0.7499 by solving only the reduced SDPs. Since each SDP takes at least 0.08 seconds to solve, this search would take at least 69 million years to finish. By applying the techniques in this paper, we were able to run this search in a matter of days.

We see that symmetry helped dramatically reduce the number of protocols that needed to be tested. In the largest search, we were able to cut the number of protocols down from  $2.74 \times 10^{16}$  to  $3.98 \times 10^{10}$ . F1 and F2 perform very well, together cutting down the number of protocols by a factor of about 10. An interesting observation is that F2 performs much better than F1, and is also 10 times faster to compute. It may seem better to put F2 before F1 in the tests, however, we place F1 first since it is beneficial to have the more expensive strategy being computed first. This way, it only needs to be computed for every choice of  $\beta_0$  and  $\beta_1$ . If we were to calculate F2 first, we would have to calculate F1 on every  $(\alpha_0, \alpha_1, \beta_0, \beta_1)$  for those  $\alpha_0, \alpha_1$  that F2 did not filter out. Being the first strategy to rely on all four probability distributions, F3

performs very well by reducing the number of protocols by another factor of 10. F4, F5, and F6 do not perform well (F5 being the same as F4 but with  $\beta_0$  swapped with  $\beta_1$ ); they cut down the number of protocols by a very small number. F7 and F8 perform so well that no SDPs were needed to be solved.

These numbers suggest a conjecture along the lines of

$$\min_{\alpha_0, \alpha_1, \beta_0, \beta_1 \in \text{Prob}^9} \max \{F1, \dots, F8\} \geq 0.7499.$$

However, in the Supplemental Material we give evidence that this may not be true if we replace 0.7499 with 0.75 by conducting “zoning-in searches” with much higher precisions.

**Six-round search.** We list the filter cheating strategies in Table 4 and give an estimate for how long it takes to compute the success probability for each strategy by taking the average over 1000 random instances. We then present a table showing how well the filter performs for six-round protocols with  $d = 2$  and  $\nu = 1/15$  and for  $d = 3$  and  $\nu = 1/4$ . The measure of performance of the filter that we use is as before. For each prefix of cheating strategies in the filter, we count the number of protocols in the mesh that are *not* determined to have bias greater than 0.2499 by that prefix.

Again, we choose which strategy to put first, G1 or G2. Preliminary tests show that placing G1 first results in a much faster search, similar to the four-round case. Even though G5 takes longer to compute than G6, tests show that it is better to have G5 first. We calculate  $P_{B,0}^*$  before  $P_{A,0}^*$  since G9 and G10 are close approximations of  $P_{A,0}^*$  and  $P_{A,1}^*$ , respectively. It will be evident that the order of solving the SDPs does not matter much.

We note here a few omissions as compared to the four-round tests. First, we have removed the two returning strategies, F4 and F5. These did not perform well in the four-round tests and preliminary tests show that they did not perform well in the six-round search either. Also, we do not have all the lower bounds for the eigenstrategies. Preliminary tests show that these lower bounds (omitted from the six-round search) take just as long or longer to compute than the eigenstrategies themselves, so there is no advantage in calculating them. Also, the marginal probabilities take approximately  $5.49 \times 10^{-6}$  seconds to compute which is negligible compared to the other times. Thus, we need not be concerned whether the strategies rely on the full probability distributions or marginal distributions.

The summary of results for  $d \in \{2, 3\}$  can be found in Table 5. Tables giving the exact numbers for the above searches and also for lower precision searches can be found in the Supplemental Material.

**Observations on the six-round search.** We first note that the filter does not work as effectively as in the four-round case. The six-round search for  $d = 3$  ran for about a month. In comparison, all the four-round searches ran in the matter of days.

The symmetry arguments cut down the number of protocols we need to examine significantly, by a factor of roughly 100. Note that in the four-round

Description or Equation	Comp. Time (s)	Code
(6)	0.000036128	G1
(7)	0.000005552	G2
(11)	0.000015667	G3
(5)	0.000028408	G4
(2)	0.000052325	G5
(4)	0.000044243	G6
(10)	0.000879119	G7
(10) with $\beta_0, \beta_1$ switched	0.000797106	G8
(1)	0.256377981	G9
(1) with $\beta_0, \beta_1$ switched	0.249946219	G10
$P_{B,0}^*$	0.164744870	SDPB0
$\frac{1}{2P_{B,0}^*}$	0.000000996	G11
$P_{A,0}^*$	0.276034548	SDPA0
$P_{B,1}^*$	0.162818974	SDPB1
$\frac{1}{2P_{B,1}^*}$	0.000001075	G12
$P_{A,1}^*$	0.271631913	SDPA1

**Table 4** Average running times for filter strategies in a 6-round protocol for  $d = 3$  over random protocol states. Equation references are to Theorems 4 and 5 and the codes (on the right) are used for reference in the numerical search experiments.

6-Rounds	$d = 2, \nu = \frac{1}{15}$	$d = 3, \nu = \frac{1}{4}$
No. Protocols	$4.43 \text{ e} + 11$	$6.00 \text{ e} + 10$
Symmetry	$9.37 \text{ e} + 09$	$2.79 \text{ e} + 08$
G1	$1.04 \text{ e} + 09$	$1.80 \text{ e} + 08$
G2	$8.77 \text{ e} + 08$	$8.61 \text{ e} + 07$
G3	$7.39 \text{ e} + 08$	$5.80 \text{ e} + 07$
G4	$3.50 \text{ e} + 08$	$3.07 \text{ e} + 07$
G5	$4.37 \text{ e} + 07$	$1.53 \text{ e} + 07$
G6	$4.31 \text{ e} + 07$	$1.53 \text{ e} + 07$
G7	$1.97 \text{ e} + 06$	$6.55 \text{ e} + 06$
G8	$4.79 \text{ e} + 05$	$5.44 \text{ e} + 06$
G9	$4.11 \text{ e} + 05$	$5.39 \text{ e} + 06$
G10	$3.86 \text{ e} + 05$	$5.39 \text{ e} + 06$
SDPB0	$5.94 \text{ e} + 02$	$2.40 \text{ e} + 04$
G11	$5.94 \text{ e} + 02$	$2.40 \text{ e} + 04$
SDPA0	0	0

**Table 5** A summary of the number of 6-round protocols that get past symmetry reductions and each strategy in the filter for  $d \in \{2, 3\}$  and the highest precision tested for each  $d$ .

case it was a factor of 1,000,000 (for the  $d = 9$  case). This can be explained by the weaker index symmetry in the six-round version.

Cheating strategy G1 cut the number of protocols down by a factor of 10 with G2 performing less well than the corresponding strategy in the four-round tests. G5 also performed well, but after this, G6 was not much help. G7 and G8 cut down the number of protocols by a factor of 10 each in the  $d = 2$  case, but not as much in the  $d = 3$  case. The next notable strategy was G10, being G9 with  $\beta_0$  and  $\beta_1$  swapped, which performed very poorly. It seems that the swapped strategies do not help much in the filters, that is, there is not much discrepancy between cheating towards 0 or 1. SDPB0 almost filtered out the rest of the protocols, relying on SDPA0 to stop the rest. The implicit strategy from Kitaev’s bound, G11, did not perform well after SDPB0 (note that it relies on SDPB0 so it is computed afterwards). Again, we notice that no protocols with bias less than 0.2499 were found.

We notice that G9 and G10, the improved eigenstrategies for Alice, hardly filter out any protocols, if any at all, in the low-precision tests (presented in the Supplemental Material). In these strategies, we compute a value on the concave hull  $\text{conc} \left\{ \frac{1}{2} \beta_{0,y} F(\cdot, \alpha_0), \frac{1}{2} \beta_{1,y} F(\cdot, \alpha_1) \right\}$ , for every value of  $y$ . In the eigenstrategy, we approximate the concave hull with the one of the two that has the larger constant. When we choose these constants according to a coarse mesh, e.g.,  $\nu = 1/3$  or  $\nu = 1/4$ , the one with the larger constant is a very good approximation of the concave hull. It appears that, we need finer precisions to bring out the power of this strategy in the filter.

In all of our searches, we did not find any protocols with bias less than 0.2499, and it seems that  $1/4$  might be the least bias achievable by the class of protocols we study. In the Supplemental Material, we perform two different kinds of searches to test this conjecture. First, we perform “random offset” searches to test protocols defined by possibly less structured parameters. That is, we test the filter strategies (and our conjecture above) by offsetting the mesh by a random constant. We give tables detailing the performance of the filter strategies and give further evidence that  $1/4$  is the least bias within this family of protocols. Second, we “zone-in” on protocols with bias exactly  $1/4$  to see if small perturbations allow a decrease in bias. Since we are searching over a smaller region, finer precisions can be used. We give tables detailing the performance of the filter strategies for these tests and note here that no tested perturbations allow a decrease in bias.

Inspired by the numerical results, in the Supplemental Material we prove a lower bound for four-round qubit protocols and discuss computer aided bounds on bias.

## 8 Conclusions

We introduced a parameterized family of quantum coin-flipping protocols based on bit-commitment, and formulated the cheating probabilities of Alice and Bob as simple semidefinite programs. Using these semidefinite programming formulations, we designed an algorithm to search for parameters yielding a protocol with small bias. We exploited symmetry and developed

cheating strategies to create a protocol filter so that a wider array of protocols can be searched. For example, without the heuristics used in this paper, it would have taken over 69 million years to search the same  $3 \times 10^{16}$  protocols that we tested.

Using the search algorithm, we searched four and six-round protocols from a mesh over the parameter space, with messages of varying dimension and with varying fineness for the mesh. After the systematic searches, no protocols having all four cheating probabilities less than 0.7499 were found.

An obvious open problem is to resolve the conjecture that all the protocols in the family we study have bias at least  $1/4$ . It seems the smallest bias does not decrease when the number of messages increases from four rounds to six. We conjecture the smallest bias does not decrease even if more messages are added. One way to show this is to find closed-form expressions for the optimal objective function values of the SDP formulations. This would be of great theoretical significance since very few highly interactive protocols (such as those examined in this paper) have been characterized by closed-form expressions for their bias or even by a description of optimal cheating strategies.

**Acknowledgements** We thank Andrew Childs, Michele Mosca, Peter Høyer, and John Watrous for their comments and suggestions. A.N.'s research was supported in part by NSERC Canada, CIFAR, an ERA (Ontario), QuantumWorks, and MITACS. A part of this work was completed at Perimeter Institute for Theoretical Physics. Perimeter Institute is supported in part by the Government of Canada through Industry Canada and by the Province of Ontario through MRI. J.S.'s research is supported by NSERC Canada, MITACS, ERA (Ontario), ANR project ANR-09-JCJC-0067-01, and ERC project QCC 306537. L.T.'s research is supported in part by Discovery Grants from NSERC.

Research at the Centre for Quantum Technologies at the National University of Singapore is partially funded by the Singapore Ministry of Education and the National Research Foundation, also through the Tier 3 Grant "Random numbers from quantum processes," (MOE2012-T3-1-009).

## References

1. Aharonov, D., Chailloux, A., Ganz, M., Kerenidis, I., Magnin, L.: A simpler proof of existence of quantum weak coin flipping with arbitrarily small bias (2013). Manuscript
2. Aharonov, D., Ta-Shma, A., Vazirani, U., Yao, A.C.C.: Quantum bit escrow. In: Proceedings of 32nd Annual ACM Symposium on the Theory of Computing, pp. 705–714. ACM (2000). DOI <http://doi.acm.org/10.1145/335305.335404>
3. Ambainis, A.: A new protocol and lower bounds for quantum coin flipping. In: Proceedings of 33rd Annual ACM Symposium on the Theory of Computing, pp. 134 – 142. ACM (2001). DOI <http://dx.doi.org/10.1109/FOCS.2004.13>
4. Bennett, C., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing, pp. 175–179. IEEE Computer Society (1984)
5. Bernstein, E., Vazirani, U.V.: Quantum complexity theory. *SIAM Journal on Computing* **26**(5), 1411–1473 (1997)
6. Blum, M.: Coin flipping by telephone. In: A. Gersho (ed.) *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security*, Santa Barbara, California, USA, August 24-26, 1981, pp. 11–15. U. C. Santa Barbara, Dept. of Elec. and Computer Eng., ECE Report No. 82-04, 1982 (1981)
7. Chailloux, A., Kerenidis, I.: Optimal quantum strong coin flipping. In: Proceedings of 50th IEEE Symposium on Foundations of Computer Science, pp. 527–533. IEEE Computer Society (2009)

8. Chailloux, A., Kerenidis, I.: Optimal bounds for quantum bit commitment. In: Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science, pp. 354–362. IEEE Computer Society Press (2011). DOI 10.1109/FOCS.2011.42
9. Gutoski, G., Watrous, J.: Toward a general theory of quantum games. In: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, pp. 565–574. ACM, New York, NY, USA (2007)
10. Kerenidis, I., Nayak, A.: Weak coin flipping with small bias. *Information Processing Letters* **89**(3), 131–135 (2004). DOI <http://dx.doi.org/10.1016/j.ipl.2003.07.007>
11. Kitaev, A.: Quantum coin-flipping (2002). Unpublished result. Talk in the 6th Annual workshop on Quantum Information Processing, QIP 2003, Berkeley, CA, USA, December 2002
12. Lo, H.K., Chau, H.F.: Is quantum bit commitment really possible? *Physical Review Letters* **78**(17), 3410–3413 (1997). DOI 10.1103/PhysRevLett.78.3410
13. Lo, H.K., Chau, H.F.: Unconditional security of quantum key distribution over arbitrarily long distances. *Science* **283**, 2050–2056 (1999)
14. Mayers, D.: Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters* **78**(17), 3414–3417 (1997). DOI 10.1103/PhysRevLett.78.3414
15. Mayers, D.: Unconditional security in quantum cryptography. *Journal of the ACM* **48**(3), 351–406 (2001)
16. Mochon, C.: A large family of quantum weak coin-flipping protocols. *Physical Review A* **72**(2), 022,341 (2005). DOI 10.1103/PhysRevA.72.022341
17. Mochon, C.: Quantum weak coin flipping with arbitrarily small bias. Available as arXiv.org e-Print [quant-ph/0711.4114](http://arxiv.org/abs/quant-ph/0711.4114) (2007)
18. Molina, A., Vidick, T., Watrous, J.: Optimal counterfeiting attacks and generalizations for Wiesner’s quantum money. In: Proceedings of the 7th Conference on Theory of Quantum Computation, Communication, and Cryptography, pp. 45–64 (2012)
19. Nayak, A., Shor, P.W.: On bit-commitment based quantum coin flipping. *Physical Review A* **67**(1), 012,304 (2003). DOI 10.1103/PhysRevA.67.012304
20. Nielsen, M., Chuang, I.L.: *Quantum computation and quantum information*. Cambridge University Press, New York, NY, USA (2000)
21. Preskill, J., Shor, P.W.: Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters* **85**(2), 441–444 (2000)
22. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* **26**(5), 1484–1509 (1997)
23. Spekkens, R.W., Rudolph, T.: Degrees of concealment and bindingness in quantum bit commitment protocols. *Physical Review A* **65**, 012,310 (2001). URL [doi:10.1103/PhysRevA.65.012310](http://doi.org/10.1103/PhysRevA.65.012310)
24. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software* **11**, 625–653 (1999)
25. Sturm, J.F.: Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. *Optimization Methods and Software* **17**(6), 1105–1154 (2002)
26. Tunçel, L., Wolkowicz, H.: Strong duality and minimal representations for cone optimization. *Computational Optimization and Applications* pp. 1–30 (2012). DOI 10.1007/s10589-012-9480-0. URL <http://dx.doi.org/10.1007/s10589-012-9480-0>
27. Wiesner, S.: Conjugate coding. *SIGACT News* **15**(1), 78–88 (1983). DOI 10.1145/1008908.1008920. URL <http://doi.acm.org/10.1145/1008908.1008920>
28. Wolkowicz, H., Saigal, R., Vandenberghe, L. (eds.): *Handbook of Semidefinite Programming*. Kluwer Academic Publishers (2000)
29. Yao, A.C.C.: Some complexity questions related to distributive computing. In: Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing, STOC ’79, pp. 209–213. ACM, New York, NY, USA (1979). DOI 10.1145/800135.804414. URL <http://doi.acm.org/10.1145/800135.804414>
30. Yao, A.C.C.: Quantum circuit complexity. In: Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, pp. 352–361. IEEE Computer Society Press, Los Alamitos, CA, USA (1993)