Constant Lower Bounds on the Cryptographic Security of Quantum Two-Party Computations

Sarah A. Osborn

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Jamie Sikora, Chair
Travis W. Morrison
Danfeng Yao

April 20, 2022
Blacksburg, Virginia

Keywords: Quantum Computation, Quantum Cryptography, Lower Bounds, Secure Function Evaluation

Constant Lower Bounds on the Cryptographic Security of Quantum Two-Party Computations

Sarah A. Osborn

(ABSTRACT)

In this thesis, we generate a lower bound on the security of quantum protocols for secure function evaluation. Central to our proof is the concept of gentle measurements of quantum states, which do not greatly disturb a quantum state if a certain outcome is obtained with high probability. We show how a cheating party can leverage gentle measurements to learn more information than should be allowable. To quantify our lower bound, we reduce a specific cryptographic task known as die-rolling to secure function evaluation and use the concept of gentle measurements to relate their security notions. Our lower bound is then obtained using a known security bound for die-rolling known as Kitaev's bound.

Due to the generality of secure function evaluation, we are able to apply this lower bound to obtain lower bounds on the security of quantum protocols for many quantum tasks. In particular, we provide lower bounds for oblivious transfer, XOR oblivious transfer, the equality function, the inner product function, Yao's millionaires' problem, and the secret phrase problem. Note that many of these lower bounds are the first of their kind, which is a testament to the utility of our lower bound. As a consequence, these bounds prove that unconditional security for quantum protocols is impossible for these applications, and since these are constant lower bounds, this rules out any form of boosting toward perfect security. Our work lends itself to future research on designing optimal protocols for the above listed tasks, and potentially others, by providing constant lower bounds to approximate or improve.

Constant Lower Bounds on the Cryptographic Security of Quantum Two-Party Computations

Sarah A. Osborn

(GENERAL AUDIENCE ABSTRACT)

Quantifying the cryptographic security of quantum applications is the focus of much research in the quantum cryptography discipline. Quantum protocols might have better security than their classical counterparts, and this advantage might make the adoption of quantum cryptographic protocols a viable option. In this thesis, we introduce a method for generating constant lower bounds on the security of a variety of quantum applications. This is accomplished through finding a lower bound on the security of a protocol that is general, and by virtue of its generality, can be scoped to quantum applications such that the lower bound can be applied, and constant lower bounds generated for these applications. The significance of the work in this thesis is that many of the constant lower bounds presented are the first of their kind for these quantum applications, thus proving the impossibility of them having unconditional security. This also proves that one cannot asymptotically boost towards perfect security in these quantum tasks by any means. These constant lower bounds also provide a foundation for future work in the study of these quantum applications, specifically in the search for upper and lower bounds on their cryptographic security, as well as in the search for protocols that approximate these bounds.

Dedication

For Mom, Dad, and Savannah.

Acknowledgments

I would like to thank my advisor, Dr. Jamie Sikora, first, for taking a chance on me, and second, for his invaluable mentorship and immeasurable patience throughout this project. I would also like to thank the other members of my committee, Dr. Travis Morrison and Dr. Danfeng Yao, for their guidance and input. Finally, I would like to thank the Department of Defense, who funded my pursuit in higher education.

Contents

List of Figures

Chapter 1

Introduction

In the late 1960s, Stephen Wiesner had an idea. Or at least, that was when he wrote a paper about the idea [1]. The paper introduced what is now known as "quantum multiplexing," which allows a party to send two messages, only one of which can be read by the receiver. The other is irrevocably destroyed in the process. Deemed ahead of its time, this paper was not published until 1983. The quantum cryptography discipline can trace its origins back to this paper, but it was not its eventual publication that signaled its onset - it was Wiesner's discussions with fellow physicist Charles H. Bennett about the paper and its concepts. These discussions eventually led to multiple collaborative publications by Bennett and Gilles Brassard, which, in turn, triggered the belated publication of Wiesner's work. Bennett and Brassard's first paper, which was published in 1982, officially gave a name to the discipline and used aspects from Wiesner's work to introduce a quantum subway token authentication system [2]. Two years later, Bennett and Brassard introduced quantum key distribution via what is known as the BB84 protocol, which was shown to be theoretically, unconditionally secure [3]. Surprisingly, this discovery was not what grabbed the attention of the scientific community - it was the first working prototype, which was produced 5 years later in 1989. A testament to the limitations of quantum computers at the time, the exchange of messages took place over a distance of a mere 32.5 centimeters.

Regardless, for fellow physicists, the ability to bridge theory and practice was exactly what was needed to jumpstart quantum cryptography. Other applications began to surface, es-

pecially those relevant to two-party cryptography. Two-party cryptography is a branch of cryptography where two parties do not trust each other to communicate honestly. One popular application of two-party cryptography is bit commitment. In a standard bit commitment protocol, there are two parties, known typically as Alice and Bob, and two phases: the commit phase, and the reveal phase. In the commit phase, Alice "commits" to a bit b, i.e. the value of b is set. Alice sends some information that is reliant on b, but does not reveal its value, to Bob. An arbitrary amount of time passes, during which both parties may attempt to "cheat." In the reveal phase, Alice reveals b to Bob, and Bob accepts or rejects the value of b, depending on whether he believes Alice has been honest. For Alice to be considered dishonest, she would try to change the value of b after the commit phase. For Bob to be considered dishonest, he would try to learn the value of b before the reveal phase.

Another popular application of two-party cryptography is known as oblivious transfer. The advent of oblivious transfer has been attributed to both Stephen Wiesner and Michael O. Rabin [4]. Rabin introduced the first classical protocol for oblivious transfer. Wiesner conceived the first quantum protocol, interestingly some 10-15 years prior to Rabin's work, though his work ended up being published two years after Rabin's. Despite the respective ideas being considered independent, they both have the same underlying structure. In a standard 1-out-of-2 oblivious transfer protocol, Alice has two messages, and Bob has a choice as to which he learns, though he does not know the contents. Alice and Bob communicate, and at the end of the protocol, Bob unambiguously learns the content of the message he chose. If Alice and Bob are both honest, Alice does not learn anything about Bob's choice, and Bob does not learn anything about the other message's contents.

Unfortunately, any hopes of securing the distinction "unconditionally secure" for quantum bit commitment were dashed in the late 1990s with work by Dominic Mayers [5] and, independently, Hoi-Kwong Lo and Hoi Fung Chau [6, 7]. Lo quickly followed this proof with

another disproving unconditional security for one-sided secure quantum two-party computations [8], which include quantum oblivious transfer and all applications discussed in Chapter 4. While these discoveries were disappointing, it did not completely negate the contribution of these applications. Some quantum applications have proven a greater level of security than their classical counterparts [9]. Even if that were not the case, it still provides for interesting research into the definition of lower bounds on the security of these quantum cryptographic protocols. But how can we quantify the security of these protocols? Luckily, we have at our disposal a primitive known as secure function evaluation, which we use in this thesis to extrapolate lower bounds on the class of quantum protocols it reduces.

1.1 Background

Before we discuss the main result, it is important to include necessary linear algebra and quantum theoretic preliminaries.

1.1.1 Linear algebra

Quantum theory is largely described using linear algebra concepts and definitions.

Complex numbers are integral to quantum theory, as are their complex conjugates.

Definition 1.1 (Complex conjugate of a complex number). The complex conjugate of a complex number $c = a + bi$ is denoted $\overline{c} = a - bi$.

One way of grouping numbers, real, complex, or otherwise, is through a vector.

Definition 1.2 (Vector). An n-dimensional vector v can be denoted

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix}, \tag{1.1}$$

where $v_1, v_2, \ldots, v_{n-1}, v_n$ form the components of v. A vector $v \in \mathbb{R}^n$ has n real components. Similarly, a vector $v \in \mathbb{C}^n$ has n complex components.

Next, we define the complex conjugate transpose of a vector.

Definition 1.3 (Complex conjugate transpose of a vector). The complex conjugate transpose of a vector

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix} \tag{1.2}$$

is denoted as

$$v^* = \begin{bmatrix} \overline{v_1} & \overline{v_2} & \ldots & \overline{v_{n-1}} & \overline{v_n} \end{bmatrix}. \tag{1.3}$$

Next, we define the inner product of two complex vectors.

Definition 1.4 (Inner product of two complex vectors). The inner product of two vectors u, $v \in \mathbb{C}^n$, is defined as follows:

$$\langle u, v \rangle = \sum_{i=1}^{n} \overline{u_i} \cdot v_i. \tag{1.4}$$

We also define the Euclidean norm for vectors.

Definition 1.5 (Euclidean norm for vectors). The Euclidean norm for a vector v is given by

$$\|v\| = \sqrt{\langle v, v \rangle}. \tag{1.5}$$

The Euclidean norm for a vector is also commonly denoted $\|v\|_2$.

Having defined the Euclidean norm for a vector, we now state the Cauchy-Schwarz inequality.

Definition 1.6 (Cauchy-Schwarz inequality). Let u and v be vectors. The following inequality holds:

$$|\langle u, v \rangle| \leq \|u\| \, \|v\| . \tag{1.6}$$

Next, we define the complex conjugate transpose of a matrix.

Definition 1.7 (Complex conjugate transpose of a matrix). The complex conjugate transpose of a matrix A where

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \tag{1.7}$$

is given by

$$A^* = \begin{bmatrix} \overline{a_{11}} & \overline{a_{21}} & \dots & \overline{a_{m1}} \\ \overline{a_{12}} & \overline{a_{22}} & \dots & \overline{a_{m2}} \\ \overline{a_{13}} & \overline{a_{23}} & \dots & \overline{a_{m3}} \\ \vdots & \vdots & \vdots & \vdots \\ \overline{a_{1n}} & \overline{a_{2n}} & \dots & \overline{a_{mn}} \end{bmatrix}. \tag{1.8}$$

Next, we define some important matrix operations.

Definition 1.8 (Trace of a matrix). The trace of an $m \times m$ matrix A is given by the sum of the elements along the diagonal, i.e.

$$\mathrm{Tr}\,(A) = \sum_{i=1}^{m} a_{ii}. \tag{1.9}$$

In general, the trace operation is cyclic.

Definition 1.9 (Cyclic property of trace). For matrices A, B, C, we have

$$\mathrm{Tr}\,(ABC) = \mathrm{Tr}\,(BCA) = \mathrm{Tr}\,(CAB)\,. \tag{1.10}$$

Next, we define the Frobenius inner product.

Definition 1.10 (Frobenius inner product). Consider two matrices A and $B \in \mathbb{C}^{m\times n}$. The Frobenius inner product of these matrices is defined as

$$\langle A, B\rangle = \sum_{i,j} \overline{a_{ij}} b_{ij} = \mathrm{Tr}\,(A^* B)\,. \tag{1.11}$$

The Frobenius inner product is distributive.

Definition 1.11 (Distributive property of the Frobenius inner product). Consider three matrices A, B and $C \in \mathbb{C}^{m\times n}$. Then,

$$\langle A - B, C\rangle = \langle A, C\rangle - \langle B, C\rangle, \tag{1.12}$$

and

$$\langle A + B, C\rangle = \langle A, C\rangle + \langle B, C\rangle. \tag{1.13}$$

Next, we define the trace norm.

Definition 1.12 (Trace norm of a matrix). The trace norm of a matrix A is given by

$$\|A\|_{tr} = \mathrm{Tr}\left(\sqrt{AA^*}\right). \tag{1.14}$$

The trace norm is also commonly denoted $\|A\|_1$.

We now define the operator norm.

Definition 1.13 (Operator norm). The operator norm of a matrix A that operates on a vector space V for $V \neq \{0\}$ is given by

$$\|A\|_{op} = \sup\{\|Av\| : \|v\| = 1 \text{ and } v \in V\}. \tag{1.15}$$

The operator norm is also commonly denoted $\|A\|_\infty$.

The operator norm is submultiplicative.

Definition 1.14 (Submultiplicative property of the operator norm). Consider two matrices A and B. Then

$$\|AB\|_{op} \leq \|A\|_{op} \|B\|_{op}. \tag{1.16}$$

We also define the trace distance.

Definition 1.15 (Trace distance). The trace distance of two matrices A and B is half the trace norm of their difference, i.e.

$$\Delta(A, B) = \frac{1}{2}\|A - B\|_{tr}. \tag{1.17}$$

We define the Kronecker product, one of the most important combining operations in quantum computing.

Definition 1.16 (Kronecker product). The Kronecker product of an $m \times n$ matrix A and a $p \times q$ matrix B results in a $mp \times nq$ matrix given by

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & a_{13}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & a_{23}B & \dots & a_{2n}B \\ a_{31}B & a_{32}B & a_{33}B & \dots & a_{3n}B \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1}B & a_{m2}B & a_{m3}B & \dots & a_{mn}B \end{bmatrix}. \tag{1.18}$$

One definition integral to the proof of our main result is that of the Matrix Hölder inequality.

Definition 1.17 (Matrix Hölder inequality [10]). Let A, B be two $m \times m$ matrices and let $p,\ q \in [1, \infty]$ where $\frac{1}{p} + \frac{1}{q} = 1$. Then, the following inequality holds true: $|\langle A, B\rangle| = |\mathrm{Tr}\,(A^* B)| \leq \|A\|_p \|B\|_q$.

Another inequality used in this thesis is the triangle inequality.

Definition 1.18 (Triangle inequality). Given real values a and b, it holds that

$$|a| - |b| \leq |a - b|\,. \tag{1.19}$$

Now, we define some matrices prevalent in quantum computing. First, we define the identity matrix.

Definition 1.19 (Identity matrix). The identity matrix is the matrix with 1s across the diagonal, and 0 otherwise. This matrix is denoted I_m, where m is an indicator of the size of the matrix (which is square, so its size is $m \times m$), or simply I, and the size is inferred.

The identity matrix has the property such that for any $m \times n$ matrix A, $I_m A = A I_n = A$. A few examples of I are shown below:

$$I_1 = \begin{bmatrix} 1 \end{bmatrix}, I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, ..., I_n = \begin{bmatrix} 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & ... & 1 \end{bmatrix}. \tag{1.20}$$

Hermitian matrices tend to make appearances.

Definition 1.20 (Hermitian matrix). A matrix A is Hermitian if it is $n \times n$ (square) and $A = A^*$.

Postive semi-definite matrices are also important.

Definition 1.21 (Postive semi-definite matrix and ordering). A matrix A is positive semi-definite if it is Hermitian, and all of its eigenvalues are non-negative. If a matrix A is positive semi-definite, this property is denoted by $A \geq 0$. If $B \geq A$ for some Hermitian matrix B, this means $B - A \geq 0$.

1.1.2 Quantum theory

With the relevant linear algebra concepts established, it is also important to present some essential tenets of quantum theory and relevant definitions.

First, we define a quantum state.

Definition 1.22 (Quantum state). A quantum state encodes quantum information. More specifically, it encodes a probability distribution for all measurement outcomes of a quantum system.

Next, we define density matrices.

Definition 1.23 (Density matrix). A density matrix ρ can be used to represent quantum states. Density matrices are Hermitian, positive semi-definite and have the property $\mathrm{Tr}(\rho) = 1$.

In quantum mechanics, measurement of a quantum state produces a classical output, and often disturbs the state to some extent. The most general form of measurement are Positive Operator-Valued Measures (POVMs).

Definition 1.24 (POVM). Let us define the measurement operators $P_1, \ldots, P_n$. If P_i is positive semi-definite for all $i \in \{1, \ldots, n\}$, and $\sum_{i=1}^{n} P_i = I$, then $\{P_1, \ldots, P_n\}$ is a Positive Operator-Valued Measure (POVM). When measuring a quantum state ρ with this POVM, the probability of receiving classical outcome i is given by $\langle \rho, P_i \rangle$.

1.2 Secure function evaluation

Having presented the relevant concepts and definitions from linear algebra and quantum theory, we define secure function evaluation, which forms the foundation of the research in this thesis.

Secure function evaluation (SFE) is a cryptographic primitive that allows two parties to compute a deterministic function on their inputs.

Formally, Alice has an input $x \in X$, and Bob has an input $y \in Y$. Each input is chosen uniformly at random. Bob has a deterministic function $f : X \times Y \to B$, i.e., Bob's function f evaluates over Alice and his inputs to produce an output $b \in B$. For the purposes of this work, we take X, Y, and B to have finite cardinality. SFE is illustrated in Figure 1.1.

It is important to note that in this thesis we examine the case of one-sided secure function evaluation, where Bob is the only one with an output. There is another form of secure

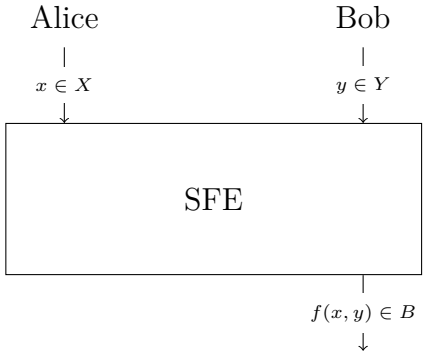

Figure 1.1: Secure function evaluation.

function evaluation where Alice also has a function $g : X \times Y \to A$, but for our purposes, we take $g(x, y) = \emptyset$, and delegate the study of two-sided secure function evaluation to future work.

We define the following goals with respect to SFE:

1. Completeness. If both Alice and Bob are honest, then the protocol performs as expected, i.e., Bob learns f on inputs x and y.

2. Soundness against cheating Bob. If Bob is dishonest, he obtains no more information about honest Alice's input x other than that which follows from learning f on inputs x and y.

3. Soundness against cheating Alice. If Alice is dishonest, she obtains no information about honest Bob's input y.

The latter two goals imply the possibility of dishonest parties. To this end, we define the following cheating probabilities for Bob:

- $\mathrm{B}_{\mathrm{SFE}}$: the maximum probability that cheating Bob can correctly learn honest Alice's input x.

- B'_{SFE}: the maximum probability that cheating Bob can correctly learn $f(x, y)$ $\forall y \in Y$.

In general, for deterministic function f, $B_{SFE} \leq B'_{SFE}$, since if Bob correctly learns Alice's input, he can compute any function over it he wants.

We also define the following cheating probability for Alice:

- A_{SFE}: the maximum probability that cheating Alice can correctly learn honest Bob's input y.

We are now well-equipped to venture on to the main result. Before that, we review the literature on lower bounds for related quantum cryptographic tasks.

Chapter 2

Review of literature

There have been many advancements in the discovery of lower bounds for quantum cryptographic applications, and the security of these applications is a subject of much research. These discoveries give important context to the work presented in this thesis.

There are three applications that have pre-existing lower bounds which we discuss here: oblivious transfer, coin-flipping, and die-rolling.

The first quantum protocol for oblivious transfer was developed by Stephen Wiesner, and the paper written on it was published in 1983 [1]. He implies the insecurity of the protocol, though does not quantify it. In 1997, Hoi-Kwong Lo proved the insecurity of all one-sided quantum two-party computations [8], which include quantum oblivious transfer. In 2013, it was found that dishonest Alice or Bob can learn the other's input with probability at least 0.5852 in 1-out-of-2 oblivious transfer [11]. In 2015, this was improved to $\frac{2}{3}$ [12]. For 1-out-of-2 oblivious transfer protocols where Alice's two messages are bit strings, it was found in 2018 that dishonest Alice or Bob can learn the other's input with probability at least 0.61 [13].

Buhrman, Christandl, and Schaffner expanded Lo's proof of insecurity for one-sided quantum two-party computations, and proved the insecurity of all quantum two-party computations [14], which include applications such as coin-flipping and die-rolling. Quantum coin-flipping is a cryptographic primitive where Alice and Bob flip a "coin," customarily by quantum telephone, and try to agree on the output. In weak coin-flipping, dishonest parties

try to influence the other's output to opposite values; in strong coin-flipping, dishonest parties try to influence the other's output to any value. Kitaev previously proved in 2002 that secure quantum strong coin-flipping was impossible. He also proved that dishonest Alice or Bob can influence the other's outcome with probability at least $\frac{1}{\sqrt{2}}$ [15]. In 2001, Ambainis presented a protocol where dishonest Alice or Bob could influence the other's outcome with maximum probability $\frac{3}{4}$ [16]. Kerenidis and Nayak simplified Ambainis' protocol and achieved the same result a year later [17]. In weak coin-flipping, Mochon proved there exist protocols such that near-perfect security can be achieved [18]. Chailloux and Kerenidis built upon the protocol presented by Mochon for weak coin-flipping, which allowed them to generate optimal quantum strong coin-flipping protocols, the security of which comes arbitrarily close to the bounds presented by Kitaev [19].

Finally, we discuss lower bounds on quantum die-rolling. Die-rolling is the expansion of coin-flipping to an N-sided die. Aharon and Silman generalized Kitaev's bound for strong coin-flipping to N outcomes. In this work, we use this lower bound for die-rolling to give us a lower bound on SFE. This also implies lower bounds on oblivious transfer. Additionally, Aharaon and Silman extended upon the work in [19] to generate optimal strong die-rolling protocols. Finally, Sikora showed how to create simple protocols for quantum die-rolling in 2018, the security of which closely approximate Kitaev's generalized bound, especially with large N [20].

Chapter 3

Result

The main result of this thesis is the proof of the following theorem.

Theorem 3.1. In any quantum protocol for SFE, it holds that

$$\mathrm{B}'_{\mathrm{SFE}} \geq \frac{1}{|Y|\,\mathrm{A}_{\mathrm{SFE}}} - 2\,(|Y|-1)\sqrt{1-\frac{1}{|Y|\,\mathrm{A}_{\mathrm{SFE}}}}, \tag{3.1}$$

where $|Y|$ is the cardinality of Bob's input set, and $\mathrm{B}'_{\mathrm{SFE}}$ and $\mathrm{A}_{\mathrm{SFE}}$ are Bob and Alice's cheating probabilities defined in section 1.2.

There are a few notes to make about the components of the above inequality. Since Alice can always randomly guess Bob's input, it holds that

$$\mathrm{A}_{\mathrm{SFE}} \geq \frac{1}{|Y|}. \tag{3.2}$$

In the case Alice can not learn anything from the protocol ($\mathrm{A}_{\mathrm{SFE}} = \frac{1}{|Y|}$), (3.1) implies $\mathrm{B}'_{\mathrm{SFE}} = 1$, indicating that Bob can cheat perfectly in computing f for all y. This tradeoff between Alice and Bob was similarly concluded by Hoi-Kwong Lo in his 1997 paper [8].

Let us define the following probabilities:

- $\mathrm{A}_{\mathrm{rand}}$: the maximum probability that cheating Alice can correctly learn honest Bob's input y with only black-box access to the SFE protocol.

- $\mathrm{B}'_{\mathrm{rand}}$: the maximum probability that cheating Bob can correctly learn $f(x, y)$ for all y with only black-box access to the SFE protocol.

In the "black-box" case, Alice's only recourse is to randomly guess Bob's input, since in this branch of SFE examples, she does not have an output. So,

$$\mathrm{A}_{\mathrm{rand}} = \frac{1}{|Y|}. \tag{3.3}$$

However, since Bob does have an output function, $\mathrm{B}'_{\mathrm{rand}}$ must be evaluated on a case-by-case basis, since different functions can reveal different levels of information about a given protocol and Alice's input, say.

We now present the following theorem.

Theorem 3.2. In any quantum protocol for SFE, either $\mathrm{B}'_{\mathrm{rand}} = 1$, or there exists a constant $c > 1$ such that

$$\mathrm{A}_{\mathrm{SFE}} \geq c \cdot \mathrm{A}_{\mathrm{rand}} \quad \text{or} \quad \mathrm{B}'_{\mathrm{SFE}} \geq c \cdot \mathrm{B}'_{\mathrm{rand}}. \tag{3.4}$$

If $\mathrm{A}_{\mathrm{SFE}} = \mathrm{A}_{\mathrm{rand}}$, by our main result, $\mathrm{B}'_{\mathrm{SFE}} = 1$. If $\mathrm{B}'_{\mathrm{rand}} < 1$, then Alice or Bob can cheat.

So how do we find this constant c? We see our lower bound is a continuous, decreasing function with respect to $\mathrm{A}_{\mathrm{SFE}}$. We make the assumption that

$$\mathrm{A}_{\mathrm{SFE}} \leq \frac{c_A}{|Y|} = c_A \cdot \mathrm{A}_{\mathrm{rand}}, \tag{3.5}$$

for a fixed constant $c_A \geq 1$. Then, it follows that

$$\mathrm{B}'_{\mathrm{SFE}} \geq \frac{1}{c_A} - 2\left(|Y| - 1\right)\sqrt{1 - \frac{1}{c_A}}. \tag{3.6}$$

Suppose

$$\mathrm{B}'_{\mathrm{SFE}} = c_B \cdot \mathrm{B}'_{\mathrm{rand}} \tag{3.7}$$

for some constant $c_B \geq 1$. Then, we rearrange Inequality (3.6) like so:

$$c_B \geq \frac{1}{\mathrm{B}'_{\mathrm{rand}}} \left(\frac{1}{c_A} - 2\left(|Y| - 1\right) \sqrt{1 - \frac{1}{c_A}} \right). \tag{3.8}$$

We have the constants c_A and c_B, and now it is time to relate the two such that we get one constant $c > 1$. For this, we assume $\mathrm{B}'_{\mathrm{rand}} < 1$. Then we have $\frac{1}{\mathrm{B}'_{\mathrm{rand}}} > 1$. When $c_A = 1$, the right-hand side of Inequality (3.8) equals $\frac{1}{\mathrm{B}'_{\mathrm{rand}}} > 1$. On the other hand, when $c_A = \frac{1}{\mathrm{B}'_{\mathrm{rand}}}$, the right-hand side is strictly less than 1, since we have

$$\begin{aligned}
\frac{1}{\mathrm{B}'_{\mathrm{rand}}} \left(\frac{1}{c_A} - 2\left(|Y| - 1\right) \sqrt{1 - \frac{1}{c_A}} \right) &= \frac{1}{\mathrm{B}'_{\mathrm{rand}}} \left(\frac{1}{\frac{1}{\mathrm{B}'_{\mathrm{rand}}}} - 2\left(|Y| - 1\right) \sqrt{1 - \frac{1}{\frac{1}{\mathrm{B}'_{\mathrm{rand}}}}} \right) \\
&= \frac{1}{\mathrm{B}'_{\mathrm{rand}}} \left(\mathrm{B}'_{\mathrm{rand}} - 2\left(|Y| - 1\right) \sqrt{1 - \mathrm{B}'_{\mathrm{rand}}} \right) \\
&= 1 - \frac{2\left(|Y| - 1\right)}{\mathrm{B}'_{\mathrm{rand}}} \sqrt{1 - \mathrm{B}'_{\mathrm{rand}}} \\
&< 1.
\end{aligned} \tag{3.9}$$

We now make use of the intermediate value theorem, defined below.

Theorem 3.3 (Intermediate Value Theorem). Let f be a continuous function over the closed interval $[a, b]$. Let s be a value that exists between $f(a)$ and $f(b)$, i.e. the intermediate value. Then there exists a value c in the interval $[a, b]$ such that $f(c) = s$.

For this function, we now have the right-hand side of Inequality (3.8) defined between a value strictly less than 1 $\left(\text{given by } c_A = \frac{1}{\mathrm{B}'_{\mathrm{rand}}}\right)$, and strictly greater than 1 (given by $c_A = 1$). The

intermediate value theorem then purports that a value $c > 1$ exists such that

$$c = \frac{1}{\mathrm{B}'_{\mathrm{rand}}} \left(\frac{1}{c} - 2\left(|Y| - 1\right) \sqrt{1 - \frac{1}{c}} \right). \tag{3.10}$$

For this value of c, we have that if $c_A \leq c$, then $c_B \geq c$. Then, if $\mathrm{A}_{\mathrm{SFE}} \leq c \cdot \mathrm{A}_{\mathrm{rand}}$, then $\mathrm{B}'_{\mathrm{SFE}} \geq c \cdot \mathrm{B}'_{\mathrm{rand}}$, thus proving Theorem 3.2.

We can compute this value c given values of $\mathrm{B}'_{\mathrm{rand}}$ and $|Y|$, and thus give constant lower bounds on $\mathrm{A}_{\mathrm{SFE}}$ and $\mathrm{B}'_{\mathrm{SFE}}$ using Theorem 3.2. Through the generalization of quantum applications to SFE, we are able to generate constant lower bounds for those applications as well; this is explored in Chapter 4. However, we first discuss the proof of our Inequality (3.1).

3.1 Proof of main result

The proof of our lower bound has two essential components: gentle measurements and die-rolling. We also make use of an inequality known as Kitaev's bound for die-rolling. We describe each of these briefly.

3.1.1 Gentle measurement

Measurements constitute a large part of discussion in quantum computation. Part of this discussion centers around measurement disturbance, namely, how much a state has been disturbed when a measurement is applied. Measurement of a quantum state, which is a particularly unstable particle, may cause it to collapse, resulting in information loss. In the scope of quantum cryptography, the collapse of a state might not just result in this loss - it can also alert other parties that a cheating attempt has been made. In this event, parties

might abort the cryptographic protocol. But, there is a concept of gentle measurements that effectively minimizes this disturbance.

The high-level idea of gentle measurements is that if a quantum state is measured and produces a given result with high probability, then the trace distance between this post-measured state and the original state will be almost negligible. This relative intactness allows a dishonest party to glean more information from a state with successive measurements. Formally, we define the Gentle Measurement Operator Lemma below.

Lemma 3.4 (Gentle Measurement Operator Lemma [21, 22]). Consider a density operator ρ and a measurement operator Λ where $0 \leq \Lambda \leq I$. The measurement operator could be an element of a POVM. Suppose that the measurement operator Λ has a high probability of detecting state ρ:

$$\mathrm{Tr}\left(\Lambda\rho\right) \geq 1 - \varepsilon, \tag{3.11}$$

where $\varepsilon \in [0, 1]$ (the probability of detection is high if ε is close to zero). Then $\sqrt{\Lambda}\rho\sqrt{\Lambda}$ is $2\sqrt{\varepsilon}$-close to the original state ρ in trace distance:

$$\left\|\rho - \sqrt{\Lambda}\rho\sqrt{\Lambda}\right\|_{tr} \leq 2\sqrt{\varepsilon}. \tag{3.12}$$

We extend this gentle measurement operator to define probabilities related to successive measurements of a quantum state. This forms a core part of our proof of our main result.

Lemma 3.5 (Sequential Measurement Lemma). Consider a density operator ρ and measurement operators $\Lambda_1, \ldots, \Lambda_n$ where $0 \leq \Lambda_i \leq I$ for all $i \in \{1, \ldots, n\}$, where $n \geq 2$. Suppose that

$$\mathrm{Tr}\left(\Lambda_i\rho\right) \geq 1 - \varepsilon_i, \tag{3.13}$$

where $\varepsilon_i \in [0,1]$. Then we have

$$\mathrm{Tr}\left(\sqrt{\Lambda_n}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_n}\rho\right) \geq 1-\epsilon_1-2\sum_{i=2}^{n}\sqrt{\varepsilon_i}. \tag{3.14}$$

This bound is related to the quantum union bound. See [23, 24] for a good version of this bound, and also [25] for a simple proof of it. While our bound is not always stronger, it can be viewed as complementary.

Proof. We prove this by induction, starting with the base case of $n = 2$. Thus, there are two measurement operators: Λ_1 and Λ_2, where $\mathrm{Tr}(\Lambda_1\rho) \geq 1-\varepsilon_1$ and $\mathrm{Tr}(\Lambda_2\rho) \geq 1-\varepsilon_2$. Suppose Λ_2 is applied first to ρ. Then, by Lemma 3.4,

$$\left\|\rho - \sqrt{\Lambda_2}\rho\sqrt{\Lambda_2}\right\|_{tr} \leq 2\sqrt{\varepsilon_2}. \tag{3.15}$$

Let Λ_1 be applied, and consider the quantity

$$\mathrm{Tr}\left(\Lambda_1\sqrt{\Lambda_2}\rho\sqrt{\Lambda_2}\right). \tag{3.16}$$

By the cyclic property of trace,

$$\mathrm{Tr}\left(\Lambda_1\sqrt{\Lambda_2}\rho\sqrt{\Lambda_2}\right) = \mathrm{Tr}\left(\rho\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\right), \tag{3.17}$$

and by the Frobenius inner product identity,

$$\mathrm{Tr}\left(\rho\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\right) = \left\langle \rho, \sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\right\rangle. \tag{3.18}$$

By the Matrix Hölder inequality,

$$\left|\left\langle \rho - \sqrt{\Lambda_2}\rho\sqrt{\Lambda_2},\, \Lambda_1 \right\rangle\right| \leq \left\|\rho - \sqrt{\Lambda_2}\rho\sqrt{\Lambda_2}\right\|_{tr} \|\Lambda_1\|_{op}\,. \tag{3.19}$$

We know from Inequality (3.15) that

$$\left\|\rho - \sqrt{\Lambda_2}\rho\sqrt{\Lambda_2}\right\|_{tr} \leq 2\sqrt{\varepsilon_2}. \tag{3.20}$$

and by virtue of $0 \leq \Lambda_1 \leq I$, we know that

$$\|\Lambda_1\|_{op} \leq 1. \tag{3.21}$$

Putting these together, we have then that

$$\left|\left\langle \rho - \sqrt{\Lambda_2}\rho\sqrt{\Lambda_2},\, \Lambda_1 \right\rangle\right| \leq 2\sqrt{\varepsilon_2}. \tag{3.22}$$

By the distributive property of the Frobenius inner product,

$$\left|\left\langle \rho - \sqrt{\Lambda_2}\rho\sqrt{\Lambda_2},\, \Lambda_1 \right\rangle\right| = \left|\langle \rho,\, \Lambda_1\rangle - \left\langle \sqrt{\Lambda_2}\rho\sqrt{\Lambda_2},\, \Lambda_1 \right\rangle\right|. \tag{3.23}$$

By the triangle inequality,

$$|\langle \rho,\, \Lambda_1\rangle| - \left|\left\langle \sqrt{\Lambda_2}\rho\sqrt{\Lambda_2},\, \Lambda_1 \right\rangle\right| \leq \left|\langle \rho,\, \Lambda_1\rangle - \left\langle \sqrt{\Lambda_2}\rho\sqrt{\Lambda_2},\, \Lambda_1 \right\rangle\right| \leq 2\sqrt{\varepsilon_2}. \tag{3.24}$$

Using the Frobenius inner product identity a second time, we have

$$|\mathrm{Tr}\,(\rho\Lambda_1)| - \left|\mathrm{Tr}\left(\sqrt{\Lambda_2}\rho\sqrt{\Lambda_2}\Lambda_1\right)\right| \leq 2\sqrt{\varepsilon_2}, \tag{3.25}$$

and the cyclic property of trace gives

$$|\mathrm{Tr}\left(\Lambda_1\rho\right)| - \left|\mathrm{Tr}\left(\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\rho\right)\right| \leq 2\sqrt{\varepsilon_2}. \tag{3.26}$$

These terms are non-negative, so the inequality is reduced to

$$\mathrm{Tr}\left(\Lambda_1\rho\right) - \mathrm{Tr}\left(\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\rho\right) \leq 2\sqrt{\varepsilon_2}. \tag{3.27}$$

Rearranging the terms gives

$$\mathrm{Tr}\left(\Lambda_1\rho\right) - 2\sqrt{\varepsilon_2} \leq \mathrm{Tr}\left(\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\rho\right), \tag{3.28}$$

and substituting in for $\mathrm{Tr}(\Lambda_1\rho)$ completes our proof of the base case:

$$\mathrm{Tr}\left(\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\rho\right) \geq 1 - \varepsilon_1 - 2\sqrt{\varepsilon_2}. \tag{3.29}$$

If Λ_1 were applied first, the expression would instead be

$$\mathrm{Tr}\left(\sqrt{\Lambda_1}\Lambda_2\sqrt{\Lambda_1}\rho\right) \geq 1 - \varepsilon_2 - 2\sqrt{\varepsilon_1}. \tag{3.30}$$

Inductive step: Assume the lemma proves true up to $k = n-1$. Now let us prove the lemma for $k+1$. Following the steps of the base case, we have

$$\begin{aligned}
&\mathrm{Tr}\left(\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}\cdots\sqrt{\Lambda_2}\right) = \\
&\mathrm{Tr}\left(\rho\sqrt{\Lambda_{k+1}}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_{k+1}}\right) = \\
&\left\langle\rho, \sqrt{\Lambda_{k+1}}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_{k+1}}\right\rangle.
\end{aligned} \tag{3.31}$$

Using the Matrix Hölder inequality, we have

$$\begin{aligned} \left|\left\langle \rho - \sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}, \sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right\rangle\right| \leq \\ \left\|\rho - \sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}\right\|_{tr} \left\|\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right\|_{op}. \end{aligned} \tag{3.32}$$

By the submultiplicative property of the operator norm, we have

$$\begin{aligned} \left\|\rho - \sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}\right\|_{tr} \left\|\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right\|_{op} \leq \\ \left\|\rho - \sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}\right\|_{tr} \left\|\sqrt{\Lambda_k}\right\|_{op}\cdots\left\|\sqrt{\Lambda_2}\right\|_{op}\|\Lambda_1\|_{op}\left\|\sqrt{\Lambda_2}\right\|_{op}\cdots\left\|\sqrt{\Lambda_k}\right\|_{op}. \end{aligned} \tag{3.33}$$

Substituting in for the first norm, and using the fact the operator norm for all the measurement operators is less than or equal to 1, we have

$$\begin{aligned} \left|\left\langle \rho - \sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}, \sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right\rangle\right| \leq \\ \left\|\rho - \sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}\right\|_{tr} \left\|\sqrt{\Lambda_k}\right\|_{op}\cdots\left\|\sqrt{\Lambda_2}\right\|_{op}\|\Lambda_1\|_{op}\left\|\sqrt{\Lambda_2}\right\|_{op}\cdots\left\|\sqrt{\Lambda_k}\right\|_{op} \leq \\ 2\sqrt{\varepsilon_{k+1}}. \end{aligned} \tag{3.34}$$

Continuing with the steps from the base case, we have

$$
\begin{aligned}
&2\sqrt{\varepsilon_{k+1}} \\
&\geq \left|\left\langle \rho - \sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}, \sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right\rangle\right| \\
&= \left|\left\langle \rho, \sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right\rangle - \left\langle \sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}, \sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right\rangle\right| \\
&\geq \left|\left\langle \rho, \sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right\rangle\right| - \left|\left\langle \sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}, \sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right\rangle\right| \\
&= \left|\mathrm{Tr}\left(\rho\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right)\right| - \left|\mathrm{Tr}\left(\sqrt{\Lambda_{k+1}}\rho\sqrt{\Lambda_{k+1}}\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\right)\right| \\
&= \left|\mathrm{Tr}\left(\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\rho\right)\right| - \left|\mathrm{Tr}\left(\sqrt{\Lambda_{k+1}}\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\sqrt{\Lambda_{k+1}}\rho\right)\right| \\
&= \mathrm{Tr}\left(\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\rho\right) - \mathrm{Tr}\left(\sqrt{\Lambda_{k+1}}\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\sqrt{\Lambda_{k+1}}\rho\right).
\end{aligned}
\tag{3.35}
$$

Rearranging the terms, and substituting in for the first term finishes the proof:

$$
\begin{aligned}
&\mathrm{Tr}\left(\sqrt{\Lambda_{k+1}}\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\sqrt{\Lambda_{k+1}}\rho\right) \\
&\geq \mathrm{Tr}\left(\sqrt{\Lambda_k}\cdots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\cdots\sqrt{\Lambda_k}\rho\right) - 2\sqrt{\varepsilon_{k+1}} \\
&\geq \left(1 - \varepsilon_1 - 2\sum_{i=2}^{k}\sqrt{\varepsilon_i}\right) - 2\sqrt{\varepsilon_{k+1}} \\
&= 1 - \varepsilon_1 - 2\sum_{i=2}^{k+1}\sqrt{\varepsilon_i}.
\end{aligned}
\tag{3.36}
$$

□

From this lemma and its proof, we furthermore define what it means to learn information given a state, and generalize this in another lemma, which scopes to ultimately prove our lower bound in subsection 3.1.3.

Lemma 3.6. Suppose there is a function f on two inputs x and y. Suppose Bob wants to learn $f(x, y)$ for all y in a set Y, and suppose $\Pr[\text{Bob learns } f(x, y_i)] \geq 1 - \varepsilon_i$ for $i \in$

$\{1, 2, \ldots, |Y|-1, |Y|\}$, where y_i is the ith element of Y. Then,

$$\Pr\left[\text{Bob learns } f(x,y)\ \forall y\right] \geq 1 - \frac{1}{|Y|}\left(\sum_{i=1}^{|Y|}\varepsilon_i + 2\left(|Y|-1\right)\sum_{i=1}^{|Y|}\sqrt{\varepsilon_i}\right). \tag{3.37}$$

Proof. Suppose Alice and Bob share a joint state ρ, which is dependent on Alice's uniformly random input $x \in X$. The value of x must first be ascertained for use in the protocol. As such, we assume Alice will measure the state with a POVM $\{C_x : x \in X\}$ to obtain the value of x.

Suppose Bob wants to learn a function $f : X \to B$. He can measure with a POVM $\{D_b : b \in B\}$. The probability that Bob is correct in learning f is given by

$$\mathrm{Tr}\left(\left(\sum_{x\in X} C_x \otimes D_{f(x)}\right)\rho\right). \tag{3.38}$$

Now we scope to SFE by expanding this measurement to encompass the idea of multiple functions. Relative to SFE, this means we allow for Bob's input $y \in Y$, and allow him to evaluate $f(x,y)$. Let us, for simplicity, set an index $i \in \{1, 2, \ldots, |Y|-1, |Y|\}$, such that y_i is the ith element of Y. Suppose further that Bob has a set of POVMs over his input set Y: $\{\{M_b^i : b \in B\} : i \in \{1, 2, \ldots, |Y|-1, |Y|\}\}$, where $\{M_b^i : b \in B\}$ is the POVM corresponding to y_i. Suppose he learns $f(x, y_i)$, using the corresponding POVM, with probability at least $1 - \varepsilon_i$, i.e.

$$\mathrm{Tr}\left(\left(\sum_{x\in X} C_x \otimes M^i_{f(x,y_i)}\right)\rho\right) \geq 1 - \varepsilon_i. \tag{3.39}$$

Here is where Lemma 3.5 comes in. We set Λ_i such that

$$\Lambda_i = \sum_{x\in X} C_x \otimes M^i_{f(x,y_i)}, \tag{3.40}$$

and apply the lemma to see that

$$\mathrm{Tr}\left(\sqrt{\Lambda_{|Y|}}\sqrt{\Lambda_{|Y|-1}}\dots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\dots\sqrt{\Lambda_{|Y|-1}}\sqrt{\Lambda_{|Y|}}\rho\right) \geq 1-\varepsilon_1-2\sum_{i=2}^{|Y|}\sqrt{\varepsilon_i}. \tag{3.41}$$

Assuming $\{C_x : x \in X\}$ is a projective measurement without loss of generality, we have

$$\begin{aligned}\sqrt{\Lambda_{|Y|}}\sqrt{\Lambda_{|Y|-1}}\dots\sqrt{\Lambda_2}\Lambda_1\sqrt{\Lambda_2}\dots\sqrt{\Lambda_{|Y|-1}}\sqrt{\Lambda_{|Y|}} = \\ \sum_{x\in X} C_x \otimes \sqrt{M^{|Y|}_{f\left(x,y_{|Y|}\right)}}\dots\sqrt{M^2_{f(x,y_2)}}M^1_{f(x,y_1)}\sqrt{M^2_{f(x,y_2)}}\dots\sqrt{M^{|Y|}_{f\left(x,y_{|Y|}\right)}}.\end{aligned} \tag{3.42}$$

One can define a POVM

$$\left\{\widetilde{M}_{b_1,\dots,b_{|Y|}} : b_1,\dots,b_{|Y|} \in B\right\} \tag{3.43}$$

where

$$\widetilde{M}_{b_1,\dots,b_{|Y|}} = \sqrt{M^{|Y|}_{b_{|Y|}}}\dots\sqrt{M^2_{b_2}}M^1_{b_1}\sqrt{M^2_{b_2}}\dots\sqrt{M^{|Y|}_{b_{|Y|}}}. \tag{3.44}$$

This is a valid POVM, and it indicates Bob has one measurement to successfully learn $f(x,y)$ for all $y \in Y$ with probability greater than or equal to

$$1-\varepsilon_1-2\sum_{i=2}^{|Y|}\sqrt{\varepsilon_i}. \tag{3.45}$$

Let the probability that Bob learns $f(x,y)$ for all $y \in Y$ now be fixed at $1-\varepsilon$, such that

$$1-\varepsilon \geq 1-\varepsilon_1-2\sum_{i=2}^{|Y|}\sqrt{\varepsilon_i}, \tag{3.46}$$

which simplifies to

$$\varepsilon \leq \varepsilon_1+2\sum_{i=2}^{|Y|}\sqrt{\varepsilon_i}. \tag{3.47}$$

This assumes that $f(x, y_1)$ is learned last but this is not necessarily the case. Lemma 3.5 accounts for that. The following group of inequalities also holds true:

$$\begin{aligned}
\varepsilon &\leq \varepsilon_2 + 2 \sum_{i=1,3,4,\ldots}^{|Y|} \sqrt{\varepsilon_i}, \\
\varepsilon &\leq \varepsilon_3 + 2 \sum_{i=1,2,4,\ldots}^{|Y|} \sqrt{\varepsilon_i}, \\
&\vdots \\
\varepsilon &\leq \varepsilon_{|Y|} + 2 \sum_{i=1}^{|Y|-1} \sqrt{\varepsilon_i}.
\end{aligned}$$

If we take the average of all these inequalities, then

$$\varepsilon \leq \frac{1}{|Y|} \left(\sum_{i=1}^{|Y|} \varepsilon_i + 2\left(|Y| - 1\right) \sum_{i=1}^{|Y|} \sqrt{\varepsilon_i} \right). \tag{3.48}$$

Since the probability of Bob learning $f(x, y)$ for all y was established to be $1 - \varepsilon$, we finish our proof with a simple substitution:

$$\Pr\left[\text{Bob learns } f(x, y)\ \forall y\right] \geq 1 - \frac{1}{|Y|} \left(\sum_{i=1}^{|Y|} \varepsilon_i + 2\left(|Y| - 1\right) \sum_{i=1}^{|Y|} \sqrt{\varepsilon_i} \right). \tag{3.49}$$

□

3.1.2 Die-rolling

Die-rolling (DR) is a generalization of a popular two-party cryptographic primitive called coin-flipping. In coin-flipping, honest Alice and Bob communicate and effectively "flip a coin," such that their respective outputs are consistent - heads or tails, 0 or 1. Dishonest

Alice or Bob would try to influence the other party's output to their advantage. In weak coin-flipping, Alice and Bob are trying to influence the other's output toward a defined number, such as 0 for Alice and 1 for Bob. In strong coin-flipping, both parties are free to influence the other's output toward any number; Alice can try to influence Bob's output towards 0 or 1, and vice versa.

In DR, honest Alice and Bob attempt to agree on a value between 0 and $N-1$. Similar to coin-flipping, dishonest Alice and Bob would try to influence the output of the other party. In this thesis, we consider only DR protocols that perform as expected when both Alice and Bob are honest, i.e. each party's outcome distribution is uniformly random. See Figure 3.1 for an illustration of this primitive.

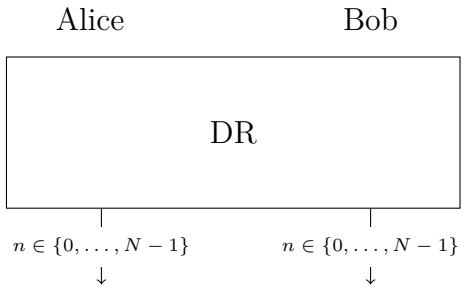

Figure 3.1: Die-rolling.

We define the following symbols for Alice and Bob's respective cheating probabilities in DR:

- $\mathrm{A}_{\mathrm{DR,n}}$: the maximum probability that cheating Alice can influence honest Bob's output to a number $n \in \{0, 1, \ldots, N-1\}$.

- $\mathrm{B}_{\mathrm{DR,n}}$: the maximum probability that cheating Bob can influence honest Alice's output to a number $n \in \{0, 1, \ldots, N-1\}$.

Kitaev proved in [15] the following inequalities for $N = 2$, i.e. strong coin-flipping:

$$\mathrm{A}_{\mathrm{DR},0}\mathrm{B}_{\mathrm{DR},0} \geq \frac{1}{2} \quad \text{and} \quad \mathrm{A}_{\mathrm{DR},1}\mathrm{B}_{\mathrm{DR},1} \geq \frac{1}{2}. \tag{3.50}$$

This proof generalizes to inequalities for greater N, namely,

$$\mathrm{A}_{\mathrm{DR,n}}\mathrm{B}_{\mathrm{DR,n}} \geq \frac{1}{N} \quad \forall n \in \{0, 1, \ldots, N-1\}. \tag{3.51}$$

Perfect security necessitates $\mathrm{A}_{\mathrm{DR,n}} = \mathrm{B}_{\mathrm{DR,n}} = \frac{1}{N}$. However, it is implied from the above bound that

$$\max\{\mathrm{A}_{\mathrm{DR,n}}, \mathrm{B}_{\mathrm{DR,n}}\} \geq \frac{1}{\sqrt{N}} \quad \forall n \in \{0, 1, \ldots, N-1\}. \tag{3.52}$$

3.1.3 Putting it all together

Now that the components of our proof have been described, we proceed with combining them.

We relate die-rolling and secure function evaluation to form what is called a reduction. This allows us to relate the cheating probabilities of die-rolling to those of secure function evaluation. This reduction is visualized in Figure 3.2. Refer back to Figure 1.1 for a visualization of SFE.

We describe this reduction at a high level. In this reduction, Alice and Bob start with standard SFE behavior: Alice and Bob's inputs x and y are chosen uniformly at random from sets X and Y, respectively. The SFE subroutine is performed, from which Bob learns $f(x, y)$. At this point, this protocol diverges from typical SFE. A value b from Bob's input

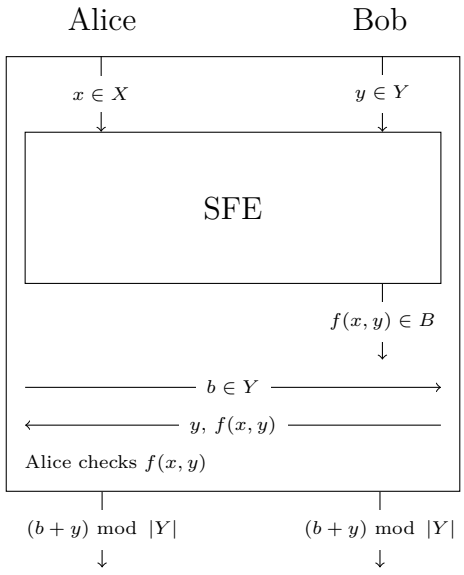

Figure 3.2: Die-rolling via secure function evaluation.

set Y is chosen uniformly at random, and Alice sends this value to Bob. Bob, in turn, sends Alice two values - his input y from the SFE protocol, and his output $f(x, y)$ from the SFE protocol. Alice checks the value of $f(x, y)$, and she can do so as she now has both input values, and knows the function f. If her computed value does not match what Bob sent, she has reasonable doubt as to Bob's honesty, and aborts the protocol. If she does not abort, both parties output $(b + y) \mod |Y|$.

This is a die-rolling reduction; here, Alice and Bob are trying to agree on a value $(b + y) \mod |Y|$. Assuming b and y are chosen uniformly at random, as well as the honesty of Alice and Bob, the output distribution will be uniformly random.

In order to derive our lower bound, we consider Bob's cheating probabilities in this reduction, firstly, $\mathrm{B}_{\mathrm{DR,n}}$. If dishonest Bob wants to influence the output of the protocol towards a value n, he needs to learn $f(x, y)$ for a y such that $(b + y) \mod |Y| = n$. For example, suppose $Y = \{0, 1, 2\}$ and Alice sends $b = 0$. If Bob's goal were for both parties to output the value 2, he would want to learn $f(x, y)$ where $y = 2$. However, Bob only realizes that he

must learn $f(x,2)$ after the SFE subroutine, and after Alice reveals b. Since b is uniformly random, for a desired outcome n, it can be seen that Bob has a 1 in $|Y|$ chance of needing to learn $f(x,y)$ such that $(b+y) \mod |Y| = n$. In the example described, b was 0, but it was equally likely for it to have been 1 or 2. Forcing an outcome of 2 means Bob would have had to learn $f(x,1)$ or $f(x,0)$, respectively. So Bob has a 1 in 3 chance of needing to learn $f(x,0)$, a 1 in 3 chance of needing to learn $f(x,1)$, and a 1 in 3 chance of needing to learn $f(x,2)$. The probability of him influencing the output towards 2 is given by the average of these probabilities:

$$\begin{aligned} \mathrm{B}_{\mathrm{DR},2} &= \frac{1}{3}\Pr\left[\text{Bob learns } f(x,0)\right] + \frac{1}{3}\Pr\left[\text{Bob learns } f(x,1)\right] + \frac{1}{3}\Pr\left[\text{Bob learns } f(x,2)\right] \\ &= \frac{1}{3}\sum_{y\in\{0,1,2\}} \Pr\left[\text{Bob learns } f(x,y)\right]. \end{aligned} \tag{3.53}$$

One can see this probability is the same, regardless of what value Bob wants him and Alice to output. Therefore, in general, the probability of him influencing the outcome towards a value n is given by

$$\mathrm{B}_{\mathrm{DR,n}} = \frac{1}{|Y|}\sum_{y\in Y} \Pr\left[\text{Bob learns } f(x,y)\right]. \tag{3.54}$$

Hopefully this probability looks a little familiar. Let us recall Lemma 3.6. For simplicity and portability, we also index the elements of Y here. Let $i \in \{1,2,\ldots,|Y|-1,|Y|\}$, and let y_i indicate the ith element of Bob's input set Y. Furthermore, let $\Pr\left[\text{Bob learns } f(x,y_i)\right] = 1-\varepsilon_i$ for each i. So we now have

$$\begin{aligned} \mathrm{B}_{\mathrm{DR,n}} &= \frac{1}{|Y|}\sum_{i=1}^{|Y|} \Pr\left[\text{Bob learns } f(x,y_i)\right] \\ &= \frac{1}{|Y|}\sum_{i=1}^{|Y|} \left(1-\varepsilon_i\right). \end{aligned} \tag{3.55}$$

Let us now formally define the inequality from Lemma 3.6 in terms of SFE. The probability that Bob learns $f(x, y)$ for all $y \in Y$ is given by $\mathrm{B}'_{\mathrm{SFE}}$, so we have

$$\mathrm{B}'_{\mathrm{SFE}} \geq 1 - \frac{1}{|Y|}\left(\sum_{i=1}^{|Y|} \varepsilon_i + 2\left(|Y| - 1\right)\sum_{i=1}^{|Y|} \sqrt{\varepsilon_i}\right). \tag{3.56}$$

We now prove a fact about sums of square roots.

Let x be the following vector:

$$x = \begin{bmatrix} \sqrt{x_1} \\ \sqrt{x_2} \\ \vdots \\ \sqrt{x_{n-1}} \\ \sqrt{x_n} \end{bmatrix}, \tag{3.57}$$

where $x_1, x_2, \ldots, x_{n-1}, x_n \geq 0$, and let y be an n-dimensional vector consisting entirely of 1s. By the Cauchy-Schwarz inequality, $\|x\|\,\|y\| \geq |\langle x, y\rangle|$. It can be seen that $|\langle x, y\rangle| = \sum_{i=1}^{n} \sqrt{x_i}$, $\|x\| = \sqrt{\sum_{i=1}^{n} x_i}$, and $\|y\| = \sqrt{n}$. Substituting in these quantities gives

$$\sum_{i=1}^{n} \sqrt{x_i} \leq \sqrt{n}\sqrt{\sum_{i=1}^{n} x_i}. \tag{3.58}$$

Using the proof of the inequality above, we have

$$\mathrm{B}'_{\mathrm{SFE}} \geq 1 - \frac{1}{|Y|}\left(\sum_{i=1}^{|Y|} \varepsilon_i + 2\left(|Y| - 1\right)\sqrt{|Y|}\sqrt{\sum_{i=1}^{|Y|} \varepsilon_i}\right). \tag{3.59}$$

Since $\varepsilon_i = 1 - \Pr\left[\text{Bob learns } f(x, y_i)\right]$, let us substitute in and simplify. Let p_i represent the probability Bob correctly learns $f(x, y_i)$ for a given i.

$$\begin{aligned}
\mathrm{B}'_{\mathrm{SFE}} &\geq 1 - \frac{1}{|Y|}\left(\sum_{i=1}^{|Y|}(1-p_i) + 2\left(|Y|-1\right)\sqrt{|Y|}\sqrt{\sum_{i=1}^{|Y|}(1-p_i)}\right) \\
&= 1 - \frac{1}{|Y|}\left(|Y| - \sum_{i=1}^{|Y|}p_i + 2\left(|Y|-1\right)\sqrt{|Y|}\sqrt{|Y| - \sum_{i=1}^{|Y|}p_i}\right) \\
&= 1 - \frac{1}{|Y|}\left(|Y| - \sum_{i=1}^{|Y|}p_i + 2\left(|Y|-1\right)|Y|\sqrt{1 - \frac{\sum_{i=1}^{|Y|}p_i}{|Y|}}\right) \\
&= 1 - \left(1 - \frac{\sum_{i=1}^{|Y|}p_i}{|Y|} + 2\left(|Y|-1\right)\sqrt{1 - \frac{\sum_{i=1}^{|Y|}p_i}{|Y|}}\right) \\
&= \frac{\sum_{i=1}^{|Y|}p_i}{|Y|} - 2\left(|Y|-1\right)\sqrt{1 - \frac{\sum_{i=1}^{|Y|}p_i}{|Y|}}.
\end{aligned} \tag{3.60}$$

Using our reduction, we substitute in $\mathrm{B}_{\mathrm{DR,n}} = \frac{1}{|Y|}\sum_{i=1}^{|Y|}p_i$ to get

$$\mathrm{B}'_{\mathrm{SFE}} \geq \mathrm{B}_{\mathrm{DR,n}} - 2\left(|Y|-1\right)\sqrt{1-\mathrm{B}_{\mathrm{DR,n}}}. \tag{3.61}$$

For this reduction, Kitaev's bound shows that

$$\mathrm{A}_{\mathrm{DR,n}}\mathrm{B}_{\mathrm{DR,n}} \geq \frac{1}{|Y|}, \tag{3.62}$$

which also translates to

$$\mathrm{B}_{\mathrm{DR,n}} \geq \frac{1}{|Y|\,\mathrm{A}_{\mathrm{DR,n}}}. \tag{3.63}$$

Substituting in, we have

$$\mathrm{B}'_{\mathrm{SFE}} \geq \frac{1}{|Y|\,\mathrm{A}_{\mathrm{DR,n}}} - 2\left(|Y|-1\right)\sqrt{1 - \frac{1}{|Y|\,\mathrm{A}_{\mathrm{DR,n}}}}. \tag{3.64}$$

We are almost done! We have not talked about Alice's goals in this reduction. It turns out that Alice's goals for the SFE subroutine and Alice's goals for the DR protocol are one and the same. For the SFE subroutine, Alice's goal is to learn Bob's input y. For the DR protocol, in order to influence the outcome towards a value n, Alice must learn Bob's input y after the SFE subroutine and before she sends b. If she learns y successfully, she can send b such that $(b + y) \mod |Y| = n$. As such, Alice's goal for the DR protocol aligns exactly with that of the SFE subroutine, so $\mathrm{A}_{\mathrm{SFE}} = \mathrm{A}_{\mathrm{DR,n}}$. One final substitution gives us our lower bound:

$$\mathrm{B}'_{\mathrm{SFE}} \geq \frac{1}{|Y|\,\mathrm{A}_{\mathrm{SFE}}} - 2\left(|Y| - 1\right)\sqrt{1 - \frac{1}{|Y|\,\mathrm{A}_{\mathrm{SFE}}}}. \tag{3.65}$$

Now we proceed in using this bound in quantum applications, both well-known and lesser-known, and some of the resulting constant lower bounds are the first of their kind.

Chapter 4

Discussion

In this chapter, we discuss how our lower bound applies to a variety of quantum applications. Each application includes a description, an illustration, a definition of symbols, the application of our lower bound, special cases, and a brief conclusion. In addition, XOR oblivious transfer contains a section describing an optimization on the application of the lower bound.

4.1 1-out-of-n oblivious transfer

The first of our applications is 1-out-of-n oblivious transfer. Here, Alice has an input "string" $(x_1, \ldots, x_n) \in W^{\times n}$ (where $|W|$ is finite). For example, if one were to scope this application to use bits, $W = \{0, 1\}$, and if one were to use integers from 1 to 5, $W = \{1, 2, 3, 4, 5\}$, etc. In any event, Bob has a choice from among the elements of the string; we call this input $y \in \{1, \ldots, n\}$. If Alice and Bob are both honest, at the end of the protocol, Bob learns x_y, or, the yth element of Alice's input. Ideally, Alice should not learn anything about y, and Bob should not learn anything more than x_y. Figure 4.1 illustrates this application. 1-out-of-n oblivious transfer is a generalization of 1-out-of-2 oblivious transfer, described in the introduction to this thesis.

To apply our lower bound, let us define the following symbols relevant to this protocol.

- B_{OT}: the maximum probability that cheating Bob can correctly learn honest Alice's

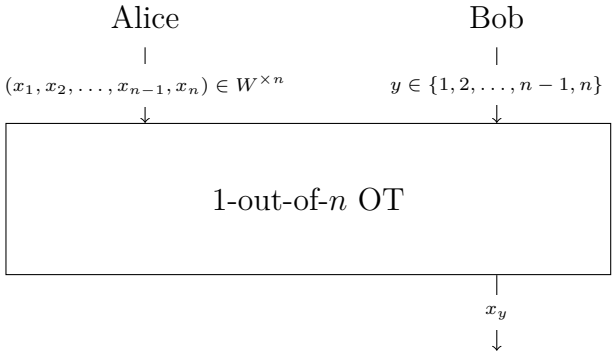

Figure 4.1: 1-out-of-n oblivious transfer.

input.

- $\mathrm{A_{OT}}$: the maximum probability that cheating Alice can correctly learn honest Bob's input.

It can be seen that 1-out-of-n oblivious transfer generalizes to SFE, which allows us to apply our lower bound. Here, $f(x, y) = x_y$. Alice and Bob's inputs map to their respective inputs in an SFE protocol. We also equate cheating probabilities. In SFE, Alice is trying to learn Bob's input y; in 1-out-of-n oblivious transfer, she is trying to do the same. So $\mathrm{A_{SFE}} = \mathrm{A_{OT}}$. In SFE, Bob is trying to learn $f(x, y)$ for all $y \in Y$. In 1-out-of-n oblivious transfer, Bob's goal is to learn x_y for all $y \in \{1, \ldots, n\}$. So $\mathrm{B'_{SFE}} = \mathrm{B_{OT}}$. Finally, we see that $|Y|$ is equal to n, the number of elements in Bob's input, and the length of Alice's input "string." It is now safe to apply our lower bound:

$$\mathrm{B_{OT}} \geq \frac{1}{n\mathrm{A_{OT}}} - 2(n-1)\sqrt{1 - \frac{1}{n\mathrm{A_{OT}}}}. \tag{4.1}$$

To evaluate a constant lower bound, we closely follow the assumptions and logic succeeding Theorem 3.2. First, we must recall the symbol $\mathrm{B'_{rand}}$, which indicates the maximum probability that cheating Bob can learn $f(x, y)$ for all $y \in Y$ with only black-box access to the

SFE protocol. For this SFE protocol, we have

$$\mathrm{B}'_{\mathrm{rand}} = \frac{1}{|W|^{n-1}}. \tag{4.2}$$

If Bob only has black-box access, the best he can do is learn x_y perfectly, and randomly guess the other $n-1$ values.

Recall also the constants c_A and c_B. Using Inequality (3.5) and Equation (3.7) as a template, we have that

$$\mathrm{B}_{\mathrm{OT}} = c_B \cdot \mathrm{B}'_{\mathrm{rand}} = \frac{c_B}{|W|^{n-1}} \quad \text{and} \quad \mathrm{A}_{\mathrm{OT}} \leq \frac{c_A}{|Y|} = \frac{c_A}{n}. \tag{4.3}$$

Having equated B_{OT} with $\mathrm{B}'_{\mathrm{SFE}}$ and A_{OT} with $\mathrm{A}_{\mathrm{SFE}}$, we substitute in and simplify Inequality (4.1), giving us

$$c_B \geq |W|^{n-1} \left(\frac{1}{c_A} - 2(n-1)\sqrt{1 - \frac{1}{c_A}} \right). \tag{4.4}$$

It is logical that an increase in one constant necessitates a decrease in the other. This tradeoff between c_A and c_B is illustrated in Figures 4.2 and 4.3.

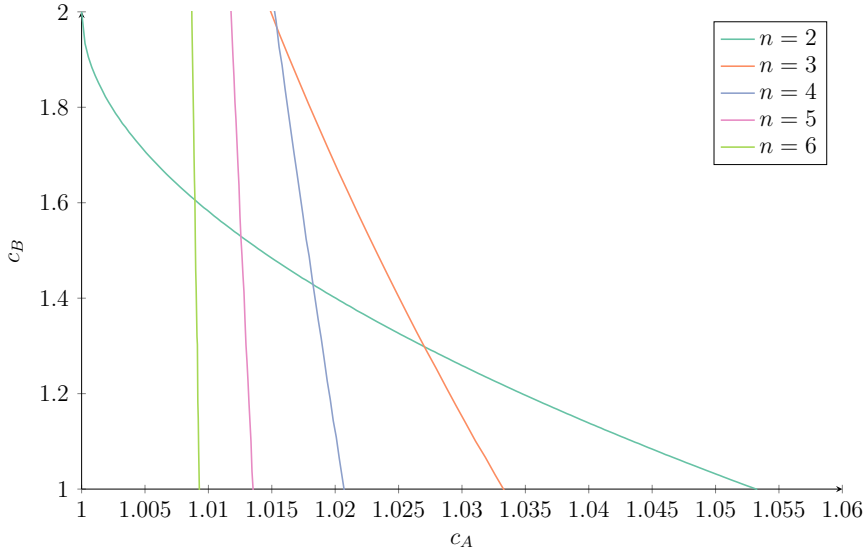

Figure 4.2: c_A vs. c_B for 1-out-of-n oblivious transfer, $|W| = 2$, and varying n.

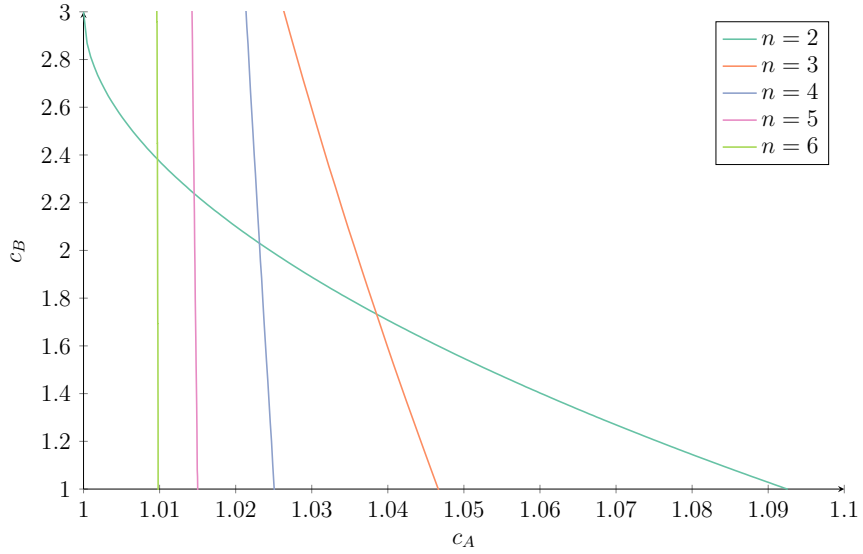

Figure 4.3: c_A vs. c_B for 1-out-of-n oblivious transfer, $|W| = 3$, and varying n.

We conclude our application of our lower bound by following the rationale behind Equation (3.10). Specifically, we introduce the constant c such that

$$c = |W|^{n-1}\left(\frac{1}{c} - 2(n-1)\sqrt{1-\frac{1}{c}}\right). \tag{4.5}$$

4.1.1 Special cases

Here, we present new bounds for popular derivatives of 1-out-of-n oblivious transfer.

1-out-of-2 (bit) OT. When $|W| = 2$, $n = 2$, we calculate c from (4.5) to get $c \approx 1.0484$. This implies

$$\mathrm{B_{OT}} \gtrapprox 0.5242 > 0.5 \quad \text{or} \quad \mathrm{A_{OT}} \gtrapprox 0.5242 > 0.5. \tag{4.6}$$

1-out-of-3 (bit) OT. When $|W| = 2$, $n = 3$, we calculate c from (4.5) to get $c \approx 1.0326$. This implies

$$\mathrm{B}_{\mathrm{OT}} \gtrapprox 0.2581 > 0.25 \quad \text{or} \quad \mathrm{A}_{\mathrm{OT}} \gtrapprox 0.3442 > 0.3333. \tag{4.7}$$

1-out-of-2 (trit) OT. When $|W| = 3$, $n = 2$, we calculate c from (4.5) to get $c \approx 1.085$. This implies that

$$\mathrm{B}_{\mathrm{OT}} \gtrapprox 0.3617 > 0.3333 \quad \text{or} \quad \mathrm{A}_{\mathrm{OT}} \gtrapprox 0.5425 > 0.5. \tag{4.8}$$

1-out-of-2 oblivious transfer with $|W| = 2^m$ has been studied previously, and constant lower bounds have been discovered. The novelty of our lower bound comes from it being applicable generally; it does not improve upon these bounds, but it does tread new ground in analysis of cases where $n > 2$, for example, and when $n = 2$ and $|W| \neq 2^m$.

4.2 k-out-of-n oblivious transfer

k-out-of-n oblivious transfer is a generalization of 1-out-of-n oblivious transfer, which is in itself a generalization of 1-out-of-2 oblivious transfer. And, like 1-out-of-n oblivious transfer, k-out-of-n oblivious transfer also generalizes to SFE, which allows us to apply our lower bound. In a k-out-of-n oblivious transfer protocol, Alice has an input "string" $(x_1, \ldots, x_n) \in W^{\times n}$, and Bob has a choice proper subset of that string, denoted $y \in Y = \{S : S \subset \{1, \ldots, n\}, |S| = k\}$, with the constraint on k that $0 < k < n$. Through this protocol, Bob learns the components of Alice's input corresponding to his input subset, i.e. $\{x_i : i \in y\}$, where x_i is the ith element of Alice's input. Ideally, Alice should not learn anything about y and Bob should not learn anything more than $\{x_i : i \in y\}$. Figure 4.4 illustrates this application.

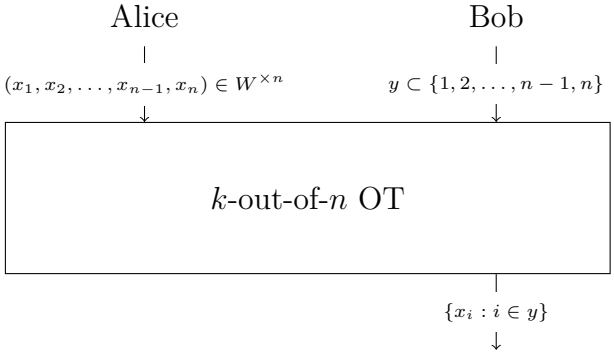

Figure 4.4: k-out-of-n oblivious transfer.

To apply our lower bound, let us define the following symbols relevant to this protocol.

- $\mathrm{B}_{\mathrm{knOT}}$: the maximum probability that cheating Bob can correctly learn honest Alice's input.

- $\mathrm{A}_{\mathrm{knOT}}$: the maximum probability that cheating Alice can correctly learn honest Bob's input.

As mentioned earlier, k-out-of-n oblivious transfer generalizes to SFE. This generalization allows us to apply our lower bound. In the context of SFE, $f(x, y) = \{x_i : i \in y\}$. Alice and Bob's inputs map to their respective inputs in an SFE protocol. We also equate cheating probabilities. In SFE, Alice is trying to learn Bob's input y; in k-out-of-n oblivious transfer, she is trying to do the same. So $\mathrm{A}_{\mathrm{SFE}} = \mathrm{A}_{\mathrm{knOT}}$. In SFE, Bob is trying to learn $f(x, y)$ for all $y \in Y$. In k-out-of-n oblivious transfer, Bob's goal is to learn x_j for all $j \in \{1, \ldots, n\}$, which he can accomplish by learning $\{x_i : i \in y\}$ for all $y \in Y$. This is overkill, as elements of a given input y may overlap with another k-sized proper subset in Y. But, we see $\mathrm{B}'_{\mathrm{SFE}} = \mathrm{B}_{\mathrm{knOT}}$. Finally, we see that $|Y|$ is equal to $\binom{n}{k}$, as it contains all possible combinations of k elements of Alice's input, which is of size n. It is now safe to apply our lower bound:

$$\mathrm{B}_{\mathrm{knOT}} \geq \frac{1}{\binom{n}{k}\mathrm{A}_{\mathrm{knOT}}} - 2\left(\binom{n}{k} - 1\right)\sqrt{1 - \frac{1}{\binom{n}{k}\mathrm{A}_{\mathrm{knOT}}}}. \tag{4.9}$$

We follow the same process as with 1-out-of-n oblivious transfer. For this SFE protocol, we have

$$\mathrm{B}'_{\mathrm{rand}} = \frac{1}{|W|^{n-k}}. \tag{4.10}$$

If Bob only has black-box access to the SFE protocol, the best he can do is learn k values of Alice's input, and randomly guess the other $n - k$ values.

At this point, we consider the constants c_A and c_B. Using Inequality (3.5) and Equation (3.7) as a template, we have that

$$\mathrm{B}_{\mathrm{knOT}} = c_B \cdot \mathrm{B}'_{\mathrm{rand}} = \frac{c_B}{|W|^{n-k}} \quad \text{and} \quad \mathrm{A}_{\mathrm{knOT}} \leq \frac{c_A}{|Y|} = \frac{c_A}{\binom{n}{k}}. \tag{4.11}$$

Having equated $\mathrm{B}_{\mathrm{knOT}}$ with $\mathrm{B}'_{\mathrm{SFE}}$ and $\mathrm{A}_{\mathrm{knOT}}$ with $\mathrm{A}_{\mathrm{SFE}}$, we substitute in and simplify Inequality (4.9), giving us

$$c_B \geq |W|^{n-k}\left(\frac{1}{c_A} - 2\left(\binom{n}{k} - 1\right)\sqrt{1 - \frac{1}{c_A}}\right). \tag{4.12}$$

The tradeoff between c_A and c_B is illustrated in Figures 4.5 and 4.6.

We conclude our application of our lower bound by following the rationale behind Equation (3.10). Specifically, we introduce the constant c such that

$$c = |W|^{n-k}\left(\frac{1}{c} - 2\left(\binom{n}{k} - 1\right)\sqrt{1 - \frac{1}{c}}\right). \tag{4.13}$$

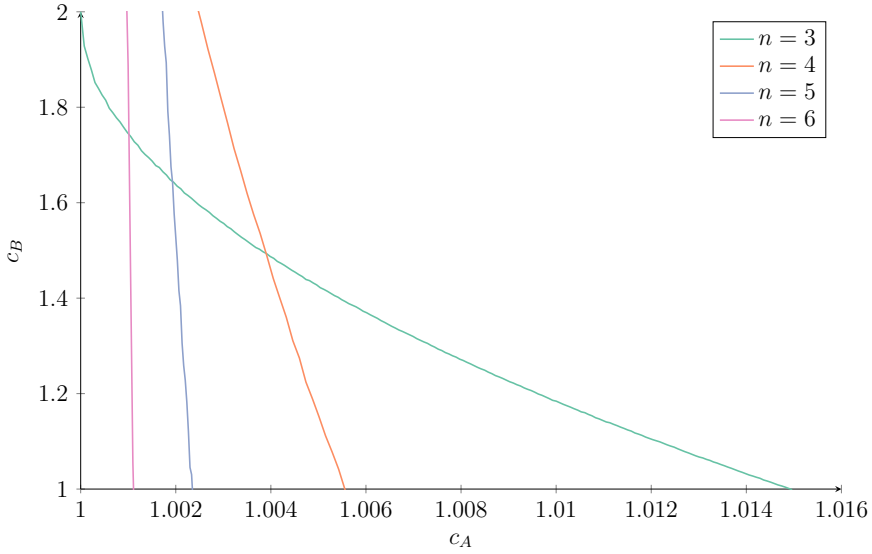

Figure 4.5: c_A vs. c_B for k-out-of-n oblivious transfer, $|W| = 2$, $k = 2$, and varying n.

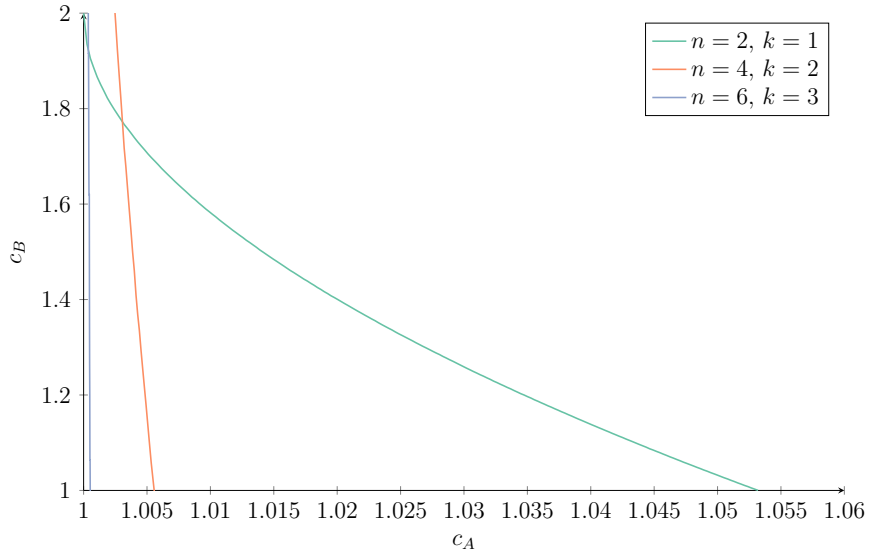

Figure 4.6: c_A vs. c_B for k-out-of-n oblivious transfer, $|W| = 2$, $k = \frac{n}{2}$, and varying n.

4.2.1 Special cases

Here, we present new bounds for derivations of k-out-of-n oblivious transfer.

2-out-of-3 (bit) OT. When $|W| = 2$, $n = 3$, and $k = 2$, we calculate c from (4.13) to get $c \approx 1.0145$. This implies that

$$\mathrm{B}_{\mathrm{knOT}} \gtrapprox 0.5073 > 0.5000 \quad \text{or} \quad \mathrm{A}_{\mathrm{knOT}} \gtrapprox 0.3382 > 0.3333. \tag{4.14}$$

2-out-of-4 (bit) OT. When $|W| = 2$, $n = 4$, and $k = 2$, we calculate c from (4.13) to get $c \approx 1.0056$. This implies that

$$\mathrm{B}_{\mathrm{knOT}} \gtrapprox 0.2514 > 0.2500 \quad \text{or} \quad \mathrm{A}_{\mathrm{knOT}} \gtrapprox 0.1676 > 0.1667. \tag{4.15}$$

3-out-of-4 (bit) OT. When $|W| = 2$, $n = 4$, and $k = 3$, we calculate c from (4.13) to get $c \approx 1.0067$. This implies that

$$\mathrm{B}_{\mathrm{knOT}} \gtrapprox 0.5034 > 0.5000 \quad \text{or} \quad \mathrm{A}_{\mathrm{knOT}} \gtrapprox 0.2517 > 0.2500. \tag{4.16}$$

As far as we know, the bounds we present here are the first of their kind for this application.

4.3 XOR oblivious transfer

Our next application is XOR oblivious transfer. In this application, Alice's inputs are two n-bit strings $x_1, x_2 \in \{0,1\}^n$. Bob's input is a choice $y \in \{1, 2, \oplus\}$, where $\oplus$ is the bitwise XOR of x_1 and x_2. A choice of $y = 1$ or $y = 2$ returns x_1 and x_2, respectively, and a choice of $\oplus$ returns the bitwise XOR of the two bit strings. Ideally, Alice should not learn anything about y and Bob should not learn anything more than x_1, x_2, or $x_1 \oplus x_2$, dependent on y. Figure 4.7 illustrates this application.

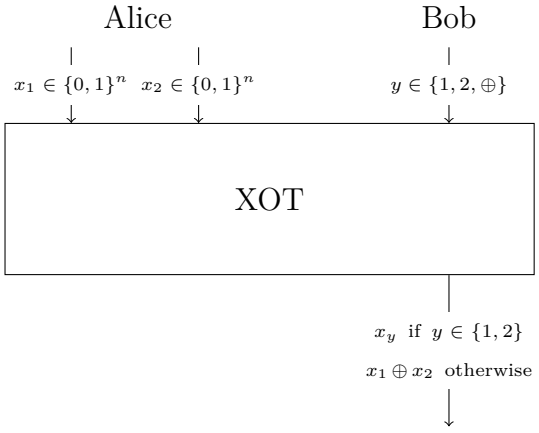

Figure 4.7: XOR oblivious transfer.

In order to apply our lower bound, we begin by defining the following symbols:

- B_{XOT}: the maximum probability that cheating Bob can correctly learn honest Alice's inputs x_1 and x_2.
- A_{XOT}: the maximum probability that cheating Alice can correctly learn honest Bob's input y.

Now, we generalize XOR oblivious transfer to a SFE protocol. In the context of SFE, we have that

$$f(x,y) = \begin{cases} x_y & \text{if } y \in \{1,2\}, \\ x_1 \oplus x_2 & \text{if } y = \oplus, \end{cases}$$

and we note that Alice and Bob's inputs map to their respective inputs in an SFE protocol. Now, we equate cheating probabilities. Alice's goal in an SFE protocol is to guess Bob's input. This goal aligns with Alice's goal in XOR oblivious transfer - Alice seeks to learn Bob's input y. As such, $A_{SFE} = A_{XOT}$. Bob's goal in an SFE protocol is to learn $f(x,y)$ for all $y \in Y$, his input set. In XOR oblivious transfer, Bob's goal is to learn x_1 and x_2, which he

can accomplish by learning $\{x_i : i \in y\}$ for all $y \in Y$. Similar to k-out-of-n oblivious transfer, Bob is overlearning, as learning two of the three possible outputs is enough to deduce the third. In fact, we optimize the analysis of the lower bound we conclude, and this is described in subsection 4.3.1. For now, it is apparent that $\mathrm{B}'_{\mathrm{SFE}} = \mathrm{B}_{\mathrm{XOT}}$. Finally, we see that $|Y| = 3$, as Bob has three choices for his input. Now we apply our lower bound:

$$\mathrm{B}_{\mathrm{XOT}} \geq \frac{1}{3\mathrm{A}_{\mathrm{XOT}}} - 4\sqrt{1 - \frac{1}{3\mathrm{A}_{\mathrm{XOT}}}}. \tag{4.17}$$

4.3.1 Optimizing our analysis

We optimize this lower bound for this application, since Bob only needs to learn two of the three possible outputs to deduce the other.

It is not as simple as substituting in the value 2 for $|Y|$ - Bob's input set is still the same size. This does not give us the bound we end up with. We must time travel all the way back to Lemma 3.6.

Suppose Bob learns $f(x, y_1) = x_1$ with probability $1 - \varepsilon_1$, $f(x, y_2) = x_2$ with probability $1 - \varepsilon_2$, and $f(x, y_3) = x_1 \oplus x_2$ with probability $1 - \varepsilon_3$. Bob only needs to make two sequential measurements to accomplish his goal laid out in the lemma of learning $f(x, y)$ for all y, i.e. $\mathrm{B}'_{\mathrm{SFE}}$, which we established was equivalent to $\mathrm{B}_{\mathrm{XOT}}$. Let us set this probability at $1 - \varepsilon$. Then, following the logic from the lemma, if he were to learn x_1 then measure again to get x_2,

$$1 - \varepsilon \geq 1 - \varepsilon_2 - 2\sqrt{\varepsilon_1}. \tag{4.18}$$

Alternatively, if he learned x_2 then x_1, it follows that

$$1 - \varepsilon \geq 1 - \varepsilon_1 - 2\sqrt{\varepsilon_2}. \tag{4.19}$$

This gives

$$\varepsilon \leq \varepsilon_2 + 2\sqrt{\varepsilon_1} \quad \text{and} \quad \varepsilon \leq \varepsilon_1 + 2\sqrt{\varepsilon_2}. \tag{4.20}$$

We get similar inequalities for if he were to learn x_1 and $x_1 \oplus x_2$, or x_2 and $x_1 \oplus x_2$. This gives a total of six different ways for him to learn $f(x, y)$ for all y using sequential measurements. The average of these six inequalities gives us

$$\varepsilon \leq \frac{1}{3}\left(\sum_{i=1}^{3} \varepsilon_i + 2\sum_{i=1}^{3} \sqrt{\varepsilon_i}\right). \tag{4.21}$$

Substituting in for ε gives

$$\mathrm{B_{XOT}} \geq 1 - \frac{1}{3}\left(\sum_{i=1}^{3} \varepsilon_i + 2\sum_{i=1}^{3} \sqrt{\varepsilon_i}\right). \tag{4.22}$$

Using the proof of Inequality (3.58), we have

$$\sum_{i=1}^{3} \sqrt{\varepsilon_i} \leq \sqrt{3}\sqrt{\sum_{i=1}^{3} \varepsilon_i}. \tag{4.23}$$

Thus,

$$\mathrm{B_{XOT}} \geq 1 - \frac{1}{3}\left(\sum_{i=1}^{3} \varepsilon_i + 2\sqrt{3}\sqrt{\sum_{i=1}^{3} \varepsilon_i}\right). \tag{4.24}$$

Now we substitute in $p_i = 1 - \varepsilon_i$, which represents the probability Bob learns $f(x, y_i)$, to

get:

$$\begin{aligned}
\mathrm{B}_{\mathrm{XOT}} &\geq 1 - \frac{1}{3}\left(\sum_{i=1}^{3}(1-p_i) + 2\sqrt{3}\sqrt{\sum_{i=1}^{3}(1-p_i)}\right) \\
&= 1 - \frac{1}{3}\left(3 - \sum_{i=1}^{3} p_i + 2\sqrt{3}\sqrt{3 - \sum_{i=1}^{3} p_i}\right) \\
&= 1 - \frac{1}{3}\left(3 - \sum_{i=1}^{3} p_i + 2(3)\sqrt{1 - \frac{\sum_{i=1}^{3} p_i}{3}}\right) \\
&= 1 - \left(1 - \frac{\sum_{i=1}^{3} p_i}{3} + 2\sqrt{1 - \frac{\sum_{i=1}^{3} p_i}{3}}\right) \\
&= \frac{\sum_{i=1}^{3} p_i}{3} - 2\sqrt{1 - \frac{\sum_{i=1}^{3} p_i}{3}}.
\end{aligned} \tag{4.25}$$

Consider the die-rolling reduction via XOR oblivious transfer illustrated in Figure 4.8.

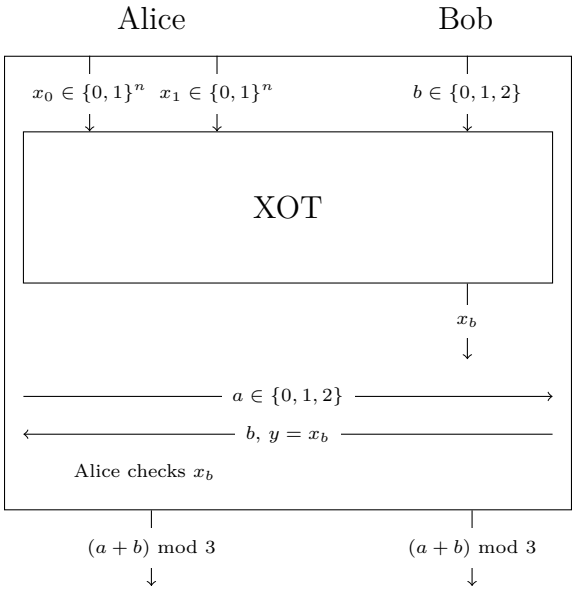

Figure 4.8: Die-rolling reduction via XOR oblivious transfer. Here we take x_2 to be $x_0 \oplus x_1$.

At the end of the XOT subroutine, Alice sends a uniformly random value a to Bob. Bob, in return, sends back b and the learned bit string x_b. Alice checks that x_b is correct, and if she

accepts it, they both output $(a+b)$ mod 3. Dishonest parties in die-rolling seek to influence the outcome of the other party to some value. For dishonest Alice, she can influence Bob's outcome by learning his input b prior to sending a - in XOR oblivious transfer, she want to learn b. So $\mathrm{A}_{\mathrm{DR,n}} = \mathrm{A}_{\mathrm{XOT}}$. For Bob to successfully influence the outcome toward some value n, he must learn the value of x_b such that $(a+b) \mod 3 = n$. a is uniformly random for honest Alice, so Bob has a 1 in 3 chance of needing to learn x_b for a given b. So, we define Bob's cheating probability as follows:

$$\mathrm{B}_{\mathrm{DR,n}} = \frac{1}{3}\sum_{i=0}^{2} \Pr\left(\text{Bob learns } x_i\right). \tag{4.26}$$

Adapting this probability to fit in Inequality (4.25) is trivial:

$$\mathrm{B}_{\mathrm{XOT}} \geq \mathrm{B}_{\mathrm{DR,n}} - 2\sqrt{1-\mathrm{B}_{\mathrm{DR,n}}}. \tag{4.27}$$

Applying Kitaev's bound gives

$$\mathrm{B}_{\mathrm{XOT}} \geq \frac{1}{3\mathrm{A}_{\mathrm{DR,n}}} - 2\sqrt{1-\frac{1}{3\mathrm{A}_{\mathrm{DR,n}}}}, \tag{4.28}$$

and substituting $\mathrm{A}_{\mathrm{XOT}}$ in for $\mathrm{A}_{\mathrm{DR,n}}$ gives an improved lower bound for XOR oblivious transfer:

$$\mathrm{B}_{\mathrm{XOT}} \geq \frac{1}{3\mathrm{A}_{\mathrm{XOT}}} - 2\sqrt{1-\frac{1}{3\mathrm{A}_{\mathrm{XOT}}}}. \tag{4.29}$$

With an improved lower bound in hand, we continue in following the same process as in preceding applications. We have

$$\mathrm{B}'_{\mathrm{rand}} = \frac{1}{2^n}. \tag{4.30}$$

Considering SFE as a generalization of XOR oblivious transfer, if Bob only has black-box

access, the best he can do is learn one of Alice's input strings with certainty, and randomly guess the other, which he does correctly with probability $\frac{1}{2^n}$.

At this point, we consider the constants c_A and c_B. Using Inequality (3.5) and Equation (3.7) as a template, we have that

$$\mathrm{B}_{\mathrm{XOT}} = c_B \cdot \mathrm{B}'_{\mathrm{rand}} = \frac{c_B}{2^n} \quad \text{and} \quad \mathrm{A}_{\mathrm{XOT}} \leq \frac{c_A}{|Y|} = \frac{c_A}{3}. \tag{4.31}$$

Having equated $\mathrm{B}_{\mathrm{XOT}}$ with $\mathrm{B}'_{\mathrm{SFE}}$ and $\mathrm{A}_{\mathrm{XOT}}$ with $\mathrm{A}_{\mathrm{SFE}}$, we substitute in and simplify Inequality (4.17), giving us

$$c_B \geq 2^n \left(\frac{1}{c_A} - 2\sqrt{1 - \frac{1}{c_A}} \right). \tag{4.32}$$

The tradeoff between c_A and c_B is illustrated in Figure 4.9.

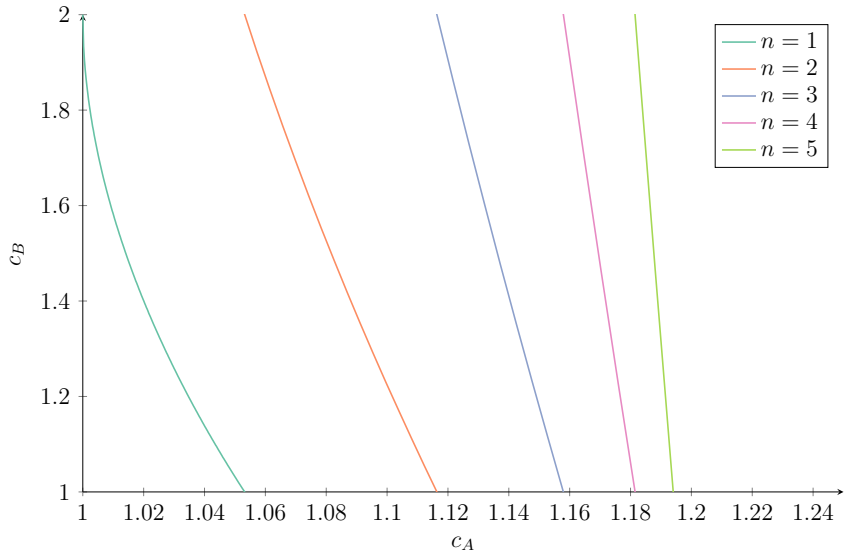

Figure 4.9: c_A vs. c_B for XOR oblivious transfer and varying n.

We conclude our application of our lower bound by following the rationale behind Equation (3.10). Specifically, we introduce the constant c such that

$$c = 2^n \left(\frac{1}{c} - 2\sqrt{1 - \frac{1}{c}} \right). \tag{4.33}$$

4.3.2 Special cases

Here, we present new bounds for derivatives of XOR oblivious transfer.

1-bit XOR oblivious transfer. When $n = 1$, we calculate c from (4.33) to get $c \approx 1.04838$. This implies that

$$\mathrm{B}_{\mathrm{XOT}} \gtrapprox 0.5242 > 0.5 \quad \text{or} \quad \mathrm{A}_{\mathrm{XOT}} \gtrapprox 0.3495 > 0.3333. \tag{4.34}$$

2-bit XOR oblivious transfer. When $n = 2$, we calculate c from (4.33) to get $c \approx 1.1083$. This implies that

$$\mathrm{B}_{\mathrm{XOT}} \gtrapprox 0.2771 > 0.25 \quad \text{or} \quad \mathrm{A}_{\mathrm{XOT}} \gtrapprox 0.3694 > 0.3333. \tag{4.35}$$

To the best of our knowledge, the lower bounds above are the first of their kind.

4.4 Equality function

In the equality function application, Bob determines whether his input is equal to Alice's. Alice has an input $x \in X$ and Bob has an input $y \in Y$. For simplicity, let us assume $X = Y$, and $|X| = |Y| = n$. The output of the protocol is the Kronecker delta function, denoted δ_{xy}, which returns 1 if x and y are equal, and 0 otherwise. Ideally, Alice should not learn anything about y, and Bob should not learn anything about Alice's input other than what follows logically from the function's output. Figure 4.10 illustrates this application.

To apply our lower bound, let us define the following symbols relevant to this protocol.

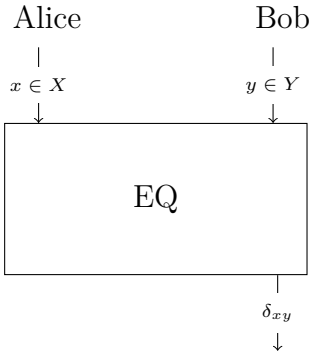

Figure 4.10: Equality function.

- $\mathrm{B_{EQ}}$: the maximum probability that cheating Bob can correctly learn honest Alice's input x.

- $\mathrm{A_{EQ}}$: the maximum probability that cheating Alice can correctly learn honest Bob's input y.

We generalize the equality function to an SFE protocol, which allows us to apply our lower bound. Here, $f(x, y) = \delta_{xy}$, and Alice and Bob's inputs map to their respective inputs in an SFE protocol. We now equate cheating probabilities. In SFE, Alice is trying to learn Bob's input y; in the equality function, she is trying to do the same. So $\mathrm{A_{SFE}} = \mathrm{A_{EQ}}$. In SFE, Bob is trying to learn $f(x, y)$ for all $y \in Y$. In the equality function, Bob's goal is to learn δ_{xy} for all $y \in Y$. So $\mathrm{B'_{SFE}} = \mathrm{B_{EQ}}$. The size of Bob's input set in SFE is equal to the size of Bob's input set in the equality function, i.e. $|Y| = n$.

We now apply the lower bound, and we get the following inequality:

$$\mathrm{B_{EQ}} \geq \frac{1}{n\mathrm{A_{EQ}}} - 2\,(n-1)\sqrt{1 - \frac{1}{n\mathrm{A_{EQ}}}}. \tag{4.36}$$

We follow the same process as in preceding applications. By generalizing to SFE, we have

$$\mathrm{B}'_{\mathrm{rand}} = \frac{2}{n}. \tag{4.37}$$

How we arrive at this value is not immediately clear. The function δ_{xy} returns 1 when Alice and Bob's inputs match, and this occurs with probability $\frac{1}{n}$. This means the probability their inputs do not match is $\frac{n-1}{n}$. If this occurs, Bob has a $\frac{1}{n-1}$ chance of correctly guessing Alice's input. Combining the two cases, Bob's probability of guessing Alice's input with only black-box access to the protocol is given by $\frac{1}{n} + \frac{n-1}{n} \cdot \frac{1}{n-1} = \frac{2}{n}$.

At this point, we consider the constants c_A and c_B. Using Inequality (3.5) and Equation (3.7) as a template, we have that

$$\mathrm{B}_{\mathrm{EQ}} = c_B \cdot \mathrm{B}'_{\mathrm{rand}} = \frac{2c_B}{n} \quad \text{and} \quad \mathrm{A}_{\mathrm{EQ}} \leq \frac{c_A}{|Y|} = \frac{c_A}{n}. \tag{4.38}$$

There is an interesting case when $n = 2$. If δ_{xy} returns 0, Bob knows that his and Alice's inputs are not equal, but since there is only one other option for Alice's input, her input is deduced anyway. This necessitates $\mathrm{B}_{\mathrm{EQ}} = 1$.

Having equated B_{EQ} with $\mathrm{B}'_{\mathrm{SFE}}$ and A_{EQ} with $\mathrm{A}_{\mathrm{SFE}}$, we substitute in and simplify Inequality (4.36), giving us

$$c_B \geq \frac{n}{2}\left(\frac{1}{c_A} - 2\,(n-1)\sqrt{1 - \frac{1}{c_A}}\right). \tag{4.39}$$

The tradeoff between c_A and c_B is illustrated in Figure 4.11.

We conclude our application of our lower bound by following the rationale behind Equa-

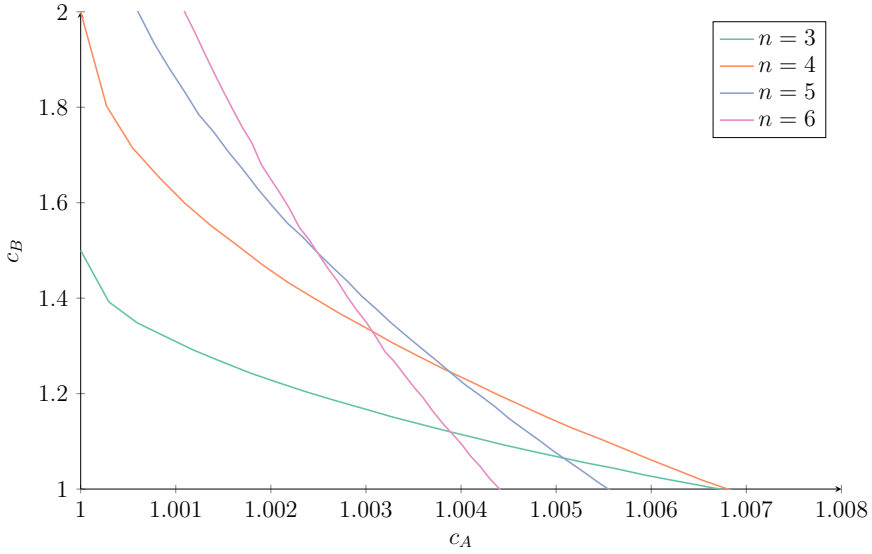

Figure 4.11: c_A vs. c_B for the equality function and varying n.

tion (3.10). Specifically, we introduce the constant c such that

$$c = \frac{n}{2}\left(\frac{1}{c} - 2\,(n-1)\sqrt{1-\frac{1}{c}}\right). \tag{4.40}$$

4.4.1 Special case

Here, we present new bounds for derivatives of the equality function.

Equality function for $n = 3$. When $n = 3$, we calculate c from (4.40) to get $c \approx 1.0065$. This implies that

$$\mathrm{B_{EQ}} \gtrapprox 0.671 > 0.667 \quad \text{or} \quad \mathrm{A_{EQ}} \gtrapprox 0.3355 > 0.3333. \tag{4.41}$$

To the best of our knowledge, the lower bounds for this application are the first of their kind.

4.5 Inner product function

The next application is the inner product function. Alice has an input string $x \in \{0,1\}^n$, and Bob has an input string $y \in \{0,1\}^n \setminus \{0\}^n$. Bob's output is the inner product function of these two inputs, defined as

$$x \cdot y := \sum_{i=1}^{n} x_i y_i \mod 2, \tag{4.42}$$

where x_i is the ith bit of x, and y_i is the ith bit of y. Ideally, Alice should not learn anything about Bob's input, and Bob should not learn anything about Alice's input other than what follows naturally from the inner product function's output. Figure 4.12 illustrates this application.

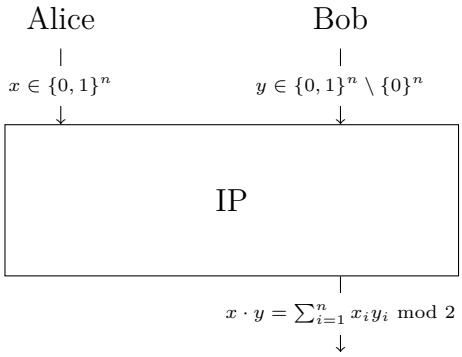

Figure 4.12: Inner product function.

To apply our lower bound, let us define the following symbols relevant to this protocol.

- B_{IP}: the maximum probability that cheating Bob can correctly learn honest Alice's input x.

- A_{IP}: the maximum probability that cheating Alice can correctly learn honest Bob's input y.

We now generalize the inner product function to an SFE protocol. Note that $f(x, y) = x \cdot y$, and Alice and Bob's inputs map to their respective inputs in an SFE protocol. Let us now equate cheating probabilities. In SFE, Alice is trying to learn Bob's input y; in the inner product function, she is trying to do the same. So $\mathrm{A}_{\mathrm{SFE}} = \mathrm{A}_{\mathrm{IP}}$. In SFE, Bob is trying to learn $f(x, y)$ for all $y \in Y$. With the inner product function, Bob can learn Alice's input by learning $x \cdot y$ for all $y \in Y$. This, however, is overkill. If Bob learns the inner product of Alice's string with n distinct inputs, each having only one bit set, he can deduce Alice's string. But, it is apparent that $\mathrm{B}'_{\mathrm{SFE}} = \mathrm{B}_{\mathrm{IP}}$. Finally, we see that $|Y| = 2^n - 1$, as the 0 string is excluded from his input set. Now we apply our lower bound:

$$\mathrm{B}_{\mathrm{IP}} \geq \frac{1}{(2^n - 1)\mathrm{A}_{\mathrm{IP}}} - 2(2^n - 2)\sqrt{1 - \frac{1}{(2^n - 1)\mathrm{A}_{\mathrm{IP}}}}. \tag{4.43}$$

We follow the same process as in preceding applications. By generalizing to SFE, we have

$$\mathrm{B}'_{\mathrm{rand}} = \frac{2}{2^n}. \tag{4.44}$$

There are two cases here for Bob, who has black-box access to this protocol. In both cases, the possibilities for Alice's input are halved. For any nonzero y, $x \cdot y$ has a 50% chance of returning 0 or 1. So from the output, Bob's chance of correctly guessing Alice's input with only black-box access is given by $\frac{1}{2^{n-1}} = \frac{2}{2^n}$.

At this point, we consider the constants c_A and c_B. Using Inequality (3.5) and Equation (3.7) as a template, we have that

$$\mathrm{B}_{\mathrm{IP}} = c_B \cdot \mathrm{B}'_{\mathrm{rand}} = \frac{2c_B}{2^n} \quad \text{and} \quad \mathrm{A}_{\mathrm{IP}} \leq \frac{c_A}{|Y|} = \frac{c_A}{2^n - 1}. \tag{4.45}$$

Having equated B_{IP} with $\mathrm{B}'_{\mathrm{SFE}}$ and A_{IP} with $\mathrm{A}_{\mathrm{SFE}}$, we substitute in and simplify Inequal-

ity (4.43), giving us

$$c_B \geq \frac{2^n}{2}\left(\frac{1}{c_A} - 2\left(2^n - 2\right)\sqrt{1 - \frac{1}{c_A}}\right). \tag{4.46}$$

The tradeoff between c_A and c_B is illustrated in Figure 4.13.

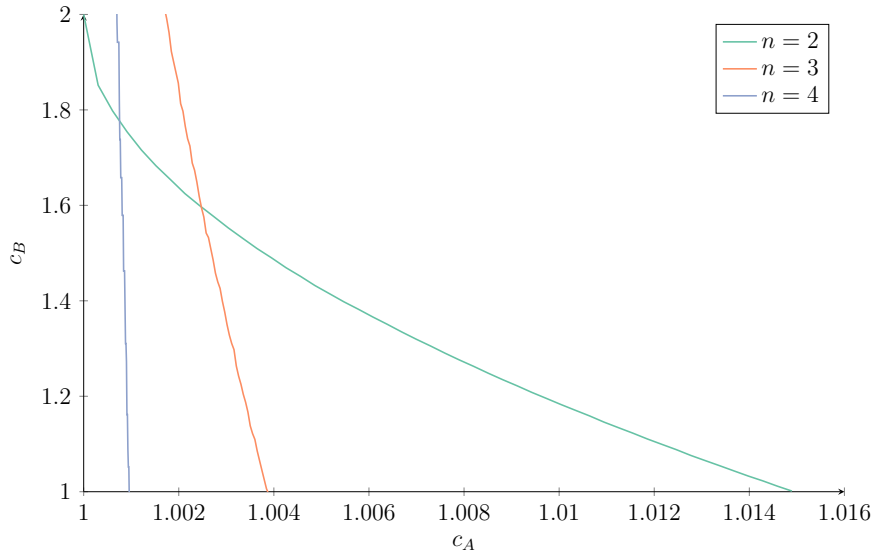

Figure 4.13: c_A vs. c_B for the inner product function and varying n.

We conclude our application of our lower bound by following the rationale behind Equation (3.10). Specifically, we introduce the constant c such that

$$c = \frac{2^n}{2}\left(\frac{1}{c} - 2\left(2^n - 2\right)\sqrt{1 - \frac{1}{c}}\right). \tag{4.47}$$

4.5.1 Special case

Here, we present new bounds for derivatives of the inner product function.

Inner product function for $n = 3$. When $n = 3$, we calculate c from (4.47) to get $c \approx 1.0039$. This implies that

$$\mathrm{B}_{\mathrm{IP}} \gtrapprox 0.251 > 0.25 \quad \text{or} \quad \mathrm{A}_{\mathrm{IP}} \gtrapprox 0.1434 > 0.1429. \tag{4.48}$$

To the best of our knowledge, the lower bounds presented here for this application are the first of their kind.

4.6 Millionaire's problem

The next application is Andrew Yao's millionaires' problem. Here, Bob wants to determine whether he has the same or more money than Alice. Alice has an input $x \in \{1, 2, \ldots, n-1, n\}$, and Bob has an input $y \in \{1, 2, \ldots, n-1\}$. Bob's output returns 1 if $y \geq x$ and 0 otherwise. Bob's input restricts a value n. We pose that honest Bob would not play if he had the maximum amount of money, and dishonest Bob would choose a value that would give him more information, noting that $f(x, n) = 1$ for all x. Ideally, Alice does not learn anything about Bob's input and Bob does not learn anything about Alice's input other than whether it is the same or less than his input. Figure 4.14 illustrates this application.

To apply our lower bound, let us define the following symbols relevant to this protocol.

- $\mathrm{B}_{☺_\$}$: the maximum probability that cheating Bob can correctly learn honest Alice's input x.
- $\mathrm{A}_{☺_\$}$: the maximum probability that cheating Alice can correctly learn honest Bob's input y.

Similar to the other applications, the millionaires' problem generalizes to an SFE protocol.

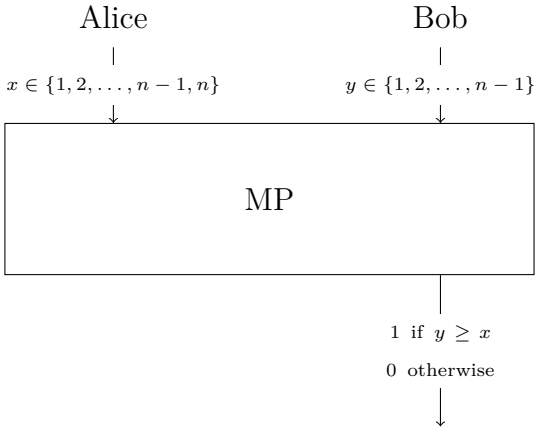

Figure 4.14: Millionaires' problem.

Here, we see

$$f(x, y) = \begin{cases} 1 & \text{if } y \geq x, \\ 0 & \text{otherwise.} \end{cases} \tag{4.49}$$

Alice and Bob's inputs map to their respective inputs in an SFE protocol. Now, we equate cheating probabilities. In SFE, Alice is trying to learn Bob's input y; in the millionaires' problem, she is trying to do the same. So $\mathrm{A}_{\mathrm{SFE}} = \mathrm{A}_{☺\$}$. In SFE, Bob is trying to learn $f(x, y)$ for all $y \in Y$. In the millionaires' problem, Bob can learn Alice's input x by learning the output of the above function for all values of his input y. So, it can be seen that $\mathrm{B}'_{\mathrm{SFE}} = \mathrm{B}_{☺\$}$. Finally, $|Y| = n - 1$, the number of possible inputs for Bob. We now apply our lower bound:

$$\mathrm{B}_{☺\$} \geq \frac{1}{(n-1)\,\mathrm{A}_{☺\$}} - 2\,(n-2)\sqrt{1 - \frac{1}{(n-1)\,\mathrm{A}_{☺\$}}}. \tag{4.50}$$

We follow the same process as in preceding applications. By generalizing to SFE, we have

$$\mathrm{B}'_{\mathrm{rand}} = \frac{2}{n}. \tag{4.51}$$

For black-box Bob with a given input y, he sees $f(x, y) = 1$ with probability $\frac{y}{n}$. When this

occurs, his resort is to randomly guess Alice's input as a value in $\{1, \ldots, y\}$. Black-box Bob sees $f(x, y) = 0$ with probability $\frac{n-y}{n}$. When this occurs, his resort is to randomly guess Alice's input as a value in $\{y + 1, \ldots, n\}$.

Combining these two cases, we have

$$\mathrm{B}'_{\mathrm{rand}} = \frac{n-y}{n} \cdot \frac{1}{n-y} + \frac{y}{n} \cdot \frac{1}{y} = \frac{2}{n}. \tag{4.52}$$

At this point, we consider the constants c_A and c_B. Using Inequality (3.5) and Equation (3.7) as a template, we have that

$$\mathrm{B}_{☺\$} = c_B \cdot \mathrm{B}'_{\mathrm{rand}} = \frac{2c_B}{n} \quad \text{and} \quad \mathrm{A}_{☺\$} \leq \frac{c_A}{|Y|} = \frac{c_A}{n-1}. \tag{4.53}$$

Having equated $\mathrm{B}_{☺\$}$ with $\mathrm{B}'_{\mathrm{SFE}}$ and $\mathrm{A}_{☺\$}$ with $\mathrm{A}_{\mathrm{SFE}}$, we substitute in and simplify Inequality (4.50), giving us

$$c_B \geq \frac{n}{2}\left(\frac{1}{c_A} - 2\,(n-2)\sqrt{1 - \frac{1}{c_A}}\right). \tag{4.54}$$

The tradeoff between c_A and c_B is illustrated in Figure 4.15.

We conclude our application of our lower bound by following the rationale behind Equation (3.10). Specifically, we introduce the constant c such that

$$c = \frac{n}{2}\left(\frac{1}{c} - 2\,(n-2)\sqrt{1 - \frac{1}{c}}\right). \tag{4.55}$$

4.6.1 Special cases

Here, we present new bounds for derivatives of the millionaires' problem.

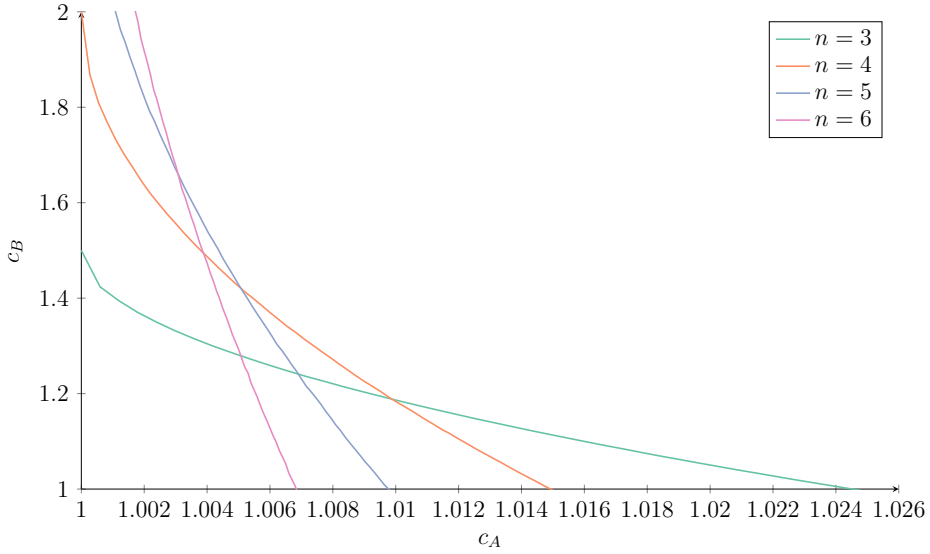

Figure 4.15: c_A vs. c_B for the millionaire's problem and varying n.

Millionaire's problem for $n = 10^9$. When $n = 10^9$, we calculate c from (4.55) to get $c \approx 1 + 2.5 \times 10^{-19}$. This implies that

$$\begin{aligned} \mathrm{B}_{\text{☺}_\$} &\gtrapprox 2 \times 10^{-9} + 5 \times 10^{-28} > 2 \times 10^{-9} \quad \text{or} \\ \mathrm{A}_{\text{☺}_\$} &\gtrapprox 1 \times 10^{-9} + 1 \times 10^{-18} + 1.25 \times 10^{-27} > 1 \times 10^{-9} + 1 \times 10^{-18} + 1 \times 10^{-27}. \end{aligned} \tag{4.56}$$

Here we have capped the wealth of Alice and Bob at a billion dollars, but our bound works for any cap.

Millionaire's problem, academics' version. When $n = 10^1$, we calculate c from (4.55) to get $c \approx 1.0025$. This implies that

$$\mathrm{B}_{\text{☺}_\$} \gtrapprox 0.2005 > 0.2 \quad \text{or} \quad \mathrm{A}_{\text{☺}_\$} \gtrapprox 0.1114 > 0.1111. \tag{4.57}$$

To the best of our knowledge, the lower bounds presented for this application are the first of their kind.

4.7 Secret phrase

In the secret phrase application, Alice has a tuple $(i, c) \in W \times N$, and Bob has an input $y \in N$. Bob's output is i if $c = y$, and NULL otherwise. Ideally, Alice does not learn anything about Bob's input and Bob does not learn anything about i, excepting the case when $c = y$. Figure 4.16 illustrates this application.

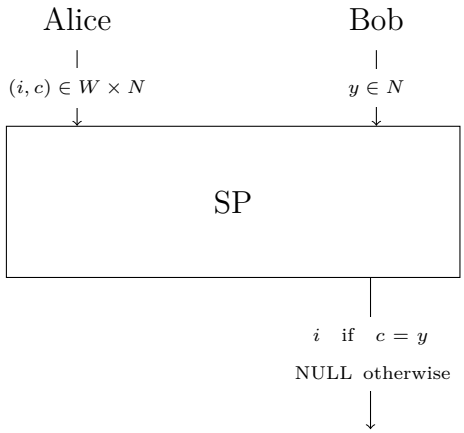

Figure 4.16: Secret phrase.

To apply our lower bound, let us define the following symbols relevant to this protocol.

- $\widetilde{\mathrm{B}}_{\mathrm{SP}}$: the maximum probability that cheating Bob can correctly learn the i part of honest Alice's input.

- $\mathrm{B}'_{\mathrm{SP}}$: the maximum probability that cheating Bob can correctly learn honest Alice's input (i, c).

- A_{SP}: the maximum probability that cheating Alice can correctly learn honest Bob's input y.

The secret phrase application also generalizes to an SFE protocol. Here, we see

$$f(i,c) = \begin{cases} i & \text{if } c = y, \\ \text{NULL} & \text{otherwise.} \end{cases} \tag{4.58}$$

Alice and Bob's inputs map to their respective inputs in an SFE protocol. Now, we equate cheating probabilities. In SFE, Alice is trying to learn honest Bob's input y. In this application, Alice is trying to do the same. So $\mathrm{A}_{\mathrm{SFE}} = \mathrm{A}_{\mathrm{SP}}$. In SFE, Bob is trying to learn $f(x,y)$ for all $y \in Y$. In this application, Bob learning $f(x,y)$ for every input $y \in N$ means he will have learned (i,c). So, it is seen that $\mathrm{B}'_{\mathrm{SFE}} = \mathrm{B}'_{\mathrm{SP}}$. However, in the secret phrase application, Bob's goal is to learn i; he does not care about learning c. So $\widetilde{\mathrm{B}}_{\mathrm{SP}}$ more accurately reflects his goal. We can say $\widetilde{\mathrm{B}}_{\mathrm{SP}} \geq \mathrm{B}'_{\mathrm{SP}} = \mathrm{B}'_{\mathrm{SFE}}$, since Bob knowing (i,c) implies him knowing i. Finally, $|Y| = |N|$. We now apply our lower bound:

$$\widetilde{\mathrm{B}}_{\mathrm{SP}} \geq \mathrm{B}'_{\mathrm{SP}} \geq \frac{1}{|N|\,\mathrm{A}_{\mathrm{SP}}} - 2\left(|N| - 1\right)\sqrt{1 - \frac{1}{|N|\,\mathrm{A}_{\mathrm{SP}}}}. \tag{4.59}$$

To accompany the new cheating probability for Bob, we also introduce the corresponding black-box cheating probability:

- $\widetilde{\mathrm{B}}_{\mathrm{rand}}$: the maximum probability that cheating Bob can correctly learn the i part of honest Alice's input with only black-box access to the SFE protocol.

Note that $\widetilde{\mathrm{B}}_{\mathrm{rand}} \geq \mathrm{B}'_{\mathrm{rand}}$ by the same logic supporting $\widetilde{\mathrm{B}}_{\mathrm{SP}} \geq \mathrm{B}'_{\mathrm{SP}}$.

We follow the same process as in preceding applications. By generalizing to SFE, we have

$$\widetilde{\mathrm{B}}_{\mathrm{rand}} = \frac{1}{|N|} + \frac{|N| - 1}{|N|}\left(\frac{1}{|W|}\right). \tag{4.60}$$

The case of Bob's input y equalling c occurs with probability $\frac{1}{|N|}$; the probability of y not equalling c is $\frac{|N|-1}{|N|}$. In the latter case, Bob has a $\frac{1}{|W|}$ chance of correctly guessing i. Combining these probabilities gives the equation above.

At this point, we consider the constant c_A and introduce a constant $\widetilde{c}_B \geq 1$. Using Inequality (3.5) and Equation (3.7) as a template, we can set

$$\widetilde{\mathrm{B}}_{\mathrm{SP}} = \widetilde{c}_B \cdot \widetilde{\mathrm{B}}_{\mathrm{rand}} \quad \text{and} \quad \mathrm{A}_{\mathrm{SP}} \leq \frac{c_A}{|Y|} = \frac{c_A}{|N|}. \tag{4.61}$$

Having related $\widetilde{\mathrm{B}}_{\mathrm{SP}}$ with $\mathrm{B}'_{\mathrm{SFE}}$ and equated A_{SP} with $\mathrm{A}_{\mathrm{SFE}}$, we substitute in and simplify Inequality (4.59), giving us

$$\widetilde{c}_B \geq \frac{|W|\,|N|}{|W|+|N|-1}\left(\frac{1}{c_A} - 2\left(|N|-1\right)\sqrt{1-\frac{1}{c_A}}\right). \tag{4.62}$$

The tradeoff between c_A and $\widetilde{c}_B$ is illustrated in Figure 4.17.

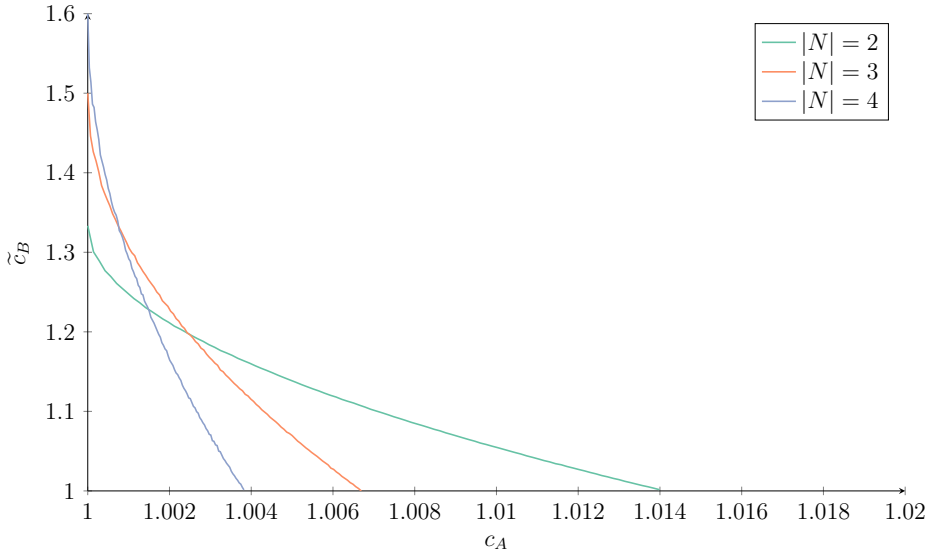

Figure 4.17: c_A vs. $\widetilde{c}_B$ for the secret phrase application with fixed $|W| = 2$ and varying $|N|$.

We conclude our application of our lower bound by following the rationale behind Equa-

tion (3.10). Specifically, we introduce the constant c such that

$$c = \frac{|W|\,|N|}{|W| + |N| - 1}\left(\frac{1}{c} - 2\left(|N| - 1\right)\sqrt{1 - \frac{1}{c}}\right). \tag{4.63}$$

4.7.1 Special cases

Here, we present new bounds for derivatives of the secret phrase application.

Rabin-OT. In Rabin's oblivious transfer protocol, Alice sends Bob a message with probability $\frac{1}{2}$, but Alice does not know whether Bob received a message [4]. As such, we can derive a constant lower bound for Rabin-OT through this secret phase protocol by setting $|W| = 2$ and $|N| = 2$. We calculate c from (4.63) to get $c \approx 1.01308$. This implies that

$$\widetilde{\mathrm{B}}_{\mathrm{SP}} \gtrapprox 0.7598 > 0.75 \quad \text{or} \quad \mathrm{A}_{\mathrm{SP}} \gtrapprox 0.5065 > 0.5. \tag{4.64}$$

James Bobd. Sean Bobbery, Bober Moore, Pierce Bobsnan - take your pick; James Bobd is back and he has a new mission! Alice is Ralph Fiennes, the current M. Bobd drew outside the lines a little on his last mission, so Ralph is not the happiest with him, and he tells Bobd he has given his new assignment and the relevant mission information to his coworker Sean Bean, code-named 006.

To access mission information, an MI6 agent must relay their valid 5-letter catchphrase to MI6 via payphone, who then provides the agent with their mission-essential information. If the catchphrase is invalid, the agent learns nothing. To catch the bad guy, Bobd needs the mission information, but it is now locked behind Sean Bean's catchphrase. Ralph wants to know whether that information was accessed, so he can check with Sean Bean that Bobd has not figured out his catchphrase. If he catches Bobd, Ralph can gleefully put him on

paperwork duty under close supervision.

Here, we set $|W|$ arbitrarily to 10, assuming there are only 10 possible scenarios surrounding the mission-essential information. We let N be the set of all 5-letter words (numbering 158,390), minus the word corresponding to Bobd's catchphrase, which Bobd knows is not Sean Bean's. We calculate c from (4.63) to get $c \approx 1 + 8.072 \times 10^{-12}$. This implies that

$$\begin{aligned} \mathrm{B}_{007} &:= \widetilde{\mathrm{B}}_{\mathrm{SP}} \gtrapprox 0.1000056822136 > 0.1000056822128 \quad \text{or} \\ \mathrm{R}_{007} &:= \mathrm{A}_{\mathrm{SP}} \gtrapprox 6.31356975553 \times 10^{-6} > 6.31356975548 \times 10^{-6}. \end{aligned} \tag{4.65}$$

Of course, Ralph is ultimately thwarted, James Bobd saves the day, and Sean Bean dies. Good job Bobd!

To the best of our knowledge, the lower bounds presented for this application are the first of their kind. And this makes sense, because this application is brand spanking new!

Chapter 5

Conclusions

The years since the dawn of quantum cryptography have seen unconditional security disproved for a myriad of quantum applications. Though disheartening discoveries, some, if not many of these applications have the potential to improve on the security of classical protocols. To study the limits of quantum protocols, we require both upper and lower bounds. In our research, we introduced a method for the generation of constant lower bounds for a variety of one-sided two-party quantum applications: some new, some old. Many of these lower bounds break new ground for these applications, and it is our hope that these bounds pave the way for future research in their improvement and eventual optimization, as well as the discovery of quantum protocols that comply with them.

Chapter 6

Summary

In this thesis, we made use of the cryptographic primitives secure function evaluation and die-rolling to derive constant lower bounds for one-sided two-party quantum applications that generalize to secure function evaluation. This was accomplished through reducing die-rolling to a secure function evaluation protocol, using the principle of gentle measurements of quantum states, and applying Kitaev's bound for die-rolling to derive a lower bound for secure function evaluation. The generalization of quantum applications to secure function evaluation allow for the specifying of the lower bound, and consequently, the generation of constant lower bounds.

Bibliography

[1] S. Wiesner, "Conjugate coding," SIGACT News, 15(1), pp. 78-88, 1983.

[2] C. H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner, "Quantum Cryptography, or Unforgeable Subway Tokens," Advances in Cryptology: Proceedings of CRYPTO '82, pp. 267-275, 1982.

[3] C.H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing, (India), pp. 175–179, 1984.

[4] M. O. Rabin, "How to exchange secrets by oblivious transfer," Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[5] D. Mayers, "Unconditionally Secure Quantum Bit Commitment is Impossible," Physical Review Letters, 78(17):3414, 1997.

[6] H.-K. Lo and H.F. Chau, "Is quantum bit commitment really possible?" Physical Review Letters, 78(17):3410-3413, 1997.

[7] H.-K. Lo and H.F. Chau, "Why quantum bit commitment and ideal quantum coin tossing are impossible," Physica D: Nonlinear Phenomena, 120(1-2), pp. 177-187, 1998.

[8] H.-K. Lo, "Insecurity of quantum secure computations," Physical Review A, 56(2):1154, 1997.

[9] A. Chailloux and I. Kerenidis, "Optimal bounds for quantum bit commitment," IEEE 52nd Annual Symposium on Foundations of Computer Science, pp. 354-362, 2011.

[10] B. Baumgartner, "An inequality for the trace of matrix products, using absolute values," available as arXiv.org e-Print math-ph/1106.6189, 2011.

[11] A. Chailloux, I. Kerenidis, and J. Sikora, "Lower Bounds for Quantum Oblivious Transfer," Quantum Information & Computation, 13(1-2), pp. 158-177, 2013.

[12] A. Chailloux, G. Gutoski, and J. Sikora, "Optimal bounds for semi-honest quantum oblivious transfer," Chicago Journal of Theoretical Computer Science, 2016:13, 2016.

[13] G. Gutoski, A. Rosmanis, and J. Sikora, "Fidelity of quantum strategies with applications to cryptography," Quantum, 2:89, 2018.

[14] H. Buhrman, M. Christandl, and C. Schaffner, "Complete Insecurity of Quantum Protocols for Classical Two-Party Computation," Physical Review Letters, 109(16):160501, 2012.

[15] A. Kitaev, "Quantum coin-flipping," unpublished result, talk at the 6th Annual workshop on Quantum Information Processing (QIP 2003), 2002.

[16] A. Ambainis, "A new protocol and lower bounds for quantum coin flipping," in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, pp. 134-142, 2001.

[17] I. Kerenidis and A. Nayak, "Weak coin flipping with small bias," Information Processing Letters, 89:131-135, 2002.

[18] C. Mochon, "Quantum weak coin flipping with arbitrarily small bias," available as arXiv.org e-Print quant-ph/0711.4114, 2007.

[19] A. Chailloux and I. Kerenidis, "Optimal quantum strong coin flipping," in Proceedings of 50th IEEE Symposium on Foundations of Computer Science, pp. 527-533, IEEE Computer Society, 2009.

[20] J. Sikora, "Simple, near-optimal quantum protocols for die-rolling," Cryptography, 1(2):11, 2017.

[21] M. Wilde, "Quantum Information Theory (second edition)," Cambridge University Press, 2017.

[22] A. Winter, "Coding theorem and strong converse for quantum channels," IEEE Transactions on Information Theory, 45(7):2481-2485, 1999.

[23] S. Oskouei, S. Mancini, and M. Wilde, "Union bound for quantum information processing," in Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 475(2221):20180612, 2018.

[24] J. Gao, "Quantum union bounds for sequential projective measurements," Physical Review A, 92(5):052331, 2015.

[25] R. O'Donnell and R. Venkateswaran, "The Quantum Union Bound made easy," available as arXiv.org e-Print quant-ph/2103.07827, 2021.

[26] J. Sikora, A. Chailloux, and I. Kerenidis, "Strong connections between quantum encodings, nonlocality, and quantum cryptography," Physical Review A, 89(2):022334, 2014.

[27] D. Aharonov, A. Ta-Shma, U.V. Vazirani, and A.C. Yao, "Quantum bit escrow," in Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, STOC '00, (New York, NY, USA), pp. 705–714, Association for Computing Machinery, 2000.

[28] A. Nayak and P. Shor, "Bit-commitment-based quantum coin flipping," Physical Review A, 67(1):012304, 2003.

[29] I. Damgaard, S. Fehr, L. Salvail, and C. Schaffner, "Cryptography In the Bounded Quantum-Storage Model," in Proceedings of the 46th IEEE Symposium on Foundations of Computer Science, pp. 449-458, 2005.

[30] C. Schaffner, "Cryptography in the Bounded-Quantum-Storage Model," available as arXiv.org e-Print quant-ph/0709.0289, 2007.

[31] S. Wehner, C. Schaffner, and B. Terhal, "Cryptography from Noisy Storage," Physical Review Letters, 100(22):220502, 2008.

[32] C. Schaffner, B. Terhal, S. Wehner, "Robust cryptography in the noisy-quantum-storage model," Quantum Information & Computation, 9:963-996, 2009.

[33] S. Kundu, J. Sikora, and E. Tan, "A Device-Independent Protocol for XOR Oblivious Transfer," in Proceedings of the 15th Conference on the Theory of Quantum Computation, Communication, and Cryptography, 158(12):1-15, 2020.

[34] N. Aharon and J. Silman, "Quantum dice rolling: a multi-outcome generalization of quantum coin flipping," New Journal of Physics, 12(3):033027, 2010.

[35] D. Aharonov, A. Chailloux, M. Ganz, I. Kerenidis, and L. Magnin, "A Simpler Proof of the Existence of Quantum Weak Coin Flipping with Arbitrarily Small Bias," SIAM Journal on Computing, 45(3):633-679, 2016.

[36] M. Wilde, "Sequential decoding of a general classical-quantum channel," in Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 469(2157):20130259, 2013.

[37] A. Ambainis, A. Nayak, A. Ta-Shma, and U. V. Vazirani, "Dense quantum coding and quantum finite automata," Journal of the ACM, 49:496-511, 2002.

[38] S. Aaronson, "QMA/qpoly is contained in PSPACE/poly: de-Merlinizing quantum protocols," in Proceedings of the Conference on Computational Complexity, pp. 261-273, 2006.

[39] M. Blum, "Coin flipping by telephone," in Allen Gersho, editor, 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.

[40] R.W. Spekkens and T. Rudolph, "Degrees of concealment and bindingness in quantum bit commitment protocols," Physical Review A, 65:012310, 2001.