Revisiting the Efficiency of Malicious Two-Party Computation

David P. Woodruff *

MIT Computer Science and Artificial Intelligence Laboratory
dpwood@mit.edu

Abstract. In a recent paper Mohassel and Franklin study the efficiency of secure two-party computation in the presence of malicious behavior. Their aim is to make classical solutions to this problem, such as zero-knowledge compilation, more efficient. The authors provide several schemes which are the most efficient to date. We propose a modification to their main scheme using expanders. Our modification asymptotically improves at least one measure of efficiency of all known schemes. We also point out an error, and improve the analysis of one of their schemes.

Keywords: secure function evaluation, malicious model, efficiency, expander graphs

1 Introduction

Two parties, Alice with input x and Bob with input y, wish to evaluate a function $f(x, y)$ in such a way that neither learns more information than what can be deduced from the output $f(x, y)$. This problem, known as general two-party secure computation, generalizes many important cryptographic tasks. A celebrated result is *Yao's garbled circuit protocol* [34, 21], which provides a solution to this problem for any efficiently computable function f.

Yao's protocol provides security in the *semi-honest model*, that is, a model in which parties must follow the instructions of the protocol, though they may keep message histories in an attempt to learn more than what is prescribed. A more reaslistic security model is the *malicious model* in which parties may behave arbitrarily. The textbook solution to achieve security in the malicious model is to perform the zero-knowledge compilation of Goldreich *et al* [15, 16, 17] to Yao's protocol. This yields a protocol with communication and computation cost bounded by a polynomial in the size of a circuit for computing f. This results in optimal efficiency, up to polynomial factors, but the polynomial factors are rather large and so this approach may not be useful in practice.

This motivates alternative methods for protecting Yao's protocol against malicious behavior, as suggested in [24, 26, 28]. These techniques provide a well-defined tradeoff between security and efficiency, and are useful in practice.

* Supported by an N.D.S.E.G. fellowship.

These protocols all use the following *cut-and-choose* technique. Alice creates m independently garbled circuits $C_1, \ldots, C_m$, each computing the same function f. These garbled circuits are transmitted to Bob, along with various commitments. Bob chooses a subset $S \subset [m] = \{1, \ldots, m\}$, and asks Alice to reveal the secrets of all circuits C_i (along with their corresponding commitments) with $i \in S$. This gives Bob confidence that Alice correctly formed most of the garbled circuits and commitments. Alice then sends her garbled inputs for the circuts in $[m] \setminus S$, and Alice and Bob perform oblivious transfer for Bob to receive his garbled inputs for these circuits. Finally, Bob evaluates the garbled circuits and outputs the majority value.

There are a number of subtleties and complexities within this framework. As pointed out by Mohassel and Franklin in [26], the *Fairplay* scheme [24] designed in this framework has a subtle bug allowing one of the parties to cheat undetectably. Moreover, Kiraz and Schoenmakers [20] found an error that occurs in both Mohassel and Franklin's work and Fairplay. Recently, Lindell and Pinkas [22] have pointed out new flaws in many existing protocols. In this paper we also present an error in [26], showing a flaw in their estimated concrete costs.

This framework poses the following problems. How do we ensure Alice provides the same garbled input to most of the circuits in $[m] \setminus S$? How do we ensure Bob receives the same garbled input to most of the circuits in $[m] \setminus S$? If neither of these conditions hold, Alice can fool Bob into outputting an incorrect value or having to abort the protocol depending on his input.

Let f be a function computable by a Boolean circuit with g gates and I inputs. We want a protocol which achieves both privacy and correctness. Intuitively, the privacy aspect is that nothing is learned from the output, and the correctness aspect is that the output is distributed according to the described functionality, see [17]. Following previous work [26, 22], we will have two security parameters. The first is the input length, n, which is the security parameter for our commitment schemes, encryption, and oblivious transfer protocols. The second parameter, ϵ, is a statistical security parameter specifying the number of garbled circuits used in the cut-and-choose framework. Here, n depends on computational assumptions, whereas ϵ indicates the error probability incurred in this framework, and is therefore a "statistical" security parameter (this term was coined in [22]). Note that one can set ϵ independently of n, and this can be used to "trade" security for efficiency, as discussed below.

We note that when both parties are honest, it suffices to only have Bob output $f(x, y)$. Indeed, as shown in Section 2.2 of [22], this is without much loss of generality because given such a secure protocol, it can be efficiently transformed so that Alice also obtains $f(x, y)$, or even $f'(x, y)$ for some other function f'. We thus assume that only Bob has output in the remainder of the paper.

1.1 Our contributions

We study the efficiency of protocols in this framework. We measure three quantities: the number of symmetric encryptions, the number of exponentiations, and the communication complexity.

We are aware of four schemes in this framework - *Fairplay* [24], *Committed-input* [26], *Equality-checker* [26], and the very recent protocol of Lindell and Pinkas [22]. These schemes differ in the way the set S is chosen, together with their methods of enforcing Alice and Bob to have consistent inputs.

The main result of this paper is a new scheme, *Expander-checker*, which asymptotically improves at least one measure of efficiency of all known schemes. It results in fewer symmetric encryptions and smaller communication complexity than *Fairplay*, *Equality-checker*, and Lindell and Pinkas' protocol, while achieving fewer exponentiations[1] than *Committed-input*. See the section on other related work below for a more detailed account of Lindell and Pinkas' new protocol. Our results are summarized by the following table.

Scheme	**Symmetric Enc.**	**Exponentiations**	**Communication**
Fairplay [24]	$O(\frac{1}{\epsilon}g)$	$O(I)$	$O(\frac{1}{\epsilon}g)$
Committed-input [26]	$O(\ln(\frac{1}{\epsilon})g)$	$O(\ln(\frac{1}{\epsilon})I)$	$O(\ln(\frac{1}{\epsilon})g)$
Equality-checker [26]	$O(\ln(\frac{1}{\epsilon})g + \ln(\frac{1}{\epsilon})^2 I)$	$O(I)$	$O(\ln(\frac{1}{\epsilon})g + \ln(\frac{1}{\epsilon})^2 I)$
Lindell-Pinkas [22]	$O(\ln(\frac{1}{\epsilon})g + \ln(\frac{1}{\epsilon})^2 I)$	$O(I)$	$O(\ln(\frac{1}{\epsilon})g + \ln(\frac{1}{\epsilon})^2 I)$
Expander-checker (new)	$O(\ln(\frac{1}{\epsilon})g)$	$O(I)$	$O(\ln(\frac{1}{\epsilon})g)$

Our scheme is built off of *Equality-checker*. In that scheme, S is a random subset of size $m/2$. With a suitable commitment scheme, Mohassel and Franklin ensure that Bob's garbled inputs to the different circuits correspond to the same ungarbled input in each of the oblivious transfer steps. The more interesting part is how they ensure that Alice's garbled inputs to the different circuits correspond to the same ungarbled input. Their method only assumes a generic commitment scheme and can be implemented without any exponentiations.

Alice commits to tuples $(j, j', K^j_{i,b}, K^{j'}_{i,b})$ for all distinct $j, j' \in [m]$, where $K^j_{i,b}$ refers to the key in Yao's garbled circuit protocol associated with the ith input wire of Alice with value b in circuit j. When Bob is given purported keys $K^j_{i,b}$ and $K^{j'}_{i,b'}$, which correspond to Alice's garbled ith input for circuits j and j' respectively, Bob can use the witness to verify that $b = b'$.

If Alice creates enough commitments $(j, j', K^j_{i,b}, K^{j'}_{i,b'})$ with $b \neq b'$, then the set S likely contains a pair of circuits $C_j, C_{j'}$ with this property, and she will be caught when forced to reveal the circuits in S and the commitments between them. On the other hand, suppose most of the commitments $(j, j', K^j_{i,b}, K^{j'}_{i,b'})$ satisfy $b = b'$. Consider the complete graph G with vertex set $[m] \setminus S$, each vertex indexing a circuit not chosen by Bob to reveal. Since every pair of circuits C_j, C'_j with $j, j' \in [m] \setminus S$ has a commitment $(j, j', K^j_{i,b}, K^{j'}_{i,b'})$, there is a large connected component C for which for each edge $\{j, j'\} \in C$, for each i and each b, in the commitment $(j, j', K^j_{i,b}, K^{j'}_{i,b'})$, $b = b'$. By transitivity, Alice's input is the same

[1] Some care needs to be taken when measuring the number of exponentiations since under certain assumptions it is possible to obtain a large number of exponentiations by only performing a small number of exponentiations and a few simpler operations, see [5, 18]. In this work we follow previous work and make the simplifying, practical assumption that there are $O(I)$ exponentiations in the oblivious transfer stage.

to every circuit in C. If C is large enough, then the majority of circuits Bob evaluates (those in $[m] \setminus S$) have the same input from Alice, and the protocol will be simulatable.

The drawback of this scheme is the number of commitments computed and transmitted. This is $\Theta(mg + m^2 I)$, where I is the number of input wires owned by Alice. To achieve probability of undetected cheating at most ϵ, we need $m = \Omega(\ln(\frac{1}{\epsilon}))$, and thus $\Omega(g \ln \frac{1}{\epsilon} + I \ln^2 \frac{1}{\epsilon})$ commitments. Each commitment involves at least one symmetric encryption and one transfer from Alice to Bob, resulting in a total of $\Omega(g \ln(\frac{1}{\epsilon}) + I \ln^2(\frac{1}{\epsilon}))$.

Our idea is instead of computing commitments to all tuples $(j, j', K^j_{i,b}, K^{j'}_{i,b})$, we only commit to tuples for which $\{j, j'\}$ is an edge in an *expander graph*. Suppose G is an expander with vertex set $[m]$ and $O(m)$ edges. We commit only to pairs of circuits with a corresponding edge in G, and thus the number of symmetric encryptions and communication drop to $O(\ln(\frac{1}{\epsilon})g)$. For many circuits g is not much larger than I, and in this case we save a factor of $\ln(\frac{1}{\epsilon})$ in both efficiency measures.

Why is the new protocol secure? If Alice commits to enough $(j, j', K^j_{i,b}, K^{j'}_{i,b'})$ with $b \neq b'$, then as in *Equality-checker*, she is likely to get caught when Bob chooses a random subset of circuits and commitments to expose. On the other hand, if many of the $(j, j', K^j_{i,b}, K^{j'}_{i,b'})$ satisfy $b = b'$, then, since the corresponding graph G is an expander, it contains a large connected component of such edges. Thus, as before, the majority of circuits Bob evaluates will have the same Alice input, and the protocol will be simulatable.

Mohassel and Franklin [26] evaluated concrete costs for some practical settings of parameters. We point out an error in their analysis for Equality-checker, which is not obvious to us how to fix within their framework. We present a new graph-theoretic framework which fixes this and gives sharper bounds. We show the probability Alice can cheat is at most $2 \cdot 2^{-\frac{m}{4}}$, whereas it was previously thought this probability was at most $2 \cdot 2^{-\frac{m}{6}}$. Since the communication and number of symmetric encryptions of Equality-checker are proportional to $mg + m^2 I$, for a given security level we achieve at least a $(1/4)/(1/6) = 3/2$ factor efficiency improvement. This implies that Equality-checker is superior in practice to Fairplay for a wider range of parameters than Tables 2 and 4 of [26] suggest. To provide a good comparison with previous schemes, it is essential that we also lower bound the probability that Alice can cheat. We give a lower bound that is within a factor of 2 of our upper bound on this probability.

For Expander-checker we show the probability that Alice can cheat is at most $2^{-\frac{m}{4}+O\left(\frac{m \log d}{\sqrt{d}}\right)}$. With the present analysis, for a practical setting of parameters Equality-checker is still superior. We discuss barriers in derandomization and protocol design that need to be overcome in order to provably make Expander-checker superior in practice. We leave it as an open question to improve the analysis or provide an implementation to determine which protocol is more practical.

1.2 Other related work

Expanders have been used in other contexts for enforcing equality constraints. For example, see [19, 32, 10]. As far as we are aware though, this is the first time they have been used in the cut-and-choose framework.

Very recently, Lindell and Pinkas [22] have given the first rigorous proof of security of a protocol in the cut-and-choose framework that meets a simulation-based definition. Our protocol has the same security as Equality-checker [26] (with the fix pointed out by [20]), and it seems that a rigorous simulation-based security proof for Equality-checker has not appeared anywhere.

We stress that the focus of this work is a new way of proving input consistency using expander graphs, which we hope can be of use in other protocols in the cut-and-choose framework. It seems likely, though we have yet to formally verify, that one can combine our approach with Lindell and Pinkas' protocol to achieve the improved efficiency of Expander-checker while also achieving a full simulation-based proof of security. We sketch how this might be done in the proof of Theorem 3.

1.3 Organization

Section 2 reviews secure two-party computation, Yao's garbled circuit protocol, the *Equality-checker* scheme, and expander graphs. In Section 3 we present *Expander-checker*, and prove its security. In Section 4 we discuss efficiency, both in theory and in practice. This work also appears as a technical report in [33].

2 Preliminaries

2.1 Two-party secure computation

For an excellent treatment of secure two-party computation, the reader is referred to [17]. Here we summarize the model. A two-party computation is a random process mapping pairs of inputs to pairs of outputs. We refer to this process as the desired *functionality*, denoted $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ where $f = (f_1, f_2)$. For any two inputs $x, y \in \{0,1\}^n$, the output $(f_1(x,y), f_2(x,y))$ is a random variable ranging over pairs of strings. The interpretation here is that the first party wants to learn $f_1(x,y)$ and the second party wants to learn $f_2(x,y)$.

In this paper we consider the *malicious* model of security. The formal definitions can be found in [17]. In this model one of the parties can behave in an arbitrary way. We will, however, assume that both parties are computationally bounded (i.e., randomized polynomial-time Turing machines). Security is achieved by comparing the adversaries in the *real model* with those in an *ideal-model* in which both parties have a *trusted party* to interact with. Informally, a two-party protocol is secure if for any admissible pair of parties (A, B) in the real-model, there is an admissible pair of parties (A', B') in the ideal model where the outputs of the two executions are indistinguishable. A pair is admissible if at least one of the parties in the pair is honest. Thus, intuitively, the protocol is secure if it provides the correct output behavior, and provides privacy to honest parties.

In our protocols we need a specific protocol called *oblivious transfer*, which has been extensively studied [11, 27, 29]. We only need *1-out-of-2* oblivious transfer. In this case, $x = (z_0, z_1), y = \sigma, f_1(x, y) = \emptyset$, and $f_2(x, y) = z_\sigma$. Efficient oblivious transfer protocols secure in the malicious model exist [27].

2.2 Yao's garbled circuit protocol

Here we review Yao's garbled circuit protocol [34]. Let f be an efficiently-computable function two parties wish to securely compute. Then f can be represented as a polynomial-size circuit. The first party computes a garbled form of this circuit as follows.

For every wire j in the circuit, she chooses two random strings $K_{j,0}$ and $K_{j,1}$. These random strings correspond to a value of 0 and a value of 1 on wire j, respectively. Next, for every gate in the circuit, she computes a garbled truth table as follows. Let E be a symmetric encryption scheme. Then she uses E together with the keys corresponding to the values on the input wires to encrypt the value of the corresponding output wire. For example, if the gate is an AND gate on two input wires j, j' with output wire ℓ, then there are four entries

$$E_{K_{j,0}}(E_{K_{j',0}}(K_{\ell,0})), E_{K_{j,0}}(E_{K_{j',1}}(K_{\ell,0})), E_{K_{j,1}}(E_{K_{j',0}}(K_{\ell,0})), E_{K_{j,1}}(E_{K_{j',1}}(K_{\ell,1})).$$

These entries should be permuted so that the second party does not learn intermediate values of the computation. Also, she creates a table which translates the garbled output values to their actual values (0 or 1). She sends the garbled circuit and her garbled inputs to the second party.

The second party learns his garbled inputs through an oblivious transfer step. This ensures that only his garbled inputs are learned, and nothing else, while the first party learns nothing about the second party's inputs. The second party then computes the garbled circuit gate by gate, obtaining his garbled output. Finally, using the translation table, he obtains the actual output of the circuit. See [21] for detail, and a proof of security in the semi-honest model.

It is well-known that Yao's garbled protocol is not secure in the malicious model. The standard way of fixing this is to apply the zero-knowledge compiler of [15, 16]. The first party needs to supply a zero-knowledge proof that her circuit was constructed correctly and computes the desired functionality. The second party needs to supply zero-knowledge proofs that show he correctly evaluated the circuit. These zero-knowledge proofs, though theoretically feasible, are very inefficient and motivate the search for practical solutions.

2.3 Equality-checker scheme

The following is the *Equality-checker* scheme of [26]. Here, $z_{j,j',i,b}$ is Alice's commitment to the tuple $(j, j', i, K^j_{i,b}, K^{j'}_{i,b})$ for every distinct pair of circuits $j, j' \in [m]$, every input wire i of Alice, and every input value $b \in \{0, 1\}$. $w_{j,j',i,b}$

is the corresponding witness for decommittal. $z_{j,i,b}$ is the commitment of the tuple $(j, i, b, K_{i,b}^{j})$ for every circuit $j \in [m]$, every input wire i of Bob, and every input value $b \in \{0, 1\}$. $w_{j,i,b}$ is the corresponding witness for decommittal.

In the original paper [26], a generic oblivious transfer scheme was chosen in step 6, and this was shown to be insecure [20]. One fix is to use a committed oblivious transfer scheme (as stated below), or a *committing* scheme. See [20] for the details. Yet another approach will be discussed in the proof of Theorem 3.

We note that this does not affect our asymptotic analysis, and only marginally affects our concrete costs.

Equality-checker:

1. Alice creates m garbled circuits $C_1, \ldots, C_m$. She sends the C_j, $(j, j', i, z_{j,j',i,b})$, and $(j, i, b, z_{j,i,b})$ to Bob. The $(j, j', i, z_{j,j',i,b})$ should be sent in a random order so that Alice cannot distinguish $z_{j,j',i,0}$ from $z_{j,j',i,1}$.
2. Bob chooses a random subset $S \subset [m]$ with $|S| = m/2$ and sends S to Alice.
3. Alice exposes the secrets of the C_i for every $i \in S$. She also sends witnesses $w_{j,j',i,b}$ and $w_{j,i,b}$ for all i, b and all $j, j' \in S$. Bob verifies the garbled circuits and commitments are correct.
4. Renumber the remaining garbled circuits $C_1, \ldots, C_{m/2}$. Alice sends the keys K_{i,b_i}^{j} and the witnesses w_{j,j',i,b_i} for every distinct $j, j' \in [m/2]$ and each of her input wires i, where b_i is her input for wire i.
5. Bob uses the witnesses w_{j,j',i,b_i} to verify that Alice's input to all the circuits is the same.
6. Alice and Bob engage in committed oblivious transfers in order for Bob to receive his garbled input bits. For every input wire i of Bob, Alice uses a single oblivious transfer to give Bob one of two tuples: $(K_{i,0}^{1}, w_{1,i,0}, K_{i,0}^{2}, w_{2,i,0}, \ldots, K_{i,0}^{m/2}, w_{m/2,i,0})$ or $(K_{i,1}^{1}, w_{1,i,1}, K_{i,1}^{2}, w_{2,i,1}, \ldots, K_{i,1}^{m/2}, w_{m/2,i,1})$, depending on Bob's value for input i.
7. Bob evaluates the $m/2$ garbled circuits and prints the majority output.

We assume, as in [26], that computing the commitments $z_{j,j',i,b}$ and $z_{j,i,b}$ does not require exponentiation, but rather, just a symmetric encryption. We also assume a single oblivious transfer requires $O(1)$ exponentiations.

Theorem 1. *([26])* Equality-checker *is secure in the malicious model with inverse exponential (in m) probability of undetected cheating. The number of symmetric encryptions and the communication complexity are $O(mg + m^2 I)$, and the number of exponentiations is $O(I)$, where g and I are the number of gates and inputs of the circuit to be computed, respectively.*

2.4 Expander properties

Let $G = (V, E)$ be a d-regular graph on n vertices. Let $A = (a_{uv}),\ u, v \in V$, be its adjacency-matrix, that is, $a_{uv} = 1$ if $(u, v) \in E$ and $a_{uv} = 0$ otherwise. Since G is

d-regular, the largest eigenvalue of A is d, corresponding to the all 1s eigenvector. Let $\lambda = \lambda(G)$ denote the second largest *absolute value* of an eigenvalue of G. We need the following discrepancy theorem, known as the *expander-mixing lemma* (see, e.g, [1, 7], for the proof).

Theorem 2. *For any subsets $X, Y \subseteq V$,*

$$|e(X,Y) - \frac{d}{n}|X||Y|| \leq \frac{\lambda}{n}\sqrt{|X|(n-|X|)|Y|(n-|Y|)},$$

where $e(X,Y)$ is the number of edges with one endpoint in X and one endpoint in Y.

In our asymptotic analysis, we use explicit expander graphs known as *Ramanujan graphs*. The construction we use is essentially due to Lubotzky, Phillips, and Sarnak [23], and independently discovered by Margulis [25]. However, the form of these graphs [23, 25] is not so convenient to work with. We use a slight variant of these graphs described in section II of [2].

Fact 1 *[2] Let p, q be any distinct primes congruent to 1 modulo 4, with p a quadratic residue modulo q, and $q \geq 2\sqrt{p}$. Let $d = p+1$. Then for every positive integer ℓ, there is an explicit $(p+1)$-regular graph on $\frac{1}{2}(q^{3\ell} - q^{3\ell-2})$ vertices such that $\lambda \leq 2\sqrt{p}$.*

For fixed p, q as we vary ℓ we get an infinite family of graphs, and there is a positive constant α such that for any integer m, there is a graph in the family with m' vertices, where $m' \leq m \leq \alpha m'$. For a description of how to efficiently compute these graphs, see section II of [2].

We note that one can also obtain Ramanujan graphs by random sampling, and testing with Gaussian elimination. See [13] for how to sample such graphs.

2.5 Combinatorial identities

Fact 2 *(see [12, 30]) For integers $n > 0$, $\sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n} \leq n! \leq \sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n}e^{\frac{1}{12}}$.*

3 Expander-checker

Alice associates her m garbled circuits with the vertices of a d-regular Ramanujan graph $G = (V, E)$ on m vertices. The difference between our protocol and Equality-checker is that instead of committing to every pair of circuits $\{j, j'\}$, Alice only commits to the edges of G. Equality-checker is a special case of our protocol, which corresponds to setting $d = m - 1$. Since G has $dm/2$ edges, Alice performs $dm/2$ commitments.

We borrow some notation from Equality-checker, as described in Section 2.3. Let $z_{j,i,b}, w_{j,i,b}, z_{j,j',i,b}$, and $w_{j,j',i,b}$ be the commitments and witnesses as defined in that section. Alice only computes $z_{j,j',i,b}$ and $w_{j,j',i,b}$ for those $\{j, j'\}$ for which $\{j, j'\}$ is an edge of G.

For a subset S of the vertices V, let $G(S)$ denote the induced subgraph of G on vertex set S.

Expander-checker:

1. Alice creates m garbled circuits $C_1, \ldots, C_m$. For edges $\{j, j'\}$ in G, she sends the C_j, $(j, j', i, z_{j,j',i,b})$, and $(j, i, b, z_{j,i,b})$ to Bob. The $(j, j', i, z_{j,j',i,b})$ should be sent in a random order so that Alice cannot distinguish $z_{j,j',i,0}$ from $z_{j,j',i,1}$.
2. Bob chooses a (uniformly) random subset $S \subseteq [m]$ of size $m/2$. Bob sends S to Alice.
3. Alice exposes the secrets of the C_i for every $i \in S$. She also sends witnesses $w_{j,j',i,b}$ and $w_{j,i,b}$ for all i, b, all $j \in S$, and all $\{j, j'\} \in G(S)$. Bob verifies the garbled circuits and commitments are correct.
4. Renumber the remaining garbled circuits $C_1, \ldots, C_{m/2}$. Alice sends the keys K^j_{i,b_i} and the witnesses w_{j,j',i,b_i} for every $j \in V \setminus S$, every edge $\{j, j'\} \in G(V \setminus S)$, and each of her input wires i, where b_i is her input for wire i.
5. Bob uses the witnesses w_{j,j',i,b_i} to verify that Alice's input to all the circuits is the same.
6. Alice and Bob engage in committed oblivious transfers in order for Bob to receive his garbled input bits. For every input wire i of Bob, Alice uses a single oblivious transfer to give Bob one of the two tuples $(K^1_{i,0}, w_{1,i,0}, K^2_{i,0}, w_{2,i,0}, \ldots, K^{m/2}_{i,0}, w_{m/2,i,0})$ or $(K^1_{i,1}, w_{1,i,1}, K^2_{i,1}, w_{2,i,1}, \ldots, K^{m/2}_{i,1}, w_{m/2,i,1})$, depending on Bob's value for input i.
7. Bob evaluates the $m/2$ garbled circuits and prints the majority output.

If both parties are honest, the above protocol is correct, so we turn to security. We first develop a framework for proving the security of Equality-checker that is more powerful than that given in [26] (leading to better bounds, see Section 1), and which generalizes to Expander-checker.

3.1 Security analysis for Equality-checker

We will show that in order for a malicious Alice to cheat with non-negligible probability, the following must be true: *Alice does not provide the same input for more than $\frac{m}{4}$ of the correctly-garbled circuits that Bob will evaluate.* If this is not true then Bob will respond with the output corresponding to the majority input of Alice, in which case the protocol will be simulatable in the ideal model by sending the majority input to the trusted third party.

Let $\mathcal{F}$ be a family of complete graphs where each $G \in \mathcal{F}$ has some of its edges labeled bad, and some of its vertices labeled incorrect. We will use the observation above to construct a family $\mathcal{F}$ containing all of the (labeled) complete graphs G for which a malicious Alice can cheat with non-negligible probability.

If Alice can cheat by sending a graph G with exactly ϵm incorrect cicuits, then there must be some subset S of $\frac{m}{2}$ vertices of G which Bob can sample, so that if we remove S from G, Alice can assign her inputs to the remaining vertices so that no more than $\frac{m}{4}$ of the remaining vertices are assigned the same input. Partition the set of remaining vertices into groups $B, C_1, C_2, \ldots, C_r$, where B denotes the set of incorrect circuits (here, $|B| = \epsilon m$), and for each C_i, all vertices in C_i are assigned the same input. Then, all of the edges connecting C_i to C_j, for any $i \neq j$, must be bad edges, as otherwise Alice will get caught. Moreover, by the observation above, $|C_i| \leq \frac{m}{4}$ for all i. For a given G, there may be more than one choice of S, each giving rise to different sets $B, C_1, C_2, \ldots, C_r$ with the above properties. For our purposes, what matters is that there is at least one such $S, B, C_1, C_2, \ldots, C_r$ for the graph G. Let $\mathcal{F}$ be the family of all such graphs G.

Lemma 1. *If Alice chooses any graph $G \in \mathcal{F}$, she will get caught when Bob samples $\frac{m}{2}$ vertices of G with probability at least $1 - 2\binom{\frac{3m}{4}}{\frac{m}{2}}/\binom{m}{\frac{m}{2}}$.*

Proof. Fix any $G \in \mathcal{F}$, and let $S, B, C_1, C_2, \ldots, C_r$ be a partition of the vertices of G with the properties described above. We compute the probability that Alice does not get caught. Note that $|S| = \frac{m}{2}$ and $|B| = \epsilon m$. For all i, let $c_i = |C_i| \leq \frac{m}{4}$. As observed above, all of the edges between C_i and C_j for $i \neq j$ are bad, and therefore in order for Alice not to get caught, Bob can sample vertices from at most one C_i. Since B contains only incorrect circuits, Bob's samples must all be drawn from S and at most one C_i. Define an *elusive* set E to be a set of vertices of G not containing any incorrect vertices and such that no two endpoints of a bad edge lie in E. For Alice not to get caught, Bob must sample an elusive set. The number of elusive sets is at most $\sum_{j=0}^{\frac{m}{2}} \binom{\frac{m}{2}}{j} \sum_{i=1}^{r} \binom{c_i}{\frac{m}{2}-j}$.

We claim this expression is maximized when $r = 2$, $c_1 = \frac{m}{4}$, and $c_2 = \frac{m}{4} - \epsilon m$ (recall that $\sum_{i=1}^{r} c_i = \left(\frac{1}{2} - \epsilon\right) m$). First, if $r = 0$, the number of elusive sets is 1, namely, the set S. Second, if $r = 1$, then since $c_1 \leq \frac{m}{4}$, the expression evaluates to at most $\binom{\frac{3m}{4}}{\frac{m}{2}}$. This follows from the identity: $\sum_{j=0}^{\ell} \binom{n_1}{j}\binom{n_2}{\ell-j} = \binom{n_1+n_2}{\ell}$. For the remainder of the proof, assume $r \geq 2$.

We now use the identity for $a \geq b$: $\binom{a}{x} + \binom{b}{x} \leq \binom{a+1}{x} + \binom{b-1}{x}$. Since $c_i \leq \frac{m}{4}$ for all i, we may inductively apply the identity so that $r = 2$, $c_1 = \frac{m}{4}$, and $c_2 = \frac{m}{4} - \epsilon m$. It follows that the number of elusive sets is at most

$$\sum_{j=0}^{\frac{m}{2}} \binom{\frac{m}{2}}{j} \left(\binom{\frac{m}{4}}{\frac{m}{2} - j} + \binom{\frac{m}{4} - \epsilon m}{\frac{m}{2} - j} \right) = \binom{\frac{3m}{4}}{\frac{m}{2}} + \binom{\frac{3m}{4} - \epsilon m}{\frac{m}{2}} \leq 2\binom{\frac{3m}{4}}{\frac{m}{2}}.$$

It follows that the probability that Alice does not get caught is at most $2\binom{\frac{3m}{4}}{\frac{m}{2}}/\binom{m}{\frac{m}{2}}$.

Corollary 1. *With probability at least $1 - 2\binom{\frac{3m}{4}}{\frac{m}{2}}/\binom{m}{\frac{m}{2}}$, there are more than $\frac{m}{4}$ correctly-garbled circuits that Bob evaluates for which Alice will provide the same input, or Alice will get caught.*

Proof. If Alice does not use the same input for more than $\frac{m}{4}$ of the correctly-garbled circuits that Bob will evaluate, she will be caught unless she sends some graph $G \in \mathcal{F}$. But then, by the previous lemma, she will get caught with probability at least $1 - 2\binom{\frac{3m}{4}}{\frac{m}{2}}/\binom{m}{\frac{m}{2}}$, as needed.

Theorem 3. Equality-checker *is secure when Alice is malicious with probability of undetected cheating by Alice at most* $2\binom{\frac{3m}{4}}{\frac{m}{2}}/\binom{m}{\frac{m}{2}} \leq 2 \cdot 2^{-\frac{m}{4}}$.

Proof. By the previous corollary, with probability at least $1 - 2\binom{\frac{3m}{4}}{\frac{m}{2}}/\binom{m}{\frac{m}{2}} \geq 1 - 2 \cdot 2^{-\frac{m}{4}}$, the majority of inputs to the correctly-garbled circuits that Bob evaluates have the same input, or Alice will get caught, and thus Bob will output the value outputted by the circuits on this input.

The security, at this point, reduces to the original argument for Equality-checker given in Claim 3 of [26]. As the proof in [26] is incomplete, we refer the reader to [22]. To make the protocol simulatable, one needs to change step 2 of the protocol so that Alice and Bob run a standard coin-tossing protocol to generate the subset of circuits to evaluate. This ensures that if Bob is malicious, the circuits evaluated are still uniformly chosen (this sub-protocol is very efficient, and doesn't affect the overall efficiency). Also, instead of using committed oblivious transfer, another approach (analyzed in [22]) is for Bob to receive his inputs before Alice sends the garbled circuits. This amounts to removing step 6, and inserting it after step 1 in the protocol. Since the circuits to be evaluated have not yet been chosen, Bob should simply receive his inputs for every circuit.

Theorem 4. Equality-checker *is secure when Bob is malicious.*

Proof. The security reduces to the original argument for Equality-checker given in Claim 4 of [26]. For a formal proof, we refer the reader to [22].

Theorem 5. *In* Equality-checker*, Alice can cheat with probability at least* $\binom{\frac{3m}{4}}{\frac{m}{2}}/\binom{m}{\frac{m}{2}}$.

Proof. Alice will send the following labeled graph $G \in \mathcal{F}$ to Bob. She will not create any incorrect circuits. She will partition the vertices into two groups V_1, V_2, with $|V_1| = \frac{3m}{4}$ and $|V_2| = \frac{m}{4}$ (assume m is a multiple of 4). An edge is labeled bad if and only if it connects V_1 to V_2. Consider the following event $\mathcal{E}$: Bob samples all $\frac{m}{2}$ of his circuits from V_1. This occurs with probability $\binom{\frac{3m}{4}}{\frac{m}{2}}/\binom{m}{\frac{m}{2}}$.

Assume the circuit being evaluated has only one bit of input from Alice. Suppose $\mathcal{E}$ occurs. Alice may then assign all remaining vertices in V_1 the input 0 and all vertices in V_2 the input 1. If the function being evaluated differs on its output (for a given Bob input) when Alice's input is a 0 or a 1, then there is no majority output of Bob's evaluations (there are two outputs, and each one occurs for exactly half of the circuits). Thus, Bob will have to abort (and this behavior cannot be hidden from Alice), and this may reveal information to Alice about Bob's input. For instance, there may be another possible input of Bob which is insensitive to the input of Alice, in which case all circuits will have the same output, and Bob will not abort.

In Appendix 5, we present a counterexample to Lemma 3 in [26], from which their Table 4, which analyzes the performance of Equality-checker for different security levels, is constructed.

3.2 Security analysis for Expander-checker

We generalize the analysis of the previous section. The difficulty is that now the family $\mathcal{F}$ of graphs for which Alice can cheat with non-negligible probability is more complex. The graphs are no longer labeled complete graphs, but rather labeled expander graphs. We bound the new probability that Alice gets caught if she chooses a graph $G \in \mathcal{F}$ to send to Bob.

As before, for Alice to cheat, she cannot provide the same input for more than $\frac{m}{4}$ of the correctly-garbled cicuits that Bob will evaluate. Corollary 1, Theorem 3, and Theorem 4 are unchanged, except for the probability that Alice does not get caught, which will increase. We prove the new version of Lemma 1 in Theorem 6 below.

In Expander-checker, if Alice can cheat by sending a graph G, then as before, we can find a vertex partition $S, B, C_1, C_2, \ldots, C_r$ with $|S| = \frac{m}{2}$, $|B| = \epsilon m$ for some ϵ where B denotes the set of incorrect circuits, all edges in the expander connecting C_i to C_j for $i \neq j$ are bad, and $|C_i| \leq \frac{m}{4}$ for all i. Let $\mathcal{F}$ be the family of all such labeled graphs G.

We assume the expander graph satisfies $\lambda \leq 2\sqrt{d}$.

Theorem 6. *Let G be a d-regular Ramanujan graph for a sufficiently large constant d. If Alice chooses any graph $G \in \mathcal{F}$, she will get caught when Bob samples $\frac{m}{2}$ vertices of G with probability at least*

$$1 - 3\left(\frac{m}{4} + 1\right)\sqrt{\frac{\pi m e^{1/3}}{2}} \cdot 2^{-\frac{m}{4} + 2m\sqrt{\frac{2}{d}}\log\left(\frac{e}{4}\sqrt{\frac{d}{2}}\right)}.$$

Remark 1. Recall that our bound on the probability of undetected cheating by Alice for Equality-checker was $2 \cdot 2^{-\frac{m}{4}}$. Comparing this to our bound for Expander-checker, we see that when the degree $d = \omega(1)$, our new bound has the form $2^{-\frac{m}{4} + o(m)}$, close to that of Equality-checker.

Proof. Fix a graph $G \in \mathcal{F}$ with corresponding $S, B, C_1, C_2, \ldots, C_r$, where $|S| = \frac{m}{2}, |B| = \epsilon m$, and $c_i \stackrel{\text{def}}{=} |C_i| \leq \frac{m}{4}$ for all i. The difference between this proof and the previous is that now Bob can actually sample vertices from more than one C_i without Alice getting caught. This is because the graph G is not complete, so there may not be any edges connecting Bob's samples in the different C_i. However, using the expander-mixing lemma, we will show that if Bob samples too many vertices from different C_i, there will be bad edges connecting some of them, and Alice will get caught.

Define an elusive set as in the proof of Lemma 1. In order for Alice not to get caught, Bob must sample an elusive set, i.e., his vertices must come from $S \cup C_1 \cup C_2 \cup \cdots \cup C_r$ and there must be no edge between any of his samples lying in different C_i. We seek an upper bound on the number of elusive sets in G.

If $r = 0$, the number of elusive sets of G is 1. If $r = 1$, since $c_1 \leq \frac{m}{4}$, as in the proof of Lemma 1 for $r = 1$, the number of elusive sets is at most $\binom{\frac{3m}{4}}{\frac{m}{2}}$. For the remainder of the proof, $r \geq 2$.

We consider a labeled graph G' which has at least as many elusive sets as G. It will be easier to upper bound the number of elusive sets of G'. We want G' to have the property that its vertices can be partitioned into sets S, B, D_0, D_1, D_2 or sets S, B, D_0, D_1 such that $|S| = \frac{m}{2}$, $|B| = \epsilon m$, all edges between D_i and D_j with $i \neq j$ are bad, and $d_i = |D_i| \leq \frac{m}{4}$ for all i.

If $r = 2$ or $r = 3$, then put $G' = G$. Otherwise, $r \geq 4$. By averaging, there exist distinct C_i and C_j in G with $c_i + c_j \leq \frac{m}{4}$. Suppose we create G' from G by removing all bad edges between C_i and C_j, and by grouping vertices in C_i and C_j into a single set D of size $d = c_i + c_j \leq \frac{m}{4}$. It follows that r has decreased by 1. If r is still more than 3, repeat this process on G'. We eventually end up with the desired labeled graph G'. We will assume that $r = 3$. If actually $r = 2$, we may just set $D_2 = \emptyset$. We introduce some notation.

Definition 1. *We say that three integers i_0, i_1, i_2, where $i_0 \leq d_0$, $i_1 \leq d_1$, and $i_2 \leq d_2$, are* **harmonious** *if there exist sets $S_0 \subseteq D_0$, $S_1 \subseteq D_1$, and $S_2 \subseteq D_2$, where $|S_j| = i_j$ for $j = 0, 1, 2$, such that $e(S_0, S_1) = e(S_0, S_2) = e(S_1, S_2) = 0$. That is, there are no edges in G' between them.*

The number of elusive sets in G', and thus in G, is at most

$$\sum_{j=0}^{\frac{m}{2}} \binom{\frac{m}{2}}{\frac{m}{2} - j} \sum_{\substack{i_0+i_1+i_2=j \\ harmonious\ i_0,i_1,i_2}} \binom{d_0}{i_0}\binom{d_1}{i_1}\binom{d_2}{i_2} \leq$$

$$\sum_{j=0}^{\frac{m}{2}} \binom{\frac{m}{2}}{j} \sum_{r=0}^{2} \sum_{\substack{i_0+i_1+i_2=j \\ i_r=\max(i_0,i_1,i_2) \\ harmonious\ i_0,i_1,i_2}} \binom{d_0}{i_0}\binom{d_1}{i_1}\binom{d_2}{i_2}$$

We will choose d_0, d_1, d_2 to maximize this expression, subject to $\sum_i d_i = \frac{m}{4} - \epsilon m$ and $d_i \leq \frac{m}{4}$ for all i. As before, it is clear that the expression is maximized when $\epsilon = 0$. We start by bounding the following expression.

$$\sum_{j=0}^{\frac{m}{2}} \binom{\frac{m}{2}}{j} \sum_{\substack{i_0+i_1+i_2=j \\ i_0\ \geq\ i_1,i_2 \\ harmonious\ i_0,i_1,i_2}} \binom{d_0}{i_0}\binom{d_1}{i_1}\binom{d_2}{i_2}. \tag{1}$$

The following is the only place where we use the fact that G is an expander.

Claim. For fixed harmonious i_0, i_1, i_2 with $i_0 + i_1 + i_2 = j$ and $i_0 \geq i_1, i_2$, we have,

$$\binom{d_1}{i_1}\binom{d_2}{i_2} \leq \binom{\frac{m}{2}}{2m\sqrt{2/d}}.$$

Proof. Suppose first that $i_0 \leq m\sqrt{2/d}$. Then since $i_0 \geq i_1, i_2$, we have $i_1 + i_2 \leq 2i_0 \leq 2m\sqrt{2/d}$. We arrive at

$$\binom{d_1}{i_1}\binom{d_2}{i_2} \leq \binom{d_1 + d_2}{i_1 + i_2} \leq \binom{\frac{m}{2}}{2m\sqrt{2/d}},$$

where we have used that $i_1 + i_2 \leq 2m\sqrt{2/d} \leq \frac{m}{4}$ since d is sufficiently large. Now suppose that $i_0 \geq m\sqrt{2/d}$. This is where we use the fact that G is an expander. Suppose T is a subset of $D_0 \cup D_1 \cup D_2$, and set $X = T \cap D_0$ and $Y = T \cap (D_1 \cup D_2)$. Suppose $|X| = i_0$ and $|Y| = i_1 + i_2$. We first note that the edgeset in G' connecting X to Y is identical to that in G. By the expander-mixing lemma, there is at least one edge from X to Y provided[2] that

$$\frac{d}{m}|X||Y| > \frac{\lambda}{m}\sqrt{|X|(m-|X|)|Y|(m-|Y|)}.$$

This is equivalent to the condition $|X||Y| > \left(\frac{\lambda}{d}\right)^2(m-|X|)(m-|Y|)$. As we will choose λ so that $\lambda \leq 2\sqrt{d}$, this is in turn implied by the simpler $|X||Y| > \frac{4m^2}{d}$. This is just $i_0(i_1+i_2) > \frac{4m^2}{d}$. Since $i_0 \geq m\sqrt{2/d}$, this holds if $i_1+i_2 > 2m\sqrt{2/d}$. Thus, i_0, i_1, and i_2 are not harmonious if $i_1 + i_2 > 2m\sqrt{2/d}$, and so we again have $\binom{d_1}{i_1}\binom{d_2}{i_2} \leq \binom{d_1+d_2}{i_1+i_2} \leq \binom{\frac{m}{2}}{2m\sqrt{2/d}}$.

By the previous claim, expression 1 simplifies to

$$\sum_{j=0}^{\frac{m}{2}} \binom{\frac{m}{2}}{j} \sum_{\substack{i_0+i_1+i_2=j \\ i_0 \geq i_1, i_2 \\ harmonious\ i_0,i_1,i_2}} \binom{d_0}{i_0}\binom{\frac{m}{2}}{2m\sqrt{2/d}} =$$

$$\binom{\frac{m}{2}}{2m\sqrt{2/d}} \sum_{j=0}^{\frac{m}{2}} \binom{\frac{m}{2}}{j} \sum_{\substack{i_0+i_1+i_2=j \\ i_0 \geq i_1, i_2 \\ harmonious\ i_0,i_1,i_2}} \binom{d_0}{i_0}$$

In expression 1, we took $i_0 \geq i_1, i_2$, but we could've equally well taken $i_1 \geq i_0, i_2$ or $i_2 \geq i_0, i_1$. It follows that the number of elusive sets in G is at most

$$\binom{\frac{m}{2}}{2m\sqrt{2/d}} \sum_{j=0}^{\frac{m}{2}} \binom{\frac{m}{2}}{j} \sum_{r=0}^{2} \sum_{\substack{i_0+i_1+i_2=j \\ i_r \geq i_{r+1}, i_{r+2} \\ harmonious\ i_0,i_1,i_2}} \binom{d_r}{i_r}, \tag{2}$$

where the subscripts should be understood modulo 3. At this point, our task is to maximize expression 2 subject to $\sum_i d_i = \frac{m}{2}$ and $d_i \leq \frac{m}{4}$ for all i.

By switching the order of summations, we have shown that the number of elusive sets is at most

$$\binom{\frac{m}{2}}{2m\sqrt{2/d}} \sum_{r=0}^{2} \sum_{j=0}^{\frac{m}{2}} \sum_{\substack{i_0+i_1+i_2=j \\ i_r \geq i_{r+1}, i_{r+2} \\ harmonious\ i_0,i_1,i_2}} \binom{\frac{m}{2}}{j}\binom{d_r}{i_r}. \tag{3}$$

---

[2] One can do slightly better than the expander-mixing lemma by using Tanner's inequality [31]. This does not affect our bound much, so we omit this improvement.

Then, since there are at most $j+1$ pairs (i_{r+1}, i_{r+2}) for a given i_r for which $i_r + i_{r+1} + i_{r+2} = j$, we can bound the inner sum by $\binom{\frac{m}{2}}{j}(j+1)2^{d_r}$. We may then pull out the 2^{d_r} term and, ignoring the terms that we have pulled out, we are left with $\sum_{j=0}^{\frac{m}{2}} \binom{\frac{m}{2}}{j}(j+1)$. We recall the identity: $\sum_{i=0}^{n} i\binom{n}{i} = n2^{n-1}$. This implies

$$\sum_{j=0}^{\frac{m}{2}} (j+1)\binom{\frac{m}{2}}{j} = \frac{m}{2} \cdot 2^{\frac{m}{2}-1} + 2^{\frac{m}{2}} = \left(\frac{m}{4}+1\right) 2^{\frac{m}{2}}.$$

We can now simplify expression 3 to the following,

$$\binom{\frac{m}{2}}{2m\sqrt{2/d}} \left(\frac{m}{4}+1\right) 2^{\frac{m}{2}} \left(2^{d_0} + 2^{d_1} + 2^{d_2}\right).$$

This expression is clearly maximized when $d_r = d_{r+1} = \frac{m}{4}$ and $d_{r+2} = 0$ for some value of r. Since $2^{\frac{m}{4}} \geq 1$ for any $m \geq 0$, this expression is at most

$$3\binom{\frac{m}{2}}{2m\sqrt{2/d}} \left(\frac{m}{4}+1\right) 2^{\frac{3m}{4}}.$$

Using the identity $\binom{a}{b} \leq \left(\frac{ae}{b}\right)^b$, we further upper bound this expression as

$$3\left(\frac{m}{4}+1\right) 2^{\frac{3m}{4} + 2m\sqrt{\frac{2}{d}}\log\left(\frac{e}{4}\sqrt{\frac{d}{2}}\right)},$$

which upper bounds the total number of elusive sets. Thus, the probability that Alice does not get caught is at most this quantity divided by $\binom{m}{\frac{m}{2}}$. Using Fact 2, after some algebraic manipulation, $\binom{m}{m/2} = \frac{m!}{(m/2)!^2} \geq 2^m \left(\frac{2}{\pi m e^{1/3}}\right)^{1/2}$. We conclude that the probability that Alice does not get caught is at most

$$3\left(\frac{m}{4}+1\right)\sqrt{\frac{\pi m e^{1/3}}{2}} \cdot 2^{-\frac{m}{4} + 2m\sqrt{\frac{2}{d}}\log\left(\frac{e}{4}\sqrt{\frac{d}{2}}\right)}.$$

and the proof of the theorem is complete.

4 Efficiency

To compare Expander-checker with Equality-checker, we would like to achieve inverse exponential (in m) probability of undetected cheating, where m is an input parameter we use to measure our protocol's efficiency. m corresponds to the number of garbled circuits in the above.

The probability Alice can cheat in Expander-checker is at most

$$3\left(\frac{m}{4}+1\right)\sqrt{\frac{\pi m e^{1/3}}{2}} \cdot 2^{-\frac{m}{4} + 2m\sqrt{\frac{2}{d}}\log\left(\frac{e}{4}\sqrt{\frac{d}{2}}\right)}.$$

One can write a short computer program to find a constant $d = p+1$ with p a prime congruent to 1 mod 4, for which we can instantiate the graphs G in the previous section with those of Fact 1 on $\Theta(m)$ vertices, so that this probability is at most $2^{-\Omega(m)}$. One can also find such a graph by random sampling [13].

To achieve error probability ϵ, we may set $m = O(\ln \frac{1}{\epsilon})$. Recall that g and I denote the number of gates and inputs to the circuit to be computed, respectively.

Step 1 requires $O(mg) = O(\ln(\frac{1}{\epsilon})g)$ symmetric encryptions and communication for the garbled circuits. The commitments require $O(dmI + 2mI) = O(mI) = O(mg)$ symmetric encryptions and communication. Step 2 requires communication $O(m)$. Similar to step 1, step 3 requires $O(mg)$ communication. Step 4 requires $O(mI)$ communication. Step 6 requires $O(I)$ exponentiations.

Theorem 7. Expander-checker *is secure in the malicious model with inverse exponential (in m) probability of undetected cheating. The number of symmetric encryptions and communication complexity are $O(mg)$, and the number of exponentiations is $O(I)$.*

Recall that Equality-checker achieves $2^{-\Omega(m)}$ probability of undetected cheating with $O(mg + m^2 I)$ communication and number of symmetric encryptions, while the number of exponentiations is $O(I)$ (see Theorem 1). Suppose we want error probability ϵ. Let m be such that we achieve error probablity ϵ in Equality-checker. Then in Expander-checker we achieve error probability ϵ for $m' = O(m)$. Moreover, our communication and number of symmetric encryptions is $O(m'g) = O(mg)$, which improves the $\Omega(mg + m^2 I)$ of Equality-checker for sufficiently large m and I.

4.1 Practical issues and open questions

For a practical setting of parameters our bounds on the probability that Alice can cheat in Expander-checker may not be good enough to make Expander-checker favorable to Equality-checker. This is due in part to a certain suboptimality of our Ramanujan graphs. In Claim 3.2 we argued that any two disjoint sets of vertices in a Ramanujan graph on m vertices, one of size at least $m\sqrt{2/d}$ and one of size at least $2m\sqrt{2/d}$, have an edge between them. However, a counting argument shows there exist graphs on m vertices for which there is an edge between any two disjoint sets of vertices of size at least $2m \ln d/d$. Such an explicit graph would significantly reduce the $2^{2m\sqrt{2/d}\log(\frac{e}{4}\sqrt{\frac{d}{2}})}$ factor in our probability bound.

We cannot even rule out that there exist graphs on m vertices for which there is an edge between any two disjoint subsets of $\Theta(m/d)$ vertices. As far as we are aware, the best explicit construction of such graphs can be obtained from [6], and show there exist graphs on m vertices for which any two disjoint sets of vertices of size $\Omega(m \cdot \text{polylog}(d)/d)$ have an edge between them. We leave it as an open problem to see if the work of [6] can be of practical use in this context.

Besides directly trying to construct such graphs, it may be possible to slightly change the protocol. The natural thing to do would be to have Bob sample a d-regular graph on m vertices at random, and send it to Alice to use instead of our explicit Ramanujan graph. Then with high probability it is such that any two disjoint subsets of vertices of size $2m \ln d/d$ have an edge between them. The problem with this approach is that the probability of sampling such a graph is only $1 - 2^{-\Theta(m/d)}$, which is smaller than the $1 - 2^{-\Theta(m)}$ we are looking for. We leave it as an open problem to see if a probabilistic approach can be effective here.

Acknowledgments: We thank Payman Mohassel, Benny Pinkas, and the anonymous referees for many helpful comments.

References

[1] N. Alon, *Eigenvalues and expanders,* Combinatorica **6**, 1986, pp. 86–96.

[2] N. Alon, J. Bruck, J. Naor, M. Naor, and R. Roth, *Construction of asymptotically good, low-rate error-correcting codes through pseudo-random graphs*, IEEE Transactions on Information Theory **38** (192), pp. 509-516.

[3] N. Alon and V. D. Milman. *Eigenvalues, expanders, and superconcentrators,* FOCS, 1984.

[4] N. Alon and J. Spencer. *The Probabilistic Method*, 2000.

[5] D. Beaver. *Correlated pseudorandomness and the complexity of private computations*, STOC, 1996.

[6] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. *Randomness conductors and constant-degree lossless expanders*, STOC, 2002.

[7] F. Chung, *Spectral Graph Theory*, CBMS Lecture Notes, AMS Publications, 1997.

[8] F. Chung and L. Lu. *Concentration inequalities and martingale inequalities - a survey,* Internet Mathematics, to appear.

[9] R. Diestel. *Graph Theory*, Springer-Verlag, 2005.

[10] I. Dinur. *The PCP Theorem by Gap Amplification*, STOC, 2006.

[11] S. Even, O. Goldreich and A. Lempel. *A randomized protocol for signing contracts,* Communications of the ACM, 1985.

[12] W. Feller, *Stirling's Formula*, Section 2.9 in An Introduction to Probability Theory and its Applications **1**, 3rd edition, New York: Wiley, pp. 50 -53, 1968.

[13] J. Friedman, *A Proof of Alon's Second Eigenvalue Conjecture,* STOC, 2003.

[14] O. Gabber and Z. Galil. *Explicit constructions of linear-sized superconcentrators,* JCSS, **22(3)**:407-420, 1981.

[15] O. Goldreich, S. Micali, and C. Rackoff. *Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proofs*, FOCS, 1986.

[16] O. Goldreich, S. Micali, and A. Wigderson. *How to play any mental game or a completeness theorem for protocols with honest majority*, STOC, 1987.

[17] O. Goldreich. *Foundations of cryptography* - volume 2, ch. 7, 2004.

[18] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. *Extending oblivious transfers efficiently*, Crypto, 2003.

[19] J. Kilian and E. Petrank. *An efficient noniteractive zero-knowledge proof system for NP with general assumptions*, Journal of Cryptology, **11**:1–27, 1998.

[20] M. Kiraz and B. Schoenmakers, *A protocol issue for the malicious case of Yao's garbled circuit construction*, in the 27th Symposium on information theory in the BENELUX (WIC), 2006.

[21] Y. Lindell, and B. Pinkas. *A proof of Yao's protocol for secure two-party computation*, Cryptology ePrint Archive, Report 2004/175, 2004.

[22] Y. Lindell, and B. Pinkas. *An efficient protocol for secure two-party computation in the presence of malicious adversaries*, to appear in these proceedings, Eurocrypt, 2007.

[23] A. Lubotzky, R. Phillips, and P. Sarnak. *Explicit expanders and the Ramanujan conjectures*, STOC, 1986. See also: A. Lubotzky, R. Phillips, and P. Sarnak, Ramanujan graphs, Combinatorica 8, 1988, pp. 261-277.
[24] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. *Fairplay - a secure two-party computation system*, Usenix, 2004.
[25] G. A. Margulis. *Explicit group-theoretical constructions of combinatorial schemes and their application to the design of expanders and superconcentrators*, Problemy Peredachi Informatsii **24:** 51-60 (Russian). English translation in Problems of Information Transmission **24**, 1988, 39-46.
[26] P. Mohassel and M. Franklin. *Efficiency Tradeoffs for Malicious Two-Party Computation*, PKC, 2006.
[27] M. Naor and B. Pinkas. *Efficient oblivious transfer.* SODA, 2001.
[28] B. Pinkas. *Fair secure two-party computation*, Eurocrypt, 2003.
[29] M. Rabin. *How to exchange secrets by oblivious transfer*, Technical Report Tech., Memo. TR-81, Aiken Computation Laboratory, Harvard University, 1981.
[30] H. Robbins. *A remark of Stirling's Formula.*, Amer. Math Monthly **62**, pp. 26-29, 1955.
[31] R. M. Tanner. *Explicit Construction of Concentrators from Generalized N-Gons,* SIAM J. Alg. Discr. Math **5**, 1984, pp. 287-293.
[32] L. Trevisan. *Inapproximability of Combinatorial Optimization Problems*, Optimisation Combinatiore 2.
[33] D. Woodruff *Revisiting the efficiency of malicious two-party computation*, Cryptology ePrint Archive, Report 2006/397, 2006.
[34] A. C. Yao. *How to generate and exchange secrets*, FOCS, 1986.

5 Appendix: a counterexample

We've restated the lemma of [26] in our language (in this paper we have swapped the roles of Alice and Bob):

Lemma 3 of [26]: *With probability $\geq 1 - 2^{-\frac{m}{6}}$, at least $\frac{5}{6}$ of Alice's $\frac{m}{2}$ inputs are the same, or Alice will get caught.*

Consider the following behavior of a malicious Alice. Label the garbled circuits $C_1, ..., C_m$. Suppose m is a multiple of 8. For the first $\frac{7m}{8}$ circuits $C_1, ..., C_{7m/8}$, Alice will use the input 0 (assume Alice has only one input to the circuits), and for every other circuit, Alice will use the input 1. Thus, the bad edges are exactly those between one of the first $\frac{7m}{8}$ circuits and one of the last $\frac{m}{8}$ circuits.

Since all the circuits are correctly garbled, Alice only gets caught if a bad commitments is exposed in step 3. Consider the following event $\mathcal{E}$: Bob samples all $\frac{m}{2}$ of his circuits from the first $\frac{7m}{8}$ garbled circuits. Observe that if $\mathcal{E}$ occurs, no bad commitment is exposed in step 3, and therefore Alice does not get caught. Moreover, if $\mathcal{E}$ occurs, Bob will use $\frac{7m}{8} - \frac{m}{2} = \frac{3m}{8}$ 0 inputs when he performs verification, and $\frac{m}{8}$ 1 inputs. Thus, at most $\frac{3}{4}$ of Alice's $\frac{m}{2}$ inputs are the same, contrary to the $\frac{5}{6}$ claimed by Lemma 3.

For the counterexample to go through, it remains to show $\Pr[\mathcal{E}] > 2^{-\frac{m}{6}}$. But $\Pr[\mathcal{E}]$ is just $\binom{\frac{7m}{8}}{\frac{m}{2}} / \binom{m}{\frac{m}{2}}$. It is then straightforward to show $\binom{\frac{7m}{8}}{\frac{m}{2}} / \binom{m}{\frac{m}{2}} > 2^{-\frac{m}{6}}$, as needed.

The above presentation was done for simplicity. One can replace $\frac{7m}{8}$ by any value less than $\frac{11m}{12}$ in the above to get a "stronger" counterexample.