Electronic Colloquium on Computational Complexity, Report No. 92 (2015)

Compressing Communication in Distributed Protocols

Yael Tauman Kalai * Ilan Komargodski †

Abstract

We show how to compress communication in distributed protocols in which parties do not have private inputs. More specifically, we present a generic method for converting any protocol in which parties do not have private inputs, into another protocol where each message is *"short"* while preserving the same number of rounds, the same communication pattern, the same output distribution, and the same resilience to error. Assuming that the output lies in some universe of size M, in our resulting protocol each message consists of only $\mathsf{polylog}(M, n, d)$ many bits, where n is the number of parties and d is the number of rounds. Our transformation works in the full information model, in the presence of either static or adaptive Byzantine faults.

In particular, our result implies that for any such $\mathsf{poly}(n)$-round distributed protocol which generates outputs in a universe of size $\mathsf{poly}(n)$, long messages are not needed, and messages of length $\mathsf{polylog}(n)$ suffice. In other words, in this regime, any distributed task that can be solved in the $\mathcal{LOCAL}$ model, can also be solved in the $\mathcal{CONGEST}$ model with the *same* round complexity and security guarantees.

As a corollary, we conclude that for any $\mathsf{poly}(n)$-round collective coin-flipping protocol, leader election protocol, or selection protocols, messages of length $\mathsf{polylog}(n)$ suffice (in the presence of either static or adaptive Byzantine faults).

Keywords: Communication complexity, distributed computing, compression, coin-flipping.

*Microsoft Research. Email: yael@microsoft.com.

†Weizmann Institute of Science, Israel. Email: ilan.komargodski@weizmann.ac.il. Part of this work done while an intern at MSR New England. Supported in part by a grant from the I-CORE Program of the Planning and Budgeting Committee, the Israel Science Foundation, BSF and the Israeli Ministry of Science and Technology.

ISSN 1433-8092

1 Introduction

In classical algorithmic design the goal is to design efficient algorithms, where the common complexity measures are time and space. In distributed algorithms, where a set of parties tries to perform a predefined task, there are more parameters of interest, such as round complexity, message complexity, fault-tolerance, and more.

These measures have been studied in the literature under two main models: $\mathcal{LOCAL}$ and $\mathcal{CONGEST}$ [Pel00]. The $\mathcal{LOCAL}$ model is aimed at studying "localized" executions of distributed protocols, and thus, messages of unlimited size are allowed. The $\mathcal{CONGEST}$ model is geared towards understanding the effect of congestion in the network, and thus, messages of poly-logarithmic size (in the number of parties) are allowed.[1]

Most of the work in distributed computing assumes one of the models above and focuses on optimizing resources such as round complexity, message complexity and fault-tolerance. We initiate the study of the following question:

Is there a generic way to transform protocols in the $\mathcal{LOCAL}$ model to protocols in the $\mathcal{CONGEST}$ model, without negatively affecting the round complexity, fault-tolerance and other resources?

We give a positive answer to this question for protocols in which parties do not have private inputs, without incurring *any* cost to the round complexity or the resilience to errors. More details follow.

Our model. In this work, our focus is on the synchronous, full information model. Namely, we consider a distributed model in which n parties are trying to perform a predefined task. Each party is equipped with a source of private randomness and a unique ID. We assume the existence of a global counter which synchronizes parties in between rounds, but the parties are asynchronous within each round. The goal is to fulfill the task even in the presence of Byzantine faults. In the full information model no restrictions are made on the computational power of the faulty parties or the information available to them. Namely, the faulty parties may be infinitely powerful, and we do not assume the existence of private channels connecting pairs of honest parties.

We model faulty parties by a computationally unbounded adversary who controls a subset of parties and whose aim is to bias the output of the protocol. We assume that the adversary has access to the entire transcript of the protocol, and once a party is corrupted, the adversary gains complete control over the party and can send any messages on its behalf, and the messages can depend on the entire transcript so far. In addition, we allow our adversary to be "rushing", i.e., it can schedule the delivery of the messages within each round. We consider two classes of adversaries: *static* and *adaptive*. A static adversary is an adversary that chooses which parties to corrupt ahead of time, before the protocol begins. An adaptive adversary, on the other hand, is allowed to choose which parties to corrupt *adaptively* in the course of the protocol as a function of the messages seen so far.

The focus of this work, is on protocols in which parties do not have private inputs. Many classical distributed tasks fall in this category, including collective coin-flipping, leader election, selection and more.

A concrete motivation: adaptively-secure coin-flipping. An important distributed task that was extensively studied in the full information model, is that of *collective coin-flipping*. In this problem, a set of n parties use private randomness and are required to generate a common random

[1] We note that often the term $\mathcal{CONGEST}$ is a short-hand writing for $\mathcal{CONGEST}(B)$, where B is a bandwidth constraint. In many cases, the convention is to set B to be bounded by $O(\log n)$, where n is the number of parties. In this paper, we take the more liberal interpretation, which allows for messages of size bounded by $\mathsf{polylog}(n)$ (see e.g., [SMPU15]).

bit. The goal of the parties is to jointly output a somewhat uniform bit even in the case that some of the parties are faulty and controlled by a static (resp. adaptive) adversary whose goal is to bias the output of the protocol in some direction.

This problem was first formulated and studied by Ben-Or and Linial [BL85]. In the case of static adversaries, collective coin-flipping is well studied and almost matching upper and lower bounds are known [Fei99, RSZ02], whereas the case of adaptive adversaries has received much less attention. Ben-Or and Linial [BL85] showed that the majority protocol (in which each party sends a uniformly random bit and the output of the protocol is the majority of the bits sent) is resilient to $\Theta(\sqrt{n})$ adaptive corruptions. Furthermore, they conjectured that this protocol is optimal, that is, they conjectured that any coin-flipping protocol is resilient to at most $O(\sqrt{n})$ adaptive corruptions. Shortly afterwards, Lichtenstein, Linial and Saks [LLS89] proved the conjecture for protocols in which each party is allowed to send only *one* bit. Very recently, Goldwasser, Kalai and Park [GKP15] proved a different special-case of the aforementioned conjecture: any *symmetric* (many-bit) one-round collective coin-flipping protocol[2] is resilient to at most $\widetilde{O}(\sqrt{n})$ adaptive corruptions. Despite all this effort, proving a general lower bound, or constructing a collective coin-flipping protocol that is resilient to at least $\omega(\sqrt{n})$ adaptive corruptions, remains an intriguing open problem.

The result of [LLS89] suggests that when seeking for a collective coin-flipping protocol that is resilient to at least $\omega(\sqrt{n})$ adaptive corruptions, to focus on protocols that consist of many communication rounds, or protocols in which parties send long messages. Our main result (Theorem 1.1) is that long messages are not needed in adaptively secure coin-flipping protocols with $\mathsf{poly}(n)$ rounds, and messages of length $\mathsf{polylog}(n)$ suffice. This is true more generally for leader election protocols, and for selection protocols where the output comes from a universe of size at most quasi-polynomial in n.

1.1 Our Results

Our main result is that "long" messages are not needed for distributed tasks in which parties do not have private inputs. More specifically, we show how to convert any n-party d-round protocol, where parties do not have private inputs, and whose output comes from a universe of size M, into a d-round protocol, with the same communication pattern, the same output distribution, the same security guarantees, and where each message is of length $\mathsf{polylog}(M, n, d)$. Note that for many well studied distributed tasks, such as coin-flipping, leader election, and more, the output is from a universe of size at most $\mathsf{poly}(n)$, in which case our result says that if we consider $\mathsf{poly}(n)$-round protocols, then messages of length $\mathsf{polylog}(n)$ suffice.

Our results in more detail. Formally, we say that a protocol Π, in which parties do not have private inputs, is (t, δ, s)*-statically (resp., adaptively) secure* if for any adversary $\mathcal{A}$ that *statically (resp., adaptively)* corrupts at most $t = t(n)$ parties, and any subset S of the output universe such that $|S| = s$, it holds that

$$\left| \Pr\left[\text{Output of } \mathcal{A}(\Pi) \in S\right] - \Pr\left[\text{Output of } \Pi \in S\right] \right| \leq \delta,$$

where "Output of $\mathcal{A}(\Pi)$" means the output of the protocol when executed in the presence of the adversary $\mathcal{A}$, "Output of Π" means the output of the protocol when executed honestly, and the probabilities are taken over the internal randomness of the parties. In addition, we say that a protocol Π *simulates* a protocol Π' if the outcomes of the protocols are statistically close (when executed honestly) and their communication patterns are the same.

[2] A symmetric protocol Π is one that is oblivious to the order of its inputs: namely, for any permutation $\pi\colon [n] \to [n]$ of the parties, it holds that $\Pi(r_1, \ldots, r_n) = \Pi(r_{\pi(1)}, \ldots, r_{\pi(n)})$.

Our main result is a generic communication compression theorem which, roughly speaking, states that (t, δ, s)-statically (resp., adaptively) secure protocols in the above model *do not* need "long" messages. Namely, we show that any secure protocol which sends arbitrary long messages can be simulated by a protocol which is almost as secure and sends short messages.

Theorem 1.1 (Main theorem — informal)**.** *Any (t, δ, s)-statically (resp., adaptively) secure d-round protocol that outputs m bits (or more generally, has an output universe of size 2^m), can be simulated by a d-round (t, δ', s)-statically (resp., adaptively) secure protocol, where $\delta' = \delta + \mathsf{negl}(n)$, and in which parties send random messages of length at most $m \cdot \mathsf{polylog}(n, d)$.*

Our results can also be seen as a transformation of protocols (in which parties do not have private inputs) in the $\mathcal{LOCAL}$ model to protocols in the $\mathcal{CONGEST}$ model, as discussed above. Our main theorem (Theorem 1.1) implies that any task, whose output consists of at most $\mathsf{polylog}(n)$ bits, and in which parties do not have private inputs, that can be solved in the $\mathcal{LOCAL}$ model with $d \leq \mathsf{poly}(n)$ rounds, can also be solved in the $\mathcal{CONGEST}$ model with d rounds.

Corollary 1.2. *Any n-party (t, δ, s)-statically (resp., adaptively) secure $\mathsf{poly}(n)$-round protocol that outputs $\mathsf{polylog}(n)$ bits in the $\mathcal{LOCAL}$ model, can be simulated by a (t, δ', s)-statically (resp., adaptively) secure protocol in the $\mathcal{CONGEST}$ model, where $\delta' = \delta + \mathsf{negl}(n)$.*

We emphasize that our results holds for any underlying communication pattern including the broadcast channel or the message-passing model with any underlying communication graph.

Finally, we note that the transformation in Theorem 1.1 preserves the computational efficiency of the honest parties, but the resulting protocol is *non-uniform*, even if the protocol we started with is uniform. We elaborate on this in Section 1.3.

1.2 Related Work

The resource of communication is central in several fields of computer science. The field of communication complexity is devoted to the study of which problems can be solved with as little communication as possible. We refer to the book of Kushilevitz and Nisan [KN97] for an introduction to the field. In cryptography, minimizing communication has been the focus of several works in several contexts, including private information retrieval [KO97], random access memory machines [NN01], and more.

Interestingly, in the setting of distributed computing most of the work focuses on optimizing other resources such as round complexity, fault-tolerance, and the quality of the outcome. Very few works focus on optimizing the maximal message length being sent during the protocols. Moreover, most of the work in the literature focuses on *static* adversaries, and very few papers study distributed protocols with respect to *adaptive* adversaries. Our results hold in both settings.

Finally, we mention that separations between the $\mathcal{LOCAL}$ and $\mathcal{CONGEST}$ models are known for general tasks. For example, for network graphs of diameter $D = \Omega(\log n)$, computing the minimum spanning tree (MST) in the $\mathcal{LOCAL}$ model requires $\Theta(D)$ rounds, whereas in the $\mathcal{CONGEST}$ model every distributed MST algorithm has round complexity $\Omega(D + \sqrt{n}/\log^2 n)$ [PR00].

1.3 Overview of Our Techniques

In this section we provide a high-level overview of our main ideas and techniques. First, we observe that one can assume, without loss of generality, that any protocol in which parties do not have private inputs, can be transformed into a public-coin protocol, in which honest parties' messages consist only of random bits. This fact is a folklore, and for the sake of completeness we include a proof sketch of it in Section 4.

Our main result is a generic transformation that converts any public-coin protocol, in which parties send arbitrarily long messages, into a protocol in which parties send messages of length $m \cdot \mathsf{polylog}(n \cdot d)$, where m is the number of bits the protocol outputs, n is the number of parties participating in the protocol, and d is the number of communication rounds. The resulting protocol simulates the original protocol, has the same round complexity and satisfies the same security guarantees. Next, we elaborate on how this transformation works.

Suppose for simplicity that in our underlying protocol each message sent is of length $L = L(n)$ (and thus the messages come from a universe of size 2^L), and think of L as being very large. We convert any such protocol into a new protocol where each message consists of only ℓ bits, where think of ℓ as being significantly smaller than L. This is done by a priori choosing 2^ℓ messages within the 2^L-size universe, and restricting the parties to send messages from this restricted universe. Thus, now each message is of length ℓ, which is supposedly significantly smaller than L. We note that a similar approach was taken in [New91] in the context of transforming public randomness into private randomness in communication complexity, in [GS10] to reduce the number of random bits needed for property testers, and most recently in [GKP15] to prove a lower bound for coin-flipping protocols in the setting of strong adaptive adversaries.

A priori, it may seem that such an approach is doomed to fail, since by restricting the honest parties to send messages from a small universe within the large 2^L-size universe, we give the adversary a significant amount of information about future messages (especially in the multi-round case). Intuitively, the reason security is not compromised is that there are *many* possible restrictions, and it suffices to prove that a few (or only one) of these restrictions is secure. In other words, very loosely speaking, since we believe that most of the bits sent by honest parties are not "sensitive", we believe that it is safe to post some information about each message ahead of time.

For the sake of simplicity, in this overview we focus on static adversaries, and to simplify matters even further, we assume the adversary always corrupts the first t parties. This simplified setting already captures the high-level intuition behind our security proof in Section 3.

Let us first consider one-round protocols. Note that for one-round protocols restricting the message space of honest parties does not affect security at all since we consider rushing adversaries, who may choose which messages to send based on the content of the messages sent by all honest parties in that round. Thus, reducing the length of messages is trivial in this case, assuming the set of parties that the adversary corrupts is predetermined. We mention that even in this extremely simplified setting, we need ℓ to be linear in m for correctness ("simulation"), i.e., in order to ensure that the output is distributed correctly.

Next, consider a multi-round protocol Π. We denote by H the restricted message space, i.e., H is a subset of the message universe of size 2^ℓ, and denote by Π_H the protocol Π, where the messages are restricted to the set H. Suppose that for any set H there exists an adversary $\mathcal{A}^H$ that biases the outcome of Π_H, say towards 0.[3] We show that in this case there exists an adversary $\mathcal{A}$ in the underlying protocol that biases the outcome towards 0. Loosely speaking, at each step the adversary $\mathcal{A}$ will simulate one of the adversaries $\mathcal{A}^H$. More specifically, at any point in the underlying protocol, the adversary will randomly choose a set H such that the transcript so far is consistent with a run of protocol Π_H with the adversary $\mathcal{A}^H$, and will simulate the adversary $\mathcal{A}^H$. The main difficulty is to show that with high probability there exists such H (i.e., the remaining set of consistent H's is non-empty). This follows from a counting argument and basic probability

[3]Of course, it may be that for different sets H, the adversary $\mathcal{A}^H$ biases the outcome to a different value. For simplicity we assume here that all the adversaries bias the outcome towards a fixed message, which we denote by 0.

analysis.

In our actual construction, we have a distinct set H of size 2^ℓ corresponding to *each* message of the protocol. Thus, if the underlying protocol Π has d rounds, and all the parties send a message in each round, then the resulting (short-message) protocol is associated with $d \cdot n$ sets $H_1, \ldots, H_{d \cdot n}$ each of size 2^ℓ, where the message of the j^{th} party in the i^{th} round is restricted to be in the set $H_{i,j}$. We denote all these sets by a matrix $H \in \left(\{0,1\}^L\right)^{d \cdot n \times 2^\ell}$, where the row (i,j) of H corresponds to the set of messages that the j^{th} party can send during the i^{th} round.

Note that there are $2^{L \cdot 2^\ell \cdot d \cdot n}$ such matrices. Each time an honest party sends a uniformly random message in Π it reduces the set of consistent matrices by approximately a 2^L-factor (with high probability). Any time the adversary $\mathcal{A}$ sends a message, it also reduces the set of consistent matrices H, since his message is consistent only with some of the adversaries $\mathcal{A}^H$, but again a probabilistic argument can be used to claim that it does not reduce the set of matrices by too much, and hence, with high probability there always exist matrices H that are consistent with the transcript so far.

We briefly mention that the analysis in the case of adaptive corruptions follows the same outline presented above. One complication is that the mere decision of whether to corrupt or not reduces the set of consistent matrices H. Nevertheless, we argue that many consistent matrices remain.

We emphasize that the above is an over-simplification of our ideas, and the actual proof is more complex. We refer to Section 3 for more details.

2 Preliminaries

In this section we present the notation and basic definitions that are used in this work. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. Similarly, for a set X we denote by $x \leftarrow X$ the process of sampling a value x from the uniform distribution over X. Unless explicitly stated, we assume that the underlying probability distribution in our equations is the uniform distribution over the appropriate set. We let $\mathbf{U}_L$ denote the uniform distribution over $\{0,1\}^L$. We use $\log x$ to denote a logarithm in base 2.

A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is said to be *negligible* if for every constant $c > 0$ there exists an integer N_c such that $\mathsf{negl}(n) < n^{-c}$ for all $n > N_c$.

The *statistical distance* between two random variables X and Y over a finite domain Ω is defined as

$$\mathsf{SD}(X, Y) \triangleq \frac{1}{2} \sum_{\omega \in \Omega} \left| \Pr[X = \omega] - \Pr[Y = \omega] \right|. \quad (2.1)$$

The Model

The communication model and distributed tasks. We consider the *synchronous model* where a set of n parties $\mathsf{P}_1, \ldots, \mathsf{P}_n$ run protocols. Each protocol consists of *rounds* in which parties send messages. We assume the existence of a global counter which synchronizes parties in between rounds (but they are asynchronous within a round).

The focus of this work is on tasks where parties do not have any private inputs. Examples of such tasks are coin-flipping protocols, leader election protocols, Byzantine agreement protocols, etc.

Throughout this paper, we restrict ourselves to public-coin protocols.

Definition 2.1 (Public-coin protocols)**.** A protocol is *public-coin* if all honest parties' messages consist only of random bits.

Jumping ahead, we consider adversaries in the full infomration model. In Section 4 we argue that the restriction to public-coin protocols is without loss of generality since in the full information model any protocol (in which parties do not have private inputs) can be converted into a public-coin one, without increasing the round complexity and without degrading security (though this transformation may significantly increase the communication complexity).

The adversarial model. We consider the *full information model* where it is assumed the adversary is all powerful, and may see the entire transcript of the protocol. The most common adversarial model considered in the literature is the Byzantine model, where a bound $t = t(n) \leq n$ is specified, and the adversary is allowed to corrupt up to t parties. The adversary can see the entire transcript, has full control over all the corrupted parties, and can send any messages on their behalf. Moreover, the adversary has control over the order of the messages sent within each round of the protocol.[4] We focus on the Byzantine model throughout this work.

Within this model, two types of adversaries were considered in the literature: *static* adversaries, who need to specify the parties they corrupt *before* the protocol begins, and *adaptive* adversaries, who can corrupt the parties *adaptively* based on the transcript so far. Our results hold for both types of adversaries. Throughout this work, we focus on the adaptive setting, since the proof is more complicated in this setting. In Subsection 3.2 we mention how to modify (and simplify) the proof for the static setting.

Correctness and security. For any protocol Π and any adversary $\mathcal{A}$, we denote by

$$\mathsf{out}(\mathcal{A}_\Pi \mid \mathsf{r}_1, \ldots, \mathsf{r}_n)$$

the output of the protocol Π when executed with the adversary $\mathcal{A}$, and where each honest party P_i uses randomness r_i.

Let Π be a protocol whose output is a string in $\{0,1\}^m$ for some $m \in \mathbb{N}$. Loosely speaking, we say that an adversary is "successful" if he manages to bias the output of the protocol to his advantage. More specifically, we say that an adversary is "successful" if he chooses a predetermined subset $M \subseteq \{0,1\}^m$ of some size s, and succeeds in biasing the outcome towards the set M. To this end, for any set size s, we define

$$\begin{aligned}\mathsf{succ}_{\mathsf{s}}(\mathcal{A}_\Pi) &\stackrel{\text{def}}{=} \max_{M \subseteq \{0,1\}^m \text{ s.t. } |M|=s} \mathsf{succ}_M(\mathcal{A}_\Pi) \\ &\stackrel{\text{def}}{=} \max_{M \subseteq \{0,1\}^m \text{ s.t. } |M|=s} \left(\Pr_{\mathsf{r}_1,\ldots,\mathsf{r}_n}[\mathsf{out}(\mathcal{A}_\Pi \mid \mathsf{r}_1, \ldots, \mathsf{r}_n) \in M] - \Pr_{\mathsf{r}_1,\ldots,\mathsf{r}_n}[\mathsf{out}_\Pi(\mathsf{r}_1, \ldots, \mathsf{r}_n) \in M] \right),\end{aligned}$$

where $\mathsf{out}_\Pi(\mathsf{r}_1, \ldots, \mathsf{r}_n)$ denotes the outcome of the protocol Π if all the parties are honest, and use randomness $\mathsf{r}_1, \ldots, \mathsf{r}_n$.

Intuitively, the reason we parameterize over the set size s is that we may hope for different values of $\mathsf{succ}_M(\mathcal{A}_\Pi)$ for sets M of different sizes, since for a large set M it is often the case that $\Pr_{\mathsf{r}_1,\ldots,\mathsf{r}_n}[\mathsf{out}_\Pi(\mathsf{r}_1, \ldots, \mathsf{r}_n) \in M]$ is large, and hence $\mathsf{succ}_M(\mathcal{A}_\Pi)$ is inevitably small, whereas for small sets M the value $\mathsf{succ}_M(\mathcal{A}_\Pi)$ may be large.

For example, for coin-flipping protocols (where $m = 1$ and the outcome is a uniformly random bit in the case that all parties are honest), often an adversary is considered successful if it biases the outcome to his preferred bit with probability close to 1, and hence an adversary is considered successful if $\mathsf{succ}_M(\mathcal{A}_\Pi) \geq \frac{1}{2} - o(1)$ for either $M = \{0\}$ or $M = \{1\}$, whereas for general selection protocols (where m is a parameter) one often considers subsets $M \subseteq \{0,1\}^m$ of size $\gamma \cdot 2^m$ for some constant $\gamma > 0$, and an adversary is considered successful if there exists a constant $\delta > 0$ such that $\mathsf{succ}_M(\mathcal{A}_\Pi) \geq \delta$.

[4] Such an adversary is often referred to as "rushing."

Definition 2.2 (Security)**.** Fix any constant $\delta > 0$, any $t = t(n) \leq n$, and any n-party protocol Π whose output is an element in $\{0,1\}^m$. Fix any $s = s(m)$. We say that Π is (t, δ, s)-*adaptively secure* if for any adversary $\mathcal{A}$ that adaptively corrupts up to $t = t(n)$ parties, it holds that

$$\mathsf{succ_s}(\mathcal{A}_\Pi) \leq \delta.$$

We note that this definition generalizes the standard security definition for coin-flipping protocols and selection protocols. We emphasize that our results are quite robust to the specific security definition that we consider, and we could have used alternative definitions as well. Intuitively, the reason is that we show how to transform any d-round protocol Π into another d-round protocol with short messages, that simulates Π (see Definition 2.3 below), where this transformation is *independent* of the security definition. Then, in order to prove that the resulting protocol is as secure as the original protocol Π, we show that if there exists an adversary for the short protocol that manages to break security according to some definition, then there exists an adversary for Π that "simulates" the adversary of the short protocol and breaches security in the same way. (See Section 1.3 for more details, and Section 3 for the formal argument).

Finally, we mention that an analogous definition to Definition 2.2 can be given for static adversaries. Our results hold for the static definition as well.

Definition 2.3 (Simulation)**.** Let Π be an n-party protocol with outputs in $\{0,1\}^m$. We say that an n-party protocol Π' simulates Π if

$$\mathsf{SD}\left(\mathsf{out}_\Pi, \mathsf{out}_{\Pi'}\right) = \mathsf{negl}(n),$$

where out_Π is a random variable that corresponds to the output of protocol Π assuming all parties are honest, and $\mathsf{out}_{\Pi'}$ is a random variable that corresponds to the output of protocol Π' assuming all parties are honest.

Probabilistic Tools

In the analysis we will use the following simple claims.

Claim 2.4. *For any $k, M \in \mathbb{N}$, any $U \subseteq \{0,1\}^k$, any $f : U \to [M]$, and any $\varepsilon > 0$, the following holds. For every $i \in [M]$, denote by*

$$\alpha_i = \Pr_{u \leftarrow U}\left[f(u) = i\right].$$

Then,

$$\mathop{\mathbb{E}}_{u \leftarrow U}\left[\alpha_{f(u)}\right] \geq \frac{1}{M},$$

and

$$\Pr_{u \leftarrow U}\left[\alpha_{f(u)} \geq \frac{\varepsilon}{M}\right] \geq 1 - \varepsilon.$$

Proof. We begin with the proof of the first part. By the definition of expectation

$$\mathop{\mathbb{E}}_{u \leftarrow U}\left[\alpha_{f(u)}\right] = \sum_{u \in U} \Pr[U = u] \cdot \alpha_{f(u)} = \sum_{i=1}^{M} \alpha_i \cdot \Pr_{u \leftarrow U}\left[\alpha_{f(u)} = \alpha_i\right] \geq \sum_{i=1}^{M} \alpha_i^2.$$

This, together with the the Cauchy-Schwarz inequality, implies that

$$\begin{aligned}\mathop{\mathbb{E}}_{u \leftarrow U}\left[\alpha_{f(u)}\right] \geq \sum_{i=1}^{M} \alpha_i^2 &= \sum_{i=1}^{M} \alpha_i^2 \cdot \sum_{i=1}^{M} \left(\frac{1}{\sqrt{M}}\right)^2 \\ &\geq \left(\sum_{i=1}^{M} \alpha_i \cdot \frac{1}{\sqrt{M}}\right)^2 = \frac{1}{M},\end{aligned}$$

where the last equality follows from the fact that $\sum_{i=1}^{M} \alpha_i = 1$.

For the second part, let

$$B = \left\{ i \in [M] \mid \alpha_i \leq \frac{\varepsilon}{M} \right\}.$$

Then,

$$\Pr[i \in B] \leq \sum_{i \in B} \alpha_i \leq |B| \cdot \frac{\varepsilon}{M} \leq \varepsilon,$$

as desired, where the first equation follows from the union bound, the second equation follows from the definition of B, and the third equation follows from the fact that $|B| \leq M$. ■

Definition 2.5 (Entropy)**.** Let X be a random variable with finite support. The (Shannon) entropy of X is defined as

$$\mathsf{entropy}(X) = \sum_{x \in \mathsf{supp}(X)} \Pr[X = x] \cdot \log \frac{1}{\Pr[X = x]} = \mathop{\mathbb{E}}_{x \leftarrow X}\left[\log \frac{1}{\Pr[X = x]}\right]$$

Claim 2.6. *Let X be a random variable with domain $\{0,1\}^k$. If $\mathsf{entropy}(X) \geq k - \varepsilon$, then*

$$\mathsf{SD}(X, \mathbf{U}_k) \leq \sqrt{\frac{\varepsilon}{2}},$$

where $\mathbf{U}_k$ is the uniform distribution over k bits, and where $\mathsf{SD}(X, \mathbf{U}_k)$ denotes the statistical distance between X and $\mathbf{U}_k$ (see Equation (2.1) *for the definition of statistical distance).*

Proof. The *relative entropy* (a.k.a. the Kullback-Leibler divergence) between two distributions $\mathcal{D}_1, \mathcal{D}_2 \subseteq \{0,1\}^k$ is defined as

$$\mathbf{D}_{\mathsf{KL}}(\mathcal{D}_1 \| \mathcal{D}_2) = \sum_{x \in \{0,1\}^k} \mathcal{D}_1(x) \cdot \log\left(\frac{\mathcal{D}_1(x)}{\mathcal{D}_2(x)}\right).$$

A well known relation between relative entropy and the statistical distance is known as Pinsker's inequality which states that for any two distributions $\mathcal{D}_1, \mathcal{D}_2$ as above, it holds that

$$\mathsf{SD}(\mathcal{D}_1, \mathcal{D}_2) \leq \sqrt{\frac{\ln 2}{2} \cdot \mathbf{D}_{\mathsf{KL}}(\mathcal{D}_1 \| \mathcal{D}_2)}. \tag{2.2}$$

Thus, it remains to bound the relative entropy of X and $\mathbf{U}_k$. Let $p_x = \Pr_{x \in \{0,1\}^k}[X = x]$. We get that

$$\begin{aligned}\mathbf{D}_{\mathsf{KL}}(X \| \mathbf{U}_k) &= \sum_{x \in \{0,1\}^k} p_x \cdot \log\left(p_x \cdot 2^k\right) \\ &= \sum_{x \in \{0,1\}^k} p_x \cdot (\log(p_x) + k) \\ &= -\mathsf{entropy}(X) + k.\end{aligned}$$

Since $\mathsf{entropy}(X) \geq k - \varepsilon$, we get that

$$\mathbf{D}_{\mathsf{KL}}(X \| \mathbf{U}_k) \leq -k + \varepsilon + k = \varepsilon.$$

Plugging this into Pinsker's inequality (see Equation (2.2)), we get that

$$\mathsf{SD}(X, \mathbf{U}_k) \leq \sqrt{\frac{\ln 2}{2} \cdot \varepsilon} \leq \sqrt{\frac{\varepsilon}{2}}.$$

■

3 Compressing Communication in Distributed Protocols

In this section we show how to transform any n-party d-round t-adaptively secure public-coin protocol, that outputs messages of length m and sends messages of length L, into an n-party d-round t-adaptively secure public-coin protocol in which every party sends messages of length $\ell = m \cdot \mathsf{polylog}(n, d)$.

Theorem 3.1. *Fix any $m = m(n)$, $d = d(n)$, $L = L(n)$, and any n-party d-round public-coin protocol Π that outputs messages in $\{0,1\}^m$ and in which all parties send messages of length $L = L(n)$. Then, for any constant $\delta > 0$, any $t = t(n) < n$, and any $s = s(m)$, if Π is (t, δ, s)-adaptively secure then for every constant $c \in \mathbb{N}$ there exists a constant $b \in \mathbb{N}$ and an n-party d-round (t, δ', s)-adaptively secure public-coin protocol, that simulates Π, where all parties send messages of length $\ell = m \cdot \log^4(n \cdot d)$, and where $\delta' \leq \delta + \frac{b}{n^c}$.*

Proof. Fix any $m = m(n)$, $d = d(n)$, $L = L(n)$, and any n-party d-round public-coin protocol Π that outputs messages in $\{0,1\}^m$ and in which all parties send messages of length $L = L(n)$. Fix any constant $\delta > 0$, any $t = t(n) < n$, and any $s = s(m)$ such that Π is (t, δ, s)-adaptively secure. We start by describing the construction of the (short message) protocol. Let

$$N = 2^\ell = 2^{m \cdot \log^4(n \cdot d)}. \tag{3.1}$$

Let

$$\mathcal{H} = \{H : [d \cdot n] \times \{0,1\}^\ell \to \{0,1\}^L\}$$

be the set all possible $[d \cdot n] \times \{0,1\}^\ell \equiv [d \cdot n] \times [N]$ matrices, whose elements are from $\{0,1\}^L$. Note that $|\mathcal{H}| = 2^{d \cdot n \cdot N \cdot L}$. We often interpret $H : [d \cdot n] \times \{0,1\}^\ell \to \{0,1\}^L$ as a function

$$H : [d] \times [n] \times \{0,1\}^\ell \to \{0,1\}^L,$$

or as a matrix where each row is described by a pair from $[d] \times [n]$. We abuse notation and denote by

$$H(i, j, \mathsf{r}) \triangleq H((i-1)n + j, \mathsf{r}).$$

As a convention, we denote by R a message from $\{0,1\}^L$ and by r and a message from $\{0,1\}^\ell$.

From now on, we assume for the sake of simplicity of notation, that in protocol Π, in each round, all the parties send a message. Recall that we also assume for the sake of simplicity (and without loss of generality) that Π is a public-coin protocol (see Definition 2.1). For any $H \in \mathcal{H}$ we define a protocol Π_H that simulates the execution of the protocol Π, as follows.

The Protocol Π_H. In the protocol Π_H, for every $i \in [d]$ and $j \in [n]$, in the i^{th} round, party P_j sends a random string $\mathsf{r}_{i,j} \leftarrow \{0,1\}^\ell$. We denote the resulting transcript in round i by

$$\mathsf{Trans}_{H,i} = (\mathsf{r}_{i,1}, \ldots, \mathsf{r}_{i,n}) \in \left(\{0,1\}^\ell\right)^n,$$

and denote the entire transcript by

$$\mathsf{Trans}_H = (\mathsf{Trans}_{H,1} \ldots, \mathsf{Trans}_{H,d}).$$

We abuse notation, and define for every round $i \in [d]$,

$$H(\mathsf{Trans}_{H,i}) = (H(i, 1, \mathsf{r}_{i,1}), \ldots, H(i, n, \mathsf{r}_{i,n})).$$

Similarly, we define

$$H(\mathsf{Trans}_H) = (H(\mathsf{Trans}_{H,1}) \ldots, H(\mathsf{Trans}_{H,d})).$$

The outcome of protocol Π_H with transcript Trans_H is defined to be the outcome of protocol Π with transcript $H(\mathsf{Trans}_H)$.

It is easy to see that the round complexity of Π_H (for every $H \in \mathcal{H}$) is the same as that of Π. Moreover, we note that with some complication in notation we could have also preserved the exact communication pattern (instead of assuming that in each round all parties send a message).

In order to prove Theorem 1.1 it suffices to prove the following two lemmas.

Lemma 3.2. *For every constant $c \in \mathbb{N}$ there exists a constant $b \in \mathbb{N}$ and a subset $\mathcal{H}_0 \subseteq \mathcal{H}$ of size $\frac{|\mathcal{H}|}{2}$, such that for every matrix $H \in \mathcal{H}_0$ it holds that Π_H is (t, δ', s)-adaptively secure for $\delta' = \delta + \frac{b}{n^c}$.*

Lemma 3.3. *There exists a negligible function $\mu = \mu(n)$ such that,*

$$\Pr_{H \leftarrow \mathcal{H}}[\mathsf{SD}(\mathsf{out}_{\Pi_H}, \mathsf{out}_\Pi) \le \mu] \ge \frac{2}{3}.$$

Indeed, given Lemmas 3.2 and 3.3, we obtain that there exists an $H \in \mathcal{H}$ such that Π_H is (t, δ', s)-adaptively secure and it simulates Π. We proceed with the proof of Lemma 3.3 and then (in Section 3.1) give the proof of Lemma 3.2.

Proof of Lemma 3.3. By the definition of statistical distance, in order to prove Lemma 3.3 it suffices to prove that there exists a negligible function $\mu = \mu(n)$ such that,

$$\Pr_{H \leftarrow \mathcal{H}}\left[\forall z \in \{0,1\}^m, |\Pr[\mathsf{out}_{\Pi_H} = z] - \Pr[\mathsf{out}_\Pi = z]| \le \frac{\mu}{2^m}\right] \ge \frac{2}{3}.$$

Note that

$$\begin{aligned}
&\Pr_{H \leftarrow \mathcal{H}}\left[\forall z \in \{0,1\}^m, |\Pr[\mathsf{out}_{\Pi_H} = z] - \Pr[\mathsf{out}_\Pi = z]| \le \frac{\mu}{2^m}\right] = \\
&1 - \Pr_{H \leftarrow \mathcal{H}}\left[\exists z \in \{0,1\}^m, |\Pr[\mathsf{out}_{\Pi_H} = z] - \Pr[\mathsf{out}_\Pi = z]| > \frac{\mu}{2^m}\right] \ge \\
&1 - \sum_{z \in \{0,1\}^m} \Pr_{H \leftarrow \mathcal{H}}\left[|\Pr[\mathsf{out}_{\Pi_H} = z] - \Pr[\mathsf{out}_\Pi = z]| > \frac{\mu}{2^m}\right].
\end{aligned}$$

Therefore it suffices to prove that there exists a negligible function μ such that for every $z \in \{0,1\}^m$,

$$\Pr_{H \leftarrow \mathcal{H}}\left[|\Pr[\mathsf{out}_{\Pi_H} = z] - \Pr[\mathsf{out}_\Pi = z]| > \frac{\mu}{2^m}\right] \le \frac{1}{3 \cdot 2^m}.$$

To this end, for any $z \in \{0,1\}^m$, we denote by $p_z = \Pr[\mathsf{out}_\Pi = z]$ and $p_{z,H} = \Pr[\mathsf{out}_{\Pi_H} = z]$. Using this notation, it suffices to prove that there exists a negligible function μ such that for every $z \in \{0,1\}^m$,

$$\Pr_{H \leftarrow \mathcal{H}}\left[|p_{z,H} - p_z| > \frac{\mu}{2^m}\right] \leq \frac{1}{3 \cdot 2^m}.$$

For any $H \in \mathcal{H}$, consider the experiment, where we run the protocol Π_H independently $B = 2^{m \cdot \log^3(nd)}$ times, and check how many times the output is z. Denote by $X_1, \ldots, X_B$ the identically distributed random variables, where $X_i = 1$ if in the i^{th} run of the protocol the outcome is z, and $X_i = 0$ otherwise. The Chernoff bound[5] implies that for every $H \in \mathcal{H}$ and for every $\gamma > 0$,

$$\Pr\left[\left|\frac{1}{B}\sum_{i=1}^{B} X_i - p_{z,H}\right| \geq \gamma\right] \leq e^{-\frac{\gamma^2 \cdot B}{3}}.$$

In particular, setting $\gamma = 2^{-m \cdot \log^2(nd)}$ we deduce that

$$\Pr\left[\left|\frac{1}{B}\sum_{i=1}^{B} X_i - p_{z,H}\right| \geq \gamma\right] \leq e^{-2^{m \cdot \log^2(nd)}}. \tag{3.2}$$

We next define random variables $Y_1, \ldots, Y_B$ as follows: We run the protocol Π independently B times, and we set $Y_i = 1$ if in the i^{th} run the outcome is z, and otherwise we set $Y_i = 0$. We note that the same argument used to deduce Equation (3.2) can be used to deduce that

$$\Pr\left[\left|\frac{1}{B}\sum_{i=1}^{B} Y_i - p_z\right| \geq \gamma\right] \leq e^{-2^{m \cdot \log^2(nd)}}. \tag{3.3}$$

Note that,

$$\begin{aligned}
&\Pr\left[|p_{z,H} - p_z| > 4\gamma\right] \leq \\
&\Pr\left[\left|p_{z,H} - \frac{1}{B}\sum_{i=1}^{B} X_i\right| + \left|\frac{1}{B}\sum_{i=1}^{B} X_i - \frac{1}{B}\sum_{i=1}^{B} Y_i\right| + \left|\frac{1}{B}\sum_{i=1}^{B} Y_i - p_z\right| > 4\gamma\right] \leq \\
&\Pr\left[\left|p_{z,H} - \frac{1}{B}\sum_{i=1}^{B} X_i\right| > \gamma\right] + \Pr\left[\left|\frac{1}{B}\sum_{i=1}^{B} X_i - \frac{1}{B}\sum_{i=1}^{B} Y_i\right| > 2\gamma\right] + \Pr\left[\left|\frac{1}{B}\sum_{i=1}^{B} Y_i - p_z\right| > \gamma\right] \leq \\
&2 \cdot e^{-2^{m \cdot \log^2(nd)}} + \Pr\left[\left|\frac{1}{B}\sum_{i=1}^{B} X_i - \frac{1}{B}\sum_{i=1}^{B} Y_i\right| > 2\gamma\right],
\end{aligned}$$

where the first inequality follows from the triangle inequality, the second inequality follows from the union bound, and the third inequality follows from Equations (3.2) and (3.3). Thus, it suffices to prove that there exists a negligible function $\mu = \mu(n)$ such that

$$\Pr\left[\left|\frac{1}{B}\sum_{i=1}^{B} X_i - \frac{1}{B}\sum_{i=1}^{B} Y_i\right| > 2\gamma\right] \leq \frac{\mu}{2^m}.$$

[5]The Chernoff bound states that for any identical and independent random variables $X_1, \ldots, X_B$, such that $X_i \in \{0,1\}$ for each i, if we denote by $p = \mathbb{E}[X_i]$ then $\Pr[\left|\frac{1}{B}\sum_{i=1}^{B} X_i - p\right| \geq \delta] \leq e^{-\frac{\delta^2}{3}B}$.

To this end, notice that for a random $H \leftarrow \mathcal{H}$,

$$
\begin{aligned}
&\mathsf{SD}\left((X_1,\ldots,X_B),(Y_1,\ldots,Y_B)\right) \leq \\
&\sum_{i=1}^{B} \mathsf{SD}\left((X_1,\ldots,X_{i-1},X_i,Y_{i+1},\ldots,Y_B),(X_1,\ldots,X_{i-1},Y_i,Y_{i+1},\ldots,Y_B)\right) = \\
&\sum_{i=1}^{B} \mathsf{SD}\left((X_1,\ldots,X_{i-1},X_i),(X_1,\ldots,X_{i-1},Y_i)\right) \leq \\
&B \cdot \mathsf{SD}\left((X_1,\ldots,X_{B-1},X_B),(X_1,\ldots,X_{B-1},Y_B)\right) \leq \\
&B \cdot nd \cdot \frac{(B-1)nd}{Nnd} \leq \\
&\frac{B^2 \cdot nd}{N} \leq \\
&\frac{2^{2m\log^3(nd)} \cdot nd}{2^{m\log^4(nd)}} \leq \\
&2^{-m\log^3(nd)},
\end{aligned}
$$

where the first equation follows from a standard hybrid argument. The second equation follows from the fact that $Y_{i+1},\ldots,Y_B$ are independent of $X_1,\ldots,X_i,Y_i$. The third equation follows from the fact that the statistical distance between $(X_1,\ldots,X_{i-1},X_i)$ and $(X_1,\ldots,X_{i-1},Y_i)$ is maximal for $i = B$. The forth equation follows from the fact that $(X_1,\ldots,X_{B-1},X_B)$ and $(X_1,\ldots,X_{B-1},Y_B)$ are identically distributed if the following event, which we denote by Good, occurs: Recall that each X_i depends only on nd *random* coordinates of $H \leftarrow \mathcal{H}$. We say that Good occurs if the nd coordinates that X_B depends on are disjoint from all the $nd(B-1)$ coordinates that $X_1,\ldots,X_{B-1}$ depend on. The forth equation follows from the fact that $\Pr[\neg\mathsf{Good}] \leq nd \cdot \frac{(B-1)nd}{Nnd}$. The rest of the equations follow from basic arithmetics and from the definition of B and N.

In particular, this implies that

$$
\mathsf{SD}\left(\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right),\left(\frac{1}{B}\sum_{i=1}^{B} Y_i\right)\right) \leq 2^{-m\log^3(nd)}. \tag{3.4}
$$

Consider the algorithm $\mathcal{D}$ that given p'_z, supposedly distributed according to $\frac{1}{B}\sum_{i=1}^{B} X_i$ or distributed according to $\frac{1}{B}\sum_{i=1}^{B} Y_i$, outputs 1 if $|p'_z - p_z| \leq \gamma$, and otherwise outputs 0. Equation (3.3) implies that

$$
\Pr\left[\mathcal{D}\left(\frac{1}{B}\sum_{i=1}^{B} Y_i\right) = 1\right] \geq 1 - e^{-2^{m\cdot\log^2(nd)}}.
$$

This together with Equation (3.4), implies that

$$
\Pr\left[\mathcal{D}\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right) = 1\right] \geq 1 - e^{-2^{m\cdot\log^2(nd)}} - 2^{-m\log^3(nd)} \geq 1 - 2^{-m\log^2(nd)},
$$

which by the definition of $\mathcal{D}$, implies that

$$
\Pr\left[\left|\frac{1}{B}\sum_{i=1}^{B} X_i - p_z\right| \leq \gamma\right] \geq 1 - 2^{-m\log^2(nd)}.
$$

This in particular implies that

$$\Pr\left[\left|\frac{1}{B}\sum_{i=1}^{B} X_i - \frac{1}{B}\sum_{i=1}^{B} Y_i\right| \leq 2\gamma\right] \geq 1 - 2 \cdot 2^{-m\log^2(nd)},$$

as desired. ■

3.1 Proof of Lemma 3.2

Assume towards contradiction that there exists a constant $c \in \mathbb{N}$ such that for every constant $b \in \mathbb{N}$ and for every set $\mathcal{H}_0 \subseteq \mathcal{H}$ of size $\frac{|\mathcal{H}|}{2}$ there exists $H \in \mathcal{H}_0$ such that Π_H is *not* (t, δ', s)-adaptively secure, for $\delta' = \delta + \frac{b}{n^c}$. This implies that for every constant $b \in \mathbb{N}$ there exists a set $\mathcal{H}_0 \subseteq \mathcal{H}$ of size $\frac{|\mathcal{H}|}{2}$ such that for every $H \in \mathcal{H}_0$ there exists an adversary $\mathcal{A}^H$ that adaptively corrupts at most t parties and satisfies

$$\mathsf{succ}_{\mathsf{s}}((\mathcal{A}^H)_{\Pi_H}) \geq \delta'.$$

This implies that for every constant $b \in \mathbb{N}$ there exists a set $M \subseteq \{0,1\}^m$ of size s such that for at least $\binom{2^m}{s}$-fraction of the H's in $\mathcal{H}_0$ the adversary $\mathcal{A}^H$ satisfies that $\mathsf{succ}_M((\mathcal{A}^H)_{\Pi_H}) \geq \delta'$. We denote this set of H's by $\mathcal{H}_1$. Notice that

$$|\mathcal{H}_1| \geq \frac{|\mathcal{H}|}{2^{2^m}} = 2^{dnNL - 2^m}. \tag{3.5}$$

The proof proceeds as follows: we show how to use these adversaries $\{\mathcal{A}^H\}_{H \in \mathcal{H}_1}$ to construct an adversary $\mathcal{A}$ such that $\mathsf{succ}_M(\mathcal{A}_\Pi) \geq \delta' - \mu$, where $\mu = \mu(n)$ is a negligible function that does not depend on δ', and in particular is independent of the constant $b \in \mathbb{N}$. Hence we can take a constant b such that $\delta' - \mu > \delta$, contradicting the (t, δ, s)-adaptive security of Π.

The idea is for the adversary $\mathcal{A}$ to simulate the execution of one of the $\mathcal{A}^H$'s. The problem is that we do not know ahead of time which H will be consistent with the transcript of the protocol, since we have no control over the (long) random messages of the honest parties. We overcome this problem by choosing H *adaptively*. Namely, at any point in the protocol, $\mathcal{A}$ simulates a random adversary $\mathcal{A}^H$, where H is a random matrix that is consistent (in some sense that we explain later) with the transcript up to that point.

More specifically, for every $i \in [d]$ and every $j \in [n]$, we denote by $\mathcal{H}_{i,j-1}$ the set of matrices that are consistent with the transcript up until the point where the j^{th} message of the i^{th} round is about to be sent. Fix any round $i \in [d]$ and any $j \in [n]$. Roughly speaking, in the i^{th} round before the j^{th} message is to be sent, the adversary $\mathcal{A}$ simulates $\mathcal{A}^{H^*}$ where $H^* \leftarrow \mathcal{H}_{i,j-1}$ is chosen uniformly at random. If $\mathcal{A}^{H^*}$ corrupts a party P_u then $\mathcal{A}$ also corrupts P_u. If $\mathcal{A}^{H^*}$ sends a message r_i^* on behalf of a corrupted party P_u, then $\mathcal{A}$ will send the message $\mathsf{R}_i^* = H^*(i, u, \mathsf{r}_i^*)$ on behalf of party P_u. In this case, we define $\mathcal{H}_{i,j}$ to be all the matrices in $\mathcal{H}_{i,j-1}$ which are consistent with the transcript so far and agree with H^* on row (i, u). If $\mathcal{A}^{H^*}$ asks an honest party P_u to send its message, the adversary $\mathcal{A}$ will also ask honest party P_u to send a message. Upon receiving a message R^* from P_u, we choose a random matrix $H \leftarrow \mathcal{H}_{i,j-1}$ that is consistent with the transcript so far, and set $\mathcal{H}_{i,j}$ to be all the matrices in $\mathcal{H}_{i,j-1}$ that are consistent with the transcript so far, and where we fix the (i, u) row to be the (i, u) row of H.

Before giving the precise description of the adversary $\mathcal{A}$, we provide some useful notation. We denote the transcript generated in an execution of the protocol Π with an adversary $\mathcal{A}$ by $\mathsf{Trans}_{\mathcal{A}}$. Note that $\mathsf{Trans}_{\mathcal{A}}$ consists of d vectors (one per each round), where each vector consists of n pairs of the form

$$((\mathsf{P}_{j_1}, \mathsf{R}_1), \ldots, (\mathsf{P}_{j_n}, \mathsf{R}_n)),$$

where $\mathsf{R}_1, \dots \mathsf{R}_n \in \{0,1\}^L$ and $j_1, \dots, j_n \in [n]$, where the order means that in this round party P_{j_1} sent his message first, then party P_{j_2} sent his message, and so on (recall that in our model, the adversary has control over the scheduling of the messages within each round). We sometimes consider a partial transcript $\mathsf{Trans}_{i,j}$ (i.e., a prefix of a transcript) which corresponds to a partial execution of the protocol Π with the adversary $\mathcal{A}$ until after the j^{th} message in the i^{th} round was sent. For $H \in \mathcal{H}$, we denote by

$$\mathsf{MAP}_H \colon [d] \times [n] \times \{0,1\}^L \to \{0,1\}^\ell \cup \{\perp\}$$

the mapping that takes as input a row number $(i,j) \in [d] \times [n]$ and a (long) message in $\mathsf{R} \in \{0,1\}^L$, and converts it into a (short) message $\mathsf{r} \in \{0,1\}^\ell$ such that $H(i,j,\mathsf{r}) = \mathsf{R}$. If no such message exists, MAP_H outputs $\perp$.

Let $\mathsf{Trans}_{i,j}$ be a (long) partial transcript of Π. The corresponding (short) transcript of Π_H, denoted by $\mathsf{MAP}_H(\mathsf{Trans}_{i,j})$, is defined recursively, as follows. Let $\mathsf{Trans}_{i,j} = (\mathsf{Trans}_{i,j-1}, (\mathsf{P}_u, \mathsf{R}))$. Then,

$$\mathsf{MAP}_H(\mathsf{Trans}_{i,j}) = \big(\mathsf{MAP}_H(\mathsf{Trans}_{i,j-1}), (\mathsf{P}_u, \mathsf{MAP}_H(i,u,\mathsf{R}))\big).$$

We initialize $\mathsf{Trans}_{1,0} = \emptyset$ and $\mathcal{H}_{1,0} = \mathcal{H}_1$. Using this notation, a formal description of the adversary $\mathcal{A}$ is given in Figure 1.

In order to prove Lemma 3.2 (and thus to complete the proof of Theorem 1.1), it suffices to prove the following lemma.

Lemma 3.4. *The adversary $\mathcal{A}$ makes at most t adaptively-chosen corruptions, and $\mathsf{succ}_M(\mathcal{A}_\Pi) \geq \delta' - \mu(n)$, where $\mu = \mu(n)$ is a negligible function such that $\delta' - \mu > \delta$ for every $n \in \mathbb{N}$.*

Proof. We first note that $\mathcal{A}$ always makes at most t corruptions. This follows from the fact that $\mathcal{A}$ is always consistent with some adversary $\mathcal{A}^H$, for some $H \in \mathcal{H}_1$ (or else $\mathcal{A}$ aborts), and by our assumption, every $\mathcal{A}^H$ makes at most t corruptions.

We next prove that $\mathsf{succ}_M(\mathcal{A}_\Pi) \geq \delta' - \mathsf{negl}(n)$. Recall that we denote by $\mathsf{Trans}_\mathcal{A}$ the random variable that corresponds to the transcript generated by running the protocol Π with the adversary $\mathcal{A}$ (described in Figure 1).

Let $\mathsf{Trans}_{\mathsf{ideal}}$ be an "ideal" transcript, generated as follows: Choose a random $H \leftarrow \mathcal{H}_1$, run the protocol Π_H with the adversary $\mathcal{A}^H$. Denote the resulting transcript by Trans_H. As above, Trans_H consists of d vectors (one per each round), where each vector consists of n pairs of the form

$$((\mathsf{P}_{j_1}, \mathsf{r}_1), \dots, (\mathsf{P}_{j_n}, \mathsf{r}_n)),$$

where $\mathsf{r}_1, \dots \mathsf{r}_n \in \{0,1\}^\ell$ and $j_1, \dots, j_n \in [n]$. We define

$$\mathsf{Trans}_{\mathsf{ideal}} = H(\mathsf{Trans}_H)$$

where $H(\mathsf{Trans}_H)$ is the transcript obtained by applying $H(i,u,\cdot)$ to each element in the $(i,u)^{\text{th}}$ row of Trans_H. Formally, $H(\mathsf{Trans}_H)$ is defined recursively, as follows: For every $i \in [d]$ and every $j \in [n]$, we let $\mathsf{Trans}_{H,i,j}$ denote the transcript Trans_H up until after the j^{th} message in the i^{th} round is sent. We define $H(\mathsf{Trans}_{H,i,j})$ recursively, as follows: For $\mathsf{Trans}_{H,i,j} = (\mathsf{Trans}_{H,i,j-1}, (\mathsf{P}_u, \mathsf{r}))$, we define

$$H(\mathsf{Trans}_{H,i,j}) = (H(\mathsf{Trans}_{H,i,j-1}), (\mathsf{P}_u, H(i,u,\mathsf{r}))).$$

In order to prove Lemma 3.4 it suffices to prove the following claim.

The adversary $\mathcal{A}(\mathsf{Trans}_{i,j-1})$ before the j^{th} message of round i

1. If $\mathcal{H}_{i,j-1} = \emptyset$, output $\perp$ and HALT.
2. Choose $H^* \leftarrow \mathcal{H}_{i,j-1}$ uniformly at random. Let $\mathsf{Trans}_{H^*} = \mathsf{MAP}_{H^*}(\mathsf{Trans}_{i,j-1})$ denote the (short) transcript in the protocol Π_{H^*} that corresponds to the (long) transcript $\mathsf{Trans}_{i,j-1}$.
3. If $\mathcal{A}^{H^*}(\mathsf{Trans}_{H^*})$ corrupts a party P_u then corrupt P_u.
4. If $\mathcal{A}^{H^*}(\mathsf{Trans}_{H^*})$ sends a message on behalf of a corrupt party P_u, then do the following:
   (a) Denote by $\mathsf{r}^* \in \{0,1\}^\ell$ the message that $\mathcal{A}^{H^*}(\mathsf{Trans}_{H^*})$ sends on behalf of P_u. Let $\mathsf{R}^* = H^*(i,u,\mathsf{r}^*)$.
   (b) Send the message R^* on behalf of party P_u.
   (c) Add $(\mathsf{P}_u, \mathsf{R}^*)$ to the partial transcript. Namely, set
   $$\mathsf{Trans}_{i,j} = (\mathsf{Trans}_{i,j-1}, (\mathsf{P}_u, \mathsf{R}^*))\,.$$
   (d) Define $\mathcal{H}_{i,j}$ to be the set of all $H \in \mathcal{H}_{i,j-1}$ that are consistent with the transcript so far, and for which $H(i,u,\cdot) = H^*(i,u,\cdot)$. Namely, set
   $$\begin{aligned}\mathcal{H}_{i,j} = \big\{H \in \mathcal{H}_{i,j-1} \mid \forall \mathsf{r}\colon\ & H(i,u,\mathsf{r}) = H^*(i,u,\mathsf{r}), \text{and}\\ & \mathcal{A}^H(\mathsf{Trans}_H) \text{ sends } \mathsf{r}^* \text{ on behalf of } \mathsf{P}_u,\\ & \text{where } \mathsf{Trans}_H = \mathsf{MAP}_H(\mathsf{Trans}_{i,j-1})\big\}.\end{aligned}$$
5. If $\mathcal{A}^{H^*}(\mathsf{Trans}_{H^*})$ does not corrupt, and orders an honest party P_u to send a message, then do the following:
   (a) Do not corrupt, and order honest party P_u to send a message. Denote the message it sends by R^*.
   (b) Add $(\mathsf{P}_u, \mathsf{R}^*)$ to the partial transcript. Namely, set
   $$\mathsf{Trans}_{i,j} = (\mathsf{Trans}_{i,j-1}, (\mathsf{P}_u, \mathsf{R}^*)).$$
   (c) Choose a random matrix
   $$\begin{aligned}H' \leftarrow \{H \in \mathcal{H}_{i,j-1} \mid\ & \mathcal{A}^H(\mathsf{Trans}_H) \text{ orders honest } \mathsf{P}_u \text{ to send a message, and}\\ & \exists \mathsf{r} \text{ s.t. } H(i,u,\mathsf{r}) = \mathsf{R}^*\}.\end{aligned}$$
   (d) Define $\mathcal{H}_{i,j}$ to be the set of all $H \in \mathcal{H}_{i,j-1}$ that are consistent with the transcript so far, and agree with H' on row (i,u). That is,
   $$\begin{aligned}\mathcal{H}_{i,j} = \{H \in \mathcal{H}_{i,j-1} \mid \forall \mathsf{r}\colon\ & H(i,u,\mathsf{r}) = H'(i,u,\mathsf{r}), \text{ and}\\ & \mathcal{A}^H(\mathsf{Trans}_H) \text{ orders honest } \mathsf{P}_u \text{ to send a message}\}.\end{aligned}$$
6. If $j = n$, set $\mathcal{H}_{i+1,0} = \mathcal{H}_{i,j}$ and $\mathsf{Trans}_{i+1,0} = \mathsf{Trans}_{i,j}$.

Figure 1: The adversary $\mathcal{A}$ before the j^{th} message of round i.

Claim 3.5.

$$\mathsf{SD}(\mathsf{Trans}_{\mathcal{A}}, \mathsf{Trans}_{\mathsf{ideal}}) = \mu(n),$$

where $\mu(n)$ is a negligible function that does not depend on δ'.

Proof of Claim 3.5. We prove Claim 3.5 using a hybrid argument. Specifically, we define a sequence of $d \cdot (n+1)$ experiments. For every $i \in [d]$ and every $j \in \{0, 1, \ldots, n\}$, we define the experiment $\mathsf{Exp}^{(i,j)}$ as follows:

1. Generate $\mathsf{Trans}_{i,j}$ and $\mathcal{H}_{i,j}$, as defined in Figure 1.
2. Choose a random $H \leftarrow \mathcal{H}_{i,j}$, and let $\mathsf{Trans}_{H,i,j} = \mathsf{MAP}_H(\mathsf{Trans}_{i,j})$.
3. Run the protocol Π_H with the adversary $\mathcal{A}^H$, given the partial transcript $\mathsf{Trans}_{H,i,j}$. Namely, run Π_H with $\mathcal{A}^H$ from after the j^{th} message in the i^{th} round was sent, and assume the transcript up until that point is $\mathsf{Trans}_{H,i,j}$. Denote the entire transcript (including $\mathsf{Trans}_{H,i,j}$) by Trans_H.
4. Output $H(\mathsf{Trans}_H)$.

Notice that

$$\mathsf{Exp}^{(d,n)} \equiv \mathsf{Trans}_{\mathcal{A}},$$

and

$$\mathsf{Exp}^{(1,0)} \equiv \mathsf{Trans}_{\mathsf{ideal}}.$$

It remains to argue that for every $i \in [d]$ and every $j \in [n]$ the statistical distance between any two consecutive experiments $\mathsf{Exp}^{(i,j-1)}$ and $\mathsf{Exp}^{(i,j)}$ is small. In particular, it suffices to prove that

$$\mathsf{SD}\left(\mathsf{Exp}^{(i,j-1)}, \mathsf{Exp}^{(i,j)}\right) = \mathsf{negl}(n,d), \tag{3.6}$$

for a negligible function that is independent of δ'. The reason is that given this inequality, we obtain that

$$\mathsf{SD}(\mathsf{Trans}_{\mathcal{A}}, \mathsf{Trans}_{\mathsf{ideal}}) \leq \sum_{i\in[d],j\in[n]} \mathsf{SD}(\mathsf{Exp}^{(i,j-1)}, \mathsf{Exp}^{(i,j)}) \leq d \cdot n \cdot \mathsf{negl}(n,d) = \mathsf{negl}(n,d) = \mathsf{negl}(n),$$

which completes the claim. We note that the first inequality follows from the union bound together with the fact that $\mathsf{Exp}^{(i,n)} = \mathsf{Exp}^{(i+1,0)}$ for every $i \in [d-1]$ (see Figure 1 Item 6).

We proceed with the proof of Equation (3.6). To this end, fix any $i \in [d]$ and $j \in [n]$. Let $k \triangleq (i-1) \cdot d + j$. Note that in both $\mathsf{Exp}^{i,j-1}$ and $\mathsf{Exp}^{i,j}$ the first $k-1$ messages are generated according to $\mathsf{Trans}_{\mathcal{A}}$.

Denote by $\mathsf{corrupt}_k$ the event that the k^{th} message is sent by a corrupted party. We first argue that

$$\Pr\left[\mathsf{corrupt}_k \mid \mathsf{Exp}^{(i,j-1)}\right] = \Pr\left[\mathsf{corrupt}_k \mid \mathsf{Exp}^{(i,j)}\right].$$

This follows immediately from the definition of the two experiments. In $\mathsf{Exp}^{(i,j)}$ (according to Figure 1, Items 2-4), before sending the k^{th} message, a random function is chosen $H^* \leftarrow \mathcal{H}_{i,j-1}$ and the k^{th} message is sent by a corrupted party if and only if $\mathcal{A}^{H^*}$ chooses the k^{th} message to be sent by a corrupted party (given the transcript so far). Note that in $\mathsf{Exp}^{(i,j-1)}$, the same exact process occurs (see Items 2 to 4 at the beginning of the proof of Claim 3.5).

We next argue

$$\mathsf{SD}\left(\left(\mathsf{Exp}^{(i,j-1)} \mid \mathsf{corrupt}_k\right), \left(\mathsf{Exp}^{(i,j)} \mid \mathsf{corrupt}_k\right)\right) = 0. \tag{3.7}$$

To see why Equation (3.7) holds, note that according to Figure 1 (see Items 2 to 4), the k^{th} message in $\left(\mathsf{Exp}^{(i,j)} \mid \mathsf{corrupt}_k\right)$ is chosen by sampling a random matrix $H^* \leftarrow \mathcal{H}_{i,j-1}$ conditioned on the fact that the k^{th} message sent in Π_{H^*} with $\mathcal{A}^{H^*}$ is sent by a corrupted party. Denote this corrupted party by P_u and denote by r^* the message that $\mathcal{A}^{H^*}$ sends on behalf of P_u. Then the k^{th} message in $\mathsf{Exp}^{(i,j)}$ is set to be $H^*(i,u,\mathsf{r}^*)$. Note that the k^{th} message in $\mathsf{Exp}^{(i,j-1)}$ is chosen in exactly the same way (see Items 2 to 4 at the beginning of the proof of Claim 3.5). Moreover, the distribution of the set $\mathcal{H}_{i,j}$ in both cases is identical, which implies that the distributions of the rest of the messages in $\left(\mathsf{Exp}^{(i,j-1)} \mid \mathsf{corrupt}_k\right)$ and in $\left(\mathsf{Exp}^{(i,j)} \mid \mathsf{corrupt}_k\right)$ are identical as well.

It remains to prove that

$$\mathsf{SD}\left(\left(\mathsf{Exp}^{(i,j-1)} \mid \neg\mathsf{corrupt}_k\right), \left(\mathsf{Exp}^{(i,j)} \mid \neg\mathsf{corrupt}_k\right)\right) = \mathsf{negl}(n,d), \tag{3.8}$$

for a negligible function that does not depend on δ'.

Recall that in $\left(\mathsf{Exp}^{(i,j)} \mid \neg\mathsf{corrupt}_k\right)$ the k^{th} message is uniformly distributed in $\{0,1\}^L$. Denote by R' the k^{th} message in $\left(\mathsf{Exp}^{(i,j-1)} \mid \neg\mathsf{corrupt}_k\right)$. Recall that R' is distributed as follows: Choose a random $H \leftarrow \mathcal{H}_{i,j-1}$ such that the adversary $\mathcal{A}$ (given the partial transcript $\mathsf{MAP}_H(\mathsf{Trans}_{i,j-1})$) orders an honest party P_u to send the j^{th} message in the i^{th} round. Choose a random $r' \leftarrow \{0,1\}^\ell$, and and set $R' = H(i,u,r')$.

Notice that in order to prove Equation (3.8), it suffices to prove that

$$\mathsf{SD}(R', \mathbf{U}_L) = \mathsf{negl}(n,d), \tag{3.9}$$

for a negligible function that does not depend on δ'.

Fix $\varepsilon = 2^{-\log^2(dn)}$. We first argue that in order to prove Equation (3.9) it suffices to prove that,

$$\Pr\left[|\mathcal{H}_{i,j-1}| \geq \frac{2^{dnNL}}{2^{(k-1)NL} \cdot \left(\frac{4nN}{\varepsilon^2}\right)^{k-1} \cdot 2^{2^m}}\right] \geq (1-\varepsilon)^{k-1}, \tag{3.10}$$

where the probability is over the randomness of the honest parties.

To this end, suppose that Equation (3.10) holds. Denote by E the event that

$$|\mathcal{H}_{i,j-1}| \geq \frac{2^{dnNL}}{2^{(k-1)NL} \cdot \left(\frac{4nN}{\varepsilon^2}\right)^{k-1} \cdot 2^{2^m}}. \tag{3.11}$$

By Equation (3.10),

$$\Pr[\mathsf{E}] \geq (1-\varepsilon)^{k-1} = 1 - \mathsf{negl}(n,d),^{6}$$

[6] Note that this negligible function does not depend on δ'.

where the latter equality follows from the definition of $\varepsilon = 2^{-\log^2(nd)}$ and from the fact that $k \leq nd$. Therefore,

$$\begin{aligned}
&\mathsf{SD}(R', \mathbf{U}_L) \leq \\
&\mathsf{SD}((R' \mid \mathsf{E}), \mathbf{U}_L) \cdot \Pr[\mathsf{E}] + \mathsf{SD}((R' \mid \neg\mathsf{E}), \mathbf{U}_L) \cdot \Pr[\neg\mathsf{E}] \leq \\
&\mathsf{SD}((R' \mid \mathsf{E}), \mathbf{U}_L) + \Pr[\neg\mathsf{E}] = \\
&\mathsf{SD}((R' \mid \mathsf{E}), \mathbf{U}_L) + \mathsf{negl}(n, d).
\end{aligned}$$

Thus, in order to prove Equation (3.9) it suffices to prove that

$$\mathsf{SD}((R' \mid \mathsf{E}), \mathbf{U}_L) = \mathsf{negl}(n, d),$$

for a negligible function that does not depend on δ'. This, together with Claim 2.6, implies that it suffices to prove that

$$\mathsf{entropy}(\mathsf{R}' \mid \mathsf{E}) \geq L - \varepsilon. \tag{3.12}$$

To this end, let $H \leftarrow \mathcal{H}_{i,j-1}$. Then,

$$\begin{aligned}
&\mathsf{entropy}(H \mid \mathsf{E}) \geq \\
&dnNL - (k-1)NL - (k-1)(\log 4nN) - (k-1)\log\frac{1}{\varepsilon^2} - 2^m = \\
&(dn - k + 1)NL - (k-1)\left(\log 4nN + 2\log\frac{1}{\varepsilon}\right) - 2^m,
\end{aligned}$$

where the first inequality follows from Equation (3.11) together with the definition of entropy (see Definition 2.5), and the latter equality follows from basic arithmetics.

For every $\alpha \in [d]$ and every $\beta \in [n]$, we denote by $\mathsf{Row}_{\alpha,\beta} \in \{0,1\}^{NL}$ the random variable obtained by choosing a random matrix $H \leftarrow \mathcal{H}_{i,j-1}$, and setting $\mathsf{Row}_{\alpha,\beta}$ to be the $(\alpha,\beta)^{\text{th}}$ row of H. Note that

$$\mathsf{entropy}(H \mid \mathsf{E}) \leq \sum_{\alpha\in[d],\beta\in[n]} \mathsf{entropy}(\mathsf{Row}_{\alpha,\beta} \mid \mathsf{E}) \leq \mathsf{entropy}(\mathsf{Row}_{i,u} \mid \mathsf{E}) + NL(dn - k),$$

where the first inequality follows from the basic property of Shannon entropy, that for any random variables X and Y, it holds that $\mathsf{entropy}(X, Y) \leq \mathsf{entropy}(X) + \mathsf{entropy}(Y)$, and the second equality follows from the fact that $k-1$ of the rows in $\mathcal{H}_{i,j-1}$ are fixed. This, together with the equations above, implies that

$$\begin{aligned}
&\mathsf{entropy}(\mathsf{Row}_{i,u} \mid \mathsf{E}) \geq \\
&(dn - k + 1)NL - (k-1)\left(\log 4nN + 2\log\frac{1}{\varepsilon}\right) - 2^m - NL(dn-k) = \\
&NL - (k-1)\left(\log 4nN + 2\log\frac{1}{\varepsilon}\right) - 2^m = \\
&NL - (k-1)\left(\log 4nN + 2\log^2(dn)\right) - 2^m.
\end{aligned}$$

Recall that $(R' \mid \mathsf{E})$ is the random variable defined by choosing $H \leftarrow \mathcal{H}_{i,j-1}$ (where we assume that event E holds for $\mathcal{H}_{i,j-1}$), choosing a random $\alpha \leftarrow [N]$, and setting $R' = H(i, u, \alpha)$. Thus,

$$\begin{aligned}
&\mathsf{entropy}(R' \mid \mathsf{E}) \geq \\
&\frac{NL - (k-1)(\log 4nN + 2\log^2(dn)) - 2^m}{N} = \\
&L - \frac{(k-1)(\log 4nN + 2\log^2(dn)) + 2^m}{N} \geq \\
&L - \varepsilon,
\end{aligned}$$

contradicting Equation (3.12), where the latter inequality follows from the definition of N (see Equation (3.1)).

It remains to prove Equation (3.10). We prove that Equation (3.10) holds for any $(i,j) \in [d] \times \{0, 1, \ldots, n\}$. The proof is by induction on $k = (i-1) \cdot n + j$. The base case is $k = 0$, which corresponds to $(i,j) = (1,0)$. In this case, it is always holds that

$$|\mathcal{H}_{i,j}| = |\mathcal{H}_{1,0}| = |\mathcal{H}_1| \geq \frac{2^{dnNL}}{2^{2^m}},$$

where the latter inequality follows from the definition of $\mathcal{H}_1$.

Next, assume that Equation (3.10) holds for $k-1$, and we prove that it holds for k. Fix $i \in [d]$ and $j \in [n]$ such that $k = (i-1) \cdot n + j$. By the induction hypothesis,

$$\Pr\left[|\mathcal{H}_{i,j-1}| \geq \frac{2^{dnNL}}{2^{(k-1)NL} \cdot \left(\frac{4nN}{\varepsilon^2}\right)^{k-1} \cdot 2^{2^m}}\right] \geq (1-\varepsilon)^{k-1}.$$

We denote by E the event that indeed

$$|\mathcal{H}_{i,j-1}| \geq \frac{2^{dnNL}}{2^{(k-1)NL} \cdot \left(\frac{4nN}{\varepsilon^2}\right)^{k-1} \cdot 2^{2^m}}. \tag{3.13}$$

Thus, by our induction hypothesis,

$$\Pr[\mathsf{E}] \geq (1-\varepsilon)^{k-1}.$$

In what follows, fix *any* $\mathcal{H}_{i,j-1}$ such that event E holds. By Claim 2.4, it suffices to prove that

$$\mathbb{E}\left[|\mathcal{H}_{i,j}|\right] \geq \frac{|\mathcal{H}_{i,j-1}|}{2^{NL} \cdot \frac{4nN}{\varepsilon}}.$$

To this end, we define $\mathcal{H}_{i,j}$ in two stages. We first define $\mathcal{H}_{i,j}^{\mathsf{tmp}}$ (corresponding to Items 2 and 3 in Figure 1), as follows: Recall that $\mathcal{H}_{i,j-1}$ is generated with an associated partial transcript $\mathsf{Trans}_{i,j-1}$ corresponding to the protocol Π. The set $\mathcal{H}_{i,j}^{\mathsf{tmp}} \subseteq \mathcal{H}_{i,j-1}$ is defined by choosing a random $H^* \leftarrow \mathcal{H}_{i,j-1}$, and setting $\mathsf{Trans}_{H^*} = \mathsf{MAP}_{H^*}(\mathsf{Trans}_{i,j-1})$. If $\mathcal{A}^{H^*}(\mathsf{Trans}_{H^*})$ corrupts a party P_u or sends a message r^* on behalf of a corrupted party P_u, then $\mathcal{H}_{i,j}^{\mathsf{tmp}}$ consists of all the matrices $H \in \mathcal{H}_{i,j-1}$ such that $\mathcal{A}^H(\mathsf{Trans}_H)$ behaves the same way, i.e., corrupts P_u or sends the message r^* on behalf of the corrupted P_u. If $\mathcal{A}^{H^*}(\mathsf{Trans}_{H^*})$ does not corrupt and orders an honest party P_u to send a message, then $\mathcal{H}_{i,j}^{\mathsf{tmp}}$ consists of all the matrices $H \in \mathcal{H}_{i,j-1}$ such that $\mathcal{A}^H(\mathsf{Trans}_H)$ behaves the same way, i.e., does not corrupt and orders the honest party P_u to send a message.

Claim 2.4 implies that

$$\mathbb{E}\left[\left|\mathcal{H}_{i,j}^{\mathsf{tmp}}\right|\right] \geq \frac{|\mathcal{H}_{i,j-1}|}{2nN}.$$

Moreover, Claim 2.4 implies that for every $\gamma > 0$

$$\Pr\left[\left|\mathcal{H}_{i,j}^{\mathsf{tmp}}\right| \geq \frac{\gamma \cdot |\mathcal{H}_{i,j-1}|}{2nN}\right] \geq 1 - \gamma. \tag{3.14}$$

Set $\gamma = \varepsilon$. Let $\mathsf{E}_{\mathsf{tmp}}$ denote the event that

$$\left|\mathcal{H}_{i,j}^{\mathsf{tmp}}\right| \geq \frac{\varepsilon \cdot |\mathcal{H}_{i,j-1}|}{2nN}. \tag{3.15}$$

Then

$$\Pr[\mathsf{E}_{\mathsf{tmp}}] \geq 1 - \varepsilon.$$

Note that,

$$\begin{aligned}
&\mathbb{E}\left[|\mathcal{H}_{i,j}|\right] = \\
&\mathbb{E}\left[|\mathcal{H}_{i,j}| \mid \mathsf{E}_{\mathsf{tmp}}\right] \cdot \Pr[\mathsf{E}_{\mathsf{tmp}}] + \mathbb{E}\left[|\mathcal{H}_{i,j}| \mid \neg\mathsf{E}_{\mathsf{tmp}}\right] \cdot \Pr[\neg\mathsf{E}_{\mathsf{tmp}}] \geq \\
&\mathbb{E}\left[|\mathcal{H}_{i,j}| \mid \mathsf{E}_{\mathsf{tmp}}\right] \cdot (1 - \varepsilon).
\end{aligned}$$

Therefore, it suffices to prove that

$$\mathbb{E}\left[|\mathcal{H}_{i,j}| \mid \mathsf{E}_{\mathsf{tmp}}\right] \geq \frac{|\mathcal{H}_{i,j-1}|}{2^{NL} \cdot \frac{4nN}{\varepsilon} \cdot (1-\varepsilon)}. \tag{3.16}$$

Fix any $\mathcal{H}_{i,j}^{\mathsf{tmp}}$ such that event $\mathsf{E}_{\mathsf{tmp}}$ holds. We distinguish between two cases.

Case 1: The $(i,j)^{\text{th}}$ message in $\mathcal{H}_{i,j}^{\mathsf{tmp}}$ is sent by a corrupted party P_u. Recall that in this case the set $\mathcal{H}_{i,j} \subseteq \mathcal{H}_{i,j}^{\mathsf{tmp}}$ is defined by choosing a random $H^* \leftarrow \mathcal{H}_{i,j}^{\mathsf{tmp}}$ and setting

$$\mathcal{H}_{i,j} = \left\{H \in \mathcal{H}_{i,j}^{\mathsf{tmp}} \mid \forall \mathsf{r}\colon\ H(i,u,\mathsf{r}) = H^*(i,u,\mathsf{r})\right\}.$$

Claim 2.4 implies that

$$\mathbb{E}\left[|\mathcal{H}_{i,j}|\right] \geq \frac{\left|\mathcal{H}_{i,j}^{\mathsf{tmp}}\right|}{2^{NL}} \geq \frac{\varepsilon \cdot |\mathcal{H}_{i,j-1}|}{2nN \cdot 2^{NL}} = \frac{|\mathcal{H}_{i,j-1}|}{\frac{2nN}{\varepsilon} \cdot 2^{NL}} \geq \frac{|\mathcal{H}_{i,j-1}|}{\frac{4nN}{\varepsilon} \cdot 2^{NL}(1-\varepsilon)},$$

as desired. The first inequality follows from Claim 2.4, the second inequality follows from Equation (3.15), and the last inequality follows from the fact that $2(1-\varepsilon) > 1$.

Case 2: The $(i,j)^{\text{th}}$ message in $\mathcal{H}_{i,j}^{\mathsf{tmp}}$ is sent by an honest party P_u. Recall that in this case, a message $\mathsf{R}^* \leftarrow \{0,1\}^L$ is chosen uniformly at random (supposedly by P_u in the protocol Π with the adversary $\mathcal{A}$). Then a random matrix $H' \in \mathcal{H}_{i,j}^{\mathsf{tmp}}$ is chosen such that there exists r for which $H'(i,u,\mathsf{r}) = \mathsf{R}^*$, and the set $\mathcal{H}_{i,j}$ is defined to be

$$\mathcal{H}_{i,j} = \left\{H \in \mathcal{H}_{i,j}^{\mathsf{tmp}} \mid \forall \mathsf{r}\colon\ H(i,u,\mathsf{r}) = H'(i,u,\mathsf{r})\right\}.$$

For every R^* we denote by

$$\mathcal{H}_{i,j,\mathsf{R}^*} = \left\{H \in \mathcal{H}_{i,j}^{\mathsf{tmp}} \mid \exists \mathsf{r}\colon\ H(i,u,\mathsf{r}) = \mathsf{R}^*\right\}.$$

Note that

$$\sum_{\mathsf{R}^* \in \{0,1\}^L} |\mathcal{H}_{i,j,\mathsf{R}^*}| \geq \left|\mathcal{H}_{i,j}^{\mathsf{tmp}}\right|.$$

This, together with the definition of expectation, implies that

$$\mathop{\mathbb{E}}_{\mathsf{R}^* \leftarrow \{0,1\}^L}[|\mathcal{H}_{i,j,\mathsf{R}^*}|] \geq \frac{|\mathcal{H}|_{i,j}^{\mathsf{tmp}}}{2^L}. \tag{3.17}$$

This, together with Claim 2.4, implies that

$$\mathbb{E}[|\mathcal{H}_{i,j}|] \geq \mathbb{E}\left[\frac{|\mathcal{H}_{i,j,\mathsf{R}^*}|}{2^{(N-1)L}}\right] = \frac{\mathbb{E}\left[|\mathcal{H}_{i,j,\mathsf{R}^*}|\right]}{2^{(N-1)L}} \geq \frac{\left|\mathcal{H}_{i,j}^{\mathsf{tmp}}\right|}{2^{NL}},$$

as desired, where the first equation follows from Claim 2.4, the second equation follows from the linearity of expectation, and the third equation follows from Equation (3.17).

■

■

■

3.2 Static Adversaries

We note that Theorem 3.1 holds also for static adversary. For completeness, we restate the theorem for static adversaries.

Theorem 3.6. *Fix any $m = m(n)$, $d = d(n)$, $L = L(n)$, and any n-party d-round public-coin protocol Π that outputs messages in $\{0,1\}^m$ and in which all parties send messages of length $L = L(n)$. Then, for any constant $\delta > 0$, any $t = t(n) < n$, and any $s = s(m)$, if Π is (t, δ, s)-statically secure then for every constant $c \in \mathbb{N}$ there exists a constant $b \in \mathbb{N}$ and an n-party d-round (t, δ', s)-statically secure public-coin protocol, that simulates Π, where all parties send messages of length $\ell = m \cdot \log^4(n \cdot d)$, and where $\delta' \leq \delta + \frac{b}{n^c}$.*

The proof is almost identical to the proof of Theorem 3.1 except that in the static setting, the adversary $\mathcal{A}$ needs to decide which t parties to corrupt before the protocol begins.

Recall that in the proof of Theorem 3.1, the adversary $\mathcal{A}$ simulates one of the adversaries $\mathcal{A}^H$. In the static setting, the adversary $\mathcal{A}$ will choose to corrupt the t parties that are consistent with as many $\mathcal{A}^H$ as possible. More specifically, recall that in the proof of Theorem 3.1 we defined $\mathcal{H}_1$ to be the set of all matrices H such that $\mathcal{A}^H$ tries to bias the outcome towards a specific set M. Recall that $|\mathcal{H}_1| \geq \frac{|\mathcal{H}|}{2^{2m}}$.

In the static setting, for every $H \in \mathcal{H}_1$ we denote by T^H the set of parties that the adversary $\mathcal{A}^H$ corrupts. For every set $T \subseteq [n]$ of size t let

$$\alpha(T) = \left|\{H \in \mathcal{H}_1 : T^H = T\}\right|.$$

We define

$$T^* = \operatorname*{argmax}_T \{\alpha(T)\},$$

and the adversary $\mathcal{A}$ corrupts the set of parties T^*. We define $\mathcal{H}'_1 \subseteq \mathcal{H}_1$ to consist of all the matrices $H \in \mathcal{H}_1$ for which $\mathcal{A}^H$ corrupts the set of parties T^*. Note that

$$|\mathcal{H}'_1| \geq \frac{|\mathcal{H}_1|}{2^n} \geq \frac{|\mathcal{H}|}{2^{2^m} \cdot 2^n}.$$

The rest of the proof is similar to that of Theorem 3.1, except that the analysis is easier in the static setting, since the decision of who to corrupt has already been made.

4 Public-Coin Protocols

In this section we show how to convert any distributed protocol in which parties do not have private inputs into a public-coin protocol.

Theorem 4.1. *Every protocol Π in which parties do not have private inputs can be transformed into a protocol Π' which simulates Π and such that the messages sent in Π' are uniformly random. Moreover, the protocol Π' preserves the security of Π and its round complexity.*

Proof Sketch. Let Π be an n-party protocol in which parties do not have private inputs. Let $d = d(n)$ be the number of communication rounds and let us assume for simplicity that each party speaks at each round. Assume, without loss of generality, that each party samples its own randomness ahead of time, when the protocol begins. That is, for every $j \in [n]$, party P_j has randomness $r_j \in \{0,1\}^\ell$, where we let ℓ be the maximum number of random bits used by all parties during the protocol. At each round i, party P_j evaluates a function $f_{i,j}$ which depends on the transcript of the protocol so far, which we denote by Trans_{i-1} (i.e., Trans_{i-1} are the messages sent by all parties in rounds $1, \ldots, i-1$), and on its own randomness r_j. Namely, the message sent at round $i \in [d]$ by party P_j is

$$m_{i,j} = f_{i,j}(\mathsf{Trans}_{i-1}, r_j).$$

Before we define the protocol Π', we introduce some notation. We say that a random string r is *good* with respect to transcript Trans_i and party P_j if when it is used as the randomness of that party, it generates the same exact transcript.

Next, we define the protocol Π'. In round $i \in [d]$, party P_j sends a uniformly random string $u_{i,j}$ of length 2^ℓ. Specifically, each party sends a uniformly random permutation of all possible ℓ-bit strings. At the end, after the d^{th} round ends, we interpret each $u_{i,j}$ as a collection of many possible random strings for party P_j, choose one (say the first), denoted by $r_{i,j}$, which is *good* with respect to the transcript so far and think of the $(i,j)^{\text{th}}$ message as $f_{i,j}(\mathsf{Trans}_{i-1}, r_{i,j})$.

First, we observe that the round complexity of Π' is the same as that of Π. Next, we claim that in an honest execution (i.e., in the absence of an adversary), the distribution of the output of the protocol Π is identical to that of Π' (namely, Π' simulates Π). We first note that conditioned on the fact that a *good* randomness was found for all $d \cdot n$ messages, the above distributions are the same. This is true since in Π' each party sends all possible ℓ bit strings in a *uniformly random* order. Second, we note that, since each party sends *all* possible ℓ-bit strings in each round, there *always* exists *good* randomness.

Next, we argue that the protocol Π' is as secure as Π. This follows by a simple hybrid argument. We define a sequence of protocols $\Pi^{(i)}$ for $i \in \{0, \ldots, dn\}$ in which until (and including) the i^{th} message, the parties act according to Π and in the rest of the protocol they act according to Π'. Notice that $\Pi' \equiv \Pi^{(0)}$ and $\Pi \equiv \Pi^{(dn)}$. We argue that for every $i \in [dn]$, the "advantage" of any $\mathcal{A}^{(i)}$ in $\Pi^{(i)}$ over any $\mathcal{A}^{(i-1)}$ in $\Pi^{(i-1)}$ is zero.

To this end, observe that the first $i-1$ messages are distributed exactly the same. In the next message (i.e., the i^{th} one) the protocols deviate. Assume party P_j speaks in both. While in $\Pi^{(i)}$ the message sent is some function of the transcript so far and the initial randomness P_j has, in $\Pi^{(i-1)}$ it is a random permutation of all possible random strings. We first note that if party P_j is corrupted, then both the adversary $\mathcal{A}^{(i)}$ and $\mathcal{A}^{(i-1)}$ can force any message in the name of P_j and thus they have the same power in both protocols (recall that after the i^{th} message, the protocols are identical). Hence, assume that P_j is not corrupted. In this case, the adversary $\mathcal{A}^{(i)}$ sees a message which is a function of the transcript up to that point and the (private) randomness of that party, whereas $\mathcal{A}^{(i-1)}$ sees a message which is a random permutation of all possible random strings. The theorem now follows by observing that one adversary can simulate the view of the other, and recalling that that the rest of the messages in both protocols are identically distributed. ■

Acknowledgments

We thank Nancy Lynch, Merav Parter and David Peleg for helpful remarks and pointers. The second author thanks his advisor Moni Naor for his continuous support.

References

[BL85] Michael Ben-Or and Nathan Linial. Collective coin flipping, robust voting schemes and minima of banzhaf values. In *26th Annual Symposium on Foundations of Computer Science, FOCS*, pages 408–416. IEEE Computer Society, 1985.

[Fei99] Uriel Feige. Noncryptographic selection protocols. In *40th Annual Symposium on Foundations of Computer Science, FOCS*, pages 142–153. IEEE Computer Society, 1999.

[GKP15] Shafi Goldwasser, Yael Tauman Kalai, and Sunoo Park. Adaptively secure coin-flipping, revisited. *IACR Cryptology ePrint Archive*, 2015:202, 2015. To appear in ICALP 2015.

[GS10] Oded Goldreich and Or Sheffet. On the randomness complexity of property testing. *Computational Complexity*, 19(1):99–133, 2010.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, 1997.

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is NOT needed: SINGLE database, computationally-private information retrieval. In *38th Annual Symposium on Foundations of Computer Science, FOCS*, pages 364–373. IEEE Computer Society, 1997.

[LLS89] David Lichtenstein, Nathan Linial, and Michael E. Saks. Some extremal problems arising form discrete control processes. *Combinatorica*, 9(3):269–287, 1989.

[New91] Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991.

[NN01] Moni Naor and Kobbi Nissim. Communication preserving protocols for secure function evaluation. In *33rd Annual ACM Symposium on Theory of Computing, STOC*, pages 590–599. ACM, 2001.

[Pel00] David Peleg. *Distributed Computing: A Locality-sensitive Approach.* Society for Industrial and Applied Mathematics, 2000.

[PR00] David Peleg and Vitaly Rubinovich. A near-tight lower bound on the time complexity of distributed minimum-weight spanning tree construction. *SIAM J. Comput.*, 30(5):1427–1442, 2000.

[RSZ02] Alexander Russell, Michael E. Saks, and David Zuckerman. Lower bounds for leader election and collective coin-flipping in the perfect information model. *SIAM J. Comput.*, 31(6):1645–1662, 2002.

[SMPU15] Atish Das Sarma, Anisur Rahaman Molla, Gopal Pandurangan, and Eli Upfal. Fast distributed pagerank computation. *Theor. Comput. Sci.*, 561:113–121, 2015.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de