Optimally-secure Coin-tossing against a Byzantine Adversary

Hamidreza Amini Khorasgani, Hemanta K. Maji, Mingyuan Wang
Department of Computer Science
Purdue University
West Lafayette, Indiana, USA – 47907
Email: {haminikh, hmaji, wang1929}@purdue.edu

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9518276

***Abstract*—Ben-Or and Linial (1985) introduced the full information model for coin-tossing protocols involving n processors with unbounded computational power using a common broadcast channel for all their communications. For most adversarial settings, the characterization of the exact or asymptotically optimal protocols remains open. Furthermore, even for the settings where near-optimal asymptotic constructions are known, the exact constants or poly-logarithmic multiplicative factors involved are not entirely well-understood.**

This work studies n-processor coin-tossing protocols where every processor broadcasts an arbitrary-length message once. An adaptive Byzantine adversary, based on the messages broadcast so far, can corrupt $k = 1$ processor. A bias-X coin-tossing protocol outputs 1 with probability X; otherwise, it outputs 0 with probability $(1 - X)$. A coin-tossing protocol's insecurity is the maximum change in the output distribution (in the statistical distance) that a Byzantine adversary can cause. Our objective is to identify bias-X coin-tossing protocols achieving near-optimal minimum insecurity for every $X \in [0, 1]$.

Lichtenstein, Linial, and Saks (1989) studied bias-X coin-tossing protocols in this adversarial model where each party broadcasts an independent and uniformly random bit. They proved that the elegant "threshold coin-tossing protocols" are optimal for all n and k. Furthermore, Goldwasser, Kalai, and Park (2015), Kalai, Komargodski, and Raz (2018), and Haitner and Karidi-Heller (2020) prove that $k = \mathcal{O}(\sqrt{n} \cdot \mathsf{polylog}\,(n))$ corruptions suffice to fix the output of any bias-X coin-tossing protocol. These results encompass parties who send arbitrary-length messages, and each processor has multiple turns to reveal its entire message.

We use an inductive approach to constructing coin-tossing protocols using a potential function as a proxy for measuring any bias-X coin-tossing protocol's susceptibility to attacks in our adversarial model. Our technique is inherently constructive and yields protocols that minimize the potential function. It is incidentally the case that the threshold protocols minimize the potential function, even for arbitrary-length messages. We demonstrate that these coin-tossing protocols' insecurity is a 2-approximation of the optimal protocol in our adversarial model. For any other $X \in [0, 1]$ that threshold protocols cannot realize, we prove that an appropriate (convex) combination of the threshold protocols is a 4-approximation of the optimal protocol. Finally, these results entail new (vertex) isoperimetric inequalities for density-X subsets of product spaces of arbitrary-size alphabets.

A full version of this paper is accessible at: https://eprint.iacr.org/2020/519.pdf

I. Introduction

Ben-Or and Linial [1], [2] introduced the full information model to study collective coin-tossing protocols in a seminal work. Collective coin-tossing protocols upgrade each of the n processors' local private randomness into shared randomness that all processors agree. In this model, all processors have an unbounded computational power and communicate over a broadcast channel. This model for the design and analysis of coin-tossing protocols has close connections with diverse mathematics and computer science topics like isoperimetric inequalities over product spaces (refer to the full version for details).

A *bias-X n-processor coin-tossing protocol* is an interactive protocol where every complete transcript is publicly associated with output 0 or 1, and the expected output for an honest execution of the protocol is $X \in [0, 1]$. Given a bias-X n-processor coin-tossing protocol π and a model for adversarial corruption and attack, let $\varepsilon^{+}(\pi) \in [0, 1]$ represent the maximum increase in the expected output an adversarial strategy can cause. Similarly, let $\varepsilon^{-}(\pi) \in [0, 1]$ represent the maximum decrease in the expected output caused by an adversarial strategy. One defines the insecurity of a protocol π as $\varepsilon(\pi) := \max\{\varepsilon^{+}(\pi), \varepsilon^{-}(\pi)\}$. For a fixed $X \in [0, 1]$, the *optimal bias-X n-processor protocol* minimizes $\varepsilon(\pi)$ among all bias-X n-processor coin-tossing protocols.

For practical applications, given the tolerance for insecurity, one needs precise guarantees on the insecurity of coin-tossing protocols to estimate the necessary number of processors to keep the insecurity acceptably low. If the insecurity estimates for the potential coin-tossing protocols involve large latent constants or poly-logarithmic factors, such a decision needs to be overly pessimistic in calculating the necessary number of processors. Consequently, it is essential to characterize coin-tossing protocols that are either optimal or within a small constant factor of the optimal protocol for every pair (n, X).

Hamidreza Amini Khorasgani, Hemanta K. Maji, and Mingyuan Wang are supported in part by an NSF CRII Award CNS–1566499, NSF SMALL Awards CNS–1618822 and CNS–2055605, the IARPA HECTOR project, MITRE Innovation Program Academic Cybersecurity Research Awards (2019–2020, 2020–2021), a Purdue Research Foundation (PRF) Award, and The Center for Science of Information, an NSF Science and Technology Center, Cooperative Agreement CCF–0939370.

Model. We study n-processor coin-tossing protocols where every processor broadcasts a message exactly once (i.e., *single-turn*), and there are n *rounds*. That is, in every round, a unique processor broadcasts her message in our model. Which processor speaks in which round and its message distribution may depend on the previous rounds' messages. We consider *adaptive Byzantine* adversaries who can corrupt $k = 1$ processor, i.e., based on the protocol's evolution, our adversary can corrupt one processor and fix her message arbitrarily.

Illustrative Example. A *threshold coin-tossing protocol* with threshold $t \in \{0, 1, \ldots, n+1\}$ proceeds as follows. Suppose $X \in [0, 1]$ be such that $X = 2^{-n} \cdot \sum_{i=t}^{n} \binom{n}{i}$. For $1 \leqslant i \leqslant n$, the processor-$i$ broadcasts an independent and uniformly random bit $C_i \in \{0, 1\}$. The output of this bias-X coin-tossing protocol is 1 if and only if $\sum_{i=1}^{n} C_i \geqslant t$. Observe that a Byzantine adversary can increase or decrease the expected output by $2^{-n}\binom{n-1}{t-1}$ by corrupting $k = 1$ processor. So, the insecurity of this bias-X threshold coin-tossing protocol is $\varepsilon = 2^{-n}\binom{n-1}{t-1}$.

In the restriction of our model, where each processor broadcasts an independent and uniformly random bit, these threshold protocols are the optimally-secure (even for $k > 1$) [3]. In our model (as well as the generalization where an adversary reveals its message over multiple turns), a Byzantine adversary can corrupt $k = \mathcal{O}(\sqrt{n} \cdot \mathsf{polylog}\,(n))$ processors to (nearly) fix the output [4]–[6]. Observe that in a threshold protocol, corrupting $k = \mathcal{O}(\sqrt{n})$ processors allows one to fix the output, and $k = \mathrm{o}(\sqrt{n})$ does not suffice. Consequently, the threshold protocols are *near-optimal* coin-tossing protocols against high corruption threshold k. These highly elegant threshold coin-tossing protocols are believed to be optimal or near-optimal in diverse adversarial corruption settings.

Summary of Our Results. For $k = 1$, we prove that the threshold coin-tossing protocols' insecurity is a 2-approximation to the optimal protocol's insecurity in our model. For $X \in [0, 1]$ that is not realizable by a threshold coin-tossing protocol, we prove that a convex combination of appropriate threshold coin-tossing protocols is a 4-approximation to the optimal coin-tossing protocol. These results imply new vertex isoperimetric inequalities for density-X subsets of product spaces of arbitrary-size alphabets.

A. Prior Related Works

Binary Alphabet. Lichtenstein, Linial, and Saks [3] consider the restriction where the i-th processor broadcasts an independent and uniformly random bit x_i, where $1 \leqslant i \leqslant n$, and the adversary can corrupt up to k processors, where $1 \leqslant k \leqslant n$. In this case, the coin-tossing protocol is a function $f\colon \{0, 1\}^n \to \{0, 1\}$. The underlying message space is $\{0, 1\}^n$, a product space of a *small-size alphabet*, and the probability distribution induced by the transcript is the uniform distribution over the message space. Note that, for n-processor coin-tossing protocols, such a protocol's bias can only be an integral multiple of 2^{-n}. Therefore, this is a discrete optimization problem.

Given n, k, and X, they begin with the objective of minimizing $\varepsilon^+(\pi)$ over bias-X n-processor coin-tossing protocols π. Their characterization of the protocol minimizing ε^+, *incidentally*, turns out to be identical to the optimal solution for the *vertex isoperimetric inequality* over the Boolean hypercube [7]–[9]. Therefore, a threshold protocol[1] π is the optimal protocol and minimizes ε^+. The complementary protocol, which swaps the outputs 0 and 1 of π, is also a threshold protocol and, consequently, minimizes ε^-. So, threshold protocols simultaneously minimize ε^+ and ε^- and achieve optimal security. As is evident, this analysis crucially relies on the message space being the Boolean hypercube.

Significantly altering the output distribution. For symmetric functions (i.e., permuting the inputs of the function f does not change its output), Goldwasser, Kalai, and Park [4] prove that $k = \mathcal{O}(\sqrt{n} \cdot \mathsf{polylog}\,(n))$ corruptions suffice to completely fix the output of any coin-tossing protocol even if the protocol has *arbitrary-length messages*. After that, Kalai, Komargodski, and Raz [5] remove the restriction of symmetric functions. Recently, in independent work,[2] Haitner and Karidi-Heller [6] extend this result to *multi-turn* coin-tossing protocols in a ground-breaking result. These papers use global analysis techniques for martingales that are inherently non-constructive; consequently, they prove the optimality of threshold protocols up to $\mathcal{O}(\mathsf{polylog}\,(n))$ factors (i.e., the adversary corrupts at most $k = \mathcal{O}(\sqrt{n} \cdot \mathsf{polylog}\,(n))$ processors).

B. Challenge for Arbitrary-Length Messages

Our objective is to provide tight insecurity estimates for the optimal coin-tossing protocols that use *arbitrary-length messages*. Let us understand why the technical approach of [3] fails, and an entirely new approach is needed.

In the full information model, without the loss of generality, one can assume that all interactive protocols are stateless, and processors use a new block of private randomness to generate the next message at any point during the evolution of the coin-tossing protocol [10], [11]. Furthermore, the security of processors' internal state is not a concern, so, without loss of generality, every processor broadcasts its appropriate block of randomness whenever it speaks.[3] A Byzantine adversary can corrupt a processor and arbitrarily set its randomness. So, for an appropriately large alphabet Σ, which depends on the randomness complexity of generating each message, our ambient message space is Σ^n, a product space involving a large alphabet set.

However, over such product spaces, the objective of minimizing ε^+ in isolation does not entail the simultaneous minimization of ε^-. Given any $n \in \mathbb{N}$ and $X \in [0, 1]$, there exists a protocol with $(\varepsilon^+, \varepsilon^-) = (\frac{1-X}{n}, X)$. Let $\Sigma = \{0, 1, \ldots, \sigma-1\}$

[1] More generally, protocols that output 1 for all strings smaller in the *simplicial order* [9] than a threshold string are the optimal protocols.

[2] The full version of our paper appeared online on eprint on 05 May, 2020.

[3] Let π be the original coin-tossing protocol. In the compiled π', suppose parties reveal the block of randomness that they use to prepare their next-message in the protocol π. The new protocol π', first, emulates the next-message function of π to generate the entire transcript, and, then, uses π to determine the output.

such that $(1-1/\sigma)^n = X$, i.e., $\sigma = \Theta(n/\log(1/X))$. Every processor broadcasts an independent and uniformly random message from Σ. The outcome of the coin-tossing protocol is 0 if some processor broadcasts 0; otherwise, the output is 1. Observe that a Byzantine adversary can corrupt any single processor to force output 0. However, corrupting one processor can only increase the expected output by (roughly) $(1-X)/n$. Similarly, there is a protocol with $(\varepsilon^+, \varepsilon^-) = (1-X, \frac{X}{n})$. More generally, for product spaces over large alphabets, one does not expect such a well-behaved isoperimetric inequality [12], [13].

Finally, global analysis techniques of [4]–[6] analyze the case of multiple corruptions. The optimal protocol for $k = 1$ is not apriori related to the optimal protocols robust to multiple corruptions. Furthermore, the inductive proof technique of Aspnes [14], [15] is agnostic of the expected output of the coin-tossing protocol. Consequently, reconstructing the optimal protocol from the lower-bound on insecurity is not apparent.

Therefore, we follow the geometric technique of Khorasgani, Maji, and Mukherjee [16], which is inherently constructive, to obtain tight estimates of the optimally secure protocols. Ideally, one should find the coin-tossing protocol that minimizes $\varepsilon = \max\{\varepsilon^+, \varepsilon^-\}$; however, this function does not behave well. So, we consider the proxy objective of minimizing $\widetilde{\varepsilon} = \varepsilon^+ + \varepsilon^-$, instead. We characterize the optimal protocol for this proxy objective, which is a 2-approximation to the actual objective.

C. Connection to Isoperimetric Inequalities

The connection to isoperimetric inequalities [7]–[9], [13] (via the expansion of fixed density subset of product spaces) establishes relevance to theoretical computer science topics like expander graphs, complexity theory, and error-correcting codes.

Encoding Security of Coin-tossing Protocols. Every coin-tossing protocol is equivalent to a subset S of an n-dimension product space Σ^n, where the size of the alphabet set $\sigma := |\Sigma|$ depends on the randomness complexity of the coin-tossing protocol. Likewise, for every subset $S \subseteq \Sigma^n$, one can define a coin-tossing protocol corresponding to this subset S. The elements of this product space represent the complete transcript of the coin-tossing protocol. The i-th coordinate of an element corresponds to the message sent by processor i, and the subset S contains all elements of the product space on which the coin-tossing protocol outputs 1. One considers the uniform distribution over Σ^n to sample the elements. This subsection considers a *stronger* Byzantine adversary who can edit one processor's message after seeing the messages of all processors.

The discussion in this subsection extends to arbitrary corruption threshold k. However, for the simplicity of the presentation, we consider the specific case of $k = 1$. Let ∂S_k^+ be the set of elements in $\overline{S}$ (the complement of S) that are at a Hamming distance $k = 1$ from the set S. Consequently, the strong Byzantine adversary can change an element from the set $\partial S_k^+ \subseteq \overline{S}$ into some element of S by editing (at most) k coordinates. If the stronger Byzantine adversary can see *all* the messages and then perform the edits, it can increase the expected output by exactly $\varepsilon^+ = \left|\partial S_k^+\right|/\sigma^n$.

Analogously, one defines the set $\partial S_k^- \subseteq S$ that contains all elements at a Hamming distance $k = 1$ from the set $\overline{S}$. So, a stronger Byzantine adversary can reduce the expected output by $\varepsilon^- = \left|\partial S_k^-\right|/\sigma^n$.

Extremal Graph Theory Perspective. The (width-k) *vertex perimeter* of the set S, represented by $\partial_{V,k} S$, is the set of all elements in $\overline{S}$ that are at a Hamming distance of at most k from some element in S. Observe that the perimeter $\partial_{V,k} S$ is identical to the set ∂S_k^+. Similarly, the vertex perimeter of the set $\overline{S}$ (which is $\partial_{V,k}\overline{S}$) is identical to the set ∂S_k^-.

Extremal graph theory aims to characterize the optimal set S of a density-X that minimizes its vertex perimeter. This optimal set S, in turn, characterizes the bias-X coin-tossing protocol with the minimum ε^+. In Appendix A of the full version, we see that minimizing ε^+ does not automatically entail the simultaneous minimization of ε^- for general Σ.[4] That example highlights that the protocol minimizing ε^+ resulted in a protocol where the stronger Byzantine adversary can force the outcome 0 with certainty. Therefore, there is a disconnect between the cryptographic objective of minimizing $\varepsilon = \max\{\varepsilon^+, \varepsilon^-\}$ with the typical objective in extremal graph theory for large alphabet set Σ.

Cryptography-inspired Extremal Graph Theory. Instead of minimizing the vertex perimeter of a density-X set S, one should consider the alternative objective of minimizing the *symmetric perimeter* of S defined under various norms.

$$\partial^{\mathsf{sym}}_{V,k,\ell}(S) := \left(\left|\partial_{V,k} S\right|^\ell + \left|\partial_{V,k}\overline{S}\right|^\ell\right)^{1/\ell}.$$

The $\ell = \infty$ case corresponds to our cryptographic objective; however, this norm is difficult to analyze. Consequently, we study the norm $\ell = 1$ as a proxy, which is a 2-approximation of the norm $\ell = \infty$. Our results provide evidence that such symmetric perimeters may be more well-behaved in general.

Recall that, in our setting, the element in Σ^n is exposed one coordinate at a time, and our Byzantine adversaries cannot go back to edit previously exposed coordinates. So, our Byzantine adversaries have lesser power than the stronger Byzantine adversaries considered in this section. Consequently, the *minimum achievable insecurity* for bias-X n-processor coin-tossing protocols in our setting *lower-bounds* the proxy norm above. For instance, when $\ell = 1$, our results imply that the symmetric perimeter's density is $1/\sqrt{n}$ for any dense set S, irrespective of the size of the alphabet set.

Remark. We identify a density-X set with its corresponding bias-X coin-tossing protocol. Using the independent bounded differences inequality for the Hamming distance function (using Azuma's inequality [17]) on a constant-density subset S implies that $k = \mathcal{O}(\sqrt{n})$ edits suffice to achieve any constant

[4]For $\Sigma = \{0, 1\}$, this entailment holds; otherwise, it is not known to hold in general.

ε^+ and ε^-, for any σ.[5] However, for small k (for example, $k = 1$), obtaining meaningful guarantees on both ε^+ and ε^- is not possible for large σ. On the other hand, interestingly, our results entail that $\max\{\varepsilon^+, \varepsilon^-\} \geqslant 1/\sqrt{n}$ for any σ. This result lends support to the hypothesis that the symmetric perimeter is more well-behaved.

II. Our Contributions

Let us set up the minimal notations to state our results. For every partial transcript v, the *color of* v, represented by x_v, is the expected output of the coin-tossing protocol conditioned on the partial transcript being v. For example, a complete transcript has color $\in \{0, 1\}$, and the color of the empty partial transcript (i.e., the protocol has not started yet) of a bias-X coin-tossing protocol is X. The probability p_v represents the probability that the protocol evolution of π generates the partial transcript v.

Let τ be an attack strategy for a Byzantine adversary who may corrupt $k = 1$ processor. Equivalently, τ can be represented as a collection of pairs of partial transcripts, namely, $\{(u_i, v_i)\}_{i=1}^{\ell}$. That is, when a partial transcript u_j (for instance, it consists of the messages from processors $1, 2, \ldots, j$) happens, the Byzantine adversary shall decide to corrupt the next processor $j + 1$ and fix the next partial transcript to be v_i. The score of the attack strategy τ on the protocol π is

$$\mathsf{Score}\,(\pi, \tau) \;:=\; \sum_{i=1}^{\ell} p_{u_i} \cdot |x_{u_i} - x_{v_i}|.$$

The term $\mathsf{Score}\,(\pi, \tau)$ represents the vulnerability of protocol π under attack strategy τ. Furthermore, we define

$$\mathsf{Score}\,(\pi) \;:=\; \sup_{\tau}\; \mathsf{Score}\,(\pi, \tau)\,.$$

Intuitively, $\mathsf{Score}\,(\pi)$ represents the insecurity of the protocol under the most devastating attack, a.k.a., our potential function.

We emphasize that our score is not identical to the deviation in output distribution that a Byzantine adversary causes. It is a 2-approximation of that quantity. Define the insecurity as the maximum change that a Byzantine adversary can cause to the output distribution. Then, it is evident that the insecurity of π is at least $\mathsf{Score}(\pi)/2$, because $\max\{\varepsilon^+, \varepsilon^-\} \geqslant (\varepsilon^+ + \varepsilon^-)/2$.

For an arbitrary $n \in \mathbb{N}^*$ and $t \in \{0, 1, \ldots, n + 1\}$, let $\pi^{n,t}$ denote the n-processor t-threshold *threshold protocol*. In this threshold protocol, every processor broadcasts an independent and uniformly random bit. The output of this threshold protocol is 1 if and only if the total number of ones in the complete transcript is $\geqslant t$. An n-processor t-threshold protocol has color $2^{-n} \cdot \left(\sum_{i=t}^{n} \binom{n}{i}\right)$.

We prove the following theorem about the threshold protocol.

[5]Even computationally efficient attacks are known to achieve this bound [18], [19].

Theorem 1: For any bias-X n-processor protocol π^*, where $X = 2^{-n} \cdot \left(\sum_{i=t}^{n} \binom{n}{i}\right)$ and $0 \leqslant t \leqslant n + 1$, then

$$\mathsf{Score}(\pi^{n,t}) \leqslant \mathsf{Score}(\pi^*).$$

That is, the threshold protocol is the protocol that minimizes the score. Equivalently, the insecurity of the threshold protocol is a 2-approximation of the optimal insecurity in our corruption model (refer to Corollary 1 of the full version).

Furthermore, for a root color X that does not correspond to a threshold protocol, we prove the following result.

Theorem 2: Suppose X is a root-color that does not admit a threshold protocol, and X is inverse-polynomially far from both 0 and 1. Suppose X is intermediate to the bias of the threshold protocols $\pi^{n-1,t}$ and $\pi^{n-1,t-1}$. Let π be a protocol where the first processor decides to run the threshold protocol $\pi^{n-1,t}$ or $\pi^{n-1,t-1}$ with suitable probability so that the resulting protocol is a bias-X protocol. Then, we have, for any bias-X and n-processor coin-tossing protocol π^*,

$$\mathsf{Score}(\pi) \leqslant 2 \cdot \mathsf{Score}(\pi^*).$$

That is, the insecurity of this protocol π is a 4-approximation of the protocols with minimum insecurity against Byzantine adversaries (refer to Corollary 2 of the full version).

Finally, our results imply isoperimetric inequalities for the symmetric perimeter.

Corollary 3: For any alphabet Σ and dimension $n \in \mathbb{N}$, let $S \subseteq \Sigma^n$ be a density-X subset of Σ^n. Suppose $X \in 2^{-n} \cdot \left(\sum_{i=t+1}^{n-1} \binom{n-1}{i}, \sum_{i=t}^{n-1} \binom{n-1}{i}\right)$, for some t. Then,

$$\max\{\partial S_1^+, \partial S_1^-\} \geqslant \frac{1}{4} \cdot 2^{-n} \binom{n-2}{t}.$$

In particular, when X is a constant, it implies that

$$\max\{\partial S_1^+, \partial S_1^-\} \geqslant \Theta\big(1/\sqrt{n}\big).$$

III. Technical Overview

The techniques closest to our approach are those introduced by Aspnes [14], [15] and Khorasgani et al. [16], [20], [21].

Aspnes' technique [14], [15] tracks the locus of all possible $(\varepsilon^+, \varepsilon^-)$ corresponding to any n-processor k-corruption threshold protocol. However, the information regarding the root-color is lost and, consequently, the technique does not yield the optimal protocol construction. Next, one lower-bounds this space using easy-to-interpret (hyperbolic) curves and obtains bounds on the insecurity of any n-processor protocol with k corruption threshold (against adversaries who erase the messages of processors).

The technique of Khorasgani et al. [16], [20], [21] uses a potential function as a proxy to study the actual problem at hand. They maintain the locus of all n-processor bias-X protocols that minimize the potential function. Next, they inductively build the next curve of $(n+1)$-processors bias-X protocols that minimize the potential function. Their approach outrightly yields the optimal constructions that minimize the potential function, and easily handle the case of processors sending *arbitrary-length* messages.

High-level summary of our approach. We use the potential function as introduced in Section II, which is a 2-approximation of the optimal insecurity against Byzantine adversaries, for any n-processor bias-X protocol. Let $C_n(X)$ represent the minimum realizable potential for bias-X n-processor coin-tossing protocols.

Next, we prove that if an n-processor threshold protocol has potential δ and bias-X, then the point (δ, X) lies on the optimal curve $C_n(X)$. Therefore, the potential of these threshold protocols are 2-approximation of the optimal bias-X protocol against Byzantine adversaries.

After that, inductively, we prove that the linear interpolation, denoted by L_n, of the set of points (δ, X) realized by n-processor threshold protocols with potential δ and root-color X, where $0 \leqslant t \leqslant n+1$, is a lower-bound to the actual curve $C_n(X)$. Finally, we argue that a linear interpolation of appropriate threshold functions yields a protocol with potential that is 4-approximation of the optimal protocol against Byzantine adversaries.

The curves and the inductive transformation. Consider the case of $n = 1$ and arbitrary bias-X. If $X = 0$ or $X = 1$, then we have $C_1(X) = 0$. If $X \in (0, 1/2]$, then we include that edge that sets the output to 1. This observation creates a potential of $C_1(X) = 1 - X$. Similarly, we have $C_1(X) = X$, for all $X \in [1/2, 1)$. Our characterization of the curve $C_1(X)$ is complete (refer to Figure 1).

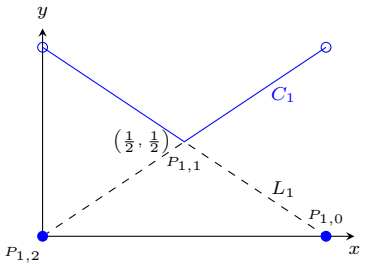

Fig. 1. The (black) dashed curve is L_1 and the (blue) solid curve is C_1. Note that $P_{1,t}$ corresponds to the point defined by $(1, t)$-threshold protocol.

Next, consider the case of $n = 2$ and bias-X. This case is sufficient to understand how to inductively build the locus of the curve $C_{n+1}(X)$ inductive from $C_n(X)$. Consider any arbitrary 2-processor bias-X coin-tossing protocol. Suppose the first processor sends message $1, 2, \ldots, \ell$. Let x_i, for $1 \leqslant i \leqslant \ell$, be the expected output conditioned on the first message being i. At the root of this protocol, we have two options. Corrupt processor one and send the message that achieves the highest potential. Or, defer the intervention to a later point in time.

Corrupting the root of this protocol causes the potential to become $\max_{i=1}^{\ell} |X - x_i|$. Deferring the intervention to a later point in time results in the potential becoming at least $\sum_{i=1}^{\ell} p_i \cdot C_1(x_i)$, where p_i is the probability that processor 1 outputs i. The actual potential of π is the maximum of these two quantities. Our objective is to characterize the choice of $x_1, \ldots, x_\ell$ such that the potential is minimized (refer to Figure 2).

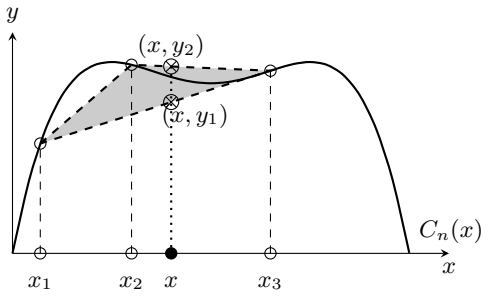

Fig. 2. An intuitive example of the geometric transformation

The solution to this optimization problem motivates the definition of our geometric transformation,

$$T(C)(x) := \inf_{\substack{x_1,\ldots,x_\ell \in [0,1] \\ p_1,\ldots,p_\ell \in [0,1] \\ p_1+\cdots+p_\ell = 1 \\ p_1 x_1 + \cdots + p_\ell x_\ell = x}} \max\left(|x - x_1|, \ldots, |x - x_\ell|, \sum_{i=1}^{\ell} p_i \cdot C(x_i) \right).$$

Towards proving our main theorem, at a high-level, we inductively prove that, if L_n is a lower-bound of the curve C_n, then $L_{n+1} := T(L_n)$ and is a lower-bound of the next curve $C_{n+1} = T(C_n)$. We refer the interested readers to the full version for more technical details.

References

[1] M. Ben-Or and N. Linial, "Collective coin flipping, robust voting schemes and minima of banzhaf values," in *26th FOCS*. IEEE Computer Society Press, Oct. 1985, pp. 408–416. 1

[2] M. Ben-Or and N. Linial, "Collective coin flipping," *Advances in Computing Research*, 1989. 1

[3] D. Lichtenstein, N. Linial, and M. Saks, "Some extremal problems arising from discrete control processes," *Combinatorica*, 1989. 2

[4] S. Goldwasser, Y. T. Kalai, and S. Park, "Adaptively secure coin-flipping, revisited," in *ICALP 2015, Part II*, ser. LNCS, M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, Eds., vol. 9135. Springer, Heidelberg, Jul. 2015, pp. 663–674. 2, 3

[5] Y. T. Kalai, I. Komargodski, and R. Raz, "A lower bound for adaptively-secure collective coin-flipping protocols," in *DISC*, 2018. 2, 3

[6] I. Haitner and Y. Karidi-Heller, "A tight lower bound on adaptively secure full-information coin flip," *FOCS*, 2020. 2, 3

[7] J. B. Kruskal, "The number of simplices in a complex," *Mathematical optimization techniques*, 1963. 2, 3

[8] G. Katona, "A theorem for finite sets, theory of graphs (p. erdös and g. katona, eds.)," 1968. 2, 3

[9] L. H. Harper, "Optimal numberings and isoperimetric problems on graphs," *Journal of Combinatorial Theory*, 1966. 2, 3

[10] M. Jerrum, "Random generation of combinatorial structures from a uniform distribution (extended abstract)," in *ICALP*, 1985. 2

[11] M. Bellare, O. Goldreich, and E. Petrank, "Uniform generation of np-witnesses using an np-oracle," *Inf. Comput.*, 2000. 2

[12] Y. Filmus, L. Hambardzumyan, H. Hatami, P. Hatami, and D. Zuckerman, "Biasing Boolean functions and collective coin-flipping protocols over arbitrary product distributions," in *ICALP 2019*, ser. LIPIcs, C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, Eds., vol. 132. Schloss Dagstuhl, Jul. 2019, pp. 58:1–58:13. 3

[13] L. H. Harper, "On an isoperimetric problem for hamming graphs," *Discret. Appl. Math.*, 1999. 3

[14] J. Aspnes, "Lower bounds for distributed coin-flipping and randomized consensus," in *29th ACM STOC*. ACM Press, May 1997, pp. 559–568. 3, 4

[15] James Aspnes, "Lower bounds for distributed coin-flipping and randomized consensus," *J. ACM*, 1998. 3, 4

[16] H. A. Khorasgani, H. K. Maji, and T. Mukherjee, "Estimating gaps in martingales and applications to coin-tossing: Constructions and hardness," in *TCC 2019, Part II*, ser. LNCS, D. Hofheinz and A. Rosen, Eds., vol. 11892. Springer, Heidelberg, Dec. 2019, pp. 333–355. 3, 4

[17] K. Azuma, "Weighted sums of certain dependent random variables," 1967. 3

[18] S. Mahloujifar and M. Mahmoody, "Can adversarially robust learning leveragecomputational hardness?" in *Algorithmic Learning Theory, ALT*, 2019. 4

[19] O. Etesami, S. Mahloujifar, and M. Mahmoody, "Computational concentration of measure: Optimal bounds, reductions, and more," in *SODA*, 2020. 4

[20] H. A. Khorasgani, H. K. Maji, and M. Wang, "Coin tossing with lazy defense: Hardness of computation results," *IACR Cryptol. ePrint Arch.*, 2020. 4

[21] H. K. Maji and M. Wang, "Black-box use of one-way functions is useless for optimal fair coin-tossing," Cryptology ePrint Archive, Report 2020/253, 2020, https://eprint.iacr.org/2020/253. 4