Polynomial-time targeted attacks on coin-tossing for any number of corruptions

Omid Etesami[1], Ji Gao[2], Saeed Mahloujifar[3], and Mohammad Mahmoody[2]

[1] School of Mathematics, IPM, P.O. Box 19395-5746, Tehran, Iran. etesami@gmail.com
[2] University of Virginia, Charlottesville, VA, USA. {jg6yd,mohammad}@virginia.edu
[3] Princeton University, Princeton, NJ, USA. sfar@princeton.edu

Abstract. Consider a coin tossing protocol in which n processors $P_1, \ldots, P_n$ agree on a random bit b in n rounds, where in round i P_i sends a single message w_i. Imagine a full-information adversary who prefers the output 1, and in every round i it knows all the finalized messages $w_1, \ldots, w_{i-1}$ so far as well as the prepared message w_i. A k-replacing attack will have a chance to replace the prepared w_i with its own choice $w'_i \neq w_i$ in up to k rounds. Taking majority protocol over uniformly random bits $w_i = b_i$ is robust in the following strong sense. Any k-replacing adversary can only increase the probability of outputting 1 by *at most* $O(k/\sqrt{n})$. In this work, we ask if the above simple protocol is tight. For the same setting, but restricted to uniformly random bit messages, Lichtenstein, Linial, and Saks [Combinatorica'89] showed how to achieve bias $\Omega(k/\sqrt{n})$ for any $k \in [n]$. Kalai, Komargodski, and Raz [DISC'18, Combinatorica'21] gave an alternative *polynomial-time* attack when $k \geq \Theta(\sqrt{n})$. Etesami, Mahloujifar, and Mahmoody [ALT'19, SODA'20] extended the result of KKR18 to *arbitrary* long messages. It hence remained open to find *any* attacks of bias $\Omega(k/\sqrt{n})$ in the few-corruption regime $k = o(\sqrt{n})$ when the messages are of arbitrary length, and to find such *polynomial-time* (and perhaps tight) attacks when messages are uniformly random bits. In this work, we resolve both of these problems.

- For arbitrary length messages, we show that k-replacing polynomial-time attacks can indeed increase the probability of outputting 1 by $\Omega(k/\sqrt{n})$ for *any* k, which is optimal up to a constant factor. By plugging in our attack into the framework of Mahloujifar Mahmoody [TCC'17] we obtain similar data poisoning attacks against deterministic learners when adversary is limited to changing $k = o(\sqrt{n})$ of the n training examples.
- For uniformly random bits $b_1, \ldots, b_n$, we show that whenever $\Pr[b = 1] = \Pr[\sum b_i \geq t] = \beta_n^{(t)}$ for $t \in [n]$ is the probability of a Hamming ball, then online *polynomial-time* k-replacing attacks can increase $\Pr[b = 1]$ from $\beta_n^{(t)}$ to $\beta_n^{(t-k)}$, which is optimal due to the majority protocol. In comparison, the (information-theoretic) attack of LLS89 increased $\Pr[b = 1]$ to $\beta_{n-k}^{(t-k)}$, which is optimal for adaptive adversaries who *cannot* see the message before changing it. Thus, we obtain a computational variant of Harper's celebrated vertex isoperimetric inequality.

Keywords: Coin tossing protocols, isoperimetric inequalities, poisoning attacks

1 Introduction

Collective coin tossing [6] is a fundamental problem in cryptography in which a set of n parties aim to jointly produce a random bit b that remains (close to) random even if an adversary controls a subset of these parties. The simple majority protocol $\mathrm{maj}(b_1, \ldots, b_n)$, when n is odd and each bit b_i is broadcast by party P_i, is robust in the following strong sense: Any adversary who even gets to see *all* the messages and then replaces at most $k \in [n]$ of the them can only bias the output bit by at most by $O(k/\sqrt{n})$ [4]. In a nutshell, in this work we ask *how optimal is the majority protocol against such attacks?* We study this question from various angels as explained below.

Problem setting. Suppose Π is an n-round coin-tossing protocol between n parties, where party P_i sends a single message w_i in round i that could depend on all the previous messages, and the final bit b is a deterministic function of all messages.[4] Now, suppose an adversary aims to increase the probability of $\Pr[b = 1]$. We call this a *targeted* attack, as adversary can choose the target direction of the bias.[5] We deal with k-*replacing* adversaries who can replace k of the messages as follows. Suppose messages $w_1, \ldots, w_{i-1}$ are already finalized and party P_i is about to send w_i in round i. The adversary will have a chance to replace w_i, based on the knowledge of w_i.[6] Equivalently, we will think of the protocol as a *random process* $(w_1, \ldots, w_n)$ with n steps, and a k-replacing adversary will be allowed to override the content of k of the steps, in which case the rest of the random process will depend on the new values. The goal of the adversary is to increase the probability of $\Pr[b = 1]$ for a Boolean function $f(w_1, \ldots, w_n) = b \in \{0, 1\}$. Informally speaking, we would like to know what are the most robust random processes in this setting.

Targeted aspect. Studying targeted attacks is important due to several reasons. Firstly, targeted attacks allow modeling adversaries who have a particular output preferred in mind. For example, the coin tossing model's output might determine whether a contract would be signed or not. Then, a party who prefers signing the contract wants to increase the chance of outputting $b = 1$. Moreover, targeted attacks allow modeling attacks on specific "undesired" properties like $\mathcal{B}$ defined over random processes; namely, the adversary aims to increase the probability of B happening at the end. Below in the introduction we further discuss applications such as targeted poisoning attacks in adversarial machine learning and computational isoperimetry results. See the full version of this paper for formalization of these results.

Robustness of threshold functions. For a setting where w_i is a uniform random bit b_i, consider the threshold function f defined as $f(b_1, \ldots, b_n) = 1$ whenever $\sum b_i \geq t$ and let $\beta_n^{(t)} = \Pr[\sum b_i \geq t]$. Then we get a robust protocol in the following sense. Any k-replacing adversary will be limited to achieve $\Pr[b = 1] \leq \beta_n^{(t-k)}$, because all it can do is to replace k ones with zeros. In particular, it can be shown that for the majority function (for odd n) any k-replacing attack increase $\Pr[b = 1]$ by at most $O(k/\sqrt{n})$.

[4] This is also called a *single-turn* protocol.

[5] In contrast, *untargeted* adversaries can bias the output towards *either* of 0 or 1.

[6] This is also called the *strongly adaptive* corruption model [13].

In this work, we study the optimality of the simple threshold/majority protocols and ask the following.

1. If $\Pr[b=1]=1/2$ holds originally, can k-replacing adversaries increase the probability of $\Pr[b=1]$ by $\Omega(k/\sqrt{n})$ in *every* n-step random process with arbitrarily long messages?
2. For simpler models such as those with uniformly random bits, can we obtain *optimal* attacks that prove the threshold protocols to be the best possible for all $\Pr[b=1]=\beta_n^{(t)}$?

We answer both questions above affirmatively. Notably, we even obtain *polynomial-time* attacks. Before describing our results in details, we briefly discuss what was known before our work.

Previous work for uniform binary messages. Lichtenstein, Linial, and Saks [24] showed that the threshold protocols are optimal when the messages are uniform random bits, but under a weaker attack model where the adversary is supposed to corrupt parties before seeing their message. In particular, they showed that if $\Pr[f(b_1,\ldots,b_n)=1]$ without attack is the probability of the threshold function $\Pr[\sum b_i \geq t]=\beta_n^{(t)}$, then there is an *adaptive* attack with budget k that achieves $\Pr[f(b_1,\ldots,b_n)=1] \geq \beta_{n-k}^{(t-k)}$. However, this attack was information theoretic and not polynomial time. It also remained open whether k-replacing attacks can improve upon the bound of [24] and potentially match the robustness of threshold functions. In other words, prior to our work, it was not known whether threshold functions are optimal against k-replacing attacks.

Previous work on arbitrary length messages. Kalai, Komargodski, and Raz [21] showed that in the "many-replacement" regime where $k=\Omega(\sqrt{n})$, a different attack in the binary setting of [24] can be achieved in polynomial time.[7] Building upon [21], Etesami, Mahloujifar and Mahmoody [12,28] showed how to extend this result to arbitrary message length and obtain (again targeted) attacks in polynomial time, but again only when $k \geq \Omega(\sqrt{n})$. (See Section 1.1 for more discussions on why those proofs lead to many replacements.) Finally, Khorasgani, Maji, Mukherjee, and Wang [22,23] showed how to get *non-targeted* attacks for large messages when $k=1$.

Our results. Previous works left open our two main questions. In this work, we resolve both of these questions and show that (1) when $\Pr[b=1]=\Theta(1)$, then majority is optimal up to a constant factor against k-replacing adversaries for all adversary budget k (including the "few corruption regime"), and (2) when messages are uniformly random bits, for any initial probability of Hamming balls $\Pr[b=1]=\Pr[\sum b_i \geq t]$, the corresponding threshold function is optimal, *even up to exact constants*.

Theorem 1 (Main result 1 – arbitrary messages)**.** *Let Π be any single-turn polynomial-time coin-tossing protocol between n parties to obtain an output bit b in which, originally (before any attack) it holds that $\Pr[b=1]=\mu$. For any $k=O(\sqrt{n})$, there is a*

[7] Interestingly, the main result of [21] focuses on *non-targeted* attacks and shows that the output of any single-turn protocol can be attacked (only information theoretically) by a (standard) adaptive *non-targeted* adversary replacing $k=\Omega(\sqrt{n})$ parties. The recent breakthrough of Haitner and Karidi-Heller [15] generalized the main result of [21] to any general, perhaps multi-turn, protocol. Our focus in this work, however, is on single-turn protocols.

k-replacing polynomial time attack that increases the probability of outputting $b = 1$ by a probability that can get arbitrarily close to:

$$\left(1-\left(1-\frac{\mu}{\sqrt{n}}\right)^k\right)\cdot\left(1-e^{-2}-\mu\right).$$

To prove Theorem 1, we use ideas from the attack of [28] (see Section 1.1). See Theorems 14 for a formalization of the information theoretic variant. For the polynomial-time variant of this theorem see the full version.

It can be shown that, as long as $k = O(\sqrt{n})$, the biasing bound of Theorem 1 is $\Omega(k \cdot \mu/\sqrt{n})$. Therefore, Theorem 1 resolves our first main question above; i.e., the majority protocol of [3] is optimal, up to a constant factor, for targeted attacks on any single-turn protocol when $\mu = \Theta(1)$.

Our next result solves the problem completely for protocols with uniform random bits, as long as the probability of outputting 1 is that of a threshold function.

Theorem 2 (Main result 2 – uniformly random bits)**.** *Let Π be any single-turn polynomial-time coin-tossing protocol between n parties to obtain an output bit b in which the parties share uniformly random bits $b_1, \dots, b_n$. Suppose originally (before any attack) it holds that $\Pr[b = 1] = \Pr[\sum b_i \geq t] = \beta_n^{(t)}$ for $t \in [n]$. Then, for any $k \in [n]$, there is a k-replacing attack that increases the probability of outputting $b = 1$ to $\beta_n^{(t-k)}$. Moreover, if it further holds that $\Pr[b = 1] \geq 1/\operatorname{poly}(n)$ is non-negligible, then there will be* polynomial-time *k-replacing attacks that can get arbitrarily close to the same bound of $\beta_n^{(t-k)}$.*

To prove Theorem 2, we also use ideas from the recent work of [23]. See Theorem 18 for a formal version of the information theoretic variant of Theorem 2. See the full version of the paper for how our specific information theoretic attack can be adapted minimally to run in polynomial time.

Note that Theorem 2 shows something perhaps surprising about the power of *online* attacks against coin tossing protocols. It shows that online attacks are *as powerful* as offline attacks, when we consider the most robust functions with $\Pr[b = 1] = \beta_n^{(t)}$ being that of a Hamming ball. In fact, we present such attacks that run in polynomial time, and this implies a new computational variant for the celebrated vertex isoperimetry inequality of Harper [19]. Indeed, the vertex isoperimetric inequality in the Boolean hypercube states that for any set $\mathcal{S} \subseteq \{0,1\}^n$ of probability $\Pr[(b_1 \dots, b_n) \in \mathcal{S}] = \beta_n^{(t)}$, the probability of the set of points (inside or outside $\mathcal{S}$) with a neighbor in $\mathcal{S}$ of distance at most k is at least $\beta_n^{(t-k)}$. Our Theorem 2 matches this bound exactly, and even shows how to *find* such close neighbors (in $\mathcal{S}$) in *polynomial time* and even in an *online manner* for at least $\beta_n^{(t-k)}$ fraction of $\{0,1\}^n$.

Applications. We can directly apply the attacks of Theorems 1 and 2 to obtain the applications below.

- **Targeted data poisoning on learners.** Theorem 1 can model any random process $(w_1, \dots, w_m)$ that generates an object h that might or might not belong to an (undesirable) set $\mathcal{B}$ with some probability μ. In that case, we can define the output of

	Targeted	Poly-time	Corruption model	Budget k	Messages	Rounds
[24]	✓	-	Adaptive	Any	Uniform bits	Any
[21]	✓	✓	Adaptive	$\Omega(\sqrt{n})$	Uniform bits	Any
This work	✓	✓	Replacing	Any	Uniform bits	Any
[12, 28]	✓	✓	Replacing	$\Omega(\sqrt{n})$	Arbitrary	Any
This work	✓	✓	Replacing	Any	Arbitrary	Any
[8]	-	-	Replacing	1	Arbitrary	Any
[13]	-	-	Replacing	$\Omega(\sqrt{n})$	Arbitrary	1
[32, 36]	-	-	Replacing	Any	Arbitrary	1
[15, 21]	-	-	Adaptive	$\Omega(\sqrt{n})$	Arbitrary	Any
[22]	-	-	Replacing	1	Arbitrary	Any
[23]	-	-	Adaptive	1	Arbitrary	Any

Table 1. Summary of related attacks on single-turn coin tossing protocols.

the process to be $b = 1$ if $h \in \mathcal{B}$, and then an adversary can *increase* the probability of falling into $\mathcal{S}$ through a k-replacing attack. Now, suppose w_i is a batch of data provided by the i^{th} party, and let h be a model that is deterministically trained on the data set $w_1 \cup \dots \cup w_n$. Suppose there is an specific (efficiently testable) property $\mathcal{B}$ defined over h that an adversary wants to increase its probability (e.g., h makes a specific decision on a particular test instance). Theorem 14 shows that the adversary can always increase the probability of $\mathcal{B}$ from μ to $\mu + \Omega(k/\sqrt{n})$ by changing only k of the training batches. Previously, Etesami, Mahloujifar, and Mahmoody [12, 28] proved such results only for when $k \geq \Omega(\sqrt{n})$ and Diochnos, Mahloujifar, and Mahmoody [25, 26, 29] proved a weaker bound of $\mu + \Omega(k/n)$.

- **Computational isoperimetry in product spaces.** Let $\mathbf{w}_{\leq n} \equiv (\mathbf{w}_1 \times \dots \times \mathbf{w}_n)$ be a product distribution of dimension n, and let HD be the Hamming distance $\mathrm{HD}(w_{\leq n}, w'_{\leq n}) = |\{i \mid w_i \neq w'_i\}|$. Then, a basic question in functional analysis is how quickly noticeable events expand under Hamming distance. It is known, e.g., by results implicit in [1, 33] and explicit in [32, 36][8] that if a set $\mathcal{S}$ has measure μ, the k-expansion of it (i.e., the set of points with a neighbor in $\mathcal{S}$ of distance at most k) will have have measure at least $\mu + \Omega(k \cdot \mu/\sqrt{n})$ for $k = O(\sqrt{n})$. The previous works of [12, 28] introduced an *algorithmic* variant of the measure concentration phenomenon and showed how to obtain polynomial time algorithms that achieve the following. Given a random point $w_{\leq n} \in \mathbf{w}$, we can *find* a neighbor of distance at most k in $\mathcal{S}$ with probability $\mu + \Omega(k \cdot \mu/\sqrt{n})$.

  Their result above only apply to the setting where $k \geq \Omega(\sqrt{n})$, and it remained open to obtain such computational concentration for any small $k = o(\sqrt{n})$. For such small k, the problem is more suitable to be called an *isoperimetric* problem, due to historic reasons. By applying our Theorem 1 we directly get computational concentration/isoperimetry results for any $k = o(\sqrt{n})$ in any product space. For the case of uniform random bits and probabilities corresponding to Hamming balls, our Theorem 2 shows how to obtain results that match the corresponding lower bound

[8] A weaker version for uniform bits is known as the blowing-up lemma [31].

on the vertex isoperimetry [19], and we do so by using polynomial time algorithms. See the full version for the details of the polynomial time extension.

1.1 Technical overview

Here, we describe the key ideas behind our main results of Theorems 1 and 2 at a high level. We prove Theorem 1 by giving a novel inductive analysis (over adversary's budget k) for a variant of the attack of [28]. Interestingly, even though the attack of [12] improves [28] for many-replacing regime, we are not able to build our few-replacing attacks on that of [12]! We also do a modification to the [28] (by *always* looking at a message before changing or not changing it) that allows us to significantly improve the exact bound. Our modification of the attack of [28] makes the attack's description simpler and allows for sharper analysis (even in the many-replacing regime of [28], but that is not our focus here). In fact, that change is crucial to obtain our Theorem 2 which gives an *optimal* bounds for uniform binary messages.

Our proof of Theorem 2 is inspired by the recent work of Khorasgani et al. [23] who studied 1-*replacing information-theoretic non-targeted* attacks, but we still use ideas from their work in our setting. In particular, we use a concave function as the lower bound of the success probability of our attack and use induction over the number of bits n. The exact attack and the details of our inductive proof, however, are quite different from the work of Khorasgani et al. [23].

Outline. We first describe our ideas for Theorem 1 and then will do so for Theorem 2. For Theorem 1, we will first sketch the proofs of [12, 21, 28][9] and explain why they require $k = \Omega(k)$ replacements to give a meaningful bound. Then, we explain our new ideas that allow bypassing the barrier of $k = \Omega(k)$.

In the following, we explain our new ideas behind the proof of Theorems 1 and 2.

Why the attacks of [12, 21, 28] need $k = \Omega(\sqrt{n})$ corruptions. The targeted attacks of [12, 21, 28] have a similar core that make them rely on many $k = \Omega(\sqrt{n})$ number of corruptions to achieve bias towards 1. These attacks first show that certain specific attacks with *unlimited* budget can significantly bias the output of the function towards 1. Then, in the second step, they show that the number of corruptions of such ∞-replacing attacks will not be more than $O(\sqrt{n})$. To contrast our approach, the analysis of our attack for proving Theorem 1 starts from $k = 1$ and increases k, while those of [12, 21, 28] start from $k = \infty$ and show that it does not have to be more than $k = \Theta(\sqrt{n})$.

Notation. Let w_i be the i'th message sent by the i'th party, and let v_i be the possible modified version ($v_i \neq w_i$ if the adversary corrupts the i^{th} party and changes its message). We let $w_{\leq i} = (w_1, \ldots, w_i)$ and $v_{\leq i}$ is defined similarly. Let $f(v_1, \ldots, v_n) = b$ be the Boolean function that determines the final output bit b. Also $\mu = \Pr[b = 1]$ holds in the original (no-attack) protocol. (See Section 2 for all the definitions.)

The attacks of [12, 21, 28] all track the expected value $\bar{f}(v_{\leq i-1}) = \Pr[b = 1 \mid v_{\leq i-1}]$ of the final bit b conditioned on the current messages $v_{\leq i-1}$ (which forms a Doob martingale). Let w_i be the honestly prepared message of the i'th party that is

[9] In case [21], here we refer to their proof for the case of bitwise messages. Their attack for the long-message setting is (inherently) an non-targeted attack, and not a PPT one.

about to be sent in round i. If the number of corruptions has not reached k yet, with the attack parameter $\lambda \in [0,1]$, do as follows.

1. Even before looking at w_i, if there is *some* v_i that increases the expected value of b by λ (i.e., $\bar{f}(v_{\le i}) > \bar{f}(v_{\le i-1}) + \lambda$) then corrupt the i'th party and send v_i instead.
2. Otherwise, look at w_i. If, it is going to decrease the expected value of b by more than λ (i.e., $\bar{f}(v_{\le i}, w_i) < \bar{f}(v_{\le i-1}) - \lambda$), then again corrupt message w_i to v_i.
3. Otherwise, do not corrupt the i'th party, and let $v_i = w_i$ remain unchanged.

Analysis of [28]. The main ideas in the analysis of [28] are as follows.

1. Ignoring the number of corruptions, the ∞-replacing attack achieves expected value $1 - \mathrm{err}(\lambda, \mu, n)$, where $\mathrm{err}(\lambda, \mu, n) = e^{-\Omega(\mu^2/(n\lambda^2))}$ is an "Azuma error".
2. For every corruption, the expected value of the output jumps up by at least λ.

Relying on the above two keys, [28] proved that the total expected number of corruptions cannot be larger than $1/\lambda$, so by choosing $\lambda \approx \mu/\sqrt{n}$, they can achieve both (1) high expected value $1 - \mathrm{err}(\lambda, \mu, n)$ and (2) few corruptions $k \le 1/\lambda \approx \sqrt{n}/\mu$.

A candidate one-replacing targeted PPT attack. We now propose our new one-replacing attack that we will analyze using new ideas. The first version of our attack follows that of [28] and immediately stops as soon as the first corruption happens. Note that, the analysis of [28] says nothing about the power of this 1-replacing attack, as this attack is cut prematurely.

Idea 1: we gain as soon as the corruption happens. Our first key idea is that, the additive attack of [28] (as opposed to the "multiplicative" attack of [12]) *always* gains by λ, whenever a corruption happens. So, to analyze our 1-replacing attacks, all we need is to lower bound the probability p_1 of one corruption.

Idea 2: 1*-replacing is as good as* ∞*-replacing if no corruptions happens.* As long as no corruption has happened, our one-replacing attacker is actually *identical* to an attack with no limit on the number of corruptions. Also, note that the probability of outputting 1 in the ∞-replacing attack of [28] is $1 - \mathrm{err}(\lambda, \mu, n)$. Therefore, we conclude that if we run the one-replacing attack, with probability $1 - \mathrm{err}(\lambda, \mu, n)$ we *either* output 1 (which is good enough) *or* do at least one corruption (which is also good for us!). Since the probability of outputting 1 *without* any attacks is exactly μ, we can now lower bound p_1 and conclude that

$$p_1 \ge 1 - \mathrm{err}(\lambda, \mu, n) - \mu.$$

Having the above bound on p_1, we lower bond output's expected value μ_1 under our 1-replacing attack is

$$\mu_1 \ge \mu + \lambda \cdot (1 - \mathrm{err}(\lambda, \mu, n) - \mu).$$

We can now choose $\lambda = \Theta(\mu/\sqrt{n})$ which leads to up bias $\Omega(\mu/\sqrt{n})$. This attack can be made polynomial time by approximating output's Doob martingale.

Induction on k to obtain k-replacing targeted attack. Having the 1-replacing attack above, it is now tempting to apply them recursively to get k-replacing attacks. Note that this is possible only because we have a *targeted* attack, and so we can recursively apply such attack k times, each of which is a one-replacing attack, and increase the expected value of the output bit gradually. This approach, however, remains polynomial time only for $k = O(1)$. Here, we take a different approach and directly analyze the k-replacing attack of [28] using induction on k.

The idea is to allow the ∞-replacing attack of [28] run for k corruptions in total rather than one, and then trying to analyze it by induction on k. Suppose p_k is the probability that the k-replacing attack reaches its k'th corruption. Also, let μ_i be the expected value of the output b under the i-replacing targeted attack. A key idea is that all we have to do is to lower bound the probability of the corruptions happening, and by linearity of expectation we will indeed gain by at least $\lambda \cdot k$ in expected value of the outcome. In fact, we go one step further and relate the gain in the k'th corruption directly to the gain already obtained through $k - 1$ corruptions. I.e., by linearity of expectation, we have:

$$\mu_k \geq \mu_{k-1} + \lambda \cdot p_k .$$

The intuition is that before reaching the k'th corruption, the two attack are the same, and once the k'th corruption happens, the k-replacing attack gets a jump of λ up compared to the $(k - 1)$-replacing attack. Again, all we need is to lower bound p_k. To do so, we again use a generalization of the idea that we described for the case of one-replacing above. Namely, we note that as long as the k'th corruption does not happen in the k-replacing attack, it is again *indistinguishable* from the ∞-replacing attack of [28]. Also, the $(k-1)$-replacing attack reaches $b = 1$ with probability μ_{k-1} already. Using a union bound, we get:

$$p_k \geq 1 - \mathrm{err}(\lambda, \mu, n) - \mu_{k-1},$$

using which we can get that the expected value of b under the k-replacing attack is

$$\mu_k \geq \mu + \lambda \cdot (1 - \mathrm{err}(\lambda, \mu, n) - \mu_{k-1}) .$$

Solving the recursive inequalities above, we lower bound μ_k as in Theorems 1 and 14.

We now describe some of the key ideas behind our proof of Theorem 2, which deals with uniform binary messages. In this section, we mainly focus on showing the core ideas that lead to the information theoretic optimal k-replacing attacks of Theorem 2, which deals with online attacks. In the full version of this paperwe show how to use similar ideas (by approximating the Doob martingale of the final output bit) used for the polynomial-time attacks for Theorem 1 to also extend our information theoretic attacks for Theorem 2 to polynomial time variants.

Notation. First, we define the key notations that are needed for our overview of the ideas behind the proof of our Theorem 2. Here, all the original messages are *independent and uniform* random bits, which we denote with $(u_1, \ldots, u_n)$. Also, we let $\mathcal{S}$ be the set of input sequences that lead to output 1, namely $\mathcal{S} = \{x \mid f(x) = 1\}$. We know that $\Pr[(u_1, \ldots, u_n) \in \mathcal{S}] = \Pr[\sum u_i \geq t] = \beta_n^{(t)}$ is that of a Hamming ball. The goal of the adversary is to maximize the probability of falling into $\mathcal{S}$ through k-replacements in an online way. We now define the "online expansion" under optimal online k-replacing attacks, both as a function of sets, or as a function of set probabilities. (See Definition 17 for more details.) Let A be an online k-replacing adversary over the uniform distribution over $\{0, 1\}^n$. Let $\mathsf{OnExp}^{(\mathsf{A})}(\mathcal{S})$ be the probability that A can map a random input to $\mathcal{S}$ through its online k-replacing attack. Let $\mathsf{OnExp}^{(k)}(\mathcal{S})$ be the maximum over

$\mathsf{OnExp}^{(\mathsf{A})}(\mathcal{S})$ among all k-replacing attacks, and let the following be the minimum of $\mathsf{OnExp}^{(k)}(\mathcal{S})$ among all sets of measure μ.

$$\mathsf{OnExp}_n^{(k)}(\mu) = \inf_{\mathcal{S}, \Pr[\mathcal{S}] \geq \mu} \mathsf{OnExp}^{(k)}(\mathcal{S}).$$

Our key idea is to show that the following piecewise-linear function is a *lower bound* on the power of k-replacing attacks. We prove this by induction on n. In comparison, [23] also used similar piecewise-linear functions, but their goal was to obtain *1-corrupting information theoretic non-targeted* attacks. It is possible that using similar techniques, one can make the attack of [23] also polynomial time, but the key differences are due to the fact that [23] aims for a non-targeted attack, and hence it ends up with a completely different recursive relation and induction on n.

Definition 3 (The piecewise-linear lower bound – informal)**.** For any non-negative integers k, n, the function $\ell_n^{(k)} \colon [0,1] \to [0,1]$ is defined as follows.

- If $\mu = \beta_n^{(t)}$ for any $t \in [n]$, it holds that $\ell_n^{(k)}\left(\beta_n^{(t)}\right) = \beta_n^{(t-k)}$. Namely, when the input probability is that of an exact Hamming balls, $\ell_n^{(k)}$ returns their probability after expanding them to include anything within their k Hamming distance (which is also a Hamming ball).
- Connect all the $n+2$ points above to obtain a piecewise-linear function $\ell_n^{(k)}$.

See Definition 24 for a formal definition of the function above.

Recursive relation for $\mathsf{OnExp}_n^{(k)}(\mu)$. We then use a recursive relation that can be used to exactly compute $\mathsf{OnExp}_n^{(k)}(\mu)$ for all k, n, μ (see Definition 20). The idea of the recursive relation is to model adversary's decision based on optimal decisions. In fact, if an adversary is given a bit $u_i = 0$, and it holds that $\Pr[(0, u_2, \ldots, u_n) \in \mathcal{S}] = \mu_0, \Pr[(1, u_2, \ldots, u_n) \in \mathcal{S}] = \mu_1$. Then, an optimal online adversary shall decide between changing it to 1 or not, and if it knows the optimal solutions for $\mathsf{OnExp}_{n-1}^{(k)}(\mu_0)$ (reflecting the "no change" decision) and $\mathsf{OnExp}_{n-1}^{(k-1)}$ (reflecting the "change" decision) it can make the optimal decision.

Using lower bounds lead to lower bounds. We prove by induction on n, that if one uses *lower bounds* (e.g., $\ell_{n-1}^{(k)}$ and $\ell_{n-1}^{(k-1)}$) instead of $\mathsf{OnExp}_{n-1}^{(k)}(\mu_0)$ and $\mathsf{OnExp}_{n-1}^{(k-1)}(\mu_1)$ in the recursive relation that computes $\mathsf{OnExp}_n^{(k)}$, then one obtains a lower bound on $\mathsf{OnExp}_n^{(k)}(\mu)$. This part of the proof follows from the monotonicity of the recursive relation for $\mathsf{OnExp}_n^{(k)}$.

Function $\mathsf{OnExp}_n^{(k)}(\mu)$ *remains a lower bound for* $\ell_n^{(k)}$. We also show that when we apply the recursive relation over $\ell_{n-1}^{(k)}$ and $\ell_{n-1}^{(k-1)}$, the result will be an *upper bound* on $\ell_n^{(k)}$. This, together with the step above implies that $\ell_n^{(k)}$ remains a lower bound $\ell_n^{(k)}$. This is the most technical step of the proof that goes through a careful case study and heavily relies on the concavity and monotonicity of $\ell_n^{(k)}$.

Making the attack polynomial time. In the actual polynomial time attack, the adversary approximates μ, and it uses $\ell_n^{(k)}$ (which is efficiently computable) instead of $\mathsf{OnExp}_n^{(k)}(\mu)$

in the recursive relation and decides to change or not change the bits. See the full version of the paper for the details of making the attack polynomial time.

1.2 Further related work

Many of the related works were already discussed in previous sections. In this section, we discuss other works related to ours, mostly in the context of coin tossing protocols.

Adaptive corruption. As explained above, our results are proved in the *strong* adaptive corruption model. However, many works study the power of standard adaptive corruption in coin tossing protocols. The main result in [21] indeed proves the existence of such attacks that achieve *non-targeted* biasing that controls the output fully when the number of corruptions is $k \geq \sqrt{n}$. Haitner and Karidi-Heller [15] further generalized this result to multi-turn protocols, resolving a long-standing open problem of Ben-Or and Linial [4]. Dodis [11] previously proved that certain black-box methods cannot break this conjecture. The recent work of Khorasgani, Maji, and Wang [22, 23] showed that for the case of 1 replacing, (computationally unbounded) adaptive adversaries can achieve *non-targeted* bias $\Omega(1/\sqrt{n})$ in single-turn protocols.

Static corruption. A static adversary chooses the corrupted parties independently of the execution of the protocol, and hence can fix the corrupted set ahead of the execution. The previously mentioned works of [3,5,25,27,29] all fall into this framework and prove that corrupting k parties can lead to bias $\Omega(\mu k/n)$ statically. These results hold even if the statically corrupted set is chosen *at random*. For single-round protocols in which each party sends a single bit, Kahn, Kalai and Linial [20] showed that any protocol is susceptible to $\Omega(n/\log n)$ corruptions. A long line of exciting works (see [35]) showed how to achieve robustness to $(1-\delta)\cdot n$ static corruption for any $\delta < 1$.

Fair coin tossing. Another line of work in coin tossing protocols aims to study the power of *fair* protocols in which the parties *need* to output a bit even if the other party is caught cheating (e.g., by aborting in the middle of the protocol). The work of Cleve [7] showed that in any such protocol with r rounds between two parties, there is a PPT attacker that biases the output of the other party by at least $\Omega(1/r)$. The work of Moran, Naor, and Segev [34] showed how to *match* this bound assuming oblivious transfer, leading to an "optimally fair" protocol. A sequence of works [9, 10, 16, 17] showed barriers for doing so from one-way functions, and finally, the beautiful work of Maji and Wang [30] completely resolved this question for black-box constructions. For works on fair coin tossing in the multiparty settings see [2, 18].

2 Preliminaries

General notation. We use calligraphic letters (e.g., $\mathcal{X}$) for sets. All distributions and random variables in this work are discrete. We use bold letters (e.g., $\mathbf{w}$) to denote random variables that return a sample from a corresponding discrete distribution. By $w \leftarrow \mathbf{w}$ we denote sampling w from the random variable $\mathbf{w}$. By $\mathrm{Supp}(\mathbf{w})$ we denote the support set of $\mathbf{w}$. For an event $\mathcal{S} \subseteq \mathrm{Supp}(\mathbf{w})$, the probability function of $\mathbf{w}$ for $\mathcal{S}$ is denoted as $\Pr[\mathbf{w} \in \mathcal{S}] = \Pr_{w \leftarrow \mathbf{w}}[w \in \mathcal{S}]$ or simply as $\Pr[\mathcal{S}]$ when $\mathbf{w}$ is clear from the context. By $\mathbf{u} \equiv \mathbf{v}$ we denote that the random variables $\mathbf{u}$ and $\mathbf{v}$ have the same distributions. Unless stated otherwise, we denote vectors by using a bar over a variable. By $(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$ we refer to a sequence of n *jointly sampled* random variables. For a vector $(w_1 \ldots w_n)$, we use $w_{\leq i}$ to denote the prefix $(w_1, \ldots, w_i)$, and we use the same notation $\mathbf{w}_{\leq i}$ for jointly distributed random variables. For vector $x = u_{\leq i-1}$ and $y = u_i$, by, by xy we denote the vector $u_{\leq i-1}$ that appends u_i as the last coordinate of x. For a jointly distributed random variables $(\mathbf{u}, \mathbf{v})$, by $(\mathbf{u} \mid \mathbf{v} = v)$ or we denote the random variable $\mathbf{u}$ conditioned on $\mathbf{v} = v$. When it is clear from the context, we simply write $(\mathbf{u} \mid v)$ or $\mathbf{u}[v]$ instead. By $\mathbf{u} \times \mathbf{v}$ we refer to the product distribution in which $\mathbf{u}$ and $\mathbf{v}$ are sampled independently. $\mathrm{HD}(u_{\leq n}, v_{\leq n}) = |\{i \mid u_i \neq v_i\}|$ denotes the Hamming distance for vectors of n coordinates.

Random processes. Let $\mathbf{w}_{\leq n} \equiv (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ be a sequence of jointly distributed random variables. We can interpret the distribution of $\mathbf{w}_{\leq i}$ as a random process in which the i^{th} block w_i is sampled from the marginal distribution $(\mathbf{w}_i \mid w_{\leq i-1}) \equiv (\mathbf{w}_i \mid \mathbf{w}_{\leq i-1} = w_{\leq i-1}) \equiv \mathbf{w}_i[w_{\leq i-1}]$. We also use $\mathbf{w}_{\leq n}[\cdot]$ to denote an oracle sampling algorithm that given $w_{\leq i}$ returns a sample from $\mathbf{w}_{\leq n}[w_{\leq i}]$.

Attack model. Our adversaries *replace* a message/block in a random process. Namely, they observe the blocks one by one and sometimes intervene to replace them with a new value. (The new values will subsequently change the way the random process will proceed.) Hence, we refer to them as *replacing* adversaries. Such adversaries are equivalent to *strongly adaptive corrupting* adversaries as defined in [13].

Definition 4 (Online replacing attacks on random processes)**.** Let $\mathbf{w}_{\leq n} \equiv (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ be a random process. Suppose $\mathsf{A}(x, \sigma) \to (x', \sigma')$ is a (potentially randomized) algorithm with the following syntax. It takes as input some (randomness,) x and σ, where σ is interpreted as a "state", and it outputs (x', σ'). We call such algorithm an *online replacing adversary* and define the following properties for it.

We define the following notions for $\mathbf{w}_{\leq n}$.

- **The generated and output random processes under replacing attacks.** Suppose A is an replacing algorithm. We now define two random processes that result from running the replacing adversary A to influence the original random process $\mathbf{w}_{\leq n}$. For $i = 1, 2, \ldots, n$, we first sample $u_i \leftarrow (\mathbf{w}_i \mid \mathbf{w}_{\leq i-1} = v_{\leq i-1})$, and then we obtain $(v_i, \sigma_i) \leftarrow \mathsf{A}(u_i, \sigma_{i-1})$. If at any point during this process $\Pr[\mathbf{w}_{\leq i} = v_{\leq i}] = 0$, we will output $u_{i+1} = \cdots = u_n = v_{i+1} = \cdots = v_n = \perp$. We call $(\mathbf{u}_{\leq n}, \mathbf{v}_{\leq n})$ the *jointly generated* random processes under the attack. We also refer to $u_{\leq n}$ as the *original values* and $v_{\leq n}$ as the *output* of the random process under the attack A.
- **Online replacing.** We call A a *valid* (online replacing) attack on $\mathbf{w}_{\leq n}$, if with probability 1 over the generation of $u_{\leq n}, v_{\leq n}$, it holds that none of the coordinates are $\perp$ (i.e., $\Pr[\mathbf{w}_{\leq i} = v_{\leq i}] \neq 0$.) In this work we always work with valid online replacing attacks, even if they are not called valid.
- **Budget of replacing attacks.** Replacing adversary A has *budget* k, if

$$\Pr[\mathrm{HD}(\mathbf{u}_{\leq n}, \mathbf{v}_{\leq n}) \leq k] = 1,$$

where $(\mathbf{u}_{\le n}, \mathbf{v}_{\le n})$ are the jointly generated random processes that are also jointly distributed.

- **Algorithmic efficiency of attacks.** If $\mathbf{w}_{\le n}$ is indexed by n as a member of a *family* of joint distributions defined for all $n \in \mathbb{N}$, then we call an online or offline replacing algorithm *efficient*, if its running time is at most $\mathrm{poly}(N)$ where N is the total bit-length representation of any $w_{\le n} \in \mathrm{Supp}(\mathbf{w}_{\le n})$. We would also consider efficiency where the replacing algorithm uses an oracle. In particular, we say an attack $\mathsf{A}^{\mathbf{w}_{\le n}[\cdot]}$ with oracle access to sampler $\mathbf{w}_{\le n}[\cdot]$ is efficient if it runs in time $\mathrm{poly}(N)$.

We now recall the so-called Doob martingale of a (Boolean-output) random process.

Definition 5 (Doob martingale, partial averages, and their approximate variant)**.** For random process $\mathbf{w}_{\le n} \equiv (\mathbf{w}_1, \ldots, \mathbf{w}_n)$, let $f\colon\ \mathrm{Supp}(\mathbf{w}_{\le n}) \mapsto \mathbb{R}$, $i \in [n]$, and $w_{\le i} \in \mathrm{Supp}(\mathbf{w}_{\le i})$. Then we use the notation $\bar{f}(w_{\le i}) = \mathbb{E}_{w_{\le n} \leftarrow (\mathbf{w}_{\le n} | w_{\le i})}[f(w_{\le n})]$ to define the expected value of f for a sample from $\mathbf{w}_{\le n}$ conditioned on the prefix $w_{\le i}$ and refer to it as a *partial-average* of f. In particular, using notation $w_{\le 0} = \varnothing$, we have $\bar{f}(\varnothing) = \mathbb{E}[f(\mathbf{w}_{\le n})]$. The random process $(\bar{f}(\mathbf{w}_{\le 1}), \ldots, \bar{f}(\mathbf{w}_{\le n}))$ is called the Doob martingale of the function f over the random process $\mathbf{w}_{\le n}$. For the same $\mathbf{w}_{\le n}$ and $\bar{f}(\cdot)$, we call $\tilde{f}(\cdot)$ an (additive) ε-approximation of $\bar{f}(\cdot)$, if for all $w_{\le i} \in \mathrm{Supp}(\mathbf{w}_{\le i})$, it holds that $\tilde{f}(w_{\le i}) \in \bar{f}(w_{\le i}) \pm \varepsilon$.

If one is given oracle access to ℓ samples from $(\mathbf{w}_i \mid w_{\le i})$, then by averaging them, one can obtain (due to the Hoeffding inequality) an ε-approximation of $\tilde{f}(w_{\le i})$ for with probability $1 - \exp(-\ell/\varepsilon^2)$.

2.1 Useful facts

We use the following variant of the Azuma inequality which is proved in [14].

Lemma 6 (Azuma's inequality for dynamic interval lengths (Theorem 2.5 in [14]))**.** *Let $\mathbf{t}_{\le n} \equiv (\mathbf{t}_1, \ldots, \mathbf{t}_n)$ be a sequence of n jointly distributed random variables such that for all $i \in [n]$, and for all $t_{\le i-1} \leftarrow \mathbf{t}_{\le i-1}$, we have*

$$\exists t^*, \quad \Pr_{t_i \leftarrow \mathbf{t}_i | t_{\le i-1}}[t^* + \eta_i \ge t_i \ge t^* - \eta_i] = 0$$

and $\mathbb{E}[\mathbf{t}_i \mid t_{\le i-1}] \ge 1$. Then, we have

$$\Pr\left[\sum_{i=1}^{n} \mathbf{t}_i \le -s\right] \le \mathrm{e}^{\frac{-s^2}{2\sum_{i=1}^{n} \eta_i^2}}$$

Lemma 7 (Azuma's inequality for dynamic interval lengths under approximate conditions)**.** *Let $\mathbf{t}_{\le n} \equiv (\mathbf{t}_1, \ldots, \mathbf{t}_n)$ be a sequence of n jointly distributed random variables such that for all $i \in [n]$, and for all $t_{\le i-1} \leftarrow \mathbf{t}_{\le i-1}$, we have*

$$\exists t^*, \quad \Pr_{t_i \leftarrow \mathbf{t}_i | t_{\le i-1}}[|t_i| \ge 1] = 0$$

$$\exists t^*, \quad \Pr_{t_i \leftarrow \mathbf{t}_i | t_{\le i-1}}[t^* + \eta_i \ge t_i \ge t^* - \eta_i] \ge 1 - \gamma$$

and $\mathbb{E}[\mathbf{t}_i \mid t_{\leq i-1}] \geq -\gamma$. *Then, we have*

$$\Pr\left[\sum_{i=1}^{n} \mathbf{t}_i \leq -s\right] \leq \mathrm{e}^{\frac{-(s-2n\gamma)^2}{2\sum_{i=1}^{n}\eta_i^2}} + n \cdot \gamma$$

Proof. If we let $\gamma = 0$, Lemma 7 becomes equivalent to Lemma 6. Here we sketch why Lemma 7 can also be reduced to the case that $\gamma = 0$ (i.e., Azuma inequality). We build a sequence $\mathbf{t}'_i$ from $\mathbf{t}_i$ as follows: Sample $t_i \leftarrow \mathbf{t}_i \mid t_{\leq i-1}$, if $|t_i - t^*| \leq \eta_i$, output $t'_i = t_i + 2\gamma$ otherwise output $t^* + 2\gamma$. We have $\mathbb{E}[\mathbf{t}'_i \mid t'_{\leq i-1}] \geq 0$ and $\Pr[|t'_i - t^* - 2\gamma| > \tau_i] = 0$. Now we can use Lemma 7 for the basic case of $\gamma = 0$ for the sequence $\mathbf{t}'_i$ and use it to get a looser bound for sequence $\mathbf{t}_i$, using the fact that $\exists i \in [n], |t_i - t^*| \geq \eta_i$ happens with probability at most $n \cdot \gamma$. □

Lemma 8 (Composition of concave functions)**.** *Suppose ℓ_1 and ℓ_2 are two non-decreasing concave functions. Then $\ell_1(\ell_2)$ is also non-decreasing and concave.*

3 Attacking protocols with any message length

In this section, we design and analyze our k-replacing up-biasing attack on random processes with arbitrary alphabet size. We first describe our attack in an idealized model in which the partial-average oracle $\bar{f}(\cdot)$ and "maximum child" of a prefix of the process are available for free. In the full version of this paper, we show that our attack can be made polynomial-time using an approximation of the partial-average oracle that can be obtained in polynomial time.

Construction 9 (k-replacing attack using exact oracles)**.** This attack uses the exact partial-average oracle $\bar{f}(\cdot)$ and another oracle that returns "the best choice" for the next block (see u^*_{i+1} defined below). The attack is also parameterized by a vector $\lambda_{\leq k} = (\lambda_1, \ldots, \lambda_k) \in [0,1]^k$ for some integer $k \leq n$ which is adversary's budget. The attack will keep state $\sigma_i = (u_{\leq i}, v_{\leq i})$ where $u_{\leq i}$ are the original values and $v_{\leq i}$ are the output values under attack.[10] Having state $(u_{\leq i}, v_{\leq i})$ and for given u_{i+1} the algorithm A will decide on whether to keep or replace u_{i+1}, using $u^*_{i+1} = \operatorname{argmax}_{u'_{i+1}} \bar{f}(v_{\leq i}, u'_{i+1})$, $\bar{f}^* = \bar{f}(v_{\leq i}, u^*_{i+1})$, and $d = \mathrm{HD}(u_{\leq i}, v_{\leq i})$ as follows.

- (Case 0) If $d \geq k$, do not change u_{i+1} and output $v_{i+1} = u_{i+1}$.
- (Case 1) if Case 0 does not happen and $\bar{f}(v_{\leq i}, u_{i+1}) < \bar{f}^* - \lambda_{d+1}$, then $\mathsf{A}[\lambda_{\leq k}](u_{i+1})$ will return the output $v_{i+1} = u^*_{i+1}$ which is different from u_{i+1}.
- (Case 2) If Cases 0, 1 do not happen, do not change u_{i+1} and output $v_{i+1} = u_{i+1}$.

In all the cases above, A will also update the state as $\sigma_{i+1} = (u_{\leq i+1}, v_{\leq i+1})$.

Notation. Suppose we run the attack $\mathsf{A}[\lambda_{\leq k}]$ on random process $\mathbf{w}_{\leq n}$ through the process described in Definition 4. (In particular, u_{i+1} will be sampled from $(\mathbf{w}_{i+1} \mid \mathbf{w}_{\leq i} =$

[10] Attack would need $v_{\leq i}$ and the "used part of the budget" $\mathrm{HD}(u_{\leq i}, v_{\leq i})$. Both of these can be obtained from $\sigma_i = (u_{\leq i}, v_{\leq i})$.

$v_{\le i}$).) We use $(\mathbf{u}^{(k)}_{\le n}, \mathbf{v}^{(k)}_{\le n})$ to denote the jointly generated random processes under the attack $\mathsf{A}[\lambda_{\le k}]$. (This notation allows us to distinguish between the generated random processes under attacks with different budget.) We sometimes use $(\mathbf{u}^{(\infty)}_{\le n}, \mathbf{v}^{(\infty)}_{\le n})$ to denote $(\mathbf{u}^{(n)}_{\le n}, \mathbf{v}^{(n)}_{\le n})$ as they are the same distributions. Also, let

$$\mu_k = \mathop{\mathbb{E}}_{(u_{\le n}, v_{\le n}) \leftarrow (\mathbf{u}^{(k)}_{\le n}, \mathbf{v}^{(k)}_{\le n})} [f(v_{\le n})]$$

denotes the expected value of f over the sequence that is the output of k-replacing attack of Construction 9. For $k = 0$ we have and $\mu_0 = \mu = \mathbb{E}[f(\mathbf{w}_{\le n})]$.

Lemma 10 below shows that the increase in μ_k compared with μ_{k-1} can be related to the "threshold parameter" λ_k and the probability that an attack with *unlimited* (or equivalently just n) budget with threshold parameters $\lambda_1, \dots, \lambda_k, \lambda'_{k+1}, \dots, \lambda'_n$ makes at least k replacements.

Lemma 10. *We have*

$$\mu_k \ge \mu_{k-1} + \lambda_k \cdot \Pr_{(u_{\le n}, v_{\le n}) \leftarrow (\mathbf{u}^{(\infty)}_{\le n}, \mathbf{v}^{(\infty)}_{\le n})} [\mathrm{HD}(u_{\le n}, v_{\le n}) \ge k].$$

Proof. For any $j \in \{0, 1, 2\}$, let C^k_j be the Boolean random variable over (u_{i+1}, σ_i) that determines which case of the attack A with budget k happens on prefix $(v_{\le i}, u_{i+1})$ where $v_{\le i}$ is the finalized output prefix, $u_{\le i}$ is the original prefix and u_{i+1} is the original sampled block at round $i+1$. For all $(v_{\le i}, u_{\le i}, u_{i+1})$ we have $\sum_{j=0}^{2} C^k_j(u_{i+1}, \sigma_i) = 1$ because the cases complement each other.

In the rest of the proof, whenever $u_{\le i}$ and $v_{\le i}$ are clear from the context, we will use $C^k_j(u_{i+1})$ instead of $C^k_j(u_{i+1}, \sigma_i)$. In the following, when the threshold parameters $\lambda_1, \dots, \lambda_k$ are clear from the context, we will use A instead of $\mathsf{A}[\lambda_{\le k}]$.

For all $u_{\le i}, v_{\le i} \in \mathrm{Supp}(\mathbf{u}_{\le i}, \mathbf{v}_{\le i})$ we have the following qualities for different cases of the attack.

- Case 0:

$$\mathop{\mathbb{E}}_{(u_{i+1}, v_{i+1}) \leftarrow (\mathbf{u}^k_{i+1}, \mathbf{v}^k_{i+1})[u_{\le i}, v_{\le i}]} \left[\left(\bar{f}(v_{\le i}, v_{i+1}) - \bar{f}(v_{\le i}, u_{i+1}) \right) \cdot C^k_0(u_{i+1}) \right] = 0. \tag{1}$$

- Case 1:

$$C^k_1(u_{i+1}) = (C^\infty_1(u_{i+1}) \wedge \mathrm{HD}(u_{\le i}, v_{\le i}) < k). \tag{2}$$

  This is because as long as the number of replacements is fewer than k, Case 1 of the attack with budget k would go through whenever A with budget of n does so.

- Case 2:

$$\mathop{\mathbb{E}}_{(u_{i+1}, v_{i+1}) \leftarrow (\mathbf{u}^k_{i+1}, \mathbf{v}^k_{i+1})[u_{\le i}, v_{\le i}]} \left[\left(\bar{f}(v_{\le i}, v_{i+1}) - \bar{f}(v_{\le i}, u_{i+1}) \right) \cdot C^k_2(u_{i+1})) \right] = 0. \tag{3}$$

  This is correct because either $C^k_2(v_{\le i}, u_{i+1}) = 0$ or $u_{i+1} = v_{i+1}$.

We define a notation $g(v_{\leq i+1}, u_{\leq i+1}) = \bar{f}(v_{\leq i+1}) - \bar{f}(v_{\leq i}, u_{i+1})$. In the following We use the shorten forms of $\mathbb{E}_{(\mathbf{u}_{\leq i}, \mathbf{v}_{\leq i})}$ and $\mathbb{E}_{(\mathbf{u}_{\leq n}, \mathbf{v}_{\leq n})[u_{\leq i}, v_{\leq i}]}$ to refer to $\mathbb{E}_{(u_{\leq i}, v_{\leq i}) \leftarrow (\mathbf{u}_{\leq i}, \mathbf{v}_{\leq i})}$ and $\mathbb{E}_{(u,v) \leftarrow (\mathbf{u}_{\leq n}, \mathbf{v}_{\leq n})[u_{\leq i}, v_{\leq i}]}$. We have

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n}, \mathbf{v}^{(k)}_{\leq n})}[f(v_{\leq n})] - \mu = \mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n}, \mathbf{v}^{(k)}_{\leq n})} \left[\sum_{i=0}^{n-1} (\bar{f}(v_{\leq i+1}) - \bar{f}(v_{\leq i})) \right] \\
&= \mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n}, \mathbf{v}^{(k)}_{\leq n})} \left[\sum_{i=0}^{n-1} (\bar{f}(v_{\leq i+1}) - \bar{f}(v_{\leq i}, u_{i+1})) \right] \quad \text{(by the definition of } \bar{f}\text{)} \qquad (4) \\
&= \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}^{k}_{\leq i}, \mathbf{v}^{k}_{\leq i})} \mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n}, \mathbf{v}^{(k)}_{\leq n})[u_{\leq i}, v_{\leq i}]} \left[g(v_{\leq i+1}, u_{\leq i+1}) \cdot \left(\sum_{j=0}^{2} C^k_j(u_{i+1}) \right) \right] \\
&= \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}^{k}_{\leq i}, \mathbf{v}^{k}_{\leq i})} \mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n}, \mathbf{v}^{(k)}_{\leq n})[u_{\leq i}, v_{\leq i}]} \left[g(v_{\leq i+1}, u_{\leq i+1}) \cdot C^k_1(u_{i+1}) \right] \quad \text{(by (3) and (1))} \\
&\qquad (5) \\
&= \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}^{k}_{\leq i}, \mathbf{v}^{k}_{\leq i})} \mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n}, \mathbf{v}^{(k)}_{\leq n})[u_{\leq i}, v_{\leq i}]} \left[g(v_{\leq i+1}, u_{\leq i+1}) \cdot (C^{(\infty)}_1(u_{i+1}) \wedge (\mathrm{HD}(u_{\leq i}, v_{\leq i}) < k) \right] \\
&= \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}^{\infty}_{\leq i}, \mathbf{v}^{\infty}_{\leq i})} \mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n}, \mathbf{v}^{(k)}_{\leq n})[u_{\leq i}, v_{\leq i}]} [g(v_{\leq i+1}, u_{\leq i+1}) \cdot (C^{\infty}_1(u_{i+1}) \wedge (\mathrm{HD}(u_{\leq i}, v_{\leq i}) < k))] \, . \\
&\qquad (6)
\end{aligned}
$$

The last equality above holds, because for all $u_{\leq i}, v_{\leq i}$ where $\mathrm{HD}(u_{\leq i}, v_{\leq i}) < k$,

$$
\Pr[(\mathbf{u}^k_{\leq i}, \mathbf{v}^k_{\leq i}) = (u_{\leq i}, v_{\leq i})] = \Pr[(\mathbf{u}^{(\infty)}_{\leq i}, \mathbf{v}^{(\infty)}_{\leq i}) = (u_{\leq i}, v_{\leq i})].
$$

The reason for this is that as long as we have not used the full budget k, the k-replacing attack will behave as if its budget is infinite.

Similarly, for the adversary A with budget $k-1$ we have

$$
\mathop{\mathbb{E}}_{(\mathbf{u}^{(k-1)}_{\leq n}, \mathbf{v}^{(k-1)}_{\leq n})}[f(v_{\leq n})] - \mu = \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}^{(\infty)}_{\leq i}, \mathbf{v}^{(\infty)}_{\leq i})_{\leq i}} \mathop{\mathbb{E}}_{(\mathbf{u}^{(k-1)}_{\leq n}, \mathbf{v}^{(k-1)}_{\leq n})[u_{\leq i}, v_{\leq i}]} [\eta(u_{\leq i+1}, v_{\leq i+1})] \, . \qquad (7)
$$

where $\eta(u_{\leq i+1}, v_{\leq i+1}) = g(v_{\leq i+1}, u_{\leq i+1}) \cdot \big(C^{\infty}_1(u_{i+1}) \wedge (\mathrm{HD}(u_{\leq i}, v_{\leq i}) < k-1) \big)$. Therefore, by combining Equations (6) and (7) we have

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n},\mathbf{v}^{(k)}_{\leq n})}[f(v_{\leq n})] - \mathop{\mathbb{E}}_{(u_{\leq n},v_{\leq n})\leftarrow(\mathbf{u}^{(k-1)}_{\leq n},\mathbf{v}^{(k-1)}_{\leq n})}[f(v_{\leq n})] = \\
&\sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}^{(\infty)}_{\leq i},\mathbf{v}^{(\infty)}_{\leq i})} \mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n},\mathbf{v}^{(k)}_{\leq n})[u_{\leq i},v_{\leq i}]} [g(v_{\leq i+1},u_{\leq i+1}) \cdot C_1^{\infty}(u_{i+1}) \cdot (\mathrm{HD}(u_{\leq i},v_{\leq i}) = k-1)] \\
&\geq \sum_{i=0}^{n-1} \mathop{\mathbb{E}}_{(\mathbf{u}^{(\infty)}_{\leq i},\mathbf{v}^{(\infty)}_{\leq i})} \left[\lambda_k \cdot \mathop{\mathbb{E}}_{(\mathbf{u}^{(k)}_{\leq n},\mathbf{v}^{(k)}_{\leq n})[u_{\leq i},v_{\leq i}]} [C_1^{\infty}(u_{i+1}) \cdot (\mathrm{HD}(u_{\leq i},v_{\leq i}) = k-1)]\right] \\
&= \lambda_k \cdot \Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[\mathrm{HD}(u_{\leq n},v_{\leq n}) \geq k].
\end{aligned}
$$

The last equality above holds because whenever $C_1^{(\infty)}$ holds, we know that A will replace u_{i+1} with $v_{i+1} \neq u_{i+1}$ and this makes the hamming distance of $u_{\leq i+1}$ from $v_{\leq i+1}$ equal to k. □

Now we prove the following lemma about the power of attacks with infinite budget. The work of [28] also prove a similar bound (see Claim 19 in [28]) for their attack but our attack achieves a better bound because of the fact that our attack has only one step in which the replacement might happen which allows us to make a better use of Azuma's inequality with dynamic interval (See Lemma 6).

Lemma 11. *If* $\mu_\infty = \mathbb{E}_{(u_{\leq n},v_{\leq n})\leftarrow(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[f(v_{\leq n})]$ *and* $\lambda = \max_{i\in[n]} \lambda_i$*, then*

$$\mu_\infty \geq 1 - e^{-\frac{2\mu^2}{n\lambda^2}}.$$

Proof. We define a sequence of random variables $\mathbf{t}_{\leq n} = (\mathbf{t}_1, \ldots, \mathbf{t}_n)$, where $t_{i+1} = \bar{f}(v_{\leq i+1}) - \bar{f}(v_{\leq i})$ is a random variable that is dependent on $v_{\leq i+1}$. Then we have

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})[u_{\leq i},v_{\leq i}]} [\bar{f}(v_{\leq i+1}) - \bar{f}(v_{\leq i})] \\
&\geq \mathop{\mathbb{E}}_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})[u_{\leq i},v_{\leq i}]} [\bar{f}(v_{\leq i},u_{i+1}) - \bar{f}(v_{\leq i})] = 0.
\end{aligned}
$$

Therefore, $\mathbf{t}_{\leq n}$ defines a sub-martingale. Furthermore, we have

$$\bar{f}^* \geq \bar{f}(v_{\leq i+1}) \geq \bar{f}^* - \lambda.$$

Therefore, t_i always falls in an interval of size λ. Hence, applying the right variant of Azuma's Inequality (as stated in Lemma 6) over $\mathbf{t}_{\leq n}$, we have

$$\Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[f(v_{\leq n}) = 0] = \Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}\left[\sum_{i=1}^{n} t_i \leq -\mu\right] \leq e^{-\frac{2\mu^2}{n\lambda^2}}. \tag{8}$$

Now, leveraging the fact that f outputs in $\{0,1\}$ and relying on Inequality (8), we have

$$\Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[f(v_{\leq n}) = 1] = 1 - \Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[f(v_{\leq n}) - \mu \leq -\mu] \geq 1 - e^{-\frac{2\mu^2}{n\lambda^2}}.$$

□

Lemma 12. *If* $\lambda = \max_{i\in[k]} \lambda_i$, *then*

$$\Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[\mathrm{HD}(u_{\leq n}, v_{\leq n}) \geq k] \geq 1 - e^{-\frac{2\mu^2}{n\lambda^2}} - \mu_{k-1}.$$

Proof. First we have

$$\begin{aligned}
&\Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[\big(f(v_{\leq n}) = 1 \wedge \mathrm{HD}(u_{\leq n}, v_{\leq n}) < k\big) \vee (\mathrm{HD}(u_{\leq n}, v_{\leq n}) \geq k)]\\
&\quad= \Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[f(v_{\leq n}) = 1 \vee \mathrm{HD}(u_{\leq n}, v_{\leq n}) \geq k]\\
&\quad\geq \Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[f(v_{\leq n}) = 1]\\
&\quad= \mu_\infty \geq 1 - e^{-\frac{2\mu^2}{n\lambda^2}} \text{ (by Lemma 11).}
\end{aligned} \tag{9}$$

On the other hand, by a union bound we have

$$\begin{aligned}
&\Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[\big(f(v_{\leq n}) = 1 \wedge \mathrm{HD}(u_{\leq n}, v_{\leq n}) < k\big) \vee (\mathrm{HD}(u_{\leq n}, v_{\leq n}) \geq k)] \leq\\
&\Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[f(v_{\leq n}) = 1 \wedge \mathrm{HD}(u_{\leq n}, v_{\leq n}) < k] + \Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[\mathrm{HD}(u_{\leq n}, v_{\leq n}) \geq k].
\end{aligned} \tag{10}$$

The generated process under $k-1$ replacing attack is same as n-replacing attack as long as the number of replacements is less than k. Therefore, it holds that

$$\Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[\big(f(v_{\leq n}) = 1 \wedge \mathrm{HD}(u_{\leq n}, v_{\leq n}) < k\big)] \leq \Pr_{(\mathbf{u}^{(k-1)}_{\leq n},\mathbf{v}^{(k-1)}_{\leq n})}[f(v_{\leq n}) = 1] = \mu_{k-1}. \tag{11}$$

Now, combining Inequalities (9), (10) and (11) we get

$$\Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[\mathrm{HD}(u_{\leq n}, v_{\leq n}) \geq k] \geq 1 - e^{-\frac{2\mu^2}{n\lambda^2}} - \mu_{k-1}.$$

□

Corollary 13. *If* $\lambda = \max_{i\in[k]} \lambda_i$, *then we have*

$$\mu_k \geq \mu_{k-1} + \lambda_k \cdot \left(1 - e^{\frac{-2\mu^2}{n\cdot\lambda^2}} - \mu_{k-1}\right).$$

Proof. Combining Lemmas 12 and 10 we have

$$\begin{aligned}
\mu_k &\geq \mu_{k-1} + \lambda_k \cdot \Pr_{(\mathbf{u}^{(\infty)}_{\leq n},\mathbf{v}^{(\infty)}_{\leq n})}[\mathrm{HD}(u_{\leq n}, v_{\leq n}) \geq k] \quad \text{(by Lemma 10)}\\
&\geq \mu_{k-1} + \lambda_k \cdot \left(1 - e^{\frac{-2\mu^2}{n\cdot\lambda^2}} - \mu_{k-1}\right) \quad \text{(by Lemma 12).}
\end{aligned}$$

□

Theorem 14. *If* $\lambda = \max_{i\in[k]} \lambda_i$, *then we have*

$$\mu_k \geq \mu + \left(1 - \prod_{i=1}^{k}(1-\lambda_i)\right) \cdot \left(1 - e^{\frac{-2\mu^2}{n\cdot\lambda^2}} - \mu\right).$$

In particular, by setting all $\lambda_i = \frac{\mu}{\sqrt{n}}$ *we get*

$$\mu_k \geq \mu + \left(1 - \left(1 - \frac{\mu}{\sqrt{n}}\right)^k\right) \cdot \left(1 - e^{-2} - \mu\right).$$

Note that the choice of $\lambda_i = u/\sqrt{n}$ above is not optimal. The optimal choice does not have a compact closed form and is actually by setting different λ_i's for different remaining budgets.

Proof. We prove this by induction on k. The case of $k = 1$ directly follows from Corollary 13. For $k > 1$, by Corollary 13 we have

$$\mu_k \geq \mu_{k-1} + \lambda_k \cdot \left(1 - e^{\frac{-2\mu^2}{n\cdot\lambda^2}} - \mu_{k-1}\right),$$

which implies that

$$\mu_k \geq (1-\lambda_k)\cdot\mu_{k-1} + \lambda_k \cdot \left(1 - e^{\frac{-2\mu^2}{n\cdot\lambda^2}}\right).$$

Now we can use the induction's hypothesis and replace μ_{k-1} with $\mu + \Big(1 - \prod_{i=1}^{k-1}(1 - \lambda_i)\Big) \cdot \Big(1 - e^{-2\mu^2/(n\cdot\lambda^2)} - \mu\Big)$ which implies that

$$\mu_k \geq \mu + \left(1 - \prod_{i=1}^{k}(1-\lambda_i)\right) \cdot \left(1 - e^{\frac{-2\mu^2}{n\cdot\lambda^2}} - \mu\right),$$

and that proves the claim. □

4 Optimal attacks for uniform binary messages

In this section, we focus on the setting in which n parties each send a uniform random bit and then a final bit is chosen based on the published messages. We will show how to obtain *optimal* online k-replacing attacks that *match* the power of *offline* attacks.

Notation. $\mathbf{u}_{\leq n} \equiv (\mathbf{u}_1 \times \cdots \times \mathbf{u}_n)$ denotes the uniform random variable over $\{0,1\}^n$, where each $\mathbf{u}_i$ is a uniform and independent random bit. In this section, for simplicity we use notation $\mathbf{U}_n$ for this distribution. We will study k-replacing attacks on $\mathbf{U}_n$.[11] $\mathrm{HW}(x) = \mathrm{HD}(x, 0^n)$ denotes Hamming weight of $x \in \{0,1\}^n$. We let

[11] In Sections 2 and 3, we called the original random process $\mathbf{w}_{\leq n}$ and $\mathbf{U}_n$ was one of the generated random processes (modeling the original samples). However, since we are starting from a *product* distribution, it would hold that $\mathbf{U}_n \equiv \mathbf{w}_{\leq n}$, and thus we simply call the original distribution $\mathbf{u}$.

$[n] = \{1,\ldots,n\}$, $\langle n] = \{0,\ldots,n\}$ and $\langle n\rangle = \{0,\ldots,n+1\}$. For $t \in \langle n\rangle$, we define the threshold function $\tau_t \colon \{0,1\}^n \to \{0,1\}$ as $\tau_t(x) = 1$ iff $\mathrm{HW}(x) \geq t$. (τ_0 is the constant function 1 function and τ_{n+1} is the constant 0 function.) We let $\beta_n^{(t)} = 2^{-n} \cdot \sum_{i=t}^{n} \binom{n}{i}$ be the probability of the Hamming ball defined by τ_t, and when n is clear from the context we write it as $\beta^{(t)}$. We also let $s_n^{(t)} = 2^n \cdot \beta_n^{(t)}$ be the size of the same Hamming ball. For set $\mathcal{S} \subset \mathbb{R}$, $r \in \mathbb{R}$, we use the notation $r\mathcal{S} = \{rx \mid x \in \mathcal{S}\}$, e.g., $r\langle n] = \{0, r, 2r, \ldots, nr\}$. We let $\binom{n}{k} = 0$ if $k < 0$ or $k > n$. For a set $\mathcal{S} \subseteq \{0,1\}^n$ and $r \in \{0,1\}^d$ for $d \in [n]$, we let

$$\mathcal{S}[r] = \left\{x' \mid x \in \mathcal{S} \wedge \exists x' \in \{0,1\}^{n-d} \text{ such that } x = (r, x')\right\}$$

be the set of suffixes of strings in $\mathcal{S}$ of length $n-d$ with r as their prefix.

We first define the isoperimetry function that capture the power of “offline” attacks.

Definition 15 (The offline expansion and isoperimetry functions)**.** For $k \in [n], \mathcal{S} \subseteq \{0,1\}^n$, the offline *$k$-expansion* (probability) of $\mathcal{S}$ is the probability of all points within Hamming distance k of $\mathcal{S}$

$$\mathsf{OffExp}^{(k)}(\mathcal{S}) = \frac{|\{y \in \{0,1\}^n \mid \exists x \in \mathcal{S}, \mathrm{HD}(x,y) \leq k\}|}{2^n}.$$

For a given probability μ, the *k-expansion* of μ is equal to:

$$\mathsf{OffExp}^{(k)}(\mu) = \inf_{\mathcal{S}, \Pr[\mathcal{S}] \geq \mu} \mathsf{OffExp}^{(k)}(\mathcal{S}).$$

Finally, for a set $\mathcal{S}$ and probability μ, we define the (offline) *k-isoperimetry* function

$$\mathsf{OffIso}^{(k)}(\mathcal{S}) = \mathsf{OffExp}^{(k)}(\mathcal{S}) - \Pr[\mathcal{S}], \quad \mathsf{OffIso}_n^{(k)}(\mu) = \mathsf{OffExp}_n^{(k)}(\mu) - \mu.$$

Note that whenever the input is a set $\mathcal{S} \subseteq \{0,1\}^n$, it already determines n on its own, and hence we do not need to state it explicitly, but when the input is $\mu \in \mathbb{R}$, we explicitly state n as the index of the function.

Theorem 16 (Implied by the vertex isoperimetric inequality in Boolean hypercube [19])**.** *For any $t \in \langle n]$, it holds that* $\mathsf{OffExp}^{(k)}(\beta^{(t)}) = \beta^{(t-k)}$.

Online attacks vs. offline attacks. Suppose an adversary wants to increase the probability of falling into a set $\mathcal{S}$ in an “offline” attack, in which the adversary gets a point $x \leftarrow \mathbf{U}_n$ and then can replace k of the bits of x. It is easy to see that the adversary can increase the probability of falling into $\mathcal{S}$ exactly by $\mathsf{OffIso}^{(k)}(\mathcal{S})$. Accordingly, we can define the *online* variant of such attacks as defined in Section 4. In such online attacks, the adversary gets to see the independent and uniformly sampled random bits $(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ one by one, and after seeing $u_i \leftarrow \mathbf{u}_i$, it can decide to keep or change it.

Definition 17 (The online expansion OnExp and isoperimetry OnIso functions)**.** Let A be an online adversary of budget k over the uniform distribution $\mathbf{U}_n$ over $\{0,1\}^n$. Let $\mathbf{v}_{\leq n}$ be the generated output random process (distributed over $\{0,1\}^n$) under attack A (as defined in Definition 4). We define $\mathsf{OnExp}^{(\mathsf{A})}(\mathcal{S}) = \Pr[\mathbf{v}_{\leq n} \in \mathcal{S}]$. Let $\mathcal{A}_k$ be the set of *all* k-replacing attacks on $\mathbf{U}_n$. We define $\mathsf{OnExp}^{(k)}(\mathcal{S})$ as the maximum probability of points in $\{0,1\}^n$ that any online adversary can map to $\mathcal{S}$ by up to k changes to a stream of n uniformly random bits. Namely,

$$\mathsf{OnExp}^{(k)}(\mathcal{S}) = \max_{\mathsf{A}\in\mathcal{A}_k} \mathsf{OnExp}^{(\mathsf{A})}(\mathcal{S}).$$

Also, for any $\mu \in [0,1]$, we define

$$\mathsf{OnExp}_n^{(k)}(\mu) = \inf_{\mathcal{S},\Pr[\mathcal{S}]\geq\mu} \mathsf{OnExp}^{(k)}(\mathcal{S})$$

as the minimum $\mathsf{OnExp}^{(k)}(\mathcal{S})$ among all sets of probability at least μ. Finally, for any set $\mathcal{S}$ and probability μ, we define the *online k-isoperimetry* functions as follows

$$\mathsf{OnIso}^{(k)}(\mathcal{S}) = \mathsf{OnExp}^{(k)}(\mathcal{S}) - \Pr[\mathcal{S}], \quad \mathsf{OnIso}_n^{(k)}(\mu) = \mathsf{OnExp}_n^{(k)}(\mu) - \mu$$

as the *growth* in probability of falling into sets (of probability μ) under optimal online k-replacing attacks.

Since offline adversaries know as much as online adversaries when making decision to change or not, it always holds that $\mathsf{OffIso}(\mathcal{S}) \geq \mathsf{OnIso}(\mathcal{S})$, and hence $\mathsf{OffIso}_n^{(k)}(\mu) \geq \mathsf{OnIso}_n(\mu)$ for all n, $\mathcal{S} \subseteq \{0,1\}^n$, and $\mu \in [0,1]$. The surprising phenomenon stated in the next theorem is that when μ is the probability of a Hamming ball, online and offline attacks have the *same exact* power as a function of the measure μ, and consequently the online and offline k-isoperimetry functions would be equal.

Theorem 18 (Power of online vs. offline attacks for the uniform distribution over $\{0,1\}^n$)**.** *For all $n \in \mathbb{N}, t \in [n], k \leq t$, if $\beta^{(t)} = \Pr[\mathrm{HW}(\mathbf{U}_n) \geq t]$ be the probability of a Hamming ball. Then it holds that*

$$\mathsf{OnExp}_n^{(k)}(\beta^{(t)}) = \mathsf{OffExp}_n^{(k)}(\beta^{(t)}) = \beta^{(t-k)}.$$

In words, if $\mu = \beta^{(t)}$, then the power of online k-replacing adversaries to increase the probability of falling into a set $\mathcal{S}$, in the minimum over all sets of probability at least $\beta^{(t)}$, is equal to that of offline attacks.

Reaching a target probability. Suppose $\Pr[\mathcal{S}] = \mu$, and suppose we want to increase the probability of falling into $\mathcal{S}$ to $\mu' > \mu$. How much budget an adversary needs? Theorem 18 shows that as long as μ is the probability of a Hamming ball (i.e., $\mu = \beta^{(t)}$), then in the worst case (among all possible sets $\mathcal{S}$ of probability μ) the power of online and offline attacks are *exactly the same*. Therefore, this brings up the natural question of what happens in general, when μ is *not* exactly the probability of a Hamming ball. As stated in Corollary 19 below, Theorem 18 already shows that the power of offline and online attacks is different by at most one. In fact, as we will see later, these quantities are not equal in general. In particular, Figure 1 compares $\mathsf{OnIso}_n(\mu)$ and $\mathsf{OffIso}_n^{(k)}(\mu)$ for all μ when $n = 10$ (and $k = 1$).

Corollary 19 (Budget of online vs. offline attacks to reach a target probability)**.** *For* $0 < \mu < \mu' \leq 1$*, let*

$$\mathsf{OfBud}_n(\mu \to \mu') = \min_{k \in [n]} [\mathsf{OffExp}_n^{(k)}(\mu) \geq \mu']$$

be the minimum budget k *that an offline adversary needs to increase the probability of falling into any set* $\mathcal{S}$ *of probability at least* μ *to* μ'*. Let*

$$\mathsf{OnBud}_n(\mu \to \mu') = \min_{k \in [n]} [\mathsf{OnExp}_n^{(k)}(\mu) \geq \mu']$$

be the similar quantity for online attacks. Then, it always holds that

$$\mathsf{OfBud}_n(\mu \to \mu') \leq \mathsf{OnBud}_n(\mu \to \mu') \leq \mathsf{OfBud}_n(\mu \to \mu') + 1$$

and $\mathsf{OfBud}_n(\beta^{(t)} \to \mu') = \mathsf{OnBud}_n(\beta^{(t)} \to \mu')$ *for all* $t \in [n+1]$*.*

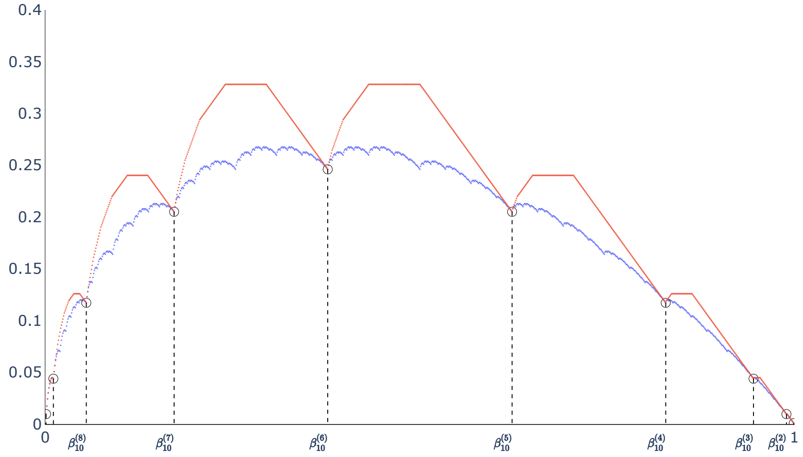

Fig. 1. Comparing the online isoperimetric function OnIso (blue) versus the offline isoperimetric function OffIso (red) for $n = 10$.

We first prove Corollary 19 using Theorem 18 and then will prove Theorem 18.

Proof of Corollary 19. Let $k = \mathsf{OfBud}_n(\mu \to \mu')$, we have $\mathsf{OffExp}_n^{(k)}(\mu) \geq \mu'$ and $\mathsf{OffExp}_n^{(k-1)}(\mu) < \mu'$. Let $t \in \langle n \rangle$ be the minimum such t that $\beta^{(t)} \geq \mu$, and so we have $\beta^{(t+1)} \leq \mu \leq \beta^{(t)}$. By the monotonicity of $\mathsf{OnExp}_n^{(k)}$ function, we have

$$\mathsf{OnExp}_n^{(k+1)}(\beta^{(t+1)}) \leq \mathsf{OnExp}_n^{(k+1)}(\mu). \tag{12}$$

By Theorem 18 it holds that $\mathsf{OnExp}_n^{(k+1)}(\beta^{(t+1)}) = \mathsf{OffExp}_n^{(k+1)}(\beta^{(t+1)})$. Now, because $\beta^{(t+1)} \leq \mu \leq \beta^{(t)}$, by the monotonicity of $\mathsf{OffExp}_n^{(k)}(\mu)$ we have

$$\mathsf{OffExp}_n^{(k)}(\mu) \leq \mathsf{OffExp}_n^{(k)}(\beta^{(t)}) = \mathsf{OffExp}_n^{(k+1)}(\beta^{(t+1)}) = \mathsf{OnExp}_n^{(k+1)}(\beta^{(t+1)}). \tag{13}$$

Combining (12) and (13), we have

$$\mathsf{OffExp}_n^{(k)}(\mu) \leq \mathsf{OnExp}_n^{(k+1)}(\beta^{(t+1)}) \leq \mathsf{OnExp}_n^{(k+1)}(\mu).$$

Therefore we have,

$$\begin{aligned}\mathsf{OfBud}_n(\mu \to \mu') + 1 &= \min_{k\in[n]}[\mathsf{OffExp}_n^{(k)}(\mu) \geq \mu'] + 1\\ &\geq \min_{k\in[n]}[\mathsf{OnExp}_n^{(k+1)}(\mu) \geq \mu'] + 1\\ &= \min_{k+1\in[n]}[\mathsf{OnExp}_n^{(k+1)}(\mu) \geq \mu']\\ &= \mathsf{OnBud}_n(\mu \to \mu').\end{aligned}$$

The inequality holds because let $k' = \min_{k\in[n]}[\mathsf{OffExp}_n^{(k)}(\mu) \geq \mu']$, we have $\mathsf{OnExp}_n^{(k'+1)}(\mu) \geq \mu'$, and therefore $\min_{k\in[n]}[\mathsf{OnExp}_n^{(k+1)}(\mu) \geq \mu'] \leq k'$. Since we also have $\mathsf{OffExp}_n^{(k)}(\mu) \geq \mathsf{OnExp}_n^{(k)}(\mu)$ for any μ, $\mathsf{OfBud}_n(\mu \to \mu') \leq \mathsf{OnBud}_n(\mu \to \mu') \leq \mathsf{OfBud}_n(\mu \to \mu') + 1$.

Finally, because $\mathsf{OffExp}_n^{(k)}(\beta^{(t)}) \geq \mathsf{OnExp}_n^{(k)}(\beta^{(t)})$ holds for any k and t, we have $\mathsf{OfBud}_n(\beta^{(t)} \to \mu') = \mathsf{OnBud}_n(\beta^{(t)} \to \mu')$ for all $t \in [n+1]$. □

In the rest of this section, we prove Theorem 18.

Proof of Theorem 18. In order to prove Theorem 18, we start by deriving a recursive relation for $\mathsf{OnExp}_n^{(k)}(\cdot)$. Before doing so, we define some mathematical notation.

Definition 20 (Definitions related to the recursive relation of online expansion)**.** For $s \in \langle 2^n]$, let

$$\mathcal{D}iv_{n-1}(s) = \left\{(s_0, s_1) \mid s_0, s_1 \in \langle 2^{n-1}], 0 \leq s_0 \leq s_1 \leq 2^{n-1}, s = s_0 + s_1\right\}$$

be the set of ways in which a "set size" $s \in \langle 2^{n-1}]$ can be divided into two sizes. For $(s_0, s_1) \in \mathcal{D}iv_{n-1}(s)$ and a *fixed* pair of integers n, k let $\mathsf{Rec}_n^{(k)}(\cdot,\cdot)$ be defined as

$$\mathsf{Rec}_n^{(k)}(s_0, s_1) = \frac{\mathsf{Rec}_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\mathsf{Rec}_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \mathsf{Rec}_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \tag{14}$$

based on functions $\mathsf{Rec}_{n-1}^{(k)}, \mathsf{Rec}_{n-1}^{(k-1)}$ to be specified later. Finally, for $\mu \in 2^{-n}\langle 2^n]$ let

$$\mathsf{Rec}_n^{(k)}(\mu) = \inf_{(s_0,s_1)\in\mathcal{D}iv_n(2^n\cdot\mu)} \mathsf{Rec}_n^{(k)}(s_0, s_1). \tag{15}$$

Transformation $\mathsf{Rec}_n^{(k)}[p, q]$. For functions p, q defined on input space $2^{-n}\langle 2^n]$. Suppose we use p instead of $\mathsf{Rec}_{n-1}^{(k)}$ and q instead of $\mathsf{Rec}_{n-1}^{(k-1)}$ in Equation (14). Then by $\mathsf{Rec}_n^{(k)}[p, q](\cdot,\cdot)$ (resp. $\mathsf{Rec}_n^{(k)}[p, q](\cdot)$) we denote the function that one obtains in Equation (14) (resp. Equation (15)).

Interpretation. $\mathsf{Rec}_n^{(k)}(s_0, s_1)$ represents the optimal choice that a tampering adversary can make to increase the probability of falling into a set of size s, when $\mathcal{S}[0] = \mathcal{S}_0, \mathcal{S}[1] = \mathcal{S}_1$ are adversarially chosen based on their sizes s_0, s_1 where $s_0 \leq s_1$, and when the optimal online expansions for s_0, s_1 can be applied by (appropriate use of) functions $\mathsf{Rec}_{n-1}^{(k)}, \mathsf{Rec}_{n-1}^{(k-1)}$.

Notation. Let f, g be defined over the same input domain $\mathcal{D}$. We say $f \leq g$, if $\forall \mu \in \mathcal{D}, f(\mu) \leq g(\mu)$.

We now show that the transformation of Definition 20 has some desired properties.

Claim 21 (Transformation of Definition 20 is monotone)**.** *Let* $\mathsf{u}_{n-1}^{(k)} \leq \mathsf{v}_{n-1}^{(k)}$ *and* $\mathsf{u}_{n-1}^{(k-1)} \leq \mathsf{v}_{n-1}^{(k-1)}$*, and let*

$$\mathsf{u}_n^{(k)} = \mathsf{Rec}_n^{(k)}[\mathsf{u}_{n-1}^{(k)}, \mathsf{u}_{n-1}^{(k-1)}], \quad \mathsf{v}_n^{(k)} = \mathsf{Rec}_n^{(k)}[\mathsf{v}_{n-1}^{(k)}, \mathsf{v}_{n-1}^{(k-1)}]$$

as defined in Definition 20. Then, it holds that $\mathsf{u}_n^{(k)} \leq \mathsf{v}_n^{(k)}$.

Proof. We first show that for any $s_0, s_1 \in \mathcal{D}iv_n(2^n \cdot \mu)$, we have $\mathsf{u}_n^{(k)}(s_0, s_1) \leq \mathsf{v}_n^{(k)}(s_0, s_1)$. Because $\mathsf{u}_{n-1}^{(k)} \leq \mathsf{v}_{n-1}^{(k)}$ and $\mathsf{u}_{n-1}^{(k-1)} \leq \mathsf{v}_{n-1}^{(k-1)}$, we have $\mathsf{u}_{n-1}^{(k)}\left(s_1/2^{n-1}\right) \leq \mathsf{v}_{n-1}^{(k)}\left(s_1/2^{n-1}\right)$ and

$$\max\{\mathsf{u}_{n-1}^{(k)}\left(s_0/2^{n-1}\right), \mathsf{u}_{n-1}^{(k)}\left(s_1/2^{n-1}\right)\} \leq \max\{\mathsf{v}_{n-1}^{(k)}\left(s_0/2^{n-1}\right), \mathsf{v}_{n-1}^{(k)}\left(s_1/2^{n-1}\right)\}.$$

Therefore, $\mathsf{u}_n^{(k)}(s_0, s_1) \leq \mathsf{v}_n^{(k)}(s_0, s_1)$ holds for any s_0, s_1.

From Eq. (15), let $(s_0', s_1') = \arg\inf_{(s_0,s_1)\in\mathcal{D}iv_n(2^n\cdot\mu)} \mathsf{v}_n^{(k)}(s_0, s_1)$ be the partition where $\mathsf{v}_n^{(k)}(\mu)$ achieves its minimum. Then we have $\mathsf{u}_n^{(k)}(\mu) \leq \mathsf{u}_n^{(k)}(s_0', s_1') \leq \mathsf{v}_n^{(k)}(s_0', s_1') = \mathsf{v}_n^{(k)}(\mu)$. □

Claim 22 (Recursive relation for online expansion)**.** *One can recursively compute* $\mathsf{OnExp}_n^{(k)}(\mu)$ *for all* $\mu \in 2^{-n}\langle 2^n]$ *as follows.*

- *If* $k = 0$ *and* $n \geq 0$*, then* $\mathsf{OnExp}_n^{(0)}(\mu) = \mu$.
- *If* $k \geq 1$ *and* $k \geq n$*, then:* $\mathsf{OnExp}_n^{(k)}(0) = 0$ *and* $\mathsf{OnExp}_n^{(k)}(\mu) = 1$ *for* $\mu > 0$.
- *If* $k \geq 1$ *and and* $k < n$*, then* $\mathsf{OnExp}_n^{(k)} = \mathsf{Rec}_n^{(k)}[\mathsf{OnExp}_{n-1}^{(k)}, \mathsf{OnExp}_{n-1}^{(k-1)}]$ *as in Definition 20.*

Proof sketch. The extremal cases of the recursive relation stated in the first two bullets hold trivially. Below we argue why the inductive step as stated in the third bullet holds as well.

Suppose by fixing the first bit to b we get a subset of size s_b, and $s_0 \leq s_1$, and suppose in both cases the residual subsets $\mathcal{S}[0], \mathcal{S}[1]$ are chosen in the "worst" case (against the adversary) based on their sizes s_0, s_1, minimizing the success probability of an online adversary. Since $\mathsf{OnExp}(\cdot)$ is a monotone function, then when the first bit is selected to be 1, the adversary has no motivation to replace it with 0. When the first bit is selected to be 0, the adversary has choose between maximum of the expansions that arise from changing or not changing the bit to 1. Once we consider all ways that s can be split into $s = s_0 + s_1$, this leads to the definition of the recursion of Eq. (15) and the transformation of Definition 20. □

Claim 23. *Suppose $p \le \mathsf{OnExp}^{(k)}_{n-1}, q \le \mathsf{OnExp}^{(k-1)}_{n-1}$ for functions p, q. Then, it holds that $\mathsf{Rec}^{(k)}_n[p, q] \le \mathsf{OnExp}^{(k)}_n$ (see Definition 20).*

Proof. The proof directly follows from Claims 22 and 21. □

We now define a piecewise-linear function $\ell^{(k)}_n$ to later prove to be a lower bound for $\mathsf{OnExp}^{(k)}_n$.

Definition 24 (The piecewise-linear (lower bound) function)**.** For any non-negative integers k, n, the function $\ell^{(k)}_n : [0, 1] \to [0, 1]$ is defined as follows.

- If $\mu = \beta^{(t)}_n$ for any $t \in \langle n \rangle$, it holds that $\ell^{(k)}_n\left(\beta^{(t)}_n\right) = \mathsf{OffExp}^{(k)}_n\left(\beta^{(t)}_n\right)$. Namely, $\ell^{(k)}_n\left(\beta^{(n+1)}_n\right) = \mathsf{OffExp}^{(k)}_n(0) = 0$, and for any $t \in \langle n]$, $\ell^{(k)}_n\left(\beta^{(t)}_n\right) = \beta^{(t-k)}_n = \Pr\left[\mathrm{HW}(\mathbf{U}_n) \ge t - k\right]$.
- If $\mu = \alpha\beta^{(t)}_n + (1-\alpha)\beta^{(t-1)}_n$ for $0 < \alpha < 1$ and any $t \in \langle n \rangle$, then $\ell^{(k)}_n(\mu) = \alpha \cdot \ell^{(k)}_n\left(\beta^{(t)}_n\right) + (1-\alpha) \cdot \ell^{(k)}_n\left(\beta^{(t-1)}_n\right)$.

Proposition 25 (Composition of the lower bound function)**.** *For any $k_1, k_2, n \ge 0$ and $\mu \in [2^{-n}, 1]$, it hold that $\ell^{(k_1+k_2)}_n(\mu) = \ell^{(k_1)}_n\left(\ell^{(k_2)}_n(\mu)\right)$.*

Proof. Consider every case,

- If $\mu = \beta^{(t)}_n$. By Definition 24 we have $\ell^{(k_1)}_n\left(\ell^{(k_2)}_n(\mu)\right) = \ell^{(k_1)}_n\left(\mathsf{OffExp}^{(k_2)}_n\left(\beta^{(t)}_n\right)\right)$. As $\mu \in [2^{-n}, 1]$, we have $t \le n$. Therefore, $\mathsf{OffExp}^{(k_2)}_n\left(\beta^{(t)}_n\right) = \beta^{(t-k_2)}_n$. Therefore, we have
$$\ell^{(k_1)}_n\left(\ell^{(k_2)}_n\left(\beta^{(t)}_n\right)\right) = \ell^{(k_1)}_n\left(\beta^{(t-k_2)}_n\right) = \beta^{(t-(k_2+k_1))}_n = \ell^{(k_1+k_2)}_n\left(\beta^{(t)}_n\right).$$
- If $\mu = \alpha\beta^{(t)}_n + (1-\alpha)\beta^{(t-1)}_n$ for $0 < \alpha < 1$, In this case, by Definition 24 we have $\ell^{(k_2)}_n(\mu) = \alpha \cdot \ell^{(k_2)}_n\left(\beta^{(t)}_n\right) + (1-\alpha) \cdot \ell^{(k_2)}_n\left(\beta^{(t-1)}_n\right)$. As $\mu \in [2^{-n}, 1]$, we have $t \le n$. Therefore, we have $\ell^{(k_2)}_n(\mu) = \alpha \cdot \beta^{(t-k_2)}_n + (1-\alpha) \cdot \beta^{(t-1-k_2)}_n$. We then have
$$\begin{aligned}\ell^{(k_1)}_n\left(\ell^{(k_2)}_n(\mu)\right) &= \ell^{(k_1)}_n\left(\alpha \cdot \beta^{(t-k_2)}_n + (1-\alpha) \cdot \beta^{(t-1-k_2)}_n\right)\\ &= \alpha \cdot \beta^{(t-k_2-k_1)}_n + (1-\alpha) \cdot \beta^{(t-1-k_2-k_1)}_n\\ &= \ell^{(k_1+k_2)}_n(\mu).\end{aligned}$$

□

Lemma 26. *$\ell^{(k)}_n$ is concave for all $n, k \ge 0$.*

Proof. $\ell^{(0)}_n$ is linear, and hence concave, so suppose $k \ge 1$. Let fix n, and define $\hat{\ell}(\mu) = \ell^{(1)}_n(\mu) - \mu$ for $\mu \in [0, 1]$. To prove that $\ell^{(k)}_n(\mu)$ is concave over $[2^{-n}, 1]$, it is sufficient to show that $\hat{\ell}(\mu)$ is concave over $[2^{-n}, 1]$, because:

1. If $\hat{\ell}(\mu)$ is concave, then $\hat{\ell}(\mu) + \mu = \ell_n^{(1)}(\mu)$ is concave as well.
2. If $\ell_n^{(1)}(\mu)$ is concave, since it is non-decreasing, by repeated applications of Lemma 8 and Proposition 25, it follows that $\ell_n^{(k)}$ is also concave for all $k \geq 1$ as well, when we limit the inputs to $\mu \geq 2^{-n}$.

Therefore, in the following, we only aim to prove that (1) $\hat{\ell}(\mu)$ is concave over $[2^{-n}, 1]$, and (2) the left and right derivatives of $\hat{\ell}(\mu)$ over $\mu = 2^{-n}$ do not violate its concavity.

In the following, we will fix n and $k = 1$. Because n, k are both fixed, in the rest of the proof of Lemma 26 we do not represent them explicitly as indexes anymore.

It holds that $\hat{\ell}(\beta^{(t)}) = \mathsf{OffIso}(\beta^{(t)})$ for all $t \in \langle n \rangle$. Also, for $\mu \in (\beta^{(t)}, \beta^{(t-1)})$ (recall that $\beta^{(t)} < \beta^{(t-1)}$) where $\mu = \alpha\beta^{(t)} + (1-\alpha)\beta^{(t-1)}$, we have

$$\begin{aligned}\hat{\ell}(\mu) &= \alpha\ell(\beta^{(t)}) + (1-\alpha)\ell(\beta^{(t-1)}) - \alpha\beta^{(t)} - (1-\alpha)\beta^{(t-1)} \\ &= \alpha\mathsf{OffIso}(\beta^{(t)}) + (1-\alpha)\mathsf{OffIso}(\beta^{(t-1)}).\end{aligned}$$

Since the curve $\hat{\ell}$ is linear over every interval $\mu \in [\beta^{(t)}, \beta^{(t-1)}]$ for all $t \in [n+1]$, to prove its concavity, we only have to compare its left and right derivatives at every $\beta^{(t)}, t \in [n]$, where it holds that $\hat{\ell}(\beta^{(t)}) = \mathsf{OffIso}(\beta^{(t)})$. Hence, for all $t \in [n]$, we need to prove the following.

$$\frac{\mathsf{OffIso}(\beta^{(t)}) - \mathsf{OffIso}(\beta^{(t+1)})}{\beta^{(t)} - \beta^{(t+1)}} \geq \frac{\mathsf{OffIso}(\beta^{(t-1)}) - \mathsf{OffIso}(\beta^{(t)})}{\beta^{(t-1)} - \beta^{(t)}} \tag{16}$$

Note that by letting $t = n$ in Inequality (16), we have $\hat{\ell}$ is still concave for point 2^{-n}. We first verify Inequality (16) for extreme cases of $t = 1, n$. If $t = 1$, then Inequality (16) holds because

$$\frac{1-n}{n} = \frac{\mathsf{OffIso}(\beta^{(t)}) - \mathsf{OffIso}(\beta^{(t+1)})}{\beta^{(t)} - \beta^{(t+1)}} \geq \frac{\mathsf{OffIso}(\beta^{(t-1)}) - \mathsf{OffIso}(\beta^{(t)})}{\beta^{(t-1)} - \beta^{(t)}} = \frac{0-1}{1}.$$

If $t = n$, a generalization of Inequality 16 for any k holds because

$$\frac{\sum_{i=0}^{k}\binom{n}{i} - 0}{1-0} = \frac{\ell^{(k)}(\beta^{(t)}) - \ell^{(k)}(\beta^{(t+1)})}{\beta^{(t)} - \beta^{(t+1)}} \geq \frac{\ell^{(k)}(\beta^{(t-1)}) - \ell^{(k)}(\beta^{(t)})}{\beta^{(t-1)} - \beta^{(t)}} = \frac{\binom{n}{k+1}}{n},$$

which in turn is correct because $\sum_{i=0}^{k}\binom{n}{i} > \binom{n}{k} \geq \binom{n}{k+1}/n$.

For the intermediate cases, for all $t \in \{n-1, \ldots, 2\}$, we have to prove:

$$\begin{aligned}\frac{\binom{n}{t-k} - \binom{n}{t}}{\binom{n}{t}} &= \frac{\mathsf{OffIso}(\beta^{(t)}) - \mathsf{OffIso}(\beta^{(t+1)})}{\beta^{(t)} - \beta^{(t+1)}} \\ &\geq \frac{\mathsf{OffIso}(\beta^{(t-1)}) - \mathsf{OffIso}(\beta^{(t)})}{\beta^{(t-1)} - \beta^{(t)}} = \frac{\binom{n}{t-2} - \binom{n}{t-1}}{\binom{n}{t-1}}\end{aligned}$$

which is equivalent to proving the following true statement

$$\begin{aligned}\frac{t}{n-t+1} = \frac{t!(n-t)!}{(t-1)!(n-t+1)!} &= \frac{\binom{n}{t-1}}{\binom{n}{t}} \geq \frac{\binom{n}{t-2}}{\binom{n}{t-1}} \\ &= \frac{(t-1)!(n-t+1)!}{(t-2)!(n-t+2)!} = \frac{t-1}{n-t+2}.\end{aligned}$$

□

The main step of the proof of Theorem 18 is to show the following claim.

Claim 27. *It holds that* $\ell_n^{(k)} \leq \mathsf{Rec}_n^{(k)}\left[\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}\right]$.

Proof. In the following, for simplicity we let $\mathsf{Rec} = \mathsf{Rec}_n^{(k)}\left[\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}\right]$.

Case of exact Hamming ball probabilities. We first prove

$$\forall t \in \langle n \rangle,\ \ell_n^{(k)}(\beta^{(t)}) \leq \mathsf{Rec}(\beta^{(t)}) \tag{17}$$

and then will extend the proof of this inequality to an arbitrary $\mu \in 2^{-n}\langle 2^n]$. We only need to prove Inequality 17 for $t \in [n]$, because $\beta_n^{(n+1)} = 0, \beta^{(0)} = 1$, and so

$$\ell_n^{(k)}(0) = \mathsf{Rec}(0) = 0, \quad \ell_n^{(k)}(1) = \mathsf{Rec}(1) = 1.$$

Recall that $\ell_n^{(k)}\left(\beta_n^{(t)}\right) = \mathsf{OffExp}_n^{(k)}\left(\beta_n^{(t)}\right) = \beta_n^{(t-k)}$. Hence, for $s = s_n^{(t)} = \beta^{(t)} \cdot 2^n$ where $t \in [n]$, our goal is to prove the following

$$\beta_n^{(t-k)} \leq \inf_{(s_0,s_1)\in \mathcal{D}iv_n(s)} \mathsf{Rec}(s_0, s_1). \tag{18}$$

Case studies. Note that $\beta_n^{(t)}2^n = s_n^{(t)} = s_{n-1}^{(t)} + s_{n-1}^{(t-1)}$ because of the Pascal equality. Also by the definition of $\mathcal{D}iv_n$, we have $s_0 \leq s_1$ and $s_0 + s_1 = s_n^{(t)}$ for any choice of $(s_0, s_1) \in \mathcal{D}iv_n(s)$ in the right hand side of Equation 18. Then, one of the following three cases must hold: (1) $s_0 = s_{n-1}^{(t)}$, (2) $s_0 < s_{n-1}^{(t)}$, or (3) $s_0 > s_{n-1}^{(t)}$. Hence, we divide our analysis to the same three cases, and then prove that $\beta_n^{(t-k)} \leq \mathsf{Rec}(s_0, s_1)$ holds in all of them. We will also use the Pascal equality in the form of $\beta_n^{(t-k)} = (\beta_{n-1}^{(t-k-1)} + \beta_{n-1}^{(t-k)})/2$.

1. $s_{n-1}^{(t)} = s_0 < s_1 = s_{n-1}^{(t-1)}$. In this case, we have

$$\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right) = \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right) = \beta_{n-1}^{(t-k)}$$

   which, informally speaking means that, it does not matter if the adversary intervenes to change 0 to 1 when the first bit is fixed to 0. Formally, we have

$$\begin{aligned}\mathsf{Rec}(s_0, s_1) &= \mathsf{Rec}(s_{n-1}^{(t)}, s_{n-1}^{(t-1)}) \\ &= \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\ &= \frac{\beta_{n-1}^{(t-k-1)} + \beta_{n-1}^{(t-k)}}{2} = \beta_n^{(t-k)}.\end{aligned}$$

2. $s^{(t)}_{n-1} < s_0 \le s_1 < s^{(t-1)}_{n-1}$. Informally speaking, in this case the adversary does not change the bit and we use the piece-wise linearity of the ℓ function on $[\beta^{(t)}_{n-1}, \beta^{(t-1)}_{n-1}]$. More formally,

$$\begin{aligned}
\mathsf{Rec}(s_0, s_1) &= \frac{\ell^{(k)}_{n-1}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell^{(k)}_{n-1}\left(\frac{s_0}{2^{n-1}}\right), \ell^{(k-1)}_{n-1}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\
&\ge \frac{\ell^{(k)}_{n-1}\left(\frac{s_1}{2^{n-1}}\right) + \ell^{(k)}_{n-1}\left(\frac{s_0}{2^{n-1}}\right)}{2} \\
&= \frac{\ell^{(k)}_{n-1}\left(\frac{s^{(t-1)}_{n-1}}{2^{n-1}}\right) + \ell^{(k)}_{n-1}\left(\frac{s^{(t)}_{n-1}}{2^{n-1}}\right)}{2} \text{ (by piece-wise linearity of } \ell^{(k)}_{n-1}) \\
&= \mathsf{Rec}\left(s^{(t)}_{n-1}, s^{(t-1)}_{n-1}\right) = \beta^{(t-k)}_n.
\end{aligned}$$

3. $s_0 < s^{(t)}_{n-1} < s^{(t-1)}_{n-1} < s_1$. Informally speaking, in this case the adversary does change the bit 0 into 1, and we also use the fact that $\ell^{(k)}_{n-1}$ is monotone. More formally,

$$\begin{aligned}
\mathsf{Rec}(s_0, s_1) &= \frac{\ell^{(k)}_{n-1}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell^{(k)}_{n-1}\left(\frac{s_0}{2^{n-1}}\right), \ell^{(k-1)}_{n-1}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\
&\ge \frac{\ell^{(k)}_{n-1}\left(\frac{s_1}{2^{n-1}}\right) + \ell^{(k-1)}_{n-1}\left(\frac{s_1}{2^{n-1}}\right)}{2} \\
&\ge \frac{\ell^{(k)}_{n-1}\left(\frac{s^{(t-1)}_{n-1}}{2^{n-1}}\right) + \ell^{(k-1)}_{n-1}\left(\frac{s^{(t-1)}_{n-1}}{2^{n-1}}\right)}{2} \text{ (by monotonicity of } \ell^{(k)}_{n-1}) \\
&= \mathsf{Rec}\left(s^{(t)}_{n-1}, s^{(t-1)}_{n-1}\right) = \beta^{(t-k)}_n.
\end{aligned}$$

Case of other probabilities. Here we no longer assume that $\mu = \beta^{(t)}$ for some $t \in [n]$, and assume $\mu = \alpha\beta^{(t)}_n + (1-\alpha)\beta^{(t-1)}_n$ for some $t \in [n+1]$ and $0 < \alpha < 1$. Recall that $\beta^{(t)}2^n = s^{(t)}_n = s^{(t)}_{n-1} + s^{(t-1)}_{n-1}$ and $\beta^{(t-1)}2^n = s^{(t-1)}_n = s^{(t-1)}_{n-1} + s^{(t-2)}_{n-1}$. We define

$$s'_0 = \alpha \cdot s^{(t)}_{n-1} + (1-\alpha) \cdot s^{(t-1)}_{n-1}, \quad s'_1 = \alpha \cdot s^{(t-1)}_{n-1} + (1-\alpha) \cdot s^{(t-2)}_{n-1}.$$

By the definition of μ, it holds that $\mu \cdot 2^n = s = s'_0 + s'_1$ because

$$s'_0 + s'_1 = \alpha \cdot \left(s^{(t)}_{n-1} + s^{(t-1)}_{n-1}\right) + (1-\alpha) \cdot \left(s^{(t-1)}_{n-1} + s^{(t-2)}_{n-1}\right) = \alpha \cdot s^{(t)}_n + (1-\alpha) \cdot s^{(t-1)}_n = s.$$

In general, s'_0, s'_1 are not integers, but intuitively, $s'_0 + s'_1$ gives the critical way of splitting s into two numbers at which the replacing and no-replacing strategies give the same bound and we can do the case studies. (In particular s'_0, s'_1 take the role of $s^{(t)}_{n-1}, s^{(t-1)}_{n-1}$ when we previously assumed that $\mu = \beta^{(t)}$.)

Useful observations. By the piecewise linearity of $\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}$ we have

$$\ell_{n-1}^{(k)}\left(\frac{s'_0}{2^{n-1}}\right) = \alpha\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t)}}{2^{n-1}}\right) + (1-\alpha)\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right),$$

$$\ell_{n-1}^{(k)}\left(\frac{s'_1}{2^{n-1}}\right) = \alpha\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right) + (1-\alpha)\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-2)}}{2^{n-1}}\right),$$

$$\ell_{n-1}^{(k-1)}\left(\frac{s'_1}{2^{n-1}}\right) = \alpha\ell_{n-1}^{(k-1)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right) + (1-\alpha)\ell_{n-1}^{(k-1)}\left(\frac{s_{n-1}^{(t-2)}}{2^{n-1}}\right),$$

$$\ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t)}}{2^{n-1}}\right) = \beta_{n-1}^{(t-k)} = \ell_{n-1}^{(k-1)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right),$$

$$\text{and} \quad \ell_{n-1}^{(k)}\left(\frac{s_{n-1}^{(t-1)}}{2^{n-1}}\right) = \beta_{n-1}^{(t-k-1)} = \ell_{n-1}^{(k-1)}\left(\frac{s_{n-1}^{(t-2)}}{2^{n-1}}\right).$$

Therefore, we get the following.

$$\ell_{n-1}^{(k)}\left(\frac{s'_0}{2^{n-1}}\right) = \ell_{n-1}^{(k-1)}\left(\frac{s'_1}{2^{n-1}}\right) = \alpha\beta_{n-1}^{(t-k)} + (1-\alpha)\beta_{n-1}^{(t-k-1)}, \tag{19}$$

$$\ell_{n-1}^{(k)}\left(\frac{s'_1}{2^{n-1}}\right) = \alpha\beta_{n-1}^{(t-k-1)} + (1-\alpha)\beta_{n-1}^{(t-k-2)}. \tag{20}$$

Case studies. We now again partition into three different categories and separately prove that $\ell_n^{(k)}(\mu) \leq \mathsf{Rec}(s_0, s_1)$ holds for each category.

1. $s'_0 = s_0 < s_1 = s'_1$. In this case, using Equations (19) and (20) we get

$$\begin{aligned}
&\mathsf{Rec}(s'_0, s'_1) \\
&= \frac{\ell_{n-1}^{(k)}\left(\frac{s'_1}{2^{n-1}}\right)}{2} + \frac{\max\left\{\ell_{n-1}^{(k)}\left(\frac{s'_0}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s'_1}{2^{n-1}}\right)\right\}}{2} \\
&= \frac{\alpha\cdot\beta_{n-1}^{(t-k-1)} + (1-\alpha)\cdot\beta_{n-1}^{(t-k-2)}}{2} + \frac{\alpha\cdot\beta_{n-1}^{(t-k)} + (1-\alpha)\cdot\beta_{n-1}^{(t-k-1)}}{2} \\
&= \alpha\cdot\beta_n^{(t-k)} + (1-\alpha)\cdot\beta_n^{(t-k-1)} \\
&= \alpha\cdot\ell_n^{(k)}(\beta^{(t)}) + (1-\alpha)\cdot\ell_n^{(k)}(\beta^{(t-1)}) = \ell_n^{(k)}(\mu).
\end{aligned}$$

2. $s'_0 < s_0 \leq s_1 < s'_1$. Informally speaking, in this case the adversary does not tamper and leave the bit 0 unchanged. We will use the fact that $\ell_{n-1}^{(k)}$ is *concave*, which was proved in Lemma 26. Note that in the corresponding Case 2 when the probability μ was that of an exact ball ($\mu = \beta_n^{(t)}$) we could have also used the fact that $\ell_{n-1}^{(k)}$ is concave, but in that case we only used the concavity over a linear

part of $\ell_{n-1}^{(k)}$. However, in our current case, we could no longer only rely on the piecewise linearity of $\ell_{n-1}^{(k)}$ and we would use its concavity. More formally,

$$\begin{aligned}\mathsf{Rec}(s_0, s_1) &= \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\ &\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right)}{2} \\ &\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s_1'}{2^{n-1}}\right) + \ell_{n-1}^{(k)}\left(\frac{s_0'}{2^{n-1}}\right)}{2} \text{ (by \textbf{concavity} of } \ell_{n-1}^{(k)}) \\ &= \mathsf{Rec}(s_0', s_1') = \ell_n^{(k)}(\mu).\end{aligned}$$

3. $s_0 < s_0' < s_1' < s_1$. Informally speaking, in this case the adversary does change the bit 0 into 1, and we rely on the monotonicity of $\ell_{n-1}^{(k)}$. More formally,

$$\begin{aligned}\mathsf{Rec}(s_0, s_1) &= \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \max\left\{\ell_{n-1}^{(k)}\left(\frac{s_0}{2^{n-1}}\right), \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)\right\}}{2} \\ &\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s_1}{2^{n-1}}\right) + \ell_{n-1}^{(k-1)}\left(\frac{s_1}{2^{n-1}}\right)}{2} \\ &\geq \frac{\ell_{n-1}^{(k)}\left(\frac{s_1'}{2^{n-1}}\right) + \ell_{n-1}^{(k-1)}\left(\frac{s_1'}{2^{n-1}}\right)}{2} \text{ (by monotonicity of } \ell_{n-1}^{(k)}) \\ &= \mathsf{Rec}(s_0', s_1') = \ell_n^{(k)}(\mu).\end{aligned}$$

□

Claim 28. $\ell_n^{(k)} \leq \mathsf{OnExp}_n^{(k)}$.

Proof. The proof is by induction on n. The claim hold for $n = 0$. Using Claim 23 and 27 and induction we get:

$$\mathsf{OnExp}_n^{(k)} \geq \mathsf{Rec}_n^{(k)}\left[\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}\right] \geq \ell_n^{(k)}.$$

□

Now we can finish the proof of Theorem 18. If $\mu = \beta_n^{(t)}$ for some $t \in \langle n \rangle$, it then always holds that $\ell_n^{(k)}(\mu) \geq \mathsf{OnExp}_n^{(k)}(\mu)$ simply because $\ell_n^{(k)}(\mu)$ describes how much one particular protocol (i.e., τ_t) can bound adversary's power, while $\mathsf{OnExp}_n^{(k)}(\mu)$ is equal to the *minimum* of the same quantity among all protocols. Therefore, by Claim 28, $\mathsf{OnExp}_n^{(k)}\left(\beta_n^{(t)}\right) = \ell_n^{(k)}\left(\beta_n^{(t)}\right) = \beta_n^{(t-k)}$. □

Relaxing the last message to non-binary. Here we discuss an extension to Theorem 18 that follows essentially from the same proof. Theorem 18 shows that online attacks are as powerful as offline attacks when we focus on protocols with uniform binary messages. Now, suppose we allow the last message of the protocol to be an arbitrary long message, while every other message is supposed to be a uniform bit. We refer to such protocols as *binary-except-last-message* (BELM) protocols. Note that BELM protocols constitute a *larger* set of protocols, and hence they potentially could include more robust protocols that further limits the power of (offline or online) attacks. We observe that, essentially the same proof as that of Theorem 18 shows that we can strengthen Theorem 18 as follows.

Theorem 29 (Informally stated: extending Theorem 18 to BELM protocols)**.** *Suppose a random process $\mathbf{w}_{\leq n} = (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ has the property that all the first $n-1$ blocks are independent and uniform random bits, and suppose f is a Boolean function defined over this random process. Suppose $\Pr[f(\mathbf{w}_{\leq n}) = 1] = \beta_n^{(t)}$ for some $t \in [n]$. Then, there is an online k-replacing adversary over $\mathbf{u}_{\leq n}$ that generates joint random process $(\mathbf{u}_{\leq n}, \mathbf{v}_{\leq n})$ with $\mathbf{v}_{\leq n}$ being the output process, such that $\Pr[f(\mathbf{w}_{\leq n}) = 1] \geq \beta_n^{(t-k)}$. Note that this is optimal in a strong sense: there is a fully binary protocol (i.e., the threshold function τ_t) for which even* offline *k-replacing adversaries are limited to achieve offline expansion at most $\beta_n^{(t-k)}$.*

Proof Sketch. The proof of the above improved variant of Theorem 18 relies on two observations. One of them is the basis of the induction, when $n = 1$, and the other one is the improved induction step which follows from the improve variant of Claim 27 as explained below.

Relaxing transformation of Definition 20. Claim 27 was the heart of the proof of Theorem 18. In this claim, we deal with the recursion of Eq. (15) which is defined by splitting integer s into smaller integers, computing some recursive expansions and taking the minimum. It is easy to see that Claim 27 holds even if we relax the way we split s into smaller quantities and pick such pairs as *real* values

$$\widetilde{\mathcal{D}iv}_{n-1}(s) = \left\{(s_0, s_1) \mid s_0, s_1 \in \mathbb{R}, 0 \leq s_0 \leq s_1 \leq 2^{n-1}, s = s_0 + s_1\right\}.$$

In particular, let $\widetilde{\mathsf{Rec}}_n^{[k]}$ be the similar transformation using this relaxed variant $\widetilde{\mathcal{D}iv}_{n-1}(s)$ instead. First, note that by this relaxation instead, we might end up getting *smaller* expansions; namely, $\widetilde{\mathsf{Rec}}_n^{(k)} \leq \mathsf{Rec}_n^{(k)}$. Yet, the same proof shows that Claim 27 holds even if we use $\widetilde{\mathsf{Rec}}_n^{(k)}$ instead of $\mathsf{Rec}_n^{(k)}$. Moreover, in (both variants of) Case 1, it is now always possible to achieve the equality using some pair in $\widetilde{\mathcal{D}iv}_{n-1}(s)$. Therefore, this time we obtain a slightly stronger statement than that of Claim 27 for BELM protocols as follows.

Claim 30 (Variant of Claim 27 for BELM protocols)**.** $\ell_n^{(k)} = \widetilde{\mathsf{Rec}}_n^{(k)}[\ell_{n-1}^{(k)}, \ell_{n-1}^{(k-1)}]$.

The proof of the claim above is identical to that of Claim 27. □

References

1. Amir, D., Milman, V.: Unconditional and symmetric sets in n-dimensional normed spaces. Israel Journal of Mathematics **37**(1-2), 3–20 (1980) 5

2. Beimel, A., Haitner, I., Makriyannis, N., Omri, E.: Tighter bounds on multi-party coin flipping via augmented weak martingales and differentially private sampling. In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS). pp. 838–849. IEEE (2018) 10
3. Ben-Or, M., Linial, N.: Collective coin flipping. Advances in Computing Research **5**, 91–115 (1989) 4, 10
4. Ben-Or, M., Linial, N.: Collective coin flipping. Advances in Computing Research **5**, 91–115 (1990) 2, 10
5. Bentov, I., Gabizon, A., Zuckerman, D.: Bitcoin beacon. arXiv preprint arXiv:1605.04559 (2016) 10
6. Blum, M.: How to exchange (secret) keys. ACM Transactions on Computer Systems **1**, 175–193 (1984) 2
7. Cleve, R.: Limits on the security of coin flips when half the processors are faulty. In: Proceedings of the eighteenth annual ACM symposium on Theory of computing. pp. 364–369. ACM (1986) 10
8. Cleve, R., Impagliazzo, R.: Martingales, collective coin flipping and discrete control processes. Manuscript (1993) 5
9. Dachman-Soled, D., Lindell, Y., Mahmoody, M., Malkin, T.: On the black-box complexity of optimally-fair coin tossing. In: Theory of Cryptography Conference. pp. 450–467. Springer (2011) 10
10. Dachman-Soled, D., Mahmoody, M., Malkin, T.: Can optimally-fair coin tossing be based on one-way functions? In: Lindell, Y. (ed.) TCC 2014: 11th Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 8349, pp. 217–239. Springer, Heidelberg, Germany, San Diego, CA, USA (Feb 24–26, 2014). https://doi.org/10.1007/978-3-642-54242-8_10 10
11. Dodis, Y.: New imperfect random source with applications to coin-flipping. In: International Colloquium on Automata, Languages, and Programming. pp. 297–309. Springer (2001) 10
12. Etesami, O., Mahloujifar, S., Mahmoody, M.: Computational concentration of measure: Optimal bounds, reductions, and more. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 345–363. SIAM (2020) 3, 5, 6, 7
13. Goldwasser, S., Kalai, Y.T., Park, S.: Adaptively secure coin-flipping, revisited. In: International Colloquium on Automata, Languages, and Programming. pp. 663–674. Springer (2015) 2, 5, 11
14. Habib, M., McDiarmid, C., Ramirez-Alfonsin, J., Reed, B.: Probabilistic methods for algorithmic discrete mathematics, vol. 16. Springer Science & Business Media (2013) 12
15. Haitner, I., Karidi-Heller, Y.: A tight lower bound on adaptively secure full-information coin flip. Foundations of Computer Science (FOCS), IEEE 61st Annual Symposium on (2020) 3, 5, 10
16. Haitner, I., Makriyannis, N., Omri, E.: On the complexity of fair coin flipping. In: Theory of Cryptography Conference. pp. 539–562. Springer (2018) 10
17. Haitner, I., Nissim, K., Omri, E., Shaltiel, R., Silbak, J.: Computational two-party correlation: A dichotomy for key-agreement protocols. In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS). pp. 136–147. IEEE (2018) 10
18. Haitner, I., Tsfadia, E.: An almost-optimally fair three-party coin-flipping protocol. SIAM Journal on Computing **46**(2), 479–542 (2017) 10
19. Harper, L.H.: Optimal numberings and isoperimetric problems on graphs. Journal of Combinatorial Theory **1**(3), 385–393 (1966) 4, 6, 19
20. KAHN, J.: The influence of variables on boolean functions. In: Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science, 1988 (1988) 10

21. Kalai, Y.T., Komargodski, I., Raz, R.: A lower bound for adaptively-secure collective coin-flipping protocols. In: 32nd International Symposium on Distributed Computing (2018) 3, 5, 6, 10
22. Khorasgani, H.A., Maji, H.K., Mukherjee, T.: Estimating gaps in martingales and applications to coin-tossing: constructions and hardness. In: Theory of Cryptography Conference. pp. 333–355. Springer (2019) 3, 5, 10
23. Khorasgani, H.A., Maji, H.K., Wang, M.: Optimally-secure coin-tossing against a byzantine adversary. Cryptology ePrint Archive, Report 2020/519 (2020), https://eprint.iacr.org/2020/519 3, 4, 5, 6, 9, 10
24. Lichtenstein, D., Linial, N., Saks, M.: Some extremal problems arising from discrete control processes. Combinatorica **9**(3), 269–287 (1989) 3, 5
25. Mahloujifar, S., Diochnos, D.I., Mahmoody, M.: Learning under p-Tampering Attacks. In: ALT. pp. 572–596 (2018) 5, 10
26. Mahloujifar, S., Mahmoody, M.: Blockwise p-tampering attacks on cryptographic primitives, extractors, and learners. In: Theory of Cryptography Conference. pp. 245–279. Springer (2017) 5
27. Mahloujifar, S., Mahmoody, M.: Blockwise p-tampering attacks on cryptographic primitives, extractors, and learners. In: Theory of Cryptography Conference. pp. 245–279. Springer (2017) 10
28. Mahloujifar, S., Mahmoody, M.: Can adversarially robust learning leverage computational hardness? In: Garivier, A., Kale, S. (eds.) Proceedings of the 30th International Conference on Algorithmic Learning Theory. Proceedings of Machine Learning Research, vol. 98, pp. 581–609. PMLR, Chicago, Illinois (22–24 Mar 2019), http://proceedings.mlr.press/v98/mahloujifar19a.html 3, 4, 5, 6, 7, 8, 16
29. Mahloujifar, S., Mahmoody, M., Mohammed, A.: Universal multi-party poisoning attacks. In: Proceedings of the 36th International Conference on Machine Learning. vol. 97, pp. 4274–4283 (2019) 5, 10
30. Maji, H.K., Wang, M.: Black-box use of one-way functions is useless for optimal fair coin-tossing. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12171, pp. 593–617. Springer (2020). https://doi.org/10.1007/978-3-030-56880-1_21, https://doi.org/10.1007/978-3-030-56880-1_21 10
31. Margulis, G.A.: Probabilistic characteristics of graphs with large connectivity. Problemy peredachi informatsii **10**(2), 101–108 (1974) 5
32. McDiarmid, C.: On the method of bounded differences. Surveys in combinatorics **141**(1), 148–188 (1989) 5
33. Milman, V.D., Schechtman, G.: Asymptotic theory of finite dimensional normed spaces, vol. 1200. Springer Verlag (1986) 5
34. Moran, T., Naor, M., Segev, G.: An optimally fair coin toss. In: TCC. pp. 1–18 (2009) 10
35. Russell, A., Saks, M., Zuckerman, D.: Lower bounds for leader election and collective coin-flipping in the perfect information model. SIAM Journal on Computing **31**(6), 1645–1662 (2002) 10
36. Talagrand, M.: Concentration of measure and isoperimetric inequalities in product spaces. Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques **81**(1), 73–205 (1995) 5