Cleve's Bound Is Tight: Optimally Fair Coin Flipping for Every Fixed Number of Parties

Yuwen Zhou
University of British Columbia Okanagan
zhouyw07@student.ubc.ca

Abstract

Proof-of-stake chains, committee selection and lotteries draw public randomness from protocols whose participants may collude. Cleve [STOC 1986] showed that without an honest majority, aborting alone lets a coalition bias the outcome of any r-round n-party coin flip by $\Omega(1/r)$. Moran, Naor and Segev [TCC 2009] matched this bound for two parties, and later work came within a $\sqrt{\log r}$ factor of it for three. For four or more parties, the best known bias is about $r^{-1/2-1/(2^{n-1}-2)}$, which approaches the $r^{-1/2}$ of the simple majority protocol as n grows. The obstacle is recovery: after an abort, the remaining parties must carry on without giving the coalition a fresh chance to bias the outcome. In the best earlier protocol they reconstruct a noisy copy of the current expected outcome, and with more than three parties this costs a power of r at each level. We show that Cleve's bound is tight for every fixed number of parties. Assuming constant-round oblivious transfer and constant-round passively secure computation, we give an n-party r-round protocol over a broadcast channel whose bias against $n-1$ corrupted parties is $O(n^3 2^n/r)$. The protocol draws the outcome at the start and keeps it secret, and the coins it reveals form a random walk conditioned on that outcome. After an abort, the remaining parties reconstruct nothing: they start a new run whose outcome probability is a public table evaluated on bits they hold in secret-shared form. Our main technical result is that the entries of every such table stay within a factor of two of each other, however many aborts came before, so each level of recovery costs only a constant factor. The protocol therefore improves on majority for up to about $\frac{1}{2}\log_2 r$ parties, and a variant for any number of parties has an exponential factor set only by how many more parties are corrupted than honest.

Declaration of generative AI use (IACR category: Substantial AI contribution). The proof presented in this paper was developed through an interactive dialogue with Claude Opus 5.5 (Anthropic) and ChatGPT-6 (OpenAI). The author used them to explore multiple possible proof strategies and directions, and the AI generated the initial draft under the author's guidance. The author then critically examined the generated proof, verified each step for correctness and rigor, and revised it where necessary. The author takes full responsibility for the final content.

1. Introduction

Public randomness decides who proposes the next block of a proof-of-stake chain, which validators sit on a committee, who wins a lottery and which ballots are audited [19]. The protocols that produce it are judged first by whether a participant can steer the output. The simplest design, commit–reveal [12], lets every party commit to a random string and then open it, and outputs the XOR of the strings. It has a weakness that is easy to state: the party that opens last computes

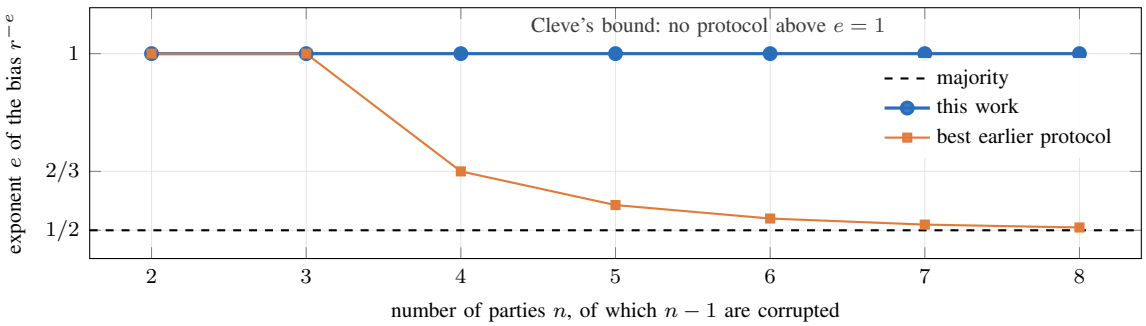

Figure 1. Earlier protocols fall back toward the majority protocol as parties are added; ours stays at Cleve's bound. The plot shows the exponent e of the best known bias r^{-e} against $n-1$ corrupted parties out of n, for fixed n and with logarithmic factors dropped.

the output before the others do, and can refuse to open when it dislikes what it sees. Ethereum's RANDAO beacon, which secures billions of dollars of stake, answers this *last-revealer attack* with penalties for parties that fail to reveal [15], [19].

The weakness is not particular to commit–reveal. Cleve [22] showed that in every r-round coin-flipping protocol in which half or more of the parties may collude, some coalition can bias the output by $\Omega(1/r)$, and it does so by following the protocol and choosing the round at which it stops. Deployed beacons therefore pay for an unbiased output with an additional assumption: that fewer than a threshold of the participants collude [17], [27], [46], [48], that some function cannot be evaluated faster than a set delay [4], [14], [15], or that parties who stop can be fined [3], [11]. Whether such an assumption is acceptable depends on the deployment, whereas rounds are available to every protocol. What rounds alone can buy is thus the baseline against which the three assumptions are measured:

How small can the bias of an r-round coin flip be when all parties but one may collude?

For two parties the answer has been known since 2009, when Moran, Naor and Segev [42] matched Cleve's bound with a protocol of bias $O(1/r)$. For more parties it has remained open. The textbook protocol, in which the parties flip r public coins and output their majority [1], has bias $O(n/\sqrt{r})$. Haitner and Tsfadia [32] reached $O(\log^3 r/r)$ for three parties. Buchbinder, Haitner, Levi and Tsfadia (BHLT) [16] improved this to $O(\sqrt{\log r}/r)$ and gave an n-party protocol with bias about $r^{-1/2-1/(2^{n-1}-2)}$ against $n-1$ corruptions. At four parties this is about $r^{-2/3}$, and with every further party the exponent falls back toward the $1/2$ of majority (Fig. 1), so that the protocol improves on majority only for $n \le \frac{1}{2}\log\log r$. BHLT call the optimal bias for more than two parties "the main open question in this area," and Beimel et al. [7] point out that for $\log\log r < n \le \log r$ the only known bounds are Cleve's $1/r$ and the $n/\sqrt{r}$ of majority.

1.1. Our Results

We show that Cleve's bound is tight for every fixed number of parties, with an explicit dependence on that number.

Theorem 1 (informal). *Assume constant-round oblivious transfer and constant-round passively secure computation. There is a universal constant c such that for every n and every polynomially bounded $T \ge 2^n$ there is an n-party coin-flipping protocol over a broadcast channel with $O(n^3T)$*

TABLE 1. COIN FLIPPING WITH NO HONEST MAJORITY IN r ROUNDS. IN THE ROWS WITH N PARTIES AND t CORRUPTIONS, $k = 2t - N$ IS THE GAP BETWEEN CORRUPTED AND HONEST PARTIES AND $s = N - k - 1$.

Protocol	Corrupted	Bias
Majority [1]	$n-1$ of n	$O(n/\sqrt{r})$
Moran et al. [42]	1 of 2	$O(1/r)$
Beimel et al. [8]	$t < 2N/3$ of N	$O(2^{2^{k+1}}/r)$
Haitner–Tsfadia [32]	2 of 3	$O(\log^3 r/r)$
Alon–Omri [2]	$t < 3N/4$ of $N = O(1)$	$O(\log^3 r/r)$
BHLT [16]	2 of 3	$O(\sqrt{\log r}/r)$
BHLT [16]	$n-1$ of n	$\tilde{O}(r^{-1/2-1/(2^{n-1}-2)})$
This work	$n-1$ of n	$O(n^3 2^n/r)$
This work	$t \geq N/2$ of N	$O((k+2)^3 2^k s^2/r)$
Lower bound [22]	half or more	$\Omega(1/r)$

rounds, whose bias against any static coalition of up to $n-1$ parties is at most $c\,2^n/T$ plus a negligible term.

In r rounds the bias is thus $O(n^3 2^n/r)$, and for every fixed n it is $O(1/r)$, which by Cleve's bound cannot be improved. For three parties the theorem removes the last $\sqrt{\log r}$, a question that BHLT leave open, and for every $n \geq 4$ it improves the best known bias by a power of r. The constant matters as much as the rate: the bias is below that of majority as long as $n \leq \frac{1}{2}\log_2 r - 2\log_2\log_2 r - \omega(1)$ (Appendix K). This is exponentially more parties than the $\frac{1}{2}\log\log r$ of BHLT, and up to lower-order terms it covers the lower half of the range singled out by Beimel et al.

Our second result shows that the exponential factor is set by the gap between corrupted and honest parties, not by the number of parties.

Theorem 2 (informal). *Under the same assumptions, let N parties communicate over a broadcast channel, let $t \geq N/2$ of them be statically corrupted, and let $k = 2t - N$ and $s = N - k - 1$. There is an r-round N-party coin-flipping protocol with bias $O((k+2)^3 2^k s^2/r)$ plus a negligible term. If constant-round computation with an honest majority and guaranteed output is available (Assumption 2), the factor s^2 can be dropped.*

The earlier protocol of this kind, by Beimel, Omri and Orlov [8], requires $t < 2N/3$ and has bias $O(2^{2^{k+1}}/r)$; we remove the restriction on t and replace the doubly exponential dependence on the gap by a singly exponential one.

What the results mean in practice. With the majority protocol, halving the bias costs four times the rounds; with ours it costs twice the rounds. We implemented the protocol for fail-stop adversaries and computed the best fail-stop attack on it and on majority: at three parties, a run of our protocol uses fewer broadcast rounds than majority once the target bias is below about 0.5%, and 4.96 times fewer at a bias of 0.1% (Section 9). For a large committee, Theorem 2 says that if the corrupted parties outnumber the honest ones by a constant, the bias is $O(s^2/r)$, at the rate of Cleve's bound, where majority-type protocols have $O(t/\sqrt{r})$.

Contributions. This paper contributes

- *a protocol with a hidden outcome*, in which the remaining parties answer an abort by a new run on a public table evaluated on bits they hold in shares (Section 4);

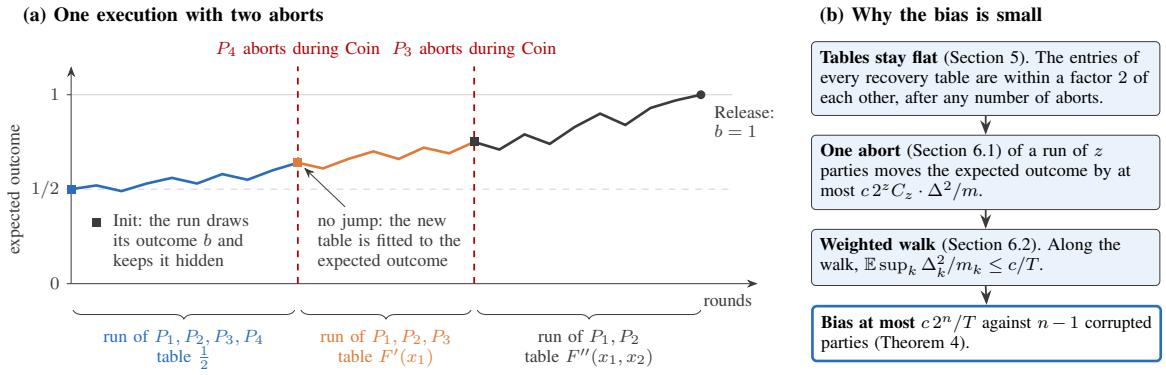

Figure 2. The protocol and its analysis (schematic). (a) Each run draws its outcome at Init and reveals it only at Release; the public coins move the expected outcome toward it. When a party aborts while a coin is being revealed, the coin stays unpublished and the remaining parties start a new run from a public table evaluated on weak bits x_i that they hold in shares about the unpublished coins. The table is fitted so that the expected outcome does not move at the abort. (b) The three steps of the proof.

- *an analysis of recovery that does not degrade with the number of aborts*: every recovery table has a closed form, and its entries stay within a factor two of each other (Section 5);
- *a tight bound for every fixed number of parties*, in a hybrid model and, through a compilation with identified abort, against malicious adversaries over broadcast (Sections 6 and 7);
- *an extension to any number of parties*, with a bias that depends on the gap between corrupted and honest parties (Section 8), and *a prototype* with a comparison against majority (Section 9).

All proofs are in the appendices, and the code is in the repository at https://anonymous.4open.science/r/fair-coin-flipping-artifact-8F7C.

2. Overview: A Hidden Outcome

Throughout the section the adversary is *fail-stop*: the corrupted parties follow the protocol and only choose when to abort. We also let the parties compute any efficient functionality securely with abort, where the corrupted parties receive their outputs first and may then abort.

2.1. Where Earlier Protocols Lose

Majority. In the protocol of Awerbuch et al. [1] the parties flip public coins and output the sign of their sum, and if some parties abort, the remaining parties re-toss the current coin. Since the sum of r coins is close to zero with probability $\Theta(1/\sqrt{r})$, a party that aborts after seeing the first coin biases the output by $\Theta(1/\sqrt{r})$.

Defense values. To improve on this, Haitner and Tsfadia (HT) [32], building on Moran et al. and Beimel et al., give the parties *defense values* before each coin. Every proper subset of the parties receives shares of a value that leans slightly toward the coin about to be flipped, and if some parties abort after seeing the coin, the remaining parties finish with the expected outcome that the coin produced. An abort after seeing the coin then gains nothing. The defense values are correlated with the coin, however, so the weak point moves to the step in which the corrupted parties receive them: if the coin moves the expected outcome by Δ, an abort at that step gains

Input: a public table F; the parties of Z hold shares of hidden bits h.

Init: draw the outcome $b \sim \mathrm{Bern}(F(h))$ and the coins $(r_T, \ldots, r_1)$ of a weighted walk conditioned on outcome b, and keep them shared in Z.

For $k = T, \ldots, 1$**:**
1. *Defense:* every proper nonempty $Y \subset Z$ receives shares of a fresh bit that agrees with r_k with probability $1/2 + \kappa_k$.
2. *Coin:* reveal r_k.

Output: Release reveals b, and all parties output b.

Abort: let Z' be the remaining parties. If $|Z'| = 1$, the party outputs a bit according to a public table evaluated on its defense bits. Otherwise Z' starts a new run on a public table evaluated on its hidden defense bits.

Figure 3. The protocol, simplified. Rounds that skip Defense are omitted.

about Δ^2 rather than Δ. Giving more weight to the coins of early rounds prevents large moves in the last rounds.

Recovery. With two or more remaining parties the recovery must itself be fair, so it needs defense values of its own, and with three or more those need a recovery of their own. In BHLT the remaining parties reconstruct a noisy copy of the current expected outcome and continue with a protocol biased accordingly. With too little noise the corrupted parties learn the expected outcome early, and with too much noise the reconstruction is itself a large jump. With more than three parties the compromise costs a power of r at each level of the recursion, and this is the source of the exponent in the bias of BHLT.

2.2. Drawing the Outcome First

Our protocol follows the paradigm of HT and BHLT, with weighted coins, defense values before each coin, recursion on the remaining parties, exact matching of expected outcomes and the view of the attack as a stopping game. The difference is in what a recovery starts from. Fig. 3 gives a simplified description and Fig. 2(a) shows an execution.

Namely, the outcome b is drawn from the table once, inside a functionality, and remains hidden until the last call. The coins in between are a walk conditioned on b, so everything the parties see along the way is evidence about a bit that already exists, and the coins depend on the hidden bits only through b. If some parties abort, the remaining parties reconstruct nothing. They start a new run on a public table that says, for each value of the bits they hold in shares, with what probability the new run draws the outcome 1. We refer to this design as a *hidden outcome*.

2.3. One Abort, Worked Out

Take three parties, and let one of them abort while the first coin is being revealed, so that the coin is never published; we call such a coin *pending*. Before this coin the expected outcome is $1/2$, and the coin would have moved it to $1/2 \pm \Delta$. The two remaining parties hold shares of a defense bit $x \in \{\pm 1\}$ that agrees with the pending coin with probability $1/2 + \kappa$. Their table F' must return, for each value of the coin, the expected outcome that this value would have produced, that is, $\mathbb{E}[F'(x) \mid \text{coin}] = 1/2 \pm \Delta$. This forces

$$F'(x) = \frac{1}{2} + x\,\frac{\Delta}{2\kappa}.$$

Our protocol sets $\kappa = 16\Delta$ at this point, so the table is $(17/32,\ 15/32)$, independent of Δ. The example shows the two properties that the analysis rests on. First, the abort does not move the expected outcome, because the table is fitted to it. Second, the table is close to constant, so the new run is close to a fresh run with a fair table.

2.4. The Strength of the Defense Bits

Let y be the current expected outcome, Δ the move of the next coin and $m = \min(y, 1-y)$. An abort after the corrupted parties read a defense bit of strength κ gains about $\kappa\Delta$, and the next table swings by about $\Delta/(2\kappa)$. A run of z parties sets $\kappa = C_z\Delta/m$ for a constant C_z. The gain is then $C_z g$ with $g = \Delta^2/m$, and the swing is about $m/(2C_z)$, a fixed fraction of the distance of the expected outcome from 0 and 1. Hence the ratio between entries of the next table is bounded by a constant that does not depend on the length of the walk. We call such a table *flat*. Under a flat table the law of the public coins is within a constant factor of the plain walk (Lemma 3), and this is what lets us analyze every run as if it were the first.

2.5. Many Aborts: the Tables Stay Flat

After several aborts the remaining parties hold weak bits about several pending coins, and the table must undo the noise on all of them at once. Undoing noise divides by the strengths, which can be tiny, and a bound that degrades with each pending coin would bring back a loss at each level.

The core of our analysis is a closed form. Seen through the noise on the defense bits, every table of the protocol is the posterior that the parties would hold had the pending coins been published: a logistic function of a linear form in which each pending coin contributes its log-likelihood ratio (Lemma 4). For tables of this shape the division is paid for. A log-likelihood ratio is itself of the order of the strength of its coin, and in a product of two functions every coin that both depend on contributes the square of its strength. Measuring a table by the sum of the absolute values of its Fourier coefficients, we obtain that all entries are within a factor $1 \pm \epsilon$ of a common value, where ϵ depends only on the sum of the ratios between log-likelihood ratios and strengths, and not on the number of pending coins (Lemma 5). A run of z parties adds about $1/C_z$ to this sum, so the constants only need to be summable. We take $C_z = 4(n-z+1)(n-z+2)$, which is 8 at the top run for every n, and every table of every execution is flat with ratio at most 2 (Lemma 6).

2.6. Bounding the Attack

Since a fail-stop adversary only chooses when to abort, its bias is at most the expected sum, over the at most $n-1$ aborts, of the jump in the expected outcome that each abort causes. Apart from a rare tie at the last call, the jump of an abort in a run of z parties is at most a constant times $2^z C_z g$ (Lemma 7); the factor 2^z counts the defense bits that the coalition reads in one Defense call, one for each of its subsets. It remains to bound the expected largest g along the walk. Let the coin with k rounds left have weight k. Near the center the expected outcome then moves by about $1/\sqrt{k}$, so that g is of order $1/k$, and the walk is near the center with probability about $(k/T)^{3/2}$, where T is the length of the walk. Summing over dyadic k gives an expected supremum of order $1/T$ (Theorem 3). Adding up over the levels, with $\sum_z 2^z C_z = O(2^n)$, gives the bias $c\,2^n/T$ (Theorem 4); Fig. 2(b) summarizes the route.

Malicious adversaries and more parties. Against malicious adversaries, each functionality becomes one secure computation over broadcast in which every abort names a corrupted party, and the parties carry their shares from call to call under one-time signatures (Section 7). For N parties of which t are corrupted, with gap $k = 2t - N$, the parties emulate our $(k+2)$-party protocol: $k+1$ of them play one party each and the others play the last party jointly, and a coalition of t parties cannot control all $k+2$ emulated parties (Section 8).

3. Model and Main Results

Model. We consider n parties in synchronous rounds with an authenticated broadcast channel and no setup; a broadcast channel is necessary for fair coin flipping against one third or more corruptions [25]. The corrupted parties are controlled by a malicious adversary that runs in time polynomial in the security parameter λ, is static (the corrupted set is fixed before the protocol starts) and is rushing (it sees the honest messages of a round before sending its own). All honest parties must output a common bit, even after an abort, and the *bias* of a protocol is the largest value of $|\Pr[\text{out} = 1] - 1/2|$ over adversaries. When a majority of the parties is honest, protocols with negligible bias follow from secure multi-party computation [10], [31], so we consider the case of no honest majority.

Assumptions. Oblivious transfer (OT) is the standard primitive of secure computation: a sender holds two messages and a receiver learns one of them, without the sender learning which. OT is the assumption of every optimal-rate protocol in this line, and black-box use of one-way functions does not suffice [26], [41]. OT gives the one-way functions and pseudorandom generators we use [33], [36]. Our compilation also uses a passive protocol whose number of rounds does not grow with the number of parties.

Assumption 1 (Constant-round passive computation)**.** *Given a constant-round OT protocol, every polynomial-time functionality among z parties can be computed with security against any number of passive corruptions, in a number of rounds bounded by an absolute constant that depends neither on z nor on the functionality.*

This is the passive, dishonest-majority form of the garbling protocol of Beaver, Micali and Rogaway [6]. Ben-Efraim, Lindell and Omri state it as Theorems 2.1 and 3.2 of [9], for a garbling with free XOR, which uses a circular correlation-robust hash function [20]; the proofs are in the extended version of [9].

Theorem 1 (Main theorem)**.** *Assume a constant-round protocol for oblivious transfer and Assumption 1. There is a universal constant c with the following property. Let $n = n(\lambda) \geq 2$ with $2^n \leq \text{poly}(\lambda)$, and let $T = T(\lambda) \leq \text{poly}(\lambda)$. The n-party protocol of Sections 4 and 7 with walk length T runs in polynomial time over a broadcast channel and has $O(n^3 T)$ rounds. All honest parties terminate with a common bit, which is uniform when no party aborts, and against every static polynomial-time coalition of at most $n-1$ parties*

$$\left|\Pr[\text{out} = 1] - \tfrac{1}{2}\right| \leq \frac{c\,2^n}{T} + \text{negl}(\lambda).$$

In particular, for every fixed n and every polynomially bounded r there is an $O_n(r)$-round protocol with bias at most $c\,2^n/r + \text{negl}(\lambda)$.

The price of each additional party is a factor of two in the bias and a polynomial factor in the rounds. In terms of the number of rounds $r = O(n^3 T)$, the bias is $O(n^3 2^n / r) + \text{negl}(\lambda)$, which is below the $O(n/\sqrt{r})$ of majority whenever $n^2 2^n = o(\sqrt{r})$.

TABLE 2. NOTATION OF SECTIONS 4 TO 6.

T, k	walk length; rounds left, counting down
R_k, σ_k^2	weighted sum $\sum_{i\le k}\varepsilon_i i$ and its variance
$y_k(S)$	expected outcome at state S, k rounds left
Δ_k, m_k	next move of y_k; distance of y_k to $\{0,1\}$
g_k	Δ_k^2/m_k, which bounds the gain of an abort
Z, z	remaining parties in the current run, $z=\lvert Z\rvert$
F, A	public table on $\{\pm1\}^p$; reference parameter
v, h	pending coins; Z's hidden defense bits about them
κ_i, u_i	strength of a defense bit; $u_i=2\kappa_i$
b, τ	hidden outcome of a run; $\Pr[b=1 \mid v,\mathrm{path}]$
N_u, $\mathcal{I}$	noise operator and its inverse
C_z	strength constant of a run of z parties

Theorem 2 (Any number of parties)**.** *Assume a constant-round protocol for oblivious transfer and Assumption 1. Let $N=N(\lambda)\le\mathrm{poly}(\lambda)$ parties communicate over an authenticated broadcast channel, let $N/2\le t\le N-1$, and let $k=2t-N$ and $s=N-k-1$, with $2^k\le\mathrm{poly}(\lambda)$. For every $T'\le\mathrm{poly}(\lambda)$, the N-party protocol $\Pi^\star$ of Section 8 has at most $r\le C_L(\rho+1+s)s^2+\rho$ rounds, where $\rho\le C_1(k+2)^3T'$ and C_1, C_L are universal constants. All honest parties terminate with a common bit, which is uniform up to a negligible term when all parties are honest, and against every static polynomial-time coalition of at most t parties*

$$\bigl|\Pr[\mathrm{out}=1]-\tfrac12\bigr|\le\frac{c\,2^{k+2}}{T'}+\mathrm{negl}(\lambda),$$

with the constant c of Theorem 1. If $\rho\ge s+1$, the right-hand side is at most $c'(k+2)^3 2^{k+2}s^2/r+\mathrm{negl}(\lambda)$ with $c'=3cC_1C_L$.

Section 8 also shows that the factor s^2 disappears under an assumption on honest-majority computation (Theorem 5).

4. The Hidden-Outcome Protocol

We now describe the protocol in a hybrid model with fail-stop adversaries; Table 2 lists the notation.

4.1. The Reference Walk

Let $R_k=\sum_{i=1}^k\varepsilon_i i$ with independent fair signs ε_i, and let $\sigma_k^2=\sum_{i\le k}i^2$, so that $k^3/3\le\sigma_k^2\le k^3$. For an integer S and $\alpha\in[0,1]$ define

$$y_k(S)=\Pr[S+R_k>0]+\alpha\Pr[S+R_k=0],$$

and $f_k(S)=1-y_k(S)$. Then $y_k(S)=\bigl(y_{k-1}(S+k)+y_{k-1}(S-k)\bigr)/2$. For every $A\in(0,1)$ there are (t_0,α) with $y_T(t_0)=A$: the function y_T is nondecreasing in S, and α sweeps the gap at each atom.

The law Ref_A draws a fair path $(\varepsilon_T,\dots,\varepsilon_1)$ and then a terminal bit $\mathrm{out}\sim\mathrm{Bern}\bigl(y_0(t_0+\sum_i\varepsilon_i i)\bigr)$. Rounds count down, so k is the number of rounds left. With $S_k=t_0+\sum_{i>k}\varepsilon_i i$ we have

$\Pr_{\mathrm{Ref}_A}[\mathrm{out} = 1 \mid \text{prefix}] = y_k(S_k)$, the *expected outcome* at state S_k. The bit out is determined by the path except when $S_0 = 0$.

We write $\Delta_k(S) = |y_{k-1}(S+k) - y_{k-1}(S-k)|/2$ for the size of the next move of the expected outcome, and $m_k(S) = \min(y_k(S), f_k(S))$ for its distance to the nearer end of $[0,1]$. Always $\Delta \le m$, because y_k is the midpoint of two numbers in $[0,1]$. Let $g_k = \Delta_k^2/m_k$, with $0/0 = 0$; this is the ratio that governs the attack.

4.2. The Protocol in the Hybrid Model

A functionality is called by a set Z with shares as input. It returns output to the corrupted parties first; the adversary may then abort, possibly for several parties at once, and the honest parties get a broadcast abort notice naming the aborting parties (identified abort). All parameters of a call are determined by public data. This hybrid model, with fail-stop adversaries, follows the conventions of BHLT [16].

Configurations. A run of z parties uses the constant

$$C_z = 4(n - z + 1)(n - z + 2), \qquad z = 2, \dots, n, \tag{1}$$

so that $C_n = 8$ for every n (Section 5.3 explains the choice). A configuration is (Z, κ, F) with $z = |Z|$, public strengths $\kappa \in (0, 1/2)^p$ and a public table $F : \{\pm 1\}^p \to [0,1]$. The set Z holds z-out-of-z shares of $h \in \{\pm 1\}^p$, its defense bits about the *pending coins* $v \in \{\pm 1\}^p$, which are coins of earlier runs that were never published. The reference parameters (t_0, α) of the run are chosen with $A = y_T(t_0)$ equal to the mean $2^{-p}\sum_h F(h)$ of the table. If F is constantly 0 or 1, every party in Z outputs that bit at once, with no call.

Run(Z, κ, F)**.**

- *Init.* The functionality reconstructs h, draws $b \sim \mathrm{Bern}(F(h))$, draws $(r_T, \dots, r_1)$ from Ref_A conditioned on $\mathrm{out} = b$, and keeps b and the coins shared in Z.
- *Round $k = T, \dots, 1$, state $S = S_k$.* Let $\Delta = \Delta_k(S)$ and $m = m_k(S)$. A round is *direct* if $\Delta = 0$ or $C_z\Delta/m \ge 1/2$ (and $m = 0$ forces $\Delta = 0$); it then only makes a Coin call, which reveals r_k, adversary first. Otherwise let $\kappa_k = C_z\Delta/m$ and make two calls. *Defense:* for every proper nonempty $Y \subset Z$, share among Y a fresh bit $d_{Y,k}$ with $\Pr[d_{Y,k} = r_k \mid r_k] = 1/2 + \kappa_k$, independent across Y given r_k. *Coin:* reveal r_k, adversary first.
- *Release.* Reveal b, adversary first. If there is no abort, output b.

Abort with remaining set Z'. Let path be the committed Coin outputs, and let $\tau(v, \mathrm{path}) = \Pr[b = 1 \mid v, \mathrm{path}]$ in the current run.

- During Coin of a non-direct round k: run $(Z', (\kappa, \kappa_k), F')$ with pending coins (v, r_k), where Z' holds its own bits about v from earlier Defense calls and $d_{Z',k}$, and $F' = \mathcal{I}[\tau(\cdot, \mathrm{path}, \cdot)]$.
- At Init, during Defense, in a direct Coin, or at Release: run (Z', κ, F') with pending coins v, where Z' holds its own bits about v, and $F' = \mathcal{I}[\tau(\cdot, \mathrm{path})]$. At Init this gives $F' = F$, read on the bits of Z'.
- If $|Z'| = 1$, the party outputs $\mathrm{Bern}(F'(\text{its bits}))$. A set Z' with no honest party is irrelevant.

Here, with $u_i = 2\kappa_i$,

$$(\mathcal{I}f)(x) = \mathbb{E}_v\Big[\prod_i \big(1 + x_i v_i/u_i\big) f(v)\Big], \quad v \text{ uniform},$$

is the inverse Bonami–Beckner noise operator [45], one coordinate at a time. It gives the unique function with $\mathbb{E}[(\mathcal{I}f)(x) \mid v] = f(v)$ when $\Pr[x_i = v_i] = (1+u_i)/2$ independently, because $\mathbb{E}_x\big[\prod_i (1 + x_i w_i/u_i) \mid v\big] = 2^p \mathbf{1}\{w = v\}$, and this holds for every prior of v.

The top configuration is $Z =$ all parties, $p = 0$, $F = 1/2$, $t_0 = 0$, $\alpha = 1/2$. Each abort removes a party, so an execution has at most $n-1$ aborts and $p \le n - z$.

Phases. A call *commits* when every party in Z has its output and none has aborted; Appendix D fixes to which call a stop between two calls belongs.

Lemma 1 (Defense invariant)**.** *At entry to a run on Z with pending coins v and entry history H (everything the adversary has seen, all coins, all b of ancestor runs, all ancestor hidden vectors): for each $Y \subseteq Z$ containing an honest party, the vector h^Y of Y's bits about v has, given (v, H), the product law $\Pr[h_i^Y = v_i] = 1/2 + \kappa_i$, independently across such Y, and its shares look uniform to the adversary. Different secrets use independent sharing randomness.*

Proof. Induction on aborts; the top is trivial. In a run on Z only h^Z influences anything sampled (b, then the path). Bits of proper subsets Y have never influenced a sampled output or the path, and fresh bits are drawn given r_k with independent randomness. The bits of all-corrupted subsets use independent randomness too. So given v, r_k, H, the run's b and path, and the adversary's view, the families $(h^Y, d_{Y,k})$ over $Y \subseteq Z'$ containing an honest party keep their product law. Choosing Z' from the adversary's view exposes nothing, since Z' has an honest member. □

5. Recovery Tables Stay Flat

The statements of this section hold for any reference parameters (t_0, α) with $A = y_T(t_0) \in (0,1)$; Section 7.4 uses this freedom.

5.1. The Expected Outcome of a Run

Lemma 2. *Let $a_v = \mathbb{E}[F(h) \mid v]$, with the law from Lemma 1. Then $\Pr[\text{path} \mid b, v, H] = \mathrm{Ref}_A(\text{path} \mid \text{out} = b)$. Hence $\operatorname{logit} \tau(v, \text{path}) = \operatorname{logit} y + \operatorname{logit} a_v - \operatorname{logit} A$, with $y = y_k(S_k)$, whenever $0 < y < 1$.*

Lemma 3. *Given (v, H), the law of $(\text{path}, \text{out})$ has density a_v/A on $\text{out} = 1$ and $(1-a_v)/(1-A)$ on $\text{out} = 0$ against Ref_A. Hence every prefix density is $y\,a_v/A + (1-y)(1-a_v)/(1-A)$, which is at most B when $a_v/A \le B$ and $(1-a_v)/(1-A) \le B$, and this includes the full path.*

Proof of Lemmas 2 and 3. Given (v, H) the invariant makes $\Pr[b = 1 \mid v, H] = a_v$, and the path is drawn given b alone, so the first claim of Lemma 2 holds. Since $A = \Pr_{\mathrm{Ref}_A}[\text{out} = 1]$, we have $\mathrm{Ref}_A(\text{path} \mid \text{out} = 1) = \mathrm{Ref}_A(\text{path})\, y/A$ and $\mathrm{Ref}_A(\text{path} \mid \text{out} = 0) = \mathrm{Ref}_A(\text{path})(1-y)/(1-A)$, and Bayes' rule gives the odds $\frac{a_v}{1-a_v} \cdot \frac{y}{1-y} \cdot \frac{1-A}{A}$, which is the logit formula. The same two identities give the densities of Lemma 3, and a mixture of two numbers that are at most B is at most B. □

5.2. Flat Tables

Definition 1. A table F is *B-flat* if $F(h)/F(h') \le B$ and $(1-F(h))/(1-F(h')) \le B$ for all h, h'.

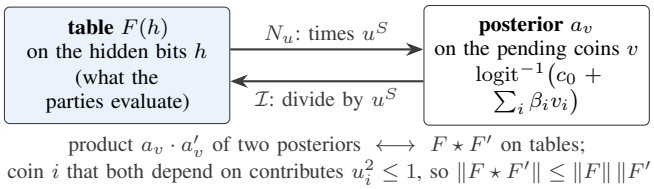

Figure 4. A table and its view through the noise. Lemma 4 shows that the view through the noise is a logistic function of the pending coins. Lemma 5 expands this function into products and uses that the product of two views corresponds to an operation on tables that does not increase the sum of the absolute Fourier coefficients.

For $f : \{\pm1\}^p \to \mathbb{R}$ write $f(x) = \sum_{S\subseteq[p]} \widehat{f}(S)x^S$ with $x^S = \prod_{i\in S} x_i$, and let $u^S = \prod_{i\in S} u_i$. The noise on the defense bits acts by

$$(N_u f)(v) = \mathbb{E}[f(h) \mid v] = \sum_S \widehat{f}(S)\, u^S v^S,$$

so that $a_v = (N_u F)(v)$, and $\mathcal{I} = N_u^{-1}$ divides the coefficient of x^S by u^S. The abort rules say that the child's table F' is the function with $N_{u'}F' = \tau$, where u' is u with the strength of the new pending coin appended, if there is one. Fig. 4 illustrates the two views of a table.

Let $\chi(C) = 2C \operatorname{artanh} \frac{1}{2C}$ for $C > 1/2$; this function decreases to 1, and $\chi(8) < 1.0014$.

Lemma 4 (Tables through the noise)**.** *Let a run of z parties have a table F with values in $(0,1)$ and*

$$a_v = \operatorname{logit}^{-1}\Big(c_0 + \sum_{i=1}^{p} \beta_i v_i\Big) \quad \textit{for all } v \in \{\pm1\}^p, \tag{2}$$

and let it end by an abort at a state with expected outcome y. If the abort leaves no new pending coin, the child's table is the constant y when $y \in \{0,1\}$, and otherwise satisfies (2) *with the same $\beta_1, \dots, \beta_p$. If the abort is during Coin of a non-direct round k, then $y_\pm = y_{k-1}(S_k \pm k)$ lie in $(0,1)$ and the child's table satisfies* (2) *with the same $\beta_1, \dots, \beta_p$ and one more coefficient*

$$\beta_{p+1} = \tfrac{1}{2}\big(\operatorname{logit} y_+ - \operatorname{logit} y_-\big).$$

In that case, with $\delta = \Delta/m < 1/(2C_z)$ and $u_{p+1} = 2\kappa_k = 2C_z\delta$,

$$\frac{|\beta_{p+1}|}{u_{p+1}} \le \frac{\operatorname{artanh}\delta}{C_z\delta} < \frac{\chi(C_z)}{C_z}.$$

Proof. Since F has values in $(0,1)$, so has a_v, and by Lemma 2 $\operatorname{logit}\tau(v,\mathrm{path}) = \operatorname{logit} y + \operatorname{logit} a_v - \operatorname{logit} A$ whenever $0 < y < 1$. By (2) this is affine in v with the coefficients β_i, which is the case with no new pending coin. If $y \in \{0,1\}$, the reached path has probability zero given $b = 1 - y$ and positive probability, so $\tau = y$.

In a non-direct round $\Delta > 0$ and $C_z\Delta/m < 1/2$, hence $\Delta < m$ because $C_z \ge 1/2$, and $\{y_+, y_-\} = \{\bar{y} + \Delta, \bar{y} - \Delta\} \subset (0,1)$ with $\bar{y} = y_k(S_k)$. Given $r_k = r$ the expected outcome of the walk is y_r, and $\operatorname{logit} y_r = \frac{1}{2}(\operatorname{logit} y_+ + \operatorname{logit} y_-) + \frac{r}{2}(\operatorname{logit} y_+ - \operatorname{logit} y_-)$, so $\operatorname{logit}\tau(v,\mathrm{path},r)$ is affine in (v,r) with the stated coefficient of r.

For the bound, replacing y by $1-y$ changes the sign of every logit and leaves Δ, m and $|\beta_{p+1}|$ unchanged, so let $\bar{y} \le 1/2$. Then $m = \bar{y}$, $\{y_+, y_-\} = \{\bar{y}(1+\delta), \bar{y}(1-\delta)\}$ and

$$2|\beta_{p+1}| = \ln\frac{1+\delta}{1-\delta} + \ln\frac{1-\bar{y}(1-\delta)}{1-\bar{y}(1+\delta)} \le 2\ln\frac{1+\delta}{1-\delta},$$

because the second term increases with $\bar{y}$ and equals the first at $\bar{y} = 1/2$. The right-hand side is $4 \operatorname{artanh} \delta$. Dividing by $2u_{p+1} = 4C_z\delta$ gives the first inequality, and the second holds because $\operatorname{artanh}(\delta)/\delta$ increases with δ and $\delta < 1/(2C_z)$. □

Lemma 5 (Flatness)**.** *Let $u \in (0,1]^p$, $c_0 \in \mathbb{R}$, $\beta \in \mathbb{R}^p$, $a = \operatorname{logit}^{-1}(c_0)$, $\Lambda = \sum_i |\beta_i|/u_i$ and $\epsilon = e^\Lambda - 1$. Let $F = \mathcal{I}[v \mapsto \operatorname{logit}^{-1}(c_0 + \sum_i \beta_i v_i)]$. If $\epsilon \le 1$, then for every $x \in \{\pm 1\}^p$*

$$|F(x) - a| \le \frac{\epsilon\, a(1-a)}{1 - \epsilon \min(a, 1-a)} \le \epsilon \min(a, 1-a),$$

and the same bound holds for every value $(N_u F)(v)$ and for the mean of F. Hence F has values in $[0,1]$. If $\epsilon < 1$, it has values in $(0,1)$ and is $\frac{1+\epsilon}{1-\epsilon}$-flat, and any two among its entries, the values $(N_u F)(v)$ and its mean have ratio at most $\frac{1+\epsilon}{1-\epsilon}$, as do their complements.

Proof. Let $\|f\| = \sum_S |\widehat{f}(S)|$, so that $|f(x)| \le \|f\|$, $|(N_u f)(v)| \le \|f\|$ and $|\widehat{f}(\emptyset)| \le \|f\|$. For two functions let $f \star g = \mathcal{I}[(N_u f)(N_u g)]$. The coefficient of x^R in $f \star g$ is

$$\sum_{S \triangle S' = R} \widehat{f}(S)\, \widehat{g}(S') \prod_{i \in S \cap S'} u_i^2 :$$

each division by u_i is paid for by two factors u_i from the product. Since $u_i \le 1$,

$$\|f \star g\| \le \|f\|\, \|g\|. \tag{3}$$

Let $E = \mathcal{I}[v \mapsto e^{\beta \cdot v}]$. Since $e^{\beta \cdot v} = \prod_i (\cosh \beta_i + v_i \sinh \beta_i)$, we get $E(x) = \prod_i (\cosh \beta_i + x_i \sinh(\beta_i)/u_i)$ and

$$\|E\| = \prod_i \Big(\cosh \beta_i + \frac{|\sinh \beta_i|}{u_i} \Big) \le \prod_i e^{|\beta_i|/u_i} = e^\Lambda,$$

using $\sinh(uw) \le u \sinh w$ for $u \in (0,1]$ and $w \ge 0$. The constant coefficient of E is at least 1, so $\|E - 1\| \le \epsilon$, and $g := e^{\beta \cdot v} - 1 = N_u(E-1)$ satisfies $|g| \le \epsilon$. Let $a \le 1/2$. Then

$$\begin{aligned} \operatorname{logit}^{-1}(c_0 + \beta \cdot v) &= a + a(1-a)\frac{g}{1+ag} \\ &= a + a(1-a) \sum_{j \ge 0} (-a)^j g^{j+1}, \end{aligned}$$

where the series converges because $|ag| \le 1/2$, and $\mathcal{I}[g^{j+1}]$ is the $(j+1)$-fold $\star$-power of $E - 1$, of norm at most ϵ^{j+1} by (3). So $\|F - a\| \le a(1-a)\epsilon/(1 - a\epsilon)$. For $a > 1/2$ the same argument applied to $1 - F$ gives $a(1-a)\epsilon/(1 - (1-a)\epsilon)$. The second inequality of the lemma is equivalent to $\epsilon \le 1$. Finally, each entry of F, each value $(N_u F)(v)$ and the mean $\widehat{F}(\emptyset)$ differ from a by at most $\|F - a\| \le \epsilon \min(a, 1-a)$, so each lies in $[a(1-\epsilon), a(1+\epsilon)]$ and its complement in $[(1-a)(1-\epsilon), (1-a)(1+\epsilon)]$. □

5.3. The Constants

By (1), $C_z \ge 8$, and with $j = n - z + 1$,

$$\sum_{z=2}^{n} \frac{\chi(C_z)}{C_z} < 1.0014 \sum_{j=1}^{n-1} \frac{1}{4j(j+1)} < \frac{1.0014}{4} < \ln \frac{4}{3}.$$

Lemma 6. *Along every execution, for any pattern of aborts, every table of a run and every table handed to a single party is constant in* $\{0,1\}$*, or has values in* $(0,1)$*, satisfies* (2) *with* $\Lambda < \ln\frac{4}{3}$*, and is* 2*-flat. In the latter case its entries, the values* a_v *and its mean are within a factor* 2 *of each other, and so are their complements.*

Proof. By induction along the execution, the three properties together. The top table is $1/2 = \mathrm{logit}^{-1}(0)$ with $\Lambda = 0$. Let a run of z parties have a table with the three properties and end by an abort. A run ends at its first abort, an abort removes at least one party, and only an abort during Coin of a non-direct round adds a pending coin, exactly one. So the pending coins of the child's table were left by runs of distinct sizes, and by Lemma 4, which applies because the parent's table has values in $(0,1)$, the child's table is constant in $\{0,1\}$ or satisfies (2) with $\Lambda \le \sum_{z=2}^{n} \chi(C_z)/C_z < \ln\frac{4}{3}$. Then $\epsilon < 1/3$, and Lemma 5 gives values in $(0,1)$ and the factor $\frac{1+\epsilon}{1-\epsilon} < 2$. □

In Lemma 3 and in the following sections $B = 2$. We also record that

$$\sum_{z=2}^{n} 2^{z-1} C_z = 2^{n+2} \sum_{j=1}^{n-1} j(j+1)2^{-j} < 64 \cdot 2^{n-1}, \tag{4}$$

because $\sum_{j\ge 1} j(j+1)2^{-j} = 8$.

6. The Bias in the Hybrid Model

6.1. The Gain of One Abort

Value. Let $\mathrm{val}(V)$ be the expected common output given a partial view V, with no further aborts. Let V_i^- be V_i without the abort messages of round i. Between aborts val is a martingale and $\mathrm{val}(V_0) = 1/2$, so

$$\left|\mathbb{E}[\mathrm{out}] - \tfrac{1}{2}\right| \le \mathbb{E} \sum_{\text{aborts}} |\mathrm{val}(V_I) - \mathrm{val}(V_I^-)|.$$

This bounds the bias in the fail-stop hybrid model directly, in the stopping-game form of HT and BHLT. We call a term of the sum the *jump* of that abort.

We bound jumps for an adversary that is also given the pending coins v and the whole entry history H; by Jensen's inequality this can only increase the expected jump.

Lemma 7 (Jumps)**.** *In a run of* z *parties, consider an abort with* k *the current round, and let* $\tau_\pm = \tau(v, \mathrm{path}, \pm 1)$*. The jump of the abort is*

(a) 0 *during Coin of a non-direct round;*
(b) 0 *at Init, during Defense with no defense bit seen, and in a direct Coin before the output is seen;*
(c) *at most* $8M_z\kappa_k \cdot |\tau_+ - \tau_-|/2$ *during Defense after seeing defense bits, of which there are at most* $M_z = 2^{z-1} - 1$*;*
(d) *at most* $|\tau_+ - \tau_-|$ *in a direct Coin after seeing* r_k*;*
(e) 0 *at Release if* $S_0 \neq 0$*, and at most* 1 *if* $S_0 = 0$*.*

Consequently, apart from (e), every jump is at most $D_z\, g_k(S_k)$ *with* $D_z = 32M_zC_z$*.*

Appendix A has the proof; the reason is as follows. An abort during Coin gains nothing, because the child's table is fitted to the expected outcome of the parent: by the defense invariant and the

identity $\mathbb{E}[(\mathcal{I}f)(x) \mid v] = f(v)$, the child's expected output is τ. What gains is reading defense bits. The M_z bits of strength κ_k move the coalition's estimate of the coin by at most $8M_z\kappa_k$, and the two values of the coin differ in expected outcome by $|\tau_+ - \tau_-| \le 8\Delta_k$, where the constant comes from the flatness of the table. The jump is half the product, which with $\kappa_k = C_z\Delta_k/m_k$ is at most $32M_zC_z\, g_k$.

Lemma 8. *Given the history at the entry of a run of z parties, the expected jump of the abort ending that run is at most*

$$2\Big(D_z\,\mathbb{E}_{\mathrm{Ref}_A}\big[\sup_k g_k(S_k)\big] + \Pr_{\mathrm{Ref}_A}[S_0 = 0]\Big).$$

Proof. We couple the execution to its preselected full path; abort decisions do not change the law of the coins before them. We then bound the jump at the stopped round by the supremum over the full path, by Lemma 7, and use the density bound of Lemma 3 with $B = 2$. □

6.2. The Weighted Walk

Theorem 3 (Walk bound)**.** *There are universal constants c_{w} and c_{t} such that for all $T \ge 1$, uniformly in the reference parameters,*

$$\mathbb{E}_{\mathrm{Ref}_A}\big[\sup_{k\le T} g_k(S_k)\big] \le \frac{c_{\mathrm{w}}}{T}, \quad \Pr_{\mathrm{Ref}_A}[S_0 = 0] \le \frac{c_{\mathrm{t}}}{T^{3/2}}.$$

The second bound is an anti-concentration estimate for R_T, whose variance is of order T^3. The first has two steps, both carried out in Appendix B. A local limit estimate for R_k gives the pointwise envelope

$$g_k(S) \le \frac{c}{k}\,\psi\big(|S|/\sigma_k\big), \qquad \psi(\zeta) = (1+\zeta)e^{-\zeta^2/16}, \tag{5}$$

for all k, S and α (Lemma 19): with k rounds left, g_k is of order $1/k$ within a few standard deviations σ_k of the center and decays like a Gaussian outside. A block argument then bounds the supremum over the path. For k in a dyadic block $(K/2, K]$ the walk is within $O(\sigma_K)$ of the center with probability $O(\sigma_K/\sigma_T) = O((K/T)^{3/2})$, so the block contributes $O(K^{-1}(K/T)^{3/2})$, and the sum over the blocks is $O(1/T)$.

6.3. The Hybrid Theorem

Theorem 4 (Hybrid model)**.** *For every $n \ge 2$ and $T \ge 1$, in the fail-stop hybrid model of Section 4.2 all honest parties end with a common bit, which is uniform when no party aborts, and against every coalition of at most $n-1$ parties, computationally bounded or not,*

$$\big|\mathbb{E}[\mathrm{out}] - \tfrac{1}{2}\big| \le \frac{2048\,c_{\mathrm{w}}\,2^n}{T} + \frac{2(n-1)\,c_{\mathrm{t}}}{T^{3/2}} \le \frac{c\,2^n}{T},$$

with $c = 2048c_{\mathrm{w}} + c_{\mathrm{t}}$. The protocol makes at most $2T+2$ calls per run and has at most n runs in an execution.

Proof. Correctness. An all-honest execution is the top run with $A = 1/2$ and outputs b, uniform. All honest parties learn the same aborts by broadcast and hold the same public state. An honest party that happened to receive a coin before an abort ignores it, following the abort rule. If several honest parties remain they stay in the same run; at the end they output b, or a single party outputs its bit. All tables are valid by Lemma 6.

Bias. The runs of an execution have strictly decreasing sizes, and a single party makes no calls. So for each $z \in \{2, \ldots, n\}$ at most one abort ends a run of size z; let J_z be its jump, or 0 if there is none. The entry of that run is a stopping time, so by Lemma 8 and Theorem 3, with $D_z = 32 M_z C_z$ and $M_z < 2^{z-1}$,

$$\mathbb{E}|J_z| \leq \frac{64 \cdot 2^{z-1} C_z \, c_{\mathrm{w}}}{T} + \frac{2 c_{\mathrm{t}}}{T^{3/2}}.$$

Summing over z with (4) gives $64 \cdot 64 \cdot 2^{n-1} c_{\mathrm{w}}/T + 2(n-1) c_{\mathrm{t}} T^{-3/2}$, which is the first bound. The second follows from $2(n-1) \leq 2^n$ and $T^{-3/2} \leq T^{-1}$. □

7. The Real Protocol

We now compile the hybrid-model protocol into a protocol over broadcast that is secure against malicious static coalitions, and prove Theorem 1. Every party carries its state between calls under one-time signatures (Section 7.1), each functionality becomes one secure computation with identified abort [37] (Section 7.2), and the calls are composed sequentially (Section 7.3).

7.1. Signed State

We turn each functionality of Section 4.2 into a signed one, which takes each party's current state tuple as input and returns its new state tuple, signed. Every signed tuple gets its own fresh one-time signature key [39], [44], generated inside the functionality and never output; its verification key is a public output. The signed message is the whole tuple, padded to a fixed length, with session, call, branch, owner and slot. The *registry*, a deterministic function of the public history, maps each input slot of the next call to its producing call, owner, version and verification key, so a party cannot select an older valid tuple. The functionality checks every supplied tuple against the registry; if tuples fail, its public output is $\mathrm{fault}(i)$ for each failing i, and $\mathrm{fault}(i)$ is treated as an abort by i. The registry advances only when a call commits, so "current" means the output of the last committed call, and the defense bits of a committed Defense call stay in force through its Coin.

Claim 1. *For every polynomial-time adversary in the signed hybrid, and every expected polynomial-time one by truncation, there is a fail-stop adversary in the hybrid of Section 4.2 with the same joint law of view and honest outputs, except on a forgery and on the sampling error of the functionalities, both negligible.*

Proof. Honest tuples always verify, since a committed call delivered them with valid signatures. A corrupted party's supplied tuple either verifies under the authorized key, and then, absent a forgery, it is the tuple that was signed, so the call receives the same input as in Section 4.2; or it fails or is missing, which is an abort by that party at that call. So the only remaining choice is when to abort. □

7.2. One Call with Identified Abort

Fix a call with set Z of z parties, corrupted set $\mathcal{C} = Z \cap \text{coalition}$, honest set $\mathcal{H} = Z \setminus \mathcal{C} \neq \emptyset$, and a polynomial-time randomized f with public output w and private outputs w_i. In the ideal process, the simulator Sim sends inputs for $\mathcal{C}$ to a trusted party TP, which computes f with fresh randomness on these and the honest inputs, and gives Sim the output w and $(w_i)_{i \in \mathcal{C}}$. At any time

Sim may send (abort, i) with $i \in \mathcal{C}$, also before sending inputs; then TP sends (abort, i) to all honest parties and stops. Otherwise TP delivers all outputs. This is the hybrid call of Section 4.2.

Lemma 9. *Assume constant-round OT and Assumption 1. There is a universal constant C_L such that the protocol of Appendix C runs over broadcast in at most $C_L z^2$ rounds and realizes this ideal process with auxiliary input: for every polynomial-time $\mathcal{A}$ there is an expected polynomial-time* Sim *such that for all inputs and auxiliary strings, (view of $\mathcal{A}$, honest outputs) in the real protocol and (output of* Sim*, honest outputs) in the ideal process are computationally indistinguishable. In the real protocol every abort names a member of $\mathcal{C}$, except with negligible probability. The running time of* Sim *and all losses are polynomial in z, λ and the size of f.*

The protocol is commit-and-prove over a passive protocol. Each party commits to its input, an output pad, randomness and tape masks under receiver strings from all parties. The passive protocol of Assumption 1, applied to a public-output version of f, then runs in the open, every message followed by Feige–Shamir arguments [29], [30] that it is the prescribed one, run one at a time in a fixed order over the $z(z-1)$ ordered pairs; this is the source of the factor z^2. Public verification names whoever fails. The simulator extracts the corrupted inputs, calls TP once, and programs one honest party's masks so that the corrupted tapes match the passive simulator's output. Appendix C gives the protocol and the proof.

7.3. Composition

Calls are sequential, with unique labels, and the protocol's other messages are public. We build the simulator call by call: it keeps $\mathcal{A}$'s main state, simulates the current call by Lemma 9, and never rewinds across a call boundary. With a uniform per-call expected bound P and at most $n(2T+2)$ calls, the total expected time is at most $n(2T+2)P$. This is Canetti's modular composition [18] for the stand-alone model with auxiliary input, applied one call at a time, and it turns any real $\mathcal{A}$ into a signed-hybrid adversary whose view and honest outputs are computationally close, with total error at most the sum of the per-call errors. Claim 1 turns it into a fail-stop adversary of Section 4.2.

The simulators run in expected polynomial time, so for every polynomial p we truncate at $2n(2T+2)Pp$ steps, which by Markov's inequality loses at most $1/(2p)$. Theorem 4 holds against unbounded adversaries, so for every polynomial p

$$\left|\Pr[\mathrm{out}=1] - \tfrac{1}{2}\right| \le \frac{c\,2^n}{T} + \frac{1}{p} + \mathrm{negl}_p(\lambda).$$

Neither the left-hand side nor the first term on the right depends on p, so the excess over $c\,2^n/T$ is negligible. A run of z parties makes at most $2T+2$ calls of at most $C_L z^2$ rounds, so the protocol has at most $\sum_{z=2}^{n}(2T+2)C_L z^2 = O(n^3 T)$ rounds. Every party outputs a fixed function of the broadcast transcript (the terminal rule of Appendix D), so all honest parties terminate with the same bit on every transcript.

For fixed n the functionalities keep all tables as exact rationals; the recursion has constant depth, so bit lengths stay polynomial in T, and conditional paths and biased bits are sampled in polynomial time with a negligible statistical error (Appendix G). This proves Theorem 1 for fixed n. For the last statement of the theorem, let $K_n = c\,2^n$: if $r \le 2K_n$ the claim is trivial, and otherwise $T = \max(r, \lceil 4K_n \rceil)$ gives $T = \Theta_n(r)$ and $K_n/T \le K_n/r$.

The theorem states bias. For the simulation-based notion of fairness, Lemma 2.15 of BHLT [16] (Lemma 2.18 of HT [32]), applied in the hybrid of Section 4.2 with $T \ge 4K_n$, gives $(K_n/T + \mathrm{negl})$-

fairness against fail-stop adversaries there, and the simulator above carries it to the real protocol. Appendix I gives the proof, also for $n = n(\lambda)$.

7.4. A Growing Number of Parties

When n grows with λ, exact means would make the bit lengths of the tables compound from level to level. We therefore let every run except the top one round its reference parameter.

Rule R. A run determines the atom of R_T and the pair (t_0, α) from the mean $\hat{A}$ of its table as before, and then replaces α by $\tilde{\alpha}$: the multiple of $2^{-(T+\lambda)}$ nearest to α, or, at the two extreme atoms of R_T, the value of α (respectively of $1-\alpha$) rounded down to λ significant bits. The run uses the reference parameters $(t_0, \tilde{\alpha})$, and $A := y_T(t_0)$ is computed with $\tilde{\alpha}$.

Lemma 10 (Rounded reference parameters)**.** *Under Rule R, $0 < A < 1$, and $A/\hat{A}$ and $(1-A)/(1-\hat{A})$ lie in $[1-2^{-\lambda}, 1+2^{-\lambda}]$. Lemma 6 holds unchanged, the four ratios a_v/A, A/a_v, $(1-a_v)/(1-A)$ and $(1-A)/(1-a_v)$ are at most $B' = 2/(1-2^{-\lambda})$, and Theorem 4 holds with its bound multiplied by $(1-2^{-\lambda})^{-3}$.*

Lemma 11 (Bit lengths)**.** *Let $\lambda \geq 2$. Under Rule R, every value $y_k(S)$, every strength and every table entry is a rational whose numerator and denominator have $O(2^n n(T+n\lambda))$ bits, and the functionalities run in time polynomial in T, 2^n and λ.*

Rule R changes each reference parameter by a relative $2^{-\lambda}$, which the flatness bounds absorb. For the bit lengths, the odds of a posterior a_v are a product, over the at most $n-1$ ancestors, of ratios of odds of walk values, by Lemma 4, so bit lengths add from level to level instead of compounding. Appendix H states Rule R in full and has both proofs.

Proof of Theorem 1. In the hybrid model the bound is Theorem 4 with Lemma 10; the top run keeps $(t_0, \alpha) = (0, 1/2)$, so the output is uniform when no party aborts. The functionalities are polynomial-time by Lemma 11, since $2^n \leq \mathrm{poly}(\lambda)$. Lemma 9 and the composition of Section 7.3 hold for $z \leq n$ growing with λ (Appendix I): every loss is polynomial in z, there are at most $n(2T+2)$ calls, and a state tuple has $O(2^z n + T)$ bits. The negligible term is a sum of polynomially many negligible terms, and the number of rounds is $O(n^3 T)$ as computed above. □

8. Any Number of Parties

There are N parties, of which at most t are corrupted, with $N/2 \leq t \leq N-1$; recall $k = 2t-N$ and $s = N-k-1 = 2(N-t)-1$, an odd number, and let $\mu = (s-1)/2$, so that a strict majority of s parties is $\mu+1 = N-t$. The parties emulate the $(k+2)$-party protocol of Theorem 1, which tolerates $k+1$ corruptions: $k+1$ parties play one emulated party each, and the remaining s parties form a group that plays the last emulated party jointly (Fig. 5). Emulating a party by a group that runs a secure computation is the player-simulation technique of Hirt and Maurer [34], and Beimel et al. [8] already partition the parties in this way: into two sets that simulate the two-party protocol of Moran et al. (their Section 3.1 and Remark 3.1), and into $k+2$ sets in their main construction (Remark 5.1). What Theorem 1 adds is a $(k+2)$-party protocol to emulate whose bias is singly exponential in k; the emulation applies to any protocol with the properties (P1) to (P3) below.

8.1. The Protocol $\Pi^\star$

Let Π be the protocol of Theorem 1 for $n' = k+2$ parties $V_1, \ldots, V_{n'}$ with walk length T', padded to exactly $\rho \leq C_1 n'^3 T'$ rounds. We use three properties of Π.

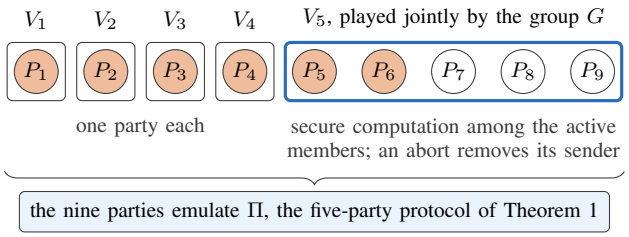

Figure 5. The emulation for $N = 9$ parties and $t = 6$ corruptions, so that $k = 3$ and $s = 5$. Four parties play one emulated party each and five play V_5 jointly. Controlling all five emulated parties needs $4 + 3 = 7 > t$ corruptions, so one emulated party is honest in every case; in the case shown it is V_5.

(P1) Π has no inputs and all its messages are broadcast: the message of V_i in round ℓ is $\mathsf{NM}_i(\omega_i, \mathrm{tr}_{<\ell})$, a deterministic polynomial-time function of its random tape and of the broadcast transcript of the earlier rounds.

(P2) The output of every party is $\mathsf{Out}(\mathrm{tr})$, one polynomial-time function of the whole broadcast transcript (the terminal rule).

(P3) The schedule and the slots are functions of the transcript, and a missing or malformed message in a slot is an abort naming its sender.

Roles. Each P_i with $i \le k+1$ plays V_i with its own tape ω_i. The group $G = \{P_{k+2}, \ldots, P_N\}$ of size s plays $V_{n'}$. If $s = 1$ there is no group and $\Pi^\star = \Pi$; let $s \ge 3$.

Group calls. The group keeps a public *active set* $\mathsf{Act} \subseteq G$, initially G. To compute a function f, $\mathsf{Call}(f)$ runs the protocol of Lemma 9 for f among the members of Act, under a fresh label. If it commits, its outputs are the outputs of the call. If it aborts naming i, then i is removed from Act and the protocol is run again; if $|\mathsf{Act}| \le \mu$, the call fails and $V_{n'}$ is declared dead. The active set, the aborts and the declaration are functions of the broadcast transcript, so all parties agree on them. Each abort removes a member, so the whole execution has at most $\rho + 1 + s$ runs of the protocol of Lemma 9.

Setup. $\mathsf{Call}(f_0)$, where f_0 has no inputs: it samples a tape ω for $V_{n'}$, shares it by Shamir's scheme [47] with threshold μ, so that any μ shares are independent of ω and any $\mu + 1$ determine it, samples a one-time signature key pair for every member, and gives member j privately its share with a signature on it, and all members the verification keys.

Epoch $\ell = 1, \ldots, \rho$**.** (a) $\mathsf{Call}(f_\ell)$, where the public data $\mathrm{tr}_{<\ell}$ and the verification keys are hardwired into f_ℓ and member j inputs its signed share: f_ℓ reconstructs ω from the $\mu + 1$ smallest indices whose signatures verify, or sets $\omega = 0$ if there are fewer, and outputs publicly $\mathrm{msg} = \mathsf{NM}_{n'}(\omega, \mathrm{tr}_{<\ell})$. The other parties are silent during (a). (b) In one announcement round, every member of G broadcasts msg, or $\perp$ if it holds none, and every P_i with $i \le k+1$ broadcasts $\mathsf{NM}_i(\omega_i, \mathrm{tr}_{<\ell})$. (c) Every party decodes the round-ℓ message of $V_{n'}$ as the value announced by at least $\mu + 1$ members of G, which is unique, or as a missing message if there is none.

Output. After epoch ρ every party outputs $\mathsf{Out}(\mathrm{tr})$. A run of the protocol of Lemma 9 among at most s parties has at most $C_L s^2$ rounds, so $\Pi^\star$ has at most $C_L(\rho + 1 + s)s^2 + \rho$ rounds. All honest parties decode the same transcript and output the same bit, and in an all-honest execution no call aborts and the transcript is that of an all-honest execution of Π up to a negligible error.

8.2. Why the Emulation Is as Fair as Π

Fix an adversary that corrupts a set I of at most t parties. It controls the emulated party V_i, $i \le k+1$, when $P_i \in I$, and it controls $V_{n'}$ when it holds a majority of the group.

Lemma 12 (Counting). *Let* $J = \{i \le k+1 : P_i \in I\} \cup \{n' : |I \cap G| \ge \mu + 1\}$. *Then* $|J| \le n' - 1$.

Proof. If $J = \{1, \ldots, n'\}$, then $|I| \ge (k+1) + (\mu+1) = (2t - N + 1) + (N - t) = t + 1$. $\square$

It remains to turn the adversary into an adversary against Π that corrupts J and biases Π as much. If the group has a corrupted majority, the new adversary plays the honest members of the group itself. If the group has an honest majority, three facts make the group behave as an honest $V_{n'}$: an abort inside a call names a corrupted member, so the honest members are never removed and every call ends with delivery; the signatures force every epoch to reconstruct the tape that the setup sampled; and the at most μ shares of the corrupted members are independent of that tape. Appendix E gives the proof.

8.3. Dropping the Factor s^2

The factor s^2 is the number of rounds of one group call. It disappears if the group has a constant-round protocol with guaranteed output. Assumption 2, stated in full in Appendix F, asks for an s-party protocol with an absolute constant number q_G of rounds, over private channels and broadcast, that computes every polynomial-size functionality with guaranteed output delivery against a malicious minority, in the stand-alone sense with auxiliary input. The constant-round protocol of Beaver, Micali and Rogaway [6] for an honest majority is the natural candidate. Let $\Pi^{\star}_{\mathrm{X}}$ be $\Pi^{\star}$ with each group call replaced by one execution of this protocol among all members of G, where the private channels are pads over broadcast that each execution sets up in C_0 rounds, for an absolute constant C_0.

Theorem 5. *Assume a constant-round protocol for oblivious transfer and Assumptions 1 and 2. In the setting of Theorem 2, the protocol* $\Pi^{\star}_{\mathrm{X}}$ *has at most* $r \le (q_G + C_0)(\rho + 1) + \rho$ *rounds, all honest parties terminate with a common bit that is uniform up to a negligible term when all parties are honest, and against every static polynomial-time coalition of at most* t *parties* $|\Pr[\mathrm{out} = 1] - 1/2| \le c\, 2^{k+2}/T' + \mathrm{negl}(\lambda)$. *If* $\rho \ge q_G + C_0$, *this is at most* $c''(k+2)^3 2^{k+2}/r + \mathrm{negl}(\lambda)$ *with* $c'' = cC_1(q_G + C_0 + 2)$.

Comparison with Beimel, Omri and Orlov. For $N/2 \le t < 2N/3$, Beimel et al. [8] obtain bias $O(2^{2^{k+1}}/r')$ with $r' = r - O(k+1)$. For constant k this is $O(1/r)$, which Theorem 5 matches and Theorem 2 matches up to the factor s^2. Theorem 2 gives the smaller bias for $k \ge \log_2 \log_2 s + O(1)$ (Appendix K), and it is below the $O(t/\sqrt{r})$ of majority-type protocols when $(k+2)^3 2^k s = o(\sqrt{r})$.

9. Implementation

We implemented the protocol for fail-stop adversaries and compared it with the majority protocol of Awerbuch et al. [1] under the same best fail-stop attack.

The prototype. Two facts, proved in Appendix M, make the protocol small enough to run. First, the draw of the whole path at Init can be replaced by one draw per round without changing the joint law of the execution, so that each call is a small circuit and, besides the shares of the

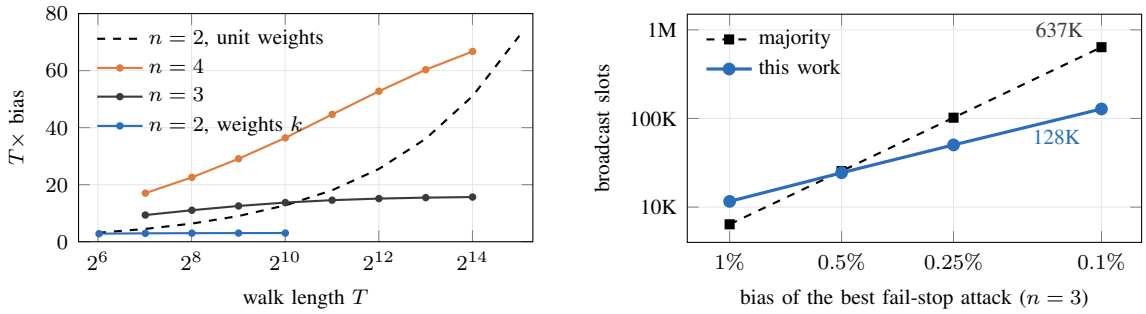

Figure 6. Left: bias of the best fail-stop attack, times the walk length T; a level curve means the rate $1/T$. Right: broadcast slots of a run without aborts at three parties for equal bias under the best fail-stop attack (values of Table 3); the curves cross at a bias of about 0.5%.

defense bits, the only secrets carried between calls are b and, between Defense and Coin, the current coin. Second, against fail-stop adversaries each call can run as GMW [31] over XOR shares with Beaver triples [5]. The prototype is written in Python, with n party processes and a broadcast board on one 16-core machine, where a *slot* is one round of the broadcast channel; Appendix L describes it and lists measured runs.

The prototype and the computations below set the strengths with the constants $C_2 = 1$ at two parties, $(C_3, C_2) = (3/2, 1)$ at three and $(C_4, C_3, C_2) = (3, 2, 1)$ at four, which are smaller than those of (1); every recovery table that occurred in the runs had values in $[0, 1]$. A coin takes at most 9 slots, and the average at $T = 4097$ is about 8.2 at $n = 3$ and 8.5 at $n = 4$. A Defense call has at most 1409 AND gates, at depth 7. At three parties and $T = 4097$, a run takes 33.5K slots and 29.7 s online, after 50.4 s of offline preparation.

The bias of the best attack. The bias values are those of the best fail-stop attack in the hybrid model, computed by backward induction at $n = 2$ and by an approximate program at $n = 3, 4$. Fig. 6 (left) shows $T \cdot \text{bias}$. At two parties it reaches 3.04 at $T = 1025$ and its local slope halves with each doubling of T, which points to a limit near 3.06. With unit weights in place of the weights k, $\sqrt{T} \cdot \text{bias}$ tends to 0.399, so that the rate is $1/\sqrt{T}$: the weights are what turns the rate of majority into the rate of Cleve's bound. At three parties $T \cdot \text{bias}$ is 15.5 at $T = 8193$, a bias of 0.19%, while at four parties the curve is still bending toward level at $T = 16385$, where it is 66.7, a bias of 0.41%.

Against majority. In the majority baseline each party posts a random bit per coin, the coin is their XOR, the output is the majority of the coins, the remaining parties re-toss a coin at which parties abort, and a single remaining party finishes with its own coins. Majority takes one slot per coin and has no offline phase. Table 3 and Fig. 6 (right) give the slots for equal bias; at four parties the crossover is lower still. Past the crossover the gap grows like $1/\epsilon$, as $1/r$ against $1/\sqrt{r}$ predicts. Aborts add at most $n - 2$ further walks, so at three parties at most twice our slots, which at 0.1% is 257K against 637K. When slots are slow, wall-clock time follows slots: at 50 ms per slot the same point would take about 1.8 h for ours and 8.8 h for majority. When slots are as fast as on one machine, the offline triples dominate and the two are about even, apart from a one-time computation of the public tables.

TABLE 3. BROADCAST SLOTS OF A RUN WITHOUT ABORTS, FOR EQUAL BIAS UNDER THE BEST FAIL-STOP ATTACK, COMPUTED. OURS: WALK LENGTH T INTERPOLATED FROM THE VALUES OF FIG. 6 (LEFT), TIMES 8.2 OR 8.5 SLOTS PER COIN. MAJORITY: $((n-1)/(\sqrt{2\pi}\,\epsilon))^2$ COINS AT ONE SLOT EACH, WHICH AGREES WITH EXACT DYNAMIC PROGRAMMING. RATIO: MAJORITY'S SLOTS OVER OURS.

n	bias	ours: T	slots	majority	ratio
3	1%	1,411	11.6K	6.4K	0.55
3	0.5%	2,972	24.4K	25.5K	1.04
3	0.25%	6,132	50.3K	102K	2.03
3	0.1%	15,652	128K	637K	4.96
4	1%	5,605	47.6K	14.3K	0.30
4	0.5%	12,879	109K	57.3K	0.52

10. Related Work

Lower bounds. Cleve and Impagliazzo [24] showed a bias of $\Omega(1/\sqrt{r})$ for two-party protocols against unbounded fail-stop adversaries in the plain model.

Drawing the outcome first. Moran et al. [42] and Beimel et al. [8] also draw the output first, but they reveal uniform noise until a hidden round, so that the expected outcome jumps by a constant there. Luby, Micali and Rackoff [40] and Cleve [23] disclose a fixed bit gradually through correlated public signals, for exchanging secrets and with no common output after an abort. Compared with HT and BHLT, our protocol differs in three places: the path depends on hidden values only through the outcome, the material of a recovery is one fresh bit per subset per round, and the strengths follow Δ/m with constants independent of the length of the walk. The inverse noise operator is textbook [45]; what we add is its use on a public table, with a bound free of the strengths and of the number of coordinates.

Randomness beacons. Threshold beacons [17], [27], [46], [48] have negligible bias against coalitions below an honest threshold, and NIST specifies a beacon that is run by a single trusted service [38]. With a dishonest majority, beating Cleve's bound requires delay functions [15], which also suffice [4]. Financial penalties deter aborts under rationality [3], [11], and game-theoretic fairness [21] asks that no coalition profits under given utilities. Our protocol is the remaining option: it adds no assumption to those of secure computation and pays with the bias that Cleve's bound makes unavoidable.

11. Conclusion and Open Problems

Cleve's bound is tight for every fixed number of parties: rounds alone bring the bias of a coin flip down to $c\,2^n/r$ when all parties but one collude. The protocol that achieves this draws its outcome first and keeps it hidden, and its analysis rests on one fact: seen through the noise, a recovery table is a posterior, and such tables stay flat however many aborts came before.

The dependence on the number of parties. The factor 2^n counts the defense bits that a coalition reads in one call. Whether it is necessary, and what the best bias is when n lies between $\frac{1}{2}\log_2 r$ and the range of the lower bound of Beimel et al. [7], is not known.

Corruptions and assumptions. Our theorems cover static corruptions. Whether adaptive corruptions can be handled at the same rate in the real protocol is open, and so is whether the rate of Theorem 5 holds without Assumption 2.

Direct protocols. Against malicious adversaries each call runs a generic secure computation; a direct protocol for Defense and Coin would bring the cost closer to that of the fail-stop prototype.

References

[1] B. Awerbuch, M. Blum, B. Chor, S. Goldwasser, and S. Micali, "How to implement Bracha's $O(\log n)$ Byzantine agreement algorithm," manuscript, 1985.

[2] B. Alon and E. Omri, "Almost-optimally fair multiparty coin-tossing with nearly three-quarters malicious," *J. Cryptology*, vol. 36, art. 24, 2023.

[3] M. Andrychowicz, S. Dziembowski, D. Malinowski, and Ł. Mazurek, "Secure multiparty computations on Bitcoin," in *IEEE S&P*, 2014, pp. 443–458.

[4] M. Ball, M. Christ, Y. Dodis, and R. Garg, "Fair multiparty coin tossing from minimal assumptions," in *EUROCRYPT*, 2026, pp. 245–272.

[5] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *CRYPTO*, 1991, pp. 420–432.

[6] D. Beaver, S. Micali, and P. Rogaway, "The round complexity of secure protocols," in *STOC*, 1990, pp. 503–513.

[7] A. Beimel, I. Haitner, N. Makriyannis, and E. Omri, "Tighter bounds on multiparty coin flipping via augmented weak martingales and differentially private sampling," *SIAM J. Comput.*, vol. 51, no. 4, pp. 1126–1171, 2022.

[8] A. Beimel, E. Omri, and I. Orlov, "Protocols for multiparty coin toss with a dishonest majority," *J. Cryptology*, vol. 28, no. 3, pp. 551–600, 2015.

[9] A. Ben-Efraim, Y. Lindell, and E. Omri, "Optimizing semi-honest secure multiparty computation for the Internet," in *ACM CCS*, 2016, pp. 578–590.

[10] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *STOC*, 1988, pp. 1–10.

[11] I. Bentov and R. Kumaresan, "How to use Bitcoin to design fair protocols," in *CRYPTO*, 2014, pp. 421–439.

[12] M. Blum, "Coin flipping by telephone: A protocol for solving impossible problems," *ACM SIGACT News*, vol. 15, no. 1, pp. 23–27, 1983.

[13] M. Blum, "How to prove a theorem so no one else can claim it," in *Proc. ICM*, 1986, pp. 1444–1451.

[14] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch, "Verifiable delay functions," in *CRYPTO*, 2018, pp. 757–788.

[15] J. Bonneau, B. Bünz, M. Christ, and Y. Efron, "Good things come to those who wait: Dishonest-majority coin-flipping requires delay functions," in *EUROCRYPT*, 2025, pp. 225–253.

[16] N. Buchbinder, I. Haitner, N. Levi, and E. Tsfadia, "Fair coin flipping: Tighter analysis and the many-party case," *J. Cryptology*, vol. 39, art. 4, 2026; preliminary version in *SODA*, 2017, pp. 2580–2600.

[17] C. Cachin, K. Kursawe, and V. Shoup, "Random oracles in Constantinople: Practical asynchronous Byzantine agreement using cryptography," *J. Cryptology*, vol. 18, no. 3, pp. 219–246, 2005.

[18] R. Canetti, "Security and composition of multiparty cryptographic protocols," *J. Cryptology*, vol. 13, no. 1, pp. 143–202, 2000.

[19] K. Choi, A. Manoj, and J. Bonneau, "SoK: Distributed randomness beacons," in *IEEE S&P*, 2023, pp. 75–92.

[20] S. G. Choi, J. Katz, R. Kumaresan, and H.-S. Zhou, "On the security of the "free-XOR" technique," in *TCC*, 2012, pp. 39–53.

[21] K.-M. Chung, Y. Guo, W.-K. Lin, R. Pass, and E. Shi, "Game theoretic notions of fairness in multi-party coin toss," in *TCC*, 2018, pp. 563–596.

[22] R. Cleve, "Limits on the security of coin flips when half the processors are faulty," in *STOC*, 1986, pp. 364–369.

[23] R. Cleve, "Controlled gradual disclosure schemes for random bits and their applications," in *CRYPTO*, 1989, pp. 573–588.

[24] R. Cleve and R. Impagliazzo, "Martingales, collective coin flipping and discrete control processes," manuscript, 1993.

[25] R. Cohen, I. Haitner, E. Omri, and L. Rotem, "Characterization of secure multiparty computation without broadcast," in *TCC*, 2016, pp. 596–616.

[26] D. Dachman-Soled, Y. Lindell, M. Mahmoody, and T. Malkin, "On the black-box complexity of optimally-fair coin tossing," in *TCC*, 2011, pp. 450–467.

[27] S. Das, V. Krishnan, I. M. Isaac, and L. Ren, "SPURT: Scalable distributed randomness beacon with transparent setup," in *IEEE S&P*, 2022, pp. 2502–2517.

[28] C.-G. Esseen, "On the Kolmogorov–Rogozin inequality for the concentration function," *Z. Wahrscheinlichkeitstheorie*, vol. 5, pp. 210–216, 1966.

[29] U. Feige and A. Shamir, "Zero knowledge proofs of knowledge in two rounds," in *CRYPTO*, 1989, pp. 526–544.

[30] U. Feige and A. Shamir, "Witness indistinguishable and witness hiding protocols," in *STOC*, 1990, pp. 416–426.

[31] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *STOC*, 1987, pp. 218–229.

[32] I. Haitner and E. Tsfadia, "An almost-optimally fair three-party coin-flipping protocol," *SIAM J. Comput.*, vol. 46, no. 2, pp. 479–542, 2017.

[33] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A pseudorandom generator from any one-way function," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1364–1396, 1999.

[34] M. Hirt and U. Maurer, "Player simulation and general adversary structures in perfect multiparty computation," *J. Cryptology*, vol. 13, no. 1, pp. 31–60, 2000.

[35] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Statist. Assoc.*, vol. 58, no. 301, pp. 13–30, 1963.

[36] R. Impagliazzo and M. Luby, "One-way functions are essential for complexity based cryptography," in *FOCS*, 1989, pp. 230–235.

[37] Y. Ishai, R. Ostrovsky, and V. Zikas, "Secure multi-party computation with identifiable abort," in *CRYPTO*, 2014, pp. 369–386.

[38] J. Kelsey, L. T. A. N. Brandão, R. Peralta, and H. Booth, "A reference for randomness beacons: Format and protocol version 2," NIST, NISTIR 8213 (draft), 2019.

[39] L. Lamport, "Constructing digital signatures from a one way function," SRI International, Tech. Rep. SRI-CSL-98, 1979.

[40] M. Luby, S. Micali, and C. Rackoff, "How to simultaneously exchange a secret bit by flipping a symmetrically-biased coin," in *FOCS*, 1983, pp. 11–22.

[41] H. K. Maji and M. Wang, "Black-box use of one-way functions is useless for optimal fair coin-tossing," in *CRYPTO*, 2020, pp. 593–617.

[42] T. Moran, M. Naor, and G. Segev, "An optimally fair coin toss," *J. Cryptology*, vol. 29, no. 3, pp. 491–513, 2016.

[43] M. Naor, "Bit commitment using pseudorandomness," *J. Cryptology*, vol. 4, no. 2, pp. 151–158, 1991.

[44] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," in *STOC*, 1989, pp. 33–43.

[45] R. O'Donnell, *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[46] P. Schindler, A. Judmayer, N. Stifter, and E. Weippl, "HydRand: Efficient continuous distributed randomness," in *IEEE S&P*, 2020, pp. 73–89.

[47] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[48] E. Syta, P. Jovanovic, E. Kokoris-Kogias, N. Gailly, L. Gasser, I. Khoffi, M. J. Fischer, and B. Ford, "Scalable bias-resistant distributed randomness," in *IEEE S&P*, 2017, pp. 444–460.

[49] T. Chou and C. Orlandi, "The simplest protocol for oblivious transfer," in *LATINCRYPT*, 2015, pp. 40–58.

[50] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, "Extending oblivious transfers efficiently," in *CRYPTO*, 2003, pp. 145–161.

Appendix A.
Proof of Lemma 7

Conditioning. An adversary may abort before seeing a Coin output, and then it does not know the child's newest pending coin. At a decision point let b^- be the output if the current run continues with no further aborts, and b^+ the output if the chosen abort happens and the child continues with no further aborts. Let $W = V \vee \sigma(v, H)$. The outgoing abort decision adds nothing to the adversary's own view, so $J_V := \mathbb{E}[b^+ - b^- \mid V] = \mathbb{E}[J_W \mid V]$ with $J_W := \mathbb{E}[b^+ - b^- \mid W]$, and Jensen's inequality gives $\mathbb{E}|J_V| \le \mathbb{E}|J_W|$. We bound J_W. W does not contain the current unrevealed coin.

Lemma 13. *In a run let V contain (v, H), the revealed coins, the adversary's shares and the bits of its all-corrupted subsets. With no defense bit about an unrevealed coin in V,* $\mathrm{val}(V) = \tau(v, \mathrm{path})$. *With such bits d about an unrevealed r_k,* $\mathrm{val}(V) = \mathbb{E}[\tau(v, \mathrm{path}, r_k) \mid v, \mathrm{path}, d]$.

Proof. Given (v, H) the path depends on h only through b, and the adversary's other bits are independent of h by the invariant. Shares of sets with an honest member are uniform, and defense bits about earlier coins are independent of the rest given those coins. □

Proof of Lemma 7. (a) The value before is $\mathbb{E}[\tau(v, \mathrm{path}, r_k) \mid V]$, by Lemma 13. After the abort, the child's output has conditional expectation $\mathbb{E}[F'(x, x') \mid v, r_k, V] = \tau(v, \mathrm{path}, r_k)$, by the invariant and the inversion identity. Adaptive choice of Z' is covered, since every surviving set has an honest member. (b) follows by the same argument. (c) The jump is $|\mathbb{E}[r_k \mid v, \mathrm{path}, d] - \mathbb{E}[r_k \mid v, \mathrm{path}]| \cdot |\tau_+ - \tau_-|/2$, with d the bits seen. There are at most M_z such bits. Their likelihood ratio between $r_k = +1$ and $r_k = -1$ lies between $((1-2\kappa)/(1+2\kappa))^{M_z}$ and its inverse. For $\kappa \le 1/4$ the posterior mean moves by at most $4M_z\kappa$, uniformly in the prior; for $\kappa > 1/4$ use $2 \le 8\kappa$. So the first factor is at most $8M_z\kappa_k$. (d) The values before and after the abort both lie between τ_- and τ_+. (e) If $S_0 \neq 0$, then τ is 0 or 1 and b is known.

For the last statement, by Lemma 2, $\operatorname{logit}\tau = \operatorname{logit} y + \xi$ with $\xi = \operatorname{logit} a_v - \operatorname{logit} A$, and $|\xi| \le 2\ln 2$ by Lemma 6. So $y \mapsto \tau$ has derivative at most 4 and $|\tau_+ - \tau_-| \le 8\Delta_k$. In (c), $\kappa_k = C_z\Delta/m$, so the jump is at most $32M_zC_z\Delta^2/m$. In (d), if $\Delta > 0$ then $\Delta/m \ge 1/(2C_z)$ and the jump is at most $16C_z\Delta^2/m$; if $\Delta = 0$ the jump is 0. □

Appendix B.
Supporting Estimates for the Walk

Lemma 14 (Sum)**.** *There is N_0 such that for $N \ge N_0$, every arithmetic progression $J \subseteq \{1, \dots, N\}$ of difference $d \in \{1, 2\}$ containing all terms of its class in $(N/2, N]$, and $0 \le |\theta| \le 1$ (or $|\theta| \le \pi/2$ when $d = 1$): $\sum_{i \in J} \sin^2(\theta i) \ge c \min(\theta^2 N^3, N)$.*

Proof. Take $\theta > 0$. For $\theta \le \pi/(2N)$, $\sin(\theta i) \ge 2\theta i/\pi$, and half of J's mass of i^2 lies in $(N/2, N]$. For $\theta \ge C/N$, $|\sum_{i \in J} \cos(2\theta i)| \le 1/|\sin(d\theta)| \le |J|/4$, as $\sin(d\theta) \ge c\theta$ on the stated range, so the sum is at least $|J|/4$. In between, $a = \theta N$ lies in the compact $[\pi/2, C]$: by Riemann sums, $(1/N)\sum_J \sin^2(ai/N) \to (1/d)\int_I \sin^2(ax)\,dx$ with I an interval of length at least $1/2$, and this integral has a positive minimum over a. The Riemann error is $O(C/N)$, so N_0 large suffices. □

Lemma 15 (Anti-concentration)**.** *For J as in Lemma 14, or $J = (K, T]$ with $K \le T/2$, and $\ell \ge 1$:* $\sup_x \Pr[\sum_{i \in J} \varepsilon_i i \in [x, x + \ell]] \le c\,\ell/\sigma_J$, *where $\sigma_J^2 = \sum_{i \in J} i^2$.*

Proof. For $N < N_0$ the claim is trivial by the choice of c. Otherwise use Esseen's concentration inequality [28]: $\Pr \le c\,\ell \int_{|\theta|\le 1/\ell} \prod_{i\in J} |\cos(\theta i)|\, d\theta$, together with $|\cos s| \le \exp(-\sin^2(s)/2)$ and Lemma 14. The integral is at most $c/\sigma_J + (2/\ell)\exp(-cN)$, and $\exp(-cN) \le c/\sigma_J$. □

Lemma 16 (Window)**.** *For all $k \ge 1$ and S:* $\Pr[|S + R_{k-1}| \le k] \le c\,(k/\sigma_k)\exp(-S^2/(8\sigma_k^2))$.

Proof. Split $R_{k-1} = A' + B'$ by odd and even weights. $|\sigma_{A'}^2 - \sigma_{B'}^2| = k(k-1)/2 \le k^2$, so $\sigma_{A'}^2, \sigma_{B'}^2 \le \sigma_k^2/2$. If $|S| \ge 4k$, the event forces $|A'| \ge 3|S|/8$ or $|B'| \ge 3|S|/8$. Say the first: Hoeffding's inequality [35] gives $2\exp(-9S^2/(128\sigma_{A'}^2)) \le 2\exp(-S^2/(8\sigma_k^2))$, and Lemma 15 applied to the independent B' with $\ell = 2k+1$ gives the factor ck/σ_k. If $|S| < 4k$ the exponential is bounded below and Lemma 15 suffices. Tiny k are absorbed in c. □

Lemma 17 (Local limit)**.** *There are k_0 and c such that for $k \ge k_0$ and every x in the parity lattice of R_k with $|x| \le \sigma_k\sqrt{11\log k}$:* $(1/c)(1/\sigma_k)\phi(x/\sigma_k) \le \Pr[R_k = x] \le c(1/\sigma_k)\phi(x/\sigma_k)$.

Proof. Let η solve $\sum_{i\le k} i\tanh(\eta i) = x$. The left side is monotone, and at $\eta = 1/k$ it is of order k^2, beyond the range, so $|\eta|k \le 1$. Then $\tanh(s) \ge s\tanh(1)$ on $[0,1]$ gives $|\eta| \le c\sqrt{\log k}/\sigma_k$ and $|\eta|k \le c\sqrt{\log k/k}$. Under P_η (independent, $\Pr[\varepsilon_i = +1] = e^{\eta i}/(2\cosh\eta i)$), $\Pr[R_k = x] = \mathcal{Z}e^{-\eta x}P_\eta[R_k = x]$ with $\mathcal{Z} = \prod\cosh(\eta i)$. Expanding, $\log\mathcal{Z} - \eta x = -x^2/(2\sigma_k^2) + O(\eta^4 k^5)$ with $\eta^4k^5 = O(\log^2 k/k)$, and $\mathrm{Var}_\eta = \sigma_k^2(1+o(1))$. The lattice has span 2: $P_\eta[R_k = x] = (1/\pi)\int_{-\pi/2}^{\pi/2}\mathbb{E}_\eta[\exp(\mathrm{i}\theta(R_k - x))]\,d\theta$. Also $|\mathbb{E}_\eta\exp(\mathrm{i}\theta\varepsilon_i i)|^2 = 1 - \mathrm{sech}^2(\eta i)\sin^2(\theta i) \le \exp(-\sin^2(\theta i)/2)$ for k large. On $|\theta| \le \log k/k^{3/2}$, the centered expansion gives $(\sqrt{2\pi}/\sigma_k)(1 + o(1))$ with error $O(|\theta|^3k^4) = O(\log^3 k/\sqrt{k})$. For larger $|\theta|$ up to $\pi/2$, Lemma 14 with $d = 1$ bounds the integrand by $\exp(-c\log^2 k)$, negligible against $1/\sigma_k$. □

Lemma 18 (Hazard)**.** *For $k \ge k_0$ and $\zeta = |S|/\sigma_k \le \sqrt{10\log k}$:* $\Pr[|S+R_{k-1}| \le k] \le c(k/\sigma_k)\phi(\zeta)$ *and* $\Pr[R_k < -|S|] \ge c^{-1}\phi(\zeta)/(1+\zeta)$.

Proof. First: apply Lemma 17 to R_{k-1}. There are at most $k+1$ lattice points in the window. Changing σ_{k-1} to σ_k and moving the location by at most k changes the exponent by $O(\zeta^2/k + \zeta/\sqrt{k}) = o(1)$. Second: the upper tail contains the lattice points in $(|S|, |S| + \sigma_k/(1+\zeta)]$, in the range of Lemma 17, each of mass at least $(c/\sigma_k)\phi(\zeta + 1/(1+\zeta)) \ge (c'/\sigma_k)\phi(\zeta)$; there are at least $c\sigma_k/(1+\zeta)$ of them. □

Lemma 19 (Envelope)**.** *There is a universal c such that* (5) *holds for all $k \ge 1$, all S and all α.*

Proof. Let $\zeta = |S|/\sigma_k$. $\Delta_k(S) \le \Pr[|S+R_{k-1}| \le k]$, because $y_{k-1}(S+k) - y_{k-1}(S-k)$ is a mass of R_{k-1} in the closed window, with endpoints weighted by α or $1-\alpha$. And $m_k(S) \ge \Pr[R_k < -|S|]$ for every α, by symmetry. Let $k \ge k_0$. If $\zeta \le \sqrt{10\log k}$, Lemma 18 gives $g = \Delta(\Delta/m) \le c(k/\sigma_k)^2(1+\zeta)\phi(\zeta) \le (c/k)(1+\zeta)e^{-\zeta^2/16}$, with ϕ the standard normal density. If $\zeta > \sqrt{10\log k}$, then $g \le \Delta$ since $\Delta \le m$, and Lemma 16 gives $g \le ck^{-1/2}e^{-\zeta^2/8} \le ck^{-1/2}k^{-10/16}e^{-\zeta^2/16} \le (c/k)e^{-\zeta^2/16}$. For $k < k_0$: $g_k(S) \le 1$, and $g_k(S) = 0$ when $|S| > k(k+1)/2$, since then all of $S \pm k + R_{k-1}$ have the sign of S and $\Delta = 0$. So $|S|/\sigma_k$ is bounded on the support, and after enlarging c, (5) holds for all $k \ge 1$. □

Proof of Theorem 3. The second bound is Lemma 15 with $\ell = 1$ and $\sigma_T^2 \ge T^3/3$. For the first, fix $K = 2^i \le 2T$ and $k \in (\lfloor K/2\rfloor, K]$. Since $\sigma_k \le \sigma_K$, (5) gives $g_k(S) \le (2c/K)\tilde\psi(|S|/\sigma_K)$, with $\tilde\psi(w) = \sup_{\zeta\ge w}\psi(\zeta)$, and $\sum_j \tilde\psi(j)$ is finite. Let E_j be the event that some k in the block has $|S_k|/\sigma_K \in [j, j+1)$. Take the first such k in reveal order, that is, decreasing k. Given the history at that stopping time, the remaining increment down to $\lfloor K/2\rfloor$ is independent with variance at most σ_K^2, so by Chebyshev's inequality it has magnitude at most $2\sigma_K$ with probability at least $3/4$.

So $\Pr[E_j] \le (4/3)\Pr[|S_{\lfloor K/2 \rfloor}|/\sigma_K \in [j-2, j+3)] \le c\,\sigma_K/\sigma_T$, by Lemma 15 on $J = (\lfloor K/2 \rfloor, T]$ if $K \le T$ (a union of two intervals of length $5\sigma_K$, uniformly in t_0), and trivially if $K > T$. Thus $\mathbb{E}\sup_{\text{block}} g \le (c/K)(K/T)^{3/2} = cK^{1/2}T^{-3/2}$, and summing over dyadic K gives c/T. □

Appendix C.
Proof of Lemma 9

Passive protocol Π_0. Each party's input is (x_i, p_i, r_i): its input to f, an independent one-time pad and randomness. Let f' have only public output: w together with $e_i = w_i \oplus p_i$ for all i, where $(w, (w_i)_i) = f((x_i)_i; \bigoplus_i r_i)$. A party decodes $w_i = e_i \oplus p_i$. Π_0 is the passive protocol of Assumption 1 for f': multiparty garbling [6], [9], with the garbled tables computed by GMW [31] from the assumed OT, instances in parallel. A garbled entry is the XOR of pseudorandom values that the parties evaluate locally and of a key that is selected by one AND of two shared bits followed by one AND of a shared bit with a string; so the inner computation has two layers of AND gates, each of them one parallel round of pairwise OTs, whatever the circuit and the number of parties. Π_0 is thus secure against any passive $\mathcal{C}$, with a number q of rounds that is an absolute constant. Private channels are pairwise pads over broadcast, from OT of two random seeds followed by disclosure of the choice, with each seed stretched into a pad stream and disjoint positions per message. The selected seed stays hidden from an outsider: switch the choice while keeping the disclosed bit, which receiver privacy hides even given both seeds; that seed is then unselected, so sender privacy hides it from the receiver's larger view. All messages of Π_0 are broadcast, and each is a deterministic function of $(x_i, p_i, r_i, \vartheta_i)$ and the broadcast transcript of the preceding rounds, with ϑ_i the tape.

Slots. Every message below has a fixed slot and fixed length, and a missing or badly formatted message causes a common abort naming its sender at that slot. Every check is a deterministic function of the broadcast transcript, so all parties compute the same verdict, including parties already removed.

Commitments. Naor's scheme [43] with receiver string σ: hiding for every σ, binding except with probability $2^{-\lambda}$ for uniform σ. $\mathrm{COM}(v)$ is one commitment to v under each σ_i, $i \in Z$, with independent randomness.

Proofs. $\mathrm{PROVE}(i, j, X)$ is a Feige–Shamir argument [29], [30], with fresh s_0, s_1 in each invocation. The verifier j broadcasts $y_0 = \mathsf{g}(s_0)$, $y_1 = \mathsf{g}(s_1)$, with g a one-way function, and gives a witness-indistinguishable proof of knowledge (WI-PoK) of a preimage of y_0 or y_1. Then the prover i gives a WI-PoK of "X, or a preimage of y_0 or y_1". The WI-PoK is parallel Blum Hamiltonicity [13] with Naor commitments under the challenger's own receiver string, with λ parallel copies: constant rounds, the challenger's messages are a receiver string and a challenge, every string of the right length is valid, verification is public, and completeness is perfect for every valid challenge. So a rejected inner proof names j, a rejected final proof names i, a missing challenge names whoever owes it, and an announced "reject" has no effect. All proofs of a call run one at a time in a fixed order over ordered pairs, so no concurrent composition arises.

Steps.

A. Each i broadcasts a uniform σ_i.

B. In fixed order each i broadcasts $\mathrm{COM}(x_i, p_i, r_i, a_i, (c_{i,j})_{j \ne i})$, with a_i its tape seed and $c_{i,j}$ its mask for j's tape. Then $\mathrm{PROVE}(i, j,$ "all copies of i's commitment open to one value") for all ordered pairs.

C. In fixed order each i broadcasts $(c_{i,j})_{j \ne i}$ and runs $\mathrm{PROVE}(i, j,$ "these are the committed masks") for all j. Define $\vartheta_i = a_i \oplus \bigoplus_{j \ne i} c_{j,i}$.

D. For each round of Π_0: every i broadcasts its batch; then $\mathrm{PROVE}(i, j,$ "this batch is the message Π_0 prescribes for this round, given the committed (x_i, p_i, r_i, a_i), the disclosed masks and the transcript of Π_0's preceding rounds") for all ordered pairs. These proofs finish before anyone computes the next round.
E. If nothing aborted, the call commits and outputs are decoded from the transcript.

There are $q + 2$ batches of messages, each followed by $z(z - 1)$ arguments of a constant number of rounds, so the call has at most $C_L z^2$ rounds for a universal constant C_L.

No honest party is blamed. Its messages are well formed and its statements are true, since a batch is a function of its committed data and the public transcript. An accepted proof by a corrupted party rules out a malformed earlier batch, up to the soundness below, so an honest party can always compute its next batch.

Extraction. To extract from an accepted WI-PoK, Sim keeps the accepted transcript and tries at most $\Gamma = 2^\lambda$ further uniform challenges with the same first messages, stopping at a second distinct accepted challenge. Two of them yield a witness, except for binding. Suppose d' of the Γ challenges accept for a fixed prefix and tape. If $d' = 1$, the Γ tries occur with probability $1/\Gamma$, contributing one expected try. If $d' \geq 2$, the weighted expected number of tries is at most $(d'/\Gamma)(\Gamma/(d' - 1)) \leq 2$. Failure has probability at most $(d'/\Gamma)(1 - (d' - 1)/\Gamma)^\Gamma \leq 1/\Gamma$. So extraction takes expected polynomial time for every prefix and auxiliary input, and fails with negligible probability. These tries are computation of Sim, not protocol rounds.

Claim 2. *Let i be corrupted and j honest. In the real protocol and in every hybrid below, whenever the final proof of* $\mathrm{PROVE}(i, j, X)$ *accepts, extraction yields a witness for X, except with negligible probability.*

Proof. Extraction yields a witness for the disjunction. Suppose it is a preimage with non-negligible probability. Reduce to the one-wayness of g in the given hybrid: put the inversion challenge at one of y_0, y_1, at a random position, generate the other with a known preimage, and use that preimage in j's inner proof. Witness indistinguishability hides the position, so with about half that probability the extracted preimage inverts g. The reduction can run every hybrid: honest false statements use the trapdoors of their own PROVE invocations; the targeted invocation has no honest final prover; the passive view does not involve its unknown preimage; extraction rewinds only the final challenge, after the inner proof has finished. □

Simulator. Sim runs $\mathcal{A}$, rewinds only inside one PROVE sub-proof at a time and plays honest verifiers honestly.

1) Honest prover i, corrupted verifier j: if j's inner proof accepts, Sim extracts a preimage, restores the state, and uses it as the witness in the final proof. Honest prover, honest verifier: Sim uses the verifier's s.
2) Steps A, B: honest σ as in the protocol, honest commitments are $\mathrm{COM}(0)$. From each corrupted i's final proof to the first honest verifier, Sim extracts an opening $(x_i, p_i, r_i, a_i, c_{i,\cdot})$, by Claim 2. An abort at any slot naming i: Sim sends (abort, i) to TP.
3) After Step B: Sim sends $(x_i)_{i\in\mathcal{C}}$ to TP and receives $w, (w_i)_{i\in\mathcal{C}}$. It sets $e_i = w_i \oplus p_i$ for $i \in \mathcal{C}$ and samples e_i uniform for $i \in \mathcal{H}$. It runs the passive simulator of Π_0 on $\mathcal{C}$, $(x_i, p_i, r_i)_{i\in\mathcal{C}}$ and the public output $(w, (e_i)_i)$, which returns tapes $(\vartheta_i)_{i\in\mathcal{C}}$ and all honest batches. These results are cached and lie outside all later rewinding.
4) Step C: with h^* the first honest party, sample all other honest masks uniform, and for corrupted j set $c_{h^*,j} = \vartheta_j \oplus a_j \oplus \bigoplus_{l\neq j,h^*} c_{l,j}$, using the extracted corrupted masks. The

proofs are as in item 1. $\mathcal{A}$ cannot later disclose other values and continue, by binding and Claim 2.

5) Step D: Sim posts the honest batches. A corrupted batch is now determined by the extracted data and the transcript. If it differs and the proof to an honest verifier accepts, Sim halts with failure. An abort naming i: Sim sends (abort, i) to TP.
6) If everything passes, Sim lets TP deliver.

Hybrids. H_0 is the real protocol.

H_1: honest provers use the witnesses of item 1. Extract the corrupted verifier's witness, restore the state, then switch the honest final proof's witness by witness indistinguishability, one PROVE at a time. For an honest verifier the preimage is already available.

H_2: honest commitments are $\mathrm{COM}(0)$. The reduction to hiding, which holds for every σ including one chosen by $\mathcal{A}$, keeps its external challenge commitment fixed. It knows the genuine honest data used later, honest proofs use trapdoors, so it never opens the challenge, and extraction rewinds only subsequent corrupted proofs. So extraction lies inside the distinguisher, and the extracted corrupted values are indistinguishable between H_1 and H_2, not only the visible commitments. Sequential scheduling is what allows this: the corrupted values acquire no dependence on honest inputs through these messages, though they may depend on auxiliary information correlated with those inputs.

H_3: fix the adversary-visible prefix of Step B and the extracted corrupted data; after H_2 these are independent of fresh honest $p_i, r_i, a_i, c_i,..$ The effective tapes $\vartheta_{\mathcal{C}}$ are not yet fixed; honest masks determine them. In H_2, for corrupted j, the map from $c_{h^*,j}$ to ϑ_j is an invertible reparameterization: sampling ϑ_j uniform and setting $c_{h^*,j}$ by item 4 gives the same joint law as uniform $c_{h^*,j}$, with all other masks unchanged. For honest j', $\vartheta_{j'}$ is uniform and independent of everything disclosed, because $a_{j'}$ is never disclosed. So $\mathcal{A}$'s whole view and the honest outputs in H_2 are one efficient function of the prefix, independent uniform masks and the real passive view with its output. Replace the real passive view by the passive simulator's view on the same public output of f'; indistinguishability is passive security, averaged over uniform honest inputs p_i, r_i. Honest outputs stay decoded as $e_i \oplus p_i$.

H_4 is the ideal process: the public output of f' and the honest outputs come from TP as in item 3. The law is the same: $\bigoplus_i r_i$ is uniform given $r_{\mathcal{C}}$, as honest r_i are fresh and independent of the prefix, and for honest i, p_i occurs nowhere else after H_2, so e_i is uniform given $(w, (w_i)_i)$. The transcript and honest outputs are replaced jointly.

The failure in item 5 is negligible: statistical binding under an honest σ makes the statement false, so Claim 2 gives a contradiction. $\mathcal{A}$ may stop after seeing its tape. H_3 uses the uniformity of the tape before this selection, and the stop is an abort in the ideal process. Every count above is polynomial in z, and every loss is a sum over $O(qz^2)$ arguments. □

Appendix D.
Phases and the Terminal Rule

Phases. After a Defense call commits, the protocol is in the Coin phase of that round, so a stop before invoking Coin counts as an abort during that Coin: the child gets the new pending coin, whose defense bits already exist. After Coin commits, its coin joins the public path, and a later stop belongs to the next call. A stop before Init commits uses the Init rule and a stop before Release commits uses the Release rule, so that the public path is the same for all honest parties.

Terminal rule. Every party, including one marked as aborted, keeps reading the broadcast channel until the schedule ends, and outputs the following function of the broadcast transcript. The

transcript determines the current run, its table and every abort. If the last run has a constant table, output that constant. If the last run completes Release, output the bit decoded from the transcript. If the last run is on a single party, that party has one broadcast slot for its bit, and the output is that bit, or 0 if the slot is empty or malformed. If the set is empty, output 0. Any condition with no defined continuation, such as failed decoding, an invalid table, an unsupported path or undefined arithmetic, ends the execution with output 0. A party that cannot compute its next message stays silent. The schedule is bounded by the call count times a fixed call length, so all parties terminate on every transcript, and they agree literally because the rule reads only broadcast data. It changes nothing in Theorem 4. Outside a soundness failure no honest party is removed, so every removed party is corrupted, a single remaining honest party broadcasts its prescribed bit, and the set is never empty while an honest party is active. A soundness failure has negligible probability, so it changes the output probability by a negligible amount.

Appendix E.
Proof of Theorem 2

Fix a polynomial-time adversary that corrupts a set I with $|I| \leq t$, and let J be as in Lemma 12. We turn the adversary into an adversary $\mathcal{A}'$ against Π that corrupts the emulated parties in J and biases Π as much, up to a negligible term and the truncation at the end. Theorem 1 for Π then gives the bound.

Case 1: the group has a corrupted majority. Then $n' \in J$. $\mathcal{A}'$ runs the adversary and plays every honest member of G itself, with fresh randomness and the code of $\Pi^\star$. In round ℓ of Π it first receives the messages of the honest emulated parties, which are the announcements of the honest parties P_i, $i \leq k+1$; it runs epoch ℓ internally, shows these announcements to the adversary in step (b) and not earlier, decodes as in (c), and sends the decoded messages of the parties in J. The decoded transcript has the same law as in $\Pi^\star$, and both protocols output Out of it.

Case 2: the group has an honest majority. Then J consists of corrupted parties P_i with $i \leq k+1$. We change the execution in three steps.

Step 1. Every run of the protocol of Lemma 9 is replaced by its ideal process, call by call, by the composition of Section 7.3. The runs happen one after another and no honest party sends other messages meanwhile; each uses fresh internal randomness and a fresh channel setup, and the state that persists between them, namely the signed shares, the verification keys, the transcript and the active set, enters as input and auxiliary input. In the ideal process an abort names a corrupted member, so no honest member is ever removed, $|\mathsf{Act}| \geq \mu + 1$ throughout, and every call ends with delivery.

Step 2. Each f_ℓ is replaced by the function that outputs $\mathsf{NM}_{n'}(\omega, \mathrm{tr}_{<\ell})$ for the tape ω of the committed run of f_0 and ignores its inputs. The honest members always supply at least $\mu + 1$ genuine shares, so the two differ only if a corrupted member supplies a different share with a valid signature. This is a forgery under a one-time key of which it saw one signature; the reduction guesses the committed run of f_0, the member and the epoch.

Step 3. The corrupted members hold at most μ shares of the committed run, which are independent of ω, and outputs of aborted runs of f_0, which concern independent tapes. The adversary can sample all of these itself. Now the group announces in every epoch the message $\mathsf{NM}_{n'}(\omega, \mathrm{tr}_{<\ell})$ for a uniform ω that is independent of everything else, and since at least $\mu + 1$ honest members announce it, it is the decoded message. So $V_{n'}$ behaves as an honest party of Π, and an adversary $\mathcal{A}'$ that corrupts J, receives this message as a rushing adversary of Π and hands it to the adversary as the output of the ideal calls of epoch ℓ, produces the same execution.

Conclusion. The simulators of Step 1 run in expected polynomial time. As in Section 7.3, for every polynomial p we truncate them and obtain an adversary $\mathcal{A}'_p$ against Π with

$$\left|\Pr[\text{out} = 1] - \tfrac{1}{2}\right| \leq \frac{c\,2^{n'}}{T'} + \frac{1}{p} + \mathrm{negl}_p(\lambda),$$

where the left-hand side refers to $\Pi^\star$. Since it does not depend on p, the excess over $c\,2^{n'}/T'$ is negligible. If $\rho \geq s+1$, the number of rounds is $r \leq 3C_L\rho s^2 \leq 3C_LC_1n'^3T's^2$, which gives the last statement of the theorem. □

Appendix F.
Group Calls in a Constant Number of Rounds

Assumption 2 (Constant-round computation with an honest majority)**.** *There is an absolute constant q_G with the following property. For every polynomial $s = s(\lambda)$ and every polynomial-size randomized circuit f with one public output and one private output per party, there is an s-party protocol with q_G rounds and complexity $\mathrm{poly}(s, |f|, \lambda)$, over pairwise private channels and an authenticated broadcast channel, that securely computes f with guaranteed output delivery against static, rushing, malicious polynomial-time adversaries corrupting fewer than $s/2$ parties, in the stand-alone sense of [18] with auxiliary input: for every such adversary there is a (possibly expected) polynomial-time ideal adversary such that, for all inputs and auxiliary inputs, the pairs (adversary output, honest outputs) in the real execution and in the ideal process without abort are computationally indistinguishable.*

The security of the protocol of Beaver, Micali and Rogaway [6] is stated as privacy and correctness separately and without auxiliary input, and the composition of Section 7.3 needs the joint form above.

In $\Pi^\star_{\mathrm{X}}$ every private message of a group call is sent on the broadcast channel encrypted under a pad. Each execution sets up fresh pads for every ordered pair of members, from an OT of two random seeds followed by disclosure of the choice.

Lemma 20 (Private channels over broadcast)**.** *Assume Assumption 2 and an OT protocol secure against passive adversaries. A group call of $\Pi^\star_{\mathrm{X}}$ securely computes f, in the sense of Assumption 2, against static adversaries that corrupt fewer than $s/2$ members of G and read the whole broadcast channel.*

For the lemma, an adversary against the protocol with ideal private channels plays the honest side of the setup OT for every pair with a corrupted member, and so translates between ciphertexts and plaintexts; for a pair of honest members it shows a fresh setup and uniform strings, and a hybrid over these pairs replaces their pads by uniform strings using the privacy of the OT and the pseudorandom generator. The proof of Theorem 5 is that of Theorem 2, with Lemma 20 in place of Lemma 9 in Step 1 and no aborts inside the group. Both proofs are in Appendix J.

Appendix G.
Efficient Functionalities

We keep all tables as exact rationals, and y_k is computable by dynamic programming over integers of polynomial range. Conditional path sampling uses the transition probabilities $y_{k-1}(S \pm k)/(2y_k(S))$ for $b = 1$ and likewise for $b = 0$, not rejection until the terminal output matches. A

rational Bernoulli returns 0 deterministically at probability 0 and 1 deterministically at probability 1; otherwise it is sampled by rejection with acceptance at least $1/2$, capped after $\lambda + O(\log(T+1))$ attempts, with fallback 0. So every sampled transition stays supported, and the total statistical error is negligible in λ. The reference parameters are chosen deterministically from the support atom j with $\Pr[R_T > j] \le A \le \Pr[R_T \ge j]$: $t_0 = -j$ and $\alpha = (A - \Pr[R_T > j])/\Pr[R_T = j]$. Parallel Defense bits go in one aggregated call, as in BHLT. Coin and Release bits are public outputs of their calls.

Bit lengths for fixed n. With exact means, let b_d bound the bit lengths of the numerators and denominators of the odds of the values a_v in a run at depth d, that is, after d aborts. The mean $A = 2^{-p}\sum_v a_v$, which is also the mean of the table, and its odds have at most $2^p(2b_d + 1) + p$ bits. The parameter $\alpha = (A - \Pr[R_T > j])/\Pr[R_T = j]$ adds $O(T)$ bits, and so does each value $y_k(S) = \Pr[R_k > -S] + \alpha\Pr[R_k = -S]$ of the run. By Lemma 4, the odds of the child's values a_v are those of the parent multiplied by the odds of y and divided by the odds of A. So $b_{d+1} \le (1 + 2^{p+3})\, b_d + O(2^p + T)$ with $p \le d$, and therefore $b_{n-1} \le O(T)\prod_{d<n}(1 + 2^{d+3}) = T\, 2^{O(n^2)}$, which is polynomial in T for fixed n. The same bound holds for A, α, the values $y_k(S)$ and the table entries of every run. The constants C_z of (1) are integers, so the strengths $\kappa_k = C_z\Delta/m$ are rationals with $T\, 2^{O(n^2)}$ bits as well.

Appendix H.
Rule R and Bit Lengths

Rule R in full. Let $\hat{A}$ be the mean of the table of the run. Let j be the largest atom of R_T with $P_> := \Pr[R_T > j] \le \hat{A} \le \Pr[R_T \ge j]$, let $P_= = \Pr[R_T = j]$, $\alpha = (\hat{A} - P_>)/P_=$ and $t_0 = -j$. If j is not one of the extreme atoms $\pm T(T+1)/2$, let $e = T + \lambda$ and let $\tilde{\alpha}$ be the multiple of 2^{-e} nearest to α, with ties broken downward. If $j = T(T+1)/2$, let $e = \lambda + \lceil \log_2(1/\alpha) \rceil$ and $\tilde{\alpha} = \lfloor \alpha 2^e \rfloor 2^{-e}$; if $j = -T(T+1)/2$, do the same with $1 - \alpha$ in place of α. The run uses the reference parameters $(t_0, \tilde{\alpha})$, and $A := y_T(t_0)$ is computed with $\tilde{\alpha}$. In all cases $\tilde{\alpha}$ is a multiple of 2^{-e}; we call e the precision exponent of the run.

Proof of Lemma 10. Every atom of R_T has probability a positive multiple of 2^{-T}, and $A - \hat{A} = (\tilde{\alpha} - \alpha)P_=$. At a non-extreme atom $|\tilde{\alpha} - \alpha| \le 2^{-(T+\lambda+1)}$, while $\hat{A} \ge P_> \ge 2^{-T}$ and $1 - \hat{A} \ge \Pr[R_T < j] \ge 2^{-T}$, so both relative errors are at most $2^{-\lambda-1}$; also $A \ge P_> > 0$ and $1 - A \ge \Pr[R_T < j] > 0$. At the top atom $2^{-e} \le 2^{-\lambda}\alpha$, so $\alpha(1 - 2^{-\lambda}) < \tilde{\alpha} \le \alpha$, and $\tilde{\alpha} > 0$ because $\alpha 2^e \ge 2^\lambda$. Then $A = \tilde{\alpha}2^{-T}$ gives the claim for $A/\hat{A}$, and $(1 - A) - (1 - \hat{A}) < 2^{-\lambda-T}$ with $1 - \hat{A} \ge 1/2$ gives it for the complement; the bottom atom is symmetric. Lemmas 4 and 5 allow any $A \in (0, 1)$, so Lemma 6 is unaffected, and it bounds the four ratios with $\hat{A}$ in place of A by 2. In Lemmas 3, 7 and 8 the constant 2 becomes B', which multiplies the bound of Theorem 4 by $(B'/2)^3$. □

Claim 3 (Precision)**.** *Let e_d be the precision exponent of a run at depth d, with $e_0 = 1$ for the top run, and let $\lambda \ge 2$. Then $e_{d+1} \le \max(T + \lambda, e_d + \lambda + 4)$, and hence $e_d \le T + 5 + d(\lambda + 4)$.*

Proof. Every value $y_k(S) = \Pr[R_k > -S] + \tilde{\alpha}\Pr[R_k = -S]$ of a run with precision exponent e is a multiple of $2^{-(k+e)}$, so if $y \notin \{0, 1\}$, then $\min(y, 1-y) \ge 2^{-(T+e)}$. The values τ of the child satisfy $\operatorname{logit}\tau = \operatorname{logit} y + \xi$ with $|\xi| \le 2\ln B'$, by Lemmas 2 and 10, where $y \notin \{0, 1\}$ is a value of the walk of the parent. Hence $\tau \ge y/B'^2$ and $1 - \tau \ge (1 - y)/B'^2$, so $\min(\tau, 1 - \tau) \ge 2^{-(T+e)}/B'^2$. The exact mean $\hat{A}'$ of the child's table is the average of τ over uniform pending coins, so $\min(\hat{A}', 1 - \hat{A}') \ge 2^{-(T+e)}/B'^2$ as well. At a non-extreme atom the child has $e' = T + \lambda$. At the

top atom $\alpha' = \hat{A}'2^T \geq 2^{-e}/B'^2$, so $e' = \lambda + \lceil \log_2(1/\alpha') \rceil \leq \lambda + e + 2\log_2 B' + 1 \leq e + \lambda + 4$, because $B' \leq 2\sqrt{2}$ for $\lambda \geq 2$; the bottom atom is symmetric. The closed bound follows by induction, since $e_0 = 1 \leq T + 5$ and $\max(T + \lambda,\, T + 5 + d(\lambda + 4) + \lambda + 4) = T + 5 + (d + 1)(\lambda + 4)$. □

Proof of Lemma 11. Let $L = T + 5 + n(\lambda + 4)$, which bounds every precision exponent by Claim 3, since $d \leq n - 1$. In every run the following holds.

1) Every $y_k(S)$ and A is $M/2^{k+e}$ with an integer $0 \leq M \leq 2^{k+e}$ and $k + e \leq T + L$.
2) Iterating Lemma 4 along the $d \leq n - 1$ ancestors, the odds of a_v are the product over $j = 1, \ldots, d$ of the odds of $y^{(j)}$ divided by the odds of $A^{(j)}$. Here $A^{(j)}$ is the reference parameter of the run ended by the j-th abort, and $y^{(j)}$ is the walk value used at that abort: y_r for the coordinate r of v if that abort left a coin, and aborts at Init contribute 1. So the odds of a_v are P_v/Q_v with integers $1 \leq P_v, Q_v \leq 2^{2d(T+L)}$.
3) Δ, $\bar{y}$ and m are multiples of $2^{-(T+L+1)}$, so $u_{p+1} = 2C_z\Delta/m$ is a ratio of integers of $O(T + L + \log n)$ bits.
4) A table entry $F(x) = 2^{-p} \sum_v \prod_i (1 + x_i v_i/u_i)\, a_v$ and the exact mean $\hat{A} = 2^{-p} \sum_v a_v$ have a common denominator of $O(p(T + L + \log n) + 2^p n(T + L)) = O(2^n n(T + n\lambda))$ bits.
5) At a non-extreme atom, $\tilde{\alpha}$ is obtained by one rounding of the rational $\alpha 2^{T+\lambda}$; no logarithm of α is computed. At the top atom, $e = \lambda + \lceil \log_2(1/\alpha) \rceil$ is found by comparing α with powers of two, and $e \leq L$ by Claim 3; the bottom atom is the same with $1 - \alpha$.

The dynamic program for $y_k(S)$ has $O(T^3)$ entries of $O(T + L)$ bits. All of this is polynomial in λ when $T \leq \mathrm{poly}(\lambda)$ and $2^n \leq \mathrm{poly}(\lambda)$. The rational Bernoulli sampler of Appendix G is unchanged: each draw errs with probability at most $2^{-\lambda}$, and there are $\mathrm{poly}(T, 2^n)$ draws, so the total sampling error is negligible. □

Appendix I.
Counts for a Growing Number of Parties

The proof of Theorem 1 in Section 7.4 uses that Lemma 9, the signed state of Section 7.1 and the composition of Section 7.3 remain valid when the number of parties grows with λ. The quantities of one call are polynomial in the number $z \leq n$ of its parties, in λ and in the size of f, the others are polynomial in 2^n, T and λ, and none needs z to be constant.

One call. We go through Appendix C for a call among z parties.

- The passive protocol Π_0 has q rounds, an absolute constant by Assumption 1, and its passive simulator handles any corrupted subset of the z parties in time polynomial in z.
- The private channels use $z(z - 1)$ pairs of seeds, set up in parallel, and $\mathrm{COM}(v)$ consists of z commitments.
- Steps B and C run $2z(z - 1)$ invocations of PROVE and Step D runs $qz(z - 1)$, one at a time, so a call has $O(qz^2) = O(z^2)$ rounds.
- For every prefix and auxiliary input, the expected number of further tries of an extraction, over the verifier's first challenge and counting the case in which nothing is extracted, is at most two, and the probability that a proof is accepted but extraction fails is at most $2^{-\lambda}$. A call has $O(qz^2)$ extractions.
- Claim 2 is applied to one invocation at a time, losing about a factor two, and a union bound covers the $O(qz^2)$ invocations.
- The simulator runs in expected time polynomial in z, λ and $|f|$.

- H_1 has $O(qz^2)$ steps that use witness indistinguishability, H_2 has at most z honest parties, each with z commitments, H_3 is one step, and H_4 is perfect. The failure of item 5 of the simulator is a union over $O(qz^2)$ proofs.

So Lemma 9 holds with $C_L z^2$ rounds, and every negligible term is $\mathrm{poly}(z) \cdot \mathrm{negl}(\lambda)$.

Signed state. A party's state tuple consists of its shares of the bits that the subsets $Y \subseteq Z$ containing it hold about the pending coins and the current coin, which are $O(2^z n)$ bits, together with its shares of b and of the path coins $r_T, \ldots, r_1$; it has $O(2^z n + T)$ bits, which is polynomial when 2^n is. Every signed tuple has its own one-time key, a call signs z tuples, and the probability of a forgery in Claim 1 is at most the number of tuples times a negligible function.

Composition and rounds. There are at most $n(2T + 2)$ calls, the expected simulation time of one call is $P = \mathrm{poly}(2^n, T, \lambda)$, the truncation is at $2n(2T + 2)Pp$ steps, and the total error is at most the sum of the at most $n(2T + 2)$ per-call errors. The number of rounds is $\sum_{z=2}^{n} (2T + 2)\, O(z^2) = O(n^3 T)$, and the schedule of the terminal rule (Appendix D) consists of $n(2T + 2)$ calls of $O(n^2)$ rounds each.

Simulation-based fairness. The proof of Lemma 2.18 of HT [32], which is Lemma 2.15 of BHLT [16], builds an ideal adversary that samples whole executions until the honest output equals the bit of the trusted party, and bounds the statistical distance by induction over the number of aborting rounds. It uses only that the protocol is correct, that the protocol and the adversary run in polynomial time and that the bias is at most $1/2 - 1/p$ for a polynomial p, and BHLT state the lemma with no restriction on the number of parties. With $T \geq 4K_n$, Lemma 10 bounds the bias in the hybrid model by $(1 - 2^{-\lambda})^{-3}/4$ plus a negligible sampling error, which is below $2/5$ for $\lambda \geq 3$, so the lemma applies with $p = 10$ for $n = n(\lambda)$, and its extension to the hybrid model needs only that the functionalities run in polynomial time, which is Lemma 11.

Appendix J.
Proofs of Lemma 20 and Theorem 5

Convention. If no member of G is corrupted, we regard one member as corrupted but following the protocol. This is possible because $\mu \geq 1$ when $s \geq 3$, and it only enlarges the view of the adversary. The broadcast transcript of a group call is then always part of the view of a corrupted member.

Proof of Lemma 20. Let $\mathcal{A}$ be an adversary against a group call of $\Pi^{\star}_{\mathrm{X}}$ for f. We define an adversary $\mathcal{A}''$ against the protocol of Assumption 2, which has ideal private channels. $\mathcal{A}''$ runs $\mathcal{A}$ and does the following.

1) For every pair of an honest member h and a corrupted member c, it plays the side of h in the setup OT with $\mathcal{A}$, so it knows the pad that h uses with c.
2) A private message from h to c arrives at $\mathcal{A}''$ in the clear, because c is corrupted; $\mathcal{A}''$ shows it to $\mathcal{A}$ encrypted under the pad of h.
3) A ciphertext that $\mathcal{A}$ sends from c to h is decrypted under the pad of h and sent to h on the private channel.
4) For every pair of honest members, $\mathcal{A}''$ runs the setup OT itself with fresh inputs and shows $\mathcal{A}$ that transcript; in place of their ciphertexts it shows uniform strings of the right length.

The view that $\mathcal{A}''$ builds for $\mathcal{A}$, together with the honest outputs, differs from a real group call on the same inputs and auxiliary input only in the pairs of honest members, where ciphertexts under the pad of the pair are replaced by uniform strings. The honest outputs are computed by the same

code in both, since honest receivers get the real plaintexts. A hybrid over the at most s^2 such pairs shows that the two joint laws are computationally indistinguishable. Each step is the argument of Appendix C: switch the choice of the receiver while keeping the disclosed bit, by receiver privacy, which holds even given both seeds; the used seed is then the unselected one, which sender privacy hides; so it is a uniform seed independent of the transcript, and the pseudorandom generator replaces the pad by a uniform string. The distinguisher knows all inputs, since the definition quantifies over all inputs, and it runs everything else itself. Finally, Assumption 2 applied to $\mathcal{A}''$ gives an ideal adversary for the protocol with private channels, and composing it with the translation of $\mathcal{A}''$ gives one for the group call. □

Proof of Theorem 5. A group call of $\Pi^{\star}_{\mathrm{X}}$ has $q_G + C_0$ rounds, and there are $\rho + 1$ group calls and ρ announcement rounds, which gives $r \le (q_G + C_0)(\rho + 1) + \rho$, and $r \le (q_G + C_0 + 2)\rho$ once $\rho \ge q_G + C_0$. A member that gets no well-formed output from a call announces $\perp$. Agreement and correctness are as for $\Pi^{\star}$. For the bias we follow Appendix E. Case 1 is unchanged, with $\mathcal{A}'$ playing the honest side of every group call, including its channel setup. In Case 2, Step 1 replaces each group call by an ideal call with guaranteed output delivery: the corrupted members choose substitute inputs, the trusted party computes the function and gives every member its output, and there is no abort. This step is justified by Lemma 20 and the composition of Section 7.3: the calls run one after another, each with a fresh channel setup, and no honest party sends other messages meanwhile. Steps 2 and 3 are as before, with a single run of f_0, and the conclusion is the same; if the ideal adversary of Assumption 2 runs in strict polynomial time, no truncation is needed. With $\rho \le C_1 n'^3 T'$ and $r \le (q_G + C_0 + 2)\rho$ the bound in rounds follows. □

Appendix K.
Comparisons in Terms of Rounds

All parties but one corrupted. By Theorem 1, the protocol with $r = O(n^3 T)$ rounds has bias $O(n^3 2^n / r) + \mathrm{negl}(\lambda)$, and the majority protocol has bias $O(n/\sqrt{r})$. The ratio of the two bounds is $O(n^2 2^n / \sqrt{r})$, which tends to 0 when $n^2 2^n = o(\sqrt{r})$, that is, when $n + 2\log_2 n \le \frac{1}{2}\log_2 r - \omega(1)$. This holds whenever $n \le \frac{1}{2}\log_2 r - 2\log_2\log_2 r - \omega(1)$, since then $\log_2 n < \log_2\log_2 r$. In this range $n^3 2^n = o(r)$, so $T \ge 2^n$, as Theorem 1 requires.

Any number of parties. Let $\rho \ge s + 1$, so that by Theorem 2 the bias is at most $c'(k+2)^3 2^{k+2} s^2 / r + \mathrm{negl}(\lambda)$. For $N/2 \le t < 2N/3$, Beimel et al. [8] obtain bias $O(2^{2^{k+1}}/r')$ with $r' = r - O(k+1)$. Since $r' \le r$, our bound is the smaller one when $2^{2^{k+1}}$ exceeds a constant times $(k+2)^3 2^k s^2$. Taking logarithms, this asks that 2^{k+1} exceed $2\log_2 s + O(k+1)$, which holds for $k \ge \log_2\log_2 s + O(1)$ when $s \ge 3$. Majority-type protocols have bias $O(t/\sqrt{r})$. Since $t \ge N/2 \ge s/2$, the ratio of our bound to theirs is $O((k+2)^3 2^k s/\sqrt{r})$, which tends to 0 when $(k+2)^3 2^k s = o(\sqrt{r})$.

Appendix L.
The Prototype

The implementation is written in Python and consists of n party processes and a broadcast board on one 16-core machine, with synchronous slots over local TCP. Pairwise keys come from Diffie–Hellman, and Beaver triples from IKNP OT extension [50] over Chou–Orlandi base OTs [49]. A coin is drawn by two 48-bit comparisons of a shared random number with public thresholds and one AND gate, and a defense bit is the coin XOR one more comparison. Aborts are implemented

in full, including child runs with pending coins and the rule for a single remaining party. We verified that the circuits match plaintext evaluation, that the public tables match exact arithmetic for $T \leq 25$, and that at $n = 3$ honest runs and one abort strategy follow the law of the hybrid model within sampling error. Table 4 lists measured runs.

TABLE 4. ONE COIN FLIP, MEASURED. d IS THE DELAY ADDED PER SLOT; SLOTS AND TIMES ARE FOR THE ONLINE PHASE, EXCEPT THE OFFLINE COLUMN. THE LAST TWO ROWS ARE MAJORITY WITH m COINS ON THE SAME BOARD. THE BIAS IS THE HYBRID-MODEL VALUE OF THE BEST FAIL-STOP ATTACK (COMPUTED).

n	T	d	slots	online	offline	bias
3	1025	0	8.1–8.9K	4.2–5.0 s	8.3–8.9 s	1.34%
3	4097	0	33.5K	29.7 s	50.4 s	0.369%
4	4097	0	35.0K	39.3 s	154.6 s	1.29%
3	4097	50 ms	34.2K	30.4 min	67.4 s	0.369%
4	1025	50 ms	8.0K	7.3 min	83.9 s	3.55%
3	m=1025	0	1025	0.49 s	none	2.49%
3	m=1025	50 ms	1025	67.1 s	none	2.49%

Appendix M.
Claims for the Prototype

Claim 4 (Per-round sampling)**.** *Let Init draw only $b \sim \mathrm{Bern}(F(h))$ and keep it shared in Z. In round k, with public $S = S_k$, let the call that needs r_k draw it with*

$$\Pr[r_k = +1 \mid b = 1, S] = \frac{y_{k-1}(S+k)}{y_{k-1}(S+k) + y_{k-1}(S-k)},$$

and with f in place of y when $b = 0$: inside Defense in a non-direct round, where r_k stays shared until Coin, and inside Coin in a direct round. Then h, b, the coins, all defense bits and all call outputs have the joint law of Section 4.2, and Theorem 4 applies unchanged.

Proof. Under Ref_A, $\Pr[\text{prefix to round } k,\ \mathrm{out} = 1] = 2^{-(T-k)} y_k(S_k)$ and $y_k(S) = (y_{k-1}(S+k) + y_{k-1}(S-k))/2$, so given $\mathrm{out} = 1$ the coins form a Markov chain in S with the stated transition, whose denominator is positive, and likewise with f given $\mathrm{out} = 0$. By the chain rule the sequential draws have the law of the path given $\mathrm{out} = b$, and the path depends on h only through b. Defense bits are drawn given r_k with fresh randomness, and the abort rules read only public data, pending coins and defense bits. After an abort the run's remaining coins are never drawn; drawing them afterwards would restore the full path. □

Claim 5 (Fail-stop adversaries)**.** *Assume OT secure against passive adversaries and semantically secure pairwise encryption. Realize each call of Claim 4 by GMW [31] over XOR shares with Beaver triples [5] made by OT between pairs, with all pairwise traffic, every OT message included, encrypted on the broadcast channel, and let every party name whoever leaves a slot empty. Then against every static polynomial-time fail-stop coalition of at most $n - 1$ parties, the bias is that of the hybrid model plus a negligible term. A sampling precision of L bits adds at most 2^{-L} per biased draw.*

Argument. All parties read the same board, so they agree on who missed which slot. Before the output step of a call the coalition sees its own inputs and openings masked by fresh triples, which

are uniform given its view, so a stop there is an abort before any output. At the output step the honest shares of the output wires and of the carried state are a uniform split given the coalition's view: trivially with one honest party, and by the honest–honest OT masks inside each triple with two or more; the top-level b is made from fresh local bits. So the shares reveal only the outputs, which is the hybrid rule that the adversary gets outputs first and may then abort. Private outputs are opened to their recipients only, and order within a slot gives the coalition nothing beyond the hybrid model. A stop while the top run's triples are made is a stop before Init and follows the Init rule; a stop while triples for a later call or a child run are made is an abort of that call before any output. Calls run one after another and every abort rule depends only on the call and the aborting set, so a real fail-stop adversary maps to a hybrid one. The coin r_k stays shared until Coin, so it cannot leak before Defense commits. □