Threshold Post-Quantum E-Voting from Lattice Commitments

Jalal Alizadeh Yadigar Imamverdiyev Ali Huseynli

Department of Cybersecurity, Azerbaijan Technical University, Baku, Azerbaijan

{jalal.alizadeh, yadigar.imamverdiyev}@aztu.edu.az, ali.huseynli@student.aztu.edu.az

Corresponding author: Jalal Alizadeh

November 2026

Abstract

Lattice-based homomorphic e-voting protocols built on commitments, such as EVOLVE, keep ballots private against all but one authority and the tally verifiable even if all authorities collude, but need every authority to tally. We present T-EVOLVE, in which votes are Shamir-shared over the commitment message space: any t of n authorities that took part in the checking phase complete the tally, privacy holds against $t-1$ colluding authorities and verifiability against all. The voter proves that her committed shares form a Reed–Solomon codeword whose value at zero is a valid vote; the parity checks cost no communication, and tallying interpolates public column sums without norm growth. Complaints of authorities, including about ciphertexts that do not decrypt, are publicly verifiable given encryption with publicly verifiable opening, which we define and instantiate under LWE. We prove privacy, verifiability, individual verifiability and robustness in the classical random oracle model; the verifiability reduction is not tight and the parameters rest on classical hardness estimates. In our C implementation, a yes/no ballot for four authorities takes 109 KB without signatures, or 147 KB with the computed seed ciphertexts, and an authority aggregates 10^4 ballots in about three minutes per successful tally round on one core.

Keywords: electronic voting, post-quantum cryptography, lattice-based commitments, threshold secret sharing, zero-knowledge proofs, verifiability.

1 Introduction

Homomorphic e-voting protocols built on lattice-based commitments—EVOLVE (del Pino et al., 2017), its extension EVOLVED (Black et al., 2022), Epoque (Boyen et al., 2021) and PQKryvos (Huber et al., 2026)—share an attractive trust model: ballot privacy holds as long as one authority is honest, while the tally remains verifiable even if all authorities collude. They also share a fragility: every vote is split *additively* among all n authorities, so the tally can only be completed if every authority takes part. A single crashed, unreachable or uncooperative authority blocks the election.

This family is explicitly based on the classical protocol of Cramer, Franklin, Schoenmakers and Yung (Cramer et al., 1996), which does not have this limitation: votes are shared with a t-out-of-n verifiable secret sharing, and any t correct authorities complete the tally. This work transfers that property to the lattice setting. We give a protocol, T-EVOLVE, in which

- any t of the n authorities suffice to compute and prove the tally, and up to $n-t$ authorities may be absent or malicious without affecting the result; an authority that is offline during the checking phase, in which authorities examine their shares and may complain, is counted among these $n-t$ (Section 4);
- ballot privacy holds against coalitions of up to $t-1$ authorities;

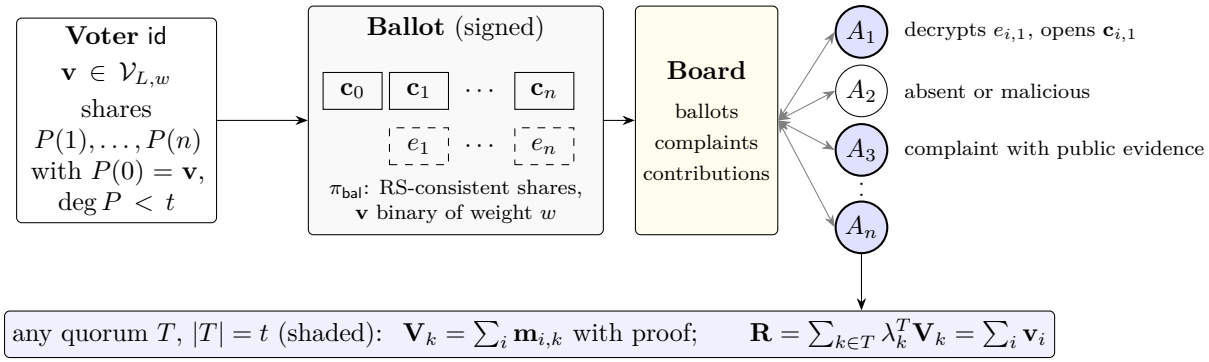


Figure 1: Overview of T-EVOLVE. The voter commits to her vote and to its Shamir shares, encrypts the opening of share k to authority k, and proves ballot validity. Any t authorities publish the sums of their columns of shares, and the result is obtained by Lagrange interpolation at 0. Privacy holds against $t-1$ authorities, and verifiability holds even if all authorities collude.

- the tally remains verifiable when *all* authorities collude (as in EVOLVE), and a voter can check that her ballot is on the board and counted, since, given an encryption scheme with publicly verifiable opening that satisfies (E3), no authority can have an honest ballot removed;
- single-choice and w-out-of-L elections are supported, with ballot validity proven by the voter.

Technical overview. Figure 1 summarizes the protocol. Each voter shares her vote with a Shamir polynomial of degree $t-1$ over R_q and commits to the vote and to each share with BDLOP commitments. She encrypts the opening of the k-th share to authority k and proves in zero knowledge that (i) the committed shares and the committed vote form a Reed–Solomon codeword and (ii) the vote is a binary vector of the prescribed weight. The consistency condition (i) is a set of $n+1-t$ linear equations over R_q with public coefficients. In a proof of knowledge of the openings with a common challenge, such equations cost no additional communication, whatever the size of the coefficients: they are verified by recomputing public values from the responses. For tallying, each authority in a quorum T opens the sum of its column of shares in the clear, using amortized proofs of knowledge and the tree of partial sums of EVOLVE. The tally is obtained by Lagrange interpolation of the t public column sums over $\mathbb{Z}_q$. Because interpolation is applied to public values and never to commitment randomness, no norm growth occurs and no special evaluation sets are needed: the points $1, \ldots, n$ suffice.

Contributions.

- A t-out-of-n variant of the EVOLVE family over BDLOP commitments, whose ballot proof ties $n+1$ commitments to a Reed–Solomon codeword with a valid vote at zero. We give definitions and proofs, in the classical random oracle model, of privacy against $t-1$ authorities, verifiability against all of them, individual verifiability, and robustness against $n-t$ absent or malicious authorities (Sections 4–6).
- Publicly verifiable complaints: an interface for encrypting the share seeds with publicly verifiable opening that also covers ciphertexts that do not decrypt, and an analysis of two instantiations used in prior voting protocols. Individual verifiability and robustness are proved relative to this interface. For release of identity-based keys, we identify a worst-case condition on decryption correctness that suffices for the soundness of complaints, and show that it yields an instantiation under LWE in the random oracle model with a

polynomially large modulus; parameter sets designed for average-case correctness do not meet the condition, and we fix parameters that do for its NTRU instantiation, at 9.6 KB per ciphertext, computed and not measured (Sections 3 and 7, Appendix D).

- Tools for EVOLVE-style proofs that apply to EVOLVE itself: one joint rejection step over all responses of a proof, a signed second challenge in the amortized proof of Baum & Lyubashevsky (2017) with extraction at slack 2 (Lemma 11), tail bounds for the witnesses (Lemma 8), including witnesses influenced by dishonest voters (Lemma 13), and the nested extraction with a labeled forking lemma (Theorem 14). Along the way, we observe that the constraints stated for EVOLVE force an opening bound above the modulus (Appendix A).

- Parameters derived from these bounds with the lattice estimator, and a C implementation with exact rejection steps and a bounded statistical error of its samplers (Section 8).

Scope and limitations. Corruption of authorities is static, the board is authenticated and seen identically by all parties, and a public-key infrastructure identifies eligible voters. The threshold protects the tally against authorities that fail after the checking phase; an authority that misses the checking phase can be excluded by a single dishonest voter, because an accepted ballot whose share it cannot open prevents it from proving its contribution (Section 4). Adaptive corruption, in which the adversary chooses whom to corrupt after seeing the board, is not covered: the reductions to hiding and to (E4) in Theorem 3 replace the honest commitments and ciphertexts by ones that cannot later be opened to the true shares, so they would need equivocable commitments and non-committing encryption, whereas the information-theoretic part of the argument (Lemma 2) does not depend on when the coalition is fixed. The protocol does not address receipt-freeness, coercion resistance or cast-as-intended verifiability. Privacy is proved in the vote-swapping sense of EVOLVE, not in the sense of BPRIV, and it is computational, not everlasting (Section 4). Public verifiability of complaints, and with it individual verifiability and robustness, is proved for any encryption scheme with publicly verifiable opening satisfying (E1)–(E3); privacy (Theorem 3) uses (E1), (E2) and (E4) of the same scheme. Release of identity-based keys satisfies the interface under LWE in the random oracle model (Appendix D), and we fix concrete parameters with worst-case decryption correctness for its NTRU instantiation, which we have not implemented; for encryption with proofs of correct decryption we know no efficient proof that a ciphertext does not decrypt. All proofs are in the classical random oracle model. The reduction for the efficient instantiation is not tight and gives no concrete security level at the proposed parameters, which, as for EVOLVE, are chosen from hardness estimates for M-LWE and M-SIS. The implementation replaces the encryption of the seeds by a placeholder, does not implement the signatures of the voters and is not constant-time.

Table 1 lists what is proved, under which assumptions, and what is instantiated and measured; the later sections do not repeat these caveats.

Relation to prior work. The architecture follows (Cramer et al., 1996; Schoenmakers, 1999), and the novelty lies in its lattice instantiation. Closest in goal are threshold approaches based on lattice MPC (Rivinius et al., 2022) or threshold FHE (Bonte et al., 2025); the latter achieves t-out-of-n tallying without public verifiability of the tally. For linear tallies, T-EVOLVE achieves threshold availability and verifiability under collusion of all authorities with commitments and an encryption scheme with publicly verifiable opening, without threshold key generation or general-purpose MPC. Section 2 compares these approaches.

2 Related Work

Homomorphic voting from lattice commitments. EVOLVE (del Pino et al., 2017) instantiates the protocol of Cramer et al. (1996) with BDLOP commitments: each vote is

Property	Statement	Assumptions	Concrete parameters	Measured
Correctness	Theorem 1	(E1); completeness of the proofs; tally rounds repeated until t succeed	Table 3	attempts, rounds (Section 8)
Privacy against $t-1$ authorities (vote-swapping, not BPRIV; computational)	Theorem 3	ROM; M-LWE (hiding); (E1), (E2), (E4); EUF-CMA; zero knowledge of the proofs; static corruption; honest board	Table 3, hardness estimates for M-LWE; EO: Appendix D	—
Verifiability, all authorities corrupted	Theorem 5 (online-extractable proof); Theorem 14 (EVOLVE-style)	ROM; M-SIS (binding); EUF-CMA; invertibility of challenge differences; Theorem 14 is not tight	Table 3, hardness estimates for M-SIS; no concrete bound on the protocol	—
Individual verifiability (honest ballot cannot be removed)	Theorem 6	EUF-CMA; completeness of Π_{bal}; (E3)	EO: Appendix D (not implemented)	—
Robustness: any t honest authorities that took part in the checking phase complete the tally despite $n-t$ absent or malicious ones	Theorem 7	EUF-CMA; completeness of the proofs; share-soundness (Section 7.4, after Theorem 14); (E1)–(E3)	Table 3	rounds (Section 8)
Sizes and times	—	encoding of Section 8; seed ciphertexts as 32-byte payloads	Table 3	Tables 4 and 5
Security in the QROM; everlasting privacy; BPRIV	—	open (Sections 4 and 9)	—	—

Table 1: What is proved, assumed, instantiated and measured. "Concrete parameters" says whether the row is covered by the parameter sets of Table 3; the hardness estimates there concern the primitives, not the protocol.

shared additively among N_A authorities, the voter proves with an OR-proof that the shares sum to a bit, and each authority proves with amortized proofs that it holds openings of its shares. EVOLVED (Black et al., 2022) extends EVOLVE to multi-candidate formats. Chillotti et al. (2016) gave an earlier homomorphic voting scheme based on LWE encryption. Epoque (Boyen et al., 2021) gives an end-to-end verifiable post-quantum protocol and points out that in EVOLVE a malicious authority can have ballots removed. PQKryvos (Huber et al., 2026) combines BDLOP commitments with a general-purpose proof system (Ligero) to support flexible ballot formats and public tally-hiding. In PQKryvos, votes are shared among the talliers with a full-threshold sharing. In all of these protocols, completing the tally requires every authority. T-EVOLVE keeps the commitment-based design and replaces the additive sharing by a t-out-of-n sharing.

Mix-nets, threshold decryption and MPC. Lattice-based mix-nets are the main alternative to homomorphic tallying: Boyen et al. (2020) give a verifiable decryption mix-net, Aranha et al. (2021) a lattice-based proof of shuffle, Aranha et al. (2023) a verifiable mix-net and distributed decryption, and Hough et al. (2025) reduce the cost of such a protocol with NTRU encryption and verifiable distributed decryption, all applied to voting. The distributed decryption of Aranha et al. (2023) involves all decryption servers and, as they note, has no adjustable threshold (Aranha

et al., 2023, Sec. 8). Rivinius et al. (2022) give a publicly accountable MPC protocol based on BGV and lattice commitments that is robust up to a threshold. Its authors expect it to be post-quantum secure but leave the analysis to future work. Bonte et al. (2025) tally complex elections with threshold (fully) homomorphic encryption, without public verifiability of the tally. Kaim et al. (2021) follow the blind-signature paradigm, in which the voter obtains a blind signature on an encrypted ballot, with a lattice-based blind signature and a ring variant of dual-Regev encryption, both with keys shared among the authorities with a threshold t; after t authorities reveal their key shares, every ballot is decrypted individually, so the tally is neither homomorphic nor based on a mix-net, and its security is argued informally. Tally-hiding voting, where only the result of a function of the tally is revealed, was introduced by Ordinos (Küsters et al., 2020). T-EVOLVE does not hide the tally; it targets the classical homomorphic setting, but with a threshold of authorities and without threshold key generation.

Publicly verifiable secret sharing on lattices. The protocol of Cramer et al. (1996) is built on publicly verifiable secret sharing (PVSS) (Schoenmakers, 1999). Lattice-based non-interactive PVSS schemes have been given by Gentry et al. (2022), Abdolmaleki et al. (2026) and Kate et al. (2026). Kate et al. (2026) resolve complaints by releasing identity-based decryption keys. For aggregation and applications such as voting, they give a variant in which Pedersen commitments replace their hash-based commitments, so that integrity is no longer post-quantum. T-EVOLVE uses lattice commitments throughout; key release is one of the instantiations of its complaints (Section 7). Griffy et al. (2026) let the holders of Shamir shares of a witness commit to their shares with BDLOP commitments and prove statements jointly over the sum of the commitments, with the randomness handled under threshold homomorphic encryption; their proofs are made rejection-free by widening the masks under the Hint-MLWE assumption. There the share holders are the provers and there is no dealer, but their construction relies on the same compatibility of BDLOP commitments with Shamir sharing as T-EVOLVE. A PVSS scheme proves that encrypted shares are consistent; a voting protocol additionally needs homomorphic tallying of committed shares and a proof that the shared value is a valid vote. The ballot proof of T-EVOLVE can be seen as a PVSS for the vote with a validity statement, built directly on BDLOP commitments.

Shamir sharing over rings. Threshold cryptosystems from lattices were studied early with lattice commitments and zero-knowledge proofs (Baum et al., 2016). In threshold decryption from lattices, Lagrange coefficients multiply noise. Boudgoust & Scholl (2023) obtain threshold decryption with a polynomial modulus, and Threshold Raccoon (del Pino et al., 2024) shares short secrets of a signature scheme with Shamir sharing and masks the partial signatures. Boneh et al. (2018) clear denominators by a factor $(N!)^2$ or use $\{0,1\}$-linear secret sharing, and Cho et al. (2024) show that partial decryptions of Shamir-shared keys can reveal the key. When short secrets such as commitment randomness or secret keys must be reconstructed, Shamir sharing over cyclotomic rings requires evaluation points from a subtractive set, and reconstruction factors grow with the number of parties (Albrecht & Lai, 2021; Kim et al., 2026). T-EVOLVE avoids this: shares are messages of commitments, the tally interpolates public column sums over $\mathbb{Z}_q$, and commitment randomness is never combined across authorities, so the points $1, \ldots, n$ suffice.

3 Preliminaries

Notation. Table 2 collects the main symbols. λ is the security parameter. $R = \mathbb{Z}[X]/(X^N + 1)$ with N a power of two, q a prime with $q > n$, and $R_q = R/qR$. The integers $1, \ldots, n$ are embedded as constants in R_q; since $q > n$ is prime, all their pairwise differences are units. Vectors are bold. For $a \in R_q$, $\mathrm{Rot}(a) \in \mathbb{Z}_q^{N \times N}$ is the matrix of multiplication by a in the coefficient basis, applied entrywise to matrices over R_q. For a set $T \subseteq [n]$ of size t and $k \in T$, the Lagrange

Symbol	Meaning
λ; λ_k^T	security parameter; Lagrange coefficient of authority k for the quorum T
N, q, R_q, d, L, μ	ring degree, modulus, $\mathbb{Z}_q[X]/(X^N+1)$, module rank, number of candidates, $\mu = 2d + L$
n, t, T, k	authorities, threshold, quorum ($\lvert T\rvert = t$), index of an authority
$\mathcal{C}$, $\mathcal{H}$	corrupted and honest authorities
$\mathbf{C} = \binom{\mathbf{A}}{\mathbf{B}}$, $\mathbf{c}_j$, $\mathbf{m}_j$, $\mathbf{r}_j$, β	commitment key (bold), commitments, messages, randomness, bound on honest randomness
$\mathbf{H}$; H_{bal}, H_{agg}, H, G, X	parity-check matrix (bold); random oracles
Acc, Hon, Dis	accepted ballots; accepted honest and dishonest ballots
$\mathbf{V}_k$, $\mathbf{R}$, R_a	column sum of authority k, result, its entry for candidate a
EO, lb, $\perp$	encryption with verifiable opening, label, decryption failure
w, $w = *$	prescribed weight of a vote; no weight constraint
σ_J, B_J, $\bar{B}_V$, $\bar{B}_{\mathrm{OR}}$	mask width, response bound and extraction bounds of the ballot proof
Θ; Θ_1, Θ_2; τ	tail bounds on the shifts of the ballot proof and of the aggregation; failure exponent
θ, θ'	challenge bit and guess in the privacy experiment
r, α, α', M, M'	rejection parameters ($r = 309/20$, italic; not the randomness $\mathbf{r}$)
l; ψ, ℓ; E, κ, ι	fan-in of the tree; columns of a block and of C_2 in the amortized proofs; blocks, attempts per round, attempt index
$\mathbf{U}$, N_{col}	statement matrix of an amortized proof; witness columns of one authority's tree
B_A, β_{SIS}, $\eta_{\max}$	extraction bound of the aggregation, M-SIS bound, bound on slack operators
Λ, Q_a, Q_b, q_H	labels of the inner fork; query counts
N_{rd}, p_{rd}, ν	tally rounds, lower bound on the success probability of a round, round index

Table 2: Main notation. Bold letters denote vectors and matrices. Except for R_a, the entry of $\mathbf{R}$ for candidate a, the same letter in italic denotes an unrelated scalar or set.

coefficient at 0 is

$$\lambda_k^T = \prod_{j \in T \setminus \{k\}} \frac{j}{j-k} \in \mathbb{Z}_q.$$

Commitments. We use the BDLOP commitment scheme (Baum et al., 2018) in the multi-message form of EVOLVE (del Pino et al., 2017, App. B). The public key is a uniformly random $\mathbf{C} = \binom{\mathbf{A}}{\mathbf{B}} \in R_q^{(d+L)\times(2d+L)}$ with $\mathbf{A} \in R_q^{d\times(2d+L)}$, and a message $\mathbf{m} \in R_q^L$ is committed as

$$\mathsf{Com}(\mathbf{m};\mathbf{r}) = \mathbf{C}\mathbf{r} + \begin{pmatrix}\mathbf{0}^d \\ \mathbf{m}\end{pmatrix}, \qquad \mathbf{r} \leftarrow D_\sigma^{N(2d+L)}.$$

A pair $(\mathbf{m}, \mathbf{r})$ is a valid opening of $\mathbf{c}$ under bound B if $\mathsf{Com}(\mathbf{m};\mathbf{r}) = \mathbf{c}$ and $\|\mathbf{r}\|_2 \le B$. The scheme is additively homomorphic, computationally hiding under M-LWE and computationally binding under M-SIS; the binding bound grows with B. Note that $\mathbf{r}$ determines $\mathbf{m} = \mathbf{c}_2 - \mathbf{B}\mathbf{r}$, where $\mathbf{c}_2$ denotes the last L components of $\mathbf{c}$.

Shamir sharing and Reed–Solomon consistency. A sharing of $\mathbf{m} \in R_q^L$ with threshold t is obtained from $P(Y) = \mathbf{m} + \sum_{\ell=1}^{t-1} \mathbf{a}_\ell Y^\ell$, $\mathbf{a}_\ell \leftarrow R_q^L$, by setting $\mathbf{m}_k = P(k)$ for $k \in [n]$. Any t shares determine $\mathbf{m} = \sum_{k\in T} \lambda_k^T \mathbf{m}_k$, and any $t-1$ shares are uniformly distributed and independent of $\mathbf{m}$.

Write $\alpha_0 = 0$ and $\alpha_k = k$ for $k \in [n]$. A word $(\mathbf{w}_0, \ldots, \mathbf{w}_n) \in (R_q^L)^{n+1}$ is of the form $(P(\alpha_0), \ldots, P(\alpha_n))$ for some P of degree less than t if and only if $\mathbf{H}\,(\mathbf{w}_0, \ldots, \mathbf{w}_n)^\top = \mathbf{0}$, where $\mathbf{H} \in \mathbb{Z}_q^{(n+1-t)\times(n+1)}$ is the parity-check matrix of the generalized Reed–Solomon code,

$$H_{j,i} = u_i\,\alpha_i^j, \qquad u_i = \prod_{\ell \neq i} (\alpha_i - \alpha_\ell)^{-1}, \qquad 0 \le j \le n-t,\ 0 \le i \le n.$$

The condition is applied componentwise; it involves only public coefficients in $\mathbb{Z}_q$.

Encryption with publicly verifiable opening. The seeds are encrypted to the authorities with a scheme $\mathsf{EO} = (\mathsf{KGen}, \mathsf{VerKey}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Open}, \mathsf{Judge})$ that lets the key holder show publicly what a ciphertext decrypts to, including that it decrypts to nothing. $\mathsf{KGen}(\mathsf{par})$ outputs a key pair $(\mathsf{pk}, \mathsf{sk})$, and $\mathsf{VerKey}(\mathsf{pk}) \in \{0,1\}$ is a public check of the key. Encryption and decryption take a label lb, which will be the voter identity: $\mathsf{Enc}(\mathsf{pk}, \mathsf{lb}, x)$ encrypts $x \in \{0,1\}^\lambda$, and the deterministic algorithm $\mathsf{Dec}(\mathsf{sk}, \mathsf{lb}, e)$ returns an element of $\{0,1\}^\lambda$ or $\perp$. $\mathsf{Open}(\mathsf{sk}, \mathsf{lb}, e)$ outputs evidence ω, and the deterministic algorithm $\mathsf{Judge}(\mathsf{pk}, \mathsf{lb}, e, \omega)$ returns an element of $\{0,1\}^\lambda \cup \{\perp\}$ or fail. We require:

(E1) *Correctness:* for honestly generated keys, $\mathsf{VerKey}(\mathsf{pk}) = 1$, and $\mathsf{Dec}(\mathsf{sk}, \mathsf{lb}, \mathsf{Enc}(\mathsf{pk}, \mathsf{lb}, x)) = x$ except with negligible probability.

(E2) *Completeness of openings:* except with negligible probability over the choice of honestly generated keys, for *every* pair (lb, e), including malformed ciphertexts, $\mathsf{Judge}(\mathsf{pk}, \mathsf{lb}, e, \mathsf{Open}(\mathsf{sk}, \mathsf{lb}, e)) = \mathsf{Dec}(\mathsf{sk}, \mathsf{lb}, e)$.

(E3) *Soundness of openings for honest ciphertexts:* no efficient adversary that outputs a key pk with $\mathsf{VerKey}(\mathsf{pk}) = 1$, a label lb and a plaintext x, then receives $e \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{lb}, x)$, can output ω with $\mathsf{Judge}(\mathsf{pk}, \mathsf{lb}, e, \omega) \notin \{x, \mathsf{fail}\}$, except with negligible probability. The key is chosen by the adversary, and the evidence after the ciphertext is seen.

(E4) *Confidentiality under openings:* for honestly generated keys, an adversary cannot distinguish $\mathsf{Enc}(\mathsf{pk}, \mathsf{lb}^*, x_0)$ from $\mathsf{Enc}(\mathsf{pk}, \mathsf{lb}^*, x_1)$ for $(\mathsf{lb}^*, x_0, x_1)$ of its choice, even with access, before and after the challenge, to $\mathsf{Open}(\mathsf{sk}, \cdot, \cdot)$ on every pair whose label differs from lb^*. We write $\mathrm{Adv}^{\mathrm{conf}}$ for its advantage.

By (E2), an authority can always justify a complaint, whatever the voter sent; by (E3), no authority can justify a complaint against an honest ciphertext. An authority whose key fails VerKey is disqualified: voters set their ciphertext to it to a fixed dummy string, its complaints are ignored, and it belongs to no quorum; by (E1) this happens only to corrupted authorities. Section 7 discusses instantiations: identity-based encryption with release of the user key, as in Epoque (Boyen et al., 2021) and (Kate et al., 2026), and encryption with proofs of correct decryption, as in PQKryvos (Huber et al., 2026).

Ballot proof. Π_{bal} is a non-interactive zero-knowledge proof of knowledge in the random-oracle model, with a label (the voter identity) bound into the proof, for the relation $\mathcal{R}_{\mathsf{bal}}$ with statement $(\mathbf{c}_0, \ldots, \mathbf{c}_n)$ and witness $(\mathbf{m}_j, \mathbf{r}_j)_{j=0}^n$ such that

1. $\mathsf{Com}(\mathbf{m}_j; \mathbf{r}_j) = \mathbf{c}_j$ and $\|\mathbf{r}_j\|_2 \le \beta$ for $j = 0, \ldots, n$;
2. $\mathbf{H}\,(\mathbf{m}_0, \ldots, \mathbf{m}_n)^\top = \mathbf{0}$;
3. $\mathbf{m}_0 \in \mathcal{V}_{L,w}$, that is, $\mathbf{m}_0 \in \{0,1\}^L$ and, if $w \neq *$, $\sum_{a=1}^L m_{0,a} = w$.

For Theorem 5 we require that Π_{bal} is *online extractable* with respect to a possibly relaxed relation $\mathcal{R}'_{\mathsf{bal}}$, identical to $\mathcal{R}_{\mathsf{bal}}$ except that the norm bound is $\beta' \ge \beta$: there is an extractor that, observing the adversary's random-oracle queries, outputs a witness in $\mathcal{R}'_{\mathsf{bal}}$ for every accepting proof whose label differs from those of honestly generated proofs, except with negligible probability. Section 7 discusses instantiations.

Authority aggregation. We abstract the tallying proofs of EVOLVE (del Pino et al., 2017, Sec. 4.2) as a pair $(\mathsf{Agg}, \mathsf{VerAgg})$. Given commitments $(\mathbf{c}_i)_{i\in I}$ and valid openings $(\mathbf{m}_i, \mathbf{r}_i)_{i\in I}$, Agg outputs the sum $\mathbf{V} = \sum_{i\in I} \mathbf{m}_i$ together with a proof Π, consisting of amortized proofs of short openings, a tree of recommitments with fresh short randomness and proofs of their consistency, and the opening of the root commitment. $\mathsf{VerAgg}((\mathbf{c}_i)_{i\in I}, \mathbf{V}, \Pi)$ accepts or rejects. We require completeness, zero knowledge (the proof can be simulated from the statement, $\mathbf{V}$ and, if Agg publishes commitments to inner nodes of a tree, these commitments, which are hiding commitments whose randomness is never revealed), and soundness: no efficient adversary outputs commitments $(\mathbf{c}_i)_{i\in I}$, openings $(\mathbf{m}_i, \mathbf{r}_i)_{i\in I}$ with $\|\mathbf{r}_i\|_2 \le \beta'$, and $(\mathbf{V}, \Pi)$ accepted by VerAgg with $\mathbf{V} \neq \sum_{i\in I} \mathbf{m}_i$, except with negligible probability, provided M-SIS is hard for the corresponding bound. The soundness argument of EVOLVE extracts openings with a common slack 2 and relies on binding (del Pino et al., 2017, App. A).

4 Syntax and Security Model

Participants. Voters $\mathsf{id} \in \mathcal{I}$, authorities $A_1, \ldots, A_n$, and an authenticated append-only bulletin board BB on which each voter writes at most once. An election is parameterized by (n, t, L, w) with $1 \le t \le n$ and $w \in [L] \cup \{*\}$. The set of admissible votes is $\mathcal{V}_{L,w} = \{\mathbf{v} \in \{0,1\}^L : \sum_a v_a = w\}$ if a weight $w \in [L]$ is prescribed, and $\mathcal{V}_{L,*} = \{0,1\}^L$ otherwise; a yes/no election is the case $L = 1$, $w = *$, and $w = 1$ gives a choice of one out of L candidates.

Board and phases. All parties see the same board. We assume that it is synchronous and does not censor: an entry that an honest party posts before a deadline appears on the board before that deadline for all parties. The keys pk_k are posted before the voting phase and cannot be changed, so all parties see the same set of disqualified authorities. Eligible voters are identified by signature keys registered in a public-key infrastructure; voters are corrupted statically, and the unforgeability assumption below is the multi-user EUF-CMA security of the signature scheme for the $|\mathcal{I}|$ keys, which loses at most a factor $|\mathcal{I}|$ with respect to single-user EUF-CMA security. An election runs in three phases—voting, checking and tallying—whose ends are fixed times measured on the board; entries posted after the end of a phase are ignored for that phase. If $\mathsf{Contrib}_k$ has several rounds, as in the EVOLVE-style instantiation, it is an interactive algorithm: each of its rounds has a deadline fixed in par (Section 7), and if a tally round does not produce t accepted contributions, the tally is repeated in a new tally round $\nu + 1$ with fresh randomness. An authority that announces a contribution in a tally round but does not complete it is recorded on the board as faulty and takes no part in later tally rounds; this is a public function of the board. For robustness, an authority that does not check the ballots during the checking phase, or does not post its messages in a tally round, is counted among the $n - t$ absent or malicious authorities. The first condition is needed because acceptance is fixed when the checking phase closes: an authority that did not complain about a ballot whose share it cannot open can no longer prove its contribution, since the aggregation proof opens every share. Nothing else excludes an authority that missed the checking phase; it can still take part in the tally if every accepted ballot carries a valid share for it. A single voter can, however, exclude an authority that is offline during the checking phase by posting a ballot whose ciphertext for this authority does not decrypt to an opening of its share, while the other ciphertexts do: the ballot proof does not cover the ciphertexts, so the ballot is accepted, and the authority can no longer contribute. Against an authority that is online, the same ballot is removed by its complaint. The checking phase can be made long enough that no honest authority misses it at little cost: checking a ballot costs, besides the public validity check, a decryption and one commitment evaluation per authority, and an authority may check ballots as they arrive or in one pass before the deadline. Acceptance and robustness below refer to this schedule. Two alternatives would weaken the requirement that an authority be present during the checking phase, and we discuss why we

did not adopt them. First, complaints could be accepted until the start of each tally round and the accepted set computed relative to the quorum, from the valid complaints of its members only. Honest ballots still could not be removed, by (E3), but the accepted set, and with it the result, would then depend on which dishonest ballots the quorum excludes, so two quorums could certify two different results from the same board; verifiability would have to be defined relative to a published quorum, and a late authority would still have to check every accepted ballot before contributing, so the deadline moves rather than disappears. Second, the ballot proof could cover the ciphertexts, as in publicly verifiable secret sharing: with a proof that each ciphertext decrypts to the randomness of the corresponding share, no checking phase is needed and every authority can contribute at any time. This requires, per authority, a lattice proof of correct encryption linked to the committed share (cf. Lyubashevsky et al., 2022), which would add a proof of that kind for each of the n ciphertexts to a ballot whose proof part is now 56 KB in total (Section 8), and would tie the ballot format to the encryption scheme; moreover, the ciphertexts of T-EVOLVE carry a seed from which the randomness is derived by the random oracle X, so the proof would have to cover that derivation, or the ciphertexts would have to carry the randomness itself, as in EVOLVE, at a multiple of their size. We keep the checking phase, whose cost is a decryption and one commitment evaluation per ballot and authority.

Syntax. A threshold voting scheme consists of the following algorithms.

- $\mathsf{Setup}(1^\lambda, n, t, L, w) \to \mathsf{par}$, derived deterministically from a public random seed, so that no trusted party is needed, and $\mathsf{ASetup}_k(\mathsf{par}) \to (\mathsf{pk}_k, \mathsf{sk}_k)$ for $k \in [n]$; pk_k is published, and voters use it only if $\mathsf{VerKey}(\mathsf{pk}_k) = 1$.
- $\mathsf{Vote}(\mathsf{par}, (\mathsf{pk}_k)_k, \mathsf{id}, \mathbf{v}) \to b$, for $\mathbf{v} \in \mathcal{V}_{L,w}$.
- $\mathsf{Valid}(\mathsf{par}, \mathsf{BB}, b) \in \{0,1\}$: public validity of a ballot.
- $\mathsf{Check}_k(\mathsf{par}, \mathsf{BB}, \mathsf{sk}_k, b)$, which outputs a complaint γ or nothing.
- $\mathsf{VerComplaint}(\mathsf{par}, \mathsf{BB}, b, \gamma) \in \{0,1\}$.
- $\mathsf{Contrib}_k(\mathsf{par}, \mathsf{BB}, \mathsf{sk}_k) \to (\mathbf{V}_k, \Pi_k)$: the contribution of authority k to the tally, possibly computed in several rounds that post messages on the board; the contribution is labeled with its tally round ν.
- $\mathsf{Combine}(\mathsf{par}, \mathsf{BB}, T, (\mathbf{V}_k, \Pi_k)_{k\in T}) \to \mathbf{R}$ for $|T| = t$.
- $\mathsf{Verify}(\mathsf{par}, \mathsf{BB}, \mathbf{R}, T, (\mathbf{V}_k, \Pi_k)_{k\in T}) \in \{0,1\}$, which accepts only contributions of one tally round.

The set of *accepted* ballots $\mathsf{Acc}(\mathsf{BB})$ consists of the ballots b posted during the voting phase with $\mathsf{Valid} = 1$ for which no complaint γ posted by a non-disqualified authority during the checking phase satisfies $\mathsf{VerComplaint} = 1$. Acceptance is a public function of the board as it stands when the checking phase closes; complaints posted later are ignored. Unlike in EVOLVE, it does not require an approval from every authority.

Correctness. If all parties are honest, every ballot is accepted, and for every $T \subseteq [n]$ with $|T| = t$, $\mathsf{Combine}$ outputs $\mathbf{R} = \sum_i \mathbf{v}_i$ and Verify accepts, with overwhelming probability. If $\mathsf{Contrib}_k$ may abort, as in the EVOLVE-style instantiation, this refers to the first tally round in which all authorities of T complete their contributions, and we require that such a round occurs within N_{rd} tally rounds except with negligible probability.

Privacy. We adapt the privacy experiment of EVOLVE (del Pino et al., 2017) to the threshold setting. Corruption is static: before the election, the adversary chooses a set $\mathcal{C} \subseteq [n]$ of at most $t-1$ corrupted authorities and their keys; the challenger generates the keys of the others. The adversary has access to BB and to oracles $\mathsf{OVote}(\mathsf{id}, \mathbf{v}^0, \mathbf{v}^1)$, which posts $\mathsf{Vote}(\ldots, \mathsf{id}, \mathbf{v}^\theta)$ and records $(\mathsf{id}, \mathbf{v}^0, \mathbf{v}^1, b)$ in HV; $\mathsf{OCheck}(b)$, which accepts only an entry b that is on BB exactly as posted and satisfies $\mathsf{Valid}(\mathsf{par}, \mathsf{BB}, b) = 1$, runs Check_k on it for all honest k and posts the resulting complaints, and on any other input returns $\perp$ without decrypting anything; and $\mathsf{OTally}(T)$, callable once, which closes the voting phase, runs Check_k for all honest k on every valid entry of the voting phase that was not passed to OCheck and posts the resulting complaints, as honest authorities do in the checking phase, then closes the checking phase and runs $\mathsf{Contrib}_k$ for the honest $k \in T$; if $\mathsf{Contrib}_k$ has several rounds, OTally runs them one at a time, returns the messages of each round, and lets the adversary post its own messages before the next round begins. OVote accepts only identities of voters that the adversary did not corrupt at the start and refuses identities that have already voted; after OTally, no further ballots or complaints are posted. The board is authenticated, so a complaint $(k, \cdot)$ is attributed to authority k. The experiment returns the adversary's guess θ' if $\sum \mathbf{v}^0 = \sum \mathbf{v}^1$, where the sums range over the entries of HV whose ballots are accepted, and $\perp$ otherwise. The scheme is private if $\left|\Pr[\theta' = 1 \mid \theta = 0] - \Pr[\theta' = 1 \mid \theta = 1]\right|$ is negligible.

Relation to BPRIV. The definition follows EVOLVE and the classical vote-swapping definitions: the adversary learns θ' only if the two sequences of honest votes give the same result. BPRIV (Bernhard et al., 2015) is now the reference game-based notion of ballot privacy. It lets the two sequences differ, returns the result computed on the left sequence, and is accompanied by strong correctness and strong consistency. BPRIV assumes that the tally is computed by an honest party holding the decryption key, whereas here up to $t-1$ tallying authorities are corrupted and take part in a multi-round aggregation. Two things stand in the way of a direct adaptation. First, the BPRIV tally oracle returns the result of the left board together with a simulated tally proof. Here the "proof" is the joint transcript of the aggregation, in which the corrupted authorities of the quorum post their own contributions. A simulator would have to produce honest contributions that pass VerAgg against column sums $\mathbf{V}_k$ consistent with the left result while the shares on the right board interpolate to another one, whereas the zero-knowledge simulator of Section 7.3 simulates one authority's proofs for given public values, not public values of its choice. Second, the multi-board variant mb-BPRIV of Cortier et al. (2020) drops the assumption of an honest board; in a threshold protocol a malicious board could show different first-round boards BB_1 to different authorities and split the quorum, so our assumption of one authenticated board seen by all parties (Section 4) is essential and would have to be replaced by a mechanism such as signed board digests. Adapting BPRIV, mb-BPRIV and their companion properties strong correctness and strong consistency to this setting is beyond this paper, and we do not claim privacy in the sense of BPRIV. Likewise, our verifiability notions follow EVOLVE; Cortier et al. (2016) survey and compare such notions.

Long-term privacy. The privacy of T-EVOLVE is computational, as in EVOLVE, EVOLVED, Epoque and PQKryvos: the board stores, for every ballot, the commitment to the vote, the n commitments to its shares and the n seed ciphertexts, and all of them hide the vote only under M-LWE (the commitments) and under the security of EO (the ciphertexts). An adversary who records the board and later solves M-LWE at the chosen parameters, or breaks the encryption, learns every vote; nothing on the board is information-theoretically hidden except through the threshold itself, which protects only against coalitions of authorities, not against a future solver. Since the board of an election is public and may be archived, this is the "harvest now, decrypt later" setting that motivates post-quantum voting, and the relevant margin is the hiding estimate of Table 3 over the lifetime of the secrecy requirement: $2^{128.9}$ at $N_V = 10^6$ with $d = 7$ and $2^{148.5}$ with $d = 8$ under the classical MATZOV model, less under the other models. Everlasting privacy in the sense surveyed by Haines et al. (2023) would require that everything public be

statistically hiding. For the commitments this means randomness with more min-entropy than the commitment can absorb, about $N(d+L)\log_2 q$ bits for $N(2d+L)$ coefficients, that is, $\sigma \approx 2^{21}$ instead of $\sigma = 1$ at the parameters for $N_V = 10^4$; every norm bound of Section 7.5, hence β_{SIS} and q, would grow by the same factor, d would have to grow with q, and we have not run the parameter chain for such a variant. For the ciphertexts it would mean removing them from the public board, which is incompatible with public complaints: the openings would have to reach the authorities over private channels, and the guarantee that no authority can have an honest ballot removed would be lost. T-EVOLVE therefore trades everlasting privacy for publicly verifiable complaints, and we list the combination as open.

Verifiability. We adapt the consistency notion of EVOLVE, including its extension to L candidates (del Pino et al., 2017, App. B). The adversary controls *all* authorities and generates all their keys, controls all voters except those created through $\mathsf{OVote}(\mathsf{id}, \mathbf{v})$, and finally outputs $(\mathbf{R}, T, (\mathbf{V}_k, \Pi_k)_{k\in T})$. Let Hon be the set of accepted ballots produced by OVote, with votes $\mathbf{v}_i$, and $\mathsf{Dis} = \mathsf{Acc}(\mathsf{BB}) \setminus \mathsf{Hon}$. The adversary wins if Verify accepts and there are no votes $\mathbf{u}_i \in \mathcal{V}_{L,w}$ for $i \in \mathsf{Dis}$ such that $\mathbf{R} = \sum_{i\in\mathsf{Hon}} \mathbf{v}_i + \sum_{i\in\mathsf{Dis}} \mathbf{u}_i$, where the entries of $\mathbf{R}$, which Verify requires to be constant polynomials, are identified with their constant coefficients in $[0, q)$. The scheme is verifiable if every efficient adversary wins with negligible probability. Thus every accepted dishonest ballot counts as exactly one admissible vote, whatever the authorities do. For $w = *$ the coordinates are independent, and the condition says that $h_a^1 \le R_a \le |\mathsf{Acc}(\mathsf{BB})| - h_a^0$ for every $a \in [L]$, where h_a^1 and h_a^0 are the numbers of ballots in Hon with $v_a = 1$ and $v_a = 0$; this is the notion of EVOLVE. For $w \neq *$ the per-candidate condition would not suffice: for $L = 2$ and $w = 1$, a single dishonest ballot with vote $(1, 1)$ keeps both counts within these ranges, although $R_1 + R_2$ exceeds $w\,|\mathsf{Acc}(\mathsf{BB})|$.

Individual verifiability. A voter checks that her ballot appears on BB and belongs to $\mathsf{Acc}(\mathsf{BB})$; together with verifiability of the tally over $\mathsf{Acc}(\mathsf{BB})$, this means that her vote is counted in any result that verifies. We require that honest ballots cannot be excluded even if *all* authorities are corrupted: in the experiment, the adversary controls all authorities and all other voters, obtains honest ballots through OVote and may post arbitrary complaints; it wins if some honest ballot is not in $\mathsf{Acc}(\mathsf{BB})$. Whether the voter's device encrypts the intended vote (cast-as-intended verifiability) is outside the model; techniques such as return codes (Boyen et al., 2021) or ballot auditing apply to T-EVOLVE as to EVOLVE.

Robustness. The adversary statically corrupts a set $\mathcal{C}$ of at most $n - t$ authorities and any number of voters, and may post arbitrary ballots and complaints. Honest voters vote through OVote, and honest authorities follow the protocol. The adversary wins if an honest ballot is not accepted, or if, within N_{rd} tally rounds, the honest authorities do not obtain some $T \subseteq [n] \setminus \mathcal{C}$ with $|T| = t$ and contributions of one round that Verify accepts. The scheme is robust with round success probability p_{rd} if every efficient adversary wins with probability at most $(1 - p_{\mathrm{rd}})^{N_{\mathrm{rd}}}$ plus a negligible term; for a one-round $\mathsf{Contrib}_k$, $p_{\mathrm{rd}} = 1$.

Privacy and robustness constrain t in opposite directions: privacy tolerates $t - 1$ corrupted authorities and robustness tolerates $n - t$ absent or malicious ones. For $n = 5$, the choice $t = 3$ tolerates two of each.

5 The Protocol

The protocol is given in Figure 2. We comment on the steps that differ from EVOLVE.

Ballots. A ballot contains $n + 1$ commitments instead of n: one to each share and one, $\mathbf{c}_0$, to the vote itself. Committing only to the t coefficients of the sharing polynomial, as in Feldman-

$\mathsf{Setup}(1^\lambda, n, t, L, w)$**:** choose N, q, d, σ, β with $q > n$ prime; expand $\mathbf{C}$ from a public random seed; compute $\mathbf{H}$; output par. Let $\mathsf{Rnd}_{\mathsf{id},k}(s) = \mathsf{SampleD}_\sigma(X(\mathsf{par}, \mathsf{id}, k, s))$ denote the vector in $D_\sigma^{N(2d+L)}$ sampled from the output of an extendable-output function X (modeled as a random oracle) on the parameters, the voter identity id, the share index k and a seed $s \in \{0,1\}^\lambda$. Different shares thus use different inputs to X, even if a voter reuses a seed.

$\mathsf{ASetup}_k(\mathsf{par})$**:** $(\mathsf{pk}_k, \mathsf{sk}_k) \leftarrow \mathsf{EO.KGen}(\mathsf{par})$; publish pk_k.

$\mathsf{Vote}(\mathsf{par}, (\mathsf{pk}_k)_k, \mathsf{id}, \mathbf{v})$**:**

1. Sample $\mathbf{a}_1, \ldots, \mathbf{a}_{t-1} \leftarrow R_q^L$; let $P(Y) = \mathbf{v} + \sum_\ell \mathbf{a}_\ell Y^\ell$ and $\mathbf{m}_0 = \mathbf{v}$, $\mathbf{m}_k = P(k)$ for $k \in [n]$.
2. Sample $\mathbf{r}_0 \leftarrow D_\sigma^{N(2d+L)}$ and, for $k \in [n]$, seeds $s_k \leftarrow \{0,1\}^\lambda$ and $\mathbf{r}_k = \mathsf{Rnd}_{\mathsf{id},k}(s_k)$. If $\|\mathbf{r}_j\|_2 > \beta$ for some j, which happens with probability below $2^{-1.16N(2d+L)}$ for $\beta = 2\sigma\sqrt{N(2d+L)}$ (Lyubashevsky, 2012, Lemma 4.4), sample again. For $j = 0, \ldots, n$ set $\mathbf{c}_j = \mathsf{Com}(\mathbf{m}_j; \mathbf{r}_j)$.
3. For $k \in [n]$: if $\mathsf{VerKey}(\mathsf{pk}_k) = 1$, $e_k \leftarrow \mathsf{Enc}(\mathsf{pk}_k, \mathsf{id}, s_k)$; otherwise A_k is disqualified and e_k is a fixed dummy string.
4. $\pi \leftarrow \Pi_{\mathsf{bal}}.\mathsf{Prove}\big((\mathbf{c}_j)_{j=0}^n; (\mathbf{m}_j, \mathbf{r}_j)_{j=0}^n\big)$, bound to id.
5. Output $b = \big(\mathsf{id}, (\mathbf{c}_j)_{j=0}^n, (e_k)_{k=1}^n, \pi\big)$, signed by id.

$\mathsf{Valid}(\mathsf{par}, \mathsf{BB}, b)$**:** check the signature, that id has not voted before, and that π verifies.

$\mathsf{Check}_k(\mathsf{par}, \mathsf{BB}, \mathsf{sk}_k, b)$**:** if b is not an entry of BB with $\mathsf{Valid}(\mathsf{par}, \mathsf{BB}, b) = 1$, output nothing and do not decrypt. Otherwise let id be the identity of b and $x \leftarrow \mathsf{Dec}(\mathsf{sk}_k, \mathsf{id}, e_k)$. If $x \neq \perp$ and $\mathsf{Rnd}_{\mathsf{id},k}(x)$ is a valid opening of $\mathbf{c}_k$ under β (that is, $\mathbf{r} = \mathsf{Rnd}_{\mathsf{id},k}(x)$ satisfies $\|\mathbf{r}\|_2 \le \beta$ and the first d components of $\mathbf{c}_k - \mathbf{C}\mathbf{r}$ vanish), output nothing; otherwise output $\gamma = (k, \mathsf{Open}(\mathsf{sk}_k, \mathsf{id}, e_k))$.

$\mathsf{VerComplaint}(\mathsf{par}, \mathsf{BB}, b, \gamma)$**:** accept iff $\gamma = (k, \omega)$ was posted by A_k, A_k is not disqualified, $x' = \mathsf{Judge}(\mathsf{pk}_k, \mathsf{id}, e_k, \omega) \neq \mathsf{fail}$, and either $x' = \perp$ or $\mathsf{Rnd}_{\mathsf{id},k}(x')$ is not a valid opening of $\mathbf{c}_k$ under β.

$\mathsf{Contrib}_k(\mathsf{par}, \mathsf{BB}, \mathsf{sk}_k)$**:** (two rounds in the EVOLVE-style instantiation, see Section 7) for each $b_i \in \mathsf{Acc}(\mathsf{BB})$, let $s_{i,k} = \mathsf{Dec}(\mathsf{sk}_k, \mathsf{id}_i, e_{i,k})$, set $\mathbf{r}_{i,k} = \mathsf{Rnd}_{\mathsf{id}_i,k}(s_{i,k})$ and $\mathbf{m}_{i,k} = \mathbf{c}_{i,k,2} - \mathbf{B}\mathbf{r}_{i,k}$; output $(\mathbf{V}_k, \Pi_k) \leftarrow \mathsf{Agg}\big((\mathbf{c}_{i,k}, \mathbf{m}_{i,k}, \mathbf{r}_{i,k})_{b_i \in \mathsf{Acc}(\mathsf{BB})}\big)$.

$\mathsf{Combine}(\mathsf{par}, \mathsf{BB}, T, (\mathbf{V}_k, \Pi_k)_{k \in T})$**:** output $\mathbf{R} = \sum_{k \in T} \lambda_k^T \mathbf{V}_k$.
$\mathsf{Verify}(\mathsf{par}, \mathsf{BB}, \mathbf{R}, T, (\mathbf{V}_k, \Pi_k)_{k \in T})$**:** recompute $\mathsf{Acc}(\mathsf{BB})$; check $|T| = t$ and $\mathsf{VerKey}(\mathsf{pk}_k) = 1$ for all $k \in T$; check $\mathsf{VerAgg}\big((\mathbf{c}_{i,k})_{b_i \in \mathsf{Acc}}, \mathbf{V}_k, \Pi_k\big) = 1$ for all $k \in T$; check $\mathbf{R} = \sum_{k \in T} \lambda_k^T \mathbf{V}_k$ and that each entry R_a of $\mathbf{R}$ is a constant polynomial whose constant coefficient, as an integer in $[0, q)$, lies in $[0, |\mathsf{Acc}(\mathsf{BB})|]$.

Figure 2: T-EVOLVE.

style verifiable secret sharing, would be shorter, and commitments to the shares would follow homomorphically. But the randomness of the commitment to share k would then be $\sum_\ell k^\ell \boldsymbol{\rho}_\ell$, with integer coefficients up to n^{t-1}. This has two drawbacks. First, the norm of this randomness would grow by a factor of about n^{t-1}, and the randomness would no longer be an output of X, so the bounds of Lemma 13 would not apply. Second, each authority would learn one such combination, so a coalition of $t-1$ authorities would obtain $t-1$ combinations of the t short vectors, including the randomness of the commitment to the vote. This is the kind of information exploited against Shamir-based threshold FHE (Cho et al., 2024). Fresh randomness per share avoids both problems. The commitment $\mathbf{c}_0$ cannot be replaced by the homomorphic combination $\sum_{k \in T} \lambda_k^T \mathbf{c}_k$ either: the Lagrange coefficients are arbitrary elements of $\mathbb{Z}_q$, so the randomness of that combination is not short and the OR-proof of a binary vote could not be given for it. The randomness of $\mathbf{c}_0$ is never revealed; $\mathbf{c}_0$ only serves as the anchor for the validity statement. The ballot proof replaces EVOLVE's OR-proof on the sum of additive shares by the statement $\mathcal{R}_{\mathsf{bal}}$, which ties the committed shares to a binary vote of the prescribed weight through the parity-check equations. Whereas EVOLVE encrypts the randomness itself, here only a seed is encrypted to each authority, from which it derives the randomness of its share; the share is

determined by the commitment and the randomness. The seed makes the ciphertexts short. It also makes the randomness of every share distributed as D_σ, including the shares of dishonest voters, up to the voter's choice among its queries to X. Since the input of X contains the identity and the share index, the randomness of different shares comes from different inputs, even if a voter reuses a seed; the bounds of the aggregation proofs in Section 7 use this.

Acceptance and complaints. In EVOLVE, a ballot is valid only once all authorities have approved it, and an authority's rejection cannot be verified publicly. As the authors of Epoque point out (Boyen et al., 2021), this allows a malicious authority to remove ballots. Here, validity of the ballot proof is public. An authority can have a ballot removed only by a complaint whose evidence shows publicly that its ciphertext decrypts to nothing ($\perp$) or to a seed that does not open its share. By (E3), such evidence cannot be produced for an honest ballot, so a false complaint is rejected by $\mathsf{VerComplaint}$ and has no effect. Conversely, by (E2) a voter cannot prevent an honest authority from complaining: whatever the voter encrypts, including a ciphertext that does not decrypt at all, the authority can show it, and the ballot is excluded. Every accepted ballot can therefore be opened by every honest authority that takes part in the checking phase.

Tallying. Each authority k that participates computes the sum $\mathbf{V}_k = \sum_i \mathbf{m}_{i,k}$ of its column of shares and publishes it with an aggregation proof, exactly as an EVOLVE authority publishes its partial tally. Since the shares of each accepted ballot lie on a polynomial of degree less than t, so do the column sums: $\mathbf{V}_k = \sum_i P_i(k)$. Interpolating any t of them at 0 yields $\sum_i P_i(0) = \sum_i \mathbf{v}_i$. The Lagrange coefficients λ_k^T are arbitrary elements of $\mathbb{Z}_q$, but they multiply only public values; commitment randomness is never combined across authorities, so no norm growth occurs. If more than t authorities publish valid contributions, any t of them give the same result.

6 Security

Throughout, par is derived from a public random seed and treated as honestly generated, ballots are signed with an EUF-CMA secure signature scheme under the voter's identity, and the number of ballots is smaller than q. We write $\mathsf{Acc} = \mathsf{Acc}(\mathsf{BB})$ for the final set of accepted ballots, and $\mathsf{Hon} \subseteq \mathsf{Acc}$ and $\mathsf{Dis} = \mathsf{Acc} \setminus \mathsf{Hon}$ for the accepted ballots produced by OVote and by the adversary, respectively.

6.1 Correctness and privacy

Theorem 1 (Correctness)**.** *T-EVOLVE is correct if* EO *satisfies (E1) and the parameters satisfy the completeness conditions of* Π_{bal} *and* $(\mathsf{Agg}, \mathsf{VerAgg})$*.*

Proof. Honest shares are valid openings, so no complaint is raised and all ballots are accepted. Each $\mathbf{V}_k$ equals $\sum_i P_i(k)$, interpolation over the t points of T recovers $\sum_i \mathbf{v}_i$, and its entries are integers at most the number of ballots, hence smaller than q. For the two-round aggregation of Section 7, a tally round in which the authorities of T are honest succeeds with probability at least $s(\kappa)^t$ minus the statistical errors of its rejection steps, independently of earlier rounds (Section 7.3), so $N_{\mathrm{rd}} = \omega(\log \lambda)$ rounds suffice except with negligible probability. □

Lemma 2 (Shares seen by a coalition)**.** *Let* $\mathcal{C} \subseteq [n]$ *with* $|\mathcal{C}| \le t-1$*, let* Hon *be a nonempty finite index set, and for* $i \in \mathsf{Hon}$ *let* $P_i(Y) = \mathbf{v}_i + \sum_{\ell=1}^{t-1} \mathbf{a}_{i,\ell} Y^\ell$ *with independent uniform* $\mathbf{a}_{i,\ell} \in R_q^L$*. Then*

1. *for each* i*, the shares* $(P_i(j))_{j\in\mathcal{C}}$ *are uniformly distributed in* $(R_q^L)^{\mathcal{C}}$*, independently of* $\mathbf{v}_i$*;*

2. *conditioned on all shares* $(P_i(j))_{i\in\mathsf{Hon},j\in\mathcal{C}}$*, the polynomial* $P_{\mathsf{Hon}} = \sum_{i\in\mathsf{Hon}} P_i$ *is uniformly distributed over the set of polynomials* P *of degree less than* t *with* $P(0) = \sum_{i\in\mathsf{Hon}} \mathbf{v}_i$ *and* $P(j) = \sum_{i\in\mathsf{Hon}} P_i(j)$ *for* $j \in \mathcal{C}$*.*

In particular, the joint distribution of these shares and of P_{Hon} *depends on the votes only through* $\sum_{i\in\mathsf{Hon}} \mathbf{v}_i$*.*

Proof. All statements are coordinatewise over $\mathbb{Z}_q$, since $R_q^L \cong \mathbb{Z}_q^{NL}$ as a $\mathbb{Z}_q$-vector space and the evaluation points lie in $\mathbb{Z}_q$. (1) The map $(\mathbf{a}_{i,1},\dots,\mathbf{a}_{i,t-1}) \mapsto (P_i(j) - \mathbf{v}_i)_{j\in\mathcal{C}}$ is linear with matrix $M = (j^\ell)_{j\in\mathcal{C},\,1\le\ell\le t-1} = \mathrm{diag}(j)_{j\in\mathcal{C}} \cdot (j^{\ell-1})_{j,\ell}$. The second factor consists of $|\mathcal{C}| \le t-1$ rows of a Vandermonde matrix at distinct points, and the first is invertible because the points are nonzero; hence M has full row rank and maps the uniform distribution to the uniform distribution on $(R_q^L)^{\mathcal{C}}$. Adding $\mathbf{v}_i$ preserves uniformity. (2) Conditioned on its shares $\mathbf{s}_i = (P_i(j))_{j\in\mathcal{C}}$, each P_i is uniform on $A_i = \{P : \deg P < t,\ P(0) = \mathbf{v}_i,\ P(j) = \mathbf{s}_{i,j}\ \forall j \in \mathcal{C}\}$, which is a coset $x_i + W$ of the linear space $W = \{P : \deg P < t,\ P(0) = 0,\ P(j) = 0\ \forall j \in \mathcal{C}\}$, and the P_i are independent. A sum of independent random variables, each uniform on a coset of W, is uniform on the sum of the cosets, $\sum_i x_i + W$, which is the set in the statement. □

Theorem 3 (Privacy)**.** *Model* X *as a random oracle and assume that* Com *is hiding (M-LWE),* EO *satisfies (E1), (E2) and (E4), the signature scheme is EUF-CMA secure, and* Π_{bal} *and* Agg *are zero-knowledge. Then T-EVOLVE is private against adversaries statically corrupting at most* $t-1$ *authorities. More precisely, for an adversary making at most* q_V *queries to* OVote*,*

$$\begin{aligned}\mathrm{Adv}^{\mathrm{priv}}_{\mathcal{A}} \;\le\; 2\Big(&\varepsilon_{\mathrm{zk}} + n\,q_V\,\mathrm{Adv}^{\mathrm{conf}}_{\mathcal{B}_1} + \frac{n\,q_V\,q_X}{2^\lambda}\\ &+ \big((n+1)\,q_V + n_{\mathrm{in}}\big)\,\mathrm{Adv}^{\mathrm{hide}}_{\mathcal{B}_2} + \mathrm{Adv}^{\text{euf-cma}}_{\mathcal{B}_3} + n\,q_V\,\varepsilon_{\mathrm{dec}}\Big),\end{aligned}$$

where $\varepsilon_{\mathrm{zk}}$ *bounds the total distinguishing advantage of the zero-knowledge simulators over all simulated proofs,* q_X *is the number of queries of* $\mathcal{A}$ *to* X*,* n_{in} *bounds, over all executions, the number of commitments to nodes of level at least* 1 *other than the root that honest authorities publish over all tally rounds (*$n_{\mathrm{in}} = 0$ *if* Agg *publishes no such commitments), and* $\varepsilon_{\mathrm{dec}}$ *bounds the probability that decryption of an honest ciphertext under an honest key fails (E1).*

Proof. Fix $\theta \in \{0,1\}$. We define a sequence of games, starting from the privacy experiment $\mathbf{G}_0^\theta$, and end with a game whose output distribution does not depend on θ. Let $\mathcal{C}$ be the set of corrupted authorities, $|\mathcal{C}| \le t-1$, and $\mathcal{H} = [n] \setminus \mathcal{C}$. For an honest ballot, P_i denotes the sharing polynomial of $\mathbf{v}_i^\theta$.

Game $\mathbf{G}_1$*: bookkeeping.* (a) When an honest authority processes an honest ballot, in OCheck or in OTally, the challenger does not decrypt: it outputs no complaint and uses the share $P_i(k)$ it generated. By correctness (E1) for the honestly generated key, decryption would return the same values, so the change is invisible except with probability at most $nq_V\varepsilon_{\mathrm{dec}}$. (b) Abort if some ballot on the board carries the identity of an honest voter but was not produced by OVote; this is bounded by $\mathrm{Adv}^{\text{euf-cma}}_{\mathcal{B}_3}$. Hence every adversarial ballot carries a label, i.e. an identity, that differs from the identities of all honest ballots. Honest authorities process only valid entries of BB, in OCheck, in the checking step of OTally and in $\mathsf{Contrib}_k$. Therefore every ballot they process is either exactly an output of OVote or carries an identity of no honest voter.

Game $\mathbf{G}_2$*: simulated proofs.* Replace the proofs produced by honest parties—the ballot proofs of honest ballots and the aggregation proofs returned by OTally—by outputs of the zero-knowledge simulators, which program the random oracle (for the aggregation proofs only the oracle G of Section 7). The simulated aggregation proof for authority k is computed from the statement $\big((\mathbf{c}_{i,k})_{i\in\mathsf{Acc}}, \mathbf{V}_k\big)$, where $\mathbf{V}_k$ is computed as in $\mathbf{G}_1$, and, if Agg publishes commitments to inner nodes of a tree, from these commitments, which are still computed honestly. The change costs at most $\varepsilon_{\mathrm{zk}}$.

Game $\mathbf{G}_2'$*: inner nodes.* In the EVOLVE-style instantiation, the inner nodes of the tree of an honest authority k commit to partial sums of its column $(P_i(k))_i$ over consecutive ballots; together with the shares of the coalition, their messages would reveal partial sums of honest votes. Replace every commitment to an inner node published by an honest authority, in every tally round, by a commitment to $\mathbf{0}$ with fresh randomness; the root, a fresh commitment to $\mathbf{V}_k$ that is opened in the clear, is kept. After $\mathbf{G}_2$ the randomness of an inner node is used only in its commitment: it is never revealed, and the proofs that involve it are simulated from the commitments. A hybrid over the n_{in} commitments reduces each step to the hiding property, and the reduction knows all other secrets. This costs at most $n_{\text{in}} \operatorname{Adv}^{\text{hide}}_{\mathcal{B}_2}$.

Game $\mathbf{G}_3$*: encryptions to honest authorities.* For every honest ballot with identity id and every $k \in \mathcal{H}$, replace $e_k = \mathsf{Enc}(\mathsf{pk}_k, \mathsf{id}, s_k)$ by $\mathsf{Enc}(\mathsf{pk}_k, \mathsf{id}, 0^\lambda)$. A hybrid over the at most nq_V such ciphertexts reduces each step to confidentiality under openings (E4). The reduction receives pk_k from its challenger and places the challenge ciphertext in the ballot; by $\mathbf{G}_1$(a), no honest ciphertext has to be decrypted. Every adversarial ballot carries an identity id' that differs from all honest identities by $\mathbf{G}_1$(b), so the reduction may query $\omega = \mathsf{Open}(\mathsf{sk}_k, \mathsf{id}', e)$, even if e is copied from an honest ballot, and obtains $\mathsf{Dec}(\mathsf{sk}_k, \mathsf{id}', e) = \mathsf{Judge}(\mathsf{pk}_k, \mathsf{id}', e, \omega)$ by (E2). It uses these values to decide on complaints, posts (k, ω) as the complaint of A_k, and uses the decrypted seeds of accepted adversarial ballots in the tally. Earlier ciphertexts of the hybrid, already replaced, are handled in the same way. This costs at most $nq_V \operatorname{Adv}^{\text{conf}}_{\mathcal{B}_1}$.

Game $\mathbf{G}_3'$*: fresh randomness.* For every honest ballot and every $k \in \mathcal{H}$, replace $\mathbf{r}_k = \mathsf{Rnd}_{\mathsf{id},k}(s_k)$ by a fresh sample of $D_\sigma^{N(2d+L)}$, conditioned on norm at most β as in Vote. After $\mathbf{G}_3$ the seed s_k is used only in the input $(\mathsf{par}, \mathsf{id}, k, s_k)$ to X. The two games differ only if X is evaluated at this input elsewhere. Honest inputs are pairwise distinct, since OVote refuses repeated identities and the inputs of one ballot differ in k. Honest authorities evaluate X also on seeds decrypted from adversarial ballots, but by $\mathbf{G}_1$(b) these ballots carry identities of no honest voter, so these inputs differ from all honest ones. The adversary queries one of the honest inputs with probability at most $nq_V q_X 2^{-\lambda}$, since the seeds are uniform.

Game $\mathbf{G}_4$*: commitments not seen by the coalition.* For every honest ballot, replace $\mathbf{c}_0$ and $\mathbf{c}_k$, $k \in \mathcal{H}$, by commitments to $\mathbf{0}$ with fresh randomness. After $\mathbf{G}_3'$ the randomness of these commitments is used nowhere else: neither it nor a seed of it is encrypted, and all proofs involving it are simulated. A hybrid over the at most $(n+1)q_V$ commitments reduces each step to the hiding property. The reduction generates the honest authorities' keys itself, so it can decrypt and open the ciphertexts of adversarial ballots, and it knows the shares $P_i(k)$, which it needs for $\mathbf{V}_k$. This costs at most $(n+1)q_V \operatorname{Adv}^{\text{hide}}_{\mathcal{B}_2}$.

In $\mathbf{G}_4$ the adversary's view depends on the honest votes only through (a) the shares $P_i(j)$, $j \in \mathcal{C}$, which appear in the commitments $\mathbf{c}_j$ and ciphertexts e_j addressed to corrupted authorities, and (b) the column sums $\mathbf{V}_k$, $k \in T \cap \mathcal{H}$, returned by OTally. Write $\mathsf{Hon} = \{i \in \mathsf{Acc} : b_i \text{ honest}\}$ and $\mathsf{Dis} = \mathsf{Acc} \setminus \mathsf{Hon}$. Then

$$\mathbf{V}_k \;=\; P_{\mathsf{Hon}}(k) + \sum_{i \in \mathsf{Dis}} \mathbf{m}_{i,k}, \qquad P_{\mathsf{Hon}} = \sum_{i \in \mathsf{Hon}} P_i,$$

where the second sum is computed from the adversarial ballots by decryption and is a deterministic function of the board and the honest keys: for $i \in \mathsf{Dis}$, $\mathbf{m}_{i,k} = \mathbf{c}_{i,k,2} - \mathbf{B}\,\mathsf{Rnd}_{\mathsf{id}_i,k}(x_{i,k})$ with $x_{i,k} = \mathsf{Dec}(\mathsf{sk}_k, \mathsf{id}_i, e_{i,k})$, so it is determined even for an adversarial ballot that copies a commitment $\mathbf{c}_{j,k}$ of an honest ballot: under the new label id_i, the decrypted seed yields an opening of $\mathbf{c}_{j,k}$ only with negligible probability, so such a ballot is removed by the complaint of authority k (Remark (1) below), and in either case $\mathbf{m}_{i,k}$ is a function of the board and the honest keys.

Game $\mathbf{G}_5$*: lazy sampling.* Instead of choosing a polynomial per honest ballot, OVote samples the shares $(P_i(j))_{j \in \mathcal{C}}$ uniformly and independently of the vote, and OTally samples P_{Hon} uniformly among the polynomials of degree less than t with $P_{\mathsf{Hon}}(0) = \sum_{i \in \mathsf{Hon}} \mathbf{v}_i^\theta$ and $P_{\mathsf{Hon}}(j) = \sum_{i \in \mathsf{Hon}} P_i(j)$ for $j \in \mathcal{C}$. If $\mathsf{Hon} = \emptyset$, set $P_{\mathsf{Hon}} = 0$. By Lemma 2, the joint distribution of all quantities in the view is unchanged, so $\mathbf{G}_5$ and $\mathbf{G}_4$ are identical.

Conclusion. In $\mathbf{G}_5$, everything the adversary sees before OTally is independent of θ. In particular, the set Acc, the set Hon and the vote pairs $(\mathbf{v}_i^0, \mathbf{v}_i^1)_{i\in\mathsf{Hon}}$ have the same distribution for both values of θ, and so does the event E that $\sum_{i\in\mathsf{Hon}} \mathbf{v}_i^0 = \sum_{i\in\mathsf{Hon}} \mathbf{v}_i^1$. If E fails, both experiments output $\perp$. If E holds, the only θ-dependent input to OTally, namely $P_{\mathsf{Hon}}(0)$, is the same for both values of θ, so the remainder of the view is identically distributed. Hence the output of $\mathbf{G}_5^0$ and $\mathbf{G}_5^1$ has the same distribution, and the bound follows by summing the differences of $\mathbf{G}_0^\theta, \ldots, \mathbf{G}_5^\theta$ for $\theta = 0, 1$. The shares of the coalition in $\mathbf{G}_4$ and $\mathbf{G}_5$ include those of corrupted authorities that do not belong to T: static corruption fixes $\mathcal{C}$ before any share is created. □

Remarks. (1) Replay attacks, in which the adversary resubmits parts of an honest ballot under its own identity so that the honest vote is counted twice, are covered by the proof: labels bind the encrypted openings to the voter identity, so copied ciphertexts decrypt, under the new label, independently of the honest openings. An adversarial ballot that reuses honest commitments without valid openings for the honest authorities is removed by their complaints. (2) The proof does not require simulation-extractability of Π_{bal}: adversarial ballots are processed through the opening oracle of (E4) rather than through extraction. (3) Privacy against t or more corrupted authorities is impossible for any threshold scheme, since t authorities can reconstruct individual votes from their shares. (4) The restriction of OCheck and Check_k to valid entries of BB is necessary. Without it, the adversary could submit an object that keeps the identity and the ciphertexts e_k of an honest ballot but replaces its commitments. Since the label is unchanged, an honest authority would decrypt the honest seed s_k, find that $\mathsf{Rnd}_{\mathsf{id},k}(s_k)$ does not open the new $\mathbf{c}_k$, and reveal s_k in its complaint. Such an object cannot become a board entry, because each identity posts at most once and signatures are unforgeable. (5) Complaints reveal the decrypted seeds of ballots that are excluded. These ballots carry identities of no honest voter, so by (E4) the revealed values say nothing about honest ciphertexts, even copied ones. (6) The proof uses only that the view depends on the honest votes through P_{Hon}. It therefore applies unchanged if honest authorities outside T also publish their column sums, as happens when every authority announces its contribution in the first round of the two-round aggregation of Section 7: every such $\mathbf{V}_k$ is $P_{\mathsf{Hon}}(k)$ plus a function of the adversarial ballots. (7) Soundness of openings (E3) is not needed for privacy. Complaints by corrupted authorities against honest ballots are judged against the commitments and ciphertexts addressed to them, which depend only on the shares of the coalition, so the set Acc is a function of a view that does not depend on θ in $\mathbf{G}_5$.

6.2 Verifiability

We prove verifiability in two forms. Theorem 5 holds for every instantiation whose ballot proof is online extractable and isolates the argument on the tally. For the EVOLVE-style instantiation, which we implement and whose Fiat–Shamir-with-aborts proofs are not known to be online extractable, the verifiability result is Theorem 14 in Section 7; its proof extracts by nested forking and reuses the argument of Theorem 5 once the openings are extracted.

Lemma 4 (Interpolation)**.** *Let $(\mathbf{m}_{i,j})_{j=0}^n$, $i \in I$, satisfy $\mathbf{H}(\mathbf{m}_{i,0}, \ldots, \mathbf{m}_{i,n})^\top = \mathbf{0}$, and let $\mathbf{V}_k = \sum_{i\in I} \mathbf{m}_{i,k}$. Then for every $T \subseteq [n]$ with $|T| = t$, $\sum_{k\in T} \lambda_k^T \mathbf{V}_k = \sum_{i\in I} \mathbf{m}_{i,0}$.*

Proof. By the parity-check characterization, for each i there is P_i of degree less than t with $P_i(\alpha_j) = \mathbf{m}_{i,j}$ for $j = 0, \ldots, n$. Then $\sum_{k\in T} \lambda_k^T \mathbf{V}_k = \sum_{i\in I} \sum_{k\in T} \lambda_k^T P_i(k) = \sum_{i\in I} P_i(0) = \sum_{i\in I} \mathbf{m}_{i,0}$, since Lagrange interpolation over T is exact for polynomials of degree less than t and all differences of points are units. □

Theorem 5 (Verifiability with an online-extractable ballot proof)**.** *Assume the signature scheme is EUF-CMA secure, Π_{bal} is online extractable for $\mathcal{R}'_{\mathsf{bal}}$ (bound β'), and $(\mathsf{Agg}, \mathsf{VerAgg})$ is sound*

for openings of norm at most β'. Then T-EVOLVE is verifiable, even if all authorities are corrupted. More precisely, for every efficient adversary $\mathcal{A}$ there are efficient $\mathcal{B}_1, \mathcal{B}_2$ with

$$\mathrm{Adv}^{\mathrm{cons}}_{\mathcal{A}} \;\le\; \mathrm{Adv}^{\text{euf-cma}}_{\mathcal{B}_1} + \varepsilon_{\mathrm{ext}} + \mathrm{Adv}^{\mathrm{agg}}_{\mathcal{B}_2},$$

where $\mathrm{Adv}^{\mathrm{cons}}_{\mathcal{A}}$ is the probability that $\mathcal{A}$ wins the verifiability experiment of Section 4, $\mathrm{Adv}^{\mathrm{agg}}_{\mathcal{B}_2}$ is the probability that $\mathcal{B}_2$ breaks the soundness of $(\mathsf{Agg}, \mathsf{VerAgg})$, and $\varepsilon_{\mathrm{ext}}$ is the failure probability of the extractor over the at most polynomially many proofs on the board.

Proof. Consider an execution of the verifiability experiment in which $\mathcal{A}$ outputs a result $\mathbf{R}$, a quorum T and contributions $(\mathbf{V}_k, \Pi_k)_{k\in T}$ accepted by Verify.

Game 1: no forged ballots. Abort if some ballot on BB carries the identity of a voter created through OVote but was not produced by OVote. An abort yields a forgery, so the probability of this event is bounded by $\mathrm{Adv}^{\text{euf-cma}}_{\mathcal{B}_1}$. After this game, the honest accepted ballots Hon are exactly the accepted ballots with honest identities, and every ballot in Dis carries a label that no honest proof carries.

Game 2: extraction. Run the online extractor of Π_{bal} on the proofs of all ballots in Dis, and abort if it fails for any of them. The abort probability is at most $\varepsilon_{\mathrm{ext}}$. For ballots in Hon the challenger knows the witnesses, since it generated them. Hence, for every $i \in \mathsf{Acc}$ we have openings $(\mathbf{m}_{i,j}, \mathbf{r}_{i,j})_{j=0}^{n}$ of the ballot's commitments with $\|\mathbf{r}_{i,j}\|_2 \le \beta'$, such that $\mathbf{H}(\mathbf{m}_{i,0}, \ldots, \mathbf{m}_{i,n})^\top = \mathbf{0}$, $\mathbf{m}_{i,0} \in \mathcal{V}_{L,w}$, and $\mathbf{m}_{i,0} = \mathbf{v}_i$ for $i \in \mathsf{Hon}$.

Game 3: correct column sums. Abort if for some $k \in T$ we have $\mathbf{V}_k \neq \sum_{i\in\mathsf{Acc}} \mathbf{m}_{i,k}$. Since Verify checks $\mathsf{VerAgg}((\mathbf{c}_{i,k})_{i\in\mathsf{Acc}}, \mathbf{V}_k, \Pi_k) = 1$, an abort directly gives $\mathcal{B}_2$ an accepted aggregation together with openings of norm at most β' whose messages do not sum to $\mathbf{V}_k$; its probability is bounded by $\mathrm{Adv}^{\mathrm{agg}}_{\mathcal{B}_2}$.

In Game 3, Lemma 4 and the check $\mathbf{R} = \sum_{k\in T} \lambda_k^T \mathbf{V}_k$ give $\mathbf{R} = \sum_{i\in\mathsf{Acc}} \mathbf{m}_{i,0}$ over R_q. All summands are binary vectors and $|\mathsf{Acc}| < q$, so the equality holds over the integers, and

$$\mathbf{R} \;=\; \sum_{i\in\mathsf{Hon}} \mathbf{v}_i + \sum_{i\in\mathsf{Dis}} \mathbf{m}_{i,0} \qquad \text{with } \mathbf{m}_{i,0} \in \mathcal{V}_{L,w}.$$

The votes $\mathbf{u}_i = \mathbf{m}_{i,0}$ witness that the adversary does not win Game 3, and the bound follows. □

The instantiation with a general-purpose proof system falls under Theorem 5 if that proof system is online extractable in the random oracle model, which has to be established for the chosen system.

Theorem 6 (Individual verifiability)**.** *Assume the signature scheme is EUF-CMA secure, Π_{bal} is complete, and EO satisfies (E3). Then no efficient adversary controlling all authorities can cause an honest ballot to be excluded from $\mathsf{Acc}(\mathsf{BB})$, except with negligible probability.*

Proof. An honest ballot b is valid: its signature and proof verify by completeness, and by unforgeability no other ballot occupies its identity. It is therefore excluded only if some complaint $\gamma = (k, \omega)$ against it satisfies $\mathsf{VerComplaint} = 1$; complaints of disqualified authorities are ignored, so $\mathsf{VerKey}(\mathsf{pk}_k) = 1$ and $x' = \mathsf{Judge}(\mathsf{pk}_k, \mathsf{id}, e_k, \omega) \neq \mathsf{fail}$ and either $x' = \perp$ or $\mathsf{Rnd}_{\mathsf{id},k}(x')$ is not a valid opening of $\mathbf{c}_k$. The honest voter encrypted s_k under pk_k only after checking $\mathsf{VerKey}(\mathsf{pk}_k) = 1$, and $\mathsf{Rnd}_{\mathsf{id},k}(s_k) = \mathbf{r}_k$ is a valid opening of $\mathbf{c}_k$ (its norm is at most β by construction). Hence $x' \notin \{s_k, \mathsf{fail}\}$. A reduction that guesses the honest ballot and the index k, embeds its challenge ciphertext there and outputs ω breaks (E3), with a loss of a factor nq_V. □

Theorems 5 and 6 together state that a voter who finds her ballot on the board knows that it is accepted, and that every result that verifies counts it, whatever the authorities do.

6.3 Robustness

Robustness needs, besides (E2) and (E3), that the shares of every accepted ballot are consistent. We say that the ballot proof is *share-sound* if no efficient adversary can output a ballot $b = (\mathsf{id}, (\mathbf{c}_j)_{j=0}^{n}, (e_k)_k, \pi)$ whose proof π verifies for the label id, a set $T \subseteq [n]$ with $|T| = t$ and valid openings $(\mathbf{m}_k, \mathbf{r}_k)$ of $\mathbf{c}_k$ under β for $k \in T$ such that $\sum_{k\in T} \lambda_k^T \mathbf{m}_k \notin \{0,1\}^L$, except with negligible probability. In the proof below, a reduction that runs the honest authorities obtains such openings by decryption. For an online-extractable Π_{bal} this follows from the extraction of Theorem 5 and binding; for the EVOLVE-style instantiation it follows from one fork on H_{bal} (Section 7, after Theorem 14).

Theorem 7 (Robustness)**.** *Assume the signature scheme is EUF-CMA secure,* EO *satisfies (E1)–(E3),* Π_{bal} *is complete and share-sound, and* $(\mathsf{Agg}, \mathsf{VerAgg})$ *is complete. Let* p_{rd} *be a lower bound on the probability that a tally round in which only the honest authorities follow the protocol yields* t *accepted contributions of honest authorities, whatever the adversary does (*$p_{\mathrm{rd}} = 1$ *for a one-round* $\mathsf{Contrib}_k$*; for the two-round aggregation of Section 7,* $p_{\mathrm{rd}} \geq s(\kappa)^t - \delta_{\mathrm{rd}}$ *with* $\delta_{\mathrm{rd}} = t\kappa(2e^{-r^2/2}/M' + (l(E+1)+1)e^{-\tau} + \varepsilon_{\mathrm{fresh}})$ *from Section 7.3 and Lemma 13, with* G *modeled as a random oracle). Then T-EVOLVE is robust with round success probability* p_{rd} *against adversaries statically corrupting at most* $n - t$ *authorities.*

Proof. Let $\mathcal{H} = [n] \setminus \mathcal{C}$, so $|\mathcal{H}| \geq t$. By (E1), honest keys pass VerKey, so only corrupted authorities can be disqualified. By Theorem 6, which uses (E3), honest ballots are accepted. Consider any $b_i \in \mathsf{Acc}$ and any $k \in \mathcal{H}$, and let $x = \mathsf{Dec}(\mathsf{sk}_k, \mathsf{id}_i, e_{i,k})$. During the checking phase, the honest authority k runs Check_k on every valid ballot of the voting phase and posts its complaints before the phase closes. If $x = \perp$, or if $\mathsf{Rnd}_{\mathsf{id}_i,k}(x)$ is not a valid opening of $\mathbf{c}_{i,k}$, it has therefore posted $\gamma = (k, \mathsf{Open}(\mathsf{sk}_k, \mathsf{id}_i, e_{i,k}))$. By (E2), Judge returns x, so $\mathsf{VerComplaint}$ accepts and b_i is not accepted, a contradiction. Hence every honest authority holds valid openings $(\mathbf{m}_{i,k}, \mathbf{r}_{i,k})$ of its share of every accepted ballot. For a one-round $\mathsf{Contrib}_k$, completeness of Agg gives contributions of all honest authorities that VerAgg accepts. For the two-round aggregation of Section 7, a round succeeds whenever at least t honest authorities have a non-aborting attempt, since their contributions alone complete it; corrupted authorities can make honest authorities publish responses in a round that fails anyway, but cannot make a round fail that would otherwise succeed. The masks of the honest attempts are hidden by G until they are opened, so each honest attempt passes both rejection steps with probability $1/M'^2$, up to the statistical error of the rejection steps and the failure term $(l(E+1)+1)e^{-\tau} + \varepsilon_{\mathrm{fresh}}$ of Lemma 13, whatever challenges the adversary induces through BB_1. An honest authority announces only attempts that pass all checks of VerAgg, so it is never recorded as faulty; hence each round succeeds with probability at least $s(\kappa)^t - \delta_{\mathrm{rd}} \geq p_{\mathrm{rd}}$, independently of earlier rounds, and the honest authorities fail in all of N_{rd} rounds with probability at most $(1 - p_{\mathrm{rd}})^{N_{\mathrm{rd}}}$. In a successful round, let $T \subseteq \mathcal{H}$ with $|T| = t$ be a set of honest authorities whose contributions are accepted, and $\mathbf{R} = \sum_{k\in T} \lambda_k^T \mathbf{V}_k = \sum_{i\in\mathsf{Acc}} \sum_{k\in T} \lambda_k^T \mathbf{m}_{i,k}$. By share-soundness, applied by a reduction that runs the honest authorities, each inner sum lies in $\{0,1\}^L$ except with negligible probability, and $|\mathsf{Acc}| < q$, so each entry of $\mathbf{R}$ is a constant polynomial with value in $[0, |\mathsf{Acc}|]$ and Verify accepts. Corrupted authorities cannot prevent this: they do not take part in T, and their complaints against honest ballots are rejected by Theorem 6. □

7 Instantiations

This section instantiates the building blocks of Section 3 and establishes the properties that Section 6 assumes. Section 7.1 discusses the encryption of the seeds. Section 7.2 gives the ballot proof, with a general-purpose proof system or with EVOLVE-style proofs and their rejection bounds. Section 7.3 gives the two-round aggregation and the bounds on its shifts. Section 7.4

proves verifiability and share-soundness of the EVOLVE-style instantiation, and Section 7.5 derives its parameters.

7.1 Encryption of the seeds

Two constructions of EO are used in the voting literature.

Release of identity-based keys. As in Epoque (Boyen et al., 2021) and in the PVSS of Kate et al. (2026), each authority runs an identity-based encryption scheme, and the label lb determines the identity $(\mathsf{par}, \mathsf{lb})$. Dec decrypts with the user key of this identity, extracted deterministically from the master key; Open outputs this user key; and Judge checks that it is a valid key for the identity and decrypts with it. For GPV-type schemes (Gentry et al., 2008), a valid key is a short preimage of the hash of the identity, with a norm bound fixed in par. (E2) holds by construction, also for malformed ciphertexts: whatever Dec returns, $\perp$ included, anyone recomputes it from the released key, so no proof of decryption is needed. (E4) follows from adaptive IND-ID-CPA security, since openings are key-extraction queries for identities other than the challenge ones. Chosen-ciphertext security and a proof of key validity are not needed, and VerKey only checks the format of the master public key. (E3) is the delicate property. The only sufficient condition we know how to establish is the following: except with negligible probability over the encryption randomness, decryption returns the plaintext for every master public key that passes VerKey and every user key that passes the check of Judge. For GPV-type schemes, the decryption error is a short noise term minus the inner product of the user key with the noise vector of the ciphertext. Independently of the master public key, its absolute value is at most $B_1 + B_K B_0$, where B_1 bounds the short term, B_K the norm of valid keys and B_0 the norm of the noise vector, so the condition holds if the modulus exceeds $4(B_1 + B_K B_0) + 2$. Proposition 15 in Appendix D makes this precise: with the identity-based encryption of Gentry et al. (2008) and deterministic extraction of user keys, the key-release construction satisfies (E1)–(E4) in the random oracle model under LWE, for a modulus that is polynomial in the dimension. Published parameter sets are designed for an average-case failure rate, which does not suffice for this argument: an authority that knows the trapdoor can recover the noise of an honest ciphertext and may be able to find, among the many valid keys of the identity, one that makes decryption fail. For instance, Jia et al. (2024, full version, Table 3) report a failure rate of $2^{-17.7}$ per bit for their set at the 128-bit level and remove the errors with an error-correcting code. Meeting the worst-case condition requires a modulus larger than the product of the norm bounds of the keys and of the noise, and hence a larger ciphertext. Appendix D fixes such parameters for the NTRU instantiation of the GPV scheme (Ducas et al., 2014): ring degree 2048, a 34-bit modulus and ternary encryption randomness, under which every honest ciphertext decrypts correctly under every valid key, for every choice of the encryption randomness. A ciphertext of a 256-bit seed then takes 9.6 KB, so the n seed ciphertexts take 38.3 KB of a ballot for $n = 4$; for comparison, the average-case parameters of (Jia et al., 2024) give 8.5 KB per ciphertext at ring degree 1024 with an error-correcting code, and the same degree does not reach the target under the worst-case condition (Appendix D).

Proofs of correct decryption. PQKryvos (Huber et al., 2026) encrypts with ML-KEM and, in a complaint, publishes the plaintext with a zero-knowledge proof of correct decryption (Lyubashevsky et al., 2021). In terms of EO, VerKey verifies a proof of knowledge of a well-formed secret key, Open outputs $\mathsf{Dec}(\mathsf{sk}, \mathsf{lb}, e)$ with a proof of this claim, and Judge returns the claimed value if the proof verifies and fail otherwise. (E3) follows from three properties: knowledge soundness of the key proof; correctness for every key in the relation extracted by it, with probability over the encryption randomness only; and soundness of the decryption proof. This correctness requirement is weaker than the worst-case condition above, because the key is fixed before the honest ciphertext is created. It is nevertheless not the average over honest keys for which the failure rate of ML-KEM is stated. (E4) follows from IND-CCA security with labels and zero knowledge of both proofs. (E2), however, also requires proofs for the claim that e decrypts to $\perp$.

When decryption succeeds, the key holder can reveal the encapsulated message with a proof of the decryption equation of the underlying CPA scheme (Lyubashevsky et al., 2021), and anyone can re-encrypt and check the rest. For this form of Open, (E4) needs, besides IND-CCA security, that the encapsulated message of a ciphertext that decrypts successfully can be extracted from the adversary's random-oracle queries. This holds for the Fujisaki–Okamoto transform in the random oracle model. This approach does not work for $\perp$: a ciphertext of an honest voter replayed under another label decrypts to $\perp$, and revealing its encapsulated message would reveal the honest seed. A proof of failure must therefore cover the hash evaluations and the re-encryption check of the Fujisaki–Okamoto transform. We are not aware of an efficient proof of this kind; PQKryvos does not discuss this case explicitly.

Theorems 3, 6 and 7 hold for every EO with the properties they list. Proposition 15 shows that the class of such schemes is not empty, and Appendix D gives concrete parameters for the key-release construction; the implementation uses a placeholder for the encryption (Section 8), so the ciphertexts are counted from their parameters and not measured.

7.2 The ballot proof

Exact proofs. Π_{bal} can be instantiated with a general-purpose proof system over the BDLOP message space, as done for single commitments in PQKryvos (Huber et al., 2026). The statement contains $n+1$ commitment circuits, $n+1-t$ linear equations with public coefficients, L binarity constraints and one weight constraint. Extraction is exact, so no invertibility condition on challenge differences is needed, and Theorem 5 applies if the proof system is online extractable in the random oracle model. We do not give concrete parameters or costs for this instantiation, since they are those of the chosen proof system; PQKryvos reports costs for statements about a single commitment (Huber et al., 2026).

EVOLVE-style proofs. Alternatively, Π_{bal} can be assembled from the following Fiat–Shamir-with-aborts proofs: proofs of knowledge of short openings of $\mathbf{c}_0, \dots, \mathbf{c}_n$ with a common challenge; for each row of $\mathbf{H}$, one additional first-message element that proves the parity-check equation; the $\{0,1\}$ OR-proof of EVOLVE (del Pino et al., 2017, Sec. 3.2) applied to each entry of $\mathbf{c}_0$, as in its extension to L candidates (del Pino et al., 2017, App. B); and a linear proof of the weight. The coefficients of a parity-check equation multiply only public values and do not affect the norms of the responses. The verifier recomputes the additional first-message elements from the responses, so they are not transmitted. All components use challenges derived from a single random-oracle query

$$H_{\mathsf{bal}}(\mathsf{par}, \mathsf{id}, (\mathbf{c}_j)_{j=0}^{n}, (e_k)_{k=1}^{n}, \mathbf{w}),$$

where $\mathbf{w}$ collects the first messages of all components. The common challenge c is taken from this output, and the challenges of the OR-proofs are expanded from the same output. Since the ballot proof is produced before the board is complete, its hash does not include the board.

The ballot proof in detail. Write $\mathbf{c}_j = (\mathbf{c}_{j,1}, \mathbf{c}_{j,2})$ with $\mathbf{c}_{j,1} = \mathbf{A}\mathbf{r}_j$ and $\mathbf{c}_{j,2} = \mathbf{B}\mathbf{r}_j + \mathbf{m}_j$, so that the message of a commitment is determined by its randomness, $\mathbf{m}_j = \mathbf{c}_{j,2} - \mathbf{B}\mathbf{r}_j$. The proof is a proof of knowledge of short $\mathbf{r}_0, \dots, \mathbf{r}_n$ with $\mathbf{A}\mathbf{r}_j = \mathbf{c}_{j,1}$, together with the linear relations that these randomnesses satisfy.

Figure 3 gives an honest run. The responses that depend on the witness are the $\mathbf{z}_j$ and, for each candidate, the response of the real branch of the OR-proof. The prover applies a single rejection step to all of them together. It is essential that $f_{a,1}$ is derived from $f_{a,0}$ by the hashed permutation (del Pino et al., 2017, VerifyOR): if both challenges could be chosen freely, a prover could simulate both branches of an OR-proof and prove any message.

Completeness. For a ballot formed as in Figure 2, the messages are the shares of a vote $\mathbf{v} \in \mathcal{V}_{L,w}$, so $\sum_j H_{\ell,j}\mathbf{m}_j = \mathbf{0}$ for every row of $\mathbf{H}$ and, if $w \neq *$, $\sum_a m_{0,a} = w$. Substituting

Prover (openings $(\mathbf{m}_j, \mathbf{r}_j)_{j=0}^n$ with $\mathbf{m}_0 \in \mathcal{V}_{L,w}$):

1. Sample $\mathbf{y}_j \leftarrow D_{\sigma_J}^{N(2d+L)}$ for $j = 0, \ldots, n$; compute $\mathbf{w}_j = \mathbf{A}\mathbf{y}_j$, $\mathbf{u}_\ell^{\mathsf{pc}} = \sum_j H_{\ell,j}\mathbf{B}\mathbf{y}_j$ for every row ℓ of $\mathbf{H}$, and, if $w \neq *$, $u^{\mathsf{wt}} = \sum_{a\in[L]}(\mathbf{B}\mathbf{y}_0)_a$; for each candidate a, with real branch $m = m_{0,a}$, sample a mask $\boldsymbol{\rho}_a \leftarrow D_{\sigma_J}^{N(2d+L)}$ for the real branch and $f_{a,1-m} \leftarrow \mathcal{S}$, $\mathbf{r}_{a,1-m} \leftarrow D_{\sigma_J}^{N(2d+L)}$ for the simulated branch, and compute the first messages of both branches of the OR-proof of (del Pino et al., 2017, Sec. 3.2), the simulated one from $(f_{a,1-m}, \mathbf{r}_{a,1-m})$; let $\mathbf{w}^{\mathsf{OR}}$ collect them.
2. $h = H_{\mathsf{bal}}\big(\mathsf{par}, \mathsf{id}, (\mathbf{c}_j)_j, (e_k)_k, (\mathbf{w}_j)_j, (\mathbf{u}_\ell^{\mathsf{pc}})_\ell, u^{\mathsf{wt}}, \mathbf{w}^{\mathsf{OR}}\big)$; expand the challenge $c \in \mathcal{S}$ and the permutations π_a^{OR} from h.
3. Respond $\mathbf{z}_j = \mathbf{y}_j + c\,\mathbf{r}_j$; for each a, set $f_{a,m}$ from $f_{a,1-m}$ so that $f_{a,1} = \pi_a^{\mathsf{OR}}(f_{a,0})$, and $\mathbf{r}_{a,m} = \boldsymbol{\rho}_a + f_{a,m}\mathbf{r}_0$.
4. Apply *one* rejection step to the concatenation of $(\mathbf{z}_j)_j$ and $(\mathbf{r}_{a,m})_a$, with shift $\mathbf{s} = (c\mathbf{r}_0, \ldots, c\mathbf{r}_n, f_{1,m}\mathbf{r}_0, \ldots, f_{L,m}\mathbf{r}_0)$; if it fails, restart at step 1.
5. Output $\pi = \big(h, (\mathbf{z}_j)_j, (\mathbf{r}_{a,0}, \mathbf{r}_{a,1}, f_{a,0})_a\big)$.

Verifier: expand c and π^{OR} from h; check $f_{a,0} \in \mathcal{S}$ and set $f_{a,1} = \pi_a^{\mathsf{OR}}(f_{a,0})$ for every $a \in [L]$; recompute $\mathbf{w}_j = \mathbf{A}\mathbf{z}_j - c\,\mathbf{c}_{j,1}$, $\mathbf{u}_\ell^{\mathsf{pc}} = \sum_j H_{\ell,j}(\mathbf{B}\mathbf{z}_j - c\,\mathbf{c}_{j,2})$, if $w \neq *$ also $u^{\mathsf{wt}} = \sum_a(\mathbf{B}\mathbf{z}_0 - c\,\mathbf{c}_{0,2})_a + c\,w$, and the first messages of the OR-proofs from their responses and $(f_{a,0}, f_{a,1})$; check that the concatenation of all responses $\big((\mathbf{z}_j)_j, (\mathbf{r}_{a,0}, \mathbf{r}_{a,1})_a\big)$ has ℓ_2-norm at most B_J; accept iff H_{bal} applied to the recomputed input returns h.

Figure 3: The EVOLVE-style ballot proof, for commitments $\mathbf{c}_j = (\mathbf{c}_{j,1}, \mathbf{c}_{j,2}) = (\mathbf{A}\mathbf{r}_j, \mathbf{B}\mathbf{r}_j + \mathbf{m}_j)$. The first messages are not transmitted.

$\mathbf{z}_j = \mathbf{y}_j + c\mathbf{r}_j$ gives $\mathbf{A}\mathbf{z}_j - c\mathbf{c}_{j,1} = \mathbf{A}\mathbf{y}_j$ and

$$\sum_j H_{\ell,j}(\mathbf{B}\mathbf{z}_j - c\mathbf{c}_{j,2}) = \sum_j H_{\ell,j}\mathbf{B}\mathbf{y}_j - c\sum_j H_{\ell,j}\mathbf{m}_j = \mathbf{u}_\ell^{\mathsf{pc}},$$

and likewise $\sum_a(\mathbf{B}\mathbf{z}_0 - c\mathbf{c}_{0,2})_a + cw = u^{\mathsf{wt}}$. Hence the verifier recomputes exactly the hashed values, for every admissible vote. If $w = *$, as in a yes/no election, the component u^{wt} is neither computed nor checked; otherwise completeness would fail for votes of another weight.

Rejection sampling. Let $M = \exp(r/\alpha + 1/(2\alpha^2))$ with $r = 309/20$ and $\alpha = 141/10$, so that $\ln M = 43669/39762$ is rational and $M \approx 3.00$. The prover rejects once, with the shift $\mathbf{s} = (c\mathbf{r}_0, \ldots, c\mathbf{r}_n, f_{1,m}\mathbf{r}_0, \ldots, f_{L,m}\mathbf{r}_0)$. Its norm depends on the challenges, and we bound it for every choice of them.

Lemma 8 (Norm of the shift)**.** *Let the coefficients of $\mathbf{r}_0, \ldots, \mathbf{r}_n$ be independent with distribution D_σ, let $c, f_{1,m}, \ldots, f_{L,m} \in \mathcal{S}$ be fixed, and let $\tau > 0$. Then $\|\mathbf{s}\|_2 \leq \Theta$ except with probability $e^{-\tau}$, where*

$$\Theta^2 = \sigma^2(t_s + 2\sqrt{\xi_s t_s \tau} + 2\xi_s\tau), \qquad t_s = 60N(2d+L)(n+1+L), \qquad \xi_s = 3600(1+L).$$

The proof, an application of the tail bound for quadratic forms of subgaussian vectors (Hsu et al., 2012), is in Appendix B.

The lemma applies to the real shift. The challenges are outputs of H_{bal} on an input that contains a fresh first message; except with probability at most $q_{\mathsf{bal}}\sigma_J^{-dN} < 2^{-26000}$ per attempt, where $q_{\mathsf{bal}} \leq 2^{64}$ counts the queries to H_{bal} of all parties, they are uniform and independent of $\mathbf{r}_0, \ldots, \mathbf{r}_n$ (Appendix B). We set $\sigma_J = \alpha\Theta$ with $\tau = 171\ln 2$, i.e. a failure probability of 2^{-171} per attempt. The inner product $\langle \mathbf{z}, \mathbf{s}\rangle$ for $\mathbf{z}$ from D_{σ_J} is $\sigma_J\|\mathbf{s}\|_2$-subgaussian (Micciancio & Peikert, 2012, Lemma 2.8), so, conditioned on $\|\mathbf{s}\|_2 \leq \Theta$, the ratio of the densities in the rejection step exceeds M with probability at most $e^{-r^2/2} < 2^{-172}$, and the output of one attempt, a response or $\perp$, is within statistical distance $e^{-r^2/2}/M$ of that of an ideal attempt that, independently of the witness, outputs a sample of D_{σ_J} over the whole concatenation with probability $1/M$ and $\perp$

otherwise; this is (Lyubashevsky, 2012, Lemma 4.5 and Thm. 4.6) with the constant 12 replaced by r. The simulator of the privacy proof samples the responses from D_{σ_J} and programs H_{bal}; its error is at most $e^{-r^2/2}/M + 2^{-171} + 3q_{\mathsf{bal}}\sigma_J^{-dN}$ per attempt of every honest ballot, plus 2^{-1300} once (Appendix B), and this is the ballot-proof part of $\varepsilon_{\mathrm{zk}}$ in Theorem 3; Section 8 and Appendix C sum all such errors over the experiment. The earlier choice $\sigma_J = \alpha\sqrt{t_s}\,\sigma$, which uses the typical norm of the shift, is smaller by a factor of 1.9 to 2.1, but it is heuristic. The rigorous choice costs about one bit per coefficient of the responses, i.e. about 3.2 KB, or 3% of the ballot and 6% of the proof for $n = 4$, $L = 1$. Rejecting each component separately, as in EVOLVE, would require all $L + 1$ steps to succeed together, about M^{L+1} attempts in expectation, whereas the single step needs about M attempts for every L. The price is a σ_J that grows like $\sqrt{n + 1 + L}$: $2^{15.0}$ for $L = 1$ and $2^{16.1}$ for $L = 10$ at $d = 7$, $n = 4$. The bound $B_J = 2\sigma_J\sqrt{N(2d + L)(n + 1 + 2L)}$ holds for the concatenation of all responses except with probability $2^{-N(2d+L)(n+1+2L)/4}$ (Baum & Lyubashevsky, 2017, (18)).

Experiment. With the implementation of Section 8, we generated 1000 independent ballots for each of the settings $(n, L) \in \{(4, 1), (4, 2), (4, 5), (4, 10), (5, 1)\}$ with $d = 7$ and $q = 2^{42} - 143$ (for $L = 10$ smaller than the modulus that Section 7.5 requires; the rejection statistics do not depend on q). Every ballot used fresh voter randomness, fresh challenges in every attempt, σ_J fixed in advance by the formula above, and the samplers of Section 8, whose statistical distance from the ideal distributions is bounded. We encoded, decoded and verified every ballot, and repeated the experiment with two other seeds and 500 ballots per setting. The mean number of attempts per setting and run lay between 2.79 and 3.11, with an overall mean of 2.96 ± 0.05 over 10,000 ballots ($M = 3.00$), and showed no dependence on L. All ballots verified. The ratio $\|\mathbf{s}\|_2/\Theta$ never exceeded 0.55, and the norm of all responses never exceeded $0.51\,B_J$. We also ran two kinds of statistical tests: a chi-square test of the accepted responses against D_{σ_J}, and a two-sample test between the response of an OR-branch when it is real (after rejection) and when it is simulated (sampled directly). Over the 15 tests of each kind, Fisher's combined p-values were 0.99 and 0.09, respectively; the smallest single p-value, 0.009, is above the Bonferroni threshold 0.0033. Such tests cannot establish zero knowledge, which rests on Lemma 8; they check that the implementation behaves as the lemma predicts.

Bounds: responses versus extraction. B_J bounds the responses of one accepting transcript. Two accepting transcripts with the same first messages and $c \neq c'$ give $\bar{\mathbf{r}}_j = \mathbf{z}_j - \mathbf{z}'_j$, and two OR-transcripts that differ in $(f_{a,0}, f_{a,1})$ give a difference of responses. Both differences have norm at most $\bar{B}_V = \bar{B}_{\mathrm{OR}} := 2B_J$. These extraction bounds, not B_J, enter β_{SIS} (del Pino et al., 2017, Thm. 3.2).

Communication. The proof consists of the 2λ-bit output h of H_{bal}, from which the challenges are expanded, of $(n + 1)(2d + L)$ ring elements with coefficients of size about σ_J, and, per candidate, of $2(2d + L)$ ring elements of the same size and the element $f_{a,0} \in \mathcal{S}$. The parity checks and the weight constraint are recomputed by the verifier and are not transmitted.

Newer proof systems. Lattice-based zero-knowledge proofs have become much shorter since EVOLVE, for instance with the framework of Lyubashevsky et al. (2022) and with LaBRADOR (Beullens & Seiler, 2023). We use EVOLVE-style proofs so that the ballot can be compared with EVOLVE's on the same code and so that the aggregation can use the amortized proofs of Baum & Lyubashevsky (2017). A newer proof system could shorten the proof part of a ballot, 56 KB of the 108.9 KB of a measured yes/no ballot for $n = 4$, but not the $n + 1$ commitments, about 52 KB, unless the shares are committed together, which would change the aggregation. We have not worked out such an instantiation; Section 9 estimates what packing the candidates into one ring element would save in the commitments.

Relaxed openings. Extraction from Fiat–Shamir-with-aborts proofs yields *relaxed openings*: a triple $(\mathbf{m}, \bar{\mathbf{r}}, f)$ is a relaxed opening of $\mathbf{c} = \binom{\mathbf{c}_1}{\mathbf{c}_2}$ with bound B if $f \in R_q$ is invertible, $\|\bar{\mathbf{r}}\|_2 \leq B$ and $f\mathbf{c} = \mathbf{C}\bar{\mathbf{r}} + f\binom{\mathbf{0}}{\mathbf{m}}$. We write $\eta(f)$ for the operator norm of multiplication by f with respect to $\|\cdot\|_2$; for instance $\eta(f) \leq \|f\|_1$, and $\eta(2) = 2$.

Accepted ballots N_V	10^4	10^5	10^6	10^6	
Blocks E per authority	24	217	2142	2142	$l = 30$, $\psi = 500$, $\ell = 517$
Ring degree N, σ		256, 1			
Modulus q	$2^{42} - 143$	$2^{43} - 175$	$2^{45} - 591$	$2^{45} - 591$	prime, $q \equiv 17 \pmod{32}$
Module rank d	7	7	7	8	width $2d + L$
Authorities n, threshold t		4, 3 and 5, 3			
σ_J, B_J, $\bar{B}_V = 2B_J$ $(n = 4)$	$2^{15.0}$, $2^{23.3}$, $2^{24.3}$			$2^{15.0}$, $2^{23.5}$, $2^{24.5}$	Lemma 8
σ_1	$2^{23.5}$	$2^{25.1}$	$2^{26.8}$	$2^{26.8}$	Lemma 13
σ_2	$2^{23.5}$	$2^{24.9}$	$2^{26.4}$	$2^{26.5}$	Lemma 13
$B_A = 36\sigma_2\sqrt{N(2d+L)}$	$2^{34.7}$	$2^{36.0}$	$2^{37.5}$	$2^{37.7}$	slack 2
β_{SIS}	$2^{41.6}$	$2^{42.9}$	$2^{44.5}$	$2^{44.6}$	Theorem 14
dominant term	$2\bar{B}_V + \eta_{\max}B_A$	$2\bar{B}_V + \eta_{\max}B_A$	β_{BL}	$2\bar{B}_V + \eta_{\max}B_A$	$\beta_{\mathrm{BL}} = 2^{41.2}$, $2^{42.8}$, $2^{44.5}$, $2^{44.6}$
M-LWE (hiding), MATZOV	$2^{138.4}$	$2^{135.0}$	$2^{128.9}$	$2^{148.5}$	classical
MATZOV, quantum	$2^{133.3}$	$2^{130.3}$	$2^{124.7}$	$2^{142.5}$	quantum
core-SVP, classical	$2^{110.7}$	$2^{107.2}$	$2^{101.0}$	$2^{121.1}$	$2^{0.292\beta}$
core-SVP, quantum	$2^{100.7}$	$2^{97.5}$	$2^{91.7}$	$2^{110.2}$	$2^{0.265\beta}$
M-SIS (binding, ℓ_2), MATZOV	$2^{144.5}$	$2^{138.2}$	$2^{134.6}$	$2^{155.0}$	classical
MATZOV, quantum	$2^{138.9}$	$2^{133.2}$	$2^{129.9}$	$2^{148.4}$	
core-SVP, classical	$2^{116.8}$	$2^{110.1}$	$2^{106.3}$	$2^{127.6}$	
core-SVP, quantum	$2^{106.0}$	$2^{99.9}$	$2^{96.5}$	$2^{115.8}$	
Commitment	10.5 KB	10.8 KB	11.3 KB	12.7 KB	
Ballot, $n = 4$	108.9 KB	110.1 KB	112.6 KB	128.1 KB	ciphertexts as 32-byte payloads
with seed ciphertexts	147.0 KB	148.2 KB	150.7 KB	166.3 KB	4×9792 bytes, Appendix D
Ballot, $n = 5$	128.4 KB	129.9 KB	132.9 KB	150.2 KB	
Ballot, $n = 4$, $L = 2$, $w = 1$	138.0 KB	139.4 KB	142.3 KB	159.2 KB	
Authority, per ballot	28.2 KB	28.3 KB	29.7 KB	33.7 KB	both rounds

Table 3: Parameters of the EVOLVE-style instantiation for $N_V = 10^4$, 10^5 and 10^6 accepted ballots, $n = 4$ and $t = 3$ unless stated otherwise; the last column is the set with $d = 8$ for $N_V = 10^6$ (Section 7.5). The row σ_J, B_J, $\bar{B}_V$ gives the values for $d = 7$ and, in the last column, for $d = 8$. All rows except the ballot row marked $L = 2$ refer to a yes/no election ($L = 1$, no weight constraint). Ballot sizes count the n seed ciphertexts as 32-byte payloads, except the row "with seed ciphertexts", which uses the 9792-byte ciphertexts of the instantiation of Appendix D; signatures are not included. All entries are computed, not measured. The M-LWE and M-SIS rows are hardness estimates for the primitives under four cost models of lattice reduction (Section 7.5; the quantum MATZOV model costs the nearest-neighbour search in the depth-times-width metric), given to one decimal; they are not lower bounds and not security guarantees for the protocol. The choice of the parameters and the size model are described in the text; a KB is 1024 bytes.

Lemma 9 (Uniqueness of messages)**.** *Let* $(\mathbf{m}, \bar{\mathbf{r}}, f)$ *and* $(\mathbf{m}', \bar{\mathbf{r}}', f')$ *be relaxed openings of the same commitment with bounds* B *and* B'*. If* $\mathbf{m} \neq \mathbf{m}'$*, then* $\mathbf{z} = f'\bar{\mathbf{r}} - f\bar{\mathbf{r}}'$ *is a nonzero vector with* $\mathbf{Az} = \mathbf{0}$ *and* $\|\mathbf{z}\|_2 \leq \eta(f')B + \eta(f)B'$*.*

Proof. Multiplying the first equation by f' and the second by f and subtracting gives $\mathbf{Az} = f'f\mathbf{c}_1 - ff'\mathbf{c}_1 = \mathbf{0}$ and $\mathbf{Bz} = ff'(\mathbf{m}' - \mathbf{m})$. The latter is nonzero because ff' is invertible, so $\mathbf{z} \neq \mathbf{0}$. The norm bound follows from the triangle inequality. This is the argument of (Baum et al., 2018, Lemma 7). □

Lemma 10 (Linear relations and weight)**.** *Suppose that, for commitments* $\mathbf{c}_0, \ldots, \mathbf{c}_n$ *and a public matrix* $\mathbf{H}$*, an extractor obtains relaxed openings* $(\mathbf{m}_j, \bar{\mathbf{r}}_j, f)$ *with a common slack* f *from two accepting transcripts of the parity-check proof, i.e.* $\bar{\mathbf{r}}_j = \mathbf{z}_j - \mathbf{z}'_j$*,* $f = c - c'$*, and the verification equations* $\sum_j H_{\ell,j}\mathbf{Bz}_j = \mathbf{u}_\ell^{\mathsf{pc}} + c\sum_j H_{\ell,j}\mathbf{c}_{j,2}$ *hold for both transcripts with the same* $\mathbf{u}_\ell^{\mathsf{pc}}$*. Here the relaxed openings come from the equations* $\mathbf{A}\bar{\mathbf{r}}_j = f\mathbf{c}_{j,1}$*, and* $\mathbf{m}_j := \mathbf{c}_{j,2} - f^{-1}\mathbf{B}\bar{\mathbf{r}}_j$*. Then* $\mathbf{H}(\mathbf{m}_0, \ldots, \mathbf{m}_n)^\top = \mathbf{0}$*. If moreover* $w \neq *$ *and the weight equation* $\sum_{a=1}^{L}(\mathbf{Bz}_0 - c\mathbf{c}_{0,2})_a + cw = u^{\mathsf{wt}}$ *holds for both transcripts with the same* u^{wt}*, then* $\sum_{a=1}^{L} m_{0,a} = w$*.*

Proof. Subtracting the two verification equations gives $\sum_j H_{\ell,j}\mathbf{B}\bar{\mathbf{r}}_j = f\sum_j H_{\ell,j}\mathbf{c}_{j,2}$. Substituting $\mathbf{B}\bar{\mathbf{r}}_j = f\mathbf{c}_{j,2} - f\mathbf{m}_j$ from the relaxed openings yields $f\sum_j H_{\ell,j}\mathbf{m}_j = \mathbf{0}$, and f is invertible. For the weight, subtracting the two weight equations gives $\sum_a(\mathbf{B}\bar{\mathbf{r}}_0)_a - f\sum_a(\mathbf{c}_{0,2})_a + fw = 0$. Substituting $\mathbf{B}\bar{\mathbf{r}}_0 = f\mathbf{c}_{0,2} - f\mathbf{m}_0$ yields $f\big(w - \sum_a m_{0,a}\big) = 0$, and f is invertible. □

In the EVOLVE-style instantiation, the ballot proof yields, for a ballot i, relaxed openings of $\mathbf{c}_{i,0}, \ldots, \mathbf{c}_{i,n}$ with a common slack f_i and bound $\bar{B}_V$ that satisfy the parity-check equations (Lemma 10) and, if $w \neq *$, the weight relation (same lemma). From the OR-proofs it yields relaxed openings of $\mathbf{c}_{i,0}$ with binary messages, whose slack is a difference of two elements of the challenge set. The OR-proof of EVOLVE is extractable from two transcripts that differ in (f_0, f_1) (del Pino et al., 2017, Thm. 3.2).

7.3 Aggregation

Aggregation tree. If an authority's openings were summed over all ballots, the bounds would grow with the number of voters (Figure 4). As in the improved scheme of EVOLVE (del Pino et al., 2017, Sec. 4.2), authority k therefore aggregates along a tree of fan-in l. The leaves are the share commitments $\mathbf{d}_{x,0} = \mathbf{c}_{x,k}$, $x \in \mathsf{Acc}$, in the order of the board. For $s \geq 1$, the u-th node of level s is a fresh commitment $\mathbf{d}_{u,s} = \mathsf{Com}(\sum_x \mathbf{m}_{x,s-1}; \boldsymbol{\rho}_{u,s})$ to the sum of the messages of its at most l children. Then $\boldsymbol{\delta}_{u,s} = \mathbf{d}_{u,s} - \sum_x \mathbf{d}_{x,s-1}$ is a commitment to $\mathbf{0}$ whose randomness the authority knows. The last level consists of a single node, the root, whose message is $\mathbf{V}_k$. The contribution Π_k consists of all node commitments, the randomness $\boldsymbol{\rho}_{\text{root}}$ of the root, and two amortized proofs. Π_{open} proves knowledge of short $\mathbf{r}$ with $\mathbf{Ar} = \mathbf{d}_{u,s,1}$ for every node of every level, including the leaves. Π_{zero} proves knowledge of short $\mathbf{r}$ with $\mathbf{Cr} = \boldsymbol{\delta}_{u,s}$ for every node of level $s \geq 1$. VerAgg recomputes the $\boldsymbol{\delta}_{u,s}$, verifies both proofs, and checks that $\|\boldsymbol{\rho}_{\text{root}}\|_2 \leq \beta$ and $\mathbf{d}_{\text{root}} = \mathbf{C}\boldsymbol{\rho}_{\text{root}} + \binom{\mathbf{0}}{\mathbf{V}_k}$.

Choice of the amortized proof. We instantiate Π_{open} and Π_{zero} with the amortized proof of Baum & Lyubashevsky (2017), split into blocks of at most ψ columns. This is a three-move protocol whose second challenge $C_2 = H(\mathbf{Z}_1, c_1)$ is derived by a random oracle H from the first response. By (Baum & Lyubashevsky, 2017, Thm. 3.5), two accepting transcripts with the same first message and different first challenges yield either the witnesses or a Module-SIS solution. This is the form of extraction that our reduction needs. EVOLVE uses the amortized proof of del Pino & Lyubashevsky (2017) instead. Its knowledge extractor combines an imperfect proof with a compiler and runs the prover many times, so it does not yield openings from two transcripts. The statement in EVOLVE that all authority proofs can be extracted with one rewinding (del Pino et al., 2017, App. A) is therefore informal.

We use the proof of Baum & Lyubashevsky (2017) with two changes. First, the second challenge $C_2 = H(\mathbf{Z}_1, c_1)$ is uniform in $\{-1,1\}^{\psi\times\ell}$ instead of $\{0,1\}^{\psi\times\ell}$; Lemma 11 adapts the extraction, and the paragraph *Why the second challenge is signed* below explains why the change is needed. Second, each of the two rejection steps is applied once per attempt to all blocks together. Blocks of Π_{open} contain up to ψ nodes of any levels; blocks of Π_{zero} contain nodes of a single level. The number of blocks is

$$E = \left\lceil \frac{\sum_{s\geq 0} n_s}{k} \right\rceil + \sum_{s\geq 1} \left\lceil \frac{n_s}{k} \right\rceil,$$

where n_s is the number of nodes of level s; for $l = 30$ and $\psi = 500$ it is $E = 24$, 217 and 2142 for $N_V = 10^4$, 10^5 and 10^6.

Lemma 11 (Signed second challenge)**.** *Assume that all nonzero elements of R_q of infinity norm at most 2 are invertible and that $18\sigma_2 < q/2$. Let H be a random oracle answered from one lazily sampled table that is never reprogrammed, in particular the same table in all runs of a rewinding argument, and let q_H bound the number of distinct queries to H in all runs, including the queries made to verify transcripts. Consider the proof of Baum & Lyubashevsky (2017) with $C_2 = H(\mathbf{Z}_1, c_1)$ uniform in $\{-1,1\}^{\psi\times\ell}$, and call a transcript accepting if it satisfies the verification equations and the norm checks described below: rows of $\mathbf{Z}_1$ of norm at most $\sqrt{2\psi}\,\sigma_1$, entries of $\mathbf{Z}_2$ of absolute value at most $9\sigma_2$, and ring components of the columns of $\mathbf{Z}_2$ of norm at most $\sqrt{2N}\sigma_2$. From two accepting transcripts with the same first message and first challenges $c_1 \neq c_1'$ in the challenge set $\mathcal{S}$ (polynomials with 60 coefficients in $\{\pm 1\}$ and all others zero) one computes, by fixed formulas, either a nonzero $\mathbf{z}$ with $\mathbf{A}\mathbf{z} = \mathbf{0}$ and $\|\mathbf{z}\|_\infty \leq 2^{5/2}\psi\sigma_1 + 2^{5/2}\sqrt{60N}\sigma_2$, or $\hat{\mathbf{S}}$ with $\mathbf{A}\hat{\mathbf{S}} = 2\mathbf{U}$. There is an event over the table of H, of probability at most $q_H^2\, 2^{-\ell-1} + 2^{-\ell}$, outside which, for all such pairs of transcripts at once, the second case satisfies $\|\hat{\mathbf{S}}\|_\infty \leq 2\|\mathbf{Z}_2 - \mathbf{Z}_2'\|_\infty$.*

Proof. We follow the proof of (Baum & Lyubashevsky, 2017, Thm. 3.5); Appendix B gives the proof in full, including the order of the oracle queries on which the probability bound rests. Let $\bar{\mathbf{S}} = \mathrm{Rot}^{-1}(c_1 - c_1')(\mathbf{Z}_1 - \mathbf{Z}_1')$, so that $\mathbf{A}\bar{\mathbf{S}} = \mathbf{U}$. As there, either $\mathbf{Z}_2 - \mathbf{Z}_2' = \bar{\mathbf{S}}(C_2 - C_2')$, or some column of $\mathrm{Rot}(c_1 - c_1')(\mathbf{Z}_2 - \mathbf{Z}_2') - (\mathbf{Z}_1 - \mathbf{Z}_1')(C_2 - C_2')$ is a nonzero preimage of $\mathbf{0}$. The columns of $C_2 - C_2'$ now have entries in $\{0, \pm 2\}$ and norm at most $2\sqrt{\psi}$, which doubles the first term of the bound; the second term is unchanged. In the first case write $C_2 - C_2' = 2D$. For fixed C_2', the entries of D are independent, and the entry (i,j) is 0 or $-(C_2')_{ij}$ with probability 1/2 each. For a row $\mathbf{m}$ of a matrix M, the j-th entry of $\mathbf{m}D$ equals $\langle \mathbf{m} \circ \boldsymbol{\epsilon}_j, \mathbf{b}_j \rangle$ with $\boldsymbol{\epsilon}_j = -(C_2')_{\cdot j}$ and $\mathbf{b}_j$ uniform in $\{0,1\}^\psi$, and $\|\mathbf{m} \circ \boldsymbol{\epsilon}_j\|_\infty = \|\mathbf{m}\|_\infty$. Hence Lemma 2.3 and Corollary 2.4 of (Baum & Lyubashevsky, 2017) give $\Pr[\|MD\|_\infty < \frac{1}{2}\|M\|_\infty] \leq 2^{-\ell}$ for every fixed M, and the argument of (Baum & Lyubashevsky, 2017, Lemma 3.6), applied to $M = 2\bar{\mathbf{S}}$, shows that $\|2\bar{\mathbf{S}}\|_\infty \leq 2\|2\bar{\mathbf{S}}D\|_\infty = 2\|\mathbf{Z}_2 - \mathbf{Z}_2'\|_\infty$ except with the stated probability. Set $\hat{\mathbf{S}} = 2\bar{\mathbf{S}}$. The argument must be applied to $2\bar{\mathbf{S}}$ and not to $\bar{\mathbf{S}}$: modulo q, $\bar{\mathbf{S}}D = 2^{-1}(\mathbf{Z}_2 - \mathbf{Z}_2')$, whose entries have representatives of size about $q/2$ wherever an entry of $\mathbf{Z}_2 - \mathbf{Z}_2'$ is odd, whereas $2\bar{\mathbf{S}}D = \mathbf{Z}_2 - \mathbf{Z}_2'$ is short. □

With the norm check below, $\|\mathbf{Z}_2 - \mathbf{Z}_2'\|_\infty \leq 18\sigma_2$, so the extracted openings have slack 2 and ℓ_2-norm at most $B_A = 36\sigma_2\sqrt{N(2d+L)}$ per column. The slack is harmless: 2 is invertible modulo q, and the lemmas below carry it along.

Lemma 12 (Aggregation tree)**.** *Suppose that, for authority k:*

- *every node $\mathbf{d}_{u,s}$ has $\hat{\mathbf{r}}_{u,s}$ with $\mathbf{A}\hat{\mathbf{r}}_{u,s} = 2\mathbf{d}_{u,s,1}$ and $\|\hat{\mathbf{r}}_{u,s}\|_2 \leq B_A$; set $\hat{\mathbf{m}}_{u,s} = \mathbf{d}_{u,s,2} - 2^{-1}\mathbf{B}\hat{\mathbf{r}}_{u,s}$, so that $(\hat{\mathbf{m}}_{u,s}, \hat{\mathbf{r}}_{u,s}, 2)$ is a relaxed opening of $\mathbf{d}_{u,s}$;*
- *every node of level $s \geq 1$ has $\hat{\mathbf{r}}'_{u,s}$ with $\mathbf{C}\hat{\mathbf{r}}'_{u,s} = 2\boldsymbol{\delta}_{u,s}$ and $\|\hat{\mathbf{r}}'_{u,s}\|_2 \leq B_A$;*
- *$\boldsymbol{\rho}_{\mathrm{root}}$ passes the checks of* VerAgg*.*

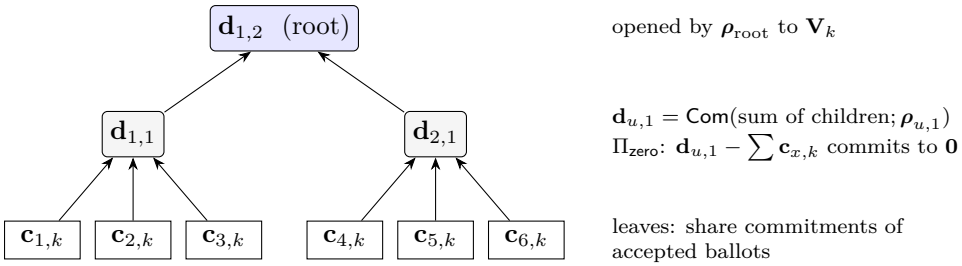


Figure 4: Aggregation tree of authority k for six accepted ballots and fan-in $l = 3$. Each inner node is a fresh commitment to the sum of its children's messages, and only the root is opened. By Lemma 12, the opened value equals the sum of the extracted shares unless Module-SIS is solved, and the bounds do not grow with the number of voters.

Round 1 (after the checking phase, deadline Δ_0): each participating authority k builds its tree over the accepted ballots with fresh node randomness, prepares κ attempts of Π_{open} and Π_{zero}, and posts the node commitments, $\mathbf{V}_k$, and $h_{k,\iota} = G(\mathbf{W}_k^{(\iota)}, s_{k,\iota})$ for $\iota \in [\kappa]$, with fresh salts $s_{k,\iota}$.

Challenges: $(c_{1,k,\iota,e})_{k\in[n],\,\iota\in[\kappa],\,e} = H_{\mathsf{agg}}(\mathsf{par}, \mathsf{BB}_1)$, where BB_1 is the board when round 1 closes.

Round 2a (deadline Δ_1): authority k computes its attempts in order, with one joint rejection step per stage over all blocks, and posts the index ι_k of its first attempt that does not abort, or $\perp$.

Round 2b (deadline Δ_2; only if at least t indices were posted): each authority that posted an index posts s_{k,ι_k}, the responses of attempt ι_k and $\boldsymbol{\rho}_{\text{root}}$. An authority that posted an index but no contribution accepted by VerAgg is recorded as faulty and takes no part in later rounds.

Result: the round succeeds if at least t contributions pass VerAgg, which recomputes $\mathbf{W}_k^{(\iota_k)}$ from the challenges and the responses and checks h_{k,ι_k}. Otherwise the tally round is repeated from round 1 with fresh randomness. Contributions carry the index ν of their round; VerAgg derives their challenges from $H_{\mathsf{agg}}(\mathsf{par}, \mathsf{BB}_1^{(\nu)})$ of that round and rejects indices $\iota_k \notin [\kappa]$; only the first round-1 entry of each authority in a round counts; and $\mathsf{Combine}$ and Verify accept only t contributions of the same round.

Figure 5: The two-round aggregation of the EVOLVE-style instantiation.

Then either $\mathbf{V}_k = \sum_{x\in\mathsf{Acc}} \hat{\mathbf{m}}_{x,0}$, *or one can compute* $\mathbf{z}$ *with* $\mathbf{Az} = \mathbf{0}$, $\mathbf{Bz} \neq \mathbf{0}$ *and* $\|\mathbf{z}\|_2 \leq \max\big((l+2)B_A,\ 2\beta + B_A\big)$.

Proof. For a node (u, s) with $s \geq 1$, let $\mathbf{z} = \hat{\mathbf{r}}'_{u,s} - \hat{\mathbf{r}}_{u,s} + \sum_x \hat{\mathbf{r}}_{x,s-1}$, where the sum ranges over the children. Since $\boldsymbol{\delta}_{u,s} = \mathbf{d}_{u,s} - \sum_x \mathbf{d}_{x,s-1}$, we get $\mathbf{Az} = \mathbf{0}$ and $\mathbf{Bz} = 2\big(\hat{\mathbf{m}}_{u,s} - \sum_x \hat{\mathbf{m}}_{x,s-1}\big)$, and $\|\mathbf{z}\|_2 \leq (l+2)B_A$. For the root, $\mathbf{z} = 2\boldsymbol{\rho}_{\text{root}} - \hat{\mathbf{r}}_{\text{root}}$ satisfies $\mathbf{Az} = \mathbf{0}$, $\mathbf{Bz} = 2(\hat{\mathbf{m}}_{\text{root}} - \mathbf{V}_k)$ and $\|\mathbf{z}\|_2 \leq 2\beta + B_A$. Since 2 is invertible modulo q, $\mathbf{Bz} = \mathbf{0}$ holds exactly when the messages agree. If none of these vectors has $\mathbf{Bz} \neq \mathbf{0}$, then the message of every node is the sum of the messages of its children and $\mathbf{V}_k = \hat{\mathbf{m}}_{\text{root}}$, so induction over the levels gives $\mathbf{V}_k = \sum_x \hat{\mathbf{m}}_{x,0}$. □

A vector with $\mathbf{Az} = \mathbf{0}$ and $\mathbf{Bz} \neq \mathbf{0}$ is nonzero modulo q, and hence is a Module-SIS solution for $\mathbf{A}$.

Hashing. EVOLVE derives the challenges of all authorities from one hash of the whole board, so that all their proofs can be extracted with one rewinding (del Pino et al., 2017, App. A). Here, $\mathsf{Contrib}_k$ is split into two rounds for the same purpose; the first messages are committed to in the first round and revealed in the second. Figure 5 gives the two rounds. There, $\mathbf{W}_k^{(\iota)}$ collects the first messages $(\mathbf{W}_1, \mathbf{W}_2)$ of all blocks of Π_{open} and Π_{zero} of attempt ι, and e ranges over the blocks of authority k. The board BB_1 consists of the ballots, the complaints and everything posted in round 1; in tally round ν we write $\mathsf{BB}_1^{(\nu)}$. The salts have λ bits; the implementation uses 256-bit salts, so that the term $q_G 2^{-\lambda}$ below, summed over 2^{40} attempts, is at most 2^{-152} for $q_G \leq 2^{64}$. The deadlines Δ_0, Δ_1 and Δ_2 are fixed in par and measured on the board. Anyone can check

whether an authority is faulty. VerAgg recomputes the first messages as in the non-interactive version of (Baum & Lyubashevsky, 2017). Attempts that abort are never opened; the choice of κ and the resulting cost are analyzed below. Since BB_1 contains the accepted ballots, the challenges depend on the leaves of every tree. The random oracles G, H_{bal}, H_{agg} and H are domain-separated.

Zero knowledge of the aggregation. In the privacy proof, the simulator of an honest authority k posts uniformly random values $h_{k,\iota}$ in the first round. Once BB_1 is fixed, it obtains the challenges by querying $H_{\mathsf{agg}}(\mathsf{par}, \mathsf{BB}_1)$, so nothing is programmed at H_{agg}. It then runs the simulator of (Baum & Lyubashevsky, 2017, Lemma 3.3) on these challenges; this simulator queries H but does not program it. Finally, the simulator of authority k programs G at $(\mathbf{W}_k^{(\iota_k)}, s_{k,\iota_k})$ to return h_{k,ι_k}. Programming fails only if G was queried at this point before, which happens with probability at most $q_G 2^{-\lambda}$ because the salt is fresh and is revealed only in the second round. The values $h_{k,\iota}$ of unopened attempts are uniformly random in both the real and the simulated protocol. Under the bounds of Lemma 13, the output of each of the two rejection steps, a response or $\perp$, is within statistical distance $e^{-r^2/2}/M'$ of that of an ideal step that, independently of the witness, outputs a Gaussian sample with probability $1/M'$ and $\perp$ otherwise (Lyubashevsky, 2012, Thm. 4.6). The simulator samples the index ι_k announced first, and whether it exists, from the same distribution. Hence these are simulated with an error of at most $2e^{-r^2/2}/M' + (l(E+1)+1)e^{-\tau} + \varepsilon_{\text{fresh}}$ per attempt. For $\tau = 187 \ln 2$ and $E \le 2142$, the largest value used here, $(l(E+1)+1)e^{-\tau} \le 2^{-171}$, since $30 \cdot 2143 + 1 < 2^{16}$; the same holds in general for $E \le 2183$. The responses of attempts that are not revealed never leave the authority. The randomness of the intermediate nodes is never revealed, so the node commitments are hidden by the hiding property. The root, a fresh commitment to $\mathbf{V}_k$ opened by $\boldsymbol{\rho}_{\text{root}}$, reveals only $\mathbf{V}_k$. These simulation errors are part of ε_{zk} in Theorem 3, which therefore depends on the number Q of oracle queries; the parameters of Table 3 use $Q = 2^{64}$.

Aborts in the aggregation. In (Baum & Lyubashevsky, 2017), each block rejects twice. If every block of an attempt had to pass its own two rejection steps, an attempt would succeed with probability about 3^{-E}, that is 2^{-38}, 2^{-344} and 2^{-3395} for the three sizes above, which is not practical. We therefore apply each of the two rejection steps once per attempt, to the concatenation of the corresponding shifts of all blocks. Here σ_i is the smallest value achievable by the sampler (Appendix C) that is at least $\alpha' \Theta_i$, where $\alpha' = 281/10$, and $M' = \exp(r/\alpha' + 1/(2\alpha'^2))$ with $r = 309/20$ as for the ballot proof, so that $\ln M' = 86929/157922$ is rational and $M' \approx 1.734 \approx \sqrt{3}$. An attempt then succeeds with probability about $1/M'^2 \approx 0.333$, independently of E. The norm check of every block requires that the rows of $\mathbf{Z}_1$ have norm at most $\sqrt{2\psi}\,\sigma_1$, that the entries of $\mathbf{Z}_2$ be at most $9\sigma_2$ in absolute value, and that every ring component of every column of $\mathbf{Z}_2$ have norm at most $\sqrt{2N}\sigma_2$. The bound on the entries is $7\sigma_2$ in (Baum & Lyubashevsky, 2017). With $9\sigma_2$, an honest attempt fails the entry check with probability about 2^{-36}, 2^{-33} and 2^{-30} for the three sizes; the prover counts such an attempt as aborting, and the prover and the simulator apply the same check. The rejection steps are only correct if Θ_1 and Θ_2 bound the shifts of the witnesses that actually occur.

Why the second challenge is signed. With a binary C_2, a column of $\mathbf{S}C_2$ is the sum of about half of the witness columns of a block, and the mean $\frac{1}{2}$ of the entries of C_2 contributes $\frac{1}{2}\|\mathbf{S1}\|_2$. For independent columns this is of the order of $\sqrt{\psi}$ times the norm of one column. But the randomness of a leaf is chosen by a voter. Dishonest voters can correlate the randomness of their shares by selecting seeds. If the seed alone determined the randomness, they could make it identical, which would make $\|\mathbf{S1}\|_2$ equal to ψ times the norm of one column. A bound on $\frac{1}{2}\|\mathbf{S1}\|_2$ that holds for every such choice needs either a union over Q^ψ choices per block or the inequality $\|\mathbf{S}C_2\|_F \le \|\mathbf{S}\|_F\, s_1(C_2)$ with $s_1(C_2) \approx \sqrt{\psi\ell}/2$, which for our parameters costs 3.1 to 3.7 bits in β_{SIS}. With a signed C_2, $\|\mathbf{S}C_2\|_F^2$ concentrates at $\ell\|\mathbf{S}\|_F^2$ for every $\mathbf{S}$, however its columns are aligned, and Lemma 13 needs only bounds on the individual columns. Since the input of X contains the identity and the share index, different leaves are outputs of X at different

inputs, which are independent in the random oracle model even if seeds repeat. A voter therefore cannot make two leaves equal by reusing a seed. The adversary can still select seeds among many queries, which affects the leaves below one node of level 1; this is covered by the union over $g_u \ln Q$ in b_u. The price of the signed challenge is the slack 2 of Lemma 11.

Lemma 13 (Norms of the aggregation shifts)**.** *Let N_V be a hard upper bound on the number of accepted ballots, for instance the number of eligible voters, and let N_{col} and E be the numbers of witness columns and of blocks of one authority for a tree with N_V leaves. Assign to each column u the class $(v_u, g_u) = (1, 1)$ if it belongs to Π_{open}, $(1 + l, l)$ if it belongs to Π_{zero} at level 1, and $(1 + l, 0)$ if it belongs to Π_{zero} above level 1; v_u bounds the number of randomness vectors summed in the column and g_u the number of leaves among them. Model X, H_{agg}, H and G as random oracles, with salts of λ bits, let Q bound the number of distinct inputs at which X is evaluated by any party and the number of distinct queries to H_{agg} in the whole experiment, and let the authority sample the randomness of inner nodes from D_σ. For $\tau > 0$ let $h(x, \xi, \tau) = x + 2\sqrt{x\xi\tau} + 2\xi\tau$, $\tau_u = \tau + \ln N_{\mathrm{col}} + g_u \ln Q$, and*

$$b_u^2 = \sigma^2 v_u \, h(N\mu, 1, \tau_u), \qquad \Theta_1^2 = \sum_u \sigma^2 v_u \, h(60N\mu, 3600, \tau_u + \ln Q),$$

$$\Theta_2^2 = \ell F + 2\sqrt{\ell\tau \textstyle\sum_e F_e^2} + 2\tau \max_e F_e,$$

where $\mu = 2d + L$, u and e range over the columns and blocks of the tree for N_V leaves, $F_e = \sum_{u \in e} b_u^2$ and $F = \sum_e F_e$. Consider one attempt of an honest authority, for any number of accepted ballots up to N_V chosen by the adversary. Then, except with probability $(l(E + 1) + 1)e^{-\tau} + \varepsilon_{\mathrm{fresh}}$, every column of the actual tree satisfies $\|\mathbf{s}_u\|_2 \le b_u$, with b_u the value of its class, the first shift satisfies $\|(c_{1,e}\mathbf{s}_u)_u\|_2 \le \Theta_1$, and for signed second challenges $\|(\mathbf{S}_e C_{2,e})_e\|_F \le \Theta_2$. Here $\varepsilon_{\mathrm{fresh}} \le q_G 2^{-\lambda} + q_H^* E \sigma_1^{-N\mu}$ bounds the probability that one of the queries $H(\mathbf{Z}_{1,e}, c_{1,e})$ of the attempt was made before, where q_H^* counts the queries to H by all parties in the experiment, including the authority's own queries, and q_G the queries to G by parties other than the authority.

Proof. We sketch the argument; Appendix B gives it in full. The answers of X, the answers of H_{agg} and the node randomness can be viewed as three independent sequences of independent samples, and every column of every tree with at most N_V leaves is a signed sum of at most v_u entries of these sequences with distinct indices; we call the positions and indices involved its description. For a fixed description, the column is $\sigma\sqrt{v_u}$-subgaussian (Micciancio & Peikert, 2012, Lemma 2.8), and (Hsu et al., 2012, Thm. 2.1) with $\Phi = I$ bounds its norm by b_u except with probability $e^{-\tau_u}$. A union bound over all descriptions, of which there are at most Q^{g_u} per position and number of children, covers every adaptive choice of seeds and every number of accepted ballots, and gives the first claim except with probability $l\,e^{-\tau}$. For the second claim, a fixed description, a fixed query to H_{agg} and a fixed block give a challenge that is uniform and independent of the column, and the same theorem applies with Φ the multiplication by c_1 (squared Frobenius norm $60N\mu$, operator norm at most 60); the union over the Q queries and the E blocks accounts for the $\ln Q$ in Θ_1 and a factor E in the probability. For the third claim, the entries of all $C_{2,e}$ form a Rademacher vector $\mathbf{x}$. If none of the queries $H(\mathbf{Z}_{1,e}, c_{1,e})$ was made before, which fails with probability at most $\varepsilon_{\mathrm{fresh}}$ because $\mathbf{Z}_{1,e}$ contains a fresh mask, $\mathbf{x}$ is independent of the witnesses. Then $\|(\mathbf{S}_e C_{2,e})_e\|_F^2 = \mathbf{x}^\top \Sigma \mathbf{x}$ with $\Sigma = \bigoplus_e I_\ell \otimes \mathbf{S}_e^\top \mathbf{S}_e$, and given the column bounds $\operatorname{tr}\Sigma \le \ell F$, $\operatorname{tr}(\Sigma^2) \le \ell \sum_e F_e^2$ and $\|\Sigma\| \le \max_e F_e$; apply (Hsu et al., 2012, Thm. 2.1) once more. Because the bounds are per class, Θ_1, F, $\sum_e F_e^2$ and $\max_e F_e$ are nondecreasing in the number of leaves, so the values for N_V cover every smaller tree, whatever its block layout. □

The bounds grow with the number of columns, so σ_1, σ_2, B_A and with them β_{SIS} and q grow with the number of voters, by about $\frac{1}{2}\log_2 E$ bits. The implementation computes all of these quantities from the formulas of Lemma 13, and in the protocol test of the implementation

($N_V = 70$, $E = 3$ blocks) the measured shifts were at most $0.17\,\Theta_1$ and $0.43\,\Theta_2$; at the sizes of Table 3 the bounds leave less room, as the ratios grow with E. Table 3 gives parameters for $N_V = 10^4$, 10^5 and 10^6.

Choice of κ and expected cost. An honest authority has a non-aborting attempt in a round with probability $s(\kappa) = 1 - (1-p)^\kappa$, where $p = 1/M'^2 \approx 0.333$ is the success probability of one attempt. In the worst case for robustness, only t authorities are honest, and the round succeeds with probability at least $s(\kappa)^t$. The expected number of rounds is then at most $s(\kappa)^{-t}$. Since an honest authority prepares κ attempts per round, its expected work is at most $\kappa\, s(\kappa)^{-t}$ attempts. We take $\kappa = 4$: for $t = 3$, on average at most 1.9 rounds and 7.8 attempts per authority (Appendix C gives the trade-off for other κ). If all n authorities are honest, the round only needs any t of them to succeed; for $n = 4$, $t = 3$ and $\kappa = 4$ this happens with probability 0.82, that is, 1.2 rounds on average. One attempt costs one computation of the first messages for all E blocks, which dominates the work of the authority. A failed round costs a repetition of both rounds; misbehaving authorities cannot make a round fail that would otherwise succeed (Theorem 7), and Appendix C gives the cost of failed rounds. The published responses belong to accepted attempts, which the simulator of the privacy proof produces exactly as in a successful round, and the values $\mathbf{V}_k$ are the same in every round. The simulation errors are summed over all attempts of all rounds, of which we assume at most 2^{40}, as in the choice of τ. Repeating a round is compatible with Theorem 14. The reduction forks at the query $H_{\mathsf{agg}}(\mathsf{par}, \mathsf{BB}_1^{(\nu)})$ of the round whose contributions are output; by the rule above, this is a single round. The index J_a counts the queries of all rounds and therefore already identifies it.

7.4 Verifiability

Theorem 14 (Verifiability of the EVOLVE-style instantiation)**.** *Assume:*

(i) all nonzero elements of R_q of infinity norm at most 2 are invertible;

(ii) ballot extraction: *from two accepting ballot proofs with the same input to H_{bal}, with $c \neq c'$ and, for every candidate a, OR-transcripts that differ in $(f_{a,0}, f_{a,1})$, one computes relaxed openings $(\mathbf{m}_j, \bar{\mathbf{r}}_j, f)$ of $\mathbf{c}_0, \ldots, \mathbf{c}_n$ with $f = c - c'$ and bound $\bar{B}_V = 2B_J$ that satisfy the parity-check equations (Lemma 10), and, row by row, relaxed openings of $\mathbf{c}_0$ with binary messages, bound $\bar{B}_{\mathrm{OR}} = 2B_J$ and a slack f' that is a difference of two elements of the challenge set $\mathcal{S}$ (del Pino et al., 2017, Thm. 3.2), where Lemma 9 is applied row by row to the matrix $\binom{\mathbf{A}}{\mathbf{b}_a}$; and, if $w \neq *$, the weight relation $\sum_{a=1}^{L} m_{0,a} = w$ for the message $\mathbf{m}_0$ of the relaxed opening $(\mathbf{m}_0, \bar{\mathbf{r}}_0, f)$ of $\mathbf{c}_0$ above (Lemma 10);*

(iii) aggregation extraction: *for every block of Π_{open} and Π_{zero} and every two accepting transcripts with the same first message and first challenges $c_1 \neq c_1'$, where H is answered from one table in all runs, the formulas of Lemma 11 compute either a vector $\mathbf{z} \neq \mathbf{0} \bmod q$ with $\mathbf{A}\mathbf{z} = \mathbf{0}$ and $\|\mathbf{z}\|_2 \leq \beta_{\mathrm{BL}}$, or columns $\hat{\mathbf{r}}$ with $\mathbf{A}\hat{\mathbf{r}} = 2\mathbf{y}$ (resp. $\mathbf{C}\hat{\mathbf{r}} = 2\mathbf{y}$ for Π_{zero}) for every statement column $\mathbf{y}$, i.e. openings with slack 2; outside an event Bad_H over the table of H, the latter satisfy $\|\hat{\mathbf{r}}\|_2 \leq B_A$. By Lemma 11, $\Pr[\mathsf{Bad}_H] \leq \varepsilon_H = q_H^2\, 2^{-\ell-1} + 2^{-\ell}$, where q_H counts the distinct queries to H in all runs of $\mathcal{A}$ by $\mathcal{B}_2$, including those made to verify transcripts. Here $B_A = 36\sigma_2\sqrt{N(2d+L)}$ and $\beta_{\mathrm{BL}} = (2^{5/2}\psi\sigma_1 + 2^{5/2}\sqrt{60N}\sigma_2)\sqrt{N(2d+L)}$.*

Let $\eta_{\max}$ bound $\eta(f)$ over differences of challenges (we take $\eta_{\max} = 120$, since $\eta(c-c') \leq \|c-c'\|_1 \leq 120$ for $c, c' \in \mathcal{S}$), let $\bar{B}_V$ and $\bar{B}_{\mathrm{OR}}$ be the extraction bounds of the ballot proof and of the OR-proof, that is, twice the respective bounds on the responses, let β be the bound on honest commitment randomness, and let

$$\beta_{\mathrm{SIS}} = \max(2\beta + B_A,\ (l+2)B_A,\ 2\bar{B}_V + \eta_{\max}B_A,\ \eta_{\max}(\bar{B}_V + \bar{B}_{\mathrm{OR}}),\ \beta_{\mathrm{BL}}).$$

Let $\mathcal{A}$ win the verifiability experiment with probability ε. Let Q_b and Q_a bound the numbers of distinct queries to H_{bal} and H_{agg} in the verifiability experiment, counting the queries of $\mathcal{A}$, of

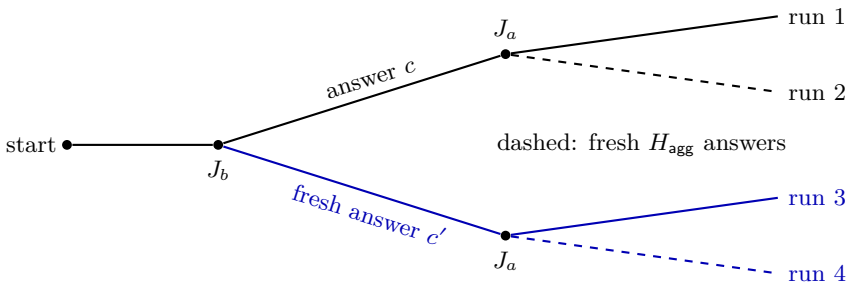
the oracles for honest parties and of verification, over all tally rounds; queries are numbered in the order of their first occurrence, and all indices below refer to this numbering. Let q_G bound the number of distinct queries to G in all runs of $\mathcal{A}$ by $\mathcal{B}_2$, and let $\Lambda = \binom{n}{t}\kappa^t$. Let $\delta_V \le (L+1)/|\mathcal{S}| + 2^{-2\lambda}$, where $2^{-2\lambda}$ bounds a repeated 2λ-bit output h of H_{bal}, from which the challenges are expanded (resp. $\delta_A \le tE/|\mathcal{S}|$, where E bounds the number of blocks of one authority and $\mathcal{S}$ is the challenge set; the ballot proof, the OR-proofs and the proof of Baum & Lyubashevsky (2017) all use the set of polynomials with 60 coefficients in $\{\pm 1\}$ and all others zero) bound the probability that a fresh output of H_{bal} (resp. H_{agg}) violates, relative to a fixed first run, the distinctness required in (ii) (resp. (iii)): for (ii), that $c' = c$ or that for some candidate a the new permutation maps the first run's $f_{a,0}$ to the first run's $f_{a,1}$, so that the second run could reuse the pair $(f_{a,0}, f_{a,1})$; for (iii), that $c'_{1,k,\iota_k,e} = c_{1,k,\iota_k,e}$ for some $k \in T$ and block e. Then there are $\mathcal{B}_1$ against EUF-CMA and $\mathcal{B}_2$ against M-SIS with bound β_{SIS} for the matrix $\mathbf{A}$ such that, with $\varepsilon' = \varepsilon - \mathrm{Adv}^{\text{euf-cma}}_{\mathcal{B}_1}$,

$$\mathrm{Adv}^{\mathrm{msis}}_{\mathcal{B}_2} \;\ge\; \mathrm{acc}\Big(\frac{\mathrm{acc}}{Q_b} - \delta_V\Big) - \varepsilon_H, \qquad \mathrm{acc} \;\ge\; \varepsilon'\Big(\frac{\varepsilon'}{\Lambda Q_a} - \delta_A\Big) - \varepsilon_G,$$

where $\varepsilon_G \le q_G^2\, 2^{-2\lambda}$ bounds a collision of G. When δ_A, δ_V, ε_G and ε_H are negligible, this gives the leading-order estimate $\mathrm{Adv}^{\mathrm{msis}}_{\mathcal{B}_2} \gtrsim \varepsilon'^4/(\Lambda^2 Q_a^2 Q_b)$. $\mathcal{B}_2$ runs $\mathcal{A}$ four times.

Proof strategy. The reduction $\mathcal{B}_2$ receives an M-SIS instance $\mathbf{A}$ and must output a short nonzero solution whenever the adversary makes a result accepted that is not the tally of the honest votes. It forks the adversary twice (Figure 6). The inner fork, at the query $H_{\mathsf{agg}}(\mathsf{par}, \mathsf{BB}_1)$ that fixes the challenges of the output tally round, gives two accepting transcripts of every amortized proof of the quorum and, through hypothesis (iii) and Lemma 12, relaxed openings with slack 2 of every share of every accepted ballot, whose sums are the published column sums $\mathbf{V}_k$; the label (T, ι) in the forking lemma accounts for the quorum and the attempts being chosen by the adversary. If the opened shares of every honest ballot are its committed ones and every dishonest ballot opens to a valid vote, the accepted result is the sum of these votes, as in Theorem 5, and the adversary has not won; so either some honest ballot has an opened share that differs from the committed one, and Lemma 9 turns the two openings into an M-SIS solution, or some dishonest ballot i^* is opened to shares whose interpolation $\mathbf{u}_{i^*}$ lies outside $\mathcal{V}_{L,w}$, and the outer fork, at the query H_{bal} of that ballot, gives relaxed openings of its commitments from its own proof by hypothesis (ii); two openings of one commitment with different messages, or a message outside the ballot format, give the M-SIS solution by Lemmas 9 and 10. The bound β_{SIS} collects the norms of the five possible solutions, and the loss $\varepsilon'^4/(\Lambda^2 Q_a^2 Q_b)$ is that of two nested forks.

Proof. We use the general forking lemma (Bellare & Neven, 2006, Lemma 1) in a variant with a label and a bad set, which we state and prove. Let $\mathcal{F}$ be an algorithm that on input $(\rho, h_1, \ldots, h_Q)$, with $h_1, \ldots, h_Q$ independent and uniform, returns $\perp$ or a pair (J, γ) with $J \in [Q]$ and a label γ from a set of size Λ, and let $\mathrm{acc} = \Pr[\mathcal{F} \neq \perp]$. For every first run, i.e. every $(\rho, h_1, \ldots, h_Q)$ with output (J, γ), let Bad be a set of answers determined by this run, with $\Pr_{h'}[h' \in \mathrm{Bad}] \le \delta$. The fork runs $\mathcal{F}(\rho, h)$; if this returns (J, γ), it runs $\mathcal{F}(\rho, h_1, \ldots, h_{J-1}, h'_J, \ldots, h'_Q)$ with fresh $h'_J, \ldots, h'_Q$, and it succeeds if the second run returns the same pair and $h'_J \notin \mathrm{Bad}$. Then the fork succeeds with probability at least $\mathrm{acc}^2/(\Lambda Q) - \delta\,\mathrm{acc}$. Indeed, let $p_{j,\gamma}$ be the probability that $\mathcal{F}$ returns (j, γ), given ρ and $h_1, \ldots, h_{j-1}$. Given these, the two runs are independent, so both return (j, γ) with probability $\mathbb{E}[p_{j,\gamma}^2] \ge \mathbb{E}[p_{j,\gamma}]^2$, and by the Cauchy–Schwarz inequality $\sum_{j,\gamma} \mathbb{E}[p_{j,\gamma}]^2 \ge \big(\sum_{j,\gamma} \mathbb{E}[p_{j,\gamma}]\big)^2/(\Lambda Q) = \mathrm{acc}^2/(\Lambda Q)$. The answer h'_j is independent of the first run, so, given that the first run returned (j, γ), it lies in the bad set with probability at most δ; this subtracts at most $\delta\,\mathrm{acc}$. For $\Lambda = 1$ this is (Bellare & Neven, 2006, Lemma 1) with a bad set. The two forks are nested as in the double-forking argument of Maxwell et al. (2019). The extraction



J_b: query H_{bal} for the proof of ballot i^*; J_a: query $H_{\mathsf{agg}}(\mathsf{par}, \mathsf{BB}_1)$.
Runs 1 and 2 (inner algorithm $\mathcal{F}$): same BB_1, so openings $\hat{\mathbf{m}}_{i,k}$ with slack 2 for all $k \in T$ (Lemma 12), and i^* with $\mathbf{u}_{i^*} \notin \mathcal{V}_{L,w}$ is found.
Runs 3 and 4 (rerun of $\mathcal{F}$): must return the same J_b; only the transcript of ballot i^* with challenge c' is used.
Same input at J_b, hence the same commitments $(\mathbf{c}_{i^*,j})_j$ and first messages; $c \neq c'$ gives relaxed openings of ballot i^*.

Figure 6: The nested forks in the proof of Theorem 14. The outer fork is at the earlier query J_b and the inner fork at the later query J_a. The ballot i^* that has to be rewound is determined only after the inner fork.

below computes its candidates by fixed formulas and checks no norms; the norm bounds of (iii) are used only for the final output, and the event Bad_H is subtracted once at the end.

Setup. $\mathcal{B}_2$ receives $\mathbf{A}$, samples $\mathbf{B}$ and sets $\mathbf{C} = \binom{\mathbf{A}}{\mathbf{B}}$ as in BDLOP. It generates the signing keys of honest voters and answers OVote honestly, so it knows exact openings $(\mathbf{m}_{i,j}, \mathbf{r}_{i,j}, 1)$ with $\|\mathbf{r}_{i,j}\|_2 \leq \beta$ of the commitments of every honest ballot. It answers the ℓ-th distinct query to H_{bal} with h_ℓ^{bal} and the ℓ-th distinct query to H_{agg} with h_ℓ^{agg}; its own queries, made for honest ballot proofs, are included. It answers H and G from lazily sampled tables that are shared by all runs below and are never forked, so that (iii) applies with q_H the total number of distinct queries to H, including those made to verify transcripts, and a collision of G in any run is bounded by ε_G; the output length of G is at least 2λ bits. As in Game 1 of Theorem 5, we abort on a forged ballot, at cost $\mathrm{Adv}^{\text{euf-cma}}_{\mathcal{B}_1}$, so every ballot in Dis carries an identity that no honest ballot carries. Since the identity is part of the input to H_{bal}, the query for the proof of a ballot in Dis is made by $\mathcal{A}$ and not by $\mathcal{B}_2$. Without loss of generality, $\mathcal{A}$ queries H_{bal} on the input of every ballot proof before it posts the ballot, and H_{agg} on $(\mathsf{par}, \mathsf{BB}_1)$ before it outputs; this adds at most polynomially many queries, which are included in Q_b and Q_a.

Inner algorithm $\mathcal{F}$ (fork on H_{agg}). On randomness ρ, consisting of the coins of $\mathcal{A}$, the coins of honest parties, the answers h^{bal} and the fresh answers used in step 2, and on answers h^{agg}:

1. Run $\mathcal{A}$. If it does not win, return $\perp$. Otherwise let J_a be the index of the query $H_{\mathsf{agg}}(\mathsf{par}, \mathsf{BB}_1^{(\nu)})$ of the round ν whose contributions are output, T the quorum and $\iota = (\iota_k)_{k \in T}$ the indices of the first messages answered by the authorities in T. The forking index is J_a, with the label (T, ι), which takes at most Λ values. Store the aggregation transcripts of $k \in T$, and the transcripts of all ballot proofs on BB_1 together with the indices of their H_{bal} queries.

2. Rerun $\mathcal{A}$ with the same ρ and $h_1^{\mathsf{agg}}, \ldots, h_{J_a-1}^{\mathsf{agg}}$, and fresh answers from index J_a on. The fork succeeds if $\mathcal{A}$ wins again with the same index J_a and label (T, ι) and the first challenges $c_{1,k,\iota_k,e}$ differ in the two runs for all $k \in T$ and all blocks e; otherwise return $\perp$. Both runs are identical up to the J_a-th query, so the input $(\mathsf{par}, \mathsf{BB}_1)$ of that query is the same in both runs, and so are the accepted ballots, which are determined by the board at the close of the checking phase, their commitments, the node commitments and the values h_{k,ι_k}. Acceptance also depends on the answers of H_{bal} to the ballot proofs; by the convention in the setup, $\mathcal{A}$ queries H_{bal} on the input of every ballot proof before it posts the ballot,

hence before the J_a-th query, so these answers are the same in both runs. The openings of h_{k,ι_k} in the two runs are equal unless they form a collision of G; in that case $\mathcal{F}$ returns $\perp$, which is accounted for by ε_G. Hence the first messages of all blocks with index ι_k coincide. For every $k \in T$ and every block, compute from the two transcripts the candidates of Lemma 11: if some block gives a nonzero $\mathbf{z}$ with $\mathbf{Az} = \mathbf{0}$, store it and return the index 1; otherwise the blocks give openings with slack 2 of all nodes. No norm is checked. Apply Lemma 12, whose algebraic part does not use the norms: either it yields $\mathbf{z}$ with $\mathbf{Az} = \mathbf{0}$ and $\mathbf{Bz} \neq \mathbf{0}$, which is stored before returning the index 1, or the leaves give relaxed openings $(\hat{\mathbf{m}}_{i,k}, \hat{\mathbf{r}}_{i,k}, 2)$ of all $\mathbf{c}_{i,k}$ with $\mathbf{V}_k = \sum_{i \in \mathsf{Acc}} \hat{\mathbf{m}}_{i,k}$. Set $\mathbf{u}_i = \sum_{k \in T} \lambda_k^T \hat{\mathbf{m}}_{i,k}$, so that $\mathbf{R} = \sum_i \mathbf{u}_i$ by the check in Verify.

3. If $\hat{\mathbf{m}}_{i,k} \neq \mathbf{m}_{i,k}$ for some $i \in \mathsf{Hon}$ and $k \in T$, Lemma 9 applied to $(\mathbf{m}_{i,k}, \mathbf{r}_{i,k}, 1)$ and $(\hat{\mathbf{m}}_{i,k}, \hat{\mathbf{r}}_{i,k}, 2)$ yields an M-SIS solution of norm at most $2\beta + B_A$; store it and return the index 1. Otherwise $\mathbf{u}_i = \mathbf{v}_i$ for all $i \in \mathsf{Hon}$ by interpolation. If $\mathbf{u}_i \in \mathcal{V}_{L,w}$ for all $i \in \mathsf{Dis}$, then $\mathbf{R} = \sum_{i \in \mathsf{Hon}} \mathbf{v}_i + \sum_{i \in \mathsf{Dis}} \mathbf{u}_i$ over the integers, as in Theorem 5, contradicting that $\mathcal{A}$ wins. Hence there is $i^* \in \mathsf{Dis}$ with $\mathbf{u}_{i^*} \notin \mathcal{V}_{L,w}$. Return the index J_b of the H_{bal} query for the proof of b_{i^*}, together with the side output $(i^*, (\hat{\mathbf{m}}_{i^*,k}, \hat{\mathbf{r}}_{i^*,k})_{k \in T})$ and the first-run transcript $(\mathbf{w}, c, \mathbf{z})$ of that proof.

By the forking lemma with $Q = Q_a$, the label (T, ι) and $\delta = \delta_A$, applied to the run in step 1, the fork in step 2 succeeds with probability at least $\varepsilon'(\varepsilon'/(\Lambda Q_a) - \delta_A)$, since $\mathcal{A}$ wins without a forged ballot with probability at least ε'. Subtracting the collisions of G, $\mathcal{F}$ returns an index with probability $\mathrm{acc} \geq \varepsilon'(\varepsilon'/(\Lambda Q_a) - \delta_A) - \varepsilon_G$. The label includes T and ι because extraction for authority k needs two answers to the same first message.

Outer fork on H_{bal}. $\mathcal{B}_2$ runs $\mathcal{F}$ on (ρ, h^{agg}). If it returns an index J_b, $\mathcal{B}_2$ reruns $\mathcal{F}$ with the same coins, the same h^{agg} and the same inner fresh answers, the same $h_1^{\mathsf{bal}}, \ldots, h_{J_b-1}^{\mathsf{bal}}$, and fresh answers to H_{bal} from index J_b on. The outer fork succeeds if the rerun also returns J_b and the new answer at J_b is not in the bad set of (ii); by the forking lemma with $Q = Q_b$, $\Lambda = 1$ and $\delta = \delta_V$ this happens with probability at least $\mathrm{acc}(\mathrm{acc}/Q_b - \delta_V)$.

Output. If either run of $\mathcal{F}$ stored an M-SIS solution in step 2 or 3, output it. Otherwise both runs returned the same index J_b from the i^* case. The J_b-th query to H_{bal} has the same input in both runs, because the runs coincide before it. This input contains the identity, the commitments $(\mathbf{c}_{i^*,j})_{j=0}^n$ and the first messages $\mathbf{w}$, so the first run gives an accepting transcript $(\mathbf{w}, c, \mathbf{z})$ and the second run gives an accepting transcript $(\mathbf{w}, c', \mathbf{z}')$ with $c \neq c'$. The second transcript is accepting because in the second run the corresponding ballot lies in $\mathsf{Dis} \subseteq \mathsf{Acc}$. By (ii), obtain relaxed openings $(\mathbf{m}_j, \bar{\mathbf{r}}_j, f)$ of $\mathbf{c}_{i^*,j}$ satisfying the parity-check equations, and binary relaxed openings of $\mathbf{c}_{i^*,0}$ with slack f'. The commitments $\mathbf{c}_{i^*,k}$ are the ones for which the first run of $\mathcal{F}$ extracted $(\hat{\mathbf{m}}_{i^*,k}, \hat{\mathbf{r}}_{i^*,k}, 2)$. Apply Lemma 9:

- if $\mathbf{m}_k \neq \hat{\mathbf{m}}_{i^*,k}$ for some $k \in T$, output $\mathbf{z} = 2\bar{\mathbf{r}}_k - f\hat{\mathbf{r}}_{i^*,k}$, of norm at most $2\bar{B}_V + \eta_{\max} B_A$;
- if $\mathbf{m}_0$ differs from the binary message in some row, output the corresponding vector, of norm at most $\eta_{\max}(\bar{B}_V + \bar{B}_{\mathrm{OR}})$, where the lemma is applied to $(\mathbf{A}; \mathbf{b}_a)$ for the row $\mathbf{b}_a$ of $\mathbf{B}$.

One of the two cases must occur. Otherwise $\mathbf{m}_k = \hat{\mathbf{m}}_{i^*,k}$ for $k \in T$ and $\mathbf{m}_0 \in \{0,1\}^L$; if $w \neq *$, also $\sum_a m_{0,a} = w$ by (ii), which holds over the integers since the $m_{0,a}$ are in $\{0,1\}$ and $L < q$, so $\mathbf{m}_0 \in \mathcal{V}_{L,w}$. By the parity-check characterization, the $\mathbf{m}_j$ are the values of one polynomial of degree less than t, so $\mathbf{u}_{i^*} = \sum_{k \in T} \lambda_k^T \mathbf{m}_k = \mathbf{m}_0 \in \mathcal{V}_{L,w}$, which contradicts the choice of i^*. Hence, whenever the outer fork succeeds, $\mathcal{B}_2$ outputs a vector $\mathbf{z}$ with $\mathbf{Az} = \mathbf{0}$ and $\mathbf{z} \neq \mathbf{0} \bmod q$, computed by fixed formulas from the transcripts. Its norm is at most β_{SIS} unless Bad_H occurs, a single event over the table of H shared by all runs, of probability at most ε_H; this gives the bound. Each run of $\mathcal{F}$ runs $\mathcal{A}$ twice, which gives four runs in total. □

The order of the two forks matters (Figure 6). The ballot to be rewound is known only after the aggregation proofs have been extracted, but its proof was produced earlier. The outer fork therefore reruns the whole inner algorithm. Only the ballot transcript of the second outer run is used, and it refers to the same commitments because the query input fixes them. The loss $\varepsilon'^4/(\Lambda^2 Q_a^2 Q_b)$ is that of double forking and is not tight. EVOLVE's consistency proof performs the same two extractions in the same order (del Pino et al., 2017, Thm. A.2). At the parameters of Table 3, the reduction gives no concrete guarantee. With $n = 4$, $t = 3$ and $\kappa = 4$ we have $\Lambda = \binom{4}{3} \cdot 4^3 = 2^8$. $\mathcal{B}_2$ runs $\mathcal{A}$ four times and succeeds with probability at least about $\varepsilon'^4/(\Lambda^2 Q_a^2 Q_b)$. The reduction therefore only guarantees a ratio of running time to success probability of at most about $4\Lambda^2 Q_a^2 Q_b\, t_{\mathcal{A}}/\varepsilon'^4$, where $t_{\mathcal{A}} \geq \max(Q_a, Q_b)$ is the running time of $\mathcal{A}$. For an adversary making 2^{64} (resp. 2^{40}) queries, this guaranteed ratio is at least 2^{274} (resp. 2^{178}) even for $\varepsilon' = 1$. This is far above the M-SIS estimates of 2^{134} to 2^{145} in Table 3, so no bound on ε' follows. As for EVOLVE, the parameters are therefore chosen heuristically: Theorem 14 establishes verifiability asymptotically, and Table 3 gives hardness estimates for the underlying problems, not a concrete security level of the protocol.

Where the loss comes from, and two ways to reduce it. The factor Q_a arises because the adversary chooses which query to H_{agg} becomes the forking point, for instance by trying several boards BB_1 before committing to one. If the aggregation challenges were instead derived from a public randomness beacon that is revealed only after round 1 has closed, the forking point would be the beacon output of the tally round, and Q_a would be replaced by the maximum number of tally rounds. In the verifiability experiment all authorities may be corrupted and can make any number of rounds fail, each with a fresh beacon value, so this maximum must be fixed in par as a bound on N_{rd} after which the election is declared failed; with at most four rounds the guaranteed ratio above becomes about 2^{150} for 2^{64} queries. This requires a beacon whose output nobody can predict or bias before the close of round 1, a trust assumption that the hash-based protocol does not make, and it still gives no concrete bound at the estimates of Table 3. The factor Q_b and one factor ΛQ_a are the price of the outer fork, which disappears under Theorem 5 if the ballot proof is online extractable: the reduction then loses only the inner fork, about $\varepsilon'^2/(\Lambda Q_a)$, and the guaranteed ratio becomes about 2^{137} for 2^{64} queries, which would bound ε' by about 2^{-4} at $N_V = 10^4$ if the M-SIS estimate of Table 3 were a lower bound on the time-to-advantage ratio and the extraction error were negligible; combined with a beacon the ratio would become about 2^{75}, that is, $\varepsilon' \lesssim 2^{-35}$ under the same proviso. Both options change the protocol or its proof system and are not analyzed further here; we note them because they locate the loss precisely in the two forks.

Share-soundness. The outer fork alone gives share-soundness, which Theorem 7 requires. Suppose an adversary outputs a ballot whose proof verifies and valid openings $(\hat{\mathbf{m}}_k, \hat{\mathbf{r}}_k)$ under β, $k \in T$, of its share commitments such that $\sum_{k \in T} \lambda_k^T \hat{\mathbf{m}}_k \notin \{0,1\}^L$. Forking at the H_{bal} query of its proof gives, by (ii), relaxed openings $(\mathbf{m}_j, \bar{\mathbf{r}}_j, f)$ of all its commitments that satisfy the parity-check equations, and binary relaxed openings of $\mathbf{c}_0$. We argue as in the step *Output* above. If $\mathbf{m}_k \neq \hat{\mathbf{m}}_k$ for some $k \in T$, Lemma 9 with slack 1 for $(\hat{\mathbf{m}}_k, \hat{\mathbf{r}}_k)$ gives $\mathbf{z} = \bar{\mathbf{r}}_k - f\hat{\mathbf{r}}_k$, of norm at most $\bar{B}_V + \eta_{\max}\beta \leq 2\bar{B}_V + \eta_{\max} B_A \leq \beta_{\mathrm{SIS}}$. If $\mathbf{m}_0$ differs from the binary message in some row, the second case of *Output* applies unchanged. Otherwise $\sum_{k \in T} \lambda_k^T \hat{\mathbf{m}}_k = \mathbf{m}_0 \in \{0,1\}^L$, a contradiction. If the adversary succeeds with probability ε_s, the forking lemma gives an M-SIS solution with probability at least $\varepsilon_s(\varepsilon_s/Q_b - \delta_V)$.

In EVOLVE, the voter's slack multiplies the sum of the authorities' extracted randomness, and the corresponding constraint grows with $\sqrt{N_A}$ (del Pino et al., 2017, Sec. 5). Here the comparison is made per share, and the bound β_{SIS} contains no explicit factor that comes from summing openings over the authorities. The bounds themselves do depend on n in one place: since the ballot proof applies one rejection step to all $n + 1$ responses, B_J grows like $\sqrt{(n+1+L)(n+1+2L)}$, that is, roughly linearly in n. In contrast, B_A does not depend on n, since each authority proves statements about its own share only.

7.5 Parameters

Choice of parameters. Table 3 lists the parameters. The rejection parameters are computed from Lemmas 8 and 13 with $\tau = 171 \ln 2$ for the ballot proof, $\tau = 187 \ln 2$ for the aggregation and $Q = 2^{64}$, where Q counts all evaluations of the random oracles, including offline evaluations by the adversary. The hardness estimates were obtained with the lattice estimator (Albrecht et al., 2015), commit 53da598, using the MATZOV cost model (MATZOV, 2022) and the Euclidean norm for M-SIS, with secret and error D_1 and $m = (d + L)N$ samples for M-LWE, and $m = (2d + L)N$ columns for M-SIS. For each N_V we took the smallest d and, for it, the largest prime $q \equiv 17 \pmod{32}$ below the smallest power of two for which both estimates reach 2^{128} and $q > \beta_{\mathrm{SIS}}$; the output for every attack is published with the implementation (Appendix C). The target of 2^{128} refers to the MATZOV model, which costs sieving with the classical nearest-neighbour search of Albrecht et al. (2020) and dimensions for free; at $N_V = 10^6$ the M-LWE estimate is only 0.9 bit above it. Table 3 gives the same estimates under three further models, computed with the same tool and the same attacks: the MATZOV model with the quantum nearest-neighbour cost of Albrecht et al. (2020) in the depth-times-width metric, which lowers the estimates by 4 to 7 bits, so that the sets for 10^4 and 10^5 stay above 2^{128} and the set for 10^6 falls 3.3 bits below it; and the core-SVP model of Alkim et al. (2016), which counts $2^{0.292\beta}$ (classical) or $2^{0.265\beta}$ (quantum) for one sieve call in dimension β and nothing else, the metric of the NIST submissions. For comparison, the same tool and models give $2^{139.7}$, $2^{134.4}$, $2^{115.5}$ and $2^{106.3}$ for ML-KEM-512 (the parameters of Kyber-512, whose specification reports core-SVP hardness 118 and 107 from the primal attack alone (Avanzi et al., 2021)). The set for $N_V = 10^4$ is thus within 1.3 bits of ML-KEM-512 under both MATZOV models and 5 to 6 bits below it under core-SVP; the set for 10^6 is 10 to 15 bits below it. For $N_V = 10^6$ we therefore also give the set with $d = 8$ at the same q (last column of Table 3): its estimates stay above 2^{128} under both MATZOV models, with M-LWE at $2^{148.5}$ and $2^{142.5}$, and its core-SVP estimates ($2^{121.1}$ and $2^{110.2}$) exceed those of ML-KEM-512, at the cost of a ballot of 128.1 KB instead of 112.6 KB and of 33.7 KB instead of 29.7 KB per ballot and authority. We recommend $d = 8$ for elections of this size; the measurements of Section 8 and the comparisons with EVOLVE refer to $d = 7$. The scripts and the raw estimator output for all four models are published with the implementation (Appendix C). In β_{SIS}, $\eta_{\max} = 120$ and the honest bound is $\beta = 2\sigma\sqrt{N(2d + L)} \approx 124$. For $n = 5$, $\sigma_J = 2^{15.0}$, $B_J = 2^{23.5}$ and $\bar{B}_V = 2^{24.5}$, and β_{SIS} is unchanged. For one out of two candidates ($L = 2$, $w = 1$) the same d and q suffice: $\sigma_J = 2^{15.2}$, $B_J = 2^{23.8}$, β_{SIS} is larger by at most 0.1 bit, the binding estimates are $2^{144.0}$, $2^{137.6}$ and $2^{134.0}$, and the hiding estimates are those for $L = 1$, $2^{138.4}$, $2^{135.0}$ and $2^{128.9}$, since the additional samples do not improve the best attack (`results/params_L2_hiding.jsonl`). The sizes are those of the encoding of the implementation (Section 8): elements of $\mathbb{Z}_q$ in $\lceil \log_2 q \rceil$ bits and Gaussian coefficients Golomb–Rice coded; for $N_V = 10^4$ they coincide with the measured encodings. The seed ciphertexts are counted with their 32-byte payload only; encryption overhead, signatures, keys and the evidence attached to complaints are not included (see Section 8).

Invertibility and the choice of q. The argument requires that differences of challenges, which have infinity norm at most 2, are invertible in R_q. By (Lyubashevsky & Seiler, 2018, Cor. 1.2), if $X^N + 1$ splits into s factors modulo q with $q \equiv 2s + 1 \pmod{4s}$, every nonzero y with $\|y\|_\infty < q^{1/s}/\sqrt{s}$ is invertible. For $N = 256$ and $s = 8$, i.e. $q \equiv 17 \pmod{32}$, which is the condition of EVOLVE's Lemma 2.8, this radius is about 5.2 for $q \approx 2^{31}$ and about 13.4 for $q \approx 2^{42}$. For $s = 16$ and $q \approx 2^{31}$ it is below 1. The modulus of EVOLVE's parameter set, $q = 2^{31} - 2^7 - 2^5 + 1$, satisfies $q \equiv 1 \pmod{32}$ and gives $s = 16$, so the invertibility of challenge differences for that modulus is not covered by the cited lemma. We therefore use a modulus with $s = 8$. The size of the modulus is determined by the binding bound below; since B_A grows with the number of blocks, so does q. For $N_V = 10^4$, 10^5 and 10^6 we use module rank 7 and the primes $q = 2^{42} - 143$, $2^{43} - 175$ and $2^{45} - 591$ (Table 3). All satisfy $q \equiv 17 \pmod{32}$, and each is the largest such prime below a power of two, so that an element of $\mathbb{Z}_q$ takes exactly $\lceil \log_2 q \rceil$

bits. The invertibility radius $q^{1/8}/\sqrt{8}$ then exceeds 13, well above 2. The price of $s = 8$ is one fewer level of the partial NTT (three stages instead of four).

Remark on the opening bound. The consistency argument of EVOLVE requires extracted openings to be valid openings of the commitment scheme. This yields lower bounds on the opening bound B_r (del Pino et al., 2017, Sec. 5), resp. B_{Com} in the revised version (del Pino, 2018, Sec. 5.5). For the proposed parameter sets, these bounds evaluate to about $2^{34.3}$, resp. $2^{35.8}$, which exceed $q \approx 2^{31}$. The kernel of $\mathbf{A}$ contains vectors that are nonzero modulo q and have norm about $q\sqrt{dN/12}$, and such vectors can be found by linear algebra alone. An opening bound of this size therefore admits openings of one commitment to two different messages (Appendix A). The construction of EVOLVE is not affected, since the constraints can be met with a larger modulus; our estimates give $\log q \approx 36$. The same requirement applies to our instantiation, and when we compare costs we use parameters that satisfy the constraints of both schemes.

8 Implementation and measurements

We implemented T-EVOLVE and, on the same code base, the ballot of EVOLVE (Alizadeh, 2026). The implementation is written in C, single-threaded, and consists of SHAKE, arithmetic in R_q for primes $q \equiv 17 \pmod{32}$ below 2^{50} with the three-level partial NTT, discrete Gaussian samplers that use integer arithmetic only, so that an authority re-derives exactly the voter's randomness from a seed on any platform, BDLOP commitments, Shamir sharing and the parity-check matrix, the ballot proof of Section 7, the aggregation tree with Π_{open} and Π_{zero}, the two-round tally, and an encoding (elements of $\mathbb{Z}_q$ packed, Gaussian coefficients Golomb–Rice coded) from which all sizes are measured. The same code computes every quantity of Lemmas 8 and 13 and of Theorem 14, and a script searches the parameters of Table 3 with the lattice estimator. The tests check SHAKE against known answers, the NTT against schoolbook multiplication, the Gaussian samplers by chi-square tests, completeness, the rejection of nine kinds of tampered ballots and seven kinds of tampered aggregation proofs, and a complete tally. Three components of the protocol are not implemented and are excluded from all sizes and times: the encryption of the seeds with publicly verifiable opening, which is replaced by a placeholder that fixes the data flow and the 32-byte payload; the signatures of the voters; and the evidence that authorities attach to complaints. Measured sizes therefore count each ciphertext as its 32-byte payload and contain no signature; with the ciphertexts of Appendix D, 9792 bytes each in place of the 32-byte payloads, a ballot grows by 38.1 KB for $n = 4$ (Table 3). As an estimate, not a measurement, an ML-DSA-44 signature adds 2.4 KB. The ciphertexts depend on the instantiation of EO (Section 7): with ML-KEM-768 (a 1088-byte encapsulation) and an authenticated encryption of the seed with the identity as associated data (16 bytes of tag), as in PQKryvos, each ciphertext grows by about 1.1 KB, and a yes/no ballot for $n = 4$ grows by about 4.3 KB for the four ciphertexts; with an ML-DSA-44 signature it grows by 6.7 KB, from 108.9 to about 115.6 KB (here and below, 1 KB is 1024 bytes), but this instantiation lacks proofs for ciphertexts that do not decrypt; with release of identity-based keys at the parameters of Appendix D, each ciphertext takes 9.6 KB. Constant-time code is also outside the scope of the implementation. The proofs refer to the ideal distributions, and the samplers approximate them with a bounded error. Their statistical distance from the ideal distributions is below 2^{-251} per coefficient for $\sigma = 1$ and below 2^{-216} per coefficient of a mask; Appendix C gives the construction and the resulting errors per attempt.

Exact tests and error budget. The rejection steps and norm checks are computed exactly in integer arithmetic, up to a cap on the counter of the Bernoulli sampler whose deviation is below $2^{-2^{23}}$ per call, and the statistical errors of all simulations sum to less than $2^{-130.3}$ over 2^{40} attempts (Appendix C, Table 7). This bounds the statistical distance of the simulations only; it does not by itself give 128-bit security of the protocol, which also rests on computational assumptions and on the non-tight reduction of Theorem 14.

Table 4 gives the measurements on the reference machine. Creating a ballot takes a few tens

Ballots ($d = 7$, $q = 2^{42} - 143$; medians over 30 runs)					
		Vote	Verify	Check_k	Size
EVOLVE	$n = 4$, yes/no	12.5 (9.1–21.1) ms	2.9 ms	—	58.6 KB
T-EVOLVE	$n = 4$, $t = 3$, yes/no	28.7 (17.8–30.1) ms	9.3 ms	1.2 ms	108.9 KB
T-EVOLVE	$n = 4$, $t = 3$, 1 of 2	39.8 (23.5–57.4) ms	12.9 ms	1.5 ms	138.0 KB
T-EVOLVE	$n = 5$, $t = 3$, yes/no	47.2 (33.5–77.1) ms	10.8 ms	1.3 ms	128.4 KB
Aggregation, per authority ($n = 4$, $t = 3$, yes/no)					
N_V	E	Round 1	Round 2	VerAgg	Published per ballot
10^3	6	25 s	8 s	5 s	47.4 KB
10^4	24	136 s	36 s	30 s	28.2 KB

Table 4: Measurements on the reference machine: one core of an Intel Xeon (KVM virtual machine with 2 virtual CPUs and 8 GB of memory, reported frequency 2.1 GHz, Ubuntu 24.04, gcc 13.3, `-O3 -march=native`), with no other load. Vote includes all attempts of the ballot proof, whose number is random ($M \approx 3$ on average), hence the wide interquartile range given in parentheses; in these 30 runs per row the mean number of attempts was 2.47, 2.07, 2.40 and 3.30, in the order of the rows, against 2.96 over the 10,000 ballots of the experiment in Section 7.2; sizes are medians of the measured encodings, which agree with Table 3 to within 0.1 KB; Verify includes decoding. The encryption of the seeds, signatures and the evidence of complaints are not implemented and not included. Round 1 builds the tree and prepares $\kappa = 4$ attempts; round 2 searches for a non-aborting attempt. The aggregation rows are single runs in which the first tally round succeeded; the three runs at $N_V = 10^4$ on the second machine (Table 5) differ by less than 3%. Commits and logs: Appendix C.

of milliseconds and is dominated by the products with $\mathbf{C}$ of the masks and of the OR-branches; its time varies with the number of attempts. Verifying takes about 10 ms. The work of an authority is dominated by the first messages of the amortized proofs: each attempt computes $\mathbf{AY}_1$ and $\mathbf{AY}_2$ for every block, so round 1 costs κ such computations, and round 2 one computation of the responses per attempt until an attempt does not abort. For 10^4 voters an authority needs about three minutes on one core for one successful tally round ($136 + 36$ s); the expected number of rounds is between 1.2 and 1.9 (Section 7.3). At the same parameters, a ballot of T-EVOLVE is 1.86 times as large as one of EVOLVE. The fastest runs of Vote, which need a single attempt, take 16.8 against 8.0 ms, 2.1 times as long, and verification takes 3.2 times as long: the ballot carries one more commitment and a proof that answers for all $n + 1$ of them, whereas the OR-proof of EVOLVE answers for one commitment, the sum of the shares. We implemented only the ballot of EVOLVE, with the samplers, the rejection step and the parameters of T-EVOLVE, so that the two ballots differ only in the protocol. The aggregation of T-EVOLVE follows the tree aggregation of EVOLVE with two amortized proofs, for the openings and for the zero relations of the tree, and adds the signed second challenge and the committed first messages; we did not implement EVOLVE's aggregation. In EVOLVE all n authorities must aggregate, in T-EVOLVE any t of them.

Scaling. We measured how the costs grow with n, L and N_V on a second machine, a laptop with an Intel Core i9-14900HX (Windows 11, gcc 15.2 of MinGW-w64, `-O3 -march=native`, one thread, 15.6 GB of memory), with the same protocol code (Table 5). The aggregation was run in a mode that processes the authorities of the quorum one after another, which bounds the memory by that of one authority; the work of each authority is the same as in the normal run. A ballot grows by about 19 KB per authority, since each authority adds one commitment and one response, and by about 29 to 45 KB per candidate, since each candidate adds one message row to every commitment, one column to every randomness vector and hence to every response, and one OR-proof; the cost per candidate grows with n and L, since the coefficients of the responses grow with $\sigma_J \propto \sqrt{n + 1 + L}$. For $L = 10$, the bound $\beta_{\mathrm{SIS}} = 2^{42.1}$ at $N_V = 10^4$ exceeds 2^{42}, so the rule of Section 7.5 selects $q = 2^{43} - 175$; the lattice estimator, run as for Table 3, gives $2^{135.0}$ for hiding and $2^{144.0}$ for binding. The times of Vote and Verify grow roughly linearly in n and L. For the aggregation, the cost per block is the same at all sizes, 0.90 to 0.93 s per block and

Ballots ($d = 7$; medians over 30 runs)				
	Size	Vote	Verify	Check_k
$n = 4$, yes/no	108.9 KB	17.3 ms	6.6 ms	0.8 ms
$n = 5$, yes/no	128.4 KB	23.6 ms	7.7 ms	0.8 ms
$n = 7$, yes/no	166.0 KB	28.7 ms	9.7 ms	0.8 ms
$n = 10$, yes/no	222.6 KB	39.7 ms	12.5 ms	0.8 ms
$n = 15$, yes/no	317.0 KB	55.4 ms	17.8 ms	0.8 ms
$n = 4$, 1 of 2	138.0 KB	27.1 ms	9.4 ms	1.0 ms
$n = 4$, 1 of 5	237.8 KB	52.3 ms	19.0 ms	1.5 ms
$n = 4$, 1 of 10†	460.5 KB	104.8 ms	38.8 ms	2.5 ms
$n = 10$, 1 of 2	262.6 KB	65.9 ms	16.5 ms	1.0 ms
Aggregation, per authority ($n = 4$, $t = 3$, yes/no, $q = 2^{42} - 143$)				
N_V (E)	Round 1	Round 2‡	VerAgg	Per ballot
10^4 (24)§	87.2 s	12.0 s	20.7 s	28.2 KB
$5 \cdot 10^4$ (111)	408 s	57 s	99 s	28.1 KB
10^5 (217)	810 s	113 s	196 s	28.3 KB
10^6 (2142)*	8000 s	1117 s	1938 s	29.7 KB

Table 5: Scaling on the second machine (Intel Core i9-14900HX laptop, Windows 11, gcc 15.2, one thread). Thresholds: $t = 3, 3, 4, 6, 8$ for $n = 4, 5, 7, 10, 15$. Ballots use $q = 2^{42} - 143$, the modulus of Table 3 for $N_V = 10^4$, whose bound β_{SIS} is determined by the aggregation and does not depend on n; the ballot marked † ($L = 10$) needs $q = 2^{43} - 175$. ‡Per attempt, whereas Table 4 gives the whole round; on the reference machine one attempt took 24.2 s at $N_V = 10^4$. §Mean of three runs, which ranged from 86.7 to 87.6 s (round 1), 11.9 to 12.2 s (round 2) and 20.6 to 20.8 s (VerAgg). All aggregation runs use $q = 2^{42} - 143$; for $N_V = 5 \cdot 10^4$ and 10^5 this modulus is below β_{SIS} ($2^{42.50}$ and $2^{42.92}$), so these runs do not use the moduli of Table 3; the sizes in these two rows are measured at $q = 2^{42} - 143$. All arithmetic is done in 64-bit words for every $q < 2^{50}$, so the running times do not depend on q; the extrapolation to $N_V = 10^6$ assumes this. *Per ballot*: published per ballot by one authority. Sizes are measured encodings and agree with those computed from the parameters to within 0.1 KB. The row marked * is extrapolated from the per-block costs measured at $N_V = 10^5$, and its size is computed from the encoding with the parameters for $N_V = 10^6$. The encryption of the seeds, signatures and the evidence of complaints are not included. Logs: Appendix C.

attempt in round 1, 0.50 to 0.52 s in round 2 and 0.86 to 0.91 s for VerAgg, so the cost grows linearly in the number of blocks E. We therefore extrapolate $N_V = 10^6$ from the per-block costs measured at $N_V = 10^5$: about 2.2 hours for round 1, 19 minutes per attempt of round 2 and 32 minutes for VerAgg, per authority and on one core. The blocks are independent, so the work parallelizes over cores; we did not measure this. In all five aggregation runs, every authority had a non-aborting attempt in the first tally round. These runs are not independent samples of the number of attempts: the random coins of the rejection steps are derived from the seed of the run, and runs with the same seed use the same coins, as do the aggregation runs of Table 4. Only the three runs at $N_V = 10^4$, with seeds 1, 2 and 3, are independent; their 9 authorities used 16 attempts. The success probability of an attempt follows from the analysis of the rejection steps (Section 7.3), not from these runs. On the same laptop, the core tests produce the same output as on the reference machine, and a ballot experiment of 5000 ballots gave the same numbers of attempts, norms, sizes and verification results for every ballot; so an authority re-derives exactly the randomness that a voter computed on another platform. In the scaling runs, the times on the laptop are 0.50 to 0.73 of those of Table 4 for the same settings; in the cross-platform run (Appendix C) they were 0.47 to 0.77.

Cost of public verification. Everyone can verify an election, but the data is large. For $N_V = 10^6$, the ballots take about 115 GB (112.6 KB each, Table 3; here 1 GB is 10^9 bytes), plus the ciphertexts of the seeds and the signatures, and the contributions of a quorum of $t = 3$ authorities, $3 \cdot 29.7$ KB per ballot, about 91 GB, plus the first-round publications of the other authorities, about 0.4 KB per ballot each. On one core of the laptop, verifying the ballots takes about $10^6 \cdot 6.6$ ms, that is 1.8 hours, and verifying the t contributions about $3 \cdot 32$ minutes;

the time of Verify was measured at the parameters for $N_V = 10^4$ (Table 5). Recomputing Acc also requires checking the posted complaints with Judge; their number is small unless voters or authorities misbehave, and their cost depends on the instantiation of EO, so it is not included. Each authority also checks its share of every ballot, about 0.8 to 1.2 ms per ballot excluding decryption, measured on the two machines. All of these checks are independent per ballot or per block and parallelize. For $N_V = 10^4$ the board takes about 2 GB, and the laptop verifies it in about two minutes.

9 Conclusion

T-EVOLVE transfers the threshold property of the protocol of Cramer et al. (1996) to lattice commitments: any t of n authorities that took part in the checking phase complete the tally, privacy holds against $t - 1$ of them, and the tally remains verifiable if all of them collude. Compared with EVOLVE, a ballot carries $n + 1$ commitments instead of n, and its proof answers for all of them; the Reed–Solomon relation itself costs no communication. An authority runs the two amortized proofs over its aggregation tree as in EVOLVE, with a signed second challenge and first messages committed in a first round, and only t authorities need to be online for tallying. Table 6 compares the protocols. For the ballot, we compare T-EVOLVE and EVOLVE at identical parameters, with the same implementation and the same encoding (Section 8). With four authorities in the yes/no setting and the parameters of Table 3 for $N_V = 10^4$, EVOLVE's ballot takes 58.6 KB and ours 108.9 KB, a factor of 1.86; creating ours takes 2.1 times as long in the fastest runs and 2.3 times as long in the median, and verifying it takes 3.2 times as long. This comparison isolates the cost of the threshold in the commitments and proofs: both ballots are produced by the same code with seed-derived randomness and n seed ciphertexts counted as 32-byte payloads, whereas EVOLVE as published encrypts the commitment randomness itself, so its encryption part would be larger. EVOLVE alone could use smaller parameters, since its authorities need no joint rejection over a tree of our size. The smallest parameters that satisfied EVOLVE's constraints in our earlier estimates are $d = 6$ and $q \approx 2^{36}$; with them ($q = 2^{36} - 303$), its ballot takes 45.9 KB in our encoding, as measured, and ours is 2.37 times as large. A comparison with the 78 KB reported by EVOLVE would mix two different size estimates, and that figure refers to parameter sets that do not satisfy EVOLVE's constraints (Section 7.5). An authority of T-EVOLVE publishes about 28 KB per ballot for 10^4 voters; EVOLVE reports 18 KB for a different amortized proof, for which we have no common implementation. All these comparisons hold only for the parameter sets and the machine above. In exchange, the tally survives the absence or misbehavior of $n - t$ authorities. The threshold also changes the privacy guarantee: for $n = 4$ and $t = 3$, privacy holds against two corrupted authorities, whereas EVOLVE tolerates three; this is the price of tolerating one absent authority.

All security proofs are in the classical random oracle model: the verifiability arguments rewind the adversary with the forking lemma, and the zero-knowledge simulators program the random oracle. We do not analyze adversaries with quantum access to the random oracle. The post-quantum claim therefore rests on the hardness of M-LWE and M-SIS against quantum algorithms, as for EVOLVE and (Baum & Lyubashevsky, 2017), whose authors treat the quantum case heuristically; security in the quantum random oracle model remains open. Two steps of our proofs stand in the way. First, the verifiability reduction rewinds the adversary twice, in a nested way, and extracts from many proofs at once; the measure-and-reprogram technique of Don et al. (2019) gives extraction from Fiat–Shamir proofs in the quantum random oracle model, but we do not know how to apply it to nested forks over many proofs with a single rewinding. Second, the zero-knowledge simulators program the random oracles, and the analysis of Fiat–Shamir with aborts in the quantum random oracle model (Kiltz et al., 2018; Barbosa et al., 2023) would have to be extended to our joint rejection steps and to the committed first messages of the aggregation. The reduction of Theorem 14 is not tight. The estimates of Table 3 concern the

	Privacy	Tally	Verifiable, all collude	Honest ballot removable	Ballot formats	Ballot (yes/no)	Authority per ballot
EVOLVE (del Pino et al., 2017)	$n-1$	all n	yes	yes (Boyen et al., 2021)	yes/no	58.6 KB† (45.9)	18 KB
PQKryvos (Huber et al., 2026)	n_t-1	all n_t	yes	no (ZK proof of decryption)	flexible, tally-hiding	comm. + Ligero	proof of result
Based on (Rivinius et al., 2022)	threshold	threshold	—	—	via MPC	—	—
Bonte et al. (2025)	threshold	threshold	no	—	complex (FHE)	—	—
T-EVOLVE (this work)	$t-1$	any t of n that took part in the checking phase	yes	no (EO)	yes/no, w-out-of-L	109 KB* (147)	28 KB

Table 6: Comparison with related protocols. "—": not reported or not applicable. *With the n seed ciphertexts counted as 32-byte payloads; in parentheses, with the 9.6 KB ciphertexts of the instantiation of Appendix D. †EVOLVE's ballot encoded by our implementation at the parameters of T-EVOLVE for $N_V = 10^4$, measured (Table 4), and, in parentheses, at $d = 6$ and $q = 2^{36} - 303$, the smallest parameters that satisfied EVOLVE's constraints in our estimates (Section 9); EVOLVE itself reports 78 KB (del Pino et al., 2017, Tab. 1) for parameter sets that do not satisfy its constraints (Section 7.5). Its 18 KB per ballot per authority is as reported, for a different amortized proof. Running times are in Table 4; PQKryvos reports about 1.5 s to prove ballot validity for a yes/no election and proof sizes from several kilobytes up to about 1 MB depending on the circuit, on a different machine and implementation. For PQKryvos, n_t is its number of talliers; its complaints publish the plaintext with a zero-knowledge proof of correct decryption (Section 7). That a dishonest authority can have an honest ballot removed in EVOLVE was pointed out by Boyen et al. (2021). The protocol of (Rivinius et al., 2022) is a general MPC protocol whose post-quantum security is left to future work by its authors; voting costs are not reported. The tally of Bonte et al. (2025) is computed with threshold FHE and is not publicly verifiable. Protocols based on mix-nets (Boyen et al., 2020; Aranha et al., 2021, 2023; Hough et al., 2025) follow a different tallying paradigm and are not listed. Our sizes are those of Table 3 for $N_V = 10^4$ in the yes/no setting with $n = 4$ authorities, the number used by EVOLVE, and $t = 3$; for a choice of one out of two candidates the ballot takes 138 KB. The 109 KB and 138 KB count the seed ciphertexts as 32-byte payloads; signatures are excluded throughout (Section 8).

hardness of M-LWE and M-SIS for the stated parameters, not the protocol itself; turning them into a statement about the protocol requires a tighter reduction, which remains open. The remaining work comprises four items: an implementation of the encryption of Appendix D and of the complaint path, whose parameters are fixed but whose running times are not measured; concrete parameters for the general-purpose instantiation; a comparison of running times with PQKryvos and with voting based on (Rivinius et al., 2022); and, for $N_V = 10^6$, measurements at $d = 8$, the rank at which the M-LWE estimate stays above 2^{128} also under the quantum cost model of Table 3. Two further directions concern the proofs. The rejection steps of the aggregation, and with them the κ attempts and the repeated tally rounds, might be avoided with the rejection-free technique of Griffy et al. (2026), which widens the masks and relies on the Hint-MLWE assumption; whether their analysis carries over to the two amortized proofs over a tree is open. For many candidates, the L message rows could be packed into the coefficients of one ring element and binarity shown with a quadratic proof in the framework of Lyubashevsky et al. (2022); the $n + 1$ commitments of a ballot would then take about 54 KB instead of 114 KB for $L = 10$ at $q = 2^{43}$, and the L OR-proofs would be replaced by one proof, whose size and whose interaction with the aggregation we have not worked out.

Summary. Table 1 lists the proved properties with their assumptions, what Table 3 covers and what is measured. The concrete estimates rest on three assumptions: that the lattice estimator reflects the hardness of M-LWE and M-SIS for the chosen (d, q), that the adversary makes at most 2^{64} random-oracle queries, and that at most 2^{40} proof attempts are made in one election; the statistical error budget of the implementation is below 2^{-128} over these attempts (Appendix C).

A Openings beyond the modulus

The constraints of EVOLVE. Section 5 of (del Pino et al., 2017) lists the bounds that correctness and security impose on the parameters. In its notation (n is the ring dimension, d the module rank, N_A the number of authorities and l the fan-in of the aggregation tree), they read:

- correctness and zero knowledge of π^V: $B_{OR} \geq 2N_A\sqrt{n(2d+1)}\,\sigma$ and $B'_{OR} \geq 44\sqrt{60n(2d+1)}\,B_{OR}$;
- correctness and zero knowledge of π^A: $B_{Amo,1} \geq 2\sqrt{n(2d+1)}\,\sigma$ and $B'_{Amo,1} \geq 2684n\sqrt{2d+1}\,B_{Amo,1}$;
- correctness and zero knowledge of $\pi^{A'}$: $B_{Amo,2} \geq 2(l+1)\sqrt{n(2d+1)}\,\sigma$ and $B'_{Amo,2} \geq 2684n\sqrt{2d+1}\,B_{Amo,2}$;
- consistency of the vote, from equations (3), (4), (7) and (8) of its Appendix A: $2B'_{OR} \leq B_r$, $2\sqrt{60}\,N_A B'_{Amo,1} \leq B_r$, $(l+1)B'_{Amo,1} \leq B_r$ and $B'_{Amo,2} \leq B_r$.

All norms are Euclidean, as in (del Pino et al., 2017, Sec. 2.1); its Section 3.1 defines the opening space of the commitment by $\|\mathbf{r}\| \leq B_r$ and reduces binding to M-SIS with bound $2B_r$. Here B_r is the opening bound of the commitment scheme: the consistency proof extracts openings and needs them to be valid openings, of norm at most B_r. The inequalities above are transcribed from Section 5 and Appendix A of (del Pino et al., 2017) without change of constants. With the parameters of its Table 2 ($n = 256$, $q = 2^{31} - 2^7 - 2^5 + 1$, $d = 7$, $\sigma = 1$, $N_A = 4$, $l = 30$), we have $\sqrt{n(2d+1)} = \sqrt{3840}$, and the smallest admissible values are

$$B_{OR} = 2^{8.95}, \quad B'_{OR} = 2^{23.32}, \quad B_{Amo,1} = 2^{6.95}, \quad B'_{Amo,1} = 2^{28.30},$$

$$B_{Amo,2} = 2^{11.91}, \quad B'_{Amo,2} = 2^{33.25}.$$

The four consistency constraints then require $B_r \geq 2^{24.32}$, $2^{34.25}$, $2^{33.25}$ and $2^{33.25}$, respectively, so $B_r \geq 2^{34.25} > q \approx 2^{31}$. For binding, the difference of two openings must then be a solution of M-SIS with bound $2B_r = 2^{35.25}$, which exceeds q, and such solutions exist and are easy to find, as the construction below shows.

Two openings of one commitment. Let $\mathbf{A} \in R_q^{d \times \mu}$ with $\mu = 2d + L$, let $\mathbf{B} \in R_q^{1 \times \mu}$, and view $\mathbf{A}$ as a $dN \times \mu N$ matrix over $\mathbb{Z}_q$. Write $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_2]$ with $\mathbf{A}_1$ square, which is invertible with high probability for random $\mathbf{A}$. Choose $\mathbf{z}_2 = \mathbf{e}_1$ and $\mathbf{z}_1 = -\mathbf{A}_1^{-1}\mathbf{A}_2\mathbf{z}_2 \bmod q$ with centered coefficients. Then $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$ satisfies $\mathbf{A}\mathbf{z} = \mathbf{0} \bmod q$ and $\mathbf{z} \neq \mathbf{0} \bmod q$. For random $\mathbf{B}$, also $\mathbf{B}\mathbf{z} \neq \mathbf{0}$ with high probability. The coefficients of $\mathbf{z}_1$ are close to uniform in $(-q/2, q/2]$, so $\|\mathbf{z}\|_2 \approx q\sqrt{dN/12}$. Let $\mathbf{r} = \lfloor \mathbf{z}/2 \rceil$ and $\mathbf{r}' = \mathbf{r} - \mathbf{z}$, so that $\|\mathbf{r}\|_2, \|\mathbf{r}'\|_2 \approx \|\mathbf{z}\|_2/2$. For any message m, the commitment $\mathbf{c} = \mathbf{C}\mathbf{r} + \binom{\mathbf{0}}{m}$ is also opened by $\mathbf{r}'$, to the message $m + \mathbf{B}\mathbf{z} \neq m$, since $\mathbf{C}\mathbf{z} = \binom{\mathbf{0}}{\mathbf{B}\mathbf{z}}$. Whenever the opening bound exceeds $\|\mathbf{z}\|_2/2$, both openings are therefore valid.

We ran this construction for $N = 256$ and $q = 2^{31} - 2^7 - 2^5 + 1$, with a random key of the shape of EVOLVE (one message row, width $2d+1$). For module rank $d = 7$ (the CCS parameters) we obtain $\|\mathbf{z}\|_2 = 2^{34.58}$, against the expected $q\sqrt{dN/12} = 2^{34.61}$, and $\|\mathbf{r}\|_2 = \|\mathbf{r}'\|_2 = 2^{33.58}$, below the required $B_r \geq 2^{34.25}$; so a commitment has two valid openings to different messages. The script `scripts/evolve_opening_check.py` of (Alizadeh, 2026) evaluates the constraints above and performs this construction; its output is in `results/evolve_opening_check.txt`.

For $d = 5$ (the parameters of (del Pino, 2018)) we obtain $\|\mathbf{r}\|_2 = \|\mathbf{r}'\|_2 = 2^{33.33}$, compared with the bound $B_{Com} \geq 2^{35.8}$ that the constraints of (del Pino, 2018, Sec. 5.5) give for these parameters; the output for $d = 5$ is in the same file.

B Detailed proofs

This appendix gives the proof of Lemma 8, the freshness and simulation arguments of the ballot proof, and self-contained proofs of the two lemmas that are new in the aggregation, Lemmas 11 and 13, with the places where they are used.

Proof of Lemma 8. D_σ is σ-subgaussian, i.e. $\mathbb{E}[e^{uX}] \leq e^{\sigma^2 u^2/2}$ for all real u (Micciancio & Peikert, 2012, Lemma 2.8). For fixed challenges, $\mathbf{s} = \Phi\mathbf{r}$ for a linear map Φ applied to the concatenation $\mathbf{r}$ of the $\mathbf{r}_j$. With $\Sigma = \Phi^\top\Phi$, the tail bound for quadratic forms of subgaussian vectors (Hsu et al., 2012, Thm. 2.1) gives $\|\Phi\mathbf{r}\|_2^2 \leq \sigma^2(\operatorname{tr}\Sigma + 2\sqrt{\operatorname{tr}(\Sigma^2)\tau} + 2\|\Sigma\|\tau)$ except with probability $e^{-\tau}$. Multiplication by an element of $\mathcal{S}$ has squared Frobenius norm $60N$ and operator norm at most $\|c\|_1 = 60$ on each ring element. The vector $\mathbf{r}_0$ enters $1 + L$ blocks of Φ and every other $\mathbf{r}_j$ one block, so $\operatorname{tr}\Sigma = t_s$ and $\|\Sigma\| \leq \xi_s$; finally $\operatorname{tr}(\Sigma^2) \leq \|\Sigma\| \operatorname{tr}\Sigma$. □

Freshness of the ballot challenges. The challenge c and the permutations of the OR-proof are outputs of H_{bal} on an input that contains the first messages, among them $\mathbf{w} = \mathbf{A}\mathbf{y}$ for a fresh mask $\mathbf{y}$ from $D_{\sigma_J}^{N(2d+L)}$; the simulated challenges $f_{a,1-m}$ are sampled uniformly by the prover, and $f_{a,m}$ is obtained from $f_{a,1-m}$ by the permutation. Let $\mathsf{BadFresh}_{\mathsf{bal}}$ be the event that this input was queried before. Write $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_2]$ with $\mathbf{A}_1 \in R_q^{d\times d}$. Since R_q is a product of 8 fields of size q^{32}, $\mathbf{A}_1$ is singular with probability at most $16q^{-32} < 2^{-1300}$ over the choice of $\mathbf{A}$, once for the whole experiment. If $\mathbf{A}_1$ is invertible, then for fixed $\mathbf{w}$ and fixed last $d+L$ components of $\mathbf{y}$, the equation $\mathbf{A}\mathbf{y} = \mathbf{w}$ determines the first d components modulo q. For a sample x of D_σ, the discrete Gaussian with weights $\rho_\sigma(x) = \exp(-x^2/(2\sigma^2))$, let $p_q(\sigma) = \max_{a\in\mathbb{Z}_q} \Pr[x \equiv a \pmod q]$. By the Poisson summation formula $\rho_\sigma(\mathbb{Z}) \geq \sqrt{2\pi}\,\sigma$, and since ρ_σ is symmetric and unimodal with maximum 1 at 0, its sum over a class $a + q\mathbb{Z}$ is at most $1 + \sqrt{2\pi}\,\sigma/q$; hence $p_q(\sigma) \leq 1/q + 1/(\sqrt{2\pi}\,\sigma)$, which is at most $1/\sigma$ when $q \geq 2\sigma$. This condition holds with a large margin ($q > 2^{41}$, $\sigma_J < 2^{17}$), so $\Pr[\mathbf{A}\mathbf{y} = \mathbf{w}] \leq p_q(\sigma_J)^{dN} \leq \sigma_J^{-dN}$. The mask is used nowhere before this query, so every earlier query is independent of it, and $\Pr[\mathsf{BadFresh}_{\mathsf{bal}}] \leq q_{\mathsf{bal}}\sigma_J^{-dN} < 2^{-26000}$ per attempt, where $q_{\mathsf{bal}} \leq 2^{64}$ counts the queries to H_{bal} of all parties. Outside this event the challenges are uniform and independent of $\mathbf{r}_0, \ldots, \mathbf{r}_n$.

Simulation of the ballot proof. By the rejection argument after Lemma 8, the simulator of the privacy proof samples the responses from D_{σ_J}, samples the simulated OR-branches as the prover does, recomputes the first messages and programs H_{bal}. Programming fails only if the programmed input, which contains $\mathbf{A}\mathbf{z}_j - c\mathbf{c}_{j,1}$ for a fresh sample $\mathbf{z}_j$ of D_{σ_J} (the same argument modulo q applies, since $c\mathbf{c}_{j,1}$ is fixed given c), was queried before, and the queries of attempts that the real prover rejects, which the simulator does not make, are hit by the adversary only if it queries their inputs; by the argument above each has probability at most $q_{\mathsf{bal}}\sigma_J^{-dN}$. The error of the simulator is therefore at most $e^{-r^2/2}/M + 2^{-171} + 3q_{\mathsf{bal}}\sigma_J^{-dN}$ per attempt of every honest ballot, plus 2^{-1300} once, and this is the ballot-proof part of $\varepsilon_{\mathrm{zk}}$ in Theorem 3; Section 8 and Appendix C sum all such errors over the experiment.

The protocol of Baum & Lyubashevsky (2017) for one block. The statement is $\mathbf{U} = \mathbf{A}\mathbf{S}$ over R_q, with ψ witness columns. The prover samples $\mathbf{Y}_1$ with ψ columns and $\mathbf{Y}_2$ with ℓ columns, with coefficients from D_{σ_1} and D_{σ_2}, and sends $\mathbf{W}_i = \mathbf{A}\mathbf{Y}_i$. On a first challenge $c_1 \in \mathcal{S}$ it computes $\mathbf{Z}_1 = c_1\mathbf{S} + \mathbf{Y}_1$, where $c_1\mathbf{S}$ multiplies every ring element of $\mathbf{S}$ by c_1, then $C_2 = H(\mathbf{Z}_1, c_1) \in \{-1, 1\}^{\psi\times\ell}$ (the implementation adds the digest of BB_1 and (k, ι, e) to the input, which does not affect the arguments) and $\mathbf{Z}_2 = \mathbf{S}C_2 + \mathbf{Y}_2$. The verifier accepts if the norm checks of Section 7 pass, $\mathbf{A}\mathbf{Z}_1 = c_1\mathbf{U} + \mathbf{W}_1$ and $\mathbf{A}\mathbf{Z}_2 = \mathbf{U}C_2 + \mathbf{W}_2$. For $a \in \mathbb{Z}_q$ we write $|a|$ for the absolute value of its representative in $(-q/2, q/2]$, and $\|\cdot\|_\infty$ for the maximum of $|\cdot|$

over the entries. We use Corollary 2.4 of (Baum & Lyubashevsky, 2017): for every matrix M over $\mathbb{Z}_q$ with ψ columns and every $B \in \{0,1\}^{\psi \times \ell}$,

$$\Pr_{B' \leftarrow \{0,1\}^{\psi \times \ell}}[\|M(B'-B)\|_\infty < \tfrac{1}{2}\|M\|_\infty] \le 2^{-\ell}. \tag{1}$$

A matrix over R_q is viewed as a matrix over $\mathbb{Z}_q$ through the coefficient embedding, which commutes with right multiplication by integer matrices.

Proof of Lemma 11. Let $(\mathbf{W}_1, \mathbf{W}_2, c_1, \mathbf{Z}_1, \mathbf{Z}_2)$ and $(\mathbf{W}_1, \mathbf{W}_2, c_1', \mathbf{Z}_1', \mathbf{Z}_2')$ be accepting transcripts with $c_1 \neq c_1'$, and let $C_2 = H(\mathbf{Z}_1, c_1)$ and $C_2' = H(\mathbf{Z}_1', c_1')$.

Step 1: a preimage of $\mathbf{U}$. The difference $c_1 - c_1'$ has infinity norm at most 2 and is invertible in R_q by the assumption of the lemma. Subtracting the first verification equations gives $\mathbf{A}(\mathbf{Z}_1 - \mathbf{Z}_1') = (c_1 - c_1')\mathbf{U}$, so $\bar{\mathbf{S}} = (c_1 - c_1')^{-1}(\mathbf{Z}_1 - \mathbf{Z}_1')$ satisfies $\mathbf{A}\bar{\mathbf{S}} = \mathbf{U}$.

Step 2: two cases. Subtracting the second verification equations gives $\mathbf{A}(\mathbf{Z}_2 - \mathbf{Z}_2') = \mathbf{U}(C_2 - C_2') = \mathbf{A}\bar{\mathbf{S}}(C_2 - C_2')$.

(a) If $\mathbf{Z}_2 - \mathbf{Z}_2' \neq \bar{\mathbf{S}}(C_2 - C_2')$, multiplying by the invertible $c_1 - c_1'$ shows that $\mathbf{X} = (c_1 - c_1')(\mathbf{Z}_2 - \mathbf{Z}_2') - (\mathbf{Z}_1 - \mathbf{Z}_1')(C_2 - C_2')$ is nonzero and satisfies $\mathbf{AX} = \mathbf{0}$; a nonzero column of $\mathbf{X}$ is the vector $\mathbf{z}$ of the lemma. Each coefficient of $(\mathbf{Z}_1 - \mathbf{Z}_1')(C_2 - C_2')$ is the inner product of a row of $\mathbf{Z}_1 - \mathbf{Z}_1'$, of norm at most $2\sqrt{2\psi}\,\sigma_1$ by the norm checks, with a column of $C_2 - C_2'$, whose ψ entries lie in $\{0, \pm 2\}$, so of norm at most $2\sqrt{\psi}$; by Cauchy–Schwarz it is at most $2^{5/2}\psi\sigma_1$. Each coefficient of the product of $c_1 - c_1'$ with a ring component of a column of $\mathbf{Z}_2 - \mathbf{Z}_2'$ is an inner product of a rotation of $c_1 - c_1'$, of norm at most $\sqrt{4 \cdot 60} = 2\sqrt{60}$, with that component, of norm at most $2\sqrt{2N}\sigma_2$; it is at most $2^{5/2}\sqrt{60N}\sigma_2$. This gives the bound of the lemma.

(b) Otherwise $\mathbf{Z}_2 - \mathbf{Z}_2' = \bar{\mathbf{S}}(C_2 - C_2')$. Write $C_2 = J - 2B$ and $C_2' = J - 2B'$, with J the all-ones matrix and $B, B' \in \{0,1\}^{\psi \times \ell}$. Then $C_2 - C_2' = 2(B' - B)$ and

$$\mathbf{Z}_2 - \mathbf{Z}_2' = \mathbf{P}(B' - B), \qquad \mathbf{P} = 2\bar{\mathbf{S}}.$$

Suppose that $\|\mathbf{P}\|_\infty \le 2\|\mathbf{P}(B'-B)\|_\infty$ (Step 3 bounds the probability that this fails). The entries of $\mathbf{Z}_2 - \mathbf{Z}_2'$ are integers of absolute value at most $18\sigma_2 < q/2$, so $|\cdot|$ of each entry of $\mathbf{P}(B' - B)$ is its absolute value as an integer, and $\|\mathbf{P}\|_\infty \le 2\|\mathbf{Z}_2 - \mathbf{Z}_2'\|_\infty$. Hence $\hat{\mathbf{S}} = \mathbf{P}$ satisfies $\mathbf{A}\hat{\mathbf{S}} = 2\mathbf{U}$ and the bound of the lemma. The argument needs $\mathbf{P} = 2\bar{\mathbf{S}}$: the matrix $\bar{\mathbf{S}}(B' - B) = 2^{-1}(\mathbf{Z}_2 - \mathbf{Z}_2')$ has entries of absolute value about $q/2$ wherever $\mathbf{Z}_2 - \mathbf{Z}_2'$ has an odd entry, so it carries no information on the size of $\bar{\mathbf{S}}$.

Step 3: probability. We bound the probability, over H, that the prover outputs transcripts with $\|\mathbf{P}\|_\infty > 2\|\mathbf{P}(B' - B)\|_\infty$. The oracle H is sampled lazily and answered consistently, also across rewound runs; order its distinct queries by the time of their first answer. If one of the queries $(\mathbf{Z}_1, c_1)$ and $(\mathbf{Z}_1', c_1')$ was never asked, its answer is uniform and independent of everything else, and (1) bounds the probability by $2^{-\ell}$. Otherwise both were asked, as queries $i < j$ in this order. The matrix $\mathbf{P}^{(i,j)} = 2(c_1^{(j)} - c_1^{(i)})^{-1}(\mathbf{Z}_1^{(j)} - \mathbf{Z}_1^{(i)})$ is a function of the two inputs, and $B^{(i)}$ of the answer to query i; both are fixed when query j is answered, and the answer to query j gives a uniform $B^{(j)}$ that is independent of them. By (1), $\|\mathbf{P}^{(i,j)}(B^{(j)} - B^{(i)})\|_\infty < \frac{1}{2}\|\mathbf{P}^{(i,j)}\|_\infty$ with probability at most $2^{-\ell}$; exchanging the roles of the two transcripts only changes the sign of $B' - B$. A union bound over the at most $q_H^2/2$ pairs $i < j$ gives $q_H^2\, 2^{-\ell-1} + 2^{-\ell}$ in total. If q_H includes the queries made to verify the output transcripts, the first case cannot occur, and the bound is a single event over all pairs of queries, which covers all blocks of all authorities and both transcripts at once. This is the argument of (Baum & Lyubashevsky, 2017, Lemma 3.6) with $C_2 - C_2'$ replaced by $B' - B$ and $\bar{\mathbf{S}}$ by $2\bar{\mathbf{S}}$. □

Proof of Lemma 13. Samples of $\mathsf{SampleD}_\sigma$ are exact here; the error of the implemented samplers is accounted for separately (Section 8). The authority k and the attempt ι are fixed, and N_V, N_{col}, E and the classes are as in the lemma.

Step 1: three independent sequences. Sampling a random oracle lazily is equivalent to sampling in advance a sequence of independent uniform answers and giving the m-th answer to the m-th

distinct input, in the order of first evaluation, however the inputs are chosen. We use this for X, with answers $a_1, a_2, \ldots$ and $\mathbf{r}_m = \mathsf{SampleD}_\sigma(a_m)$, which are independent samples of $D_\sigma^{N\mu}$, and for H_{agg}, with answers $h_1, h_2, \ldots$ Likewise, we let the authority take the randomness of node i of level $s \geq 1$ in the current round from an array $(\boldsymbol{\rho}_{s,i})$ of independent samples of $D_\sigma^{N\mu}$ drawn in advance; this does not change the distribution, since every node receives fresh samples. The three families are independent of each other. The tree is filled from the left: node i of level $s \geq 1$ has the children $li, \ldots, li + c - 1$ of level $s - 1$, with $c = l$ except possibly for the last node of a level. The number of levels and the number of nodes of every level are nondecreasing in the number of leaves, so every position (s, i) of an actual tree is a position of the tree for N_V leaves, and every class has at least as many columns in the latter. Accepted ballots have distinct identities, so the leaves of one tree are $\mathbf{r}_m$ for distinct indices m, even if seeds repeat.

Step 2: descriptions of columns (first claim). Every column of every tree with at most N_V leaves equals one of the following vectors, which we call its description: (a) $\mathbf{r}_m$ with $m \leq Q$, for a leaf column of Π_{open}; (b) $\boldsymbol{\rho}_{s,i}$, for an inner column of Π_{open}; (c) $\boldsymbol{\rho}_{1,i} - \sum_{m \in M} \mathbf{r}_m$ with a set M of 1 to l distinct indices at most Q, for a column of Π_{zero} at level 1; (d) $\boldsymbol{\rho}_{s,i} - \sum_{j<c} \boldsymbol{\rho}_{s-1,li+j}$ with $1 \leq c \leq l$, for a column of Π_{zero} at level $s \geq 2$. Which description is realized depends on the choices of the adversary, including its seeds and the number of accepted ballots, but every description is a fixed signed sum of at most v distinct entries of the independent families of Step 1, where (v, g) is its class. Its coordinates are independent, and each is $\sigma\sqrt{v}$-subgaussian (Micciancio & Peikert, 2012, Lemma 2.8). Theorem 2.1 of Hsu et al. (2012) with $\Phi = I$ ($\mathrm{tr} = N\mu$, operator norm 1) shows that its squared norm exceeds $\sigma^2 v\, h(N\mu, 1, \tau + \ln N_{\mathrm{col}} + g \ln Q)$, the value b_u^2 of its class, with probability at most $e^{-\tau} N_{\mathrm{col}}^{-1} Q^{-g}$. Let N_1 and $N_{\geq 2}$ be the numbers of inner nodes of the tree for N_V leaves at level 1 and above level 1, and $N_{\mathrm{in}} = N_1 + N_{\geq 2}$. There are at most Q^c sets M of size c, so the sum over all descriptions is

$$\frac{e^{-\tau}}{N_{\mathrm{col}}} \Big(\underbrace{Q \cdot \tfrac{1}{Q}}_{(a)} + \underbrace{N_{\mathrm{in}} \cdot \tfrac{1}{Q}}_{(b)} + \underbrace{N_1 \textstyle\sum_{c=1}^{l} Q^{c-l}}_{(c)} + \underbrace{N_{\geq 2}\, l}_{(d)} \Big) \leq \frac{1 + (l+1) N_{\mathrm{in}}}{N_{\mathrm{col}}}\, e^{-\tau} \leq l\, e^{-\tau},$$

since $Q^{-1} \leq 1$, $\sum_{c=1}^{l} Q^{c-l} \leq l$, and $N_{\mathrm{col}} = N_V + 2N_{\mathrm{in}}$ with $N_V \geq 1$ and $l \geq 2$. Outside this event, every column of the actual tree satisfies $\|\mathbf{s}_u\|_2 \leq b_u$ for the value b_u of its class, for every adaptive choice of seeds and every number of accepted ballots up to N_V. The union over the number of children is part of this sum.

Step 3: first shift (second claim). The challenges $c_{1,e}$ of the attempt are derived from the answer to the query $H_{\mathsf{agg}}(\mathsf{par}, \mathsf{BB}_1)$ and from (k, ι, e). The adversary can influence which query this is, for instance by trying several boards, and into which block a column falls, through the number of accepted ballots. Fix a description, the index $j \leq Q$ of the query and a block index $e \leq E$, and let c be the challenge of block e derived from h_j. It is uniform in $\mathcal{S}$ and independent of the vector of the description, since h_j is independent of the other two families, even though the input of the query contains commitments to their entries. Conditioned on c, the map Φ that multiplies each of the μ ring elements by c is fixed, with $\|\Phi\|_F^2 = 60N\mu$ and operator norm at most $\|c\|_1 = 60$, and Theorem 2.1 of Hsu et al. (2012) bounds the squared norm of c times the vector by $\sigma^2 v\, h(60N\mu, 3600, \tau + \ln N_{\mathrm{col}} + (g+1) \ln Q)$, except with probability $e^{-\tau} N_{\mathrm{col}}^{-1} Q^{-g-1}$. The sum over j and e multiplies the sum of Step 2 by $QE \cdot Q^{-1} = E$, so the bound fails for some triple with probability at most $lE\, e^{-\tau}$. Outside this event, the squared first shift of the actual tree is at most the sum of the class values over its columns, and hence at most Θ_1^2, since every class has at least as many columns in the tree for N_V leaves.

Step 4: second shift (third claim). We first compare the blocks. The columns of Π_{open} form one sequence, and the columns of Π_{zero} one sequence per level; each sequence is cut into consecutive blocks of ψ columns, of which only the last may be shorter, and all columns of a sequence have the same class value b^2. A sequence of n' columns thus gives blocks with $F_e = \psi b^2$, except for the last one, and the sum, the sum of squares and the maximum of these values are

nondecreasing in n'. Since n' is nondecreasing in the number of leaves for every sequence, the values F, $\sum_e F_e^2$ and $\max_e F_e$ of the actual tree, computed with the class values, are at most those of the tree for N_V leaves.

Next, freshness. In an attempt, the authority samples the masks and computes, for every block of the actual tree, $\mathbf{Z}_{1,e} = c_{1,e}\mathbf{S}_e + \mathbf{Y}_{1,e}$, the query $C_{2,e} = H(\mathbf{Z}_{1,e}, c_{1,e})$ and $\mathbf{Z}_{2,e}$; it applies the two rejection steps only at the end, to the whole attempt. The queries are therefore made on the masks as sampled, and the bound below involves no conditioning on acceptance. (In an implementation that queries H only after the first rejection step, rejection could only cancel queries and hence only exclude the event below.) Let $\mathsf{BadFresh}$ be the event that one of these queries was made before. Before these queries, the masks of the attempt enter nothing that other parties see except $h_{k,\iota} = G(\mathbf{W}_k^{(\iota)}, s_{k,\iota})$, whose salt is revealed only in round 2, after the queries. Let NoG be the event that no party other than the authority queried G with the salt $s_{k,\iota}$ before it was revealed; the authority's own evaluation of $h_{k,\iota}$ is not such a query. Its complement has probability at most $q_G 2^{-\lambda}$, with q_G the queries of the other parties to G. On NoG, every earlier query to H, by the adversary or by honest parties, whose queries involve other, independent masks, is independent of $\mathbf{Y}_{1,e}$. Given all other values, one column of $\mathbf{Z}_{1,e}$ takes any fixed value with probability at most $\sigma_1^{-N\mu}$, because every value of D_{σ_1} has probability at most $1/\rho_{\sigma_1}(\mathbb{Z}) \le 1/(\sqrt{2\pi}\,\sigma_1) \le 1/\sigma_1$; here no reduction modulo q is involved, since the query contains $\mathbf{Z}_{1,e}$ as a vector of integers. This includes the authority's own queries for the other blocks of the attempt, so the queries of the attempt are also pairwise distinct outside $\mathsf{BadFresh}$. A union bound over the at most q_H^* earlier queries and the at most E blocks gives $\Pr[\mathsf{BadFresh}] \le \varepsilon_{\text{fresh}}$.

Consider now the experiment in which these queries are answered with fresh uniform values, whether or not they were made before; it differs from the real one only on $\mathsf{BadFresh}$. In it, the entries of all $C_{2,e}$ form a Rademacher vector $\mathbf{x}$ that is independent of the witnesses $\mathbf{S}_e$, which are fixed in round 1. Then $\|(\mathbf{S}_e C_{2,e})_e\|_F^2 = \mathbf{x}^\top \Sigma \mathbf{x}$ with $\Sigma = \bigoplus_e I_\ell \otimes \mathbf{S}_e^\top \mathbf{S}_e$. On the event of Step 2 and by the comparison of the blocks, $\operatorname{tr}\Sigma = \ell \sum_e \|\mathbf{S}_e\|_F^2 \le \ell F$, $\operatorname{tr}(\Sigma^2) = \ell \sum_e \|\mathbf{S}_e^\top \mathbf{S}_e\|_F^2 \le \ell \sum_e F_e^2$ and $\|\Sigma\| = \max_e \|\mathbf{S}_e^\top \mathbf{S}_e\| \le \max_e F_e$. The vector $\mathbf{x}$ has independent 1-subgaussian coordinates, and the bound of Theorem 2.1 of Hsu et al. (2012) is increasing in these three quantities, so $\mathbf{x}^\top \Sigma \mathbf{x} \le \Theta_2^2$ except with probability $e^{-\tau}$. In the real experiment, the third claim therefore fails with probability at most $e^{-\tau} + \varepsilon_{\text{fresh}}$. With Steps 2 and 3, all three claims hold except with probability $(l + lE + 1)e^{-\tau} + \varepsilon_{\text{fresh}}$. □

Where the lemmas are used. Lemma 13 is used only for zero knowledge, in the paragraph on the zero knowledge of the aggregation and hence in ε_{zk} of Theorem 3, and for completeness. Theorem 14 does not use it: extraction relies on the norm checks of the verifier, which hold for every accepted proof, and on Lemma 11. Lemma 11 enters Theorem 14 through assumption (iii). It provides the openings with slack 2 and bound $B_A = 36\sigma_2\sqrt{N(2d+L)}$, since each entry of $\hat{\mathbf{S}}$ is at most $2 \cdot 18\sigma_2$ and a column has $N(2d+L)$ entries. It also provides the bound β_{BL}, which is $\sqrt{N(2d+L)}$ times the infinity-norm bound of Step 2(a), and the error ε_H. The slack 2 is then carried by Lemma 12, giving the term $(l+2)B_A$ of β_{SIS} and, at the root, the term $2\beta + B_A$, which also arises when the reduction compares an opening with slack 1 and one with slack 2 by Lemma 9; and by the comparison of the authorities' openings of the leaves with the openings extracted from the ballot proof, where Lemma 9 with $f' = 2$ and $f = c - c'$ gives the term $2\bar{B}_V + \eta_{\max} B_A$. The term $\eta_{\max}(\bar{B}_V + \bar{B}_{\text{OR}})$ comes from the ballot proof alone and does not involve the slack.

C Implementation details

Samplers. For $\sigma = 1$ the sampler draws r uniformly from $[0, 2^{256})$ and compares it with a table of $\lfloor 2^{256} \Pr[|X| \le x] \rfloor$, computed with 512-bit arithmetic; its statistical distance from $D_{\mathbb{Z},1}$ is below 2^{-251} per coefficient. For the masks, a table of the same kind samples $D_{\mathbb{Z},256}$ (distance below 2^{-243}), and at most two levels of the convolution of Micciancio & Walter (2017, Thm. 2.1 and

Lemma 5.1) combine four such samples as $a_2(a_1x_1 + b_1x_2) + b_2(a_1x_3 + b_1x_4)$ with coprime a_l, b_l below the bound of the convolution theorem for $\varepsilon = 2^{-220}$; the result is within statistical distance 2^{-216} of $D_{\mathbb{Z},\sigma}$ for $\sigma = 256\sqrt{(a_1^2 + b_1^2)(a_2^2 + b_2^2)}$. We use the smallest such σ above the value required by Lemmas 8 and 13, which exceeds it by at most 0.04%, and compute all bounds with it. Since the samples are independent and each is within 2^{-216} of its ideal distribution, the sampling error adds to ε_{zk} at most $(n + 1 + 2L)N(2d + L) \cdot 2^{-216}$ per attempt of the ballot proof, which draws $n+1+2L$ mask vectors of $N(2d+L)$ coefficients, and at most $E(\psi+\ell)N(2d+L)\cdot 2^{-216}$ per attempt of the aggregation, which draws at most $\psi + \ell$ mask columns per block; the commitment randomness, sampled once per ballot or node with $\sigma = 1$, adds 2^{-251} per coefficient. Lemma 13 takes a union bound over all outputs of X, so its ideal samples are substituted in one hybrid for all at most Q outputs of X at once, at a cost of $QN(2d + L) \cdot 2^{-251} \le 2^{-174.4}$ for $Q = 2^{64}$, counted once for the whole experiment. For $d = 7$ and $n = 4$ this is below $2^{-201.2}$ for $L = 1$ and below $2^{-198.7}$ for $L = 10$ (153 600 coefficients) per attempt of the ballot proof, and below $2^{-183.0}$ for $L = 1$ and below $2^{-182.3}$ (about $2^{-182.36}$) for $L = 10$ per attempt of the aggregation at $N_V = 10^6$, well below the rejection errors.

Rejection steps and norm checks. The tests that decide acceptance are exact up to one explicitly bounded deviation, the cap of the Bernoulli counter described below. A rejection step with shift $\mathbf{s}$, response $\mathbf{z}$ and parameter σ accepts with probability $\min(1, \exp((\|\mathbf{s}\|_2^2 - 2\langle \mathbf{z}, \mathbf{s}\rangle)/(2\sigma^2) - \ln M))$ (Lyubashevsky, 2012). The implementation computes $\langle \mathbf{z}, \mathbf{s}\rangle$ and $\|\mathbf{s}\|_2^2$ exactly in 128-bit integers; σ^2 is an integer, since the achievable values above are $\sigma = 1$ or $\sigma^2 = 256^2$ times at most two factors $a_l^2 + b_l^2$; and $\ln M$ is rational, by the choices $\alpha = 141/10$ for the ballot proof and $\alpha' = 281/10$ for the aggregation with $r = 309/20$, which give $\ln M = 43669/39762$ and $\ln M' = 86929/157922$. The exponent is therefore a rational number $-\gamma$, and a Bernoulli variable with parameter $\exp(-\gamma)$ is sampled exactly by Algorithm 1 of Canonne et al. (2020), whose correctness for every rational $\gamma \ge 0$ is Proposition 33 there (numbering of the full version); it needs only uniform integers in given ranges, and these are drawn from the output of SHAKE by rejection. The counter K of that algorithm is unbounded; we stop it at $K = 2^{20}$, which keeps every product in 128 bits and can change the output only if the counter reaches the cap, with probability below $1/(2^{20} - 1)! < 2^{-2^{23}}$ per call. Given ideal random bits, the acceptance probability is thus the one in the proofs up to this term. All sums of products are computed in 128-bit integers with overflow detection: the prover aborts on overflow, which cannot occur in the supported ranges of the parameters, and the verifier saturates, so that an oversized input fails the norm check. The ranges of all intermediate quantities are listed with the implementation. The norm checks compare exactly computed squared norms with the integers B_J^2, $2\psi\sigma_1^2$, $81\sigma_2^2$ and $2N\sigma_2^2$. The only idealization left is that SHAKE, which expands the seeds and provides all random bits, behaves as a random oracle, as X does in the proofs.

Error budget. Table 7 collects the statistical errors of the simulations per attempt for the largest parameter sets. Over at most 2^{40} attempts of both kinds together their sum is below $2^{-130.3}$; the one-time terms, $2^{-174.4}$ for the hybrid over the outputs of X and 2^{-1300} for the choice of $\mathbf{A}$, do not change this. The statistical part of ε_{zk} is therefore below 2^{-128} over the whole experiment; the computational terms of Theorem 3 are separate. This bounds the statistical distance of the simulations only and does not by itself give 128-bit security of the protocol, which also rests on the computational assumptions and, for verifiability, on the non-tight reduction of Theorem 14. The values of Table 7 are computed from the formulas above (docs/PARAMETERS.md, Section 7), not measured.

Reproducibility. The implementation, the logs of all measurements and the commands that produce them are published (Alizadeh, 2026); release `v1.1` of the repository contains all commits and files cited below; its protocol code is that of release `v1.0`, with which the measurements were made, unchanged. The numbers of this paper were produced with commit `d03fd81`; the logs of Table 4, the parameter chain of Table 3 (`results/params_chain_signed.jsonl` and `.log`) are in commit `996646f`, the EVOLVE ballot

Source	Ballot attempt	Aggregation attempt
Tail bounds (Lemma 8, resp. Lemma 13)	2^{-171}	$< 2^{-171.0}$
Rejection, $e^{-r^2/2}/M$ (one, resp. two steps)	$< 2^{-173.7}$	$< 2^{-171.9}$
Mask samplers	$< 2^{-198.7}$	$< 2^{-182.3}$
Freshness of the hashed first messages	$< 2^{-26000}$	2^{-192}
Cap of the Bernoulli counter	$< 2^{-2^{23}}$	$< 2^{-2^{23}}$
Sum	$< 2^{-170.7}$	$< 2^{-170.3}$

Table 7: Statistical errors of the simulations per attempt, for $N_V = 10^6$ ($E = 2142$), a ballot with $n = 4$ and $L = 10$, $Q = 2^{64}$ and 256-bit salts. Over 2^{40} attempts the sum is below $2^{-130.3}$. Computed from the formulas of this section (docs/PARAMETERS.md, Section 7), not measured.

at $d = 6$ (`results/linux_cloud/bench_evolve_d6.csv`) in commit `5d02f9e`, the logs of the cross-platform comparison in commit `1d97995`. Table 5 was measured with commit `af7e386`, which adds options for the scaling runs to the benchmark program but leaves the protocol code of `d03fd81` unchanged; its logs are in `results/windows/scaling_20260927_0833` (commit `4971977`), and `scripts/run_windows_scaling.bat` runs it. `scripts/reproduce_paper.sh` lists the other commands. The estimates of Table 3 under the quantum and core-SVP cost models, and those for ML-KEM-512 and for $d = 8$, are in `results/quantum_models/` (script `tools/quantum_models.py`, commits `f3a8137`, `8e6e375` and `47867dc`); the hardness estimates of the encryption of Appendix D are in the same place. The estimate for $L = 10$ in Section 8 is in `results/params_L10_q43.json`, and the check of Appendix A in `results/evolve_opening_check.txt`, produced by `scripts/evolve_opening_check.py`.

Choice of κ. With $p = 1/M'^2$ and $s(\kappa) = 1-(1-p)^\kappa$ as in Section 7.3, the expected number of rounds and of attempts per authority, in the worst case for robustness of $t = 3$ honest authorities, are as follows (κ also enters the loss of Theorem 14 through $\Lambda = \binom{n}{t}\kappa^t$, which appears squared, so doubling κ costs $2t$ bits there):

κ	1	2	3	4	6	8
$s(\kappa)$	0.33	0.55	0.70	0.80	0.91	0.96
rounds $s(\kappa)^{-3}$	27.2	5.9	2.9	1.9	1.3	1.1
attempts $\kappa\, s(\kappa)^{-3}$	27.2	11.7	8.6	7.8	7.9	9.0

Failed rounds. An honest authority has no non-aborting attempt in a round with probability $(1-p)^4 \approx 0.20$. A round can fail in two ways. First, if fewer than t authorities announce an index, no responses are published. Such a round costs the computation of both rounds and the publications of its first round, namely the node commitments, $\mathbf{V}_k$ and the κ hash values of every participating authority, since a repeated round starts with a new tree. For $N_V = 10^4$ this is about 3.6 MB per authority, or 0.36 KB per ballot, against 28.2 KB per ballot for a complete contribution. Second, if at least t authorities announce an index but fewer than t valid contributions arrive by Δ_2, some announcing authority has misbehaved. This cannot make a round fail that would otherwise succeed: if at least t honest authorities have a non-aborting attempt, their contributions alone complete the round. Misbehavior can only turn a round that fails anyway into one in which the honest announcing authorities have already published their responses. Such a round costs, in addition, one publication of the second round per honest announcing authority. The misbehaving authority is excluded afterwards, so this happens in at most $n - t$ rounds, and the expected number of rounds remains at most $s(\kappa)^{-t}$.

D An encryption scheme with publicly verifiable opening

We show that the key-release construction of Section 7.1 satisfies (E1)–(E4) for the identity-based encryption of Gentry et al. (2008), asymptotically. We use a separate modulus q', dimensions n' and m', and a noise distribution χ over $\mathbb{Z}$, all fixed in par; plaintexts are $x \in \{0,1\}^\lambda$. H_{id} and

H_{key} are random oracles, domain-separated from those of the protocol, with outputs in $\mathbb{Z}_{q'}^{n' \times \lambda}$ and in the coins of the preimage sampler. For a matrix $\mathbf{F}$ with trapdoor, $\mathsf{SamplePre}$ outputs, for every column of a target matrix, a short preimage under $\mathbf{F}$ (Gentry et al., 2008).

- KGen: generate $\mathbf{F} \in \mathbb{Z}_{q'}^{n' \times m'}$ with a trapdoor and a seed $\kappa_{\mathsf{key}} \in \{0,1\}^\lambda$; $\mathsf{pk} = \mathbf{F}$, $\mathsf{sk} = (\text{trapdoor}, \kappa_{\mathsf{key}})$. $\mathsf{VerKey}(\mathsf{pk}) = 1$ if and only if $\mathsf{pk} \in \mathbb{Z}_{q'}^{n' \times m'}$.
- User key: for a label lb, let $\mathbf{D} = H_{\mathsf{id}}(\mathsf{par}, \mathsf{lb})$. For $j = 1, 2, \ldots,$ compute $\mathbf{K} = \mathsf{SamplePre}(\text{trapdoor}, \mathbf{D}; H_{\mathsf{key}}(\kappa_{\mathsf{key}}, \mathsf{lb}, j))$ and stop at the first j for which every column of $\mathbf{K}$ has norm at most B_K. The key is a deterministic function of sk and lb. Since $\mathsf{SamplePre}$ meets the norm bound with overwhelming probability, the loop ends after an expected constant number of iterations, and Dec and Open run in expected polynomial time.
- A key $\mathbf{K} \in \mathbb{Z}^{m' \times \lambda}$ is *valid* for $(\mathsf{pk}, \mathsf{lb})$ if $\mathbf{FK} = \mathbf{D}$ and every column has norm at most B_K.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{lb}, x)$: sample $\mathbf{h} \leftarrow \mathbb{Z}_{q'}^{n'}$, $\mathbf{g}_0 \leftarrow \chi^{m'}$ and $\mathbf{g}_1 \leftarrow \chi^\lambda$, and output $e = (\mathbf{e}_0, \mathbf{e}_1) = (\mathbf{F}^\top \mathbf{h} + \mathbf{g}_0,\ \mathbf{D}^\top \mathbf{h} + \mathbf{g}_1 + \lfloor q'/2 \rfloor x)$.
- $\mathsf{Dec}_{\mathbf{K}}(e)$: if $e \notin \mathbb{Z}_{q'}^{m'} \times \mathbb{Z}_{q'}^{\lambda}$, return $\perp$; otherwise return the vector x' whose i-th bit is 0 if the i-th entry of $\mathbf{e}_1 - \mathbf{K}^\top \mathbf{e}_0$, as an integer in $(-q'/2, q'/2]$, has absolute value less than $q'/4$, and 1 otherwise. $\mathsf{Dec}(\mathsf{sk}, \mathsf{lb}, e) = \mathsf{Dec}_{\mathbf{K}}(e)$ for the user key $\mathbf{K}$ of lb.
- $\mathsf{Open}(\mathsf{sk}, \mathsf{lb}, e)$ outputs the user key $\mathbf{K}$ of lb. $\mathsf{Judge}(\mathsf{pk}, \mathsf{lb}, e, \omega)$ returns fail if ω is not a valid key for $(\mathsf{pk}, \mathsf{lb})$, and $\mathsf{Dec}_\omega(e)$ otherwise.

Proposition 15 (Existence of EO)**.** *Let B_0, B_1 and B_K be such that, except with negligible probability, $\|\mathbf{g}_0\|_2 \le B_0$ for $\mathbf{g}_0 \leftarrow \chi^{m'}$, every entry of $\mathbf{g}_1 \leftarrow \chi^\lambda$ has absolute value at most B_1, and $\mathsf{SamplePre}$ with an honestly generated trapdoor outputs columns of norm at most B_K with overwhelming probability. If*

$$q' > 4\,(B_1 + B_K B_0) + 2,$$

then the construction satisfies (E1), (E2) and (E3), and (E3) holds for every pk and every valid key, without computational assumptions. If moreover the identity-based encryption of Gentry et al. (2008) with these parameters is adaptively IND-ID-CPA secure in the random oracle model, which holds under the hardness of LWE with modulus q' and noise χ, the construction satisfies (E4) in the random oracle model. Both conditions can be met with q' polynomial in n'.

Proof. Decryption with any valid key. Let $e = \mathsf{Enc}(\mathsf{pk}, \mathsf{lb}, x)$ for an arbitrary $\mathsf{pk} = \mathbf{F}$, and let $\mathbf{K}$ be any valid key for $(\mathsf{pk}, \mathsf{lb})$. Since $\mathbf{FK} = \mathbf{D}$,

$$\mathbf{e}_1 - \mathbf{K}^\top \mathbf{e}_0 = \mathbf{D}^\top \mathbf{h} + \mathbf{g}_1 + \lfloor q'/2 \rfloor x - \mathbf{K}^\top \mathbf{F}^\top \mathbf{h} - \mathbf{K}^\top \mathbf{g}_0 = \mathbf{g}_1 - \mathbf{K}^\top \mathbf{g}_0 + \lfloor q'/2 \rfloor x.$$

By the Cauchy–Schwarz inequality, each entry of $\mathbf{g}_1 - \mathbf{K}^\top \mathbf{g}_0$ has absolute value at most $B := B_1 + B_K B_0$ whenever the noise satisfies its bounds. For a bit 0 the entry of $\mathbf{e}_1 - \mathbf{K}^\top \mathbf{e}_0$ then has absolute value at most $B < q'/4$. For a bit 1 it lies within B of $\lfloor q'/2 \rfloor \ge (q'-1)/2$, so its distance from 0 modulo q' is at least $(q'-1)/2 - B > q'/4$, since $q' > 4B + 2$. This holds for every valid $\mathbf{K}$ at once, including one chosen after e is seen, and the bounds fail only with negligible probability over the encryption randomness. Hence $\mathsf{Dec}_{\mathbf{K}}(e) = x$ for all valid $\mathbf{K}$, except with negligible probability, whatever $\mathbf{F}$ is.

(E1). Honest keys pass VerKey. The user key of lb is valid by construction, so decryption is correct by the first step.

(E2). For honestly generated keys, Open outputs the user key $\mathbf{K}$ that Dec uses, and this key is valid. Judge therefore computes $\mathsf{Dec}_{\mathbf{K}}(e)$, which is $\mathsf{Dec}(\mathsf{sk}, \mathsf{lb}, e)$ for every e, including e

outside the ciphertext space, for which both return $\perp$. The equality relies on the deterministic derivation of the user key: Dec and Open use the same key.

(E3). An evidence ω on which Judge does not return fail is a valid key for $(\mathsf{pk}, \mathsf{lb})$. By the first step, Judge then returns x for an honest ciphertext e, except with negligible probability, for every pk in the format checked by VerKey and every such ω, even if both are chosen by the adversary.

(E4). The oracle $\mathsf{Open}(\mathsf{sk}, \cdot, \cdot)$ on labels other than lb^* returns user keys of identities $(\mathsf{pk}, \mathsf{lb})$ with $\mathsf{lb} \neq \mathsf{lb}^*$, which are key-extraction queries in the IND-ID-CPA experiment. Since κ_{key} is secret and H_{key} is a random oracle, the derived keys are distributed as keys that the IND-ID-CPA challenger samples once per identity, with the same procedure, and stores, except with probability at most $q_{\mathsf{key}} 2^{-\lambda}$ for q_{key} queries to H_{key}. Conditioning the output of $\mathsf{SamplePre}$ on the norm bound changes its distribution only by a negligible amount, so the proof of Gentry et al. (2008), which programs H_{id} with $\mathbf{FK}$ for keys $\mathbf{K}$ sampled in advance, applies to this extraction procedure as well. An adversary against (E4) is therefore an adversary against the adaptive IND-ID-CPA security of the scheme of Gentry et al. (2008) in the random oracle model, which reduces to LWE with modulus q' and noise χ.

Parameters. As in Gentry et al. (2008), take $m' = O(n' \log q')$, a trapdoor for which $\mathsf{SamplePre}$ outputs columns of norm at most $B_K = \mathrm{poly}(n')$ with overwhelming probability, and χ a discrete Gaussian of width $\alpha q'$, for which $B_0 = O(\alpha q' \sqrt{m'})$ and $B_1 = O(\alpha q' \sqrt{\log n'})$ up to $\omega(1)$ factors. The condition $q' > 4(B_1 + B_K B_0)$ then holds for $\alpha = 1/\mathrm{poly}(n')$, a factor of about $\sqrt{m'}$ below the value that suffices for correctness with a key chosen independently of the encryption noise, and LWE with such α and $\alpha q' \geq 2\sqrt{n'}$ is hard if worst-case lattice problems are hard to approximate within polynomial factors for quantum algorithms (Regev, 2009), with q' polynomial in n'. $\square$

Concrete parameters. The proposition is asymptotic; we now fix a concrete instance. The argument for (E1)–(E3) uses only $\mathbf{FK} = \mathbf{D}$, the norm bound on $\mathbf{K}$ and the Cauchy–Schwarz inequality, so it applies to any trapdoor for $\mathbf{F}$, including the NTRU trapdoor of Ducas et al. (2014), which gives among the smallest ciphertexts of GPV-type schemes. Let $R' = \mathbb{Z}[X]/(X^{N'} + 1)$ with $N' = 2048$ and $q' = 9\,123\,495\,937$, a 34-bit prime with $q' \equiv 1 \pmod{2N'}$. The master key is an NTRU basis: $f, g \leftarrow D_{\sigma_f}^{N'}$ with $\sigma_f = 1.17\sqrt{q'/(2N')}$, rejected until the Gram–Schmidt norm of the basis is at most $1.17\sqrt{q'}$, as in (Ducas et al., 2014); $\mathbf{F} = (1, h)$ with $h = g/f \bmod q'$ is the public key, and VerKey checks $h \in R'_{q'}$. For a label, $\mathbf{D} = d = H_{\mathsf{id}}(\mathsf{par}, \mathsf{lb}) \in R'^2_{q'}$, and the user key is $\mathbf{K} = (s_1, s_2) \in R'^2$ with $s_1 + h s_2 = d$, sampled with the Gaussian sampler of (Gentry et al., 2008; Ducas et al., 2014) at $\sigma_K = \lceil 1.17\sqrt{q'} \cdot \eta \rceil = 506\,228$, where $\eta = \sqrt{\ln(4N'(1 + 1/\varepsilon))/\pi} = 4.53$ bounds the smoothing parameter $\eta_\varepsilon(\mathbb{Z}^{2N'})$ for $\varepsilon = 2^{-80}$ (Gentry et al., 2008, Lemma 3.1); the sampler is repeated, with the coins $H_{\mathsf{key}}(\kappa_{\mathsf{key}}, \mathsf{lb}, j)$, until $\|(s_1, s_2)\|_2 \leq B_K = \lfloor 1.1\, \sigma_K \sqrt{2N'} \rfloor = 35\,638\,451$, which a Gaussian vector of dimension 4096 exceeds with probability below 2^{-57} (tail bound of Lyubashevsky (2012, Lemma 4.4), which for a lattice coset carries an extra factor $(1 + \varepsilon)/(1 - \varepsilon)$). A key is valid if $s_1 + h s_2 = d$ and $\|(s_1, s_2)\|_2 \leq B_K$. Encryption of $x \in \{0,1\}^{256}$ samples $r, e_1 \in \{-1, 0, 1\}^{N'}$ and $e_2 \in \{-1, 0, 1\}^{256}$ uniformly (so that $\mathbf{g}_0 = (r, -e_1)$ in the notation of the proposition) and outputs $u = rh + e_1 \in R'_{q'}$ and $v_j = (rd)_j + e_{2,j} + \lfloor q'/2 \rfloor x_j$ for $j < 256$; decryption reads x_j from $v_j - (us_2)_j = (rs_1 - e_1 s_2)_j + e_{2,j} + \lfloor q'/2 \rfloor x_j$. Each coefficient of $rs_1 - e_1 s_2$ is an inner product of (r, e_1) with a signed rotation of (s_1, s_2), so its absolute value is at most $B_0 B_K$ with $B_0 = \sqrt{2N'} = 64$, the largest possible norm of a ternary (r, e_1), and $B_1 = 1$; the rows of the negacyclic matrix of s_i are signed permutations of its coefficients, so the Cauchy–Schwarz step of the proposition applies unchanged. Hence $B_1 + B_K B_0 = 2\,280\,860\,865 < \lfloor q'/4 \rfloor = 2\,280\,873\,984$: the condition of the proposition holds, and here it holds for every ciphertext, not only with overwhelming probability, so decryption under any valid key is correct for every honest ciphertext. The hardness of the master key against key recovery and of the ciphertexts against RLWE with ternary secret and noise, over the $N' + 256$ available samples, is estimated in Table 8 with the models of Table 3; (E4) then rests on the NTRU assumption for h and on this RLWE instance, as in (Ducas et al., 2014). A ciphertext

Cost model	Ciphertext (RLWE, $N' = 2048$, ternary)	Master key (NTRU, $\sigma_f = 1746$)
MATZOV, classical	$2^{199.0}$	$2^{621.5}$
MATZOV, quantum	$2^{188.3}$	$2^{566.8}$
core-SVP, classical	$2^{174.0}$	$2^{625.2}$
core-SVP, quantum	$2^{158.5}$	$2^{567.4}$

Table 8: Hardness estimates for the concrete instantiation of EO ($N' = 2048$, $q' = 9\,123\,495\,937$), lattice estimator commit 53da598, best attack over those of Table 3; the NTRU row uses the estimator's NTRU module for circulant instances; its dense-sublattice and uSVP attacks return no cost below the full dimension for this instance (the key is not overstretched), and the figures are those of the BDD attack. Script and output: `tools/quantum_models.py` and `results/quantum_models/` of the repository.

takes $N' \cdot 34$ bits for u and $256 \cdot 34$ bits for v, 9792 bytes in total, 9.6 KB; the opening need only contain s_2, since Judge recomputes $s_1 = d - hs_2$ before checking the norm, about 5.8 KB at 23 bits per coefficient; the master public key h takes 8.5 KB. The n ciphertexts of a ballot therefore take 38.3 KB for $n = 4$ and 47.8 KB for $n = 5$ (Table 3). The same rule with $N' = 1024$ gives the 32-bit modulus $q' = 2\,256\,363\,521$ and halves the ciphertext, to 5.0 KB, but its RLWE instance is estimated at only $2^{103.0}$ under the classical MATZOV model and $2^{75.0}$ in core-SVP, far below the 192 bits that Ducas et al. (2014) estimated in 2014, with the methodology of the time, for this degree and a 27-bit modulus; this is why we take $N' = 2048$. We have not implemented this encryption; the data flow of the protocol and the 32-byte payload are exercised by the placeholder of Section 8.

References

Abdolmaleki, B., Clark, J., Foroutani, M., Khazaei, S., & Nasirzadeh, S. (2026). A practical framework for lattice-based non-interactive publicly verifiable secret sharing. In *PQCrypto 2026*, Part II, LNCS. Springer. https://doi.org/10.1007/978-3-032-22698-3_4

Albrecht, M. R., & Lai, R. W. F. (2021). Subtractive sets over cyclotomic rings: Limits of Schnorr-like arguments over lattices. In *CRYPTO 2021*, LNCS 12826 (pp. 519–548). Springer. https://doi.org/10.1007/978-3-030-84245-1_18

Albrecht, M. R., Player, R., & Scott, S. (2015). On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3), 169–203. https://doi.org/10.1515/jmc-2015-0016. Software: https://github.com/malb/lattice-estimator

Albrecht, M. R., Gheorghiu, V., Postlethwaite, E. W., & Schanck, J. M. (2020). Estimating quantum speedups for lattice sieves. In *ASIACRYPT 2020*, LNCS 12492 (pp. 583–613). Springer. https://doi.org/10.1007/978-3-030-64834-3_20

Alizadeh, J. (2026). T-EVOLVE: reference implementation (software, release v1.1). https://github.com/futures1mple/t-evolve-cryptoalg

Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. (2016). Post-quantum key exchange—a new hope. In *USENIX Security 2016* (pp. 327–343). USENIX Association. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim

Aranha, D. F., Baum, C., Gjøsteen, K., & Silde, T. (2023). Verifiable mix-nets and distributed decryption for voting from lattice-based assumptions. In *ACM CCS 2023* (pp. 1467–1481). https://doi.org/10.1145/3576915.3616683

Aranha, D. F., Baum, C., Gjøsteen, K., Silde, T., & Tunge, T. (2021). Lattice-based proof of shuffle and applications to electronic voting. In *CT-RSA 2021*, LNCS. Springer. https://doi.org/10.1007/978-3-030-75539-3_10

Barbosa, M., Barthe, G., Doczkal, C., Don, J., Fehr, S., Grégoire, B., Huang, Y.-H., Hülsing, A., Lee, Y., & Wu, X. (2023). Fixing and mechanizing the security proof of Fiat–Shamir with aborts and Dilithium. In *CRYPTO 2023*, LNCS 14085 (pp. 358–389). Springer. https://doi.org/10.1007/978-3-031-38554-4_12

Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe,

P., Seiler, G., & Stehlé, D. (2021). CRYSTALS-Kyber: Algorithm specifications and supporting documentation (version 3.02). NIST PQC Round 3 submission. https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf

Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., & Peikert, C. (2018). More efficient commitments from structured lattice assumptions. In *SCN 2018*, LNCS 11035 (pp. 368–385). Springer. https://doi.org/10.1007/978-3-319-98113-0_20

Baum, C., Damgård, I., Oechsner, S., & Peikert, C. (2016). Efficient commitments and zero-knowledge protocols from Ring-SIS with applications to lattice-based threshold cryptosystems. *Cryptology ePrint Archive*, Report 2016/997. https://eprint.iacr.org/2016/997

Baum, C., & Lyubashevsky, V. (2017). Simple amortized proofs of shortness for linear relations over polynomial rings. *Cryptology ePrint Archive*, Report 2017/759. https://eprint.iacr.org/2017/759

Bellare, M., & Neven, G. (2006). Multi-signatures in the plain public-key model and a general forking lemma. In *ACM CCS 2006* (pp. 390–399). https://doi.org/10.1145/1180405.1180453

Bernhard, D., Cortier, V., Galindo, D., Pereira, O., & Warinschi, B. (2015). SoK: A comprehensive analysis of game-based ballot privacy definitions. In *IEEE Symposium on Security and Privacy 2015* (pp. 499–516). IEEE. https://doi.org/10.1109/SP.2015.37

Beullens, W., & Seiler, G. (2023). LaBRADOR: Compact proofs for R1CS from Module-SIS. In *CRYPTO 2023*, LNCS 14085 (pp. 518–548). Springer. https://doi.org/10.1007/978-3-031-38554-4_17

Black, I., McFall, E., Whidden, J., Xie, B., & Cartor, R. (2022). Practical quantum-safe voting from lattices, extended (EVOLVED). *Cryptology ePrint Archive*, Report 2022/1686. Preprint. https://eprint.iacr.org/2022/1686

Boneh, D., Gennaro, R., Goldfeder, S., Jain, A., Kim, S., Rasmussen, P. M. R., & Sahai, A. (2018). Threshold cryptosystems from threshold fully homomorphic encryption. In *CRYPTO 2018*, LNCS 10991 (pp. 565–596). Springer. https://doi.org/10.1007/978-3-319-96884-1_19

Bonte, C., Nicolas, G., & Smart, N. P. (2025). Complex elections via threshold (fully) homomorphic encryption. *Cryptology ePrint Archive*, Report 2025/1482, version of 5 May 2026. https://eprint.iacr.org/2025/1482

Boudgoust, K., & Scholl, P. (2023). Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus. In *ASIACRYPT 2023*, LNCS. Springer. https://doi.org/10.1007/978-981-99-8721-4_12

Boyen, X., Haines, T., & Müller, J. (2020). A verifiable and practical lattice-based decryption mix net with external auditing. In *ESORICS 2020*, LNCS. Springer. https://doi.org/10.1007/978-3-030-59013-0_17

Boyen, X., Haines, T., & Müller, J. (2021). Epoque: Practical end-to-end verifiable post-quantum-secure e-voting. In *IEEE EuroS&P 2021* (pp. 272–291). https://doi.org/10.1109/EuroSP51992.2021.00027

Canonne, C. L., Kamath, G., & Steinke, T. (2020). The discrete Gaussian for differential privacy. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*. Full version: arXiv:2004.00010v6, 2024. https://arxiv.org/abs/2004.00010

Chillotti, I., Gama, N., Georgieva, M., & Izabachène, M. (2016). A homomorphic LWE based E-voting scheme. In *PQCrypto 2016*, LNCS. Springer. https://doi.org/10.1007/978-3-319-29360-8_16

Cho, W., Kim, J., & Lee, C. (2024). (In)security of threshold fully homomorphic encryption based on Shamir secret sharing. *Cryptology ePrint Archive*, Report 2024/1858. https://eprint.iacr.org/2024/1858

Cortier, V., Galindo, D., Küsters, R., Müller, J., & Truderung, T. (2016). SoK: Verifiability notions for e-voting protocols. In *IEEE Symposium on Security and Privacy 2016* (pp. 779–798). IEEE. https://doi.org/10.1109/SP.2016.52

Cortier, V., Lallemand, J., & Warinschi, B. (2020). Fifty shades of ballot privacy: Privacy against a malicious board. In *IEEE CSF 2020* (pp. 17–32). IEEE. https://doi.org/10.1109/CSF49147.2020.00010

Cramer, R., Franklin, M., Schoenmakers, B., & Yung, M. (1996). Multi-authority secret-ballot elections with linear work. In *EUROCRYPT 1996*, LNCS 1070 (pp. 72–83). Springer. https://doi.org/10.1007/3-540-68339-9_7

del Pino, R. (2018). *Efficient lattice-based zero-knowledge proofs and applications*. PhD thesis, PSL Research University (École normale supérieure). https://theses.hal.science/tel-02445482

del Pino, R., Katsumata, S., Maller, M., Mouhartem, F., Prest, T., & Saarinen, M.-J. (2024). Threshold Raccoon: Practical threshold signatures from standard lattice assumptions. In *EUROCRYPT 2024*, LNCS. Springer. https://doi.org/10.1007/978-3-031-58723-8_8

del Pino, R., & Lyubashevsky, V. (2017). Amortization with fewer equations for proving knowledge of small secrets. In *CRYPTO 2017*, LNCS 10403 (pp. 365–394). Springer. https://doi.org/10.1007/978-3-319-63697-9_13

del Pino, R., Lyubashevsky, V., Neven, G., & Seiler, G. (2017). Practical quantum-safe voting from lattices. In *ACM CCS 2017* (pp. 1565–1581). https://doi.org/10.1145/3133956.3134101

Ducas, L., Lyubashevsky, V., & Prest, T. (2014). Efficient identity-based encryption over NTRU lattices. In *ASIACRYPT 2014*, LNCS 8874 (pp. 22–41). Springer. https://doi.org/10.1007/978-3-662-45608-8_2

Don, J., Fehr, S., Majenz, C., & Schaffner, C. (2019). Security of the Fiat–Shamir transformation in the quantum random-oracle model. In *CRYPTO 2019*, LNCS 11693 (pp. 356–383). Springer. https://doi.org/10.1007/978-3-030-26951-7_13

Gentry, C., Halevi, S., & Lyubashevsky, V. (2022). Practical non-interactive publicly verifiable secret sharing with thousands of parties. In *EUROCRYPT 2022*, LNCS 13275 (pp. 458–487). Springer. https://doi.org/10.1007/978-3-031-06944-4_16

Gentry, C., Peikert, C., & Vaikuntanathan, V. (2008). Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008* (pp. 197–206). ACM. https://doi.org/10.1145/1374376.1374407

Griffy, S., Youdom Kemmoe, V., Nguyen, N. K., & Silde, T. (2026). Threshold lattice-based zero-knowledge proofs. *Cryptology ePrint Archive*, Report 2026/1750, version of 20 August 2026. https://eprint.iacr.org/2026/1750

Haines, T., Mosaheb, R., Müller, J., & Pryvalov, I. (2023). SoK: Secure e-voting with everlasting privacy. *Proceedings on Privacy Enhancing Technologies*, 2023(1), 279–293. https://doi.org/10.56553/popets-2023-0017

Hough, P., Sandsbråten, C., & Silde, T. (2025). More efficient lattice-based electronic voting from NTRU. *IACR Communications in Cryptology*, 1(4). https://doi.org/10.62056/a69qudhdj

Hsu, D., Kakade, S. M., & Zhang, T. (2012). A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17, paper no. 52. https://doi.org/10.1214/ECP.v17-2079

Huber, N., Küsters, R., & Reisert, P. (2026). PQKryvos: Post-quantum secure e-voting with flexible ballot formats and public tally-hiding. *Proceedings on Privacy Enhancing Technologies*, *2026*(4), 1149–1171. https://doi.org/10.56553/popets-2026-0164

Jia, H., Hu, Y., Tang, C., & Wang, L. (2024). Towards compact identity-based encryption on ideal lattices. In *CT-RSA 2024*, LNCS. Springer. https://doi.org/10.1007/978-3-031-58868-6_14. Full version: *Cryptology ePrint Archive*, Report 2024/026.

Kaim, G., Canard, S., Roux-Langlois, A., & Traoré, J. (2021). Post-quantum online voting scheme. In *FC 2021 International Workshops*, LNCS 12676 (pp. 290–305). Springer. https://doi.org/10.1007/978-3-662-63958-0_25

Kate, A., Mukherjee, P., Saleem, H., Sarkar, P., & Sinha, R. (2026). Practical post-quantum secure publicly verifiable secret sharing and applications. *Cryptology ePrint Archive*, Report 2026/813, version of 9 October 2026. https://eprint.iacr.org/2026/813

Kiltz, E., Lyubashevsky, V., & Schaffner, C. (2018). A concrete treatment of Fiat–Shamir signatures in the quantum random-oracle model. In *EUROCRYPT 2018*, LNCS 10822 (pp. 552–586). Springer. https://doi.org/10.1007/978-3-319-78372-7_18

Kim, J., Kim, S., & Lee, H. T. (2026). Revisiting Shamir secret sharing for threshold fully homomorphic encryption. *Cryptology ePrint Archive*, Report 2026/1559. https://eprint.iacr.org/2026/1559

Küsters, R., Liedtke, J., Müller, J., Rausch, D., & Vogt, A. (2020). Ordinos: A verifiable tally-hiding e-voting system. In *IEEE EuroS&P 2020* (pp. 216–235). https://doi.org/10.1109/EuroSP48549.2020.00022

Lyubashevsky, V. (2012). Lattice signatures without trapdoors. In *EUROCRYPT 2012*, LNCS 7237 (pp. 738–755). Springer. https://doi.org/10.1007/978-3-642-29011-4_43

Lyubashevsky, V., Nguyen, N. K., & Plançon, M. (2022). Lattice-based zero-knowledge proofs and

applications: Shorter, simpler, and more general. In *CRYPTO 2022*, LNCS 13508 (pp. 71–101). Springer. https://doi.org/10.1007/978-3-031-15979-4_3

Lyubashevsky, V., Nguyen, N. K., & Seiler, G. (2021). Shorter lattice-based zero-knowledge proofs via one-time commitments. In *PKC 2021*, LNCS 12710 (pp. 215–241). Springer. https://doi.org/10.1007/978-3-030-75245-3_9

Lyubashevsky, V., & Seiler, G. (2018). Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *EUROCRYPT 2018*, Part I, LNCS 10820 (pp. 204–224). Springer. https://doi.org/10.1007/978-3-319-78381-9_8

MATZOV (2022). Report on the security of LWE: Improved dual lattice attack. Technical report, version 2. Zenodo. https://doi.org/10.5281/zenodo.6493704

Maxwell, G., Poelstra, A., Seurin, Y., & Wuille, P. (2019). Simple Schnorr multi-signatures with applications to Bitcoin. *Designs, Codes and Cryptography*, 87(9), 2139–2164. https://doi.org/10.1007/s10623-019-00608-x

Micciancio, D., & Peikert, C. (2012). Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, LNCS 7237 (pp. 700–718). Springer. https://doi.org/10.1007/978-3-642-29011-4_41

Micciancio, D., & Walter, M. (2017). Gaussian sampling over the integers: Efficient, generic, constant-time. In *CRYPTO 2017*, LNCS 10402 (pp. 455–485). Springer. https://doi.org/10.1007/978-3-319-63715-0_16

Regev, O. (2009). On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6), article 34. https://doi.org/10.1145/1568318.1568324

Rivinius, M., Reisert, P., Rausch, D., & Küsters, R. (2022). Publicly accountable robust multi-party computation. In *IEEE S&P 2022* (pp. 2430–2449). Full version: *Cryptology ePrint Archive*, Report 2022/436. https://doi.org/10.1109/SP46214.2022.9833608

Schoenmakers, B. (1999). A simple publicly verifiable secret sharing scheme and its application to electronic voting. In *CRYPTO 1999*, LNCS 1666 (pp. 148–164). Springer. https://doi.org/10.1007/3-540-48405-1_10