Practical Quantum-Safe Voting from Lattices

Rafaël del Pino
IBM Research – Zurich
afe@zurich.ibm.com

Vadim Lyubashevsky
IBM Research – Zurich
vad@zurich.ibm.com

Gregory Neven
IBM Research – Zurich
nev@zurich.ibm.com

Gregor Seiler
IBM Research – Zurich
grs@zurich.ibm.com

ABSTRACT

We propose a lattice-based electronic voting scheme, EVOLVE (Electronic Voting from Lattices with Verification), which is conjectured to resist attacks by quantum computers. Our protocol involves a number of voting authorities so that vote privacy is maintained as long as at least one of the authorities is honest, while the integrity of the result is guaranteed even when all authorities collude. Furthermore, the result of the vote can be independently computed by any observer.

At the core of the protocol is the utilization of a homomorphic commitment scheme with strategically orchestrated zero-knowledge proofs: voters use approximate but efficient "Fiat-Shamir with Aborts" proofs to show the validity of their vote, while the authorities use amortized exact proofs to show that the commitments are well-formed. We also present a novel efficient zero-knowledge proof that one of two lattice-based statements is true (so-called OR proof) and a new mechanism to control the size of the randomness when applying the homomorphism to commitments.

We give concrete parameter choices to securely instantiate and evaluate the efficiency of our scheme. Our prototype implementation shows that the voters require 8 milliseconds to submit a vote of size about $20KB$ to each authority and it takes each authority 0.15 seconds per voter to create a proof that his vote was valid. The size of the vote share that each authority produces is approximately $15KB$ per voter, which we believe is well within the practical bounds for a large-scale election.

CCS CONCEPTS

• **Security and privacy → Privacy-preserving protocols**;

KEYWORDS

E-Voting, Lattices, Implementation, Zero-Knowledge, Post-Quantum

1 INTRODUCTION

Given how information technology has penetrated almost every aspect of our world, one could be surprised at the primitive state of technology used in the process that most influences society: elections. Electronic voting machines, which required voters to physically turn up at polling stations to cast their ballots on dedicated machines, saw a brief rise in popularity until the early 2000s, but many countries have since then gone back to paper-and-pencil voting amidst worries about security and reliability.

These are definitely genuine concerns. Researchers discovered serious security flaws in several models of voting machines that were used in elections in the US, the Netherlands, and Germany [19, 22]. Recent news reports on massive security breaches and suspicions of foreign meddling in national elections have only aggravated those concerns.

Nevertheless, many countries are warming up to the idea of online voting, in which voters cast their ballots using their personal devices from the comfort of their couch. A handful of countries, including Estonia, Switzerland, and Australia, are already using online voting for local and national elections, and it is quite a common tool among private organizations to elect officers and board members.

There are of course many aspects to securing an online voting system, but the underlying cryptographic protocol is obviously an important ingredient. All of the currently deployed electronic voting systems (e.g., Helios [2], the Swiss voting system [31], and the Estonian one) are based on cryptographic primitives that rely on the hardness of factoring or discrete logarithms for their security. Both of these assumptions are well-known to succumb to attacks by quantum computers, meaning that, as soon as sufficiently powerful quantum computers become available, an adversary could use them to break the vote secrecy of a past election, or to tamper with the result of an ongoing one. The threat of foreign meddling in elections gives additional reason for concern: powerful nation states may very well be the first to build quantum computers, and they may not be particularly vocal about their achievement.

Fortunately, we do have some cryptographic problems that resist attacks by quantum computers. Lattices are the most prominent one, offering a good trade-off between efficiency and security for basic primitives such as signatures and encryption. For more advanced protocols, such as those required for electronic voting, lattices tend to, however, suffer from extremely high bandwidth requirements.

The main difficulty in constructing practical lattice-based privacy schemes is the lack of efficient zero-knowledge proofs, which is an important tool in electronic voting schemes to let voters prove

CCS'17, , Oct. 30–Nov. 3, 2017, Dallas, TX, USA.

ACM ISBN ISBN 978-1-4503-4946-8/17/10…$15.00
https://doi.org/https://doi.org/10.1145/3133956.3134101

	Voter	Auth/Voter	Total Size / Voter
Time	8.5ms	0.15s	
Size	78KB	18KB	150KB

Table 1: Time and space complexity of the voting scheme with 4 authorities. Using the parameters of Section 5, each voter ouputs one OR-Proof and four commitments (one per authority), while each authority outputs one proof per voter.

that they cast a valid ballot. Most lattice-based zero-knowledge proofs are either Fiat-Shamir proofs with single-bit challenges or Stern-type proofs [32] with soundness error 2/3, which have to be repeated many times to reduce the soundness error. Amortization techniques [6, 14, 18] exist when performing thousands of proofs in parallel, but these are not very useful when each voter must prove correctness of his own vote. Lyubashevsky's "Fiat-Shamir with Aborts" technique [25] yields much more efficient proofs with large challenges, but only allows to prove correctness of the statement up to a small multiple of the witnesses, which could be quite detrimental in the context of voting, as it may allow an attacker to inflate the weight of this vote.

The only quantum-safe voting protocol that we are aware of [11] therefore shuns zero-knowledge proofs completely and uses fully-homomorphic encryption [21] instead. The paper doesn't give any implementation details or concrete parameter choices, so it's hard to make statements about efficiency, but due to the "heavy machinery" being utilized, chances are that the protocol is not efficient enough for medium to large-scale elections.

1.1 Our Contributions

In this paper, we present a new lattice-based electronic voting scheme that *does* use zero-knowledge proofs, but overcomes their inefficiencies by re-organizing the proofs so that the voting authorities assist the voters by performing amortized proofs. Our protocol provably guarantees vote privacy as long as one of a number of voting authorities is honest, and guarantees consistency (i.e., that honest votes are correctly counted) even if all voting authorities are corrupt, all under standard lattice-based assumptions in the random-oracle model. We suggest concrete choices for the security parameters and implement a prototype of our protocol. Our experimental results (Table 1) show that voters need less than 10ms to cast a vote and a complete bulletin (including all commitments and votes) is of size $150KB$, though the proof and verification time is higher for the authorities these computations can be done after the vote and are easily parallelizable, which we think is well within practical bounds for a large-scale election.

To better understand the technical hurdles to obtain this result, we briefly sketch a voting protocol by Cramer et al. [15] on which our protocol is based. Let's say there are N_V voters and N_A voting authorities that assist in a binary election, i.e., where each voter votes zero or one and the result is the sum of the votes. Let's also say that there is a public bulletin board where voters can post their ballots. The authorities jointly compute the tally and post the result of the election, together with a proof of correctness. The goal is to obtain vote privacy, meaning that as long as one authority is honest, the adversary does not learn anything more about the votes of honest voters than what is already implied by the result, as well as consistency and universal verifiability, meaning that anyone can check that all honest votes were counted correctly, even if all authorities collude to rig the election.

The protocol of Cramer et al. [15] begins by letting each voter secret-share his vote among the N_A authorities and commit to each of the shares. The voter sends the share and the opening information to each authority, and performs an OR-proof [13] to show that he secret-shared a zero-or-one vote by exploiting a homomorphism in the commitment scheme. When the voting phase closes, all servers check the openings of the commitments they received. Each authority then publishes the sum of all the shares it received together with valid opening information, again using the homomorphism in the commitments. The result of the election is the sum of all these partial sums.

There are a number of hurdles to overcome when translating this approach into lattice-based primitives. The first is that, as discussed above, lattice-based zero-knowledge proofs are either inefficient or approximate, while amortization doesn't help for proofs by individual voters. The second is that commitments typically use short vectors as randomness (i.e., opening information), but applying homomorphisms accumulates the size of this randomness, which must be compensated for by choosing larger parameters, which comes at a big cost in efficiency. The third problem is that the typical OR-proof technique [13] of XOR-ing challenge values doesn't work for lattices, because challenges are polynomials with small coefficients in a ring, but do not form a group among them.

We address the first problem by strategically splitting up the burden of the proofs between voters and authorities. Namely, we let voters prove that they secret-shared a zero-or-one vote using approximate proofs, but we let the authorities prove that the commitment they received is well-formed, i.e., has short opening information. The authorities do so for all voters simultaneously, so they can use the more efficient amortized proofs [6, 14, 18].

The second problem we address by letting authorities re-commit to the sum of batches of votes, and by letting them prove in zero knowledge that the new commitment indeed contains the sum of all votes in the batch. By repetitively applying this technique, each authority can keep the randomness growth within bounds, so that it eventually ends up with a commitment to the sum of all received shares with short randomness.

Finally, we solve the problem with the OR proofs by not using polynomials with short coefficients as challenges, but rather permutations over the indices of a polynomial. These permutations do form a group, so that they can be used to build efficient OR proofs.

The proofs outlined above are constructed using quantum-secure building blocks via the Fiat-Shamir transform. While there is a known *classical* reduction from hard lattice problems to schemes constructed in this manner, there is no quantum reduction known. The underlying reason as to why a general proof is unlikely to come is due to the fact that classically-secure *computationally binding* commitments are not known to be binding for a quantum committer (c.f. [16]). Nevertheless, there are known quantum security proofs in the QROM for Fiat-Shamir schemes of the same form as ours, but in which the parameters are set differently [34]. Furthermore, there are currently no known natural counter-examples of Fiat-Shamir zero-knowledge proofs (nor of commitment schemes) which are based on quantum-hard problems via classical reductions, but are broken by quantum adversaries. It therefore seems reasonable to assume that such Fiat-Shamir schemes are secure. If one would like to have a reduction that is in the QROM, one could instantiate the schemes as in [1] and then use the reduction in [34]. This would, however, lead to a noticeable increase in the size of the proofs and public keys.

1.2 Overview of the Cryptographic Tools

Commitment Scheme with "Unbounded" Messages Sizes. We review the lattice-based instantiation of a generic commitment scheme from [17] to lattices [7].

To commit to an integer $x \in \mathbb{Z}_q$, we first interpret it as a polynomial in $\mathcal{R}_q$ whose constant coefficient is $\mathbf{x}$ and all others are 0, then pick a random vector of polynomials $\mathbf{r}$ having small coefficients, and output

$$\begin{bmatrix} \mathbf{a} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ x \end{bmatrix}$$

where the public commitment key $\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$ is a matrix of uniformly random polynomials. To open a commitment $\begin{bmatrix} \mathbf{a} \\ b \end{bmatrix}$ one simply computes the inverse operation

$$\begin{bmatrix} \mathbf{0} \\ x \end{bmatrix} = \begin{bmatrix} \mathbf{a} \\ b \end{bmatrix} - \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r}$$

This commitment is binding if finding small solutions to the equation $\mathbf{Ar} = \mathbf{0} \mod q$ is hard (since opening a commitment to two different values implies finding small $\mathbf{r}$ and $\mathbf{r}'$ such that $\mathbf{Ar} = \mathbf{a} = \mathbf{Ar}'$), this problem is called M-SIS (Module Short Integer Solution). The commitment is hiding if distinguishing $\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r}$ from uniform is hard, we will show that this problem can be reduced to the M-LWE problem (Module Learning With Error). A useful property of this scheme is that it's additively-homomorphic, i.e. the sum of two commitments is a commitment to the sum of the messages using the sum of the randomnesses, however one cannot sum an arbitrary number of commitments as finding a solution to $\mathbf{Ar} = \mathbf{0} \mod q$ (and thus breaking the binding property) becomes easier as the coefficients of $\mathbf{r}$ grow.

Approximate Zero-Knowledge Proofs. Suppose that $\mathbf{A}$ is a matrix over $R^{k \times \ell}$ and $\mathbf{t}$ is an element in $\mathcal{R}^k$ such that there exists an $\mathbf{s}$ with small coefficients satisfying $\mathbf{As} = \mathbf{t}$.[1] Then it is possible to very efficiently prove in zero-knowledge using the "Fiat-Shamir with Aborts" technique that there exist $\bar{\mathbf{s}}$ and $\bar{c}$ that satisfy $\mathbf{A}\bar{\mathbf{s}} = \bar{c}\mathbf{t}$. The coefficients of $\bar{\mathbf{s}}$ are somewhat larger than those of $\mathbf{s}$ and the coefficients of $\bar{c}$ are very small. Notice that such proofs are not homomorphic. In other words giving giving a proof for $\mathbf{t}_1$ and $\mathbf{t}_2$ does not give a proof for $\mathbf{t}_1 + \mathbf{t}_2$ because the $\bar{c}_i$ could be different. This is the main reason why the voters cannot simply use these types of proofs by themselves to commit to their votes.

Amortized Exact Zero-Knowledge Proofs. If we would like to have proofs compose homomorphically, we would need to prove the knowledge of $\bar{\mathbf{s}}$ such that $\mathbf{A}\bar{\mathbf{s}} = \gamma\mathbf{t}$ for some fixed γ. While this is quite inefficient for one proof, it can be in fact made very efficient when needing to prove many such relations simultaneously [6, 14, 18]. If one has a very large number of relations, then one can have $\gamma = 1$. It was shown in [18] that one could apply the improved zero-knowledge proof from [9] to efficiently prove these linear relations with $\gamma = 2$ when having access to only around a few thousand relations for security parameter 256. In our voting scheme, this is the proof system that the authorities will use when proving that the randomness used by the voters' commitment scheme contains small coefficients.

Approximate Zero-Knowledge Proof of an OR of Two Statements. A classic construction to prove that an element y belongs in either a language $\mathcal{L}(\Re_0)$ or a language $\mathcal{L}(\Re_1)$ is as follows. The prover will prove both that $y \in \mathcal{L}(\Re_0)$ and that $y \in \mathcal{L}(\Re_1)$ but he will cheat in one of the proofs. If the verifier can make sure that the prover cheats in exactly one of the proofs without knowing which one he will effectively be convinced that $y \in \mathcal{L}(\Re_0) \cup L(\Re_1)$. Suppose that the prover P knows a witness w_0 of the fact that $y \in \mathcal{L}(\Re_0)$, P can choose a challenge c_1 and create a fake proof that $y \in \mathcal{L}(\Re_1)$. He then starts an honest proof that $y \in \mathcal{L}(\Re_0)$ and sends its commitment along with the one from the fake proof to the verifier, the verifier answers with a challenge c, P finishes his honest proof but using the challenge $c_0 = c - c_1$ instead of c, he then sends the responses for both proofs. The verifier receives two valid proofs but he cannot guess which one is a fake as he knows that $c_0 + c_1 = c$ but he does not know which of the two challenges was created first. This proof is sound because once c_1 and c are fixed the verifier has no degree of freedom on the choice of c_0, it is Zero-Knowledge because the verifier cannot distinguish the difference of two challenges $(c - c_1)$ from a challenge (c_0). In lattice-based Zero-Knowledge the challenge is taken to be a small polynomial (e.g. with coefficients in $\{0, 1\}$) meaning that one can easily distinguish between a challenge and the difference of two challenges (which will have coefficients in $\{-1, 0, 1\}$), hence such a proof would clearly not be zero knowledge. This problem can be mended by using larger challenges and restarting the protocol if the difference of the challenges is not in the right set (i.e. by using rejection sampling), this is the approach taken in [29] but it results in very inefficient proofs. Note however that the challenge c sent by the verifier does not need to come from the same set as c_0 and c_1, what one needs is that for a bit $b \in \{0, 1\}$ and any couple (c_b, c) the value c_{1-b} is uniquely fixed (i.e. the prover can cheat on at most one proof) and that c induces a random permutation over the space of challenges (this way the verifier cannot know whether c_0 or c_1 was created first). With this in mind if the challenge space for c_0 and c_1 is e.g. the set of all polynomials with binary coefficients and fixed hamming weight (we will use a slightly larger but similar challenge space), then taking c to be a random permutation over the coefficients of these polynomials is a much better solution. In doing so we obtain efficient Or-Proofs which we will use to prove that commitments open to 0 or 1, however these proofs are still approximate. That is, they prove knowledge of $\bar{\mathbf{r}}$ with small coefficients and a polynomial

[1] The parameters that we will use in this paper will typically have $k \approx 7$ and $\ell \approx 2k$ with the ring R being $\mathbb{Z}_q[X]/(X^{256} + 1)$.

$\bar f$ with very small coefficients such that

$$\bar f \begin{bmatrix}\mathbf{a}\\ b\end{bmatrix} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}\bar{\mathbf{r}} + \bar f\begin{bmatrix}\mathbf{0}\\ x\end{bmatrix}, \text{ and } x \in \{0,1\}.$$

1.3 Overview of the Construction

We will now explain how we utilize the above building blocks to build a voting scheme.

We will make the convention that the voters (and information pertaining to the voters) are numbered 1 through N_V using a subscript, whereas the information pertaining to the authorities is numbered 1 through N_A and is labeled using a parenthesized superscript. In particular, for elements $x_i^{(j)}$, we will define $x_i = \sum\limits_{j=1}^{N_A} x_i^{(j)}$, $x^{(j)} = \sum\limits_{i=1}^{N_V} x_i^{(j)}$, and $x = \sum\limits_{i=1}^{N_V}\sum\limits_{j=1}^{N_A} x_i^{(j)} = \sum\limits_{j=1}^{N_A}\sum\limits_{i=1}^{N_V} x_i^{(j)}$.

A voter i who wishes to cast a vote v_i (which is 0 or 1), first splits v_i into N_A parts $v_i^{(j)}$ where the first $N_A - 1$ of them are chosen uniformly random modulo q and the last share $v_i^{(N_A)}$ is chosen such that $\sum\limits_{j=1}^{N_A} v_i^{(j)} = v_i (\operatorname{mod} q)$. Each $v_i^{(j)}$ is then interpreted as a polynomial in the ring $\mathcal{R}_q$ whose constant coefficient is $v_i^{(j)}$ and the other coefficients are 0. The voter i then uses the commitment scheme to commit to each share $v_i^{(j)}$ as

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \mathbf{r}_i^{(j)} + \begin{bmatrix}\mathbf{0}\\ v_i^{(j)}\end{bmatrix} = \begin{bmatrix}\mathbf{a}_i^{(j)}\\ b_i^{(j)}\end{bmatrix}. \tag{1}$$

All the commitments are published to the bulletin board. Note that because the commitment scheme is additively homomorphic, we have

$$\sum_{j=1}^{N_A}\begin{bmatrix}\mathbf{a}_i^{(j)}\\ b_i^{(j)}\end{bmatrix} = \begin{bmatrix}\mathbf{a}_i\\ b_i\end{bmatrix} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \mathbf{r}_i + \begin{bmatrix}\mathbf{0}\\ v_i\end{bmatrix},$$

which is a valid commitment to v_i (but with slightly larger randomness $\mathbf{r}_i$). Voter i now creates a zero-knowledge OR-proof that he has knowledge of a vector $\bar{\mathbf{r}}_i$ with small coefficients and a ring element $\bar f_i$ with very small coefficients such that

$$\begin{bmatrix}\mathbf{a}_i\\ b_i\end{bmatrix} \cdot \bar f_i = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \bar{\mathbf{r}}_i + \begin{bmatrix}\mathbf{0}\\ v_i \cdot \bar f_i\end{bmatrix}, \text{ and } v_i \in \{0,1\}. \tag{2}$$

This proof π_i^V also gets posted to the bulletin board.

Each voter now sends to authority j the encryption (under authority j's public key) of the share $v_i^{(j)}$ and the randomness under which this share was committed $\mathbf{r}_i^{(j)}$ from (1) (one can alternatively think that the voters simply post this encryption to the bulletin board). Upon receiving all such encryptions from every voter, authority j needs to create a proof of knowledge that the $\mathbf{r}_i^{(j)}$ all have small coefficients. He uses the Amortized Exact Zero-Knowledge proof to create proofs $\pi_{i,j}^A$ that prove the knowledge of $\hat{\mathbf{r}}_i^{(j)}$ that satisfy

$$\mathbf{A} \cdot \hat{\mathbf{r}}_i^{(j)} = 2\mathbf{a}_i^{(j)}. \tag{3}$$

If all N_A authorities provide proofs of the above statement, then using the additive homomorphism of the commitment scheme, we obtain a proof of knowledge of an $\hat{\mathbf{r}}_i$ such that

$$2\mathbf{a}_i = \sum_{j=1}^{N_A} 2\mathbf{a}_i^{(j)} = \sum_{j=1}^{N_A} \mathbf{A} \cdot \hat{\mathbf{r}}_i^{(j)} = \mathbf{A} \cdot \hat{\mathbf{r}}_i. \tag{4}$$

Combining this with (2), implies that $\mathbf{A} \cdot (2\bar{\mathbf{r}}_i - \bar f_i \hat{\mathbf{r}}_i) = \mathbf{0}$. Based on the hardness of the M-SIS problem, this implies that $2\bar{\mathbf{r}}_i = \bar f_i \hat{\mathbf{r}}_i$. One can then rewrite (2) as

$$2 \cdot \begin{bmatrix}\mathbf{a}_i\\ b_i\end{bmatrix} \cdot \bar f_i = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \bar f_i \hat{\mathbf{r}}_i + \begin{bmatrix}\mathbf{0}\\ 2v_i \cdot \bar f_i\end{bmatrix},$$

and since we choose the challenge set such that $\bar f_i$ is invertible in $\mathcal{R}_q$, we can divide by $\bar f_i$ to finally obtain

$$2 \cdot \begin{bmatrix}\mathbf{a}_i\\ b_i\end{bmatrix} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \hat{\mathbf{r}}_i + \begin{bmatrix}\mathbf{0}\\ 2m_i\end{bmatrix}, \text{ and } v_i \in \{0,1\}. \tag{5}$$

Because there is no longer the factor $\bar f_i$ which could be distinct for every voter, the commitment in (5) is additively homomorphic. In particular, if we compute

$$2 \cdot \sum_{i=1}^{N_V}\begin{bmatrix}\mathbf{a}_i\\ b_i\end{bmatrix} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \hat{\mathbf{r}} + \begin{bmatrix}\mathbf{0}\\ 2\sum\limits_{i=1}^{N_V} v_i\end{bmatrix}, \text{ and } v_i \in \{0,1\}, \tag{6}$$

and $\hat r$ is a vector with small coefficients, then the quantity

$$2 \cdot \sum_{i=1}^{N_V}\begin{bmatrix}\mathbf{a}_i\\ b_i\end{bmatrix}$$

is a commitment to twice the total number of 1-votes that have been cast. If there are many voters, then $\hat r = \sum_i \bar r_i$ is not small, but we show how to handle this issue later.

For universal verifiability, we therefore would like the value of $\hat{\mathbf{r}}$ to be publicly computable. For this to happen, each authority simply computes $\sum\limits_{i=1}^{N_V} \mathbf{r}_i^{(j)} = \mathbf{r}^{(j)}$ and reveals it by publishing it to the bulletin board. Any verifier can simply check that

$$\mathbf{A} \cdot \mathbf{r}^{(j)} = \mathbf{a}^{(j)}. \tag{7}$$

We now claim that it must be that

$$\mathbf{r} = \sum_{j=1}^{N_A} \mathbf{r}^{(j)} = 2\hat{\mathbf{r}}.$$

From (3), we know that

$$\mathbf{A} \cdot \hat{\mathbf{r}}^{(j)} = \sum_{i=1}^{N_V} \mathbf{A} \cdot \hat{\mathbf{r}}_i^{(j)} = 2\mathbf{a}^{(j)}.$$

Combining this with (7) implies that $\mathbf{A}\hat{\mathbf{r}}^{(j)} = \mathbf{A} \cdot (2\mathbf{r}^{(j)})$. Unless one can break the M-SIS problem, it must be that $\hat{\mathbf{r}}^{(j)} = 2\mathbf{r}^{(j)}$, and then we have

$$\hat{\mathbf{r}} = \sum_{j=1}^{N_A} \hat{\mathbf{r}}^{(j)} = 2\sum_{j=1}^{N_A} \mathbf{r}^{(j)} = 2\mathbf{r}.$$

Plugging the above into (6) and dividing by 2 implies that

$$\begin{bmatrix}\mathbf{a}\\ b\end{bmatrix} = \sum_{i=1}^{N_V}\begin{bmatrix}\mathbf{a}_i\\ b_i\end{bmatrix} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \mathbf{r} + \begin{bmatrix}\mathbf{0}\\ \sum\limits_{i=1}^{N_V} v_i\end{bmatrix}, \text{ and } v_i \in \{0,1\}. \tag{8}$$

If $\mathbf{r}$ is small enough, then the above implies that $\begin{bmatrix}\mathbf{a}\\ b\end{bmatrix}$ is a commitment to the full vote tally $\sum\limits_{i=1}^{N_V} v_i$, and one can obtain this tally by computing $b - \mathbf{B} \cdot \mathbf{r}$. As long as there are fewer than q voters, we can exactly recover $\sum\limits_{i=1}^{N_V} v_i$ over the integers.

1.3.1 Reducing the Randomness. An issue that we still need to deal with is how to make sure that the randomness, when summed over all the voters, does not grow too much. This is crucial in order for the final commitment in (6) to be meaningful. A trivial way to accomplish this is to simply set the parameters large enough so that a large set of voters can be accommodated. This is an extremely impractical solution that we would like to avoid.

The way that we can overcome this issue is by making the Authorities create votes of 0 using randomness that is close to the randomnesses used by the individual voters. For example, if voters $1, \ldots, l$ whose commitments to Authority j are $\begin{bmatrix}\mathbf{a}_1^{(j)}\\ b_1^{(j)}\end{bmatrix}, \ldots, \begin{bmatrix}\mathbf{a}_l^{(j)}\\ b_l^{(j)}\end{bmatrix}$ under randomnesses $\mathbf{r}_1^{(j)}, \ldots, \mathbf{r}_l^{(j)}$, then the Authority can create a vote of 0 using randomness $\mathbf{r}' = \mathbf{r} - \sum_i \mathbf{r}_i^{(j)}$, where $\mathbf{r}$ is a fresh randomness that comes from the same distribution as the commitment randomnesses. The Authority would then publish the "vote" $\begin{bmatrix}\mathbf{a}'\\ b'\end{bmatrix}$, prove that it's a zero-vote by proving that there exists some $\hat{\mathbf{r}}'$ such that

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \hat{\mathbf{r}}' = 2 \begin{bmatrix}\mathbf{a}'\\ b'\end{bmatrix}, \tag{9}$$

and also prove that there exists a small $\hat{\mathbf{r}}$ such that

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \hat{\mathbf{r}} = 2 \cdot \left(\begin{bmatrix}\mathbf{a}'\\ b'\end{bmatrix} + \sum_i \begin{bmatrix}\mathbf{a}_i^{(j)}\\ b_i^{(j)}\end{bmatrix}\right) \tag{10}$$

The proof in (10) can be "amortized-in" with the proofs $\pi_{i,j}^A$ because the size of the randomness is the same. The proof of (9), however, contains larger randomness, and so such proofs should be amortized only among themselves.[2]

Notice that because the randomness of the sum of the commitments $\begin{bmatrix}\mathbf{a}'\\ b'\end{bmatrix} + \sum_i \begin{bmatrix}\mathbf{a}_i^{(j)}\\ b_i^{(j)}\end{bmatrix} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \mathbf{r}$ is as small as in one voter commitment, we have effectively reduced the size of the sum of the randomness in l voter commitments to that of one commitment. If we do this for every block of l voters, then we effectively reduced the sum of the randomness by a factor of l.

It's easy to see that this procedure is repeatable. Once every l blocks of votes have small randomness, we can consider repeating this procedure by summing over l such blocks. This will effectively reduce the total sum of the randomnesses by another factor of l. If we continue this procedure, then we will be effectively adding $2(N_V/l + N_V/l^2 + \ldots) \approx 2N_V/(l-1)$ extra proofs. The advantage will be that we now only need to worry about the randomness growing by a factor l for the proof in (9). We set our parameters so that $l = 30$.

One can also think of the above procedure as the authority summing up 30 votes, recommitting to the sum using fresh, small randomness, and then giving a proof that he correctly recommitted to the sum of the votes. In particular, proving that the difference between his new commitment and the sum of the 30 commitments is a commitment to 0.

2 PRELIMINARIES

2.1 Notations

We fix n and q to be integers throughout this paper, we denote by $\mathbb{Z}_q$ the integers modulo q, which we will represent between $-\left\lfloor \frac{q-1}{2} \right\rfloor$ and $\left\lfloor \frac{q+1}{2} \right\rfloor$. We will denote by $\mathcal{R}$ the polynomial ring $\mathbb{Z}[X]/X^n + 1$, and by $\mathcal{R}_q$ the quotient ring $\mathcal{R}/q\mathcal{R}$. Elements in $\mathbb{Z}, \mathbb{R}, \mathcal{R}$, or $\mathcal{R}_q$ will be written in lower case, vectors over $\mathbb{Z}, \mathbb{R}, \mathcal{R}$, or $\mathcal{R}_q$ will be written in bold face lower case, and matrices will be in bold face upper case. The Euclidean norm is denoted as $\|\cdot\|$, the infinity norm and l_1 norm are denoted respectively as $\|\cdot\|_\infty$, and $\|\cdot\|_1$. All norms can be extended to a polynomial $f = \sum_0^{n-1} f_i X^i$ by considering the vector $(f_0, \ldots, f_{n-1})$ of its coefficients (if said coefficients are in $\mathbb{Z}_q$ we consider their representative that lies between $-\left\lfloor \frac{q-1}{2} \right\rfloor$ and $\left\lfloor \frac{q+1}{2} \right\rfloor$). For a set $\mathcal{S}$ we write $s \xleftarrow{\$} \mathcal{S}$ to denote that s was sampled at random from $\mathcal{S}$, for a distribution χ we write $s \leftarrow \chi$ to denote that s was sampled from χ.

2.2 E-Voting Schemes

We base our syntax and security definitions of one-pass electronic voting schemes on that of Bernhard et al. [10], but there are some differences. First, we explicitly model a multi-authority setting where each authority independently generates its own keys. Second, ballot testing in our scheme must be performed by the authorities, who use their secret key in the process. Verification of the entire election can still be done publicly, though. As ballot testing, tallying, and verification do not require interaction with the voters, the authorities do not need to be online for the first part of the election and only need to run the ballot testing and tallying algorithms once all the ballots have been cast.

A multi-authority electronic voting scheme $\mathcal{EV}$ is a tuple (**Setup**, **ASetup**, **Vote**, **TestB**, **Tally**, **Verify**) of algorithms and protocols that are used by authorities $\mathcal{A}_1, \ldots, \mathcal{A}_{N_A}$ and voters with identities $id \in \mathbb{I}$ as follows. We consider binary elections (we consider the more general case of votes in $\{0,1\}^k$ for $k \geq 1$ in Appendix B), i.e., where each voter $id \in \mathbb{I}$ casts a vote $v_{id} \in \{0,1\}$ and the result of the election is $r = \sum_{id \in \mathbb{I}} v_{id}$. We assume that all voters and authorities have read access and authenticated append-only write access to a public bulletin board *BB*, meaning that entries can only be appended to the board and entries are authenticated (e.g., signed) under the writer's identity. Moreover, each voter can only write once to the bulletin board; authorities can write as often as they want.

- **Setup**(λ) generates trusted common parameters *par*.
- **ASetup**(*par*) is used by authority $\mathcal{A}_j$ to generate a public key pk_j and corresponding secret key sk_j.

[2] It is also possible to do these proofs in a non-amortized fashion and only get an approximate proof. In this case, depending on the value of l, the parameters may have to be increased.

- **Vote**$(par, pk_1, \ldots, pk_{N_A}, id, v)$ is used by voter $id \in \mathbb{I}$ to cast his vote $v \in \{0, 1\}$. It returns a ballot b that the voter posts on the bulletin board BB.
- **TestB**$(par, pk_1, \ldots, pk_{N_A}, sk_j, b)$ allows authority $\mathcal{A}_j$ to test whether ballot b is valid or not by returning 1 or 0, respectively. The ballot is only considered valid after all N_A authorities confirm its validity on the bulletin board BB. This check can be performed as the votes come in, or only after the voting phase has ended. The tallying authorities therefore do not have to be online during the voting phase: rather than interacting directly with the voters, the tallying authorities can obtain the ballots from the bulletin board after voting has ended and discard invalid ballots if needed.
- **Tally**$(par, pk_1, \ldots, pk_{N_A}, BB, sk_j)$ is an interactive protocol run among the authorities $\mathcal{A}_j$, $j = 1, \ldots, N_A$, at the end of which they announce the tally r and proof Π.
- **Verify**$(par, pk_1, \ldots, pk_{N_A}, BB, r, \Pi)$ can be run by anyone to check the correctness of the election result.

Correctness. Correctness guarantees that, when all parties behave honestly, all ballots are deemed valid and the result of the election is correct. Let $id_1, \ldots, id_{N_V} \in \mathbb{I}$ be voter identities and $v_1, \ldots, v_{N_V}$ be their respective votes. Let $par \xleftarrow{\$} \textbf{Setup}(1^\lambda)$; and $(pk_j, sk_j) \xleftarrow{\$} \textbf{ASetup}(par)$ for $j = 1, \ldots, N_A$. For $i = 1, \ldots, N_V$ and $j = 1, \ldots, N_A$ let $b_i \xleftarrow{\$} \textbf{Vote}(par, pk_1, \ldots, pk_{N_A}, id_i, v_i)$, $BB[i] \leftarrow b_i$, and let (r, Π) be the outcome of the protocol when each authority $\mathcal{A}_j$ runs **Tally**$(par, pk_1, \ldots, pk_{N_A}, BB, sk_j)$, $j = 1, \ldots, N_A$. The scheme is correct if for all $i = 1, \ldots, N_V$ and $j = 1, \ldots, N_A$, the following conditions hold with overwhelming probability: $r = \sum_{i=1}^{N_V} v_i$, **TestB**$(par, pk_1, \ldots, pk_{N_A}, sk_j, b_i) = 1$, and **Verify**$(par, pk_1, \ldots, pk_{N_A}, BB, r, \Pi) = 1$.

Privacy. Privacy requires that an adversary who corrupts $N_A - 1$ authorities and an arbitrary number of voters does not learn anything more about the votes of honest voters than what is revealed by the election result. The single-authority BPRIV notion of Bernhard et al. [10] defines this by requiring that the adversary cannot tell a bulletin board for a first set of votes with the real election result and proof from a bulletin board for a second set of votes with the same result and a simulated proof. The BPRIV notion is not easily adapted to the multi-authority setting, because the corrupt authorities would have to be involved in computing the tally for both bulletin boards. We therefore adapt the notion to require that the adversary cannot distinguish between the bulletin boards of two different sets of votes, as long as both sets of votes yield the same election result, i.e., have the same total number of zero-votes and one-votes.

The advantage of an adversary $\mathcal{A}$ in breaking the privacy of the electronic voting scheme $\mathcal{EV}$ is defined through the experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{priv},\beta}$ below as

$$\mathbf{Adv}_{\mathcal{A}}^{\text{priv}}(\lambda) \;=\; \left| \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{priv},0}(\lambda) = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{priv},1}(\lambda) = 1] \right| ,$$

where $\mathcal{A}$ is given access to all oracles in the set $\mathcal{O} = \{\mathcal{O}\textbf{Vote}, \mathcal{O}\textbf{Cast}, \mathcal{O}\textbf{Tally}\}$ as well as read and append-only write access to the bulletin board BB. The $\mathcal{O}$**Vote** and $\mathcal{O}$**Cast** oracles can be queried as many times as $\mathcal{A}$ wants, but the $\mathcal{O}$**Tally** oracle can only be queried once.

```
Experiment $Exp_{\mathcal{A}}^{\text{priv},\beta}(\lambda)$:
    $par \xleftarrow{\$} \textbf{Setup}(1^\lambda)$ ; $pk_1 \xleftarrow{\$} \textbf{ASetup}(par)$ ; $HV \leftarrow \emptyset$
    $(pk_2, \ldots, pk_{N_A}, st) \xleftarrow{\$} \mathcal{A}(par, pk_1)$
    $\beta' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}, BB}(st)$
    $HV' \leftarrow \{(id, v_0, v_1, b) \in HV$ :
        $\forall j \in \{1, \ldots, N_A\}$ : "$\mathcal{A}_j$ approves $b$"$\in BB\}$
    $V_0 \leftarrow \sum_{(id, v_0, v_1, b) \in HV'} v_0$ ; $V_1 \leftarrow \sum_{(id, v_0, v_1, b) \in HV'} v_1$
    If $V_0 \neq V_1$ then return $\perp$ else return $\beta'$

Oracle $\mathcal{O}\textbf{Vote}(id, v_0, v_1)$:
    $b \xleftarrow{\$} \textbf{Vote}(par, pk_1, \ldots, pk_{N_A}, id, v_\beta)$
    $HV \leftarrow HV \cup \{(id, v_0, v_1, b)\}$
    $BB \leftarrow BB \| $"$id$ casts $b$"
    If $\textbf{TestB}(par, pk_1, \ldots, pk_{N_A}, sk_1, b) = 1$
        then $BB \leftarrow BB \| $"$\mathcal{A}_1$ approves $b$"
        else $BB \leftarrow BB \| $"$\mathcal{A}_1$ rejects $b$"

Oracle $\mathcal{O}\textbf{Cast}(id, b)$:
    If "$id$ casts $b$" $\in BB$ and $\textbf{TestB}(par, pk_1, \ldots, pk_{N_A}, sk_1, b) = 1$
        then $BB \leftarrow BB \| $"$\mathcal{A}_1$ approves $b$"
        else $BB \leftarrow BB \| $"$\mathcal{A}_1$ rejects $b$"

Oracle $\mathcal{O}\textbf{Tally}$:
    Run $\textbf{Tally}(par, pk_1, \ldots, pk_{N_A}, BB, sk_1)$ with $\mathcal{A}$ to obtain $(r, \Pi)$
    Return $(r, \Pi)$
```

Consistency. Consistency requires that the election result is correct with respect to the votes cast by voters. Bernhard et al.'s notion of strong consistency [10] requires that individual ballots can be extracted online. We relax this notion by requiring that, if an election finishes successfully, the result must be "realistic" with respect to the honestly cast votes. Meaning, the result must be at least the number of honest one-votes and at most the total number of votes cast minus the number of honest zero-votes. We strengthen the notion, however, by requiring that this property holds even against corrupt election authorities. Intuitively, our notion is similar to the quantitative verifiability goal of Cortier et al. [12]; a formal analysis and comparison would require further work.

Formally, the advantage of an adversary $\mathcal{A}$ in breaking the consistency of $\mathcal{EV}$ is defined through the consistency experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{cons}}$ below as

$$\mathbf{Adv}_{\mathcal{A}}^{\text{cons}}(\lambda) \;=\; \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{cons}}(\lambda) = 1] .$$

```
Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{cons}}(\lambda)$:
    $par \xleftarrow{\$} \textbf{Setup}(1^\lambda)$ ; $HV \leftarrow \emptyset$
    $(pk_1, \ldots, pk_{N_A}, st) \xleftarrow{\$} \mathcal{A}(par)$
    $(r, \Pi) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}\textbf{Vote}, BB}(st)$
    $HV' \leftarrow \{(id, v, b) \in HV$ :
        $\forall j \in \{1, \ldots, N_A\}$ : "$\mathcal{A}_j$ approves $b$"$\in BB\}$
    $h_0 \leftarrow |\{(id, 0, b) \in HV'\}|$ ; $h_1 \leftarrow |\{(id, 1, b) \in HV'\}|$
    $t \leftarrow |\{b$ : $\forall j \in \{1, \ldots, N_A\}$ : "$\mathcal{A}_j$ approves $b$"$\in BB\}|$
    If $\textbf{Verify}(par, pk_1, \ldots, pk_{N_A}, BB, r, \Pi) = 1$
        and $(r < h_1$ or $r > t - h_0)$
        then return 1 else return 0

Oracle $\mathcal{O}\textbf{Vote}(id, v)$:
    $b \xleftarrow{\$} \textbf{Vote}(par, pk_1, \ldots, pk_{N_A}, id, v)$
    $HV \longleftarrow HV \cup (id, v, b)$
    $BB \longleftarrow BB||"id \text{ casts } b"$
```

2.3 Homomorphic Commitments

Our first building block will be a commitment scheme, which will allow voters to commit to their vote. Additionally using an homomorphic commitment scheme (i.e. one in which the sum of two commitments is a commitment to the sum of the associated messages) will allow our voting protocol to use the sum of all of the voters' commitments as a commitment to the result of the election. A commitment scheme is a triple (**KeyGen**, **Com**, **Open**) such that:

- $K \longleftarrow \mathbf{KeyGen}(1^{\lambda})$ generates the public commitment key.
- $(c, d) \longleftarrow \mathbf{Com_K}(m)$ generates the commitment c and opening d for message m
- $m' \longleftarrow \mathbf{Open_K}(c, d)$ opens the commitment c using the opening d (potentially $m' = \perp$ if d is not a valid opening of c)

We will ignore the subscript K on the commitment and opening algorithm when the key is clear from the context and we will denote by $c = \mathbf{Com}(m; d)$ the commitment of m with opening d. We consider computationally binding/hiding commitment schemes, i.e. schemes with the following properties:

- **Correctness :** For any message m $\mathbf{Open_K}(\mathbf{Com_K}(m)) = m$ with overwhelming probability.
- **Computational Hiding :** A commitment hides the commited message. It is computationally hard for any PPT adversary $\mathcal{A}$ to generate messages m_0, m_1 such that $\mathcal{A}$ can distinguish between $\mathbf{Com}_K(m_0)$ and $\mathbf{Com}_K(m_1)$.
- **Computational Binding :** A commitment cannot be opened to two messages. It is computationally hard for any PPT adversary $\mathcal{A}$ to generate a triple (c, d, d') such that (c, d) opens to m and (c, d') opens to m' for $m \neq m'$.

We will consider additively homomorphic commitment schemes, i.e. schemes such that if $\mathbf{Open_K}(c, d) = m$ and $\mathbf{Open_K}(c', d') = m'$, then $\mathbf{Open_K}(c + c', d + d') = m + m'$ (this property will be restricted to openings with small enough norm, c.f Section 3.1).

2.4 Zero-Knowledge Proofs of Knowledge

Our second building block will be proofs of knowledge, they will be used to prove that the voters commited to correct votes. We will consider relaxed ZKPoK (Zero-Knowledge Proof of Knowledge) as defined in [9]. They differ from standard Σ-protocols in that the soundness extractor recovers a witness that lies in a somewhat larger language than the one used for the secret.

Definition 2.1. Let $S = (P, V)$ be a two-party protocol, where V is PPT, and let $\mathfrak{R}, \mathfrak{R}'$ be binary relations such that $\mathfrak{R} \subset \mathfrak{R}'$. Then S is called a Σ'-protocol for $\mathfrak{R}, \mathfrak{R}'$ with challenge set C, public input y and private input w, if and only if it satisfies the following conditions:

- **Three-move form:** The protocol is of the following form: The prover P computes a commitment t and sends it to V. The verifier then draws a challenge $c \xleftarrow{\$} C$ and sends it to P. The prover sends a response s to the verifier. Depending on the protocol transcript (t, c, s), the verifier finally accepts or rejects the the proof. The protocol transcript (t, c, s) is called accepting if the verifier accepts.
- **Completeness:** Whenever $(x, w) \in \mathfrak{R}$, the verifier V accepts with probability at least $1 - \alpha$.
- **Soundness:** There exists a PPT algorithm $\mathcal{E}$ (the knowledge extractor) such that, for any $(x, w) \in \mathfrak{R}$ and deterministic prover $P^*(x)$ which succeeds in making V accept with probability $p = 1/|C| + \varepsilon$ over the choice of $c \xleftarrow{\$} C$, $\mathcal{E}$ can extract $(x, w') \in \mathfrak{R}'$ in expected time $poly(|x|)/\varepsilon$
- **Special honest-verifier zero-knowledge (HVZK):** There exists a PPT algorithm Sim (the simulator) taking $x \in L(\mathfrak{R})$ and $c \in C$ as inputs, that outputs $Sim(x, c) = (t, s)$ so that the triple (t, c, s) is indistinguishable from an accepting protocol transcript generated by a real protocol run.

We will use the non-interactive flavor of the ZKPoK by applying the Fiat-Shamir transform [20], i.e. the protocol can be made non-interactive in the random oracle model by replacing the challenge sampling of V by a hash of (x, t).

2.5 Rejection Sampling and the Normal Distribution

For a protocol to be zero-knowledge, the output of the prover needs to be independent of his secret. In certain situations achieving this independence requires rejection sampling to ensure a target distribution (e.g. [25]). We use discrete Gaussians when sampling errors in our proofs of knowledge as this allows for tighter parameters.

Definition 2.2 (Continuous Normal Distribution). The continuous Normal distribution over $\mathbb{R}^r$ centered at $\mathbf{v}$ with standard deviation σ is defined by the probability density function $\rho^r_{\mathbf{v},\sigma}(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^r e^{-\frac{\|\mathbf{x}-\mathbf{v}\|^2}{2\sigma^2}}$

Definition 2.3 (Discrete Normal Distribution). The discrete Normal distribution over $\mathbb{Z}^r$ centered at $\mathbf{v}$ with standard deviation σ is defined by the probability mass function $\mathcal{D}^r_{\mathbf{v},\sigma}(\mathbf{x}) = \rho^r_{\mathbf{v},\sigma}(\mathbf{x})/\rho^r_{\sigma}(\mathbb{Z}^r)$

For a polynomial $f \in \mathcal{R}$ we will denote by $f \longleftarrow \mathcal{D}^n_{\sigma}$ the fact that the coefficients of f come from the distribution $\mathcal{D}_{\sigma}$

Lemma 2.4 (Tail-Cut Bound [5]). $\Pr\left[\|\mathbf{z}\| \geq 2\sigma\sqrt{r}; \mathbf{z} \longleftarrow \mathcal{D}^r_{\sigma}\right] < 2^{-r}$

Theorem 2.5 (Rejection sampling [25] Theorem 4.6). *Let V be a subset of $\mathbb{Z}^r$ with elements of norm less than T, let h be a distribution over V, let $\sigma = 11T$. Consider the following algorithms:*

Rej :

(1) $\mathbf{v} \longleftarrow h$
(2) $\mathbf{z} \longleftarrow \mathcal{D}^r_{\mathbf{v},\sigma}$
(3) *Output* $(\mathbf{v}, \mathbf{z})$ *with probability* $\min\left(\mathcal{D}^r_{\sigma}(\mathbf{z})/(3\mathcal{D}^r_{\mathbf{v},\sigma}(\mathbf{z})), 1\right)$

F :

(1) $\mathbf{v} \longleftarrow h$
(2) $\mathbf{z} \longleftarrow \mathcal{D}^r_{\sigma}$
(3) *Output* $(\mathbf{v}, \mathbf{z})$ *with probability* $1/3$

The distributions output by both of these algorithms are statistically close, i.e. the output $\mathbf{z}$ *of* **Rej** *is a discrete Normal distribution centered on 0. Moreover the probability that* **Rej** *outputs something is exponentially close to* $1/3$

2.6 M-SIS and M-LWE

The M-SIS (Module Short Integer Solution) and M-LWE (Module Learning With Error) problems introduced in [24] are new variants of the Ring-SIS [26, 30] and Ring-LWE [27] problems often used in lattice-based constructions. Module problems are a generalization of their Ring counterparts in that instead of using operations over a large polynomial ring (e.g. of dimension 1024 or 2048) the module variants use operations on matrices of smaller polynomials (e.g. a matrix of size 4 or 8 containing polynomials of dimension 256), this allows for more modular implementations as the underlying ring does not need to be changed to accommodate new security parameters.

Definition 2.6 (M-SIS). We define M-SIS$_{q,n,d,m,\beta}$ as follows: Given $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{d\times m}$, find $\mathbf{z} \in \mathcal{R}^m$ such that $\mathbf{Az} = \mathbf{0} \mod q$ and $0 < \|\mathbf{z}\| \leq \beta$.

Remark that w.l.o.g the last block of size $d\times d$ of $\mathbf{A}$ can be taken to be the identity matrix (simply by left multiplying by the inverse of this block which has non-zero determinant with high probability). Let ψ be an error distribution over $\mathcal{R}$, let $\mathbf{s} \in \mathcal{R}_q^d$. We define $A_{\mathbf{s},\psi}$ to be the distribution over $\mathcal{R}_q^d \times \mathcal{R}$ obtained by choosing a vector $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q^d$, and $e \leftarrow \psi$, and returning $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$. We will use the normal form of the M-LWE problem in which the secret are sampled from the same distribution as the errors.

Definition 2.7 (Decision M-LWE). Let Ψ be a family of distributions over $\mathcal{R}$, we define (the decision variant of) M-LWE$_{q,n,d,\Psi}$ as follows: Let $\psi \in \Psi$, let $\mathbf{s} \leftarrow \psi^d$; the goal is to distinguish between arbitrarily many independent samples from $A_{\mathbf{s},\psi}$ and the same number of independent samples from $U(\mathcal{R}_q^d, \mathcal{R}_q)$.

2.7 Invertible polynomials

For extraction of witnesses in our zero knowledge protocols it will be useful that challenges (which will be polynomials of infinity norm equal to 1) and even differences of challenges are invertible in $\mathcal{R}_q$. For this we use the following lemma adapted from [28]:

Lemma 2.8 (Theorem 1.1 [28]). *If q is prime, $q = 17 \mod 32$, and $q > 2^{20}$ then any $f \in \mathcal{R}_q$ such that $0 < \|f\|_\infty \leq 2$ has an inverse in the ring.*

3 CRYPTOGRAPHIC PRIMITIVES

3.1 Commitment

In this section we describe the lattice-based homomorphic commitment we will use, the security of this commitment scheme relies on the hardness of the M-LWE and M-SIS problems. Let $d \in \mathbb{N}$, let $\sigma \in \mathbb{R}$, B_r be a positive real bound. We define a commitment with key space $\mathcal{R}_q^{(d+1)\times(2d+1)}$, message space $\mathcal{R}_q$, opening space $\left\{\mathbf{r} \in \mathcal{R}^{2d+1},\ \|\mathbf{r}\| \leq B_r\right\}$, and commitment space $\mathcal{R}_q^{d+1}$ (adapted from [7], which is a lattice-based instantiation of a generic commitment proposed in [17]):

Keygen(1^λ) :

- Let $\mathbf{A}' \xleftarrow{\$} \mathcal{R}_q^{d\times(d+1)}$
- Let $\mathbf{A} = \left[\begin{array}{c|c} \mathbf{A}' & \mathbf{I}_d \end{array}\right] \in \mathcal{R}_q^{d\times(2d+1)}$
- Let $\mathbf{B} \xleftarrow{\$} \mathcal{R}_q^{1\times(2d+1)}$
- Output $\mathbf{C} := \left[\begin{array}{c} \mathbf{A} \\ \mathbf{B} \end{array}\right] \in \mathcal{R}_q^{(d+1)\times(2d+1)}$

Commit$(m \in \mathcal{R}_q)$:

- Let $\mathbf{r} \leftarrow \mathcal{D}_\sigma^{n(2d+1)}$
- Output $\mathbf{Com}(m; \mathbf{r}) := \mathbf{Cr} + \begin{bmatrix} \mathbf{0} \\ m \end{bmatrix} \in \mathcal{R}_q^{d+1}$

Open$(\mathbf{c} \in \mathcal{R}_q^{d+1}, \mathbf{r} \in \mathcal{R}_q^{2d+1})$:

- If:
  - $\exists m' \in \mathcal{R}_q$ s.t $\mathbf{c} - \mathbf{Cr} = \begin{bmatrix} \mathbf{0} \\ m' \end{bmatrix}$
  - $\|\mathbf{r}\| \leq B_r$
- Then output m'
- Else output $\perp$

Theorem 3.1. *The commitment scheme described above is computationally hiding under the M-LWE assumption and computationally binding under the M-SIS assumption.*

Proof. The proof is close to the one of [7] but we use M-LWE as a computational assumption rather than statistical security as it allows for better parameters.

Binding Property: Suppose an adversary $\mathcal{A}$ generates a triple $(\mathbf{c}, \mathbf{r}, \mathbf{r}')$ such that $\mathbf{Open}(\mathbf{c}, \mathbf{r}) = m$ and $\mathbf{Open}(\mathbf{c}, \mathbf{r}') = m'$ where m, m' are valid messages and $m \neq m'$. Using the opening algorithm we have that $\mathbf{B}(\mathbf{r} - \mathbf{r}') = m - m' \neq 0$, thus $\mathbf{r} - \mathbf{r}' \neq 0$ and $\|\mathbf{r} - \mathbf{r}'\| \leq 2B_r$. Additionally $\mathbf{A}(\mathbf{r} - \mathbf{r}') = 0$, i.e. we have a solution of norm less than $2B_r$ for the M-SIS$_{q,n,d,2d+1,2B_r}$ challenge defined by $\mathbf{A}$.

Hiding Property: We show that an adversary that can distinguish a commitment from uniform can break the M-LWE problem (which clearly implies the computational hiding property given in Section 2.3). Let $\mathcal{A}$ generate a message m, suppose $\mathcal{A}$ can distinguish $\mathbf{Com}(m)$ from the uniform distribution over $\mathcal{R}_q^{d+1}$ with non negligible probability. Let $\mathbf{C} = \left[\begin{array}{c|c} \mathbf{X} & \mathbf{Y} \end{array}\right]$, with $\mathbf{X} \in \mathcal{R}_q^{(d+1)\times d}$ and $\mathbf{Y} \in \mathcal{R}_q^{(d+1)\times(d+1)}$, be the public commitment key. Let $\mathbf{U} = \mathbf{Y}^{-1}\mathbf{X}$, since $\mathbf{X}$ is sampled uniformly, so is $\mathbf{U}$. Consider the function that maps $\mathbf{y} \in \mathcal{R}_q^{d+1}$ to $\mathbf{Y}^{-1}\left(\mathbf{y} - \begin{bmatrix} \mathbf{0} \\ m \end{bmatrix}\right)$, this function maps the uniform distribution to itself and maps the distribution of $\mathbf{Com}(m)$ to the M-LWE$_{q,n,d,\mathcal{D}_\sigma^n}$ distribution for the matrix $\mathbf{U}$. Indeed if $\mathbf{y} = \mathbf{Cr} + \begin{bmatrix} \mathbf{0} \\ m \end{bmatrix}$ then $\mathbf{Y}^{-1}\left(\mathbf{y} - \begin{bmatrix} \mathbf{0} \\ m \end{bmatrix}\right) = \left[\begin{array}{c|c} \mathbf{U} & \mathbf{I} \end{array}\right]\mathbf{r}$. $\mathcal{A}$ is thus a distinguisher for M-LWE$_{q,n,d,\mathcal{D}_\sigma^n}$. □

By using Theorem 2.4 it is clear that our commitment scheme will be correct with overwhelming probability as long as $B_r \geq 2\sqrt{n(2d+1)}\sigma$, however we will need a larger gap between σ and B_r to be able to use the homomorphic properties of the commitment scheme.

3.2 OR Proof for Homomorphic Commitments

In this section we describe an or proof for our commitment scheme, i.e. a zero knowledge proof that a commitment $\mathbf{c}$ opens to a value $m \in \{0, 1\}$. As stated in Section 1.2, the proof is relaxed in the sense that it only proves knowledge of a randomness $\mathbf{r}$ such that

$f\mathbf{c}$ opens to a message fm for some small $f \in \mathcal{R}_q$. For a fixed matrix $\mathbf{C} \in \mathcal{R}_q^{(d+1)\times(2d+1)}$, we first define the two binary relations for which we want to prove the disjunction:

$$\mathfrak{R}_0 = \left\{(\mathbf{c}, \mathbf{r}) \in \mathcal{R}_q^{d+1} \times \mathcal{R}^{2d+1}, \mathbf{c} = \mathbf{C}\mathbf{r}, \|\mathbf{r}\| \leq B_{OR}\right\}$$

$$\mathfrak{R}_1 = \left\{(\mathbf{c}, \mathbf{r}) \in \mathcal{R}_q^{d+1} \times \mathcal{R}^{2d+1}, \mathbf{c} = \mathbf{C}\mathbf{r} + \begin{bmatrix}\mathbf{0}\\1\end{bmatrix}, \|\mathbf{r}\| \leq B_{OR}\right\}$$

We also define the two relaxed binary relations from which the soundness extractor will be able to recover a witness.

$$\mathfrak{R}'_0 = \Big\{(\mathbf{c}, \mathbf{r}, f) \in \mathcal{R}_q^{d+1} \times \mathcal{R}^{2d+1} \times \mathcal{R}, \\ f\mathbf{c} = \mathbf{C}\mathbf{r}, \|\mathbf{r}\| \leq B'_{OR}, 0 < \|f\| \leq 2\sqrt{60}\Big\}$$

$$\mathfrak{R}'_1 = \Big\{(\mathbf{c}, \mathbf{r}, f) \in \mathcal{R}_q^{d+1} \times \mathcal{R}^{2d+1} \times \mathcal{R}, \\ f\mathbf{c} = \mathbf{C}\mathbf{r} + f\begin{bmatrix}\mathbf{0}\\1\end{bmatrix}, \|\mathbf{r}\| \leq B'_{OR}, 0 < \|f\| \leq 2\sqrt{60}\Big\}$$

Let C be the set of all polynomials with coefficients in $\{-1, 0, 1\}$ with exactly 60 nonzero coefficients, i.e.

$$C = \left\{f \in \mathcal{R}, \|f\|_\infty = 1, \|f\|_1 = 60\right\}.$$

Our OR-Proof will use the challenge space $\Pi = \mathrm{Perm}(n) \times \{0, 1\}^{60}$ (where $\mathrm{Perm}(n)$ is the set of all permutations over the coefficients of vectors of dimension n). An element $\pi = (s, \mathbf{b}) \in \Pi$ acts on a polynomial in C by permuting its coefficients according to s and changing the sign if the i^{th} nonzero coefficient if $\mathbf{b}_i = 1$. Remark that for any $f, g \in C$, for $\pi \xleftarrow{\$} \Pi$ we have $\Pr[\pi(f) = g] = 1/|C|$. Let σ_{OR} be a positive real parameter, B_{OR} be a positive real bound, and H be a collision resistant hash function that maps arbitrary inputs to the uniform distribution over Π. We can now define our OR-proof for homomorphic commitments:

```
$\mathbf{\Pi_{OR}}(\mathbf{c} = \mathbf{C}\mathbf{r} + \begin{bmatrix}\mathbf{0}\\m\end{bmatrix}, \mathbf{r}, m \in \{0, 1\})$:
    (1) $\mathbf{r}_{1-m} \leftarrow \mathcal{D}^{n(2d+1)}_{\sigma_{OR}}$
    (2) $f_{1-m} \xleftarrow{\$} C$
    (3) $\mathbf{t}_{1-m} := \mathbf{C}\mathbf{r}_{1-m} + f_{1-m}\begin{bmatrix}\mathbf{0}\\1-m\end{bmatrix} - f_{1-m}\mathbf{c}$
    (4) $\rho \leftarrow \mathcal{D}^{n(2d+1)}_{\sigma_{OR}}$
    (5) $\mathbf{t}_m := \mathbf{C}\rho$
    (6) $\pi := H(\mathbf{c}, \mathbf{t}_0, \mathbf{t}_1) \xleftarrow{\$} \Pi$
    (7) $f_m = \pi^{2m-1}(f_{1-m})$
    (8) $\mathbf{r}_m = \rho + f_m\mathbf{r}$
    (9) Abort with probability $1 - \min\left(\frac{\mathcal{D}^{n(2d+1)}_{\sigma_{OR}}(\mathbf{r}_m)}{3\mathcal{D}^{n(2d+1)}_{f_m\mathbf{r},\sigma_{OR}}(\mathbf{r}_m)}, 1\right)$
    (10) Output $(\mathbf{r}_0, \mathbf{r}_1, f_0, f_1)$
$\mathbf{Verify_{OR}}$ $(\mathbf{c}, (\mathbf{r}_0, \mathbf{r}_1, f_0, f_1))$:
    (1) Let $\mathbf{t}_0 := \mathbf{C}\mathbf{r}_0 - f_0\mathbf{c}$
    (2) Let $\mathbf{t}_1 := \mathbf{C}\mathbf{r}_1 + f_1\begin{bmatrix}\mathbf{0}\\1\end{bmatrix} - f_1\mathbf{c}$
    (3) Let $\pi = H(\mathbf{c}, \mathbf{t}_0, \mathbf{t}_1)$
    (4) Check $\|\mathbf{r}_0\| \leq B'_{OR}$
    (5) Check $\|\mathbf{r}_1\| \leq B'_{OR}$
    (6) Check $f_0 \in C$
    (7) Check $f_1 = \pi(f_0)$
```

Remark that we could, as in [29], use a challenge $f \in C$ and compute $f_{1-m} = f - f_m$. But since C is not stable by difference this would leak information about m which would force us to use another rejection sampling and significantly increase the slack of the proof. Using Π as the challenge space allows us to cleverly circumvent this issue and obtain much better parameters.

Theorem 3.2. *If $\sigma_{OR} \geq 22 * \sqrt{60}B_{OR}$, $B'_{OR} \geq 2\sqrt{n(2d+1)}\sigma_{OR}$, then Π_{OR} is a zero knowledge proof of knowledge for the language $\mathfrak{R}_0 \vee \mathfrak{R}_1$, with soundness extractor in $\mathfrak{R}'_0 \vee \mathfrak{R}'_1$.*

Proof.
Correctness: Using Theorem 2.5 with $\sigma_{OR} \geq 22\sqrt{60}B_{OR} \geq \|f_m\mathbf{r}\|$ we have that the rejection step accepts with probability 1/3 and thus the proof outputs a result after an average of 3 runs. By construction of the proof the condition $f_1 = \pi(f_0)$ is verified on an honest proof. Moreover using Lemma 2.4 we have for $b \in \{0, 1\}$, $\|r_b\| \leq 2\sqrt{n(2d+1)}\sigma_{OR} \leq B'_{OR}$ with overwhelming probability.
Zero-knowledge: We construct a simulator Sim. For $(\mathbf{c}, \mathbf{r}) \in \mathfrak{R}_0 \vee \mathfrak{R}_1$ and $\pi \in \Pi$, Sim does the following:

```
(1) $f_0 \xleftarrow{\$} C$
(2) $f_1 = \pi(f_0)$
(3) For $b \in \{0, 1\}$, $\mathbf{r}_b \leftarrow \mathcal{D}^{n(2d+1)}_{\sigma_{OR}}$
(4) For $b \in \{0, 1\}$, $\mathbf{t}_b = \mathbf{C}\mathbf{r}_b + f_b\begin{bmatrix}\mathbf{0}\\b\end{bmatrix} - f_b\mathbf{c}$
(5) Abort with probability 2/3
(6) output $(\mathbf{r}_0, \mathbf{r}_1, f_0, f_1)$
```

Using Theorem 2.5 the distribution of the output of the simulator is identical to the one of an honest prover.
Soundness Let $(\mathbf{c}, \mathbf{r}) \in \mathfrak{R}_0 \vee \mathfrak{R}_1$, let $\mathcal{P}^*(\mathbf{c})$ be a deterministic prover, i.e. $\mathcal{P}^*(\mathbf{c})$ always queries H on the same input and his probability of success only depends on the output of H. Suppose that $\mathcal{P}^*(\mathbf{c})$ succeeds with probability $p = 1/|C| + \varepsilon$ over the randomness of the challenge π. We construct a soundness extractor $\mathcal{E}$ which extracts $\mathbf{r}'', f''$, in $poly(|(\mathbf{c}, \mathbf{r})|)/\varepsilon$ calls to H, such that $(\mathbf{c}, \mathbf{r}'', f'') \in \mathfrak{R}'_0 \vee \mathfrak{R}'_1$.
$\mathcal{E}$ first runs $\mathcal{P}^*(\mathbf{c})$ on fresh challenges $\pi \xleftarrow{\$} \Pi$ until $\mathcal{P}^*(\mathbf{c})$ outputs a valid proof $(\mathbf{c}, (\mathbf{r}_0, \mathbf{r}_1, f_0, f_1))$, this takes expected time $O\left(\frac{1}{1/|C|+\varepsilon}\right)$.
$\mathcal{E}$ then runs $\mathcal{P}^*(\mathbf{c})$ on random challenges until it outputs a valid proof $(\mathbf{c}, (\mathbf{r}'_0, \mathbf{r}'_1, f'_0, f'_1))$ such that either $f_0 \neq f'_0$ or $f_1 \neq f'_1$. Suppose $\mathcal{P}^*(\mathbf{c})$ has produced both of these proofs, let $b \in \{0, 1\}$ be such that $f_b \neq f'_b$, let $(\mathbf{c}, \mathbf{t}_0, \mathbf{t}_1)$ be the hash query made by $\mathcal{P}^*(\mathbf{c})$, since both proofs output by $\mathcal{P}^*(\mathbf{c})$ verify correctly, we have both $\mathbf{t}_b = \mathbf{C}\mathbf{r}_b + f_b\begin{bmatrix}\mathbf{0}\\b\end{bmatrix} - f_b\mathbf{c}$ and $\mathbf{t}_b = \mathbf{C}\mathbf{r}'_b + f'_b\begin{bmatrix}\mathbf{0}\\b\end{bmatrix} - f'_b\mathbf{c}$, which implies:

$$(f_b - f'_b)\mathbf{c} = \mathbf{C}(\mathbf{r}_b - \mathbf{r}'_b) + (f_b - f'_b)\begin{bmatrix}\mathbf{0}\\b\end{bmatrix}$$

Let $\mathbf{r}'' = \mathbf{r}_b - \mathbf{r}'_b$, $f'' = f_b - f'_b$, then $(\mathbf{c}, \mathbf{r}'', f'') \in \mathfrak{R}'_0 \vee \mathfrak{R}'_1$.
We still need to prove that $\mathcal{P}^*(\mathbf{c})$ outputs a proof such that $f_b \neq f'_b$

with probability at least ε.

$$\begin{aligned}
&\Pr\left[\mathcal{P}^*(\mathbf{c})\ \textit{succeeds} \wedge (f_0 \neq f_0' \vee f_1 \neq f_1')\right]\\
&= \Pr\left[\mathcal{P}^*(\mathbf{c})\ \textit{succeeds}\right] - \Pr\left[\mathcal{P}^*(\mathbf{c})\ \textit{succeeds} \wedge (f_0 = f_0' \wedge f_1 = f_1')\right]\\
&= 1/|C| + \varepsilon - \Pr\left[\mathcal{P}^*(\mathbf{c})\ \textit{succeeds} \wedge (f_0 = f_0' \wedge \pi(f_0) = \pi'(f_0)\right]\\
&\geq 1/|C| + \varepsilon - \Pr\left[\pi(f_0) = \pi'(f_0)\right]\\
&= \varepsilon
\end{aligned}$$

□

3.3 Amortized Proof

In this section we want to give a proof of knowledge of a small preimage for a one-way function. This proof can be used to prove that a commitment $\mathbf{c}$ opens to 0, by proving knowledge of $\mathbf{r}$ such that $\mathbf{Cr} = \mathbf{c}$, or it can be used to simply prove knowledge of the value that $\mathbf{c}$ commits to, by proving knowledge of $\mathbf{r}$ such that $\mathbf{Ar} = \mathbf{a}$ (with $\mathbf{a}$ the top part of $\mathbf{c}$). The proof given in this section avoids the caveat of the OR-proof from the previous section where the extracted randomness $\mathbf{r}$ was such that there is a small polynomial f such that $f\mathbf{c} = \mathbf{Cr} + \begin{bmatrix}\mathbf{0}\\ f\mathbf{x}\end{bmatrix}$, in this section we will always have $f = 2$. This is particularly useful because it entails that proving knowledge for two commitments implies proving knowledge for their sum (with of course a larger extracted randomness), which was not true for the previous proof where summing the extracted values would give an opening of $f_1\mathbf{c}_1 + f_2\mathbf{c}_2$ for some small polynomials f_1, f_2; it is not clear how one would extract an opening for $\mathbf{c}_1 + \mathbf{c}_2$ from this. As explained in Section 1.2, there are no efficient constructions to prove knowledge of a single secret, however in the context of our E-voting scheme each authority will prove knowledge of many secrets at once (one per voter). We can thus use the construction of [14, 18] which achieves quasi-optimal slack and negligible soundness error when amortizing over enough secrets. We first define the binary relation for which we will prove knowledge of a witness, let $t \in \mathbb{N}$, let $\mathbf{X} \in \mathcal{R}_q^{t\times(2d+1)}$ (in our scheme we will use either $\mathbf{X} = \mathbf{C} \in \mathcal{R}_q^{(d+1)\times(2d+1)}$ or $\mathbf{X} = \mathbf{A} \in \mathcal{R}_q^{d\times(2d+1)}$), let $s \in \mathbb{N}$ be the number of secrets we amortize over, let B_{Amo} and B'_{Amo} be two positive real bounds.

$$\mathfrak{R}_{Amo,\mathbf{X}} = \left\{(\mathbf{x}_i, \mathbf{r}_i)_{i\in[s]} \in \left(\mathcal{R}_q^t \times \mathcal{R}_q^{2d+1}\right)^s,\right.$$
$$\left.\forall i \in [s] : \mathbf{x}_i = \mathbf{X}\mathbf{r}_i \wedge \|\mathbf{r}_i\| \leq B_{Amo}\right\}$$

The binary relation for the soundness extractor will be:

$$\mathfrak{R}'_{Amo,\mathbf{X}} = \left\{(\mathbf{x}_i, \mathbf{r}_i)_{i\in[s]} \in \left(\mathcal{R}_q^t \times \mathcal{R}_q^{2d+1}\right)^s,\right.$$
$$\left.\forall i \in [s] : 2\mathbf{x}_i = \mathbf{X}\mathbf{r}_i \wedge \|\mathbf{r}_i\| \leq B'_{Amo}\right\}$$

We use the amortized zero-knowledge scheme from [14, 18], which we will denote by $\Pi_{Amo,X}$, as a black box algorithm (the description of the algorithm is rather lengthy so we refer to the cited paper for the full scheme, we discuss our choice of parameters and its validity in Section 5). The algorithm $\Pi_{Amo,\mathbf{X}}$ takes as input $(\mathbf{x}_i, \mathbf{r}_i)_{i\in[s]} \in \mathfrak{R}_{Amo,\mathbf{X}}$ and outputs a proof P, the verification algorithm $\mathbf{Verify}_{Amo,\mathbf{X}}$ takes as input $(\mathbf{x}_i)_{i\in[s]}$ and a proof P and accepts or rejects.

Theorem 3.3. *If $B'_{Amo} \geq 2684n\sqrt{2d+1}B_{Amo}$, and $s \geq 2209$, then $\Pi_{Amo,\mathbf{X}}$ is a zero knowledge proof of knowledge for the language $\mathfrak{R}_{Amo,\mathbf{X}}$, with soundness extractor in $\mathfrak{R}'_{Amo,\mathbf{X}}$.*

We now briefly discuss the parameter choices made for our amortized proof. We do not go into much detail as the description of the proof itself is rather cumbersome. An important point however is that the construction given in [18] depends on a parameter α which does not affect security nor the size of the proof (per voter) but dictates the efficiency of the proof as well as the number of voters needed to amortize. For any $\alpha \geq 2$ the number of voters needed for amortization[3] grows in $O\left(\frac{1}{\log^2\alpha}\right)$ and the time complexity of both the prover and the verifier grow in $O(\alpha)$. It is clear that increasing α past a certain point will yield little advantage at a high cost in computation time, however simply setting $\alpha = 2$ means we would need to amortized over more than 10600 voters to have 128 bits of post quantum security. We set $\alpha = 16$, which means that a little more than 2800 voters are enough for amortization, at the cost of our proofs taking 8 times longer to compute than for $\alpha = 2$. This time cost is not much of an issue since the amortized proofs are computed by the authorities after all votes have been cast and are not an urgent matter.

We should point out that after the submission of this paper, Baum and Lyubashevsky [8] constructed a different amortized proof system that only requires approximately 500 proofs for the amortization advantages to fully kick-in. We can therefore use this latter proof system in our voting protocol when there is a small number of voters.

4 OUR E-VOTING SCHEME

4.1 The Scheme

We instantiate our voting scheme according to the definition given in Section 2.2. We split the tallying algorithm in two parts, this algorithm in our E-Voting definition is interactive between authorities $\mathcal{A}_1, \ldots, \mathcal{A}_{N_A}$. In our instantiation there is no need for interaction, each authority $\mathcal{A}_j$ can run an algorithm $\mathbf{Tally}_j(par, pk_1, \ldots, pk_{N_V}, BB, sk_j)$ and publish on the bulletin board its partial tally $t^{(j)}$ and proof $\pi^{A,(j)}$, anyone can then run $\mathbf{Tally}(par, pk_1, \ldots, pk_{N_V}, BB)$ to compute the total tally and final proof.

Setup(λ) :

- generate parameters n, q, d, σ.
- $\mathbf{C} := \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \leftarrow \mathbf{Keygen}(1^\lambda)$, with $\mathbf{C} \in \mathcal{R}_q^{(d+1)\times(2d+1)}$.
- output $par := n, q, d, \sigma, \mathbf{C}$

Let $\left(\mathbf{KGen}(1^\lambda), \mathbf{Enc}, \mathbf{Dec}\right)$ be a CCA-Secure public key encryption scheme, which the authorities will use to obtain the shares of the randomness of each voter.

$\mathbf{ASetup_j}(par)$:

- $(pk_j, sk_j) \leftarrow \mathbf{KGen}(1^\lambda)$

[3] The exact number of voters needed for amortization with k bits of post-quantum security is p^2 where p is the first prime larger than $4k\,\frac{1+1/\log\alpha}{9+\log\alpha} + 1$

- Give sk_j to $\mathcal{A}_j$
- output pk_j

To cast a bulletin, voter i will share his vote into N_A additive shares and compute a commitment $\mathbf{c}_i^{(j)}$ for each of them. He proves that the sum of these commitments is a commitment to either 0 or 1 and then encrypts the randomness $\mathbf{r}_i^{(j)}$ under the public key pk_j so that each authority can open one share of the vote. He finally posts his vote, proof, commitments, and encryptions on the bulletin board along with a signature.

$\mathbf{Vote_i}(par, pk_1, \ldots, pk_{N_A}, id_i, v_i)$:

- $v_i^{(j)} \xleftarrow{\$} \mathbb{Z}_q$ s.t. $v_i = \sum_1^{N_A} v_i^{(j)}$
- $\mathbf{r}_i^{(j)} \leftarrow \mathcal{D}_\sigma^{n(2d+1)}$
- $\mathbf{c}_i^{(j)} := \mathbf{Com}\left(v_i^{(j)}; \mathbf{r}_i^{(j)}\right)$
- $\mathbf{r}_i := \sum_1^{N_A} \mathbf{r}_i^{(j)}$
- $\mathbf{c}_i := \sum_1^{N_A} \mathbf{c}_i^{(j)}$
- $\pi_i^V = \Pi_{OR}(\mathbf{c}_i, \mathbf{r}_i)$
- $\mathbf{e}_i^{(j)} = \mathbf{Enc}(\mathbf{r}_i^{(j)}, pk_j)$
- $b_i = \left(id_i, \pi_i^V, (\mathbf{c}_i^{(j)}, \mathbf{e}_i^{(j)})_{j \in [N_A]}\right)$
- Sign and publish b_i on the bulletin board

Before tallying the votes each authority $\mathcal{A}_j$ will check whether the bulletins have been properly cast. i.e. for each bulletin b_i, $\mathcal{A}_j$ checks the signature on b_i, the proof that $v_i \in \{0, 1\}$ and that the encryption of $\mathbf{r}_i^{(j)}$ under his public key decrypts to a valid randomness.

$\mathbf{TestB_{i,j}}(par, pk_1, \ldots, pk_{N_A}, sk_j, b_i)$:

- $\left(id_i, \pi_i^V, \mathbf{c}_i^{(1)}, \mathbf{e}_i^{(1)}, \ldots, \mathbf{c}_i^{(N_A)}, \mathbf{e}_i^{(N_A)}\right) := b_i$
- Check that b_i was signed by voter id_i
- $\mathbf{Verify}(\pi_i^V)$
- $\mathbf{r}_i^{(j)} := \mathbf{Dec}(\mathbf{e}_i^{(j)}, sk_j)$
- Check $\left\|\mathbf{r}_i^{(j)}\right\| \leq 2\sqrt{n(2d+1)}\sigma$

Each authority $\mathcal{A}_j$ will compute its share $t^{(j)}$ of the total tally as well as a proof that $t^{(j)}$ has been computed correctly. To do so $\mathcal{A}_j$ first decrypts $\mathbf{e}_i^{(j)}$ for each bulletin b_i to recover randomness $\mathbf{r}_i^{(j)}$. He then proves that for each voter i, $\mathbf{r}_i^{(j)}$ is a valid opening of $\mathbf{c}_i^{(j)}$ and finally outputs $\mathbf{r}^{(j)}$, the sum over i of all $\mathbf{r}_i^{(j)}$ (the share $t^{(j)}$ of the final tally can be obtained by opening $\mathbf{c}^{(j)}$ the sum of the $\mathbf{c}_i^{(j)}$ using the $\mathbf{r}^{(j)}$ output by $\mathcal{A}_j$) as well as the proofs he computed. Note that w.l.o.g we will consider in all the following algorithms that all the bulletins on the bulletin board were tested and accepted by all the authorities (otherwise we can just discard the rejected bulletins and adjust N_V to the number of remaining bulletins).

$\mathbf{Tally}_j(par, pk_1, \ldots, pk_{N_A}, BB, sk_j)$

- $\left(id_i, \pi_i^V, \mathbf{c}_i^{(1)}, \mathbf{e}_i^{(1)}, \ldots, \mathbf{c}_i^{(N_A)}, \mathbf{e}_i^{(N_A)}\right) := b_i$, For $(b_i)_{i \in N_V} \in BB$
- $\forall i,\ \mathbf{r}_i^{(j)} := \mathbf{Dec}(\mathbf{e}_i^{(j)}, sk_j)$
- $$\begin{aligned}\pi^{A,(j)} &= \left(\pi_1^A, \ldots, \pi_{N_V}^A\right) \\ &= \Pi_{Amo,\mathbf{A}}\left(\mathbf{a}_1^{(j)}, \ldots, \mathbf{a}_{N_V}^{(j)}, \mathbf{r}_1^{(j)}, \ldots, \mathbf{r}_{N_V}^{(j)}\right)\end{aligned}$$
- $\mathbf{r}^{(j)} = \sum_1^{N_V} \mathbf{r}_i^{(j)}$
- Sign and publish $\pi^{A,(j)}, \mathbf{r}^{(j)}$ on the bulletin board

To compute the total tally, anyone can simply recover the randomnesses $\mathbf{r}^{(j)}$ published by each authority $\mathcal{A}_j$, compute the corresponding commitment $\mathbf{c}^{(j)} = \sum \mathbf{c}_i^{(j)}$ and open it to the partial tally $t^{(j)}$. The total tally is then the sum of the partial tallies.

$\mathbf{Tally}(par, pk_1, \ldots, pk_{N_A}, BB)$

- For each authority $\mathcal{A}_j$, $j \in [N_A]$ recover $\mathbf{r}^{(j)}$ on BB
- $\forall j,\ \mathbf{c}^{(j)} := \sum_{i=1}^{N_V} \mathbf{c}_i^{(j)}$
- $\forall j,\ \begin{bmatrix}\mathbf{0} \\ t^{(j)}\end{bmatrix} := \mathbf{c}^{(j)} - \mathbf{C}\mathbf{r}^{(j)}$
- $t := \sum_1^{N_A} t^{(j)}$
- publish t

The verification algorithm can be run by anyone to check that the final tally is correct (i.e. the voting scheme is publicly verifiable). To do so one simply verifies all the proofs output by the voters and authorities and checks that the opening of the total tally has been done correctly (by computing it again).

$\mathbf{Verify}(par, pk_1, \ldots, pk_{N_A}, BB, t)$

- For each $i \in [N_V]$, $j \in [N_A]$, recover $\mathbf{c}_i^{(j)}, \pi_i^V, \pi_{i,j}^A$ on BB.
- $\forall i$, verify π_i^V
- $\forall i, j$, verify $\pi_{i,j}^A$
- $\forall j,\ \mathbf{c}^{(j)} := \sum_{i=1}^{N_V} \mathbf{c}_i^{(j)}$
- $\forall j,\ \begin{bmatrix}\mathbf{0} \\ t^{(j)}\end{bmatrix} := \mathbf{c}^{(j)} - \mathbf{C}\mathbf{r}^{(j)}$
- Check that $t = \sum_1^{N_A} t^{(j)}$

For correctness we need all the proofs to verify correctly, which will be true with overwhelming probability for appropriate parameters (cf. Section 5), we also need the test on the norm of $\mathbf{r}_i^{(j)}$ to succeed, which will be true with overwhelming probability by using Lemma 2.4, and we need for the commitment of the partial tallies to open correctly, i.e. we need $\left\|\mathbf{r}^{(j)}\right\| \leq B_r$. We can fix the parameters so that this condition is verified, however the norm of $\mathbf{r}^{(j)}$ grows linearly with the number of voters which, as we discuss in the next section, heavily impacts the efficiency of this scheme.

Dealing with Misbehaving Authorities. A malicious authority could prevent a voter from casting his vote by claiming that the voter's ballot is invalid. Since the **TestB** algorithm requires the secret key of the authority, the authority's claim cannot be publicly verified. This situation can be improved by letting voters store the randomness used in the encryption of $\mathbf{e}_i^{(j)}$ and, in case their ballot is incorrectly claimed to be invalid, reveal $\mathbf{r}_i^{(j)}$ and the randomness to show that the authority is at fault.

4.2 Improved Voting Scheme

A major caveat in the scheme presented in Section 4.1 is that parameters grow linearly in the number of voters. Indeed for correctness a verifier needs to be able to open the sum over all voters of the commitments of the vote shares, this implies that the bound B_r on the size of correct openings grows linearly in the number of

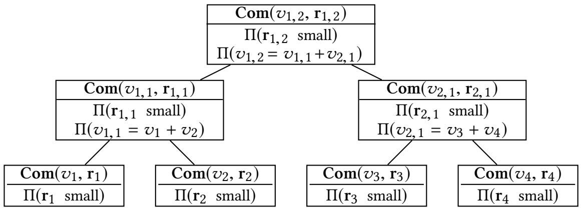
Com($v_{1,2}$, $\mathbf{r}_{1,2}$)
$\Pi(\mathbf{r}_{1,2}$ small)
$\Pi(v_{1,2} = v_{1,1} + v_{2,1})$
Com($v_{1,1}$, $\mathbf{r}_{1,1}$)
$\Pi(\mathbf{r}_{1,1}$ small)
$\Pi(v_{1,1} = v_1 + v_2)$
Com($v_{2,1}$, $\mathbf{r}_{2,1}$)
$\Pi(\mathbf{r}_{2,1}$ small)
$\Pi(v_{2,1} = v_3 + v_4)$
Com(v_1, $\mathbf{r}_1$)
$\Pi(\mathbf{r}_1$ small)
Com(v_2, $\mathbf{r}_2$)
$\Pi(\mathbf{r}_2$ small)
Com(v_3, $\mathbf{r}_3$)
$\Pi(\mathbf{r}_3$ small)
Com(v_4, $\mathbf{r}_4$)
$\Pi(\mathbf{r}_4$ small)

Figure 1: Example of the improved tallying for an authority. At each level s fresh randomnesses $\mathbf{r}_{u,s}$ are sampled and the authority commits to the sum of the votes of the previous level. The authority computes a proof that each new randomness is small and that it commits to the right value. Finally the authority publishes all the commitments and proofs as well as the opening, here $\mathbf{r}_{1,2}$, of the top level commitment which opens to the sum of the votes (i.e. $v_{1,2} = v_1 + v_2 + v_3 + v_4$).

voters. Increasing this bound heavily impacts the parameters of the scheme, e.g. if we fix $n = 256$ and $q \simeq 2^{31}$, then for $\sim$ 100 bits of security we require a dimension $d = 7$ for 100 voters and $d = 12$ for 100 000 voters. This nearly doubles the commitment size, proof size and communication cost per voter (another issue with the previous scheme is that privacy is nontrivial, indeed revealing the sum of the randomnesses used makes it so that the privacy cannot be easily proven). We avoid this problem by using the fact that the authorities know the shares of many commitments and can thus create new commitments for the sum of their associated messages (this is the solution discussed in Section 1.3.1). e.g. Imagine authority $\mathcal{A}_j$ has received the commitments and openings $\mathbf{c}_i^{(j)} = \mathbf{Com}(v_i^{(j)}; \mathbf{r}_i^{(j)},)$ from voters 1 to N_V, $\mathcal{A}_j$ can choose $l << N_V$ and compute new commitments $\mathbf{c}_{1,1}^{(j)}, \dots, \mathbf{c}_{N_V/l,1}^{(j)}$, where $\mathbf{c}_{i,1}^{(j)} = \mathbf{Com}(\sum_{l(i-1)+1}^{li} v_i^{(j)}; \mathbf{r}_{i,1}^{(j)})$ with fresh randomnesses $\mathbf{r}_{i,1}^{(j)}$ and publish these commitments on the bulletin board. Notice that $\mathbf{c}^{(j)} = \sum_1^{N_V} \mathbf{c}_i^{(j)}$ opens to the same message as $\mathbf{c}^{(j)\prime} = \sum_1^{N_V/l} \mathbf{c}_{i,1}^{(j)}$ (assuming both sums are valid commitments). However the randomness in the commitment $\mathbf{c}^{(j)\prime}$ will be approximately l times smaller than the one in $\mathbf{c}^{(j)}$ which means that $\mathbf{c}^{(j)\prime}$ can be a valid commitment even if $\mathbf{c}^{(j)}$ is not. By proving in zero knowledge that for $i \leq N_V/l$ the commitment $\mathbf{c}_{i,1}^{(j)}$ is valid and opens to the same value as $\sum_{l(i-1)+1}^{li} \mathbf{c}_i^{(j)}$ we can ensure that the scheme remains secure even if parameters are only set so that $\mathbf{c}^{(j)\prime}$ is valid and not $\mathbf{c}^{(j)}$. This effectively allows us to reduce B_r by a factor l. This process can be iterated by summing the $\mathbf{c}_{i,1}^{(j)}$ by buckets of l and outputting new commitments $\mathbf{c}_{i,2}^{(j)}$ to the sum of the corresponding messages with fresh randomnesses, once again accompanied by a proof that each commitment is valid and opens to the same value as a sum of $\mathbf{c}_{i,1}^{(j)}$ (an example of such summations with buckets of size $l = 2$ is given in Figure 1).

Each authority can repeat this process until there are less than l commitments to be summed, resulting in a bound B_r that grows linearly in l but remains independent of the number of voters. On the other hand this new protocol will output an overhead of $\sim N_V/(l-1)$ commitments (more precisely $N_V/l + N_V/l^2 + \dots$ extra commitments) and 2 additional proofs (which can be amortized over) for each new commitment.

Tally$_j(par, pk_1, \dots, pk_{N_A}, BB, sk_j)$

```
• $\left(id_i, \pi_i^V, \mathbf{c}_1^{(j)}, \mathbf{e}_1^{(j)}, \dots, \mathbf{c}_{N_V}^{(j)}, \mathbf{e}_{N_V}^{(j)}\right) := b_i$, For $(b_i)_{i \in N_V} \in BB$
• $\forall i,\ r_{i,0}^{(j)} := \mathbf{Dec}(\mathbf{e}_i^{(j)}, sk_j)$
• $\forall i,\ \begin{bmatrix} \mathbf{0} \\ v_{i,0}^{(j)} \end{bmatrix} := \mathbf{c}_i^{(j)} - \mathbf{C}\mathbf{r}_i^{(j)}$
• For $s \in \left(1, \dots, \lceil \log_l(N_V) \rceil\right)$:
    – For $u \in (1, \dots, \lceil N_V/l^s \rceil)$:
        ∗ $\mathbf{r}_{u,s}^{(j)} \leftarrow \mathcal{D}_\sigma^{n(2d+1)}$
        ∗ $\mathbf{r}_{\mathbf{u},\mathbf{s}}^{(\mathbf{j})\,\prime} := \mathbf{r}_{u,s}^{(j)} - \sum_{x=(u-1)l+1}^{ul} \mathbf{r}_{x,s-1}^{(j)}$
        ∗ $v_{u,s}^{(j)} := \sum_{x=(u-1)l+1}^{ul} v_{x,s-1}^{(j)}$
        ∗ $\mathbf{c}_{u,s}^{(j)} := \mathbf{Com}\left(v_{u,s}^{(j)}; \mathbf{r}_{u,s}^{(j)}\right)$
        ∗ $\mathbf{c}_{\mathbf{u},\mathbf{s}}^{(\mathbf{j})\,\prime} := \mathbf{c}_{u,s}^{(j)} - \sum_{x=(u-1)l+1}^{ul} \mathbf{c}_{x,s-1}^{(j)}$
        ∗ $\begin{bmatrix} \mathbf{a}_{u,s}^{(j)} \\ \mathbf{b}_{u,s}^{(j)} \end{bmatrix} := \mathbf{c}_{u,s}^{(j)}$
• $\pi^{A,(j)} = \Pi_{Amo,\mathbf{A}}\left(\mathbf{a}_{u,s}^{(j)}, \mathbf{r}_{u,s}^{(j)}\right)_{\begin{cases} s \in \left(0, \dots, \lceil \log_l(N_V) \rceil\right) \\ u \in (1, \dots, \lceil N_V/l^s \rceil) \end{cases}}$
• $\pi^{A,(j)\prime} = \Pi_{Amo,\mathbf{A}}\left(\mathbf{c}_{u,s}^{(j)\,\prime}, \mathbf{r}_{u,s}^{(j)\,\prime}\right)_{\begin{cases} s \in \left(1, \dots, \lceil \log_l(N_V) \rceil\right) \\ u \in (1, \dots, \lceil N_V/l^s \rceil) \end{cases}}$
• $\mathbf{c}_{Tot}^{(j)} = \left(\mathbf{c}_{u,s}^{(j)}\right)_{\begin{cases} s \in \left(1, \dots, \lceil \log_l(N_V) \rceil\right) \\ u \in (1, \dots, \lceil N_V/l^s \rceil) \end{cases}}$
• Sign and publish $\pi^{A,(j)}, \pi^{A,(j)\prime}, \mathbf{c}_{Tot}^{(j)}, \mathbf{r}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ on the bulletin board
```

To compute the total tally one only needs to open the commitments $\mathbf{c}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ for each $j \in N_A$ as these are commitments to the partial tallies of each authority.

Tally$(par, pk_1, \dots, pk_{N_A}, BB)$

```
• For $j \in [N_A]$ recover $\mathbf{c}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ and $\mathbf{r}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ on $BB$
• $\forall j,\ \begin{bmatrix} \mathbf{0} \\ t^{(j)} \end{bmatrix} := \mathbf{c}_{1,\lceil \log_l(N_V) \rceil}^{(j)} - \mathbf{C}\mathbf{r}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$
• $t := \sum_1^{N_A} t^{(j)}$
• publish $t$
```

To verify the election one needs to verify the OR-Proof of each user, the proof of correct opening of each $\mathbf{c}_{u,s}^{(j)}$, and the proof that $\mathbf{c}_{u,s}^{(j)\,\prime}$ opens to zero. The verifier can then recompute the tally and check that it has been done correctly.

Verify$(par, pk_1, \dots, pk_{N_A}, BB, t)$

```
• For each $i \in [N_V]$, recover $\pi_i^V$ on $BB$.
• $\forall i$, verify $\pi_i^V$
• $\forall j \in [N_A],\ \forall (s,u) \in \left(1, \dots, \lceil \log_l(N_V) \rceil\right) \times (1, \dots, \lceil N_V/l^s \rceil)$
  recover $\pi_{u,s}^{A,(j)}$ and $\pi_{u,s}^{A,(j)\prime}$ from $BB$
```

Parameter	Notation	Value
Ring dimension	n	256
Modulus	q	$2^31 - 2^7 - 2^5 + 1$
Module size	d	7
Commitment std deviation	σ	1
Number of voters	N_V	arbitrary
Number of authorities	N_A	4
"Bucket" size	l	30

Table 2: A possible set of parameters for our E-Voting scheme. These parameters achieve a post-quantum security of 119 bits in time and 93 bits in space for privacy, as well as 180 bits in time and 141 bits in space for consistency.

	Voter	Authority per voter			Verification per Voter
	total	total	sampling	OWF	total
Time	8.5ms	0.15s	33.2%	62.9%	0.15s

Table 3: Timings for our implementation of the voting scheme simulating an election with 11000 voters.

- $\forall j, u, s$ verify $\pi_{u,s}^{A,(j)}$ and $\pi_{u,s}^{A,(j)\prime}$
- For $j \in [N_A]$ recover $\mathbf{c}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ and $\mathbf{r}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ on BB
- $\forall j$, $\begin{bmatrix} \mathbf{0} \\ t^{(j)} \end{bmatrix} := \mathbf{c}_{1,\lceil \log_l(N_V) \rceil}^{(j)} - \mathbf{C}\mathbf{r}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$
- Check that $t = \sum_1^{N_A} t^{(j)}$

We prove privacy and consistency in Appendix A and we discuss how to set the parameters in Section 5.

5 PARAMETERS

In this section we review the bounds imposed on the parameters of our scheme by the correctness and security of the vote, and we propose concrete parameters in Table 2 as well as benchmarks from our implementation of the scheme.

The correctness and security of our scheme impose the following bounds on the parameters:

- Correctness of π^V: $B_{OR} \geq 2N_A\sqrt{n(2d+1)}\sigma$
- Zero-knowledge of π^V: $B'_{OR} \geq 44\sqrt{60n(2d+1)}B_{OR}$
- Correctness of π^A: $B_{Amo,1} \geq 2\sqrt{n(2d+1)}\sigma$
- Zero-knowledge of π^A: $B'_{Amo,1} \geq 2684n\sqrt{2d+1}B_{Amo,1}$
- Correctness of $\pi^{A'}$: $B_{Amo,2} \geq 2(l+1)\sqrt{n(2d+1)}\sigma$
- Zero-knowledge of $\pi^{A'}$: $B'_{Amo,2} \geq 2684n\sqrt{2d+1}B_{Amo,2}$
- Consistency of the vote (equation (3)): $2B'_{OR} \leq B_r$
- Consistency of the vote (equation (4)): $2\sqrt{60}N_AB'_{Amo,1} \leq B_r$
- Consistency of the vote (equation (7)): $(l+1)B'_{Amo,1} \leq B_r$
- Consistency of the vote (equation (8)): $B'_{Amo,2} \leq B_r$

Using the security analysis of [4] and [3] to assess the hardness of M-LWE and M-SIS, we set our parameters as in Table 2 (we arbitrarily fix the number of authorities to 4, anything larger than 2 is enough for security and this does not impact performance significantly). Due to the improved E-voting scheme of Section 4.2, the number of voters does not affect the security at all and can thus be taken arbitrarily large. Using these parameters and the cryptanalysis of [3, 4] we obtain a post-quantum security of 119 bits in time and 93 bits in space for the privacy of our scheme, as well as a post-quantum security of 180 bits in time and 141 bits in space for consistency. We have implemented the complete voting scheme in C. The main computational problems in the scheme are the sampling of discrete Gaussian vectors and multiplication of polynomials in $\mathbb{Z}_q[X]/(X^n + 1)$. For the sampling we have implemented a two-stage Knuth-Yao sampler. We have taken great care to ensure that the statistical distance between the sampled vectors and the exact discrete distribution is below 2^{-100}. This required computing the probabilities and the lookup table for the sampler with a multiprecision library. We used pari [33] for this task. For the fastest possible multiplication in rings of the given form, one usually chooses the prime q in such a way that $\mathbb{Z}_q$ contains a $2n$-th root of unity. This then implies that the modulus $X^n + 1$ splits into linear factors over $\mathbb{Z}_q$ and allows for using an NTT-based multiplication algorithm. Unfortunately, the security requirements of our scheme prevent q from being chosen in this way. Instead of completely resorting to a general algorithm that works for multiplication modulo arbitrary polynomials, we have exploited the fact that for our prime q, $X^n + 1$ does in fact split into 16 factors. This allowed us to use a general multiplication algorithm only after 4 stages of NTT. We have used our own NTT implementation and the highly optimized FLINT library [23] for the base case multiplication. FLINT uses a variant of Kronecker substitution for this task.

Table 3 gives the results of an experiment conducted with 11000 Voters. In that instance each authority $\mathcal{A}_j$ will amortize proofs over 4000 commitments (we use 4000 commitments rather than the minimum of 2809 so that the data structures used in the amortized proofs are simpler to implement) and thus compute 3 amortized proof to obtain $\pi^{A,(j)}$ as well as 1 amortized proof to compute $\pi^{\mathcal{A},(j)\prime}$. The total time per server is of 41min, we give times per voter in Table 3. We have used a laptop equipped with an Intel Skylake i7 CPU running at 2.6 GHz to perform all tests.

ACKNOWLEDGEMENTS

We thank the anonymous referees for numerous comments that helped
SNSF ERC Transfer Grant CRETP2-166734 – FELICITY.

REFERENCES

[1] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. 2012. Tightly-Secure Signatures from Lossy Identification Schemes. In *EUROCRYPT*. 572–590.

[2] Ben Adida. 2008. Helios: Web-based Open-Audit Voting. In *Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA*. 335–348.

[3] Martin R. Albrecht, Rachel Player, and Sam Scott. 2015. On the concrete hardness of Learning with Errors. *J. Mathematical Cryptology* 9, 3 (2015), 169–203. http://www.degruyter.com/view/j/jmc.2015.9.issue-3/jmc-2015-0016/jmc-2015-0016.xml

[4] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. 2016. Post-quantum Key Exchange - A New Hope. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*. 327–343. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim

[5] Wojciech Banaszczyk. 1993. New bounds in some transference theorems in the geometry of numbers. *Math. Ann.* 296 (1993), 625–635.

[6] Carsten Baum, Ivan Damgård, Kasper Green Larsen, and Michael Nielsen. 2016. How to Prove Knowledge of Small Secrets. In *CRYPTO*. 478–498.

[7] Carsten Baum, Ivan Damgård, Sabine Oechsner, and Chris Peikert. 2016. Efficient Commitments and Zero-Knowledge Protocols from Ring-SIS with Applications to Lattice-based Threshold Cryptosystems. *IACR Cryptology ePrint Archive* 2016 (2016), 997. http://eprint.iacr.org/2016/997

[8] Carsten Baum and Vadim Lyubashevsky. 2017. Simple Amortized Proofs of Shortness for Linear Relations over Polynomial Rings. *IACR Cryptology ePrint Archive* 2017 (2017), 759. http://eprint.iacr.org/2017/759

[9] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. 2014. Better Zero-Knowledge Proofs for Lattice Encryption and Their Application to Group Signatures. In *ASIACRYPT*. 551–572.

[10] David Bernhard, Véronique Cortier, David Galindo, Olivier Pereira, and Bogdan Warinschi. 2015. SoK: A Comprehensive Analysis of Game-Based Ballot Privacy Definitions. In *IEEE Symposium on Security and Privacy*. 499–516.

[11] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. 2016. A Homomorphic LWE Based E-voting Scheme. In *Post-Quantum Cryptography – PQCrypto 2016*. 245–265.

[12] Véronique Cortier, David Galindo, Ralf Küsters, Johannes Mueller, and Tomasz Truderung. 2016. SoK: Verifiability Notions for E-Voting Protocols. In *IEEE Symposium on Security and Privacy, SP 2016*. IEEE Computer Society, 779–798.

[13] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. 1994. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *CRYPTO '94*. 174–187.

[14] Ronald Cramer, Ivan Damgård, Chaoping Xing, and Chen Yuan. 2017. Amortized Complexity of Zero-Knowledge Proofs Revisited: Achieving Linear Soundness Slack. In *EUROCRYPT*. Also available at `http://eprint.iacr.org/2016/681`.

[15] Ronald Cramer, Matthew K. Franklin, Berry Schoenmakers, and Moti Yung. 1996. Multi-Autority Secret-Ballot Elections with Linear Work. In *EUROCRYPT '96*. 72–83.

[16] Ivan Damgård, Serge Fehr, and Louis Salvail. 2004. Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks. In *CRYPTO*. 254–272.

[17] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. 1993. On the Existence of Statistically Hiding Bit Commitment Schemes and Fail-Stop Signatures. In *CRYPTO*. 250–265.

[18] Rafaël Del Pino and Vadim Lyubashevsky. 2017. Amortization with Fewer Equations for Proving Knowledge of Small Secrets. *IACR Cryptology ePrint Archive* 2017 (2017), 280. To appear in CRYPTO 2017.

[19] Ariel J. Feldman, J. Alex Halderman, and Edward W. Felten. 2007. Security Analysis of the Diebold AccuVote-TS Voting Machine. In *2007 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'07*.

[20] Amos Fiat and Adi Shamir. 1986. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO*. 186–194.

[21] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *STOC*. 169–178.

[22] Rop Gonggrijp and Willem-Jan Hengeveld. Studying the Nedap/Groenendaal ES3B Voting Computer: A Computer Security Perspective. In *2007 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'07*.

[23] W. Hart, F. Johansson, and S. Pancratz. 2013. FLINT: Fast Library for Number Theory. (2013). Version 2.4.0, http://flintlib.org.

[24] Adeline Langlois and Damien Stehlé. 2015. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography* 75, 3 (2015), 565–599.

[25] Vadim Lyubashevsky. 2012. Lattice Signatures Without Trapdoors. In *EUROCRYPT*. 738–755.

[26] Vadim Lyubashevsky and Daniele Micciancio. 2006. Generalized Compact Knapsacks Are Collision Resistant. In *ICALP (2)*. 144–155.

[27] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. 2010. On Ideal Lattices and Learning with Errors over Rings. In *EUROCRYPT*. 1–23.

[28] Vadim Lyubashevsky and Gregor Seiler. 2017. Partially Splitting Rings for Faster Lattice-Based Zero-Knowledge Proofs. (2017).

[29] Carlos Aguilar Melchor, Slim Bettaieb, Xavier Boyen, Laurent Fousse, and Philippe Gaborit. 2013. Adapting Lyubashevsky's Signature Schemes to the Ring Signature Setting. In *AFRICACRYPT*. 1–25.

[30] Chris Peikert and Alon Rosen. 2006. Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In *TCC*. 145–166.

[31] Scytl R&D. 2017. Swiss Online Voting Protocol. https://www.post.ch/-/media/post/evoting/dokumente/swiss-post-online-voting-protocol.pdf. (2017).

[32] Jacques Stern. 1993. A New Identification Scheme Based on Syndrome Decoding. In *CRYPTO*. 13–21.

[33] The PARI Group 2016. *PARI/GP version* `2.9.0`. The PARI Group, Univ. Bordeaux. available from http://pari.math.u-bordeaux.fr/.

[34] Dominique Unruh. 2017. Post-Quantum Security of Fiat-Shamir. *IACR Cryptology ePrint Archive* 2017 (2017), 398. http://eprint.iacr.org/2017/398

A SECURITY ANALYSIS OF THE VOTING SCHEME

In this section we prove the privacy and consistency of our E-voting scheme as defined in Section 4. For privacy we consider the following advantages for an adversary $\mathcal{A}$:

- $\mathbf{Adv}^{CCA}_{\mathcal{A}}(\lambda)$ the advantage of $\mathcal{A}$ in the CCA security game of the encryption scheme.
- $\mathbf{Adv}^{Hid}_{\mathcal{A}}(\lambda)$ the advantage of $\mathcal{A}$ over the Hiding property of the commitment scheme.

Since the zero-knowledge of both the OR-Proof and the amortized proof are statistical, the probability of distinguishing between the simulator and the actual proof is less than $2^{-\lambda}$.

Theorem A.1. *The advantage of any PPT adversary $\mathcal{A}$ over the privacy of our E-voting scheme is at most:*

$$\mathbf{Adv}^{priv}_{\mathcal{A}}(\lambda) \leq N_V \left(2\mathbf{Adv}^{CCA}_{\mathcal{A}}(\lambda) + \frac{l}{l-1}\mathbf{Adv}^{Hid}_{\mathcal{A}}(\lambda) + 2^{-\lambda+1}\right) + 2^{-\lambda+2}$$

Proof. We use a game based proof:

Game $\mathbf{G}_0$: In this game we run $Exp^{priv,0}_{\mathcal{A}}$ as defined in Section 2.2. The voting, casting and tallying oracle are run honestly by the simulator using choice bit $\beta = 0$ and thus votes $v_{0,i}$ for $i \in [N_V]$.

Game $\mathbf{G}_{1,i \leq N_V}$: In this game we modify the honest voting oracle $\mathcal{O}\mathbf{Vote}(id, v_0, v_1)$ so that when it runs $\mathbf{Vote}(par, pk_1, \ldots, pk_{N_A}, id_i, v_0)$, the OR-proof for $\mathbf{c}_i = Com(v_i; \mathbf{r}_i)$ is not done honestly but simulated. Note that when simulated the proof is independent of the randomness $\mathbf{r}_i$ and vote $v_{0,i}$. The advantage of the adversary in distinguishing between **Game $\mathbf{G}_{1,i-1}$** and **Game $\mathbf{G}_{1,i}$** (where we consider **Game $\mathbf{G}_0$** as **Game $\mathbf{G}_{1,-1}$**) is zero if **Vote** is never called on id_i (i.e. id_i corresponds to a corrupted voter) and $2^{-\lambda}$ otherwise.

$$\mathbf{Adv}^{\mathbf{G}_{1,i}}_{\mathcal{A}} \leq \mathbf{Adv}^{\mathbf{G}_{1,i-1}}_{\mathcal{A}} + 2^{-\lambda}$$

Game $\mathbf{G}_2$: In this game we modify the tallying oracle of the first authority (the honest one) to make $\pi^{A,(1)}$ independent of the decrypted randomnesses $(\mathbf{r}^{(1)}_i)_{i \in [N_V]}$. i.e. we modify the $\mathbf{Tally}_1(par, pk_1, \ldots, pk_{N_A}, BB, sk_1)$ oracle so that the proof $\pi^{A,(1)}$ is computed using the simulator of the amortized proof.

$$\mathbf{Adv}^{\mathbf{G}_2}_{\mathcal{A}} \leq \mathbf{Adv}^{\mathbf{G}_{1,N_V}}_{\mathcal{A}} + 2^{-\lambda}$$

Game $\mathbf{G}_3$: In this game we modify the $\mathbf{Tally}_1(par, pk_1, \ldots, pk_{N_A}, BB, sk_1)$ oracle so that the proof $\pi^{A,(1)'}$ is computed using the simulator of the amortized proof.

$$\mathbf{Adv}^{\mathbf{G}_3}_{\mathcal{A}} \leq \mathbf{Adv}^{\mathbf{G}_2}_{\mathcal{A}} + 2^{-\lambda}$$

Game $\mathbf{G}_{4,i \leq N_V}$: In this game we modify the voting oracle for identity id_i so that it outputs the encryption $\mathbf{e}^{(1)}_i := \mathbf{Enc}(0, pk_1)$ instead of $\mathbf{e}^{(1)}_i := \mathbf{Enc}(\mathbf{r}^{(1)}_i, pk_1)$. The simulator also modifies the $\mathbf{Tally}_1$ oracle so that it uses $\mathbf{r}^{(1)}_i$ without decrypting $\mathbf{e}^{(1)}_i$.

$$\mathbf{Adv}^{\mathbf{G}_{4,i}}_{\mathcal{A}} \leq \mathbf{Adv}^{\mathbf{G}_{4,i-1}}_{\mathcal{A}} + \mathbf{Adv}^{CCA}_{\mathcal{A}}(\lambda)$$

Game $\mathbf{G}_{5,i \leq N_V}$ At this point all the values published by the oracles $\mathcal{O}\mathbf{Vote}$ and $\mathcal{O}\mathbf{Tally}_1$ are independent of the votes $(v_{0,i})_{i \in [N_V]}$ except for the commitments output by $\mathcal{O}\mathbf{Vote}$. We would like to use the hiding property of the commitment to change $\mathbf{c}^{(1)}_i = \mathbf{Com}(v^{(1)}_i, \mathbf{r}^{(1)}_i)$ to $\mathbf{c}^{(1)}_i = \mathbf{Com}(v^{(1)}_i + v_{1,i} - v_{0,i}, \mathbf{r}^{(1)}_i)$. Doing so implies that the commitment $\mathbf{c}_{u,s}^{(1)\,\prime}$ for $u = \lceil i/l \rceil$ and $s = 1$ will no longer be a commitment of zero but a commitment of $v_{1,i} - v_{0,i}$. This does not matter since the proof $\pi^{A,(1)\prime}$ is now simulated and thus independent of the existence of a witness that $\mathbf{c}_{u,s}^{(1)\,\prime}$ commits to zero.

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{5,i}} \leq \mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{5,i-1}} + \mathbf{Adv}_{\mathcal{A}}^{Hid}(\lambda)$$

Game $\mathbf{G}_{6,1\leq s\leq\lceil\log_l(N_V)\rceil-1,1\leq u\leq\lceil N_V/l^s\rceil}$: Now that the votes have been changed from $v_{0,i}$ to $v_{1,i}$ we need to change the values of the commitments of the partial sums in order for $\mathbf{c}_{u,s}^{(1)\,\prime}$ to be commitments to zero, this will be needed to change $\pi^{A,(1)\prime}$ back to an honest proof. To do so we let $v_{0,u,s} = \sum\limits_{x=(u-1)l+1}^{ul} v_{0,x,s-1}^{(j)}$ and $v_{1,u,s} = \sum\limits_{x=(u-1)l+1}^{ul} v_{1,x,s-1}^{(j)}$ (where $v_{0,i,0} = v_{0,i}$ and $v_{1,i,0} = v_{1,i}$). We can now change the commitments $\mathbf{c}_{u,s}^{(j)} = \mathbf{Com}\left(v_{u,s}^{(j)}; \mathbf{r}_{u,s}^{(j)}\right)$ to $\mathbf{c}_{u,s}^{(j)} = \mathbf{Com}\left(v_{u,s}^{(j)} + v_{1,u,s} - v_{0,u,s}; \mathbf{r}_{u,s}^{(j)}\right)$ and the partial sums are verified.

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{6,i}} \leq \mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{6,i-1}} + \mathbf{Adv}_{\mathcal{A}}^{Hid}(\lambda)$$

Game $\mathbf{G}_{7,i\leq N_V}$: We revert the randomness encryptions to $\mathbf{e}_i^{(1)} = \mathbf{Enc}(\mathbf{r}_\mathbf{i}^{(1)})$, this modification is consistent with the tallying scheme as the randomness used have not been modified.

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{7,i}} \leq \mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{7,i-1}} + \mathbf{Adv}_{\mathcal{A}}^{CCA}(\lambda)$$

Game $\mathbf{G}_8$: We compute the proof $\pi^{A,(1)\prime}$ honestly. This is possible because all commitments $\mathbf{c}_{u,s}^{(1)\,\prime}$ are commitments of zero made with the appropriate randomnesses.

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_8} \leq \mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{7,N_V}} + 2^{-\lambda}$$

Game $\mathbf{G}_9$: Similarly we compute the proof $\pi^{A,(1)}$ honestly.

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_9} \leq \mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_8} + 2^{-\lambda}$$

Game $\mathbf{G}_{10,i\leq N_V}$: We compute the proof π_i honestly. This is possible because $\mathbf{c}_i$ is still a commitment to either zero or one with the same randomness as before.

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{10,i}} \leq \mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{10,i-1}} + 2^{-\lambda}$$

Game $\mathbf{G}_{11}$ We run $Exp_{\mathcal{A}}^{priv,1}$, this game is identical to **Game** $\mathbf{G}_{10,N_V}$.

$$\mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{11}} = \mathbf{Adv}_{\mathcal{A}}^{\mathbf{G}_{10,N_V}}$$

□

We now consider the consistency of our voting scheme. For a tighter security proof we will assume a slight modification on the algorithm $\mathbf{Tally}_j$: Rather than using only the hash of the corresponding commitments to compute the challenges for the proofs $\pi^{A,(j)}$ and $\pi^{A,(j)\prime}$, the authorities will hash the whole bulletin board (which among other things contains the relevant commitments). Let $H(BB) = (chl_1, \ldots, chl_{N_A})$ and $H'(BB) = (chl'_1, \ldots, chl'_{N_A})$ be these hashes, the j^{th} authority will then use chl_j and chl'_j as challenges for his proofs $\pi^{A,(j)}$ and $\pi^{A,(j)\prime}$. In doing so we guarantee that we can extract witnesses for all N_A proofs $\pi^{A,(j)}$ in one rewinding of the random oracle O_H.

Theorem A.2. *Let $\mathcal{A}$ be an adversary with non negligible advantage $\mathbf{Adv}_{\mathcal{A}}^{cons}(\lambda) = \varepsilon$ in experiment $\mathbf{Exp}_{\mathcal{A}}^{cons}(\lambda)$. Using $\mathcal{A}$ we construct an extractor $\mathcal{E}$ who breaks the binding property of* **Com** *in expected time $1/\varepsilon + negl(\lambda)$*

Proof. We will assume that when $\mathcal{A}$ succeeds in $\mathbf{Exp}_{\mathcal{A}}^{cons}(\lambda)$ all the bulletins on *BB* were accepted by all the authorities, we can make this assumption because any bulletin that was not accepted is effectively discarded (the authorities do not include it in their amortized proofs nor in the final tally). We can thus use N_V as the number of accepted tallies. $\mathcal{E}$ starts by running $\mathcal{A}$ until it succeeds in $\mathbf{Exp}_{\mathcal{A}}^{cons}(\lambda)$, i.e. until it outputs a bulletin board *BB* that verifies correctly and such that $r < h_1$ or $r > N_V - h_0$. Using the soundness of the proofs $\pi^{A,(j)}$, $\mathcal{E}$ rewinds $\mathcal{A}$ and obtains witnesses $\hat{\mathbf{r}}_i^{(j)}$, for $i \leq N_V$ and $j \leq N_A$, such that $2\mathbf{A}\hat{\mathbf{r}}_i^{(j)} = 2\mathbf{a}_i^{(j)}$. Let $\hat{v}_i^{(j)} = \mathbf{b}_i^{(j)} - 2^{-1}\mathbf{B}\hat{\mathbf{r}}_i^{(j)}$, then we have:

$$2\mathbf{c}_i^{(j)} = \mathbf{C}\hat{\mathbf{r}}_i^{(j)} + 2\begin{bmatrix}\mathbf{0}\\ \hat{v}_i^{(j)}\end{bmatrix} \tag{1}$$

Suppose there exists $i \leq N_V$ such that $\hat{v}_i := \sum_{j=1}^{N_A} \hat{v}_i^{(j)} \mod q$ is not in $\{0,1\}$, $\mathcal{E}$ runs the soundness extractor for π_i^V and obtains $\bar{\mathbf{r}}_i$, $\bar{f} \in \mathcal{R}$ and $\bar{v}_i \in \{0,1\}$ such that:

$$\bar{f}\mathbf{c}_i = \mathbf{C}\bar{\mathbf{r}}_i + \bar{f}\begin{bmatrix}\mathbf{0}\\ \bar{v}_i\end{bmatrix} \tag{2}$$

By summing equations (1) over $j \in [N_A]$ and multiplying them by $\bar{f}$, and by multiplying (2) by 2 we obtain the following:

$$\mathbf{c}'_i := 2\bar{f}\mathbf{c}_i = \mathbf{C}2\bar{\mathbf{r}}_i + \begin{bmatrix}\mathbf{0}\\ 2\bar{f}\hat{v}_i\end{bmatrix} \tag{3}$$

$$\mathbf{c}'_i = 2\bar{f}\mathbf{c}_i = \mathbf{C}\bar{f}\sum_{j=1}^{N_A}\hat{\mathbf{r}}_i^{(j)} + \begin{bmatrix}\mathbf{0}\\ 2\bar{f}\bar{v}_i\end{bmatrix} \tag{4}$$

Since we assumed that $\hat{v}_i \notin \{0,1\}$ and we know $\bar{v}_i \in \{0,1\}$, if we have $\|2\bar{r}_i\| \leq B_r$ and $\left\|\bar{f}\sum_{j=1}^{N_A}\hat{\mathbf{r}}_i^{(j)}\right\| \leq B_r$ (which we will ensure in Section 5), then $\mathcal{E}$ has successfully opened $\mathbf{c}'_i$ to two different messages and thus broken the binding property of **Com**.
We can now assume that for every $i \leq N_V$, $\hat{v}_i \in \{0,1\}$. For $s \in [\lceil\log_l N_V\rceil]$, $u \in [\lceil N_V/l^s\rceil]$ and $j \in [N_A]$, let $\hat{v}_{u,s}^{(j)} = \sum\limits_{x=(u-1)l+1}^{ul} \hat{v}_{x,s-1}^{(j)}$ (where $\hat{v}_{u,0}^{(j)} := \hat{v}_u^{(j)}$), let $\mathbf{c}_{u,s}^{(j)\prime} = \mathbf{c}_{u,s}^{(j)} - \sum\limits_{x=(u-1)l+1}^{ul} \mathbf{c}_{x,s-1}^{(j)}$. $\mathcal{E}$ runs the soundness extractor for $\pi^{A,(j)\prime}$ and obtains $\hat{\mathbf{r}}_{u,s}^{(j)\prime}$ such that $2\mathbf{c}_{u,s}^{(j)\prime} = \mathbf{C}\hat{\mathbf{r}}_{u,s}^{(j)\prime}$.
Using the extraction for $\pi^{A,(j)}$ we already have $\hat{\mathbf{r}}_{u,s}^{(j)}$ such that $2\mathbf{a}_{u,s}^{(j)} = \mathbf{A}\hat{\mathbf{r}}_{u,s}^{(j)}$, from which we obtain messages $m_{u,s}^{(j)}$ such that:

$$2\mathbf{c}_{u,s}^{(j)} = \mathbf{C}\hat{\mathbf{r}}_{u,s}^{(j)} + \begin{bmatrix}\mathbf{0}\\ m_{u,s}^{(j)}\end{bmatrix} \tag{5}$$

Now suppose that for all u, s, j we have $m_{u,s}^{(j)} = \hat{v}_{u,s}^{(j)}$ this implies that the ciphertext $\mathbf{c}_{1,\lceil\log_l N_V\rceil}^{(j)}$ has an extraction:

$$2\mathbf{c}_{1,\lceil\log_l N_V\rceil}^{(j)} = \mathbf{C}\hat{\mathbf{r}}_{1,\lceil\log_l N_V\rceil}^{(j)} + 2\begin{bmatrix}0\\ \hat{v}_{1,\lceil\log_l N_V\rceil}^{(j)}\end{bmatrix} \tag{6}$$

By construction of $\hat{v}_{u,s}^{(j)}$ we have $\hat{v}^{(j)}_{1,\lceil \log_l N_V \rceil} = \sum_{i=1}^{N_V} \hat{v}_i^{(j)}$. Since the bulletin board verifies correctly we know that $\mathbf{c}^{(j)}_{1,\lceil \log_l N_V \rceil}$ opens to plaintext $v^{(j)}$ such that $r = \sum_{j=1}^{N_V} v^{(j)}$ by the binding property of **Com** we have that $v^{(j)} = \sum_{i=1}^{N_V} \hat{v}_i^{(j)}$ and thus:

$$\begin{aligned} r &= \sum_j v^{(j)} \\ &= \sum_j \sum_i \hat{v}_i^{(j)} \\ &= \sum_i \hat{v}_i \\ &= \sum_{i \in HV'} v_i + \sum_{i \in CV'} \hat{v}_i \\ &= h_1 + \sum_{i \in CV'} \hat{v}_i \end{aligned}$$

Since we have shown that for all $i \leq N_V$, $\hat{v}_i \in \{0,1\}$ this implies that $h_1 \leq r \leq N_V - h_0$ which contradicts the fact that $\mathcal{A}$ wins experiment $\mathbf{Exp}^{cons}_{\mathcal{A}}(\lambda)$. We have thus shown that there exist u, v, j such that $m_{u,s}^{(j)} \neq \hat{v}_{u,s}^{(j)}$, i.e. one of the partial sum does not commit to the proper value.

Fix a $j \leq N_A$ for which there exist such a commitment and let u, s be the smallest such triple (in lexicographic order). In particular this implies that $s \geq 1$ (as we have proven that all $\mathbf{c}_i^{(j)}$ open to $\hat{v}_i^{(j)}$) and that for $x \in ((u-1)l + 1, ul)$ we have the following witness extracted from $\pi^{A,(j)}$:

$$2\mathbf{c}^{(j)}_{x,s-1} = C\hat{\mathbf{r}}^{(j)}_{x,s-1} + \begin{bmatrix} \mathbf{0} \\ \hat{v}^{(j)}_{x,s-1} \end{bmatrix} \tag{7}$$

By summing equation (7) over $x \in ((u-1)l + 1, ul)$ and subtracting the extraction for $\mathbf{c}_{u,s}^{(j)}$ we obtain:

$$\begin{aligned} 2\mathbf{c}^{(j)\,\prime}_{u,s} &= \mathbf{c}^{(j)}_{u,s} - \sum_{x=(u-1)l+1}^{ul} \mathbf{c}^{(j)}_{x,s-1} \\ &= \mathbf{C}\left(\hat{\mathbf{r}}^{(j)}_{u,s} - \sum_{x=(u-1)l+1}^{ul} \hat{\mathbf{r}}^{(j)}_{x,s-1}\right) + \begin{bmatrix} \mathbf{0} \\ m^{(j)}_{u,s} - \hat{v}^{(j)}_{u,s} \end{bmatrix} \end{aligned} \tag{8}$$

From the extraction of $\pi^{A,(j)\prime}$ we had $2\mathbf{c}^{(j)\,\prime}_{u,s} = \mathbf{C}\hat{\mathbf{r}}^{(j)\prime}_{u,s}$. We know that $m^{(j)}_{u,s} \neq \hat{v}^{(j)}_{u,s}$, which implies that if $\left\| \hat{\mathbf{r}}^{(j)}_{u,s} - \sum_{x=(u-1)l+1}^{ul} \hat{\mathbf{r}}^{(j)}_{x,s-1} \right\| \leq B_r$ and $\left\| \hat{\mathbf{r}}^{(j)\prime}_{u,s} \right\| \leq B_r$ (which we ensure in Section 5) then we have found two distinct openings for $2\mathbf{c}^{(j)\,\prime}_{u,s}$ and broken the binding property of **Com**. □

B E-VOTING FOR MULTIPLE CANDIDATES

Though we have described our E-Voting scheme as being for votes in $\{0,1\}$, it can be extended to votes in $\{0,1\}^k$ for any constant k at a small cost. Adapting the scheme and security definitions of Section 4 is straightforward and does not pose any problem.[4] Our commitment scheme can be directly adapted to having message space $\mathcal{R}_q^k$ by taking $\mathbf{C} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$ with $\mathbf{A} \in \mathcal{R}_q^{d \times (2d+k)}$, and $\mathbf{B} \in \mathcal{R}_q^{k \times (2d+k)}$, the resulting commitments will be somewhat larger but the security remains the same. The amortized proofs can be used directly with this new commitment scheme (since they can be used with any one-way function). To obtain a (relaxed) proof that a commitment is in $\{0,1\}^k$ it is sufficient to use k parallel proofs for commitments in $\{0,1\}$, we describe this construction in more detail as its soundness is not trivial.

It is not directly clear that a proof for commitments in $\{0,1\}^k$ can be obtained by running k proofs for (partial) commitments in $\{0,1\}$. Indeed, for a public commitment key $\mathbf{C} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$, let $\mathbf{B}_i$ be the i^{th} row of $\mathbf{B}$, let $\mathbf{D}_i = \begin{bmatrix} \mathbf{A} \\ \mathbf{B}_i \end{bmatrix}$. For a commitment $\mathbf{c} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \mathbf{Cr} + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}$ we define in the same manner $\mathbf{d}_i = \mathbf{D}_i \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ m_i \end{bmatrix}$ where m_i is the i^{th} element of $\mathbf{m}$. Our aim is to run the proof Π_{OR} on each of the $\mathbf{d}_i$ to prove that m_i is in $\{0,1\}$ and conclude that $\mathbf{m}$ is in $\{0,1\}^k$. However each proof will have a different $\bar{f}_i$ in the soundness extraction. i.e. for $i \neq j$ we will obtain $\bar{\mathbf{r}}_i, \bar{\mathbf{r}}_j, \bar{f}_i, \bar{f}_j$ such that:

- $\bar{f}_i \mathbf{d}_i = \mathbf{D}_i \bar{\mathbf{r}}_i + \bar{f}_i \begin{bmatrix} \mathbf{0} \\ m_i \end{bmatrix}$
- $\bar{f}_j \mathbf{d}_j = \mathbf{D}_j \bar{\mathbf{r}}_j + \bar{f}_j \begin{bmatrix} \mathbf{0} \\ m_j \end{bmatrix}$

Remark that since the top part of both $\mathbf{D}_i$ and $\mathbf{D}_j$ are equal to $\mathbf{A}$ and the top part of both $\mathbf{d}_i$ and $\mathbf{d}_j$ are equal to $\mathbf{a}$, we have $\bar{f}_i \mathbf{a} = \mathbf{A}\bar{\mathbf{r}}_i$ and $\bar{f}_j \mathbf{a} = \mathbf{A}\bar{\mathbf{r}}_j$ and thus $\mathbf{A}(\bar{f}_j \bar{\mathbf{r}}_i - \bar{f}_i \bar{\mathbf{r}}_j) = 0$. Which entails that $\bar{f}_j \bar{\mathbf{r}}_i = \bar{f}_i \bar{\mathbf{r}}_j$ if $\|\bar{f}_j \bar{\mathbf{r}}_i - \bar{f}_i \bar{\mathbf{r}}_j\| \leq 2B_r$ (by the binding property of the commitment scheme) which will be true for our parameters. Now if we multiply our second equation by $\bar{f}_i$ and then divide by $\bar{f}_j$ (which according to Lemma 2.8 will have an inverse for well chosen parameters) we obtain:

$$\bar{f}_i \mathbf{d}_j = \mathbf{D}_j \bar{\mathbf{r}}_i + \bar{f}_i \begin{bmatrix} \mathbf{0} \\ m_j \end{bmatrix}$$

By applying the same reasoning to all $j \neq i$ and concatenating over the rows $\mathbf{b}_i$ we can extract a witness for the commitment $\mathbf{c}$:

$$\bar{f}_i \mathbf{c} = \mathbf{C}\bar{\mathbf{r}}_i + \bar{f}_i \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}$$

We thus obtain a proof of knowledge for commitments in $\{0,1\}^k$ by using k proofs for commitments in $\{0,1\}$, i.e. for the binary relation:

$$\begin{aligned} \mathfrak{R}_{OR} = \Bigg\{ &(\mathbf{c}, \mathbf{r}) \in \mathcal{R}_q^{d+k} \times \mathcal{R}^{2d+k}, \\ &\mathbf{c} = \mathbf{Cr} + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}, \mathbf{m} \in \{0,1\}^k, \|\mathbf{r}\| \leq B_{OR} \Bigg\} \end{aligned}$$

and the relation for the soundness extractor:

$$\begin{aligned} \mathfrak{R}'_{OR} = \Bigg\{ &(\mathbf{c}, \mathbf{r}, f) \in \mathcal{R}_q^{d+k} \times \mathcal{R}^{2d+k} \times \mathcal{R}_q, f\mathbf{c} = \mathbf{Cr} + f \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}, \\ &\mathbf{m} \in \{0,1\}^k, \|\mathbf{r}\| \leq B'_{OR}, 0 < \|f\| \leq 2\sqrt{60} \Bigg\} \end{aligned}$$

We have the following theorem:

[4] The consistency definition can be extended to k candidates by considering the votes v, the number of honest 1 votes h_1, the number of honest 0 votes h_0, the number of approved voters t, and the result $\mathbf{r}$ as k-dimensional vectors. An adversary then wins the consistency game if there exists $j < k$ such that $\mathbf{r}_j < h_{1,j}$ or $\mathbf{r}_j > t_j - h_{0,j}$.

Theorem B.1. *For a matrix* $\mathbf{C} \in \mathcal{R}_q^{(d+k)\times(2d+k)}$ *decomposed in matrices* $\mathbf{D}_i \in \mathcal{R}_q^{(d+1)\times(2d+k)}$ *as specified above, let* $\Pi_{OR,\mathbf{D}_i}$ *be the OR proof defined in Section 3.2 using the matrix* $\mathbf{D}_i$ *as commitment key.*
If $\sigma_{OR} \geq 22 * \sqrt{60}B_{OR}$, $B'_{OR} \geq 2\sqrt{n(2d+k)}\sigma_{OR}$, *then* $\Pi_{OR^k} = \Pi_{OR,D_1} \wedge \ldots \wedge \Pi_{OR,D_k}$ *is a zero knowledge proof of knowledge for the language* $\mathfrak{R}_{OR}$, *with soundness extractor in* $\mathfrak{R}'_{OR}$.

Now that we have a proof for message in $\{0,1\}^k$ our voting scheme can be adapted to k candidates in a completely straightforward manner.