Lattice-based proof of a shuffle

Nuria Costa[1], Ramiro Martínez[2], and Paz Morillo[2]

[1] Scytl Secure Electronic Voting
nuria.costa@scytl.com
[2] Universitat Politècnica de Catalunya
ramiro.martinez@upc.edu
paz.morillo@upc.edu

Abstract. In this paper we present the first fully post-quantum proof of a shuffle for RLWE encryption schemes. Shuffles are commonly used to construct mixing networks (mix-nets), a key element to ensure anonymity in many applications such as electronic voting systems. They should preserve anonymity even against an attack using quantum computers in order to guarantee long-term privacy. The proof presented in this paper is built over RLWE commitments which are perfectly binding and computationally hiding under the RLWE assumption, thus achieving security in a post-quantum scenario. Furthermore we provide a new definition for a secure mixing node (mix-node) and prove that our construction satisfies this definition.

Keywords: mix-nets, e-voting, post-quantum, RLWE encryption, RLWE commitment, proof of a shuffle.

1 Introduction

In the last years, several countries have been introducing electronic voting systems to improve their democratic processes, in particular, they provide voters with the chance to cast their votes from anywhere. Anonymity and verifiability are two fundamental requirements for internet voting systems that seem to be contradictory. Anonymity requires that the link between the vote and the voter who has cast it must remain secret during the whole process, while verifiability requires that all the steps of the electoral process - vote casting, vote storage and vote counting - can be checked by the voters, the auditors or external observers. One of the resources used by the actual internet voting systems to achieve anonymity are mixing networks (mix-nets). Informally we can define a mix-net as a multiparty protocol that, given a number of encrypted messages at the input, performs a permutation over them followed by a cryptographic transformation using a re-encryption and/or a decryption algorithm. This operation is called a shuffle [9] and it is done in such a way that the correlation between the input and the output of the process is hidden, and it is not possible to trace it back. The proof of the shuffle guarantees that the ciphertexts at the output of the mix-net are those at its input permuted and re-encrypted/decrypted, without revealing any secret information. One way to construct a mix-net is to define several mixing nodes (mix-nodes) each one performing in turns this operation. It is clear that if at least one of the nodes is honest, unlinkability is preserved.

On the other hand, in order to build verifiable systems one key instrument is the Bulletin Board: a public place where all the audit information of the election (encrypted votes, election configuration, proof of a shuffle, . . .) is published by authorized parties and can be verified by anyone: voters, auditors or third parties. However, once published in the Bulletin Board anyone can save a copy, and long-term privacy may not be ensured by encryption algorithms used nowadays, for example due to the efficient quantum algorithm given by Shor [29] that breaks computational problems such as the discrete logarithm (DL) or the integer factorization problems. Learning how a person voted some years ago may have political, as well as personal implications.

Some cryptosystems have appeared in the last years that are believed to be secure against quantum attacks: hash-based, code-based, lattice-based or multivariate-quadratic-equations. Lattice-based cryptography is a great promise to get cryptosystems that will remain secure in the post-quantum era [23]. These ones enjoy strong security guarantees from worst-case hardness, meaning that breaking their security implies finding an efficient algorithm for solving any instance of the underlying lattice problem, e.g., the Shortest Vector Problem (SVP) or the Closest Vector Problem (CVP). Furthermore, these constructions mainly involve linear operations such as matrix and vector sum or multiplication modulo relatively small integers, which make them highly parallelizable and consequently faster in certain contexts. Given the interest aroused by this type of cryptography, several lattice-based protocols have been proposed like public key encryption schemes, digital signatures schemes, hash functions, identity-based encryption schemes or Zero-Knowledge Proofs of Knowledge (ZKPoK). Our contribution increases the literature of the latter, providing a fully lattice-based proof of a shuffle that will remain secure in a post-quantum scenario.

To the best of our knowledge there are two proposed e-voting schemes [10,15] that are constructed using lattices. They both follow an alternative approach without shuffling, making use of the homomorphic property of their encryption schemes to compute the tally. However mix-net based schemes are more flexible and provide a better support for complex electoral processes.

On the other hand [11] and [32] give proofs of a shuffle for lattice-based cryptography. The first requires Pedersen commitments (based on the DL problem). The latter requires a Fully Homomorphic Encryption scheme, and works with any homomorphic commitment scheme, that is, using the lattice-based commitment scheme presented in [4] their proof is fully post-quantum.

We propose a proof of a shuffle that is fully constructed over lattice-based cryptography and the first for RLWE encryption schemes, which makes it secure in a post-quantum scenario. The proof uses a commitment scheme which is perfectly binding and computationally hiding under the Learning With Errors over Rings (RLWE) assumption. This lattice computational problem has been shown to be as hard as certain worst-case problems in ideal lattices (such as SVP and CVP in ideal lattices) and thus resistant to quantum attacks. We also provide a formal definition for security of a mix-node and prove security of our proposal using the sequence of games approach.

1.1 Previous work

After the introduction of the idea of a shuffle by Chaum in 1981 [9], several schemes have been proposed. The first universally verifiable mix-net is presented in [28] and gives a proof to check the correctness of the shuffle. Later, several solutions for an efficient universally verifiable mix-net are proposed [1,2,3,22] and in [17] Furukawa and Sako suggest a paradigm based on permutation matrices in the common reference string model (CRS) for proving the correctness of a shuffle, that was improved in [16,20]. The latest proposal for a CRS based proof of a shuffle is [8] by Bünz *et al.* Wikström also uses this idea of the permutation matrix and presents in [37] a proof of a shuffle that can be split in an offline and online phase in order to reduce the computational complexity in the online part.

On the other hand, Neff [24] proposes another paradigm based on polynomials being identical under permutation of their roots, obtaining Honest Verifier Zero-Knowledge (HVZK) proof and improved later in [18,25] with the drawback that these constructions are 7-move proofs. Unlike previous proposals, Groth and Ishai [19] and Bayer and Groth [6] give a practical shuffle argument with sublinear communication complexity.

The proof of a shuffle presented in this paper requires lattice-based ZKPoK to prove that some hidden elements have small norm and also that several committed elements satisfy a polynomial relation. As these proofs are generally costly we are going to use amortized protocols to reduce the communication cost. The first amortized protocol is presented in [12] by Cramer *et al.*, it is improved first by del Pino and Lyubashevsky [14] and later by Baum and Lyubashevsky in [5].

Recently, Costa *et al.* [11] have presented a proof of a shuffle based on lattices but it cannot be considered fully post-quantum since they use Pedersen commitments, whose binding property relies on the DL problem. Moreover in [11] there is no formal definition of security, necessary to precisely know how it can be embedded in a larger construction. Strand [32] presents a verifiable shuffle for the GSW cryptosystem using homomorphic commitment schemes. Using the lattice-based commitment scheme [4] makes the proof fully post-quantum. Additionally, there have been some proposals for a lattice-based universal re-encryption for mix-nets [30] but none of them give a proof of a shuffle.

In [36] Wikström provides a definition of security for a single re-encryption mix-node. It is important to note that as Wikström remarks this is not enough to completely ensure privacy since a definition of security of a complete mix-net must involve several other aspects, regarding validity of the input messages or decryption proofs.

1.2 Our contribution

We propose a proof of a shuffle fully constructed over lattices. It is based on the technique introduced by Bayer and Groth in [6] to construct a shuffle argument;

nevertheless it is not a direct adaptation of it since working with lattices requires different techniques to be applied.

The first step of the proof, that is also the first difference with [6], consists on committing the re-encryption parameters in order to demonstrate that they meet certain constraints. This is done using the commitment scheme and the ZKPoK proposed by Benhamouda *et al.* [7] which are perfectly binding and computationally hiding under the RLWE assumption and satisfy special soundness and special HVZK. The next step consists on proving knowledge of the permutation. The general idea here is to prove that two sets contain the same elements. This is done by computing two polynomials, each of them having as roots the elements of each set, and proving that both polynomials are equal.

The last step will prove knowledge of the re-encryption parameters, and this introduces another difference between Bayer and Groth's protocol and ours. While they demonstrate that there exists a linear combination of the parameters such that an equality holds, we have to use a different technique, since the re-encryption parameters in a RLWE re-encryption scheme are taken from an error distribution and a linear combination of them would imply the error grows uncontrollably, causing decryption errors.

Finally, we give a definition of security, based on the one proposed by Wikström in [36], and we provide a proof of security for our mix-node. His proposal implies that no adversary can properly compute two indices for the input and the output respectively such that the messages encrypted in the corresponding ciphertexts are the same, except with a probability negligibly close to the probability given by a random guess. In his definition the adversary might have some knowledge of correlations between the input messages. We provide a definition of security allowing the adversary to have full control over the input of the mix-node, and we prove that our construction meets this definition.

Organization of the paper. In section 2 we introduce some notation and give some cryptographic background necessary to understand the proof presented in section 4. In section 3 we describe the computational problem on which the security of our scheme is based and we also give a description of a RLWE-based commitment scheme. Finally in section 4 we present our fully post-quantum proof of a shuffle and the results about the security of the mix-node. We briefly conclude in section 5.

2 Preliminaries

We denote column vectors by boldface lower-case roman letters, $\boldsymbol{v}$ or $\boldsymbol{w}$. Matrices are represented by boldface upper-case roman letters, $\boldsymbol{M}$ or $\boldsymbol{A}$. Given two vectors $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{Z}_q^N$, we define the standard inner product in $\mathbb{Z}_q^N$ as $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \sum_{i=1}^N v_i w_i$, the l_∞ norm as $\|\boldsymbol{v}\|_\infty = \max_{1 \le i \le N} |v_i|$ and the general norm l_p as $\|\boldsymbol{v}\|_p = (\sum_{i=1}^N |v_i|^p)^{1/p}$ for $p \ge 1$.

We let $\lfloor x \rfloor$ denote the largest integer not greater than x, and $\lfloor x \rceil := \lfloor x + 1/2 \rfloor$ denote the integer closest to x, with ties broken upward.

We write $a \xleftarrow{\$} A$ when a is sampled uniformly at random from a set A, and $a \xleftarrow{\$} D$ if it is drawn according to the distribution D.

Finally, in order to avoid confusions we are going to identify the ciphertexts' elements with the subscript E, and those corresponding to the commitments with subscript C. When working with lattices we are going to follow the notation proposed in [21].

The ZKPoK between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ constructed in this paper satisfies the properties of *completeness*, *special soundness* and *special HVZK* as they are defined in [13]. We will use them to prove knowledge of valid openings of commitments that satisfy several polynomial relations.

2.1 Generalized Schwartz–Zippel lemma

The proof of a shuffle presented in this paper uses a generalized version of the Schwartz-Zippel lemma to prove polynomial equalities. This lemma works in general commutative rings that are not necessarily integral domains. Unlike Bayer and Groth we need the generalized version since we work with polynomials whose coefficients belong to another ring of polynomials.

Lemma 1. *Let $p \in R[x_1, x_2, \ldots, x_n]$ be a non-zero polynomial of total degree $d \geq 0$ over a commutative ring R. Let S be a finite subset of R such that none of the differences between two elements of S is a divisor of 0 and let $r_1, r_2, \ldots, r_n$ be selected at random independently and uniformly from S. Then:*
$\Pr[p(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|}$.

We will use this lemma to prove that two polynomials, p_1 and p_2, are equal with overwhelming probability if $p_1(r_1, r_2, \ldots, r_n) - p_2(r_1, r_2, \ldots, r_n) = 0$ for $r_1, r_2, \ldots, r_n \xleftarrow{\$} S$. The proof of this generalization directly follows from the original proof of the lemma. We have included it in appendix A for the reader interested on it.

3 Ideal Lattices

A lattice is a set of points in an n-dimensional space with a periodic structure. We are going to work with *ideal* lattices that have some extra algebraic structure and introduce some redundancy allowing a more compact representation and thus reducing significantly the storage space. We refer the interested reader to [26] for a survey on lattices.

Let $R_q = \mathbb{Z}_q[x]/\langle f(x) \rangle$ be the ring of polynomials modulo $f(x) = x^n + 1$ for n a power of 2, which makes the polynomial irreducible over the rationals. The ideal lattice $\mathcal{L}(a)$ generated by $a(x) = a_1 + a_2 x + \ldots + a_n x^{n-1} \in R_q$ is the set of polynomials $v(x)$ obtained as $v(x) = a(x) \cdot p(x) \mod x^n + 1$, where $p(x) \in R_q$.

There is currently no known way to take a significant advantage of this extra structure introduced in this class of ideal lattices, and the running time required to solve lattice problems on such lattices is comparable to that for general lattices.

3.1 RLWE problem

The security of lattice-based cryptosystems relies on the hardness of solving some computational problems on lattices, such as the Learning With Errors (LWE).

Lyubashevsky *et al.* [21] introduced in 2010 the ideal lattice based variant of LWE, called Ring Learning With Errors (RLWE). This was motivated by the necessity of constructing efficient LWE-based cryptosystems.

Definition 1 (RLWE Distribution). *For a secret $s \in R_q$, the RLWE distribution $\mathcal{A}_{s,\chi}$ over $R_q \times R_q$ is sampled choosing $a \in R_q$ uniformly at random, $e \xleftarrow{\$} \chi^n$ (that is, $e \in R_q$ with its coefficients drawn from χ), and outputting samples of the form $(a, b = a \cdot s + e \mod q) \in R_q \times R_q$.*

Analogously to LWE [27], the goal will be either to distinguish random linear equations, perturbed by a small amount of noise, from truly uniform pairs, or recover the secret $s \in R_q$ from arbitrarily many noisy products. Usually the error distribution χ is a *discrete Gaussian distribution* on $\mathbb{Z}$, that is $\chi = D_\sigma$, where σ is the standard deviation.

Hardness of RLWE . Certain instantiations of RLWE are supported by worst-case hardness theorems [21], related to the Shortest Vector Problem (SVP). For the error distribution χ where $\sigma \geq \omega(\sqrt{\log n})$, and for any ring, there exist a quantum reduction from the $\gamma(n)$-SVP problem to the RLWE problem to within $\gamma(n) = \mathcal{O}(\sqrt{n} \cdot q/\sigma)$. Additionaly, RLWE becomes no easier to solve even if the secret s is chosen from the error distribution, rather than uniformly [21].

3.2 RLWE encryption scheme

The additive homomorphic RLWE encryption scheme proposed in [21] consists of three algorithms ($\mathsf{KeyGen_E}$,$\mathsf{Encrypt}$,$\mathsf{Decrypt}$) defined below. We denote the security parameter as κ.

- $\mathsf{KeyGen_E}(1^\kappa)$: Given a uniformly random $a_\mathsf{E} \in R_q$ and two *small* elements $s, e \in R_q$ drawn from the error distribution χ^n, the public key is an RLWE sample $(a_\mathsf{E}, b_\mathsf{E}) = (a_\mathsf{E}, a_\mathsf{E} \cdot s + e) \in R_q \times R_q$ and the secret key is s.
- $\mathsf{Encrypt}((a_\mathsf{E}, b_\mathsf{E}), r_\mathsf{E}, e_{\mathsf{E},u}, e_{\mathsf{E},v}, z)$: Given three random small elements r_E, $e_{\mathsf{E},u}$, $e_{\mathsf{E},v} \in R_q$ drawn from the error distribution χ^n, the encryption of an n-bit message $z \in \{0,1\}^n$ (identified as a polynomial of degree $n-1$ with coefficients 0 or 1) is $(u, v) = (a_\mathsf{E} \cdot r_\mathsf{E} + e_{\mathsf{E},u}, b_\mathsf{E} \cdot r_\mathsf{E} + e_{\mathsf{E},v} + \lfloor \frac{q}{2} \rceil z) \in R_q \times R_q$.
- $\mathsf{Decrypt(s,(u,v))}$: Given the secret key and the ciphertext this algorithm computes: $v - u \cdot s = (r_\mathsf{E} \cdot e - s \cdot e_{\mathsf{E},u} + e_{\mathsf{E},v}) + \lfloor \frac{q}{2} \rceil z \mod q$. Then recovers each bit of z by rounding each coefficient to 0 or $\lfloor \frac{q}{2} \rceil$.

Correctness. Notice that in case of lack of error the decryption would always be correct since the algorithm will return directly 0 or $\lfloor \frac{q}{2} \rceil$ depending on the encrypted bit. Given that, a decryption error will occur if the coefficients of $(r_\mathsf{E} \cdot e - s \cdot e_{\mathsf{E},u} + e_{\mathsf{E},v})$ have magnitude greater than $q/4$.

As the messages encrypted using this scheme will pass through a mixing process we will need to also re-encrypt them. Due to the homomorphic property of the scheme we can compute the re-encryption just adding to the original ciphertext the encryption of the element 0.

- $\mathsf{Re\text{-}encrypt}((u,v),(a_{\mathsf{E}},b_{\mathsf{E}}),r'_{\mathsf{E}},e'_{\mathsf{E},u},e'_{\mathsf{E},v})$: Given the small elements $r'_{\mathsf{E}}, e'_{\mathsf{E},u}$, $e'_{\mathsf{E},v}$ drawn from the error distribution χ^n, the re-encryption of a ciphertext (u,v) is $(u',v')=(u,v)+\mathsf{Encrypt}((a_{\mathsf{E}},b_{\mathsf{E}}),r'_{\mathsf{E}},e'_{\mathsf{E},u},e'_{\mathsf{E},v},0)\in R_q\times R_q$.

Security. RLWE encryption scheme and consequently the RLWE re-encryption scheme are semantically secure based on the RLWE assumption. It is demonstrated that if there exists a polynomial-time algorithm that distinguishes between two encryptions then there exists another algorithm able to distinguish between $\mathcal{A}_{s,\chi}$ and a uniformly random distribution over R_q for a non-negligible fraction of all possible s. Notice that, even though these schemes do not achieve circuit privacy, the secrecy of the shuffle is not affected since the randomness used during the encryption and re-encryption procedures is never revealed. In order to demonstrate that the random values are of the right form, that is, that they are small enough, we use zero-knowledge proofs.

3.3 Commitments from RLWE

The commitment scheme used to build our proof of a shuffle is that described by Benhamouda *et al.* in [7] and consists of the following three algorithms:

- $\mathsf{KeyGen_C}(1^\kappa)$: given as input the security parameter κ (we omit the details about κ here and we refer the reader to [7]) this algorithm generates the public commitment key $pk_{\mathsf{C}}=(\boldsymbol{a}_{\mathsf{C}},\boldsymbol{b}_{\mathsf{C}})$ where $\boldsymbol{a}_{\mathsf{C}},\boldsymbol{b}_{\mathsf{C}}\overset{\$}{\leftarrow}(R_q)^k$, $q\equiv 3 \mod 8$ is prime and n is a power of 2.
- Com: in order to commit to a message $m\in R_q$, the algorithm chooses $r_{\mathsf{C}}\overset{\$}{\leftarrow} R_q$ and $\boldsymbol{e}_{\mathsf{C}}\overset{\$}{\leftarrow}D^k_{\sigma_e}$ conditioned on $\|\boldsymbol{e}_{\mathsf{C}}\|_\infty\le n$ and computes:

$$\boldsymbol{c}=\mathsf{Com}_{\boldsymbol{a}_{\mathsf{C}},\boldsymbol{b}_{\mathsf{C}}}(m;r_{\mathsf{C}},\boldsymbol{e}_{\mathsf{C}})=\boldsymbol{a}_{\mathsf{C}}m+\boldsymbol{b}_{\mathsf{C}}r_{\mathsf{C}}+\boldsymbol{e}_{\mathsf{C}}$$

  The opening of the commitments is defined as $(m,r_{\mathsf{C}},\boldsymbol{e}_{\mathsf{C}},1)$.
- Ver: given $(\boldsymbol{c},m',r'_{\mathsf{C}},\boldsymbol{e}'_{\mathsf{C}},f')$ the verification algorithm accepts if and only if:

$$\boldsymbol{a}_{\mathsf{C}}m'+\boldsymbol{b}_{\mathsf{C}}r'_{\mathsf{C}}+f'^{-1}\boldsymbol{e}'_{\mathsf{C}}=\boldsymbol{c}\wedge\|\boldsymbol{e}'_{\mathsf{C}}\|_\infty\le\left\lfloor\frac{n^{4/3}}{2}\right\rfloor\wedge\|f'\|_\infty\le 1\wedge\deg f'\le\frac{n}{2}$$

This commitment scheme satisfies the security requirements of correctness, perfectly binding and computational hiding as they are explained in [7].

The main reason for us to choose this commitment scheme is that [7] gives efficient ZKPoK to prove knowledge of an opening of a given commitment or to prove that the messages inside some commitments satisfy any polynomial relation.

4 Proof of a shuffle for RLWE encryptions

The existing published proposal for a universally verifiable proof of a shuffle for RLWE encryptions [11] based on [33], uses Generalized Pedersen commitments to hide the secret re-randomization elements. This would not be sound in a post-quantum scenario, as it is based on DL assumptions.

Naively replacing the commitment scheme with the one proposed by Benhamouda *et al.* yields several difficulties since it is useful when committing to polynomials, but is quite inefficient if we only want to commit to a bit, as is the case with the entries of a permutation matrix. The fact that $\mathbb{Z}_q\left[x\right] / \left\langle x^n+1\right\rangle$ is not an integral domain also has some implications for the characterization of a permutation matrix proposed in [33], that cannot be proven directly and would require additional statements different from the ones discussed in [11].

In this section we construct a post-quantum verifiable mix-node following the paradigm given by Bayer and Groth in [6] (in appendix B we give some intuitions about their construction). Once again, replacing Pedersen commitments with the ones proposed by Benhamouda *et al.* is not immediate.

We first show an overview of the shuffling protocol, then we present our proof of a shuffle and give details regarding the ZKPoK involved in the construction of the main proof and finally we prove that our mix-node is secure based on a new formal definition of security, stronger than that given in [36].

Proofs of a shuffle commonly require universal verifiability, meaning that a proof must be generated and also published, so it can be verified by any observer. Classically, this kind of interactive protocols can be transformed into non-interactive protocols by means of the Fiat-Shamir heuristics, replacing the random responses from the verifier with a hash of the previous elements in the conversation, achieving a protocol secure in the Random Oracle Model (ROM).

However, as it is exposed in [35], this method is not secure anymore in the Quantum Random Oracle Model (QROM). As far as we know the only quantum secure general transformation from an interactive protocol to a non-interactive version is the one described by [34]. Therefore, a universally verifiable version of our protocol requires further considerations.

4.1 Protocol overview

Given a permutation π and a set of re-encryption parameters $\left\{r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\right\}$ for each one of the messages, the shuffling of N RLWE encryptions is defined as $\left(u'^{(i)}, v'^{(i)}\right) = \mathsf{Re\text{-}encrypt}\left(\left(u^{\pi(i)}, v^{\pi(i)}\right), r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\right)$.

A mix-node will perform the shuffling over the input ciphertexts and will generate a proof of a shuffle, see (1), to demonstrate that it knows the permutation π and the random elements $r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}$, without revealing any information about them.

This proof will be published so everybody is convinced that the ciphertexts have been permuted and re-encrypted without modifying the encrypted plaintexts (even if some of the nodes are dishonest and leak the permutation).

The first step of the protocol will be to commit to the encryptions of 0 used to compute the RLWE re-encryptions and a ZKPoK of the resulting commitments containing valid encryptions of 0. Additionally, it will also be demonstrated that the small polynomials $r'_{\mathsf{E}}, e'_{\mathsf{E},u}, e'_{\mathsf{E},v}$ used to compute the re-encryptions have an infinity norm that is bounded by some parameter $\delta \ll q/4$.

$$\text{ZKPoK}\left[\begin{array}{c|c} \pi & \left(u'^{(i)}, v'^{(i)}\right) = \\ \left\{r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\right\}_{i=1}^{N} & \mathsf{Re\text{-}encrypt}\left(\left(u^{\pi(i)}, v^{\pi(i)}\right), r'^{(i)}_{\mathsf{E}}, e'^{(1)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\right) \\ & \left\|r'^{(i)}_{\mathsf{E}}\right\|_{\infty}, \left\|e'^{(i)}_{\mathsf{E},u}\right\|_{\infty}, \left\|e'^{(i)}_{\mathsf{E},v}\right\|_{\infty} \leq \delta \end{array}\right] \tag{1}$$

As it is explained in [7] for a suitable δ even if this additional restriction on the re-encryption parameters norm is applied, the re-encryptions remain pseudorandom, as the two probability distributions are statistically close. The last part of the protocol consists on proving that two sets contain the same elements:

$$\left\{\left(u'^{(i)}, v'^{(i)}\right) - \left(a_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},u}, b_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},v}\right)\right\}_{i=1}^{N} = \left\{\left(u^{(i)}, v^{(i)}\right)\right\}_{i=1}^{N}$$

This is done following the strategy proposed by Bayer and Groth in [6], that consists on building two polynomials, each of them having as roots the elements of each of the sets and then prove that both polynomials are equal. To convince a verifier that two polynomials are equal the prover evaluates them in a random point chosen by the verifier and uses the generalized version of Schwartz-Zippel lemma (lemma 1). Our polynomials will be evaluated and have coefficients in R_q, that is, we will work in $R_q[A]$ and the variable A takes values on R_q.

We define the mixing protocol using the following algorithms:

- $\mathsf{Setup}(1^{\kappa})$: generate parameters (n, q, σ) and run the following algorithms:
  - $\mathsf{KeyGen}_{\mathsf{E}}(1^{\kappa})$ to obtain the public and the private key of the RLWE encryption scheme: $(a_{\mathsf{E}}, b_{\mathsf{E}}) \in R_q \times R_q$ and $s \in R_q$
  - $\mathsf{KeyGen}_{\mathsf{C}}(1^{\kappa})$ to generate the public commitment key: $\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}} \xleftarrow{\$} (R_q)^k$.

  Output $\{\{(a_{\mathsf{E}}, b_{\mathsf{E}}), s\}, (\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}})\}$
- $\mathsf{MixVotes}(pk_{\mathsf{E}}, pk_{\mathsf{C}}, \{(u^{(i)}, v^{(i)})\}_{i=1}^{N})$: taking as input a list of N encrypted messages $\{(u^{(i)}, v^{(i)})\}_{i=1}^{N}$ compute the shuffling of these RLWE encryptions. Generate commitments and ZKPoK (we denote by ZK_i its corresponding protocols and by Σ_i the proofs they output) as it is explained in section 4.2 in order to demonstrate the correctness of the process. We can explicitly state the permutation and/or random elements to be used writing $\mathsf{MixVotes}(pk_{\mathsf{E}}, pk_{\mathsf{C}}, \{(u^{(i)}, v^{(i)})\}_{i=1}^{N}; \pi, \{r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\}_{i=1}^{N})$.
  Output $\left(\{(u'^{(i)}, v'^{(i)})\}_{i=1}^{N}, \{(\boldsymbol{c}_{u_0^{(i)}}, \boldsymbol{c}_{v_0^{(i)}}, \boldsymbol{c}_{\pi(i)}, \boldsymbol{c}_{\alpha^{\pi(i)}})\}_{i=1}^{N}, \Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4\right)$.
  We denote $\Sigma_0 = \{\boldsymbol{c}_{u_0^{(i)}}, \boldsymbol{c}_{v_0^{(i)}}, \boldsymbol{c}_{\pi(i)}, \boldsymbol{c}_{\alpha^{\pi(i)}})\}_{i=1}^{N}$ to unify the notation of the output of $\mathsf{MixVotes}$.
- $\mathsf{VerifyMix}(pk_{\mathsf{E}}, pk_{\mathsf{C}}, \{(u^{(i)}, v^{(i)})\}_{i=1}^{N}, \{(u'^{(i)}, v'^{(i)})\}_{i=1}^{N}, \{\Sigma_l\}_{l=0}^{4})$: given an input and an output of the mixing process and the ZKPoK generated, this algorithm outputs 1 if the proofs are valid and 0 otherwise.

4.2 Proof of a shuffle

In this subsection we present the proposed proof (see protocol 1.1) and explain in detail how it can be used as a proof of a shuffle.

Notice that each mix-node runs the algorithm $\mathsf{MixVotes}$ and acts as a prover. He first commits to N encryptions of zero. Each commitment $(\boldsymbol{c}_{u_0^{(i)}}, \boldsymbol{c}_{v_0^{(i)}})$ is:

$$\left(\boldsymbol{a}_{\mathsf{C}}\left(a_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},u}\right) + \boldsymbol{b}_{\mathsf{C}} r^{(i)}_{\mathsf{C},u} + \boldsymbol{e}^{(i)}_{\mathsf{C},u}, \boldsymbol{a}_{\mathsf{C}}\left(b_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},v}\right) + \boldsymbol{b}_{\mathsf{C}} r^{(i)}_{\mathsf{C},v} + \boldsymbol{e}^{(i)}_{\mathsf{C},v}\right)$$

That is, the commitment is a linear combination of the polynomials, with the additional condition of $r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}, \boldsymbol{e}^{(i)}_{\mathsf{C},u}, \boldsymbol{e}^{(i)}_{\mathsf{C},v}$ having small norm.

Then, $\mathcal{P}$ sends the commitments to the verifier and proves using the amortized proof of knowledge of secret small elements [14] that the public commitments are indeed commitments to encryptions of zero.

As the relation is always the same we will use the amortized proposal by del Pino and Lyubashevsky [14], which is a direct improvement of the proposal by Cramer *et al.* [12]. For a linear function f, a small vector $\boldsymbol{x}$ and its image $\boldsymbol{y} = f(\boldsymbol{x})$ we can prove knowledge of a small vector $\boldsymbol{x'}$ such that $f(\boldsymbol{x'}) = \boldsymbol{y}$. As it is usual in this kind of proofs there is a gap τ between the upper bound of the norm we use for witness $\boldsymbol{x}$ and the upper bound we get for the extracted $\boldsymbol{x'}$. This has to be taken into account when determining specific parameters so that this possible error multiplied by the number of mix-nodes does not exceed the bounds allowed for a correct decryption. We refer the reader to [14] for details, as we directly use their protocol as a building block for the ZKPoK of linear relations in ZK_1 (protocol 1.1).

Using the amortization technique of [14] as a way of proving knowledge of valid openings for [7] has some benefits and some drawbacks. On the one hand this amortized technique allows us to prove the complex structure with an amortized cost. On the other hand the gap from the bound known by the prover and the bound he is able to prove is larger than the one originally established in the ZKPoK for valid commitment openings from [7].

As a result, the prover is only able to prove knowledge of some openings that would not be valid as originally defined. However, we can prove that, in our particular case, we can further relax this definition as the openings we obtain still ensure the binding property of the commitment scheme. Details of this and a rigorous parameter analysis can be found in appendix C.

In order to commit to a permutation, $\mathcal{P}$ starts committing to $\pi(1), \ldots, \pi(N)$ in $\boldsymbol{c}_{\pi(i)}$ and receives a polynomial α chosen uniformly at random from the subset:

$$S = \{p(x) \in R_q \mid \deg p(x) < n/2\}$$

Protocol 1.1. Proof of a shuffle

$$\mathcal{P}\left(u^{(i)}, v^{(i)}, u'^{(i)}, v'^{(i)}; \pi, r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\right) \qquad\qquad \mathcal{V}\left(u^{(i)}, v^{(i)}, u'^{(i)}, v'^{(i)}\right)$$

$\forall i \in [1, \dots, N]$

$\boldsymbol{c}_{u_0^{(i)}} = \mathsf{Com}\Big(a_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},u}\Big)$

$\boldsymbol{c}_{v_0^{(i)}} = \mathsf{Com}\Big(b_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},v}\Big)$

$$\xrightarrow{\boldsymbol{c}_{u_0^{(i)}},\, \boldsymbol{c}_{v_0^{(i)}}}$$

$$\mathrm{ZKPoK}\left[\begin{array}{c|l} \begin{array}{c} r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v} \\ r^{(i)}_{\mathsf{C},u}, \boldsymbol{e}^{(i)}_{\mathsf{C},u}, r^{(i)}_{\mathsf{C},v}, \boldsymbol{e}^{(i)}_{\mathsf{C},v} \end{array} & \begin{array}{l} \boldsymbol{c}_{u_0^{(i)}} = \boldsymbol{a}_{\mathsf{C}}\left(a_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},u}\right) + \boldsymbol{b}_{\mathsf{C}} r^{(i)}_{\mathsf{C},u} + \boldsymbol{e}^{(i)}_{\mathsf{C},u} \\ \boldsymbol{c}_{v_0^{(i)}} = \boldsymbol{a}_{\mathsf{C}}\left(b_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},v}\right) + \boldsymbol{b}_{\mathsf{C}} r^{(i)}_{\mathsf{C},v} + \boldsymbol{e}^{(i)}_{\mathsf{C},v} \\ \left\| r'^{(i)}_{\mathsf{E}} \right\|_{\infty}, \left\| e'^{(i)}_{\mathsf{E},*} \right\|_{\infty} \leq \tau\delta, \quad \left\| \boldsymbol{e}^{(i)}_{\mathsf{C},*} \right\|_{\infty} \leq \tau\delta' \end{array} \end{array}\right] \ (\mathrm{ZK}_1)$$

$\forall i \in [1, \dots, N]$

$\boldsymbol{c}_{\pi(i)} = \mathsf{Com}(\pi(i))$

$$\xrightarrow{\boldsymbol{c}_{\pi(i)}}$$

$$\alpha \xleftarrow{\$} S$$

$$\xleftarrow{\alpha}$$

$\forall i \in [1, \dots, N]$

$\boldsymbol{c}_{\alpha^{\pi(i)}} = \mathsf{Com}\Big(\alpha^{\pi(i)}\Big)$

$$\xrightarrow{\boldsymbol{c}_{\alpha^{\pi(i)}}}$$

$$\beta, \gamma \xleftarrow{\$} S$$

$$\xleftarrow{\beta, \gamma}$$

$$\mathrm{ZKPoK}\left[\begin{array}{c|c} \begin{array}{c} m_i, r_i, \boldsymbol{e}_{\mathsf{C},i}, f_i \\ \widehat{m}_i, \widehat{r}_i, \widehat{\boldsymbol{e}}_{\mathsf{C},i}, \widehat{f}_i \end{array} & \begin{array}{c} \left(\prod_{i=1}^{N}\Big(\beta i + \alpha^i - \gamma\Big) = \prod_{i=1}^{N}\left(\beta m_i + \widehat{m}_i - \gamma\right)\right), \\ \bigwedge_{i=1}^{N}\Big(\mathsf{Ver}(\boldsymbol{c}_{\pi(i)}; m_i, r_i, \boldsymbol{e}_{\mathsf{C},i}, f_i) = \mathsf{accept}\Big), \\ \bigwedge_{i=1}^{N}\Big(\mathsf{Ver}(\boldsymbol{c}_{\alpha^{\pi(i)}}; \widehat{m}_i, \widehat{r}_i, \widehat{\boldsymbol{e}}_{\mathsf{C},i}, \widehat{f}_i) = \mathsf{accept}\Big), \\ m_i \in \mathbb{Z}_q \end{array} \end{array}\right] \ (\mathrm{ZK}_2)$$

$$\mathrm{ZKPoK}\left[\begin{array}{c|l} \begin{array}{c} y \in \left\{\begin{smallmatrix} \alpha^{\pi(i)} \\ u_0^{(i)} \end{smallmatrix}\right\}_i \\ r_y \\ \boldsymbol{e}_{\mathsf{C},y} \\ f_y \end{array} & \begin{array}{l} \sum_{i=1}^{N} \alpha^i u^{(i)} = \sum_{i=1}^{N} m_{\alpha^{\pi(i)}}\left(u'^{(i)} - m_{u_0^{(i)}}\right) \\ \bigwedge_y \Big(\mathsf{Ver}(\boldsymbol{c}_y; m_y, r_y, \boldsymbol{e}_{\mathsf{C},y}, f_y) = \mathsf{accept}\Big) \end{array} \end{array}\right] \ (\mathrm{ZK}_3)$$

$$\mathrm{ZKPoK}\left[\begin{array}{c|l} \begin{array}{c} y \in \left\{\begin{smallmatrix} \alpha^{\pi(i)} \\ v_0^{(i)} \end{smallmatrix}\right\}_{i,j,l} \\ r_y \\ \boldsymbol{e}_{\mathsf{C},y} \\ f_y \end{array} & \begin{array}{l} \sum_{i=1}^{N} \alpha^i v^{(i)} = \sum_{i=1}^{N} m_{\alpha^{\pi(i)}}\left(v'^{(i)} - m_{v_0^{(i)}}\right) \\ \bigwedge_y \Big(\mathsf{Ver}(\boldsymbol{c}_y; m_y, r_y, \boldsymbol{e}_{\mathsf{C},y}, f_y) = \mathsf{accept}\Big) \end{array} \end{array}\right] \ (\mathrm{ZK}_4)$$

outputs accept if all
ZKPoK are correct

Observe that the subset S meets the required conditions for lemma 1, as all differences of two different elements in S are invertible. This is true as the condition $q \equiv 3 \mod 8$ required for the Benhamouda *et al.* commitment scheme implies that $x^n + 1$ splits into two irreducible polynomials of size exactly $n/2$. Then all polynomials of degree smaller that $n/2$ have an inverse that can be computed using the Chinese Remainder Theorem.

$\mathcal{P}$ commits to each power $\alpha^{\pi(i)}$ in commitments $\boldsymbol{c}_{\alpha^{\pi(i)}}$ and publishes them. After that, $\mathcal{P}$ receives two more random polynomials $\beta, \gamma \xleftarrow{\$} S$.

At this point $\mathcal{P}$ starts proving that he knows valid integer openings $m_i \in \mathbb{Z}_q, \widehat{m}_i \in R_q$ to commitments $\boldsymbol{c}_{\pi(i)}, \boldsymbol{c}_{\alpha^{\pi(i)}}$ that satisfy the following relation (ZK$_2$ in protocol 1.1):

$$\prod_{i=1}^{N} \left(\beta i + \alpha^i - \gamma\right) = \prod_{i=1}^{N} \left(\beta m_i + \widehat{m}_i - \gamma\right). \tag{2}$$

In order to prove that some of the messages are integers we will use again the amortized proposal by del Pino and Lyubashevsky [14]. This time the linear function we need to consider maps the message, randomness and error $(m_i, r_i, \boldsymbol{e}_{\mathsf{C},i})$ to the commitment $\boldsymbol{a}_{\mathsf{C}} m_i + \boldsymbol{b} r_i + \boldsymbol{e}_{\mathsf{C}}$. The only requirement for the mapping is to be linear, therefore we can define it by construction to take only integer m_i as inputs. Originally [14] was designed for proving knowledge of small preimages, however everything works the same way if we just require part of the preimage to be small. The small part of the secret will be hidden with gaussian noise as before, while the unbounded part will be hidden with uniformly random noise. The same parameter analysis that was done in appendix C for ZK$_1$ applies here.

In order to verify equation (2) we can use the Σ-protocols from [7] that allow proving polynomial relations between committed messages.

We can consider the two sides of equation (2) as polynomials in a variable Γ evaluated in a specific $\gamma \in R_q$ with coefficients in $\mathbb{Z}_q[x] / \langle x^n + 1 \rangle$. The prover has shown that they are equal when evaluated in this specific γ chosen by the verifier, but we would like them to be equal as polynomials in $R_q[\Gamma]$. The left hand side of the equation has been determined by the choices of the verifier, and in the right hand side, by the binding property of the commitment scheme, we know that $m_i, \widehat{m}_i$ were determined before the choice for γ was made.

We have already checked that subset S satisfies the conditions of the Generalized Schwartz-Zippel lemma 1. Using this lemma the verifier is convinced that with overwhelming probability the two polynomials defined by (2) are indeed equal in $R_q[\Gamma]$.

We would still have to prove that both sets of roots, $\left\{\beta i + \alpha^i\right\}_i$, $\left\{\beta m_i + \widehat{m}_i\right\}_i$, are equal. This is not direct in general as R_q is not a unique factorization domain (in particular it is not even a domain). However, in our particular case, both sets are going to be equal with overwhelming probability over the choice of β.

For each $j \in [1, \ldots, N]$, we are going to study whether $\beta j + \alpha^j$ belongs to $\left\{\beta m_i + \widehat{m}_i\right\}_i$. We know it is a root of the polynomial so $\prod_{i=1}^{N} (\beta m_i + \widehat{m}_i - (\beta j + \alpha^j)) = 0$.

As we stated before, choosing $q \equiv 3 \mod 8$ implies that $x^n + 1$ splits into two irreducible polynomials of degree $n/2$. We are going to call these polynomials p_1

and p_2 and consider operations modulo both of them. In particular $\prod_{i=1}^{N}(\beta m_i + \widehat{m}_i - (\beta j + \alpha^j)) \equiv 0 \mod p_1$ and $\prod_{i=1}^{N}(\beta m_i + \widehat{m}_i - (\beta j + \alpha^j)) \equiv 0 \mod p_2$.

Given that p_1 and p_2 are irreducible $\mathbb{Z}_q[x]/\langle p_1\rangle$ and $\mathbb{Z}_q[x]/\langle p_2\rangle$ are fields and it is possible to ensure that at least one of the factors has to be 0. Let i_{j1} and i_{j2} be the indexes such that $\beta m_{i_{j1}} + \widehat{m}_{i_{j1}} - (\beta j + \alpha^j) \equiv 0 \mod p_1$ and $\beta m_{i_{j2}} + \widehat{m}_{i_{j2}} - (\beta j + \alpha^j) \equiv 0 \mod p_2$.

Lets write it as affine equations on β:

$$\begin{aligned} (m_{i_{j1}} - j)\beta + (\widehat{m}_{i_{j1}} - \alpha^j) \equiv 0 \mod p_1 \\ (m_{i_{j2}} - j)\beta + (\widehat{m}_{i_{j2}} - \alpha^j) \equiv 0 \mod p_2 \end{aligned} \tag{3}$$

First of all we need to see that, since m_i and $\widehat{m}_i$ were committed before β was honestly chosen uniformly from S, it is very unlikely that for any triplet $i, j \in [1, \ldots, N],\ b \in \{1, 2\}$ we have $(m_i - j)\beta + (\widehat{m}_i - \alpha^j) \equiv 0 \mod p_b$ unless $(m_i - j) \equiv 0 \mod p_b$. As we are now working in a field $\mathbb{Z}_q[x]/\langle p_b\rangle$ having $(m_i - j) \not\equiv 0 \mod p_b$ implies there is only one possible β satisfying the equation for each triplet (i, j, b). Notice that as elements of S have degree smaller than $n/2$ determining $\beta \mod p_b$ also determines it in R_q. There are $2N^2$ possible $\beta_{ijb} \equiv (m_i - j)^{-1}(\alpha^j - \widehat{m}_i) \mod p_b$, but β is chosen uniformly at random from S, that has cardinal $q^{n/2}$ and therefore the probability of choosing one of these conflicting values is negligible.

Provided that previous proofs in ZK$_2$ ensure that $m_i \in \mathbb{Z}_q$ is a constant polynomial we have that $m_{i_{jb}} \equiv j \mod p_b$ implies $m_{i_{jb}} \equiv j \mod x^n + 1$. Since for each j we have $m_{i_{j1}} = m_{i_{j2}} = j$ this implies $i_{j1} = i_{j2}$ and we can directly call it i_j and write the equations $\mod x^n + 1$.

As a direct consequence we would also have $\widehat{m}_{i_j} = \alpha^j \mod x^n + 1$ via the Chinese Reminder Theorem.

Finally we can ensure that, with overwhelming probability over the choice of β both sets commit to the same elements. Notice we have seen only one set inclusion, but since both sets contain the same number of elements and $i_j \neq i_{j'}$ if $j \neq j'$ this is everything we need.

Let $\tilde{\pi}$ be the permutation such that $j = \tilde{\pi}(i_j)$. Then, with overwhelming probability, $m_i = \tilde{\pi}(i)$ and $\widehat{m}_i = \alpha^{\tilde{\pi}(i)}$ for every $i \in [1, \ldots, N]$.

We abuse notation and call $m_{\alpha^{\pi(i)}}$ to $\widehat{m}_i$, as it has to be $\alpha^{\pi(i)}$, but understanding it is indexed by i and not the evaluation $\pi(i)$ that is unknown to the verifier.

This means that $\boldsymbol{c}_{\alpha^{\pi(i)}}$ are indeed commitments to α with exponents from 1 to N permuted in an order that was fixed by $\boldsymbol{c}_{\pi(i)}$ before α was chosen.

Then we again need to prove polynomial relations between committed messages using the Σ-protocols from [7]. We get that the input and output of the mix-node hold the following relation (ZK$_3$ and ZK$_4$ in protocol 1.1).

$$\textstyle\sum_{i=1}^{N} \alpha^i u^{(i)} = \sum_{i=1}^{N} m_{\alpha^{\pi(i)}} \left(u'^{(i)} - a_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} - e'^{(i)}_{\mathsf{E},u} \right)$$

We already know that $m_{\alpha^{\pi(i)}} = \alpha^{\pi(i)}$ for a secret π and that the claimed small elements used for the re-encryption are in fact small.

$$\sum_{i=1}^{N} \alpha^i u^{(i)} = \sum_{i=1}^{N} \alpha^{\pi(i)} \left(u'^{(i)} - a_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} - e'^{(i)}_{\mathsf{E},u} \right)$$

Once again we can see them as polynomials in $R_q[A]$ with coefficients in R_q that are equal when evaluated in α.

Both polynomials were determined before α was picked up, so we can apply lemma 1 and conclude that with overwhelming probability they are equal as polynomials, and so:

$$u'^{(i)} = u^{\pi(i)} + a_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},u} \qquad\qquad v'^{(i)} = v^{\pi(i)} + b_{\mathsf{E}} r'^{(i)}_{\mathsf{E}} + e'^{(i)}_{\mathsf{E},v}$$

The verifier $\mathcal{V}$ can conclude that the mix-net has behaved properly and the output is a permuted re-encryption of the input. Completeness, zero-knowledge and soundness follow from this reasoning and are discussed in appendix D.

4.3 Security

Finally we propose a security definition and provide a proof of security for our proposed mix-node. Informally, a mix-node should ensure that it is not possible to link an input ciphertext with its corresponding output. However, there might be more than one ciphertext encrypting the same message (this is particularly the case in an election with many voters and only a few voting options), and we have to precisely say that it is not possible to link an input of the mix-node to an output encrypting the same message.

Some security definitions assume that the original messages are independently and uniformly distributed over the message space, but it was pointed out by Wikström in [36] that there might be known correlations between some of the input plaintexts that cannot be ignored.

We base our secure mix-node definition in the one presented by Wikström in [36], but we notice that he assumes that the inputs of the mix-node are correctly computed encryptions of the messages. However the input of each mix-node comes from the (possibly malicious) previous node, and while the proofs of a shuffle ensure that the input is a set of valid encryptions we do not know if the re-encryption parameters have been drawn randomly from the adequate distributions or specifically chosen by the possibly malicious previous nodes. Therefore we present a stronger definition where we even allow an adversary $\mathcal{A}$ to choose the messages and compute something of the form of an encryption, that is, a pair of polynomials in R_q, allowing him to completely determine the input of the mix-node. Even though, he should not be able to identify an input and output index corresponding to the same message with a probability significantly greater than a random guess. Let $\mathsf{MixVotes}$ be an algorithm that performs a shuffle and outputs a zero-knowledge proof Σ. Then we can define:
$\mathbf{Exp}^{sec}_{\mathcal{A}}(\kappa)$

- $(pk, sk) \leftarrow \mathsf{Setup}(1^\kappa)$
- $(z^{(1)}, \ldots, z^{(N)}, aux) \xleftarrow{\$} \mathcal{A}(pk)$

```
– for $k \in \{1, \ldots, N\}$
    $(u^{(k)}, v^{(k)}) \xleftarrow{\$} \mathcal{A}(pk, z^{(k)}, aux)$
  end for
– $\pi \xleftarrow{\$} \mathfrak{S}_N$
– $\left(\{(u'^{(k)}, v'^{(k)})\}_{k=1}^{N}, \Sigma\right) \leftarrow \mathsf{MixVotes}(pk, \{(u^{(k)}, v^{(k)})\}_{k=1}^{N}; \pi)$
– $(i_{\mathcal{A}}, j_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}(\{(u^{(k)}, v^{(k)})\}_{k=1}^{N}, \{(u'^{(k)}, v'^{(k)})\}_{k=1}^{N}, \Sigma, aux)$
– if $z^{(i_A)} = z^{\pi(j_A)}$ then Return 1 else Return 0
```

Now we can formalize our security definition saying that no adversary can have a significant advantage over a random guess.

Definition 2 (Secure Mix-Node). *Let J be a uniform random variable taking values in $[1, \ldots, N]$. We say that a mix-node defined by an algorithm $\mathsf{MixVotes}$ is* secure *if the advantage of any PPT adversary $\mathcal{A}$ over a random guess is negligible in the security parameter. That is, for all c there exists a κ_0 such that if $\kappa \geq \kappa_0$:*

$$\begin{aligned}\boldsymbol{Adv}_{\mathcal{A}}^{sec}(\kappa) &= \left|\Pr\left[z^{(i_A)} = z^{\pi(j_A)}\right] - \Pr\left[z^{(i_A)} = z^{\pi(J)}\right]\right| \\ &= \left|\Pr\left[\boldsymbol{Exp}_{\mathcal{A}}^{sec}(\kappa) = 1\right] - \Pr\left[z^{(i_A)} = z^{\pi(J)}\right]\right| < \frac{1}{\kappa^c}\end{aligned}$$

We allow the adversary to corrupt all mix-nodes except one, and the non-corrupted one is that considered in the experiment $\mathbf{Exp}_{\mathcal{A}}^{sec}$. In order to take into account any possible control of the adversary over those other corrupted nodes and possibly a subset of the voters we even allow him to fully control all the input of the mix-node. Even though, if at least one of the mix-nodes is honest, the link between the ciphertexts at the output and those at the input of the mix-net remains completely hidden.

Observe that this security definition has to be complemented with additional security proofs when this mix-node is used as a building block in a larger scheme. For instance Wikström in [36] shows how a malleable cryptosystem can be used to break anonymity. Therefore additional validity proofs are required to enforce non-malleability, as well as strict decryption policies to prevent any leakage of information during the decryption phase.

Theorem 1. *The proposed mix-node given by our $\mathsf{MixVotes}$ algorithm is a secure mix-node according to definition 2, under the RLWE hardness assumption.*

The proof of theorem 1 is given in appendix E.

5 Conclusions

We present a shuffle that consists of a permutation and re-encryption of a set of RLWE ciphertexts. The lattice-based encryption scheme used is that proposed by Lyubashevsky *et al.* and we provide a proof of correctness of the shuffle using a lattice-based commitment scheme proposed by Benhamouda *et al.* Furthermore we give a security definition and we prove that our shuffle satisfies it.

As future work it would be worthy to have an implementation with concrete parameters in order to accurately test efficiency in a real setting. We also remark that this shuffle has to be combined with additional security requirements regarding how the input is generated as well as how the output is decrypted, in order to guarantee privacy for the overall scheme that uses this shuffle as a building block, and these requirements will depend on the specific application.

Acknowledgements. We would like to thank Kristian Gjøsteen for his helpful comments that greatly improved the proposal.

This work is partially supported by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701) and the Spanish Ministry of Economy and Competitiveness, under Project MTM2016-77213-R.

References

1. Abe, M.: Universally verifiable mix-net with verification work independent of the number of mix-servers. In: K. Nyberg (ed.) EUROCRYPT'98, *LNCS*, vol. 1403, pp. 437–447. Springer, Heidelberg, Germany, Espoo, Finland (1998). doi:10.1007/BFb0054144
2. Abe, M.: Mix-networks on permutation networks. In: K.Y. Lam, E. Okamoto, C. Xing (eds.) ASIACRYPT'99, *LNCS*, vol. 1716, pp. 258–273. Springer, Heidelberg, Germany, Singapore (1999). doi:10.1007/978-3-540-48000-6_21
3. Abe, M., Hoshino, F.: Remarks on mix-network based on permutation networks. In: K. Kim (ed.) PKC 2001, *LNCS*, vol. 1992, pp. 317–324. Springer, Heidelberg, Germany, Cheju Island, South Korea (2001). doi:10.1007/3-540-44586-2_23
4. Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., Peikert, C.: More efficient commitments from structured lattice assumptions. In: D. Catalano, R. De Prisco (eds.) SCN 18, *LNCS*, vol. 11035, pp. 368–385. Springer, Heidelberg, Germany, Amalfi, Italy (2018). doi:10.1007/978-3-319-98113-0_20
5. Baum, C., Lyubashevsky, V.: Simple amortized proofs of shortness for linear relations over polynomial rings. Cryptology ePrint Archive, Report 2017/759 (2017). `http://eprint.iacr.org/2017/759`
6. Bayer, S., Groth, J.: Zero-knowledge argument for polynomial evaluation with application to blacklists. In: T. Johansson, P.Q. Nguyen (eds.) EUROCRYPT 2013, *LNCS*, vol. 7881, pp. 646–663. Springer, Heidelberg, Germany, Athens, Greece (2013). doi:10.1007/978-3-642-38348-9_38
7. Benhamouda, F., Krenn, S., Lyubashevsky, V., Pietrzak, K.: Efficient zero-knowledge proofs for commitments from learning with errors over rings. In: G. Pernul, P.Y.A. Ryan, E.R. Weippl (eds.) ESORICS 2015, Part I, *LNCS*, vol. 9326, pp. 305–325. Springer, Heidelberg, Germany, Vienna, Austria (2015). doi:10.1007/978-3-319-24174-6_16
8. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy, pp. 315–334. IEEE Computer Society Press, San Francisco, CA, USA (2018). doi:10.1109/SP.2018.00020
9. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–90 (1981)

10. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: A homomorphic LWE based E-voting scheme. In: T. Takagi (ed.) Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, pp. 245–265. Springer, Heidelberg, Germany, Fukuoka, Japan (2016). doi:10.1007/978-3-319-29360-8_16
11. Costa, N., Martínez, R., Morillo, P.: Proof of a shuffle for lattice-based cryptography. In: Nordic Conf. on Secure IT Systems, pp. 280–296. Springer (2017)
12. Cramer, R., Damgård, I., Xing, C., Yuan, C.: Amortized complexity of zero-knowledge proofs revisited: Achieving linear soundness slack. In: J. Coron, J.B. Nielsen (eds.) EUROCRYPT 2017, Part I, *LNCS*, vol. 10210, pp. 479–500. Springer, Heidelberg, Germany, Paris, France (2017). doi:10.1007/978-3-319-56620-7_17
13. Damgard, I.: On σ-protocols. Lecture on Cryptologic Protocol Theory; Faculty of Science, University of Aarhus (2010)
14. del Pino, R., Lyubashevsky, V.: Amortization with fewer equations for proving knowledge of small secrets. In: J. Katz, H. Shacham (eds.) CRYPTO 2017, Part III, *LNCS*, vol. 10403, pp. 365–394. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (2017). doi:10.1007/978-3-319-63697-9_13
15. del Pino, R., Lyubashevsky, V., Neven, G., Seiler, G.: Practical quantum-safe voting from lattices. In: B.M. Thuraisingham, D. Evans, T. Malkin, D. Xu (eds.) ACM CCS 2017, pp. 1565–1581. ACM Press, Dallas, TX, USA (2017). doi:10.1145/3133956.3134101
16. Furukawa, J.: Efficient and verifiable shuffling and shuffle-decryption **88-A**, 172–188 (2005)
17. Furukawa, J., Sako, K.: An efficient scheme for proving a shuffle. In: J. Kilian (ed.) CRYPTO 2001, *LNCS*, vol. 2139, pp. 368–387. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (2001). doi:10.1007/3-540-44647-8_22
18. Groth, J.: A verifiable secret shuffle of homomorphic encryptions. In: Y. Desmedt (ed.) PKC 2003, *LNCS*, vol. 2567, pp. 145–160. Springer, Heidelberg, Germany, Miami, FL, USA (2003). doi:10.1007/3-540-36288-6_11
19. Groth, J., Ishai, Y.: Sub-linear zero-knowledge argument for correctness of a shuffle. In: N.P. Smart (ed.) EUROCRYPT 2008, *LNCS*, vol. 4965, pp. 379–396. Springer, Heidelberg, Germany, Istanbul, Turkey (2008). doi:10.1007/978-3-540-78967-3_22
20. Groth, J., Lu, S.: Verifiable shuffle of large size ciphertexts. In: T. Okamoto, X. Wang (eds.) PKC 2007, *LNCS*, vol. 4450, pp. 377–392. Springer, Heidelberg, Germany, Beijing, China (2007). doi:10.1007/978-3-540-71677-8_25
21. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: H. Gilbert (ed.) EUROCRYPT 2010, *LNCS*, vol. 6110, pp. 1–23. Springer, Heidelberg, Germany, French Riviera (2010). doi:10.1007/978-3-642-13190-5_1
22. Markus, J., Ari, J.: Millimix: Mixing in small batches. Tech. rep. (1999). Center for Discrete Mathematics; Theoretical Computer Science
23. Micciancio, D., Regev, O.: Lattice-based cryptography. In: D.J. Bernstein, J. Buchmann, E. Dahmen (eds.) Post-Quantum Cryptography, pp. 147–191. Springer-Verlag, Berlin, Heidelberg (2009)
24. Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: M.K. Reiter, P. Samarati (eds.) ACM CCS 2001, pp. 116–125. ACM Press, Philadelphia, PA, USA (2001). doi:10.1145/501983.502000
25. Neff, C.A.: Verifiable mixing (shuffling) of ElGamal pairs. VoteHere, Inc. (2003)
26. Peikert, C.: A decade of lattice cryptography. Cryptology ePrint Archive, Report 2015/939 (2015). `http://eprint.iacr.org/2015/939`

27. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: H.N. Gabow, R. Fagin (eds.) 37th ACM STOC, pp. 84–93. ACM Press, Baltimore, MA, USA (2005). doi:10.1145/1060590.1060603
28. Sako, K., Kilian, J.: Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In: L.C. Guillou, J.J. Quisquater (eds.) EUROCRYPT'95, *LNCS*, vol. 921, pp. 393–403. Springer, Heidelberg, Germany, Saint-Malo, France (1995). doi:10.1007/3-540-49264-X_32
29. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
30. Singh, K., Pandu Rangan, C., Banerjee, A.K.: Lattice based mix network for location privacy in mobile system. Mobile Information Systems **2015**, 1–9 (2015)
31. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: M. Matsui (ed.) ASIACRYPT 2009, *LNCS*, vol. 5912, pp. 617–635. Springer, Heidelberg, Germany, Tokyo, Japan (2009). doi:10.1007/978-3-642-10366-7_36
32. Strand, M.: A verifiable shuffle for the GSW cryptosystem. In: A. Zohar, I. Eyal, V. Teague, J. Clark, A. Bracciali, F. Pintore, M. Sala (eds.) FC 2018 Workshops, *LNCS*, vol. 10958, pp. 165–180. Springer, Heidelberg, Germany, Nieuwpoort, Curaçao (2019). doi:10.1007/978-3-662-58820-8_12
33. Terelius, B., Wikström, D.: Proofs of restricted shuffles. In: D.J. Bernstein, T. Lange (eds.) AFRICACRYPT 10, *LNCS*, vol. 6055, pp. 100–113. Springer, Heidelberg, Germany, Stellenbosch, South Africa (2010)
34. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: E. Oswald, M. Fischlin (eds.) EUROCRYPT 2015, Part II, *LNCS*, vol. 9057, pp. 755–784. Springer, Heidelberg, Germany, Sofia, Bulgaria (2015). doi:10.1007/978-3-662-46803-6_25
35. Unruh, D.: Post-quantum security of Fiat-Shamir. In: T. Takagi, T. Peyrin (eds.) ASIACRYPT 2017, Part I, *LNCS*, vol. 10624, pp. 65–95. Springer, Heidelberg, Germany, Hong Kong, China (2017). doi:10.1007/978-3-319-70694-8_3
36. Wikström, D.: The security of a mix-center based on a semantically secure cryptosystem. In: A. Menezes, P. Sarkar (eds.) INDOCRYPT 2002, *LNCS*, vol. 2551, pp. 368–381. Springer, Heidelberg, Germany, Hyderabad, India (2002)
37. Wikström, D.: A commitment-consistent proof of a shuffle. In: C. Boyd, J.M.G. Nieto (eds.) ACISP 09, *LNCS*, vol. 5594, pp. 407–421. Springer, Heidelberg, Germany, Brisbane, Australia (2009)

A Proof of Generalized Schwartz–Zippel lemma

Lemma 1 (Generalized Schwartz-Zippel lemma).

Let $p \in R\,[x_1, x_2, \ldots, x_n]$ be a non-zero polynomial of total degree $d \geq 0$ over a commutative ring R. Let S be a finite subset of R such that none of the differences between two elements of S is a divisor of 0 *and let $r_1, r_2, \ldots, r_n$ be selected at random independently and uniformly from S.*

Then $\mathsf{Pr}\,[p(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|}$.

Proof. The condition imposed on S implies that a degree d univariate non-zero polynomial $f \in R[x]$ can only have d roots in S. We can prove this by induction.

Case $d = 0$ is trivially true.

Assume the inequality holds for polynomials of degree smaller or equal to d and let $f(x)$ be a polynomial of degree $d + 1$ with $d + 2$ different roots $a_1, a_2, \ldots, a_{d+2} \in S$. The polynomial reminder theorem implies that we can write $f(x) = (x - a_{d+2})g(x)$ for some polynomial $g(x)$ of degree d.

In a field, $a_1, a_2, \ldots, a_{d+1}$ being roots of $f(x)$ and not of $(x - a_{d+2})$ would imply that they are roots of $g(x)$. But we are working with a polynomial ring, that may not be an integral domain, and this may not always be true.

However as $a_1, \ldots, a_{d+2}$ belong to S we know that $(a_i - a_{d+2})$ is not a divisor of 0 and we can ensure that $g(a_i)$ has to be 0 for $a_1, \ldots, a_{d+1}$.

Then $g(x)$ would have $d + 1$ different roots in S, this time contradicting the induction hypothesis and proving the result for the univariate case.

In order to prove the multivariate case we can follow the standard proof of the Schwartz–Zippel lemma, by induction on n.

Case $n = 1$ is the univariate case that we have already proved.

Assume the lemma is true for polynomials of n or less variables. We can write an $(n + 1)$-variate polynomial as:

$$f(x_1, \ldots, x_{n+1}) = \sum_{i \leq d'} x_{n+1}^i f_i(x_1, \ldots, x_n)$$

Where $f_{d'}$ is non-zero. As an n-variate polynomial, by the induction hypothesis, we have:

$$\mathsf{Pr}\Big[f_{d'}(x_1, \ldots, x_n) = 0\Big] \leq \frac{\deg(f_{d'})}{|S|}$$

If $f_{d'}(a_1, \ldots, a_n) \neq 0$, by the base case of the induction hypothesis we have:

$$\mathsf{Pr}\Big[f(a_1, \ldots, a_n, x_{n+1}) = 0\Big] \leq \frac{d'}{|S|}$$

And finally:

$$\begin{aligned}\mathsf{Pr}\Big[f(a_1,\dots,a_n,a_{n+1})=0\Big] &= \mathsf{Pr}\Big[f(a_1,\dots,a_n,a_{n+1})=0 \wedge f_{d'}(a_1,\dots,a_n)=0\Big]\\ &\quad+\mathsf{Pr}\Big[f(a_1,\dots,a_n,a_{n+1})=0 \wedge f_{d'}(a_1,\dots,a_n)\neq 0\Big]\\ &\leq \mathsf{Pr}\Big[f_{d'}(a_1,\dots,a_n)=0\Big]\\ &\quad+\mathsf{Pr}\Big[f(a_1,\dots,a_n,a_{n+1})=0\Big|f_{d'}(a_1,\dots,a_n)\neq 0\Big]\\ &\leq \frac{\deg(f_{d'})}{|S|}+\frac{d'}{|S|}\\ &\leq \frac{d}{|S|}\end{aligned}$$

□

Now we need to define a suitable subset $S \subseteq \mathbb{Z}_q[x]/\langle x^n+1\rangle$ for which the condition holds.

We can guarantee it if all differences of elements in S are invertible. We choose:

$$S=\Big\{p(x)\in\mathbb{Z}_q[x]/\langle x^n+1\rangle\ \Big|\ \deg p(x)<n/2\Big\}$$

Observe that the proposed subset S meets the required condition for lemma 1, as all differences of two polynomials in S are invertible. This is true as the condition $q \equiv 3 \mod 8$ implies that x^n+1 splits into two irreducible polynomials of degree exactly $n/2$ (lemma 3 in [31]). Then all polynomials of degree smaller that $n/2$ have an inverse that can be computed using the Chinese Remainder Theorem. The number of elements in S is still exponential in n, so we can use it as a set of challenges.

B Technique: Bayer and Groth Shuffle Argument

As mentioned in the introduction, our proposal is based on the shuffle argument given by Bayer and Groth in [6]. Although is not a direct adaptation of it, we want to give some intuitions here in order to better understand our construction presented in section 4.

The general idea of the shuffle argument is to demonstrate knowledge of a permutation π and some re-encryption parameters $\{\rho_i\}_{i=1}^N$ such that the set of ciphertexts at the output of the shuffle $\{C'_i\}_{i=1}^N$ are those at the input $\{C_i\}_{i=1}^N$ permuted and re-encrypted using the equation $C'_i = C_{\pi(i)}\mathsf{Encrypt}_{pk}(1;\rho_i)$. In order to construct the proof Bayer and Groth use the combination of two arguments: the multi-exponentiation ($\Sigma_{\text{multi-exp}}$) and the product argument ($\Sigma_{\text{prod-arg}}$). We are not going to enter into details about them since they are specific to ElGamal encryption and Pedersen commitment.

The proof can be divided in several steps (full proof is shown in protocol 1.2):

- The prover computes the permutation of the indexed set of elements $\{1, \ldots, N\}$: $\boldsymbol{a} = \{\pi(i)\}_{i=1}^{N}$. It also computes the commitment to this indexed set of values: $\boldsymbol{c_A}$.
- The verifier sends a challenge x and the prover computes $\boldsymbol{b} = \{x^{\pi(i)}\}_{i=1}^{N}$. Again, these values are committed: $\boldsymbol{c_B}$.
- It is demonstrated that the permutation used to compute $\boldsymbol{a}$ and $\boldsymbol{b}$ is the same, meaning that the prover has a commitment to $\{x^1, \ldots, x^N\}$ permuted in an order that was fixed before receiving x. This demonstration is done in the following way: the verifier sends two values y and z and the prover builds two polynomials, one using $y, z, \boldsymbol{a}$ and $\boldsymbol{b}$ and the second one with $y, z, \{1, \ldots, N\}$ and $\{x^i\}_{i=1}^{N}$. He proves, using the product argument, that both polynomials are equal, they have the same roots but in permuted order.
- Finally the prover demonstrates using the multi-exponentiation argument that he knows the re-encryption parameters such that

  $$\prod_{i=1}^{N} C_i^{x^i} = \mathsf{Encrypt}_{pk}(1; \rho) \prod_{i=1}^{N} (C_i')^{x^{\pi(i)}}$$

  where $\rho = -\boldsymbol{\rho} \cdot \boldsymbol{b}$. Given the homomorphic properties of the encryption scheme, the verifier can deduce from the above equation

  $$\prod_{i=1}^{N} M_i^{x^i} = \prod_{i=1}^{N} (M_i')^{x^{\pi(i)}}$$

  and taking discrete logarithms we have

  $$\sum_{i=1}^{N} \log(M_i) x^i = \sum_{i=1}^{N} \log(M'_{\pi^{-1}(i)}) x^i.$$

  As it is argued in [6], there is negligible probability over the choice of x that this equality holds true unless $M_1' = M_{\pi(1)}, \ldots, M_N' = M_{\pi(N)}$.
- The verifier accepts if the product and the multi-exponentiation arguments are both valid.

Protocol 1.2. Shuffle argument

$\mathcal{P}\,(pk, ck, \boldsymbol{C}, \boldsymbol{C}'; \pi, \boldsymbol{\rho})$ $\qquad\qquad$ $\mathcal{V}\,(pk, ck, \boldsymbol{C}, \boldsymbol{C}')$

$\boldsymbol{r} \overset{\$}{\leftarrow} \mathbb{Z}_q^m$

$\boldsymbol{a} = \{\pi(i)\}_{i=1}^N$

$\boldsymbol{c_A} = \mathsf{Com}_{ck}(\boldsymbol{a}; \boldsymbol{r})$

$$\xrightarrow{\boldsymbol{c_A}}$$

$x \overset{\$}{\leftarrow} \mathbb{Z}_q^*$

$$\xleftarrow{x}$$

$\boldsymbol{s} \in \mathbb{Z}_q^m$

$\boldsymbol{b} = \{x^{\pi(i)}\}_{i=1}^N$

$\boldsymbol{c_B} = \mathsf{Com}_{ck}(\boldsymbol{b}; \boldsymbol{s})$

$$\xrightarrow{\boldsymbol{c_B}}$$

$y, z \overset{\$}{\leftarrow} \mathbb{Z}_q^*$

$$\xleftarrow{y, z}$$

$\boldsymbol{c}_{-z} = \mathsf{Com}_{ck}(-z, \dots, -z; \boldsymbol{0})$

$\boldsymbol{c_D} = \boldsymbol{c_A}^y \boldsymbol{c_B}$

$\boldsymbol{d} = y\boldsymbol{a} + \boldsymbol{b}$

$\boldsymbol{t} = y\boldsymbol{r} + \boldsymbol{s}$

$\boldsymbol{c_D c_{-z}} = \mathsf{Com}_{ck}(\boldsymbol{d} - \boldsymbol{z}; \boldsymbol{t})$

$\prod_{i=1}^N (d_i - z) = \prod_{i=1}^N (yi + x^i - z)$

$\rho = -\boldsymbol{\rho} \cdot \boldsymbol{b}$

$\boldsymbol{x} = \left(x^1, x^2, \dots, x^N\right)$

$\boldsymbol{C^x} = \mathsf{Encrypt}_{pk}(1; \rho)\boldsymbol{C'^b}$

$$\xrightarrow{\Sigma_{\text{multi-exp}},\, \Sigma_{\text{prod-arg}}}$$

outputs accept if all
ZKPoK are correct

C Amortized Commitment Proof Analysis

Del Pino and Lyubashevsky show in [14] how to prove knowledge of small screts with an amortized cost. In order to do so their proof consists of two steps, an imperfect proof of knowledge, where the prover is able to prove knowledge of $N - \tau(\lambda)$ out of N secrets, and a compiler (adapted from [12]), used to transform an imperfect proof of knowledge into a regular proof of knowledge. The function $\tau(\lambda)$ defined for a security parameter λ is called imperfection.

Their initial imperfect proof has a soundness slack that depends on a parameter r and an imperfection $\tau(\lambda) = \frac{\lambda}{\log \alpha} + 1$. This r has to be an integer greater or equal than 128 and α is another parameter that controls the minimal amount of samples required for amortization. They provide an example that suits our demands, for $\alpha = 2^{10}$ one can create amortized proofs for as few as 853 secrets with a security parameter $\lambda = 128$. The compilation step adds extra soundness gap, and as a result [14] claims that the final ZKPoK for a secret bounded by β has a slack of $4\sqrt{r}\lambda\beta/\log\alpha$ for a security parameter λ.

In our case we use n as a security parameter and consider the error term of the commitment scheme also bounded by n. Using this kind of amortized proofs we would be able to prove that the error is bounded by $4\sqrt{128}n\left(\frac{n}{10}\right)$. This is greater than $n^{4/3}/2$, as required by the definition of a valid opening. However no invertible f is involved, and we can just redo the original binding proof and show how, for a suitable set of parameters, with overwhelming probability over the choice of the commitment public key, if a valid commitment exists and a prover uses this particular amortized proof to prove knowledge of another opening, then the message cannot be a different one. This binding property is what is required for the soundness of our protocol.

Lemma 2 (Extended binding property). *Let* $(m', r', \boldsymbol{e}', f')$*,* $(m'', r'', \boldsymbol{e}'', 1)$ *be such that* $\boldsymbol{c} = \boldsymbol{a}_{\mathsf{C}} m' + \boldsymbol{b}_{\mathsf{C}} r' + f'^{-1}\boldsymbol{e}' = \boldsymbol{a}_{\mathsf{C}} m'' + \boldsymbol{b}_{\mathsf{C}} r'' + \boldsymbol{e}''$ *where* $\|\boldsymbol{e}'\|_\infty \leq \lfloor n^{4/3}/2 \rfloor$*,* $\|f'\|_\infty \leq 1$*,* $\deg f' < n/2$ *and* $\|\boldsymbol{e}''\|_\infty \leq 4\sqrt{128}n\left(\frac{n}{10}\right)$*. Then, provided that parameters are chosen appropriately, with overwhelming probability over the choice of* $\boldsymbol{a}_{\mathsf{C}}$ *and* $\boldsymbol{b}_{\mathsf{C}}$*, we have* $m' = m''$*.*

Proof. Our goal is to find conditions on k and γ (defined as in [7], k is the dimension of $\boldsymbol{a}_{\mathsf{C}}$ and γ is such that $q \geq n^\gamma$) such that this lemma holds.

Assume by contradiction that $m' \neq m''$. Subtracting the two different expressions for $\boldsymbol{c}$ we get $\boldsymbol{a}_{\mathsf{C}} m + \boldsymbol{b}_{\mathsf{C}} r = f'^{-1}\boldsymbol{e}' - \boldsymbol{e}''$, for some $m, r \in R_q$ with $m \neq 0$. Lets fix these values $m, r, f', \boldsymbol{e}', \boldsymbol{e}''$ and check that the chances of this being possible are negligible.

Here we use again the fact that, since $q \equiv 3 \mod 8$, $x^n + 1$ splits into two irreducible polynomials p_1 and p_2 of degree $n/2$. As $m \neq 0$ we have $m \neq 0 \mod p_b$ at least for one $b \in \{1, 2\}$. Considering all possible $a_i \in R_q$ we have that $a_i m$ takes all $q^{n/2}$ possible equivalence classes $\mod p_b$ with uniform probability. This is independent for every i, as a result only a fraction $\frac{1}{q^{kn/2}}$ of all possible $(\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}})$ would satisfy the required equation.

Now, as we started fixing $m, r, f', \boldsymbol{e}', \boldsymbol{e}''$ we have to apply a union bound for all their possible values. That is q^n for m, q^n for r, $3^{n/2}$ for f', $(n^{4/3})^{kn}$ for $\boldsymbol{e}'$ and $\left(8\sqrt{128}n\left(\frac{n}{10}\right)\right)^{kn}$ for $\boldsymbol{e}''$.

If this union bound is negligible then with overwhelming probability over the choice of $(\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}})$ there are no $m, r, f', \boldsymbol{e}', \boldsymbol{e}''$ satisfiying the equation with $m \neq 0$. It would imply that m has to be 0, and the commitment would be binding even when considering this relaxed opening verifications that come from the amortized proofs.

The only missing step is to check when the following quantity is negligible:

$$\begin{aligned}
&\frac{q^{2n}3^{n/2}(n^{4/3})^{kn}\left(8n\sqrt{128}\frac{n}{10}\right)^{kn}}{q^{kn/2}} \\
&= \left(q^{2-k/2}3^{1/2}(n^{4/3})^{k}\left(\frac{2^{11/2}n^2}{5}\right)^{k}\right)^{n}
\end{aligned}$$

We know $k > 6$ from [7], then $2 - k/2 < 0$ and we can use $q \geq n^\gamma$ as defined in [7]:

$$\begin{aligned}
&\leq \left(n^{2\gamma-k\gamma/2}3^{1/2}(n^{4/3})^{k}\left(\frac{2^{11/2}n^2}{5}\right)^{k}\right)^{n} \\
&= \left(n^{2\gamma+k(10/3-\gamma/2)}3^{1/2}\left(\frac{2^{11/2}}{5}\right)^{k}\right)^{n} \\
&= \left(n^{2\gamma+k(10/3-\gamma/2+\log(2^{11/2}/5)/\log(n))}3^{1/2}\right)^{n}
\end{aligned}$$

And we want to impose that this quantity is negligible, that is:

$$\leq \left(\frac{1}{2}\right)^{n}$$

This is equivalent to:

$$\frac{1}{\sqrt{12}} \geq n^{2\gamma+k(10/3-\gamma/2+\log(2^{11/2}/5)/\log(n))}$$

And taking logarithms:

$$\begin{aligned}
\frac{\log\left(1/\sqrt{12}\right)}{\log(n)} &\geq 2\gamma + k(10/3 - \gamma/2 + \log(2^{11/2}/5)/\log(n)) \\
0 &\geq 2\gamma + k(10/3 - \gamma/2 + \log(2^{11/2}/5)/\log(n)) + \frac{\log(12)}{2\log(n)}
\end{aligned}$$

Notice how the contribution of the $\frac{1}{\log(n)}$ terms is positive. Therefore if the inequality is satisfied for some n_0 it would also be satisfied for any $n \geq n_0$. Therefore we can just plug in here the minimum value we want to consider for n, in this case $n = 2^9$ to achieve minimal security for the commitment scheme:

$$0 \geq 2\gamma + k\left(\frac{71 - 2\log(5) - 9\gamma}{18}\right) + \frac{\log(12)}{18}$$

Following the same reasoning, and using again $2 - k/2 < 0$ we notice that whenever this condition is satisfied for one γ_0 it will also be satisfied for any other $\gamma \geq \gamma_0$.

In order for the inequality to hold the coefficient of k, $\frac{71-2\log(5)-9\gamma}{18}$, has to be negative. This imposes $\gamma \geq 8$, and once we have this condition if the inequality holds for a given k_0 it will also be satisfied for any other $k \geq k_0$.

Summarizing, we just need to find the minimal pairs of $(k_0, \gamma_0) \in \mathbb{Z}^2$ satisfying the following three conditions, and that would imply that any pair (k, γ) with $k \geq k_0$ and $\gamma \geq \gamma_0$ would be feasible too.

- $\gamma \geq 8$
- $k > \frac{18\gamma}{3\gamma-16}$
- $0 \geq 2\gamma + k\left(\frac{71-2\log(5)-9\gamma}{18}\right) + \frac{\log(12)}{18}$

The region of feasible parameters can be found in figure 1. As long as we choose our parameters inside the green area the probability of the existence of non-zero solutions would be negligible and the commitment scheme will have the required extended binding property.

□

D Completeness, Zero-Knowledge and Soundness

If the prover $\mathcal{P}$ chooses all re-encryption parameters from the appropriate distribution χ conditioned to have norm smaller than δ, correctly builds the commitments to the encryptions of 0 and follows the small secrets proof the answer will be accepted. This is also the case for the proof of the committed permuted powers of α, as products $\prod_{i=1}^{N}\left(\beta i + \alpha^i - \gamma\right)$ and $\prod_{i=1}^{N}\left(\beta m_i + \widehat{m}_i - \gamma\right)$ are exactly equal, just in permuted order. Finally the two last ZKPoK are accepted as the output is exactly a permutation and re-encryption of the input, and we have built a polynomial subtracting the re-encryptions and inverting the permutation. To summarize, the protocol is complete as all the ZKPoK involved are accepted if an honest prover follows the protocols.

The special HVZK property is achieved as the only published elements are commitments (with a computationally hiding property based on the hardness of RLWE) and outputs of lattice-based ZK-protocols (that can be simulated and therefore leak no information).

Soundness follows with overwhelming probability from the soundness properties of the ZK-protocols for the commitments and the small elements, the binding property of the commitment scheme and also from the generalized Schwartz-Zippel lemma.

We start with ZK$_1$, if δ' is such that $\tau\delta' \leq \left\lfloor \frac{n^{4/3}}{2} \right\rceil$ the extractor of this zero-knowledge proof given by Del Pino and Lyubashevsky provides us with valid openings of $\boldsymbol{c}_{u_0^{(i)}}$ and $\boldsymbol{c}_{v_0^{(i)}}$ to a valid encryption of 0.

Then, we analyze ZK$_2$, using the extractor of Benhamouda *et al.* we obtain valid openings for $\boldsymbol{c}_{\pi(i)}$ and $\boldsymbol{c}_{\alpha^{\pi(i)}}$ that satisfy the equation $\prod_{i=1}^{N}\left(\beta i + \alpha^i - \gamma\right) = \prod_{i=1}^{N}\left(\beta m_i + \widehat{m}_i - \gamma\right)$. The order in which all polynomials have been determined, generalized Schwartz-Zippel and the previously discussed argument guarantees

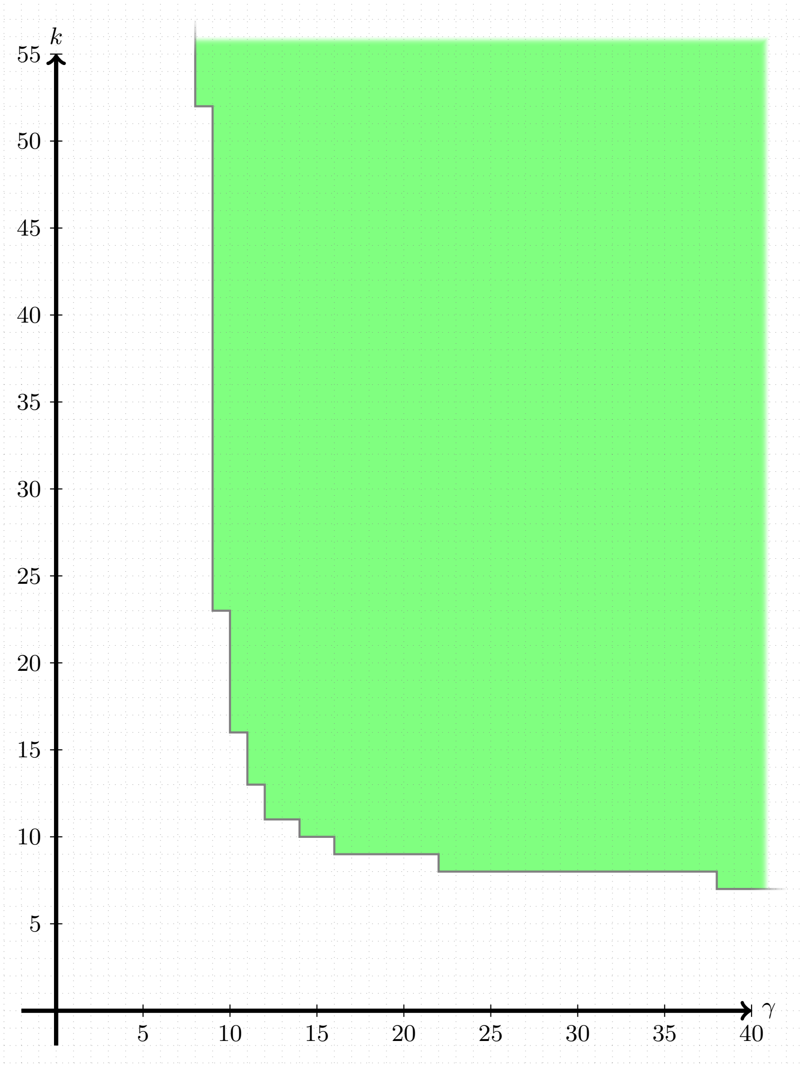

Fig. 1. Region of feasible parameters satisfying the binding property.

that, with overwhelming probability, those extracted messages are permuted integers from 1 to N and powers of α in the same order.

Finally we have ZK_3 and ZK_4, using the extractor of these proofs we obtain openings of $\boldsymbol{c}_{\pi(i)}, \boldsymbol{c}_{\alpha^{\pi(i)}}, \boldsymbol{c}_{u_0^{(i)}}, \boldsymbol{c}_{v_0^{(i)}}$. Given that the commitment scheme is binding we know from previous proofs that those openings are $\pi(i), \alpha^{\pi(i)}, u_0^{(i)}, v_0^{(i)}$. Then, the relations held by the messages committed that were written in terms of m_y are exactly $\sum_{i=1}^{N} \alpha^i u^{(i)} = \sum_{i=1}^{N} \alpha^{\pi(i)}(u'^{(i)} - u_0^{(i)})$ and $\sum_{i=1}^{N} \alpha^i v^{(i)} = \sum_{i=1}^{N} \alpha^{\pi(i)}(v'^{(i)} - v_0^{(i)})$. Applying the generalized Schwartz-Zippel lemma we can ensure with overwhelming probability that $u^{(i)} = u'^{\pi^{-1}(i)} - u_0^{\pi^{-1}(i)}$ and

$v^{(i)} = v'^{\pi^{-1}(i)} - v_0^{\pi^{-1}(i)}$. And this implies that the mix-node has performed a correct shuffle on the input votes.

E Proof of Theorem 1

Proof. We prove the security of a mix-node defining a sequence of games between a challenger and an adversary. Beginning from Game 0, that represents the original attack game with respect to a given efficient adversary, we use a sequence of hybrid arguments, Game 0, Game 1, Game 2 and Game 3, and we show that each game is indistinguishable from the previous one. Transitions between games are done applying very small changes to the defined experiment and we demonstrate that if an adversary can detect them, it would imply an efficient method of distinguishing between two distributions that are computationally indistinguishable under the corresponding assumptions. When Game 3 is reached, ciphertexts at the output of the mix-net are not RLWE samples any more, and are independent from the input.

Game 0 models the probability of an adversary getting output 1 from the experiment.

In Game 3 we have an output which is completely independent from the input and the original messages, and the permutation π is still chosen uniformly at random. Therefore the probability of guessing a correct pair of indices $(i_\mathcal{A}, j_\mathcal{A})$ is equivalent to choosing the second index uniformly at random from $[1, \dots, N]$, that is, sampling J.

This is the sequence of games:

Game (G_0)**.**

- Run Setup algorithm. $(((a_\mathsf{E}, b_\mathsf{E}), s), (\boldsymbol{a}_\mathsf{C}, \boldsymbol{b}_\mathsf{C})) \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$.

$$pk_\mathsf{E} = (a_\mathsf{E}, b_\mathsf{E}) \qquad\qquad pk_\mathsf{C} = (\boldsymbol{a}_\mathsf{C}, \boldsymbol{b}_\mathsf{C})$$

- The adversary chooses the messages. $(\{z^{(i)}\}_{i=1}^N, aux) \xleftarrow{\$} \mathcal{A}_1(pk_\mathsf{E}, pk_\mathsf{C})$.
- The adversary also computes the input of the mix-node.

$$\left(\left\{(u^{(i)}, v^{(i)})\right\}_{i=1}^N\right) \xleftarrow{\$} \mathcal{A}_2\left(\left\{z^{(i)}\right\}_{i=1}^N, aux\right)$$

- Mix the encrypted votes:
  1. Choose a random permutation $\pi \xleftarrow{\$} \mathfrak{S}_N$.
  2. Choose the re-encryption parameters $\{r'^{(i)}_\mathsf{E}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\}_{i=1}^N$ from the appropriate distribution.
  3. Compute the output of the mixing process with their corresponding proofs using the $\mathsf{MixVotes}$ algorithm. $(\{(u'^{(i)}, v'^{(i)})\}_{i=1}^N, \{\Sigma_l\}_{l=0}^4) \leftarrow \mathsf{MixVotes}\left(pk_\mathsf{E}, pk_\mathsf{C}, \{(u^{(i)}, v^{(i)})\}_{i=1}^N; \pi, \left\{r'^{(i)}_\mathsf{E}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\right\}_{i=1}^N\right)$
- $\mathcal{A}$ outputs $(i_\mathcal{A}, j_\mathcal{A}) \xleftarrow{\$} \mathcal{A}_3(\{(u^{(i)}, v^{(i)})\}_{i=1}^N, \{(u'^{(i)}, v'^{(i)})\}_{i=1}^N, \{\Sigma_l\}_{l=0}^4, aux)$.

- Check whether $z^{(i_{\mathcal{A}})} \overset{?}{=} z^{\pi(j_{\mathcal{A}})}$.

Game (G_1)**.**

- Run Setup algorithm. $(((a_{\mathsf{E}}, b_{\mathsf{E}}), s), (\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}})) \xleftarrow{\$} \mathsf{Setup}(1^{\kappa})$.

$$pk_{\mathsf{E}} = (a_{\mathsf{E}}, b_{\mathsf{E}}) \qquad\qquad pk_{\mathsf{C}} = (\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}})$$

- The adversary chooses the messages. $(\{z^{(i)}\}_{i=1}^{N}, aux) \xleftarrow{\$} \mathcal{A}_1(pk_{\mathsf{E}}, pk_{\mathsf{C}})$.
- The adversary also computes the input of the mix-node.

$$\left(\left\{(u^{(i)}, v^{(i)})\right\}_{i=1}^{N}\right) \xleftarrow{\$} \mathcal{A}_2\left(\left\{z^{(i)}\right\}_{i=1}^{N}, aux\right)$$

- Mix the encrypted votes:
  1. Choose a random permutation $\pi \xleftarrow{\$} \mathfrak{S}_N$.
  2. Choose the re-encryption parameters $\{r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\}_{i=1}^{N}$ from the appropriate distribution.

→ 3. Compute the output of the mixing process and simulate their corresponding proofs.

$$(u'^{(i)}, v'^{(i)}) \leftarrow \mathsf{Re\text{-}encrypt}\left(pk_{\mathsf{E}}, u^{\pi(i)}, v^{\pi(i)}; r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\right)$$

$$\{\Sigma_l\}_{l=1}^{4} \xleftarrow{\$} \mathsf{Simulator}\left(pk_{\mathsf{E}}, pk_{\mathsf{C}}, \left\{(u^{(i)}, v^{(i)})\right\}_{i=1}^{N}, \left\{(u'^{(i)}, v'^{(i)})\right\}_{i=1}^{N}\right)$$

Since the zero-knowledge proofs are simulated, they are now independent from the commitments in Σ_0 and we can use their hiding property to substitute each one of them by random samples, without giving to the adversary more advantage in this game than the probability of breaking the RLWE assumption.

- $\mathcal{A}$ outputs $(i_{\mathcal{A}}, j_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_3(\{(u^{(i)}, v^{(i)})\}_{i=1}^{N}, \{(u'^{(i)}, v'^{(i)})\}_{i=1}^{N}, \{\Sigma_l\}_{l=0}^{4}, aux)$.
- Check whether $z^{(i_{\mathcal{A}})} \overset{?}{=} z^{\pi(j_{\mathcal{A}})}$.

Game (G_2)**.**

→ • Run Setup algorithm. $(((a_{\mathsf{E}}, b_{\mathsf{E}}), s), (\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}})) \xleftarrow{\$} \mathsf{Setup}(1^{\kappa})$.

$$a'_{\mathsf{E}}, b'_{\mathsf{E}} \xleftarrow{\$} \mathbb{Z}_q[x] / \langle x^n + 1 \rangle \qquad pk_{\mathsf{E}} = (a'_{\mathsf{E}}, b'_{\mathsf{E}}) \qquad pk_{\mathsf{C}} = (\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}})$$

- The adversary chooses the messages. $(\{z^{(i)}\}_{i=1}^{N}, aux) \xleftarrow{\$} \mathcal{A}_1(pk_{\mathsf{E}}, pk_{\mathsf{C}})$.
- The adversary also computes the input of the mix-node.

$$\left(\left\{(u^{(i)}, v^{(i)})\right\}_{i=1}^{N}\right) \xleftarrow{\$} \mathcal{A}_2\left(\left\{z^{(i)}\right\}_{i=1}^{N}, aux\right)$$

- Mix the encrypted votes:
  1. Choose a random permutation $\pi \xleftarrow{\$} \mathfrak{S}_N$.

2. Choose the re-encryption parameters $\{r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\}_{i=1}^{N}$ from the appropriate distribution.
3. Compute the output of the mixing process and simulate their corresponding proofs.

$$(u'^{(i)}, v'^{(i)}) \leftarrow \mathsf{Re\text{-}encrypt}\left(pk_{\mathsf{E}}, u^{\pi(i)}, v^{\pi(i)}; r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\right)$$

$$\{\Sigma_l\}_{l=0}^{4} \xleftarrow{\$} \mathsf{Simulator}\left(pk_{\mathsf{E}}, pk_{\mathsf{C}}, \left\{(u^{(i)}, v^{(i)})\right\}_{i=1}^{N}, \left\{(u'^{(i)}, v'^{(i)})\right\}_{i=1}^{N}\right)$$

- $\mathcal{A}$ outputs $(i_{\mathcal{A}}, j_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_3(\{(u^{(i)}, v^{(i)})\}_{i=1}^{N}, \{(u'^{(i)}, v'^{(i)})\}_{i=1}^{N}, \{\Sigma_l\}_{l=0}^{4}, aux)$.
- Check whether $z^{(i_{\mathcal{A}})} \overset{?}{=} z^{\pi(j_{\mathcal{A}})}$.

Game $(G_{2,j})$**.** We define G_3 to be $G_{2,N}$ and observe that $G_{2,0}$ is exactly G_2.

- Run Setup algorithm. $(((a_{\mathsf{E}}, b_{\mathsf{E}}), s), (\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}})) \xleftarrow{\$} \mathsf{Setup}(1^{\kappa})$.

$$a'_{\mathsf{E}}, b'_{\mathsf{E}} \xleftarrow{\$} \mathbb{Z}_q[x] / \langle x^n + 1\rangle \qquad pk_{\mathsf{E}} = (a'_{\mathsf{E}}, b'_{\mathsf{E}}) \qquad pk_{\mathsf{C}} = (\boldsymbol{a}_{\mathsf{C}}, \boldsymbol{b}_{\mathsf{C}})$$

- The adversary chooses the messages. $(\{z^{(i)}\}_{i=1}^{N}, aux) \xleftarrow{\$} \mathcal{A}_1(pk_{\mathsf{E}}, pk_{\mathsf{C}})$.
- The adversary also computes the input of the mix-node.

$$(\{(u^{(i)}, v^{(i)})\}_{i=1}^{N}) \xleftarrow{\$} \mathcal{A}_2(\{z^{(i)}\}_{i=1}^{N}, aux)$$

- Mix the encrypted votes:
  1. Choose a random permutation $\pi \xleftarrow{\$} \mathfrak{S}_N$.

→ 2. Choose random polynomials and re-encryption parameters from the appropriate distribution.

$$w'^{(i)}_{u}, w'^{(i)}_{v} \xleftarrow{\$} \mathbb{Z}_q[x] / \langle x^n + 1\rangle \qquad \forall i \in [1, j]$$

$$\{r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}\}_{i=1}^{N} \xleftarrow{\$} \chi^n \qquad \forall i \in [j+1, N]$$

→ 3. Compute the modified output of the mixing process and simulate their corresponding proofs.

$$(u'^{(i)}, v'^{(i)}) = (u^{\pi(i)}, v^{\pi(i)}) + (w'^{(i)}_{u}, w'^{(i)}_{v}) \qquad \forall i \in [1, j]$$

$$(u'^{(i)}, v'^{(i)}) \leftarrow \mathsf{Re\text{-}encrypt}(pk_{\mathsf{E}}, u^{\pi(i)}, v^{\pi(i)}; r'^{(i)}_{\mathsf{E}}, e'^{(i)}_{\mathsf{E},u}, e'^{(i)}_{\mathsf{E},v}) \qquad \forall i \in [j+1, N]$$

$$\{\Sigma_l\}_{l=0}^{4} \xleftarrow{\$} \mathsf{Simulator}(pk_{\mathsf{E}}, pk_{\mathsf{C}}, \{(u^{(i)}, v^{(i)})\}_{i=1}^{N}, \{(u'^{(i)}, v'^{(i)})\}_{i=1}^{N})$$

- $\mathcal{A}$ outputs $(i_{\mathcal{A}}, j_{\mathcal{A}}) \xleftarrow{\$} \mathcal{A}_3(\{(u^{(i)}, v^{(i)})\}_{i=1}^{N}, \{(u'^{(i)}, v'^{(i)})\}_{i=1}^{N}, \{\Sigma_l\}_{l=0}^{4}, aux)$.
- Check whether $z^{(i_{\mathcal{A}})} \overset{?}{=} z^{\pi(j_{\mathcal{A}})}$.

Lemmas 3, 4 and 5 prove that, under RLWE assumptions, all four games above defined are equivalent. For any PPT adversary $\mathcal{A}$ the probability of winning in one of the games is at negligible distance to the probability of winning in any of the other games.

This proves the theorem and ensures that our mix-node is indeed secure. □

We let S_* be the event that $z^{(i_{\mathcal{A}})} = z^{\pi(j_{\mathcal{A}})}$ in game G_*.

Lemma 3. *G_0 and G_1 are statistically indistinguishable.*

Proof. In G_1 instead of generating the proofs $\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4$ using the witnesses, we simulate them. As simulated conversations are statistically close to real ones both games are indistinguishable in probabilistic polynomial time. Additionally, given that the commitment scheme is computationally hiding under the RLWE-assumption, we substitute each commitment in Σ_0 by random samples.

Then

$$|\Pr\{S_0\} - \Pr\{S_1\}| \leq \epsilon_{zkmix} + \epsilon_{hid}$$

where ϵ_{zkmix} is the advantage of an adversary against the zero-knowledge property of $\Sigma_1, \Sigma_2, \Sigma_3$ and Σ_4 and ϵ_{hid} is the advantage of an adversary against the RLWE problem, which are negligible. □

Lemma 4. *G_1 and G_2 are computationally indistinguishable if the RLWE problem is hard.*

Proof. This is immediate as we have just substituted the RLWE sample $(a_{\mathsf{E}}, b_{\mathsf{E}})$ by a uniform sample $(a'_{\mathsf{E}}, b'_{\mathsf{E}}) \xleftarrow{\$} R_q^2$.

Then

$$|\Pr\{S_1\} - \Pr\{S_2\}| \leq \epsilon_{dRLWE}$$

where ϵ_{dRLWE} is the advantage of an adversary against the decisional RLWE problem, which is negligible. □

Lemma 5. *G_2 and G_3 are computationally indistinguishable if the RLWE problem is hard.*

Proof. We can define N intermediate games between G_2 and G_3. $G_{2,0}$ will be G_2, $G_{2,N}$ will be G_3 and in each $G_{2,j}$ we add random $(w_u'^{(i)}, w_v'^{(i)})$ for the first j encryptions and we use the $\mathsf{Re\text{-}encrypt}$ algorithm for all the others from $j+1$ to N, with correctly chosen re-encryption parameters.

Indistinguishability follows from the indistinguishability of any pair of games $G_{2,j}$ and $G_{2,j+1}$.

If they were not indistinguishable we could use them to correctly guess if two pairs of elements (g_1, h_1) and (g_2, h_2) are RLWE samples or uniformly random samples. We would just need to modify $G_{2,j+1}$ assigning $a'_{\mathsf{E}} = g_1, b'_{\mathsf{E}} = g_2, w_u'^{(j+1)} = h_1, w_v'^{(j+1)} = h_2$. If the samples came from a RLWE distribution the game would be exactly $G_{2,j}$, while if samples are uniformly random the game would be $G_{2,j+1}$.

Then

$$|\Pr\{S_{2,j-1}\} - \Pr\{S_{2,j}\}| \leq \epsilon_{dRLWE}$$

where ϵ_{dRLWE} is the advantage of an adversary against the decisional RLWE problem, which is negligible.

Finally, as in G_3 all the re-encryptions are uniformly random samples, it is clear that

$$\Pr\{S_3\} = \Pr\left[z^{(i_A)} = z^{\pi(J)}\right].$$

□

Combining all the probabilities we obtain the advantage of the adversary

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}}^{sec}(\kappa) &= \left|\Pr\left[\mathbf{Exp}_{\mathcal{A}}^{sec}(\kappa) = 1\right] - \Pr\left[z^{(i_A)} = z^{\pi(J)}\right]\right| \\ &= \left|\Pr\{S_0\} - \Pr\{S_3\}\right| \leq \epsilon_{zkmix} + \epsilon_{hid} + (N+1)\epsilon_{dRLWE}\end{aligned}$$

which is negligible since ϵ_{zkmix}, ϵ_{hid} and ϵ_{dRLWE} are negligible.