Lattice-Based Proof of a Shuffle

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

[1] Scytl Secure Electronic Voting, Barcelona, Spain
nuria.costa@scytl.com
[2] Universitat Politècnica de Catalunya, Barcelona, Spain
{ramiro.martinez,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

A. Bracciali et al. (Eds.): FC 2019 Workshops, LNCS 11599, pp. 330–346, 2020.
https://doi.org/10.1007/978-3-030-43725-1_23

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 [31] 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 [1–3,22] 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 [36] 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 sub-linear 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 [31] 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 [35] 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 [35], 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 Sect. 2 we introduce some notation and give some cryptographic background necessary to understand the proof presented in Sect. 4. In Sect. 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 Sect. 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 Sect. 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 $\|\boldsymbol{v}\|_\infty = \max_{1 \leq i \leq N} |v_i|$ and the general norm l_p as $\|\boldsymbol{v}\|_p = (\sum_{i=1}^{N} |v_i|^p)^{1/p}$ for $p \geq 1$.

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 a full version of this paper 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 $\gamma(n) = \mathcal{O}(\sqrt{n} \cdot q/\sigma)$. Additionally, 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 $r_\mathsf{E}, e_{\mathsf{E},u}$, $e_{\mathsf{E},v} \in R_q$ 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 $(u, v) = (a_\mathsf{E} \cdot r_\mathsf{E} + e_{\mathsf{E},u}, b_\mathsf{E} \cdot r_\mathsf{E} + e_{\mathsf{E},v} + \lfloor \frac{q}{2} \rceil z) \in R_q \times R_q$.
- $\mathsf{Decrypt}(\mathsf{s}, (\mathsf{u}, \mathsf{v}))$: 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:

– $\mathsf{KeyGen}_{\mathsf{C}}(1^{\kappa})$: 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 $\boldsymbol{a}_{\mathsf{C}},\boldsymbol{b}_{\mathsf{C}}\xleftarrow{\$}(R_q)^k$, $q\equiv 3 \mod 8$ is prime and n is a power of 2.
– Com: in order to commit to a message $m\in R_q$, the algorithm chooses $r_{\mathsf{C}}\xleftarrow{\$}R_q$ and $\boldsymbol{e}_{\mathsf{C}}\xleftarrow{\$}D^k_{\sigma_e}$ conditioned on $\|\boldsymbol{e}_{\mathsf{C}}\|_\infty\leq n$ 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:

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

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 [32], 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 $\mathbb{Z}_q[x]/\langle x^n+1\rangle$ is not an integral domain also has some implications for the characterization of a permutation matrix proposed in [32], 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]. 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 [35].

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 [34], 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 [33]. 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 $\left\{r_{\mathsf{E}}'^{(i)}, e_{\mathsf{E},u}'^{(i)}, e_{\mathsf{E},v}'^{(i)}\right\}$ for each one of the messages, the shuffling of N RLWE encryptions is defined as $\left(u'^{(i)}, v'^{(i)}\right) = \mathsf{Re\text{-}encrypt}\left(\left(u^{\pi(i)}, v^{\pi(i)}\right), r_{\mathsf{E}}'^{(i)}, e_{\mathsf{E},u}'^{(i)}, e_{\mathsf{E},v}'^{(i)}\right)$.

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 $r_{\mathsf{E}}'^{(i)}, e_{\mathsf{E},u}'^{(i)}, e_{\mathsf{E},v}'^{(i)}$, 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 $\delta \ll q/4$.

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

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 Sect. 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 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).

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:

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

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. $\mathcal{P}$ receives two more random polynomials $\beta, \gamma \xleftarrow{\$} S$. Using the Σ-protocols from [7] that allow him to prove polynomial relations between committed messages, $\mathcal{P}$ proves that he knows openings $m_i, \widehat{m}_i$ to commitments $\boldsymbol{c}_{\pi(i)}, \boldsymbol{c}_{\alpha^{\pi(i)}}$ that satisfy the following relation (ZK_2 in Protocol 1.1).

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

Protocol 1.1. Proof of a shuffle

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

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

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

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

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

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

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

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

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

$\alpha \xleftarrow{\$} S$

$$\xleftarrow{\alpha}$$

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

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

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

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

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

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

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

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

outputs accept if all ZKPoK are correct

Notice that the prover claims that each $\boldsymbol{c}_{\pi(i)}$ is a commitment to $\pi(i)$ with a fixed permutation π. But until it proves that it has indeed committed to a permutation we will refer to the opening of the commitment $\boldsymbol{c}_{\pi(i)}$ as m_i and for the same reason we will call $\widehat{m}_i$ to the opening of the commitment $\boldsymbol{c}_{\alpha^{\pi(i)}}$.

We can consider the two expressions as polynomials in a variable that we can call 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 (Lemma 1). Then the verifier is convinced that with overwhelming probability the two polynomials defined by (2) are equal in $R_q[\Gamma]$ and have the same roots. These roots may be in different order, defined by a permutation $\tilde{\pi}$. For all $i \in [1, \ldots, N]$ we have:

$$\beta m_i + \widehat{m}_i = \beta\tilde{\pi}(i) + \alpha^{\tilde{\pi}(i)}, \quad \text{then} \quad \beta(m_i - \tilde{\pi}(i)) = \alpha^{\tilde{\pi}(i)} - \widehat{m}_i.$$

The polynomials m_i and $\widehat{m}_i$ were fixed before β was chosen. But the permutation $\tilde{\pi}$ was not predetermined. However looking at one i and fixed m_i and $\widehat{m}_i$ we can consider all possible j and study $\beta(m_i - j) = \alpha^j - \widehat{m}_i$. If $(m_i - j) \neq 0$ then there exists at most one $\beta_j \in S$ that fulfills the equation with this particular j (this was trivial in Bayer and Groth's proof, but in our case is again given by the condition of set S). The probability of choosing β equal to one of these (at most N) β_j is negligible. This implies that for each i there exists a j such that $m_i = j$ and $\widehat{m}_i = \alpha^j$. With this reasoning for each i and the previous equations we finally get that, with overwhelming probability $m_i = \tilde{\pi}(i)$ and $\widehat{m}_i = \alpha^{\tilde{\pi}(i)}$.

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).

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

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.

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

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:

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

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 a full version of this paper.

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 [35] 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 [35], 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* 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 [35] 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* MixVotes *algorithm is a secure mix-node according to Definition 2, under the RLWE hardness assumption.*

The proof of Theorem 1 is given in a full version of this paper.

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: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 437–447. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054144
2. Abe, M.: Mix-networks on permutation networks. In: Lam, K.-Y., Okamoto, E., Xing, C. (eds.) ASIACRYPT 1999. LNCS, vol. 1716, pp. 258–273. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-540-48000-6_21
3. Abe, M., Hoshino, F.: Remarks on mix-network based on permutation networks. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 317–324. Springer, Heidelberg (2001). https://doi.org/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: Catalano, D., De Prisco, R. (eds.) SCN 2018. LNCS, vol. 11035, pp. 368–385. Springer, Cham (2018). https://doi.org/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: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 646–663. Springer, Heidelberg (2013). https://doi.org/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: Pernul, G., Ryan, P.Y.A., Weippl, E. (eds.) ESORICS 2015, Part I. LNCS, vol. 9326, pp. 305–325. Springer, Cham (2015). https://doi.org/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 (2018). https://doi.org/10.1109/SP.2018.00020
9. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–90 (1981). https://doi.org/10.1145/358549.358563
10. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: A homomorphic LWE based E-voting scheme. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 245–265. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_16
11. Costa, N., Martínez, R., Morillo, P.: Proof of a shuffle for lattice-based cryptography. In: Lipmaa, H., Mitrokotsa, A., Matulevičius, R. (eds.) NordSec 2017. LNCS, vol. 10674, pp. 280–296. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70290-2_17
12. Cramer, R., Damgård, I., Xing, C., Yuan, C.: Amortized complexity of zero-knowledge proofs revisited: achieving linear soundness slack. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 479–500. Springer, Cham (2017). https://doi.org/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: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 365–394. Springer, Cham (2017). https://doi.org/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: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017, pp. 1565–1581. ACM Press, Dallas (2017). https://doi.org/10.1145/3133956.3134101
16. Furukawa, J.: Efficient and verifiable shuffling and shuffle-decryption. IEICE Trans. **88**–**A**, 172–188 (2005)
17. Furukawa, J., Sako, K.: An efficient scheme for proving a shuffle. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 368–387. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_22
18. Groth, J.: A verifiable secret shuffe of homomorphic encryptions. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 145–160. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36288-6_11
19. Groth, J., Ishai, Y.: Sub-linear zero-knowledge argument for correctness of a shuffle. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 379–396. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_22
20. Groth, J., Lu, S.: Verifiable shuffle of large size ciphertexts. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 377–392. Springer, Heidelberg (2007). https://doi.org/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: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_1
22. Markus, J., Ari, J.: Millimix: mixing in small batches. Technical report, Center for Discrete Mathematics, Theoretical Computer Science (1999)
23. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography, pp. 147–191. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7
24. Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: Reiter, M.K., Samarati, P. (eds.) ACM CCS 2001, pp. 116–125. ACM Press, Philadelphia (2001). https://doi.org/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: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 84–93. ACM Press, Baltimore (2005). https://doi.org/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: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 393–403. Springer, Heidelberg (1995). https://doi.org/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). https://doi.org/10.1137/S0097539795293172
30. Singh, K., Pandu Rangan, C., Banerjee, A.K.: Lattice based mix network for location privacy in mobile system. Mob. Inf. Syst. **2015**, 1–9 (2015). https://doi.org/10.1155/2015/963628

31. Strand, M.: A verifiable shuffle for the GSW cryptosystem. In: Zohar, A., et al. (eds.) FC 2018. LNCS, vol. 10958, pp. 165–180. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-58820-8_12
32. Terelius, B., Wikström, D.: Proofs of restricted shuffles. In: Bernstein, D.J., Lange, T. (eds.) AFRICACRYPT 2010. LNCS, vol. 6055, pp. 100–113. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12678-9_7
33. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 755–784. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_25
34. Unruh, D.: Post-quantum security of Fiat-Shamir. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 65–95. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_3
35. Wikström, D.: The security of a mix-center based on a semantically secure cryptosystem. In: Menezes, A., Sarkar, P. (eds.) INDOCRYPT 2002. LNCS, vol. 2551, pp. 368–381. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36231-2_29
36. Wikström, D.: A commitment-consistent proof of a shuffle. In: Boyd, C., González Nieto, J. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 407–421. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02620-1_28