One-Shot Verifiable Encryption from Lattices

Vadim Lyubashevsky(✉) and Gregory Neven

IBM Research, Zurich, Switzerland
{vad,nev}@zurich.ibm.com

Abstract. Verifiable encryption allows one to prove properties about encrypted data and is an important building block in the design of cryptographic protocols, e.g., group signatures, key escrow, fair exchange protocols, etc. Existing lattice-based verifiable encryption schemes, and even just proofs of knowledge of the encrypted data, require parallel composition of proofs to reduce the soundness error, resulting in proof sizes that are only truly practical when amortized over a large number of ciphertexts.

In this paper, we present a new construction of a verifiable encryption scheme, based on the hardness of the Ring-LWE problem in the random-oracle model, for short solutions to linear equations over polynomial rings. Our scheme is "one-shot", in the sense that a single instance of the proof already has negligible soundness error, yielding compact proofs even for individual ciphertexts. Whereas verifiable encryption usually guarantees that decryption can recover a witness for the original language, we relax this requirement to decrypt a witness of a related but extended language. This relaxation is sufficient for many applications and we illustrate this with example usages of our scheme in key escrow and verifiably encrypted signatures.

One of the interesting aspects of our construction is that the decryption algorithm is probabilistic and uses the proof as input (rather than using only the ciphertext). The decryption time for honestly-generated ciphertexts only depends on the security parameter, while the *expected* running time for decrypting an adversarially-generated ciphertext is directly related to the number of random-oracle queries of the adversary who created it. This property suffices in most practical scenarios, especially in situations where the ciphertext proof is part of an interactive protocol, where the decryptor is substantially more powerful than the adversary, or where adversaries can be otherwise discouraged to submit malformed ciphertexts.

1 Introduction

Lattice cryptography has matured to the point where the general belief is that any primitive that can be constructed from any other assumption can also be constructed based on a lattice assumption. The main question that remains is how efficient (in a practical, rather than asymptotic, sense) one can make the lattice-based constructions. A primitive that has been getting a lot of recent attention is a "proof of plaintext knowledge."

J.-S. Coron and J.B. Nielsen (Eds.): EUROCRYPT 2017, Part I, LNCS 10210, pp. 293–323, 2017.
DOI: 10.1007/978-3-319-56620-7_11

In a proof of plaintext knowledge, a prover who has a message μ produces a ciphertext $t = \mathsf{Enc}(\mu)$ and a zero-knowledge proof of knowledge π showing that he knows the value of $\mathsf{Dec}(t)$. Proving knowledge of the value of $\mathsf{Dec}(t)$ is usually the same as proving that t is a correctly formed ciphertext along with proving the knowledge of μ that was used to construct it.

By itself, a proof of plaintext knowledge is not particularly useful, and it is almost always used as a part of a primitive known as a verifiable encryption scheme. In such a scheme, there is a relation R_L and a language

$$L = \{x \ : \ \exists w \text{ s.t. } R_L(x, w) = 1\}.$$

Thus the value w is a witness to the fact that x is in the language L. The relation R_L and the element x are public, while the prover possesses the secret witness w. He then produces an encryption $t = \mathsf{Enc}(w)$ as well as a zero-knowledge proof of knowledge π of the value $w = \mathsf{Dec}(t)$ and that w satisfies $R_L(x, w) = 1$. Verifiable encryption can therefore also be seen as an extractable non-interactive zero-knowledge proof. It is a building block for many primitives, e.g.,

- group signatures [CvH91], where a group manager hands distinct signing keys to all users, using which they can anonymously sign messages. A trusted opener is able to trace back a signature to the identity of the signer. A common construction [CL06] is to let users verifiably encrypt their identity under the opener's public key together with a proof that they know a signature by the group manager on the same identity.
- key escrow protocols [YY98,PS00], where users encrypt their decryption key under the public key of a trusted escrow authority. Using verifiable encryption, communication partners or network providers can check that the ciphertext indeed encrypts the user's decryption key, and not some bogus data.
- optimistic fair exchange protocols [ASW00,BDM98], where two parties can fairly exchange secrets by, in a first step, proving that they encrypted their respective secrets under the public key of a trusted authority, who can later be called upon to recover the secret in case one of the parties aborts the protocol early;
- verifiable secret sharing, where one dealer sends verifiably encrypted shares of a secret to a set of parties, and proves to an external third party that the ciphertexts contain actual shares of the secret.

1.1 Proofs of Plaintext Knowledge from Lattices – Prior Work

If one uses a lattice-based encryption scheme based on LWE or Ring-LWE, then the encryption of a message m satisfies the linear relation $\mathbf{A}\begin{bmatrix}\mathbf{r}\\ m\end{bmatrix} = \mathbf{t} \bmod q$. There are several known techniques to prove that the ciphertext $\mathbf{t}$ is well-formed and one knows the message m. One technique is adapting Stern's protocol based on permutations [Ste93] to lattice-based schemes [LNSW13]. This approach is unfortunately very impractical due to the fact that each round of the protocol has soundness error 2/3 (and therefore needs to be repeated 192 times to achieve

128-bit security). Furthermore, if proving relations where the secret has coefficients of size k, the size of the proof increases by a factor of $\log k$ [LNSW13]. This makes schemes using Stern's protocol unsuitable for most practical applications.

Another approach is to use the "Fiat-Shamir with Aborts" zero-knowledge proof technique from [Lyu09,Lyu12] with 0/1 challenges. This also has the problem of having soundness error 1/2 and needing to be repeated 128 times for 128-bit security. This approach, however, is more algebraic than Stern's proof of knowledge and it was shown to admit several improvements. If one uses a Ring-LWE encryption scheme, then it was shown in [BCK+14] how the soundness error can be reduced to $1/(2n)$, where n is the dimension of the ring being used (typically 1024 or 2048). For 128-bit security, one then only needs to run the protocol around a dozen times. A scenario where the Fiat-Shamir with Aborts technique leads to truly practical protocols is when one wants to simultaneously do many proofs of plaintext knowledge. If one then considers the amortized cost of the proof of knowledge, then the number of iterations is only a small constant (approaches 2 as the number of instances increases) per proof [DPSZ12,BDLN16,CD16].

Despite having received considerable attention in the literature, there seems to be no satisfactory solution for the most natural scenario where the prover has a single instance to prove and would like to do it in "one shot"—that is, without repeating a protocol to amplify soundness. It is therefore conceivable that lattice-based encryption schemes are not compatible with efficient proofs of plaintext knowledge, which would make all the applications much less efficient than their number theoretic counterparts.

1.2 Proofs of Plaintext Knowledge – Our Results

In this work, we introduce a very efficient "one-shot" protocol for proving plaintext knowledge. The caveat is that the running time of the decryption algorithm depends on the running time of the prover. More precisely, our decryption algorithm is *randomized* in that it tries to decrypt ciphertexts that are "close" to the one provided by the prover. And we show that the *expected* number of decryption tries our decryptor needs is within a small factor (essentially 1) of the number of *random oracle queries* that the prover makes while constructing the proof of knowledge π. If q is the number of queries made by the prover, then Markov's inequality implies that the probability that the decryptor will need more than $\alpha \cdot q$ decryption tries is less than $1/\alpha$. If the prover is honest, though, then the decryptor will succeed from the first try.

While tying the decryption time to the adversary's running time is unusual, this should be acceptable in many scenarios. Apart from creating out-of-band incentives such as fines to prevent cheating, there are also technical ways to limit the power of the adversary. If the protocol in which the proof of knowledge is being used is interactive, then the verifier can send the prover a fresh salt during every interaction that has to be included in the cryptographic hash function (modeled as a random oracle) and require that the prover performs the proof

within a certain small amount of time. Thus the adversary will have a limited time-frame during which he can make queries to the random oracle (because each new salt in essence creates a new random oracle). The decryption algorithm, on the other hand, is almost always off-line and is therefore allowed more time. In non-interactive settings, the prover can be required to use a salt from a public "randomness beacon" (such as one provided by NIST) at the time the proof was created.

In our scheme, the verification algorithm uses one random oracle query and the decryption algorithm uses none. Thus another simple way of preventing an adversary from using too many random oracle queries during encryption would be to artificially make the computational complexity of computing the hash function high (e.g. by iterating SHA-256 some number of times to produce one output). This has the effect of significantly slowing down a cheating prover, while keeping the decryption time exactly the same.

We also show that, if one wishes, one can upper-bound the running time of the decryptor by making the protocol "k-shot" rather than one shot. In this scenario, the length of the proof of knowledge would go up by a factor k, but one could bound the decryption algorithm's running time to $k \cdot 2^{\lambda/k}$ for λ-bit security.

1.3 Verifiable Encryption Schemes – Our Results

We build upon our proof of plaintext knowledge to construct a verifiable encryption scheme that is adapted to be used as a building block for lattice constructions. The relations that are most common in lattice cryptography are those of the form

$$\mathbf{Bm} = \mathbf{u} \bmod p \tag{1}$$

where $\mathbf{B}$ is a matrix over some ring, $\mathbf{m}$ is a vector with small coefficients, and $\mathbf{u}$ is the product of $\mathbf{Bm}$ modulo p. For example, in (Ring)-LWE encryption $\mathbf{B}, \mathbf{u}$ is the public key and $\mathbf{m}$ is the secret key. In full domain hash signatures, $\mathbf{B}$ is the public key, $\mathbf{m}$ is the signature, and $\mathbf{u} = \mathsf{H}(\mu)$ is derived from the message μ. Giving a verifiable encryption scheme for such relations is a main building block for many of the protocols listed in the introduction.

While verifiable encryption would normally guarantee that decrypting a valid ciphertext yields a witness satisfying (1), our construction relaxes this guarantee to only yield a witness $(\overline{\mathbf{m}}, \bar{\mathrm{c}})$ with small coefficients satisfying

$$\mathbf{B}\overline{\mathbf{m}} = \bar{\mathrm{c}}\mathbf{u} \bmod p. \tag{2}$$

This relaxation actually turns out to be sufficient for many applications of verifiable encryption. Lattice schemes can often be slightly augmented to allow for relations of the form (2) to be "useful" whenever those of the form (1) are. We will see this in the two examples provided in Sect. 6.

Notice also how it appears as if the decryption and the proof of knowledge are disjoint. Indeed, the proof of knowledge π may prove the existence of some witness $(\overline{\mathbf{m}}, \bar{\mathrm{c}})$, whereas the decryption algorithm may obtain a completely different

witness $(\overline{\mathbf{m}}', \overline{\mathrm{c}}')$. But in addition to still being sufficient for many applications, there is also a connection between the two tuples that is actually crucial to our construction – we have that

$$\overline{\mathbf{m}}/\overline{\mathrm{c}} = \overline{\mathbf{m}}'/\overline{\mathrm{c}}' \bmod p.$$

While this property is not needed in many applications, the presence of this relationship may be useful when constructing group signatures or other primitives where it is important that the decryption recovers some specific attribute of the prover rather than just a witness to a relation.

1.4 Paper Organization

We present the Ring-LWE encryption scheme and the non-interactive "Fiat-Shamir with Aborts" zero-knowledge proof protocol in Sects. 2.5 and 2.6. Slight variations of these two primitives are used throughout our constructions.

We then present the definitions of our relaxed version of verifiable encryption in Sect. 3.1, and describe all the elements of the scheme in Sect. 3.2. In Sect. 7, we give some example instantiations for the proofs of plaintext knowledge and verifiable encryption schemes. The proof of plaintext knowledge scheme requires 9 KB for the ciphertext and 9 KB for the proof. This is quite efficient since this ciphertext is only around 4 times larger than a regular Ring-LWE ciphertext.

The efficiency of the verifiable encryption scheme is mostly affected by the size of the modulus p and the witness $\mathbf{m}$ in the relation. The larger these values, the larger the proofs and ciphertexts will be. The sample instantiations in Sect. 7 are meant to support the two sample applications in Sect. 6, where we describe how our relaxed verifiable encryption scheme can be used to build key escrow schemes and verifiably encrypted signatures. In Sects. 4 and 5 we describe two variants of our schemes, the former trading longer ciphertexts for bounded decryption time, the latter adding simulatability under chosen-ciphertext to the scheme.[1]

2 Preliminaries

For a set S, we write $a \xleftarrow{\$} S$ to mean that a is chosen uniformly at random from S. If D is a distribution, then $a \xleftarrow{\$} D$ signifies that a is randomly chosen according to the distribution D. The assignment operator $a \leftarrow b$ signifies that a gets assigned the value b. We will also sometimes write column vectors of the form $\begin{bmatrix} a_1 \\ \dots \\ a_k \end{bmatrix}$ as $[\, a_1 \; ; \; \dots \; ; \; a_k \,]$.

[1] Unlike Camenisch and Shoup [CS03], we cannot use standard indistinguishability security notions, because our decryption algorithm needs the proof to be included in the ciphertext.

2.1 The Ring $\mathbb{Z}[\mathrm{x}]/(\mathrm{x}^n + 1)$.

Consider the ring $R = \mathbb{Z}[\mathrm{x}]/(\mathrm{x}^n + 1)$ and $R_q = \mathbb{Z}_q[\mathrm{x}]/(\mathrm{x}^n + 1)$ where n is a power of 2 integer and q is some prime. The elements of the latter ring are polynomials of degree at most $n-1$ with coefficients between $-(q-1)/2$ and $(q-1)/2$ (for the ring R, there is no restriction on the sizes of coefficients). All definitions that follow apply both to R and R_q. We will denote elements of $\mathbb{Z}$ and of R by lower-case letters, elements of vectors in R^k by bold lower-case letters, and of matrices in $R^{k\times l}$ by bold upper-case letters.

We will define the ℓ_1, ℓ_2, and ℓ_∞ lengths of an element $\mathrm{a} = \sum_{i=0}^{n-1} a_i \mathrm{x}^i \in R$ as

$$\|\mathrm{a}\|_1 = \sum_{i=0}^{n-1} |a_i|, \quad \|\mathrm{a}\| = \sqrt{\sum_{i=0}^{n-1} a_i^2} \quad \text{and} \quad \|\mathrm{a}\|_\infty = \max_i(|a_i|)$$

respectively.[2] For k-dimensional vectors $\mathbf{a} = [\ \mathrm{a}_1 \mid \ldots \mid \mathrm{a}_k\ ] \in R^k$, we write $\|\mathbf{a}\|_1 = \|\mathrm{a}_1\|_1 + \ldots + \|\mathrm{a}_k\|_1$, $\|\mathbf{a}\| = \sqrt{\|\mathrm{a}_1\|^2 + \ldots + \|\mathrm{a}_k\|^2}$ and $\|\mathbf{a}\|_\infty = \max_i \|\mathrm{a}_i\|_\infty$. We will denote by S_i (respectively S_i^k), the set of elements of R (resp. of R^k) whose ℓ_∞ length is at most i.

It is not hard to check that for any two polynomials $\mathrm{a}, \mathrm{b} \in R$, we have $\|\mathrm{ab}\|_\infty \leq \|\mathrm{a}\|_1 \cdot \|\mathrm{b}\|_\infty$ and $\|\mathrm{ab}\|_\infty \leq \|\mathrm{a}\| \cdot \|\mathrm{b}\|$. Similarly for $\mathbf{a}, \mathbf{b} \in R^k$, we have the same inequalities on the ℓ_∞ norms of their inner products: that is, $\|\mathbf{a} \cdot \mathbf{b}\|_\infty \leq \|\mathbf{a}\|_1 \cdot \|\mathbf{b}\|_\infty$ and $\|\mathbf{a} \cdot \mathbf{b}\|_\infty \leq \|\mathbf{a}\| \cdot \|\mathbf{b}\|$.

2.2 Special Properties of $\mathbb{Z}_q[\mathrm{x}]/(\mathrm{x}^n + 1)$

The algebraic properties of the ring $R_q = Z_q[\mathrm{x}]/(\mathrm{x}^n + 1)$, where n is a power of 2, depend on the prime q. For efficiency, one often takes $q = 1 \bmod (2n)$, which results in the polynomial $\mathrm{x}^n + 1$ splitting into n linear factors modulo q. Operations within the ring can then be performed extremely efficiently using the number theory transform. On the other hand, one sometimes wants the ring to be "almost a field". In particular, it is sometimes desirable for the ring to have many invertible elements. While there do not exist q that will make R_q a field, using $q = 3 \bmod 8$ (as first suggested in [SSTX09]) has the effect that $\mathrm{x}^n + 1$ factors into two irreducible polynomials of degree $n/2$ and so the ring R_q contains $q^n - 2q^{n/2} + 1$ invertible elements. By the Chinese Remainder theorem, it is also easy to see that all elements of degree less than $n/2$ are invertible.

We have not seen this used in previous works, but it turns out that setting $q = 5 \bmod 8$ may be even more convenient. Modulo such a q, the polynomial $\mathrm{x}^n + 1$ also factors into two irreducible polynomials of degree $n/2$. And in addition to all elements of degree less than $n/2$ being invertible, one can also show that all elements (of degree up to n) with small coefficients are invertible as well. We

[2] We point out that since $\mathbb{Z}_q$ is a finite group, these do not correspond exactly to norms when working in R_q because we do not have $\|\alpha \cdot \mathrm{a}\| = \alpha \cdot \|\mathrm{a}\|$. The other two properties of norms (i.e. that the norm of 0 is 0 and the triangle inequality) do hold.

present the proof of this statement in Lemma 2.2. To the best of our knowledge, this lemma was first proven in an unpublished manuscript of Lyubashevsky and Xagawa.

Lemma 2.1 ([LN86], special case of Theorem 3.35, p. 88). *Let* $q = 5 (\mathrm{mod}\ 8)$ *be prime and* r *be an integer such that* $r^2 = -1 (\mathrm{mod}\ q)$. *Then for all positive integers* κ, *the polynomials* $\mathrm{x}^{2^\kappa} - r$ *and* $\mathrm{x}^{2^\kappa} + r$ *are irreducible over* $\mathbb{Z}_q[\mathrm{x}]$. *And in particular, the complete factorization into irreducibles over* $\mathbb{Z}_q[\mathrm{x}]$ *of the polynomial* $\mathrm{x}^{2^{\kappa+1}} + 1$ *is* $\mathrm{x}^{2^{\kappa+1}} + 1 = (\mathrm{x}^{2^\kappa} - r)(\mathrm{x}^{2^\kappa} + r) \bmod q$.

Lemma 2.2. *Let* $R_q = \mathbb{Z}_q[\mathrm{x}]/(x^n + 1)$ *where* $n > 1$ *is a power of* 2 *and* q *is a prime congruent to* $5 (\mathrm{mod}\ 8)$. *This ring has exactly* $2q^{n/2} - 1$ *elements without an inverse. Moreover, every non-zero polynomial* a *in* R_q *with* $\|\mathrm{a}\|_\infty < \sqrt{q/2}$ *has an inverse.*

Proof. In all that follows, the reduction modulo q will be implicit. By Lemma 2.1, $\mathrm{x}^n + 1 = (\mathrm{x}^{n/2} - r)(\mathrm{x}^{n/2} + r)$ where $r^2 = -1$ and $\mathrm{x}^{n/2} \pm r$ are irreducible. Any element $\mathrm{a} \in R_q$ can be written as $\mathrm{a} = \mathrm{a}_0 + \mathrm{x}^{n/2}\mathrm{a}_1$, where $\mathrm{a}_0, \mathrm{a}_1$ are polynomials in $\mathbb{Z}[\mathrm{x}]$ of degree less than $n/2$. Then the Chinese remainder decomposition of a is

$$CRT(\mathrm{a}) = (\mathrm{a} \bmod (\mathrm{x}^{n/2} - r), \mathrm{a} \bmod (\mathrm{x}^{n/2} + r)) = (\mathrm{a}_0 + r\mathrm{a}_1, \mathrm{a}_0 - r\mathrm{a}_1).$$

If a is not invertible, it means that either $\mathrm{a}_0 + r\mathrm{a}_1 = 0$ or $\mathrm{a}_0 - r\mathrm{a}_1 = 0$. If $\mathrm{a}_1 = 0$, then $\mathrm{a}_0 = 0$ and a is the zero polynomial. If $\mathrm{a}_1 \neq 0$, then some coefficient of a_0, say α_0, must be equal to $\pm r\alpha_1$, where α_1 is a non-zero coefficient of a_1. Therefore we have $\alpha_0^2 = (\pm r\alpha_1)^2 = -\alpha_1^2$. In other words, $\alpha_0^2 + \alpha_1^2 = 0$. But since we assumed that $|\alpha_0|, |\alpha_1| < \sqrt{q/2}$, this is not possible, and thus proves the second part of the lemma by contradiction. The first part of the lemma follows from the fact that CRT is a bijection and all the elements without an inverse must be 0 modulo at least one of $\mathrm{x}^{n/2} \pm r$. □

2.3 Lattices and the Discrete Gaussian Distribution

A full-rank integer lattice Λ of dimension n is an additive subgroup of $\mathbb{Z}^n$ that is generated by some basis $\mathbf{B} = [\mathbf{b}_1 \mid \ldots \mid \mathbf{b}_n] \in \mathbb{Z}^{n \times n}$ of linearly-independent vectors. If a basis $\mathbf{B}$ is a generator for a lattice Λ, we will write $\mathcal{L}(\mathbf{B}) = \Lambda$.

For a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$, we define

$$\mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \ : \ \mathbf{A}\mathbf{y} = 0 \bmod q\}. \tag{3}$$

It's easy to see that $\mathcal{L}^\perp(\mathbf{A})$ is a full-rank lattice of dimension m.

For a full-rank integer lattice Λ, we define the discrete Gaussian distribution $D_{\Lambda,\mathbf{c},\sigma}(\mathbf{v}) = e^{\frac{-\|\mathbf{v}-\mathbf{c}\|^2}{2\sigma^2}} \Big/ \sum_{\mathbf{w}\in\Lambda} e^{\frac{-\|\mathbf{w}-\mathbf{c}\|^2}{2\sigma^2}}$ for any $\mathbf{v} \in \Lambda$, and 0 on all other points in space.

For the special case of $\Lambda = \mathbb{Z}^m$, we know that

$$\Pr_{\mathbf{s} \xleftarrow{\$} D_{\mathbb{Z}^m,0,\sigma}} [\|\mathrm{s}\|_\infty > t\sigma] < 2m \cdot e^{-t^2/2},$$

which implies that for $t = 6$, the probability that any coefficient of $\mathbf{s}$ is greater than 6σ is less than $m \cdot 2^{-25}$.

2.4 Polynomial Lattices and Sampling over the Ring R

In this paper, rather than working over the ring $\mathbb{Z}_q$ (with the usual addition and multiplication operation modulo q), we will be working over the ring $R = \mathbb{Z}_q[\mathrm{x}]/(\mathrm{x}^n + 1)$ with the usual addition and multiplication operations modulo q and $x^n + 1$. Analogously to (3), for a vector $\mathbf{A} \in R^{1\times m}$, a lattice $\mathcal{L}^\perp(\mathbf{A})$ can be defined as

$$\mathcal{L}^\perp(\mathbf{A}) = \{\mathrm{y} \in (\mathbb{Z}[\mathrm{x}]/(\mathrm{x}^n + 1))^m : \mathbf{A}\mathrm{y} = 0 \bmod q\}.$$

Note that while it is an m-dimensional lattice over $\mathbb{Z}[\mathrm{x}]/(\mathrm{x}^n + 1)$, it is really an nm-dimensional lattice over $\mathbb{Z}$.

If we want to generate a discrete Gaussian sample over $\mathbb{Z}[\mathrm{x}]/(\mathrm{x}^n + 1)$, we can simply generate it over $\mathbb{Z}^n$ and then map into $\mathbb{Z}[\mathrm{x}]/(\mathrm{x}^n + 1)$ using the obvious embedding of coordinates into coefficients of the polynomials. We will slightly abuse notation and write $\mathrm{y} \xleftarrow{\$} D_{R,0,\sigma}$ to mean that y is generated according to $D_{\mathbb{Z}^n,0,\sigma}$ and then interpreted as an element of R. Similarly, we write $(\mathrm{y}_1, \ldots, \mathrm{y}_l) \xleftarrow{\$} D_{R^l,0,\sigma}$ to mean that z is generated according to $D_{\mathbb{Z}^{ln},0,\sigma}$ and then gets interpreted as l polynomials y_i.

2.5 Ring-LWE Encryption Scheme

We recall the Ring-LWE encryption scheme from [LPR13]. For simplicity, we take the distribution of the secret keys and the randomness to be uniformly-random elements with ℓ_∞ norm 1. The secret keys are chosen as $\mathrm{s}_1, \mathrm{s}_2 \xleftarrow{\$} S_1$ and the public keys are $\mathrm{a} \xleftarrow{\$} R_q$ and $\mathrm{t} \leftarrow \mathrm{as}_1 + \mathrm{s}_2$. There is also a public parameter $p > 2$, which is a positive integer. To encrypt a message $\mathrm{m} \in R_p$, the encryptor chooses $\mathrm{r}, \mathrm{e}, \mathrm{e}' \xleftarrow{\$} S_1$ and outputs (v, w) where $\mathrm{v} \leftarrow p(\mathrm{ar} + \mathrm{e})$ and $\mathrm{w} \leftarrow p(\mathrm{tr} + \mathrm{e}') + \mathrm{m}$. The decryption procedure computes

$$\mathrm{w} - \mathrm{vs}_1 \bmod q \bmod p = p(\mathrm{rs}_2 + \mathrm{e}' - \mathrm{es}_1) + \mathrm{m} \bmod p = \mathrm{m}, \tag{4}$$

where the last equality holds in the case that $\|p(\mathrm{rs}_2 + \mathrm{e}' - \mathrm{es}_1) + \mathrm{m}\|_\infty < q/2$.

From the above equations, we see that the encryption of a plaintext m under public keys a, t is a ciphertext v, w satisfying the equation

$$\begin{bmatrix} \mathrm{v} \\ \mathrm{w} \end{bmatrix} = \left[\begin{array}{c|c|c|c} p\mathrm{a} & p & 0 & 0 \\ p\mathrm{t} & 0 & p & 1 \end{array}\right] \begin{bmatrix} \mathrm{r} \\ \mathrm{e} \\ \mathrm{e}' \\ \mathrm{m} \end{bmatrix} \bmod q, \tag{5}$$

Extending this, the encryption of k messages $\mathrm{m}_1, \ldots, \mathrm{m}_k$ under the same public key a, t satisfies the following relation:

$$\overbrace{\left\{2k\left\{\begin{bmatrix} \mathrm{pa} & & & \mathrm{p} & & & & & & \\ \cdots & & & \cdots & & & & & & \\ & \mathrm{pa} & & & \mathrm{p} & & & & & \\ \mathrm{pt} & & & & & \mathrm{p} & & 1 & & \\ \cdots & & & & & & \cdots & & \cdots & \\ & \mathrm{pt} & & & & & & \mathrm{p} & & 1 \end{bmatrix}\right.\right.}^{4k} \begin{bmatrix} \mathrm{r}_1 \\ \cdots \\ \mathrm{r}_k \\ \mathrm{e}_1 \\ \cdots \\ \mathrm{e}_k \\ \mathrm{e}'_1 \\ \cdots \\ \mathrm{e}'_k \\ \mathrm{m}_1 \\ \cdots \\ \mathrm{m}_k \end{bmatrix} = \begin{bmatrix} \mathrm{v}_1 \\ \cdots \\ \mathrm{v}_k \\ \mathrm{w}_1 \\ \cdots \\ \mathrm{w}_k \end{bmatrix} \bmod q, \tag{6}$$

which we will write in abbreviated form as

$$\begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{e} \\ \mathbf{e}' \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \bmod q, \tag{7}$$

where $\mathbf{I}_k$ corresponds to an identity matrix of dimension k and $0^{\ell\times k}$ corresponds to an $\ell \times k$ matrix of all zeroes. The decryption procedure is then simply the vector analogy of (4), i.e.

$$\mathbf{m} = \mathbf{w} - \mathbf{v}\mathrm{s}_1 \bmod q \bmod p.$$

2.6 "Fiat-Shamir with Aborts" Proofs of Knowledge of Linear Relations

In [Lyu09,Lyu12], Lyubashevsky introduced a technique for constructing practical digital signatures (in the random oracle model) based on the hardness of lattice problems. At the heart of the construction is a zero-knowledge proof of knowledge that, given an $\mathbf{s} \in R^k$ satisfying the relation

$$\mathbf{A}\mathbf{s} = \mathbf{t} \bmod q, \tag{8}$$

proves the knowledge of low-norm $\bar{\mathbf{s}}$ and $\bar{\mathrm{c}}$ that satisfy

$$\mathbf{A}\bar{\mathbf{s}} = \bar{\mathrm{c}}\mathbf{t} \bmod q.$$

The idea in [Lyu09,Lyu12] was to construct a Σ-protocol with the main twist being that the prover does not always output the result. In particular, the protocols use rejection sampling to tailor the distribution so that it does not depend on the secret $\mathbf{s}$. This rejection sampling can be done by making

Algorithm 1. "Fiat-Shamir with Aborts" zero-knowledge proof of knowledge

Input: A matrix $\mathbf{A} \in R^{\ell \times k}$, vector $\mathbf{s} \in S \subset R^k$, a vector $\mathbf{t} \in R^\ell$, and a vector $\mathbf{q} \in \mathbb{Z}^\ell$ such that $\mathbf{As} = \mathbf{t} \bmod \mathbf{q}$. Challenge domain $\mathcal{C} \subset R$. Standard deviation $\sigma \in \mathbb{R}^+$ such that $\sigma \geq 11 \cdot \max_{\mathbf{s} \in S, \mathrm{c} \in \mathcal{C}} \|\mathbf{sc}\|$. Cryptographic hash function $\mathsf{H} : \{0,1\}^* \to \mathcal{C}$.
Output: $\mathbf{z} \in R^k$ such that $\mathbf{z} \sim D_{R^k,0,\sigma}$ and $\mathrm{c} \in \mathcal{C}$ such that $\mathrm{c} = \mathsf{H}(\mathbf{A}, \mathbf{t}, \mathbf{Az} - \mathbf{t}\mathrm{c} \bmod \mathbf{q})$.

1: $\mathbf{y} \xleftarrow{\$} D_{R^k,0,\sigma}$
2: $\mathrm{c} \leftarrow \mathsf{H}(\mathbf{A}, \mathbf{t}, \mathbf{Ay} \bmod \mathbf{q})$
3: $\mathbf{z} \leftarrow \mathbf{sc} + \mathbf{y}$
4: with probability $\frac{D_{R^k,0,\sigma}(\mathbf{z})}{3 \cdot D_{R^k,\mathbf{sc},\sigma}(\mathbf{z})}$, **goto** 1
5: **if** $\|\mathbf{z}\|_\infty > 6\sigma$, **goto** 1
6: output $(\mathrm{c}, \mathbf{z})$

Algorithm 2. "Fiat-Shamir with Aborts" Verification Algorithm

Input: A matrix $\mathbf{A} \in R^{\ell \times k}$, a vector $\mathbf{t} \in R^\ell$, a vector $\mathbf{q} \in \mathbb{Z}^\ell$, $\sigma \in \mathbb{R}^+$. A tuple $(\mathrm{c}, \mathbf{z}) \in \mathcal{C} \times R^k$. Cryptographic hash function $\mathsf{H} : \{0,1\}^* \to \mathcal{C}$.
Output: Bits 0 or 1 corresponding to Reject/Accept.

1: **if** $\|\mathbf{z}\|_\infty > 6\sigma$, **return** 0
2: **if** $\mathrm{c} \neq \mathsf{H}(\mathbf{A}, \mathbf{t}, \mathbf{Az} - \mathbf{t}\mathrm{c} \bmod \mathbf{q})$, **return** 0
3: **return** 1

the resulting distribution uniform in a box (as in [Lyu09]), or the more efficient approach in [Lyu12] of making it a discrete Gaussian. The interactive protocol is then converted to a non-interactive one in the random-oracle model [BR93] using the Fiat-Shamir technique [FS86]. This combined technique is sometimes referred to as "Fiat-Shamir with Aborts".

A variation of the signing protocol from [Lyu12] is given in Algorithm 1. It was shown in that work that the output $\mathbf{z}$ is distributed according to $D_{R^k,0,\sigma}$. In particular, the rejection sampling stem on line 4 has the effect that the distribution of $\mathbf{z}$ is independent of the secret vector $\mathbf{s}$ (and the challenge c). This algorithm is therefore honest-verifier zero knowledge since a simulator can simply output $\mathbf{z} \xleftarrow{\$} D_{R^k,0,\sigma}, c \xleftarrow{\$} \mathcal{C}$ and program $\mathrm{c} = \mathsf{H}(\mathbf{A}, \mathbf{t}, \mathbf{Az} - \mathbf{t}\mathrm{c} \bmod \mathbf{q})$. [3] We also make the observation that one does not need to use the same modulus q for every row of the relation in (8). One can instead use a different modulus for each row, and we represent this in the protocol by a vector $\mathbf{q}$ – having different moduli is crucial to the application in this paper.

We also need simulation soundness [Sah99], meaning that an adversary cannot create proofs of incorrect statements, even after seeing simulated proofs of incorrect statements. Faust et al. [FKMV12] showed that Fiat-Shamir proofs are simulation-sound if the underlying three-move protocol is honest-verifier zero-knowledge and has "quasi-unique responses", meaning that an adversary cannot

[3] Because the entropy of $\mathbf{z}$ is high, there is a very low probability that the value for $\mathsf{H}(\mathbf{A}, \mathbf{t}, \mathbf{Az} - \mathbf{t}\mathrm{c} \bmod \mathbf{q})$ was previously assigned.

create two accepting transcripts that are different only in the response value. For our proof system, this translates into finding $\mathbf{z} \neq \mathbf{z}'$ such that $\mathbf{Az} = \mathbf{Az}' \bmod q$ be hard. Finding such $\mathbf{z}, \mathbf{z}'$ would imply that $\mathbf{A}(\mathbf{z} - \mathbf{z}') = 0 \bmod q$ where $\mathbf{A}$ is the matrix in Eq. 7. Thus, there is either a non-zero tuple $(\mathrm{y}_1, \mathrm{y}_2) \in R_q$ with l_∞ norm less than 12σ such that $\mathrm{a}\mathrm{y}_1 + p\mathrm{y}_2 = 0 \bmod q$ or $p\mathrm{y}_1 + \mathrm{y}_2 = 0 \bmod q$. In our applications $p > 12\sigma$ and $12\sigma p + 12\sigma < q$, which implies that the second equality is not possible. Also, for most of the parameter sets (see the table in Sect. 7), $(24\sigma)^2 < q$, and therefore a standard probabilistic argument can be used to show that for all $\mathrm{y}_1, \mathrm{y}_2$ of ℓ_∞ norm less than 12σ,

$$\Pr_{\mathrm{a} \xleftarrow{\$} R_q} [\mathrm{a}\mathrm{y}_1 + p\mathrm{y}_2 = 0 \bmod q] = 2^{-\Omega(n)}.$$

Thus for almost all a, there will not be a short solution $(\mathrm{y}_1, \mathrm{y}_2)$ that satisfies $\mathrm{a}\mathrm{y}_1 + p\mathrm{y}_2 = 0$.

If $(24\sigma)^2 > q$, then the probabilistic argument no longer applies, but then finding such $(\mathrm{y}_1, \mathrm{y}_2)$ gives a solution to Ring-SIS [LM06] problem for a random a, which is a computationally hard problem when the norm of y_i is small-enough with relation to q (which it is in all applications).

3 One-Shot Verifiable Encryption for Linear Relations from Ring-LWE

We follow Camenisch and Shoup [CS03] in defining verifiable encryption as encrypting a witness for a member of a language. The class of languages that we'll be looking at will be the linear relations of short vectors in a ring. While Camenisch and Shoup defined soundness by requiring that decryption of a valid ciphertext always recovers a valid witness, we will only achieve a relaxed property that recovers a witness for a related "extended" language that includes the original language. As we will see in Sect. 6, however, this weaker property suffices for many practical applications of verifiable encryption.

3.1 Definition of Relaxed Verifiable Encryption

We relax Camenisch and Shoup's [CS03] definitions for verifiable encryption in two ways. First, as mentioned above, and analogous to relaxed knowledge extraction for proofs of knowledge [CKY09], the encryption algorithm encrypts a witness w for a member x of a language L, but soundness only guarantees that decryption of a valid ciphertext returns a witness $\bar{w}$ of an extented language $\bar{L}$ instead of L. Second, rather than looking at verifiable encryption as a combination of a standard public-key encryption scheme with an associated proof system, we consider encryption and proof as a single algorithm, producing a verifiable ciphertext that includes the proof. This generalization allows for more efficient schemes, in particular our construction that speeds up decryption using information from the proof.

Let $L \subseteq \{0,1\}^*$ be a language with witness relation R_L, i.e., $x \in L$ iff there exists a witness w such that $(x,w) \in R_L$. Let $\bar{L}$ with witness relation $R_{\bar{L}}$ be an extension of L, meaning that $L \subseteq \bar{L}$ and $R_L \subseteq R_{\bar{L}}$. For our language of linear relations over short vectors, we will consider the language L with relation

$$R_L = \{((\mathbf{B},\mathbf{u}),(\mathbf{m},1)) \in (R_p^{\ell \times k} \times R_p^{\ell}) \times (R_p^k \times R_p) \ : \ \mathbf{Bm} = \mathbf{u} \bmod p \ \wedge \mathbf{m} \in S_\gamma^k\}$$

and the extended language $\bar{L}$ with relation

$$R_{\bar{L}} = \{((\mathbf{B},\mathbf{u}),(\bar{\mathbf{m}},\bar{\mathrm{c}})) \in (R_p^{\ell \times k} \times R_p^{\ell}) \times (R_p^k \times R_p) : \\ \mathbf{B}\bar{\mathbf{m}} = \bar{\mathrm{c}}\mathbf{u} \bmod p \ \wedge \ \|\bar{\mathbf{m}}\|_\infty < 6\sigma \ \wedge \ \bar{\mathrm{c}} \in \bar{\mathcal{C}}\},$$

where $\bar{\mathcal{C}} = \{\mathrm{c} - \mathrm{c}' \ : \ \mathrm{c},\mathrm{c}' \in \mathcal{C}\}$ for $\mathcal{C} = \{\mathrm{c} \in R : \|\mathrm{c}\|_\infty = 1, \|\mathrm{c}\|_1 \leq 36\}$ and other parameters are described in Algorithm 3.

A *relaxed verifiable encryption scheme* for languages $L, \bar{L}$ is a tuple of algorithms $(\mathsf{Kg}, \mathsf{Enc}, \mathsf{V}, \mathsf{Dec})$ where the key generation algorithm $\mathsf{Kg}(1^\lambda)$ returns a public and secret key (pk, sk); the encryption algorithm $\mathsf{Enc}(pk, x, w)$ returns a verifiable ciphertext t that encrypts the witness w of language member $x \in L$; the verification algorithm $\mathsf{V}(pk, x, t)$ returns 1 or 0 indicating whether t encrypts a witness for x; the decryption algorithm $\mathsf{Dec}(sk, x, t)$ returns a witness $\bar{w}$ or a failure symbol $\perp$. We will focus on the case where the ciphertext t includes a Fiat-Shamir proof of a Σ-protocol, where the proof $\pi = (cmt, c, rsp)$ consists of a commitment cmt, a challenge $c = \mathsf{H}(pk, x, t, cmt, \ldots)$ generated through a random oracle H, and a response rsp. We require that the algorithms satisfy the following adapted properties from [CS03]:

Correctness. Correctness requires that $\mathsf{Dec}(sk, x, \mathsf{Enc}(pk, x, w)) = w$ with probability one for all $(x,w) \in R_L$ and all key pairs $(pk, sk) \xleftarrow{\$} \mathsf{Kg}(1^\lambda)$.

Completeness. For all $(x,w) \in R_L$ and all key pairs $(pk, sk) \xleftarrow{\$} \mathsf{Kg}(1^\lambda)$, $\mathsf{V}(pk, x, \mathsf{Enc}(pk, x, w)) = 1$ with probability one.

Soundness. Soundness requires that a ciphertext with a valid proof for $x \in L$ can with overwhelming probability be decrypted to a valid witness $\bar{w}$ such that $(x, \bar{w}) \in R_{\bar{L}}$, i.e., the following probability is negligible:

$$\Pr\left[b = 1 \ \wedge \ (x,\bar{w}) \notin R_{\bar{L}} \ : \ \begin{array}{c}(pk, sk) \xleftarrow{\$} \mathsf{Kg}(1^\lambda), (x,t) \xleftarrow{\$} \mathsf{A}(pk, sk), \\ b \leftarrow \mathsf{V}(pk, x, t), \bar{w} \xleftarrow{\$} \mathsf{Dec}(sk, x, t)\end{array}\right].$$

Simulatability. There exists a simulator Sim such that no adversary A can distinguish real from simulated ciphertexts, i.e., the following advantage of A is negligible:

$$\left|\Pr\left[b' = b \ : \ \begin{array}{c}b \xleftarrow{\$} \{0,1\}, (pk, sk) \xleftarrow{\$} \mathsf{Kg}(1^\lambda), (,,x,w) \xleftarrow{\$} \mathsf{A}(pk), \\ t_0 \xleftarrow{\$} \mathsf{Enc}(pk, x, w), t_1 \xleftarrow{\$} \mathsf{Sim}(pk, x), b' \xleftarrow{\$} \mathsf{A}(, t_b)\end{array}\right] - \frac{1}{2}\right|.$$

3.2 Construction

Given a linear relation

$$\mathbf{Bm} = \mathbf{u} \bmod p, \tag{9}$$

for a matrix $\mathbf{B} \in R_p^{\ell \times k}$, our goal is to produce a ciphertext and a proof that the decryption of this ciphertext is $(\overline{\mathbf{m}}, \overline{\mathrm{c}})$ that satisfies the relation

$$\mathbf{B}\overline{\mathbf{m}} = \mathbf{u}\overline{\mathrm{c}} \bmod p. \tag{10}$$

Key generation. Key pairs are generated as for the Ring-LWE encryption scheme from Sect. 2.5, i.e., by choosing $\mathrm{s}_1, \mathrm{s}_2 \xleftarrow{\$} S_1$ and computing a $\xleftarrow{\$} R$ and $\mathrm{t} \leftarrow \mathrm{as}_1 + \mathrm{s}_2$. The public key is $pk = (\mathrm{a}, \mathrm{t}, p, q)$, where p is the same value as the modulus that we are proving our linear relation over. The secret key is $sk = \mathrm{s}_1$.

Encryption and verification. The prover encrypts a witness $w = \mathbf{m}$ for language member $x = (\mathbf{B}, \mathbf{u})$ satisfying (9) with randomness $(\mathbf{r}, \mathbf{e}, \mathbf{e}') \xleftarrow{\$} S_1^{3k}$ as in (7). The prover then concatenates this with (9) to form the relation below:

$$\begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{e} \\ \mathbf{e}' \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix} \begin{matrix} \bmod q \\ \bmod q \\ \bmod p \end{matrix} \tag{11}$$

As discussed, there is no practical proof of knowledge for the above relation, and so the prover instead uses the "Fiat-Shamir with Aborts" approach from Sect. 2.6 to construct a proof of knowledge π of low-weight $\overline{\mathbf{r}}, \overline{\mathbf{e}}, \overline{\mathbf{e}}', \overline{\mathbf{m}}$, and $\overline{\mathrm{c}}$ that satisfy

$$\begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{r}} \\ \overline{\mathbf{e}} \\ \overline{\mathbf{e}}' \\ \overline{\mathbf{m}} \end{bmatrix} = \overline{\mathrm{c}} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix} \begin{matrix} \bmod q \\ \bmod q \\ \bmod p \end{matrix} \tag{12}$$

This procedure and the corresponding verification is presented in Algorithms 3 and 4.

Decryption. The main result of our work is showing that, when given the ciphertext $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}, \mathbf{z})$, the decryptor can recover some $(\overline{\mathbf{m}}, \overline{\mathrm{c}})$ that satisfies (10). Because the proof of knowledge $(\mathrm{c}, \mathbf{z})$ does not imply that $\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}$ is a valid Ring-LWE ciphertext, we cannot simply use the Ring-LWE decryption algorithm from (4).

Instead, the intuition is to guess a value for $\overline{\mathrm{c}}$ and then attempt to decrypt the ciphertext $\overline{\mathrm{c}} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \bmod q$ in hopes of recovering $\overline{\mathbf{m}}$. The problem with this straightforward approach is that the decryption algorithm will always return something, and so one needs a way to decide whether this decryption is something

Algorithm 3. One-Shot Verifiable encryption $\mathsf{Enc}(pk, x, w)$

Input: Public key $pk = (\mathrm{a}, \mathrm{t}, p, q)$, language member $x = (\mathbf{B}, \mathbf{u})$, witness $w = \mathbf{m} \in S_\gamma^k$. Challenge domain $\mathcal{C} = \{\mathrm{c} \in R : \|\mathrm{c}\|_\infty = 1, \|\mathrm{c}\|_1 \leq 36\}$. Cryptographic hash function $\mathsf{H} : \{0,1\}^* \to \mathcal{C}$. Standard deviation $\sigma = 11 \cdot \max_{\mathrm{c} \in \mathcal{C}} \|\mathrm{c}\|_1 \cdot \sqrt{kn(3+\gamma)}$.

1: $\mathbf{r}, \mathbf{e}, \mathbf{e}' \xleftarrow{\$} S_1^k$

2: $\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \leftarrow \begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{e} \\ \mathbf{e}' \\ \mathbf{m} \end{bmatrix} \bmod q$

3: $\mathbf{y} \leftarrow \begin{bmatrix} \mathbf{y_r} \\ \mathbf{y_e} \\ \mathbf{y_{e'}} \\ \mathbf{y_m} \end{bmatrix} \xleftarrow{\$} D_{R^{4k},0,\sigma}$

4: $\mathrm{c} \leftarrow \mathsf{H}\left(\begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix}, \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix}, \begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{y_r} \\ \mathbf{y_e} \\ \mathbf{y_{e'}} \\ \mathbf{y_m} \end{bmatrix} \begin{matrix} \bmod q \\ \bmod q \\ \bmod p \end{matrix} \right)$

5: $\mathbf{s} \leftarrow \begin{bmatrix} \mathbf{r} \\ \mathbf{e} \\ \mathbf{e}' \\ \mathbf{m} \end{bmatrix} \mathrm{c}$

6: $\mathbf{z} \leftarrow \mathbf{s} + \mathbf{y}$

7: with probability $\frac{D_{R^{4k},0,\sigma}(\mathbf{z})}{3 \cdot D_{R^{4k},\mathbf{s},\sigma}(\mathbf{z})}$, continue, else goto 3

8: **if** $\|\mathbf{z}\|_\infty > 6 \cdot \sigma$, goto 3

9: **return** $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}, \mathbf{z})$

Algorithm 4. One-Shot Verification $\mathsf{V}(pk, x, t)$

Input: Public key $pk = (\mathrm{a}, \mathrm{t}, p, q)$, language member $x = (\mathbf{B}, \mathbf{u})$, ciphertext $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}, \mathbf{z})$. Cryptographic hash function H, positive real σ as in Algorithm 3.

1: **if** $\|\mathbf{z}\|_\infty > 6 \cdot \sigma$, **return** 0

2: **if** $\mathrm{c} \neq \mathsf{H}\left(\begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix}, \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix}, \begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix} \mathbf{z} - \mathrm{c} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix} \begin{matrix} \bmod q \\ \bmod q \\ \bmod p \end{matrix} \right)$, **return** 0

3: **return** 1

valid or just garbage. In Lemma 3.1, we show that if the parameters of the Ring-LWE encryption scheme are set in a particular way, then the decryptor can test whether a particular ciphertext $\overline{\mathrm{c}} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \bmod q$ is "valid", and for any $\overline{\mathrm{c}}$ and $\overline{\mathrm{c}}'$ that lead to valid ciphertexts decrypting to $\overline{\mathbf{m}}$ and $\overline{\mathbf{m}}'$, respectively, we have the equality

$$\overline{\mathbf{m}}/\overline{\mathrm{c}} = \overline{\mathbf{m}}'/\overline{\mathrm{c}}' \bmod p \tag{13}$$

The implication of the above equation is that once the decryptor decrypts some pair $(\overline{\mathbf{m}}', \overline{\mathrm{c}}')$, it is a valid solution to (10). This is because the proof of knowledge π proves knowledge of some $(\overline{\mathbf{m}}, \overline{\mathrm{c}})$ that satisfies $\mathbf{B}\overline{\mathbf{m}} = \overline{\mathrm{c}}\mathbf{u} \bmod p$, or equivalently $\mathbf{B}\overline{\mathbf{m}}/\overline{\mathrm{c}} = \mathbf{u} \bmod p$. Equation (13) then implies that

$$\mathbf{B}\overline{\mathbf{m}}' = \overline{\mathrm{c}}'\mathbf{u} \bmod p.$$

The second issue is how to find a valid $\overline{\mathrm{c}}$. In particular, if we would like the proof of knowledge to be "one-shot", then the challenge space should be exponentially large, and so it is impractical to simply try all the possible $\overline{\mathrm{c}}$ (of which there are actually even more than in the challenge space). We show in Lemma 3.2, however, that the decryptor can try random $\overline{\mathrm{c}}$ (there is some relation between π and which $\overline{\mathrm{c}}$ should be tried), and then the *expected* number of tries is essentially the number of random oracle queries that the prover makes when constructing π, where the probability is taken over the randomness of the random oracle (modeled as a random function) and the coins of the decryptor. Algorithm 5 is the decryption algorithm that guesses a random c' from $\mathcal{C}$, constructs $\overline{\mathrm{c}} = \mathrm{c} - \mathrm{c}'$, where c is part of the proof π, and then checks whether $\overline{\mathrm{c}} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \bmod q$ is a valid ciphertext (actually k valid ciphertexts because the plaintext $\mathbf{m}$ is encrypted as k independent plaintexts). If it is, then it decrypts it, and otherwise it guesses a new random c'.

Algorithm 5. One-Shot Decryption $\mathsf{Dec}(sk, x, t)$

Input: Secret key $sk = \mathrm{s}_1$, language member $x = (\mathbf{B}, \mathbf{u})$, ciphertext $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}, \mathbf{z})$, constant $C = \max_{\mathrm{c},\mathrm{c}' \in \mathcal{C}} \|\mathrm{c} - \mathrm{c}'\|_1$.

if $\mathsf{V}(pk, x, t) = 1$ **then**
  loop
    $\mathrm{c}' \xleftarrow{\$} \mathcal{C}$
    $\overline{\mathrm{c}} \leftarrow \mathrm{c} - \mathrm{c}'$
    $\overline{\mathbf{m}} \leftarrow (\mathbf{w} - \mathbf{v}\mathrm{s}_1)\overline{\mathrm{c}} \bmod q$
    if $\|\overline{\mathbf{m}}\|_\infty < q/2C$ **then**
      $\overline{\mathbf{m}} \leftarrow \overline{\mathbf{m}} \bmod p$
      return $(\overline{\mathbf{m}}, \overline{\mathrm{c}})$
    end if
  end loop
end if

If the prover is honest, then of course $\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}$ will already be a valid ciphertext, and then it's not hard to see that any $\overline{\mathrm{c}}$ will result in a valid decryption (or the decryptor can try $\overline{\mathrm{c}} = 1$ first). On the other hand, what Lemma 3.2 roughly implies is that if the prover can only query the random oracle a few times, then the decryptor will also expect to recover a solution to (10) within a few queries.

In Sect. 4, we propose a modification to our protocol that still retains the relation between the number of random-oracle queries the prover makes and the expected number of decryption tries that the decryptor needs, but also puts an upper-bound on the decryption time. The idea is to reduce the space of challenges in the zero-knowledge proof from, say, 2^{128} down to 2^{32} and then doing the proof in parallel 4 times. This increases the proof size by a factor of 4, but upperbounds the number of decryptions tries to $4 \cdot 2^{32}$. One can of course adjust the trade-off between the decryption-time upper bound and the size of the proof to suit the particular scenario. This is the main advantage of this parallelized protocol over the earlier idea in [BCK+14]. In the latter scheme, the size of the challenge space could not be varied – if one were working over the ring $\mathbb{Z}[\mathrm{x}]/(\mathrm{x}^n + 1)$, then the challenge space was exactly $2n + 1$.

3.3 Interlude: Proofs of Plaintext Knowledge

One can see proofs of plaintext knowledge as a verifiable encryption scheme without a relation, or where the relation is trivially satisfied. In our case, one could consider the scheme from the previous section with $\mathbf{B}$ and $\mathbf{u}$ being 0, or simply the row(s) containing $\mathbf{B}$ and $\mathbf{u}$ not being present in relation (11).

The soundness requirement that a valid ciphertext must decrypt to a valid witness obviously makes no sense if the relation is trivial. Instead, soundness for a proof of plaintext knowledge requires that decryption returns the same value as can be extracted from the proof of knowledge. Our randomized decryption algorithm as described in Algorithm 5 does not satisfy such a requirement, as it potentially returns a different pair $(\overline{\mathbf{m}}, \overline{\mathrm{c}})$ at each execution. However, because of the property that $\overline{\mathbf{m}}/\overline{\mathrm{c}} = \overline{\mathbf{m}}'/\overline{\mathrm{c}}' \bmod p$ for any $(\overline{\mathbf{m}}, \overline{\mathrm{c}}), (\overline{\mathbf{m}}', \overline{\mathrm{c}}')$ returned by the decryption algorithm, we can make the decryption deterministic by letting it return $\overline{\mathbf{m}}/\overline{\mathrm{c}} \bmod p$. Because this unique value can also be extracted from the proof, this turns our verifiable encryption scheme into a proof of plaintext knowledge.

3.4 Correctness and Security

Soundness. We first show the soundness property of our relaxed verifiable encryption scheme by showing that decryption of a valid ciphertext, if it finishes, yields a witness from $R_{\bar{L}}$. In Sect. 3.5, we prove that the expected running time of the decryption algorithm is proportional to the number of random-oracle queries made by the adversary who created the ciphertext.

If an adversary $\mathcal{A}$ who is trying to break the soundness of the scheme outputs a ciphertext $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}, \mathbf{z})$ that is valid for $x = (\mathbf{B}, \mathbf{u})$, then by the verification procedure described in Algorithm 4 we have that $\|\mathbf{z}\|_\infty \leq 6 \cdot \sigma$ and

$$\mathrm{c} = \mathsf{H}\left(\mathbf{B}', \begin{bmatrix}\mathbf{v}\\ \mathbf{w}\\ \mathbf{u}\end{bmatrix}, \mathbf{B}'\mathbf{z} - \mathrm{c}\begin{bmatrix}\mathbf{v}\\ \mathbf{w}\\ \mathbf{u}\end{bmatrix} \begin{matrix}\bmod q\\ \bmod q\\ \bmod p\end{matrix}\right) \tag{14}$$

where

$$\mathbf{B}' = \begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix}. \tag{15}$$

Let $\mathbf{A}$ denote the first argument of the above random-oracle query and $\mathbf{y}$ the last, i.e., the above equation can be rewritten as $\mathrm{c} = \mathsf{H}(\mathbf{A}, [\ \mathbf{v}\ ;\ \mathbf{w}\ ;\ \mathbf{u}\ ], \mathbf{y})$.

With overwhelming probability, there exists a second challenge $\mathrm{c}' \in \mathcal{C}\setminus\{\mathrm{c}\}$ for which there exists a vector $\mathbf{z}'$ with $\|\mathbf{z}'\|_\infty \leq 6\cdot\sigma$ and $\mathbf{y} = \mathbf{A}\mathbf{z}' - \mathrm{c}'[\ \mathbf{v}\ ;\ \mathbf{w}\ ;\ \mathbf{u}\ ]$. Indeed, if c were the only such challenge, then at the moment of making the above random-oracle query, A would have had probability $1/|\mathcal{C}|$ of hitting the only challenge c for which a valid proof exists. The probability that A outputs a proof for which only one such challenge c exists is therefore at most $q_{\mathsf{H}}/|\mathcal{C}|$.

So with overwhelming probability such $\mathrm{c}', \mathbf{z}'$ does exist, and we have that $\mathbf{y} = \mathbf{A}\mathbf{z} - \mathrm{c}[\ \mathbf{v}\ ;\ \mathbf{w}\ ;\ \mathbf{u}\ ] = \mathbf{A}\mathbf{z}' - \mathrm{c}'[\ \mathbf{v}\ ;\ \mathbf{w}\ ;\ \mathbf{u}\ ]$ with $\|\mathbf{z}\|_\infty \leq 6\cdot\sigma$ and $\|\mathbf{z}'\|_\infty \leq 6\cdot\sigma$. Hence, letting $\bar{\mathrm{c}} = \mathrm{c} - \mathrm{c}'$ and $\bar{\mathbf{z}} = \mathbf{z}' - \mathbf{z} = [\ \bar{\mathbf{r}}\ ;\ \bar{\mathbf{e}}\ ;\ \bar{\mathbf{e}}'\ ;\ \bar{\mathbf{m}}\ ]$, we have that $\mathbf{A}\bar{\mathbf{z}} = \bar{\mathrm{c}}[\ \mathbf{v}\ ;\ \mathbf{w}\ ;\ \mathbf{u}\ ]$ with $\|\bar{\mathbf{z}}\|_\infty \leq 12\cdot\sigma$.

By choosing the scheme parameters appropriately, e.g., such that $(36p + 12)\sigma < q/2C$, one can satisfy the preconditions of the following crucial lemma that shows that for any $(\bar{\mathbf{m}}', \bar{\mathrm{c}}')$ returned by the decryption algorithm, we have that $\bar{\mathbf{m}}'/\bar{\mathrm{c}}' = \bar{\mathbf{m}}/\bar{\mathrm{c}}$, and, because $\mathbf{B}\bar{\mathbf{m}} = \bar{\mathrm{c}}\mathbf{u}$, that $\mathbf{B}\bar{\mathbf{m}}' = \bar{\mathrm{c}}'\mathbf{u}$.

Lemma 3.1. *Let* $(\mathrm{a}, \mathrm{t}, p, q)$ *and* $(\mathrm{s}_1, \mathrm{s}_2)$ *be generated keys as in Sect. 3.2. If for given* $\mathrm{v}, \mathrm{w} \in R_q$ *there exist* $\overline{\mathrm{r}}, \overline{\mathrm{e}}, \overline{\mathrm{e}}', \overline{\mathrm{m}}, \overline{\mathrm{c}}$ *such that*

$$\begin{bmatrix} p\mathrm{a} & p & 0 & 0 \\ p\mathrm{t} & 0 & p & 1 \end{bmatrix} \begin{bmatrix} \overline{\mathrm{r}} \\ \overline{\mathrm{e}} \\ \overline{\mathrm{e}}' \\ \overline{\mathrm{m}} \end{bmatrix} = \overline{\mathrm{c}} \begin{bmatrix} \mathrm{v} \\ \mathrm{w} \end{bmatrix} \bmod q$$

and

$$\|p(\overline{\mathrm{r}}\mathrm{s}_2 + \overline{\mathrm{e}}' - \overline{\mathrm{e}}\mathrm{s}_1) + \overline{\mathrm{m}}\|_\infty < q/2C \tag{16}$$

where $C = \max_{\overline{\mathrm{c}}\in\overline{\mathcal{C}}} \|\overline{\mathrm{c}}\|_1 = \max_{\mathrm{c},\mathrm{c}'\in\mathcal{C}} \|\mathrm{c} - \mathrm{c}'\|_1$, *then*

1. $\|(\mathrm{w} - \mathrm{v}\mathrm{s}_1)\overline{\mathrm{c}} \bmod q\|_\infty < q/2C$
2. *For any* $\overline{\mathrm{c}}' \in \overline{\mathcal{C}}$ *for which* $\|(\mathrm{w} - \mathrm{v}\mathrm{s}_1)\overline{\mathrm{c}}' \bmod q\|_\infty < q/2C$,

$$(\mathrm{w} - \mathrm{v}\mathrm{s}_1)\overline{\mathrm{c}}' \bmod q / \overline{\mathrm{c}}' \bmod p = \overline{\mathrm{m}}/\overline{\mathrm{c}} \bmod p.$$

□

Proof. To prove the first part, we note that by the definition of Ring-LWE decryption,

$$(\mathrm{w} - \mathrm{v}\mathrm{s}_1)\overline{\mathrm{c}} \bmod q = p(\overline{\mathrm{r}}\mathrm{s}_2 + \overline{\mathrm{e}}' - \overline{\mathrm{e}}\mathrm{s}_1) + \overline{\mathrm{m}},$$

which has ℓ_∞ length less than $\frac{q}{2C}$ by the hypothesis of the lemma.

To prove the second part, we first note that

$$\begin{aligned}(\mathrm{w} - \mathrm{vs}_1)\overline{\mathrm{cc}}' \bmod q \bmod p &= (p(\bar{\mathrm{r}}\mathrm{s}_2 + \bar{\mathrm{e}}' - \bar{\mathrm{e}}\mathrm{s}_1) + \overline{\mathrm{m}})\bar{\mathrm{c}}' \bmod q \bmod p \\ &= \overline{\mathrm{mc}}' \bmod p. \end{aligned} \tag{17}$$

Then we can write

$$\begin{aligned}((\mathrm{w} - \mathrm{vs}_1)\bar{\mathrm{c}}' \bmod q) \,/\, \bar{\mathrm{c}}' \bmod p &= ((\mathrm{w} - \mathrm{vs}_1)\bar{\mathrm{c}}' \bmod q) \cdot \bar{\mathrm{c}}/(\overline{\mathrm{cc}}') \bmod p \\ &= ((\mathrm{w} - \mathrm{vs}_1)\overline{\mathrm{cc}}' \bmod q)/(\overline{\mathrm{cc}}') \bmod p \\ &= \overline{\mathrm{mc}}'/(\overline{\mathrm{cc}}') \bmod p = \overline{\mathrm{m}}/\bar{\mathrm{c}} \bmod p \end{aligned}$$

The first equality is an identity, the second equality holds since $\|(\mathrm{w} - \mathrm{vs}_1)\bar{\mathrm{c}}' \bmod q\|_\infty < \frac{q}{2C}$ and therefore multiplication by $\bar{\mathrm{c}}$ does not cause a reduction modulo q. The third equality follows from (17). □

By checking that $\|(\mathrm{w} - \mathrm{vs}_1)\bar{\mathrm{c}} \bmod q\|_\infty < q/2C$ in line 6 of the Dec algorithm, we ensure that the condition of the second part of Lemma 3.1 is satisfied for decryption, so that the value $(\bar{\mathbf{m}}'(\mathbf{w} - \mathbf{v}\mathrm{s}_1)\bar{\mathrm{c}}' \bmod q \bmod p, \bar{\mathrm{c}}')$ is indeed a witness for $(\mathbf{B}, \mathbf{u}) \in \bar{L}$. This proves the soundness of our scheme.

Correctness. Correctness is straightforward because a valid encryption (see (5)) satisfies the preconditions of Lemma 3.1 with $[\ \bar{\mathbf{r}}\ ;\ \bar{\mathbf{e}}\ ;\ \bar{\mathbf{e}}'\ ;\ \bar{\mathbf{m}}\ ] = [\ \mathbf{r}\ ;\ \mathbf{e}\ ;\ \mathbf{e}'\ ;\ \mathbf{m}\ ]$ and $\bar{\mathrm{c}} = 1$; and it's clear that $\|p(\mathbf{r}\mathrm{s}_2 + \mathbf{e}' - \mathbf{e}\mathrm{s}_1) + \mathbf{m}\|_\infty \leq \|p(\bar{\mathbf{r}}\mathrm{s}_2 + \bar{\mathbf{e}}' - \bar{\mathbf{e}}\mathrm{s}_1) + \bar{\mathbf{m}}\|_\infty$.

Completeness. Completeness follows from the completeness of the proof system of Sect. 2.6.

Simulatability. The simulator Sim creates a Ring-LWE encryption $[\ \mathbf{v}\ ;\ \mathbf{w}\ ]$ of $\mathbf{m} = 1$ using the scheme of Sect. 2.5 and runs the zero-knowledge simulator for the proof system of Sect. 2.6 to create a valid-looking proof $(\mathrm{c}, \mathbf{z})$ for $(\mathbf{B}, \mathbf{u})$. The indistinguishability from the real proof follows from the IND-CPA security of Ring-LWE encryption and the zero-knowledge property of the proof system.

3.5 Decryption Running Time

Even though the running time of the decryption algorithm is unbounded in principle, we show that its *expected* running time is proportional to the number of times that the adversary queries the random oracle. More precisely, we show that if an adversary uses q_H random-oracle queries to construct a ciphertext, then the probability that the decryption algorithm requires more than $\alpha \cdot q_\mathsf{H}$ iterations is less than $1/\alpha$.

We prove the above information-theoretic statement for any adversary $\mathcal{A}$, and we can therefore limit the analysis to deterministic adversaries, since the coins that maximize the adversary's success can always be hardwired into its code.

Lemma 3.2. *For a given key pair* $(pk, sk) \in \mathsf{Kg}(1^\lambda)$*, consider the following experiment with an adversary* $\mathcal{A}$*:*

$(x, t) \xleftarrow{\$} \mathcal{A}^{\mathsf{H}}(pk)$
If $\mathsf{V}(pk, x, t) = 1$ *then* $\bar{w} \xleftarrow{\$} \mathsf{Dec}(sk, x, t)$.

Let $\widehat{H}$ *be the random coins of the function* H *(when it is modeled as a random function) and* $\widehat{D}$ *be the random coins of the Decryption algorithm* Dec*. Let* T *be the number of loop iterations in the execution of* Dec *(see Algorithm 5 lines 2–10) until it produces its output* $\bar{w}$*. Then there is an event* G*, such that for all algorithms* $\mathcal{A}$ *that make at most* $q_{\mathsf{H}} - 1$ *queries to* H*, all key pairs* (pk, sk)*, and any positive integer* f*, it holds that*

1. $\mathrm{Exp}_{\widehat{H},\widehat{D}}[\ T \mid G\ ] \leq \left(1 + \frac{1}{f}\right) \cdot q_{\mathsf{H}}$.
2. $\Pr_{\widehat{H},\widehat{D}}[\neg G] \leq q_{\mathsf{H}} \cdot f/|\mathcal{C}|$

By Markov's inequality and optimization over f*, this implies that for any positive* α*,*

$$\Pr_{\widehat{H},\widehat{D}}[\ T \geq \alpha \cdot q_{\mathsf{H}}] \leq \frac{1}{\alpha} + 2 \cdot \sqrt{\frac{q_{\mathsf{H}}}{\alpha \cdot |\mathcal{C}|}} + \frac{q_{\mathsf{H}}}{|\mathcal{C}|}.$$

Proof. For a given public key pk, language member $x = (\mathbf{B}, \mathbf{u})$, and valid ciphertext $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}, \mathbf{z})$, let $\mathbf{A}$ and $\mathbf{y}$ be the matrix and vector in the verification algorithm (Algorithm 4) so that $c = \mathsf{H}(\mathbf{A}, [\ \mathbf{v}\ ;\ \mathbf{w}\ ;\ \mathbf{u}\ ], \mathbf{y})$ and $\mathbf{y} = \mathbf{A}\mathbf{z} - \mathrm{c}[\ \mathbf{v}\ ;\ \mathbf{w}\ ;\ \mathbf{u}\ ]$. Let $\mathcal{G}_t$ be the set of "good" challenges c' for which a valid zero-knowledge proof response $\mathbf{z}'$ exists, i.e.,

$$\mathcal{G}_t \ = \ \{\mathrm{c}' \in \mathcal{C} \ : \ \exists\, \mathbf{z}' : \mathbf{y} = \mathbf{A}\mathbf{z}' - \mathrm{c}'[\ \mathbf{v}\ ;\ \mathbf{w}\ ;\ \mathbf{u}\ ] \wedge \|\mathbf{z}'\|_\infty \leq 6\sigma\}\ .$$

Let G be the "good" event that the adversary $\mathcal{A}$ produces a ciphertext t (for the decryption algorithm) with $|\mathcal{G}_t| > f$. Let $\widehat{D}$ denote the coins of the decryption algorithm Dec and let $\widehat{H}$ denote the coins determining the random oracle H. For any ciphertext t, the probability over $\widehat{D}$ that one particular iteration of Dec decrypts successfully, i.e., hits a good challenge $c' \in \mathcal{G}_t \setminus \{c\}$, is $\frac{|\mathcal{G}_t|-1}{|\mathcal{C}|}$. We therefore have that the expected number of iterations for a ciphertext t is

$$\mathrm{Exp}_{\widehat{D}}[\ T \mid \mathcal{A}^{\mathsf{H}} \text{ outputs } t\ ] = \frac{|\mathcal{C}|}{|\mathcal{G}_t| - 1}$$

and therefore, conditioned on the event G, that

$$\mathrm{Exp}_{\widehat{D}}[\ T \mid \mathcal{A}^{\mathsf{H}} \text{ outputs } t \ \wedge\ G\ ] \leq \frac{|\mathcal{C}|}{f}\ . \tag{18}$$

Below, when we say that "$\mathcal{A}^{\mathsf{H}}$ outputs t_i", we mean that $\mathcal{A}^{\mathsf{H}}$ outputs a language member $x = (\mathbf{B}, \mathbf{u})$ and a ciphertext $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}, \mathbf{z})$ such that $\mathcal{A}$'s i-th random-oracle query is

$$\mathrm{c} = \mathsf{H}\left(\mathbf{B}', \begin{bmatrix}\mathbf{v}\\ \mathbf{w}\\ \mathbf{u}\end{bmatrix}, \mathbf{B}'\mathbf{z} - \mathrm{c}\begin{bmatrix}\mathbf{v}\\ \mathbf{w}\\ \mathbf{u}\end{bmatrix} \begin{matrix}\bmod q\\ \bmod q\\ \bmod p\end{matrix}\right),$$

where $\mathbf{B}'$ is the matrix defined in (15). Also, for any adversary making at most $q_{\mathsf{H}} - 1$ queries, there exists an adversary making at most q_{H} queries that include the above query; we consider the latter adversary $\mathcal{A}$ in the rest of the analysis.

Because we are conditioning on the event G, we can assume without loss of generality that $\mathcal{A}$ only makes random-oracle queries for ciphertexts t_i with $|\mathcal{G}_{t_i}| > f$. (It is easy to see that for any $\mathcal{A}$ that does not obey these rules, there exists an adversary $\mathcal{A}'$ producing the same output that does.) We now have that

$$\operatorname*{Exp}_{\widehat{H},\widehat{D}}[\ T \mid G\ ] = \sum_{i=1}^{q_{\mathsf{H}}} \Pr_{\widehat{H}}[\mathcal{A}^{\mathsf{H}} \text{ outputs } t_i \mid G\ ] \cdot \operatorname*{Exp}_{\widehat{D}}[\ T \mid \mathcal{A}^{\mathsf{H}} \text{ outputs } t_i \ \wedge\ G\ ] \quad (19)$$

The above is true because

$$\begin{aligned}
&\sum_{i=1}^{q_{\mathsf{H}}} \Pr_{\widehat{H}}[\mathcal{A}^{\mathsf{H}} \text{ outputs } t_i \mid G\ ] \cdot \operatorname*{Exp}_{\widehat{D}}[\ T \mid \mathcal{A}^{\mathsf{H}} \text{ outputs } t_i \ \wedge\ G\ ] \\
&\quad = \sum_{i=1}^{q_{\mathsf{H}}} \Pr_{\widehat{H}}[\mathcal{A}^{\mathsf{H}} \text{ outputs } t_i \mid G\ ] \cdot \sum_{j \in \mathbb{Z}^+} \Pr_{\widehat{D}}[\ T = j \mid \mathcal{A}^{\mathsf{H}} \text{ outputs } t_i \ \wedge\ G\ ] \cdot j \\
&\quad = \sum_{j \in \mathbb{Z}^+} j \cdot \sum_{i=1}^{q_{\mathsf{H}}} \Pr_{\widehat{H},\widehat{D}}[\ T = j \ \wedge \mathcal{A}^{\mathsf{H}} \text{ outputs } t_i \mid G\ ] \\
&\quad = \sum_{j \in \mathbb{Z}^+} j \cdot \Pr_{\widehat{D},\widehat{H}}[\ T = j \mid G\ ] \\
&\quad = \operatorname*{Exp}_{\widehat{H},\widehat{D}}[\ T \mid G\ ]
\end{aligned}$$

For each random-oracle query that $\mathcal{A}$ makes for a ciphertext t_i (all the ciphertexts need not be distinct), the probability that $\mathcal{A}$ can output t_i (over the randomness $\widehat{H}$) is at most the probability that the output of the random-oracle query is in $\mathcal{G}_{t_i}$, because otherwise no valid response $\mathbf{z}$ exists. Thus each t_i has the probability of being output at most $|\mathcal{G}_{t_i}|/|\mathcal{C}|$, regardless of the strategy of the adversary. Plugging this and (18) into (19), we obtain

$$\operatorname*{Exp}_{\widehat{H},\widehat{D}}[\ T \mid G\ ] \le \sum_{i=1}^{q_{\mathsf{H}}} \frac{|\mathcal{G}_{t_i}|}{|\mathcal{C}|} \cdot \frac{|\mathcal{C}|}{|\mathcal{G}_{t_i}| - 1} \le q_{\mathsf{H}} \cdot \max_{i=1,\ldots,q_{\mathsf{H}}} \left(\frac{|\mathcal{G}_{t_i}|}{|\mathcal{G}_{t_i}| - 1}\right) \le \frac{q_{\mathsf{H}} \cdot (f+1)}{f}. \quad (20)$$

By applying Markov's inequality, we have that for any positive β,

$$\begin{aligned}\Pr_{\widehat{H},\widehat{D}}[T \geq \beta] &= \Pr_{\widehat{H},\widehat{D}}[T \geq \beta \mid G\,] \cdot \Pr_{\widehat{H},\widehat{D}}[G] + \Pr_{\widehat{H},\widehat{D}}[T \geq \beta \mid \neg G] \cdot \Pr_{\widehat{H},\widehat{D}}[\neg G] \\ &\leq \frac{\mathrm{Exp}_{\widehat{H},\widehat{D}}[T \mid G\,]}{\beta} + \Pr_{\widehat{H},\widehat{D}}[\neg G] \end{aligned} \tag{21}$$

It is furthermore easy to see that

$$\Pr_{\widehat{H},\widehat{D}}[\neg G] = 1 - \Pr_{\widehat{H}}[G] \leq 1 - \left(\frac{|\mathcal{C}| - f}{|\mathcal{C}|}\right)^{q_{\mathsf{H}}} \leq \frac{q_{\mathsf{H}} f}{|\mathcal{C}|}.$$

Plugging this and (20) into (21) and letting $\beta = \alpha \cdot q_{\mathsf{H}}$ yields

$$\Pr_{\widehat{H},\widehat{D}}[\ T \geq \alpha \cdot q_{\mathsf{H}}] \leq \frac{1}{\alpha} \cdot \left(1 + \frac{1}{f}\right) + \frac{q_{\mathsf{H}} f}{|\mathcal{C}|}.$$

To minimize this expression, we set $f = \left\lceil \sqrt{\frac{|\mathcal{C}|}{\alpha q_{\mathsf{H}}}} \right\rceil$, which gives us the claim in the Lemma. □

4 Multi-shot Verifiable Encryption – Construction with a Bounded Decryption Time

If one would like to put a limit on how much computational time decryption takes in the worst case, then the idea is to reduce the size of the challenge space and repeat the proof-of-knowledge protocol from Algorithm 3 in parallel α times using the standard approach. This protocol is described in Algorithm 6. The verification procedure simply checks if all the α copies are verified correctly.

One can show that with probability approximately $1 - |\mathcal{C}|^{-\alpha}$, there will be at least one $\overline{\mathrm{c}}$ for which $\overline{\mathrm{c}} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}$ is a valid ciphertext satisfying Lemma 3.1, and so one simply needs to find it (and check its validity) analogously to the way $\overline{\mathrm{c}} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}$ was found in Algorithm 5. The main difference is that the challenge space is no longer exponentially large, and so one can search through all the $\overline{\mathrm{c}} = \mathrm{c}^{(i)} - \mathrm{c}'$ in time $\alpha \cdot |\mathcal{C}|$. This procedure is described in Algorithm 8. If one wants to also maintain the relationship between the number of random oracle queries of the prover to the number of decryption tries, one could "dovetail" between Algorithm 8 which systematically goes through all $\mathrm{c}' \in \mathcal{C}$ with one that randomly guesses them at random.

The trade-off between having the decryption running time be upper-bounded by $\alpha \cdot |\mathcal{C}|$ is that the proof of knowledge is now approximately α times larger.

Algorithm 6. α-shot Verifiable encryption $\mathsf{Enc}(pk, x, w)$

Input: Public key $pk = (\mathrm{a}, \mathrm{t}, p, q)$, language member $x = (\mathbf{B}, \mathbf{u})$, witness $w = \mathbf{m} \in S_\gamma^k$. Challenge domain $\mathcal{C}$. Cryptographic hash function $\mathsf{H} : \{0,1\}^* \rightarrow \mathcal{C}^\alpha$. Standard deviation $\sigma = 11 \cdot \max_{\mathrm{c} \in \mathcal{C}} \|\mathrm{c}\|_1 \cdot \sqrt{\alpha k n (3+\gamma)}$

1: $\mathbf{r}, \mathbf{e}, \mathbf{e}' \xleftarrow{\$} S_1^k$

2: $\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \leftarrow \begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ \mathrm{t}\mathbf{I} & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{e} \\ \mathbf{e}' \\ \mathbf{m} \end{bmatrix} \bmod q$

3: **for** $i = 1$ **to** α **do**

4: $\quad \mathbf{y}^{(i)} \leftarrow \begin{bmatrix} \mathbf{y_r}^{(i)} \\ \mathbf{y_e}^{(i)} \\ \mathbf{y_{e'}}^{(i)} \\ \mathbf{y_m}^{(i)} \end{bmatrix} \xleftarrow{\$} D_{R^{4k},0,\sigma}$

5: $\quad \mathbf{f}^{(i)} \leftarrow \begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ \mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{y_r}^{(i)} \\ \mathbf{y_e}^{(i)} \\ \mathbf{y_{e'}}^{(i)} \\ \mathbf{y_m}^{(i)} \end{bmatrix} \begin{matrix} \bmod q \\ \bmod q \\ \bmod p \end{matrix}$

6: **end for**

7: $(\mathrm{c}^{(1)}, \ldots, \mathrm{c}^{(\alpha)}) \leftarrow \mathsf{H}\left(\begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ \mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix}, \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(\alpha)} \right)$

8: **for** $i = 1$ **to** ℓ **do**

9: $\quad \mathbf{s}^{(i)} \leftarrow \begin{bmatrix} \mathbf{r} \\ \mathbf{e} \\ \mathbf{e}' \\ \mathbf{m} \end{bmatrix} \mathrm{c}^{(i)}$

10: $\quad \mathbf{z}^{(i)} \leftarrow \mathbf{s}^{(i)} + \mathbf{y}^{(i)}$

11: **end for**

12: $\mathbf{s} \leftarrow \begin{bmatrix} \mathbf{s}^{(1)} \\ \ldots \\ \mathbf{s}^{(\alpha)} \end{bmatrix}, \mathbf{z} \leftarrow \begin{bmatrix} \mathbf{z}^{(1)} \\ \ldots \\ \mathbf{z}^{(\alpha)} \end{bmatrix}$

13: with probability $\frac{D_{R^{4k\alpha},0,\sigma}(\mathbf{z})}{3 \cdot D_{R^{4k\alpha},\mathbf{s},\sigma}(\mathbf{z})}$, continue, else goto 3

14: **if** $\|\mathbf{z}\|_\infty > 6 \cdot \sigma$, goto 3

15: **return** $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}^{(1)}, \ldots, \mathrm{c}^{(\alpha)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\alpha)})$

5 Chosen-Ciphertext Security

Many applications require a verifiable ciphertext to hide the encrypted witness, even when the adversary has access to decryptions of other ciphertexts. As a natural analog of indistinguishability under chosen-ciphertext attack (IND-CCA) for standard public-key encryption schemes, we define chosen-ciphertext simulatability and describe a construction that satisfies it.

Algorithm 7. α-shot Verification $\mathsf{V}(pk, x, t)$

Input: Public key $pk = (\mathrm{a}, \mathrm{t}, p, q)$, language member $x = (\mathbf{B}, \mathbf{u})$, ciphertext $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}^{(1)}, \ldots, \mathrm{c}^{(\alpha)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\alpha)})$. Cryptographic hash function $\mathsf{H} : \{0,1\}^* \to \mathcal{C}^\alpha$. Positive real σ as in Algorithm 6.

1: $\mathbf{z} \leftarrow \begin{bmatrix} \mathbf{z}^{(1)} \\ \cdots \\ \mathbf{z}^{(\alpha)} \end{bmatrix}$

2: **if** $\|\mathbf{z}\|_\infty > 6 \cdot \sigma$, **return** 0

3: **for** $i = 1$ **to** α **do**

4: $\quad \mathbf{f}^{(i)} \leftarrow \begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ \mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix} \mathbf{z}^{(i)} - \mathrm{c}^{(i)} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix} \begin{matrix} \bmod q \\ \bmod q \\ \bmod p \end{matrix}$

5: **end for**

6: **if** $(\mathrm{c}^{(1)}, \ldots, \mathrm{c}^{(\alpha)}) \neq \mathsf{H}\left(\begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix}, \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(\alpha)} \right)$, **return** 0

7: **return** 1

Algorithm 8. α-shot Decryption $\mathsf{Dec}(sk, x, t)$

Input: Secret key $sk = \mathrm{s}_1$, language member $x = (\mathbf{B}, \mathbf{u})$, ciphertext $t = (\mathbf{v}, \mathbf{w}, \mathrm{c}^{(1)}, \ldots, \mathrm{c}^{(\alpha)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\alpha)})$.

if $\mathsf{V}(pk, t, \pi) = 1$ **then**
  for $i = 1$ **to** α **do**
    for all $c' \in \mathcal{C}$ **do**
      $\overline{c} \leftarrow c^{(i)} - c'$
      $\overline{\mathbf{m}} \leftarrow (\mathbf{w} - \mathbf{v}\mathrm{s}_1)\overline{\mathrm{c}} \bmod q$
      if $\|\overline{\mathbf{m}}\|_\infty < q/2C$ **then**
        $\overline{\mathbf{m}} \leftarrow \overline{\mathbf{m}} \bmod p$
        return $(\overline{\mathbf{m}}, \overline{\mathrm{c}})$
      end if
    end for
  end for
end if

Our construction essentially follows the Naor-Yung paradigm [NY90] where the sender encrypts the message twice under different public keys and adds a non-interactive zero-knowledge (NIZK) proof that both ciphertexts encrypt the same message. Naor and Yung only proved their approach secure under non-adaptive chosen-ciphertext (CCA1), but Sahai [Sah99] later showed that if the NIZK proof is simulation-sound, then the resulting encryption scheme is secure against adaptive chosen-ciphertext (CCA2) attacks. Faust et al. [FKMV12] showed that Fiat-Shamir proofs are simulation-sound in the random-oracle model if the underlying proof system has quasi-unique responses.

Furthermore, because the verifiable encryption scheme for a CPA-secure encryption scheme already includes a NIZK, this conversion from CPA to CCA2

security is rather cheap, increasing the size of the proof and ciphertext by factors less than 2 (see (22)).

Chosen-ciphertext simulatability. We say that a relaxed verifiable encryption scheme $(\mathsf{Kg}, \mathsf{Enc}, \mathsf{V}, \mathsf{Dec})$ is chosen-ciphertext simulatable when there exists a simulator Sim such that the following advantage is negligible for all PPT adversaries A:

$$\left| \Pr \left[b' = b \ : \ \begin{array}{c} b \xleftarrow{\$} \{0,1\}, (pk, sk) \xleftarrow{\$} \mathsf{Kg}(1^\lambda), (st, x, w) \xleftarrow{\$} \mathsf{A}(pk), \\ t_0 \xleftarrow{\$} \mathsf{Enc}(pk, x, w), t_1 \xleftarrow{\$} \mathsf{Sim}(pk, x), b' \xleftarrow{\$} \mathsf{A}^{\mathsf{Dec}(sk,\cdot,\cdot)}(st, t_b) \end{array} \right] - \frac{1}{2} \right| ,$$

where A is not allowed to query its Dec oracle on the challenge ciphertext t_b. In the random-oracle model, Sim can additionally program the random oracle.

Construction. The receiver generates a Ring-LWE key pair by choosing the secrets $\mathrm{s}_1, \mathrm{s}'_1, \mathrm{s}_2, \mathrm{s}'_2 \xleftarrow{\$} S_1$ and a $\mathrm{a} \xleftarrow{\$} R$, and computing $\mathrm{t}_1 \leftarrow \mathrm{as}_1 + \mathrm{s}_2$ and $\mathrm{t}_1 \leftarrow \mathrm{as}'_1 + \mathrm{s}'_2$. The public key is $pk = (\mathrm{a}, \mathrm{t}_1, \mathrm{t}_2, p, q)$, where p is modulus for proving the linear relation. The secret key is $sk = \mathrm{s}_1$.

The sender encrypts a witness $w = \mathbf{m}$ for language member $x = (\mathbf{B}, \mathbf{u})$ by choosing randomness $(\mathbf{r}_1, \mathbf{e}_1, \mathbf{e}'_1, \mathbf{r}_2, \mathbf{e}_2, \mathbf{e}'_2) \xleftarrow{\$} S_1^{6k}$, computing

$$\begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}_1\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & \mathbf{I}_k \\ 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & p\mathrm{t}_2\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{e}_1 \\ \mathbf{e}'_1 \\ \mathbf{r}_2 \\ \mathbf{e}_2 \\ \mathbf{e}'_2 \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{w}_1 \\ \mathbf{v}_2 \\ \mathbf{w}_2 \end{bmatrix} \begin{array}{l} \bmod q \\ \bmod q \\ \bmod q \\ \bmod q \end{array} \tag{22}$$

and concatenating a proof $(\mathrm{c}, \mathbf{z})$ using the relaxed NIZK proof system of Sect. 2.6 for the language element:

$$\begin{bmatrix} p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & 0^{k\times k} \\ p\mathrm{t}_1\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & \mathbf{I}_k \\ 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & p\mathrm{a}\mathbf{I}_k & p\mathbf{I}_k & 0^{k\times k} & 0^{k\times k} \\ 0^{k\times k} & 0^{k\times k} & 0^{k\times k} & p\mathrm{t}_2\mathbf{I}_k & 0^{k\times k} & p\mathbf{I}_k & \mathbf{I}_k \\ 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & 0^{\ell\times k} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{e}_1 \\ \mathbf{e}'_1 \\ \mathbf{r}_2 \\ \mathbf{e}_2 \\ \mathbf{e}'_2 \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{w}_1 \\ \mathbf{v}_2 \\ \mathbf{w}_2 \\ \mathbf{u} \end{bmatrix} \begin{array}{l} \bmod q \\ \bmod q \\ \bmod q \\ \bmod q \\ \bmod p \end{array} . \tag{23}$$

Verification of a ciphertext $(\mathbf{v}_1, \mathbf{w}_1, \mathbf{v}_2, \mathbf{w}_2, \mathrm{c}, \mathbf{z})$ is done by verifying the zero-knowledge proof $(\mathrm{c}, \mathbf{z})$ for the language element (23). Decryption works exactly as in Algorithm 5, using $\mathbf{w}_1$ instead of $\mathbf{w}$.

Security. Correctness, completeness, and soundness all hold under the same assumptions as the CPA-secure scheme in Sect. 3.4. The following theorem states the chosen-ciphertext simulatability of the scheme. The proof can be found in the full version.

Theorem 5.1. *If the Ring-LWE encryption scheme is IND-CPA secure and the relaxed NIZK proof system is unbounded non-interactive zero-knowledge and unbounded simulation-sound, then the above construction is chosen-ciphertext simulatable.*

6 Applications

6.1 Key Escrow for Ring-LWE Encryption

A verifiable escrow scheme for decryption keys [YY98,PS00] allows a key owner to encrypt his private decryption key under the public key of a trusted authority so that anyone can check that the ciphertext is indeed an encryption of the private key corresponding to the owner's public key, but only the trusted authority can actually recover the private key. Intuitively, the owner is giving a proof that all messages sent to his public key can also be decrypted by the trusted third party. Note that a key escrow scheme cannot prevent parties from communicating securely, because even when forced to use escrowed keys, the parties can choose to double-encrypt messages under a non-escrowed key, or apply steganography to hide the fact that they are communicating altogether. The goal, therefore, is rather to prevent "dishonest" usage of public-key infrastructures, e.g., by using it to certify non-escrowed keys.

We show how our verifiable encryption scheme can be used to verifiably escrow Ring-LWE decryption keys. While, due to our relaxation of verifiable encryption, we cannot guarantee that the trusted authority recovers the actual decryption key, we show that whatever he recovers suffices to decrypt messages encrypted under the corresponding public key.

Let the authority have a Ring-LWE public key $\mathrm{t} = \mathrm{a}\mathrm{s}_1 + \mathrm{s}_2 \bmod q$ as described in Sect. 2.5. Users also have Ring-LWE encryption keys, but in R_p instead of R_q. Meaning, a secret key is a pair $(\mathrm{m}_1, \mathrm{m}_2) \xleftarrow{\$} S_1^2$, while the public key is $\mathrm{u} = \mathrm{b}\mathrm{m}_1 + \mathrm{m}_2 \bmod p$ for $\mathrm{b} \xleftarrow{\$} R_p$ together with a prime $p' < p$. Encryption and decryption work as in regular Ring-LWE, i.e., the sender chooses $\mathrm{r}, \mathrm{e}, \mathrm{e}' \xleftarrow{\$} S_1$ and computes

$$\begin{aligned} \mathrm{v} &= p'(\mathrm{b}\mathrm{r} + \mathrm{e}) \bmod p \\ \mathrm{w} &= p'(\mathrm{u}\mathrm{r} + \mathrm{e}') + \mu \bmod p \ . \end{aligned} \tag{24}$$

To decrypt, the receiver computes $\mu \leftarrow \mathrm{w} - \mathrm{v}\mathrm{s}_1 \bmod p \bmod p'$.

To escrow his decryption key, the key owner creates a verifiable encryption of his secret key $\mathbf{m} = [\ \mathrm{m}_1\ ;\ \mathrm{m}_2\ ]$ using our scheme from Sect. 3.2 under the authority's public t with a proof that $\mathbf{m}$ is a witness for the relation

$$[\ \mathrm{b}\ \ 1\ ] \begin{bmatrix} \mathrm{m}_1 \\ \mathrm{m}_2 \end{bmatrix} \ = \ \mathrm{u} \bmod p \ .$$

The soundness property of our relaxed verifiable encryption scheme guarantees that the authority can decrypt a witness $(\bar{\mathbf{m}}, \bar{\mathrm{c}})$ such that

$$\mathrm{b}\bar{\mathrm{m}}_1 + \bar{\mathrm{m}}_2 \ = \ \bar{\mathrm{c}}\mathrm{u} \bmod p \ .$$

The authority can decrypt an honestly generated ciphertext of the form (24) by computing

$$\begin{aligned}\bar{c}w - v\bar{m}_1 \bmod p &= \bar{c}p'(ur + e') + \bar{c}\mu - p'(br + e)\bar{m}_1 \bmod p \\ &= p'((b\bar{m}_1 + \bar{m}_2)r + \bar{c}e') + \bar{c}\mu - p'(b\bar{m}_1 r + e\bar{m}_1) \bmod p \\ &= p'(\bar{m}_2 r + \bar{c}e' - e\bar{m}_1) + \bar{c}\mu \bmod p\end{aligned}$$

from which μ can be recovered by reducing modulo p' and then dividing by $\bar{c}$ modulo p' (note that it is important that p' is chosen such that all differences of challenges in the challenge space are invertible), as long as the parameters are chosen such that $\|p'(\bar{m}_2 r + \bar{c}e' - e\bar{m}_1) + \bar{c}\mu\|_\infty < p/2$.

6.2 Verifiably Encrypted Signatures

Suppose two parties want to engage in a contract together and exchange signatures on the agreed contract. Neither of the parties wants to be the first to send his signature, however, fearing that the other party may not reciprocate and hold the first party liable to the conditions in the contract, without being held liable himself. Fair signature exchange protocols [ASW00, BDM98] ensure that no party can obtain a significant advantage over the other party by aborting the protocol early.

Verifiably encrypted signatures [ASW00, BDM98, BGLS03] are an important tool to build optimistic fair exchange protocols. The first party initially sends his signature encrypted under the key of a trusted adjudicator such that the other party can verify that the ciphertext indeed contains a valid signature on the agreed contract, but cannot recover the signature itself. The second party responds by sending his signature, after which the first party also sends over his signature. In case the first party refuses to send his signature in the last step, the second party can contact the adjudicator to have the encrypted signature from the first decrypted.

We show how our relaxed verifiable encryption scheme can be used to build verifiably encrypted signatures for the ring-based variant of Gentry-Peikert-Vaikuntanathan (Ring-GPV) signature scheme [GPV08] based on the hardness of the Ring-SIS or NTRU problems [SS11, DLP14]. Here, the signer's public key is a polynomial $b \in R_p$, while the secret key is a trapdoor allowing to find, for a given $u \in R_p$, short polynomials m_1, m_2 such that $bm_1 + m_2 = u$. A signature on a message μ in the usual scheme is a short vector (m_1, m_2) such that $bm_1 + m_2 = \mathsf{H}(\mu) \bmod p$, where $\mathsf{H} : \{0,1\}^* \to R_p$ is a random oracle. It is easy to show, however, that the scheme remains secure if one relaxes the verification algorithm to accept any tuple of short polynomials (m_1, m_2, c) such that $bm_1 + m_2 = c\mathsf{H}(\mu) \bmod p$.

In the usual security proof, when the adversary produces a forgery, $bm_1 + m_2 = \mathsf{H}(\mu) \bmod p$, the simulator already possesses another equality $bm'_1 + m'_2 = \mathsf{H}(\mu) \bmod p$, and thus obtains a solution to Ring-SIS as $b(m_1 - m'_1) + (m_2 - m'_2) = 0 \bmod p$. If, on the other hand, the adversary produces a forgery $bm_1 + m_2 = c\mathsf{H}(\mu) \bmod p$, then the simulator can obtain the equation $b(cm_1 - m'_1) + (cm_2 - m'_2) = 0 \bmod p$, which is still a (slightly longer) solution to Ring-SIS.

For this modified signature scheme that we build a verifiably encrypted signature scheme using our CCA-secure relaxed verifiable encryption scheme from Sect. 5. Namely, to encrypt an honest signature $(\mathrm{m}_1, \mathrm{m}_2, 1)$ under the adjudicator's public key, one encrypts the witness $\mathbf{m} = [\ \mathrm{m}_1\ ;\ \mathrm{m}_2\ ]$ with the encryption scheme from Sect. 5 while proving that $[\ \mathrm{b}\quad 1\ ]\ \mathbf{m} = \mathsf{H}(\mu) \bmod p$. When the adjudicator decrypts the signature, it recovers $(\bar{\mathbf{m}}, \bar{\mathrm{c}})$ such that $[\ \mathrm{b}\quad 1\ ]\ \bar{\mathbf{m}} = \bar{\mathrm{c}}\mathsf{H}(\mu) \bmod p$, which is also a valid signature on μ. Unforgeability follows from the unforgeability of the relaxed Ring-GPV scheme, while the security against extraction follows from the security of Ring-LWE encryption.

6.3 Other Applications

One of the most prominent applications of verifiable encryption is in group signatures [CvH91], where group members can sign anonymously in name of the entire group, but their anonymity can be lifted by a dedicated opening authority. A common construction paradigm [CL06, BCK+14] is to let a user's signing key consist of a signature by the group manager on the user's identity. To sign a message, the user encrypts his identity under the public key of the opener and creates a NIZK proof of knowledge of a valid signature for the encrypted identity. To recover the identity of the signer, the opener simply decrypts the ciphertext included in the signature.

Our verifiable encryption scheme could be very useful to group signatures in principle, what is missing is a *practical* signature scheme where the message $\mathbf{m}$ and the signature $\mathbf{s}$ are short vectors for which the verification equation can be expressed as a linear relation $\mathbf{B}\begin{bmatrix}\mathbf{m}\\ \mathbf{s}\end{bmatrix} = \mathbf{u}$.

7 Concrete Parameters

In this section we give some sample concrete instantiations of proofs of plaintext knowledge and verifiable encryption schemes (see Table 1). We express the

Table 1. Sample parameter sets for the verifiable encryption scheme

	I	II	III
n	1024	2048	2048
k	1	2	2
p	13	2^{15}	2^{30}
$\|\mathbf{m}\|_\infty$	1	1	2^{18}
σ	25344	50688	$\approx 2^{23.6}$
q	$\approx 2^{34}$	$\approx 2^{47}$	$\approx 2^{70}$
gamma factor	≈ 1.0046	≈ 1.0033	≈ 1.0052
proof size	9 KB	38 KB	54 KB
Ciphertext size	9 KB	48 KB	71 KB

security of each scheme in terms of the "gamma factor" from [GN08]. Values of 1.01 can be broken today, 1.007 seem to be fairly secure (conjectured at least 80-bits), and those less than 1.005 are believed to require more than 2^{128} time even for quantum computers. There have been some other attacks considered (e.g. in [ADPS16]), but those require as much memory as time, and are at this point not as useful in practice as variations of lattice-reduction attacks based on LLL (e.g. [CN11]). It is of course possible that attacks that are currently impractical can be made more practical, and at that point the concrete parameters (for all lattice-based schemes) would have to be adjusted. But the ratio between parameter sizes for verifiable encryption and regular encryption (and zero-knowledge authentication) should remain the same. One caveat would be if the algorithms for "overstretched" NTRU would become applicable to Ring-LWE. It was recently shown that when the modulus in NTRU is larger than the secrets by a sub-exponential (i.e. $2^{O(\sqrt{d})}$, where d is the lattice dimension) factor, then the NTRU problem becomes easy [ABD16, CJL16, KF16]. This is in contrast to LWE and Ring-LWE, for which efficient algorithms are only known in the case that the modulus is $2^{\omega(d)}$). If these attacks are transferred to the Ring-LWE setting, then this would have implications toward all constructions (e.g. those in this paper, most FHE schemes based on Ring-LWE, etc.) in which the secrets are significantly smaller than the modulus.

Our schemes are instantiated from the Ring-LWE cryptosystem where we take the secret and error parameters to be from the set $\{-1, 0, 1\}$. While the worst-case to average-case hardness of Ring-LWE (and LWE) was only proven with larger parameters [Reg09, LPR13], there haven't been any weaknesses found when constructing public-key encryption schemes with smaller errors. In particular, the part of the NTRU cryptosystem [HPS98] that is based on the Ring-LWE assumption has never been attacked due to having secret and error vectors having coefficients from the set $\{-1, 0, 1\}$. The most practical attacks is still to rewrite the Ring-LWE instance as a lattice problem and apply lattice reduction.

For all the parameter sets, we analyze the hardness of recovering the vector $[\mathbf{r};\ \mathbf{e};\ \mathbf{e}';\ \mathbf{m}]$ in (11). In column I, we give parameters for a proof of plaintext knowledge where there is no message $\mathbf{m}$. The exact parameters for verifiable encryption will of course depend on the parameters of the relation in (9). In columns II and III, we give the parameters that are large enough to instantiate the two example cases in Sect. 6. All the parameters are as defined in Algorithm 3 with the value of q is taken so as to satisfy (16) in the statement of Lemma 3.1 which is required for the decryption algorithm to function correctly. We point out that in the application to key escrow, there is also an encryption in the key escrow itself. But because that encryption works over modulus p, which is smaller than q, the hardness of breaking it is at least as hard as breaking the verifiable encryption scheme.

Acknowledgements. We would like to thank the anonymous reviewers for their detailed reviews and helpful feedback. This work was supported by the SNSF ERC Transfer Grant CRETP2-166734 – FELICITY and by the European Commission's PERCY grant (agreement #321310).

References

[ABD16] Albrecht, M., Bai, S., Ducas, L.: A subfield lattice attack on overstretched NTRU assumptions. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 153–178. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53018-4_6

[ADPS16] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - a new hope. In: USENIX, pp. 327–343 (2016)

[ASW00] Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures. IEEE J. Sel. Areas Commun. **18**(4), 593–610 (2000)

[BCK+14] Benhamouda, F., Camenisch, J., Krenn, S., Lyubashevsky, V., Neven, G.: Better zero-knowledge proofs for lattice encryption and their application to group signatures. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 551–572. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45611-8_29

[BDLN16] Baum, C., Damgård, I., Larsen, K.G., Nielsen, M.: How to prove knowledge of small secrets. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9816, pp. 478–498. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53015-3_17

[BDM98] Bao, F., Deng, R.H., Mao, W.: Efficient and practical fair exchange protocols with off-line TTP. In: IEEE Symposium on Security and Privacy, pp. 77–85 (1998)

[BGLS03] Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003). doi:10.1007/3-540-39200-9_26

[BR93] Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: CCS 1993, pp. 62–73 (1993)

[CD16] Cramer, R., Damgård, I.: Amortized complexity of zero-knowledge proofs revisited: achieving linear soundness slack. IACR Cryptology ePrint Archive, 2016:681 (2016)

[CJL16] Cheon, J.H., Jeong, J., Lee, C.: An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without an encoding of zero. IACR Cryptology ePrint Archive, 2016:139 (2016)

[CKY09] Camenisch, J., Kiayias, A., Yung, M.: On the portability of generalized Schnorr proofs. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 425–442. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01001-9_25

[CL06] Chase, M., Lysyanskaya, A.: On signatures of knowledge. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 78–96. Springer, Heidelberg (2006). doi:10.1007/11818175_5

[CN11] Chen, Y., Nguyen, P.Q.: BKZ 2.0: better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 1–20. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25385-0_1

[CS03] Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003). doi:10.1007/978-3-540-45146-4_8

[CvH91] Chaum, D., Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991). doi:10.1007/3-540-46416-6_22

[DLP14] Ducas, L., Lyubashevsky, V., Prest, T.: Efficient identity-based encryption over NTRU lattices. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 22–41. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45608-8_2

[DPSZ12] Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 643–662. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32009-5_38

[FKMV12] Faust, S., Kohlweiss, M., Marson, G.A., Venturi, D.: On the non-malleability of the Fiat-Shamir transform. In: Galbraith, S., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 60–79. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34931-7_5

[FS86] Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). doi:10.1007/3-540-47721-7_12

[GN08] Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 31–51. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78967-3_3

[GPV08] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)

[HPS98] Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998). doi:10.1007/BFb0054868

[KF16] Kirchner, P., Fouque, P.-A.: Comparison between subfield and straightforward attacks on NTRU. IACR Cryptology ePrint Archive 2016:717 (2016)

[LM06] Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006). doi:10.1007/11787006_13

[LN86] Lidl, R., Niederreiter, H.: Introduction to Finite Fields and their Applications. Cambridge University Press, Cambridge (1986)

[LNSW13] Ling, S., Nguyen, K., Stehlé, D., Wang, H.: Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 107–124. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36362-7_8

[LPR13] Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. J. ACM **60**(6), 43 (2013). Preliminary version appeared in EUROCRYPT 2010

[Lyu09] Lyubashevsky, V.: Fiat-shamir with aborts: applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer, Heidelberg (2009). doi:10.1007/978-3-642-10366-7_35

[Lyu12] Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29011-4_43

[NY90] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC, pp. 427–437 (1990)

[PS00] Poupard, G., Stern, J.: Fair encryption of RSA keys. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 172–189. Springer, Heidelberg (2000). doi:10.1007/3-540-45539-6_13

[Reg09] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (2009)

[Sah99] Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS, pp. 543–553 (1999)

[SS11] Stehlé, D., Steinfeld, R.: Making `NTRU` as secure as worst-case problems over ideal lattices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011). doi:10.1007/978-3-642-20465-4_4

[SSTX09] Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 617–635. Springer, Heidelberg (2009). doi:10.1007/978-3-642-10366-7_36

[Ste93] Stern, J.: A new identification scheme based on syndrome decoding. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 13–21. Springer, Heidelberg (1994). doi:10.1007/3-540-48329-2_2

[YY98] Young, A., Yung, M.: Auto-recoverable auto-certifiable cryptosystems. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 17–31. Springer, Heidelberg (1998). doi:10.1007/BFb0054114