


# Short Paper: Verifiable Decryption for BGV

Tjerand Silde 

Department of Mathematical Sciences  
Norwegian University of Science and Technology  
tjerand.silde@ntnu.no

**Abstract.** In this work we present a direct construction for verifiable decryption for the BGV encryption scheme by combining existing zero-knowledge proofs for linear relations and bounded values. This is one of the first constructions of verifiable decryption protocols for lattice-based cryptography, and we give a protocol that is simpler and at least as efficient as the state of the art when amortizing over many ciphertexts. To prove its practicality we provide concrete parameters, resulting in proof size of less than  $44\tau$  KB for  $\tau$  ciphertexts with message space 2048 bits. Furthermore, we provide an open source implementation showing that the amortized cost of the verifiable decryption protocol is only 76 ms per message when batching over  $\tau = 2048$  ciphertexts.

**Keywords:** lattice cryptography · verifiable decryption · zero-knowledge

## 1 Introduction

Many privacy preserving applications require one to prove that a ciphertext is correctly decrypted without revealing the secret key. This is called *verifiable decryption*, formalized by Camenisch and Shoup [CS03]. Example use-cases are electronic voting [Adi08], mixing networks [HM20], DC-networks [CWF13] and fully homomorphic encryption [LW18]. These applications usually require decrypting a large number of ciphertexts.

Unfortunately, the above systems are either not secure against quantum computers or very inefficient. Recent works in lattice-based cryptography are leading towards voting protocols achieving security even against quantum adversaries, see, e.g., the shuffles by Aranha *et al.* [ABG<sup>+</sup>21], Costa *et al.* [CMM19] and Farzaliyev *et al.* [FWK21]. However, few constructions provides verifiable decryption for lattice-based encryption schemes.

### 1.1 Contribution

We present a new and efficient verifiable decryption protocol for batches of ciphertext using the lattice-based encryption scheme by Brakerski, Gentry and Vaikuntanathan [BGV12]. The protocol is direct; the decryption procedure consists of computing a linear equation involving the ciphertext and the key, and

then the message is extracted by rounding the result modulo the plaintext modulus. This procedure gives the correct result if the noise-level in the ciphertext is bounded. We use lattice-based commitments to commit to the secret key, and then we prove two relations in zero-knowledge: 1) we prove that the linear equation holds with respect to a fresh commitment to the ciphertext-noise, and 2) prove that the noise is bounded. Together, this leads to an efficient verifiable decryption protocol. We give concrete parameters and estimate the size in Sec. 4.1 and give timings from our proof-of-concept implementation in Sec. 4.2.

## 1.2 Related Work

We compare to the works on verifiable decryption for lattices by Lyubashevsky *et al.* [LNS21], Gjøsteen *et al.* [GHM<sup>+</sup>22] and Boschini *et al.* [BCOS20] in Sec 4.3.

## 2 Lattice-Based Cryptography

Let  $N$  be a power of 2 and  $q = 1 \pmod{2N}$  a prime. We define the ring  $R_q = \mathbb{Z}_q[X]/\langle X^N + 1 \rangle$ . For  $f \in R_q$  we choose coefficients as the representatives in  $[-\frac{q-1}{2}, \frac{q-1}{2}]$ , and compute inner products  $\langle \cdot, \cdot \rangle$  and norms as vectors over  $\mathbb{Z}$ :

$$\|f\|_1 = \sum |\alpha_i|, \quad \|f\|_2 = \left( \sum \alpha_i^2 \right)^{1/2}, \quad \|f\|_\infty = \max\{|\alpha_i|\}.$$

We furthermore define the sets  $S_{\beta_\infty} = \{x \in R_q \mid \|x\|_\infty \leq \beta_\infty\}$  as well as

$$\mathcal{C} = \{c \in R_q \mid \|c\|_\infty = 1, \|c\|_1 = \nu\} \text{ and } \bar{\mathcal{C}} = \{c - c' \mid c \neq c' \in \mathcal{C}\}.$$

### 2.1 Rejection Sampling

We want to output vectors  $\mathbf{z} = \mathbf{y} + \mathbf{v}$  such that  $\mathbf{z}$  is independent of  $\mathbf{v}$ , and hence,  $\mathbf{v}$  is masked by the vector  $\mathbf{y}$ . If  $\mathbf{y}$  is sampled according to a Gaussian distribution  $\mathcal{N}_\sigma^k$  with standard deviation  $\sigma$ , then we want  $\mathbf{z}$  to be from the same distribution.  $1/M$  is the success probability for rejection sampling, and  $M$  is computed as

$$\max \frac{\mathcal{N}_\sigma^k(\mathbf{z})}{\mathcal{N}_{\mathbf{v},\sigma}^k(\mathbf{z})} = \exp \left[ \frac{-2\langle \mathbf{z}, \mathbf{v} \rangle + \|\mathbf{v}\|_2^2}{2\sigma^2} \right] \leq \exp \left[ \frac{24\sigma\|\mathbf{v}\|_2 + \|\mathbf{v}\|_2^2}{2\sigma^2} \right] = M,$$

so that  $|\langle \mathbf{z}, \mathbf{v} \rangle| < 12\sigma\|\mathbf{v}\|_2$  with probability at least  $1 - 2^{-100}$ . Hence, for  $\sigma = 11\|\mathbf{v}\|_2$ , we get  $M \approx 3$ . This is the standard way to choose parameters. If the procedure is only done once for the vector  $\mathbf{v}$ , we can decrease the parameters, to the cost of leaking only one bit of information about  $\mathbf{v}$  from the given  $\mathbf{z}$ .

Lyubashevsky *et al.* [LNS21] suggest to require that  $\langle \mathbf{z}, \mathbf{v} \rangle \geq 0$ . Then we can set  $M = \exp(\|\mathbf{v}\|_2/2\sigma^2)$ . For  $\sigma = 0.675\|\mathbf{v}\|_2$ , we get  $M \approx 3$ , with the effect of rejecting about half of the vectors up front. See [LNS21, Figure 2] for details.

## 2.2 Hardness Assumptions

We first define the Search Knapsack problem in the  $\ell_2$  norm, also denoted as  $\text{SKS}^2$ . The  $\text{SKS}^2$  problem is the Ring-SIS problem in its Hermite Normal Form.

**Definition 1.** *The  $\text{SKS}_{N,q,\beta}^2$  problem is to find a short vector  $\mathbf{x}$  of  $\ell_2$  norm less than or equal to  $\beta$  in  $R_q^2$  satisfying  $[a \ 1] \cdot \mathbf{x} = 0$  for a given uniformly random  $a$  in  $R_q$ . An algorithm  $\mathcal{A}$  has advantage  $\epsilon$  in solving the  $\text{SKS}_{N,q,\beta}^2$  problem if*

$$\Pr \left[ \begin{array}{l} [a \ 1] \cdot \mathbf{x} = 0 \\ \wedge \ \|x_i\|_2 \leq \beta \end{array} \mid \begin{array}{l} a \leftarrow_{\$} R_q; \\ \mathbf{0} \neq \mathbf{x} \in R_q^2 \leftarrow \mathcal{A}(a) \end{array} \right] \geq \epsilon.$$

We also define the Decisional Knapsack problem ( $\text{DKS}^\infty$ ) in the  $\ell_\infty$  norm.  $\text{DKS}^\infty$  is equivalent to the Ring-LWE problem when the number of samples is limited.

**Definition 2.** *The  $\text{DKS}_{N,q,\beta_\infty}^\infty$  problem is to distinguish the distribution  $[a \ 1] \cdot \mathbf{x}$ , for a short  $\mathbf{x}$ , from the uniform distribution when given uniformly random  $a$  in  $R_q$ . An algorithm  $\mathcal{A}$  has advantage  $\epsilon$  in solving the  $\text{DKS}_{N,q,\beta_\infty}^\infty$  problem if*

$$\begin{aligned} & |\Pr[b = 1 \mid a \leftarrow_{\$} R_q; \mathbf{x} \leftarrow_{\$} S_{\beta_\infty}; b \leftarrow \mathcal{A}(a, [a \ 1] \cdot \mathbf{x})] \\ & - \Pr[b = 1 \mid a \leftarrow_{\$} R_q; u \leftarrow_{\$} R_q; b \leftarrow \mathcal{A}(a, u)]| \geq \epsilon. \end{aligned}$$

See [LM06, LPR10] for more details about knapsack problems over rings.

## 2.3 BGV Encryption

Let  $p \ll q$  be primes, let  $R_q$  and  $R_p$  be defined as above for a fixed  $N$ , let  $\mathcal{D}$  be a bounded distribution over  $R_q$ , let  $\beta_\infty \in \mathbb{N}$  be a bound and let  $\lambda$  be the security parameter. The (plain) BGV encryption scheme [BGV12] consists of three algorithms: key generation ( $\text{KGen}$ ), encryption ( $\text{Enc}$ ) and decryption ( $\text{Dec}$ ), where

- $\text{KGen}$  samples  $a \leftarrow_{\$} R_q$  uniformly at random, samples a short  $s \leftarrow_{\$} S_{\beta_\infty}$  and samples noise  $e \leftarrow \mathcal{D}$ . It outputs keys  $\text{pk} = (a, b) = (a, as + pe)$  and  $\text{sk} = s$ .
- $\text{Enc}$ , on input  $\text{pk}$  and a message  $m$  in  $R_p$ , samples a short  $r \leftarrow_{\$} S_{\beta_\infty}$ , samples noise  $e', e'' \leftarrow \mathcal{D}$ , and outputs ciphertext  $c = (u, v) = (ar + pe', br + pe'' + m)$ .
- $\text{Dec}$ , on input  $\text{sk} = s$  and  $c = (u, v)$ , outputs  $m = (v - su \bmod q) \bmod p$ .

The decryption is correct if  $\max\|v - su\|_\infty = B_{\text{Dec}} < \lfloor q/2 \rfloor$ . The encryption scheme is CPA-secure if the  $\text{DKS}_{N,q,\beta}^\infty$  problem is hard for some  $\beta = \beta(N, q, p, \beta_\infty)$ .

## 2.4 Lattice-Based Commitments

Let  $\mathcal{N}_{\sigma_C}$  be a Gaussian distribution over  $R_q$  with standard deviation  $\sigma_C$ . The commitment scheme by Baum *et al.* [BDL<sup>+</sup>18] consists of three algorithms: key generation ( $\text{KGen}$ ), committing ( $\text{Com}$ ) and opening ( $\text{Open}$ ), where

- KGen outputs a public key  $\text{pk}$  to commit to messages in  $R_q$ . We define

$$\begin{aligned} \mathbf{A}_1 &= [\mathbf{I}_n \quad \mathbf{A}'_1] && \text{where } \mathbf{A}'_1 \leftarrow_{\$} R_q^{n \times (k-n)} \\ \mathbf{a}_2 &= [\mathbf{0}^n \quad 1 \quad \mathbf{a}'_2] && \text{where } \mathbf{a}'_2 \leftarrow_{\$} R_q^{(k-n-1)}, \end{aligned}$$

for height  $n + 1$  and width  $k$  and let  $\text{pk}$  be  $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{a}_2 \end{bmatrix}$ .

- Com commits to messages  $m \in R_q$  by sampling  $\mathbf{r}_m \leftarrow_{\$} S_{\beta_\infty}$ , and computes

$$\text{Com}_{\text{pk}}(m; \mathbf{r}_m) = \mathbf{A} \cdot \mathbf{r}_m + \begin{bmatrix} \mathbf{0} \\ m \end{bmatrix} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \llbracket m \rrbracket.$$

Com outputs commitment  $\llbracket m \rrbracket$  and opening  $\mathbf{d} = (m, \mathbf{r}_m, 1)$ .

- Open verifies whether  $(m, \mathbf{r}_m, f)$ , with  $f \in \mathcal{C}$ , is a valid opening of  $\llbracket m \rrbracket$  with respect to  $\text{pk}$  by checking that  $\|\mathbf{r}_m[i]\|_2 \leq 4\sigma_C \sqrt{N}$ , for  $i \in [k]$ , and if

$$f \cdot \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} \stackrel{?}{=} \mathbf{A} \cdot \mathbf{r}_m + f \cdot \begin{bmatrix} \mathbf{0} \\ m \end{bmatrix}.$$

Open outputs 1 if all these conditions holds, and 0 otherwise.

The commitment scheme is hiding if the  $\text{DKS}_{N,q,\beta_\infty}^\infty$  problem is hard and it is binding if the  $\text{SKS}_{N,q,16\sigma_C\sqrt{N}}^2$  problem is hard, see [BDL<sup>+</sup>18, Section 4].

## 2.5 Zero-Knowledge Proof of Linear Relations

Let  $\llbracket y \rrbracket, \llbracket y' \rrbracket$  be commitments such that  $y' = \alpha y + \beta$  for some public values  $\alpha, \beta \in R_q$ . The protocol  $\Pi_{\text{LIN}}$  in [ABG<sup>+</sup>21, Figure 1] is a zero-knowledge proof of knowledge, with  $\ell_2$  bound  $B_C = 2\sigma_C \sqrt{N}$  on the responses  $\mathbf{z}_i$ , for the relation:

$$\mathcal{R}_{\text{Lin}} = \left\{ (x, w) \mid \begin{array}{l} x = (\alpha, \beta, \llbracket y \rrbracket, \llbracket y' \rrbracket), w = (y, \mathbf{r}_y, \mathbf{r}_{y'}, f, f') : \\ \text{Open}(\llbracket y \rrbracket, y, \mathbf{r}_y, f) = \text{Open}(\llbracket y' \rrbracket, \alpha \cdot y + \beta, \mathbf{r}_{y'}, f') = 1 \end{array} \right\}$$

When applying the Fiat-Shamir transform [FS87], we let the challenge  $c \in \mathcal{C}$  be the output of a hash function applied to the full transcript. Then, we get the proof  $\pi_L = (c, \mathbf{z}_1, \mathbf{z}_2)$ , where each  $\mathbf{z}_i$  is of size  $kN \log_2(6\sigma_C)$  bits. We can compress each  $\mathbf{z}_i$  to get a proof of total size  $2(k-n)N \log_2(6\sigma_C)$  bits by checking an approximate equality instead [ABG<sup>+</sup>21, Section 3.2]. We denote by

$$\begin{aligned} \pi_L &\leftarrow \Pi_{\text{LIN}}((y, \mathbf{r}_y, \mathbf{r}_{y'}, f_y, f_{y'}); (\alpha, \beta, \llbracket y \rrbracket, \llbracket y' \rrbracket)), \text{ and} \\ 0 \vee 1 &\leftarrow \Pi_{\text{LINV}}((\alpha, \beta, \llbracket y \rrbracket, \llbracket y' \rrbracket); \pi_L), \end{aligned}$$

the run of the proof and verification protocols, respectively, where the verification protocol  $\Pi_{\text{LINV}}$  performs the checks as in the last step in [ABG<sup>+</sup>21, Figure 1] and also verifies that  $c$  was computed correctly with respect to the transcript.  $\Pi_{\text{Lin}}$  is a sound proof of knowledge in the ROM if the  $\text{SKS}_{N,q,2B_C}^2$  problem is hard.

## 2.6 Amortized Zero-Knowledge Proof of Bounded Openings

Let  $\mathbf{A}$  be a publicly known  $r \times v$ -matrix over  $R_q$ , let  $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_\tau$  be bounded elements in  $R_q^v$  and let  $\mathbf{A}\mathbf{s}_i = \mathbf{t}_i$  for  $i \in [\tau]$ . Letting  $\mathbf{S}$  be the matrix whose columns are  $\mathbf{s}_i$  and  $\mathbf{T}$  be the equivalent matrix for  $\mathbf{t}_i$ , Baum *et al.* [BBC<sup>+</sup>18] give a efficient amortized zero-knowledge proof of knowledge for the relation:

$$\mathcal{R}_A = \left\{ (x, w) \mid \begin{array}{l} x = (\mathbf{A}, \mathbf{T}), w = \mathbf{S}: \\ \forall i \in [\tau] : \mathbf{t}_i = \mathbf{A}\mathbf{s}_i \wedge \|\mathbf{s}_i\|_2 \leq 2 \cdot B_A \end{array} \right\}$$

The protocol  $\Pi_A$  is depicted in [BBC<sup>+</sup>18, Figure 1]. We use a challenge matrix  $\mathbf{C}$  with entries sampled from  $\mathcal{C}_A = \{0, 1\}$ . For security parameter  $\lambda$ , we define the number of parallel protocol instances to be  $\hat{n} = \lambda + 2$ . Denote by

$$\pi_A \leftarrow \Pi_A(\mathbf{S}; (\mathbf{A}, \mathbf{T})), \text{ and } 0 \vee 1 \leftarrow \Pi_{AV}((\mathbf{A}, \mathbf{T}); \pi_A),$$

the run of the proof and verification protocols, respectively, where the  $\Pi_A$ -protocol, using Fiat-Shamir, produces a proof of the form  $\pi_A = (\mathbf{C}, \mathbf{Z})$ , where  $\mathbf{C}$  is the output of a hash-function applied to the full transcript, and the  $\Pi_{AV}$ -protocol consists of the two checks in the last step in [BBC<sup>+</sup>18, Figure 1]. The verification bound on each column of  $\mathbf{Z}$  is  $B_A = \sqrt{2N}\sigma_A$ . Note that  $\sigma_A$ , and also  $B_A$ , depends on the norm of  $\mathbf{S}$  (see rejection sampling in Section 2.1). Hence, the bound we can prove depends on the number of equations in the statement.  $\Pi_A$  is a sound proof of knowledge in the ROM if  $\text{SKS}_{N,q,2B_A}^2$  is hard.

## 3 The Verifiable Decryption Protocol

The protocol is direct. The prover starts by decrypting the ciphertext  $(u, v)$  to obtain the underlying plaintext  $m$  as  $m = (v - us \bmod q) \bmod p$ . Then, he commits to the noise  $d (= er + e'' - se')$  in the ciphertexts as  $\llbracket d \rrbracket$ . Finally, he proves two statements in zero-knowledge: 1) the linear relation  $p\llbracket d \rrbracket = v - m - u\llbracket s \rrbracket$  holds modulo  $q$  with respect to the noise and a public commitment to the secret key, and 2) the value committed to in  $\llbracket d \rrbracket$  is shorter than some bound  $B < q/2p$ .

More precisely, we present a proof protocol for the following relation:

$$R_{\text{DEC}} = \left\{ (x, w) \mid \begin{array}{l} x = ((a, b), \llbracket s \rrbracket, (u_1, v_1), \dots, (u_\tau, v_\tau), m_1, \dots, m_\tau), \\ w = (s, \mathbf{r}_s, f_s) \text{ such that } \text{Open}(\llbracket s \rrbracket; s, \mathbf{r}_s, f_s) = 1 \\ \wedge \forall i \in [\tau] : pd_i = v_i - m_i - u_i s \wedge \|d_i\|_\infty < q/2p. \end{array} \right\}$$

Here, we assume that either a trusted dealer generated the public key and secret key together with a commitment to the secret key, or that the prover already has proved in zero-knowledge that the public key is well formed and that the secret key is committed to in  $\llbracket s \rrbracket$ , using any exact proof from the literature.

The verifiable decryption protocol  $\Pi_{\text{DEC}}$ , for prover  $\mathcal{P}$ , goes as following:

1.  $\mathcal{P}$  takes as input a set of ciphertexts  $(u_1, v_1), \dots, (u_\tau, v_\tau)$  and  $(\llbracket s \rrbracket, s, \mathbf{r}_s, f_s)$ .
2.  $\mathcal{P}$  runs  $\text{Dec}$  on input  $s$  and  $(u_i, v_i)$  for all  $i \in [\tau]$  to obtain messages  $m_1, \dots, m_\tau$ .

3.  $\mathcal{P}$  extracts noise  $d_i$  by computing  $d_i = (v_i - m_i - u_i s)/p \pmod q$  for all  $i \in [\tau]$ .
4.  $\mathcal{P}$  commits to all  $d_i$  as  $\llbracket d_i \rrbracket$ , and proves  $p\llbracket d_i \rrbracket = v_i - m_i - u_i \llbracket s \rrbracket$  using  $\Pi_{\text{LIN}}$ .
5.  $\mathcal{P}$  uses protocol  $\Pi_A$  to prove that all  $\|d_i\|_2$  are bounded by  $B_A \leq \sqrt{2vN}\sigma_A$ .
6.  $\mathcal{P}$  outputs messages  $\{m_i\}_{i=1}^\tau$ , commitments  $\{\llbracket d_i \rrbracket\}_{i=1}^\tau$  and proofs  $\{\pi_{L_i}\}_{i=1}^\tau, \pi_A$ .

A verifier  $\mathcal{V}$  runs the verification protocol  $\Pi_{\text{DECV}}$  which checks that all proofs  $\{\pi_{L_i}\}_{i=1}^\tau$  and  $\pi_A$  are valid with respect to  $(a, b)$ ,  $\{(u_i, v_i)\}_{i=1}^\tau$  and  $\{m_i\}_{i=1}^\tau$ .

**Theorem 1.** *The verifiable decryption protocol  $\Pi_{\text{DEC}}$  is a complete, sound and zero-knowledge proof protocol in the ROM for relation  $R_{\text{DEC}}$  when  $B_A < q/(4p)$ .*

*Proof.* We argue each of the properties as following:

*Completeness.* It follows directly that  $\Pi_{\text{DEC}}$  is complete if the encryption scheme is correct, which is the case when  $\|v - su\| < q/2$ , and the protocols  $\Pi_{\text{LIN}}$  and  $\Pi_A$  are complete. Hence, we only need to make sure that  $\|v - su\| < q/2$ . The protocol  $\Pi_A$  guarantees that the noise is bounded as  $\|d_i\|_2 \leq 2B_A$ . It follows that if  $B_A < q/(4p)$  then  $\|d_i\|_\infty < q/2p$ , and the decryption is correct.

*Special soundness.* The soundness of the protocol follows directly from the underlying zero-knowledge protocols  $\Pi_{\text{LIN}}$  and  $\Pi_A$ . With the use of rewinding we can either extract the secret key  $s$  or the noise  $d_i$  (which reveals the secret key) or some short vectors breaking the  $\text{SKS}^2$  problem for the given parameters.

*Honest-verifier zero-knowledge.* The zero-knowledge property follows directly from the underlying zero-knowledge protocols  $\Pi_{\text{LIN}}$  and  $\Pi_A$ , which are both honest-verifier zero-knowledge. Hence, with input messages  $m_1, \dots, m_\tau$  we can simulate the decryption proof by sampling uniformly random values  $d_i$ , committing to them as  $\llbracket d_i \rrbracket$  and then simulating all the proofs  $\pi_{L_i}$  and  $\pi_A$  subsequently.  $\square$

## 4 Performance

### 4.1 Parameters and Size

From the verifiable decryption protocol in Section 3 we get that the statement consists of  $\tau$  ciphertexts  $(u_i, v_i)$  and messages  $m_i$ . Each element  $u_i$  and  $v_i$  are uniformly elements in  $R_q$  of size  $N \log_2 q$  bits each. The messages are elements in  $R_q$  with coordinates modulo  $p$ , and hence, are of size  $N \log_2 p$  bits. Each proof  $\pi_L$  are of size  $2(k-n)N \log_2(6\sigma_C)$  bits, for  $\sigma_C = 11\nu\beta_\infty\sqrt{kN}$ , and the proof  $\pi_A$  is of size  $(k+1)\hat{n}N \log_2(6\sigma_A)$  bits. However, the norm bound  $B_A$  depends on the number of equations being proved at once, and hence, if  $\tau$  is large it is beneficial to prove smaller batches, e.g., of size  $N$ , instead of all equations at once.

As a concrete example, we set  $p = 2$ ,  $\beta_\infty = 1$  and let  $\mathbb{D}$  be the ternary distribution over  $R_q$ . It then follows that, for honestly generated ciphertexts,

$p$	$q$	$N$	$\beta_\infty$	$M$	$k$	$n$	$\hat{n}$	$\nu$	$\sigma_C$	$B_C$	$\sigma_A$	$B_A$
2	$\approx 2^{44}$	2048	1	3	3	1	130	36	$\approx 2^{15.9}$	$\approx 2^{22.4}$	$\approx 2^{28}$	$\approx 2^{40.5}$

**Table 1.** Example parameters for the verifiable decryption protocol with more than 128 bits of security against quantum adversaries ensuring correct decryption for honestly generated ciphertexts. Rejection sampling success probability is set to be  $\approx 1/3$ .

$\|v - us\|_\infty \leq p(2N + 1)$ . Furthermore, we get the following bound for  $\|d_i\|_\infty$ :

$$\begin{aligned} \|d_i\|_\infty &\leq 2B_A = \sqrt{8N}\sigma_A \leq \sqrt{8N} \cdot 0.675 \|S' C'\|_2 \leq 2\sqrt{N} \cdot \|S'\|_1 \|C'\|_\infty \\ &\leq 2\sqrt{N} \cdot (\beta_\infty kN + p(2N + 1)N)\tau. \end{aligned}$$

Thus, setting  $k = 3, n = 1, \hat{n} = 130, \nu = 36$  and  $\tau = N = 2048$  gives us  $\|d_i\|_\infty \leq 2^{41.5}$ , and we can safely set  $q \approx 2^{44}$  to get correctness. We claim more than 128 bits security against a quantum adversary for these parameters using the LWE estimator by Albrecht *et al.* [APS15] with the **BKZ.qsieve** cost model. A smaller  $N$  results in smaller noise, but the size of  $q$  would give lower security.

Message $m_i$	Ciphertext $(u_i, v_i)$	Commitment $\llbracket d_i \rrbracket$	Proof $\pi_{L_i}$	Proof $\pi_A$	Proof $\pi_{DEC}$
0.256 KB	22.6 KB	22.6 KB	19 KB	2 $\tau$ KB	43.6 $\tau$ KB

**Table 2.** Sizes for parameters  $p = 2, q \approx 2^{44}$  and  $N = 2048$  computing proof  $\pi_{DEC} = (\{\llbracket d_i \rrbracket, \pi_{L_i} \}_{i=1}^\tau, \pi_A)$ , where shortness proofs  $\pi_A$  is amortized over batches of size 2048.

## 4.2 Implementation and Timings

We provide a proof-of-concept implementation of our protocol in C++ using the NTL-library [Sho21]. The implementation was benchmarked on an Intel Core i5 running at 2.3GHz with 16 GB RAM. The timings are given in Table 3. The implementation is very simple, consists of a total of 350 lines of code, and is available online\*. A comparison of NTL to NFLlib [ABG+16] indicates that an optimized implementation could provide speedup by an order of magnitude.

Noise $\llbracket d_i \rrbracket$	Proof $\Pi_{LIN}$	Verification $\Pi_{LINV}$	Proof $\Pi_A$	Verification $\Pi_{AV}$	Proof $\pi_{DEC}$
5 $\tau$ ms	47 $\tau$ ms	12 $\tau$ ms	24 $\tau$ ms	12 $\tau$ ms	76 $\tau$ ms

**Table 3.** Amortized time per instance over  $\tau = 2048$  ciphertexts.

## 4.3 Comparison

We compare to the verifiable decryption protocols by Lyubashevsky *et al.* [LNS21] and Gjøsteen *et al.* [GHM+22]. As noted by [GHM+22, Section 8], the protocol by Boschini *et al.* [BCOS20] give proof sizes of approximate 90 KB, which is roughly twice the size of  $\pi_{DEC}$ . Furthermore, the run time is several minutes per ciphertext, which would deem it unusable for larger sets of ciphertexts.

\* [github.com/tjesi/verifiable-decryption-BGV](https://github.com/tjesi/verifiable-decryption-BGV).

**Comparison to Lyubashevsky *et al.* (PKC 2021).** They give a verifiable decryption protocol for the Kyber encapsulation scheme for a ring of dimension  $N = 256$  and modulus  $q = 3329$  with secret and noise values bounded by  $\beta_\infty = 2$ . The proof of correct decryption is of size 43.6 KB. We observe that our proof is of exactly the same size but with a plaintext space of 2048 bits instead of only 256 bits. We expect our proof size to be smaller than theirs for ciphertexts encoding larger messages, but note that they can provide efficient proofs for single ciphertexts for small moduli while our protocol is only efficient in the amortized setting for ciphertext moduli at least 44 bits. Furthermore, our protocol is much simpler, as [LNS21] make use of partially splitting rings and automorphisms by combining proofs of multiplication and range proofs – making the protocol difficult to implement in practice. They do not provide timings.

**Comparison to Gjøsteen *et al.* (ACISP 2022).** They give a verifiable decryption protocol  $\Pi_{\text{ZKPCD}}$  for the BGV encryption scheme. However, because of their noise drowning techniques, they are forced to use a moduli of at least  $q \approx 2^{55}$ . Their proof size is also depending on the soundness parameter  $\lambda$ , giving a proof of size  $14\lambda$  KB per ciphertext. For an interactive protocol with  $\lambda = 10$  they get a proof of size  $3.2\times$  larger than our proof, and for a non-interactive protocol with  $\lambda = 128$  their proof size is  $41\times$  larger than ours. They have implemented their protocol, and give a cost of at least  $4\lambda$  ms per ciphertext using NTL, which is similar to our protocol for  $\lambda = 19$  and otherwise slower.

They also sketch a protocol  $\Pi_{\text{DistDec}}$  [GHM<sup>+</sup>22, Section 8], requiring  $q \approx 2^{78}$  and  $N = 4096$ . This protocol gives a proof of size  $\approx 363$  KB per ciphertext, a factor 8 larger than our proof. They do not provide timings for this protocol.

## References

- ABG<sup>+</sup>16. Carlos Aguilar Melchor, Joris Barrier, Serge Guelton, Adrien Guinet, Marc-Olivier Killijian, and Tancrede Lepoint. NFLlib: NTT-based fast lattice library. In Kazuo Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 341–356. Springer, Heidelberg, February / March 2016.
- ABG<sup>+</sup>21. Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, Tjerand Silde, and Thor Tunge. Lattice-based proof of shuffle and applications to electronic voting. In Kenneth G. Paterson, editor, *CT-RSA 2021*, volume 12704 of *LNCS*, pages 227–251. Springer, Heidelberg, May 2021.
- Adi08. Ben Adida. Helios: Web-based open-audit voting. In Paul C. van Oorschot, editor, *USENIX Security 2008*, pages 335–348. USENIX Association, July / August 2008.
- APS15. Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 2015.
- BBC<sup>+</sup>18. Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 669–699. Springer, Heidelberg, August 2018.
- BCOS20. Cecilia Boschini, Jan Camenisch, Max Ovsiankin, and Nicholas Spooner. Efficient post-quantum SNARKs for RSIS and RLWE and their applications to privacy. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum*



- Cryptography - 11th International Conference, PQCrypto 2020*, pages 247–267. Springer, Heidelberg, 2020.
- BDL<sup>+</sup>18. Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 368–385. Springer, Heidelberg, September 2018.
- BGV12. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012.
- CMM19. Núria Costa, Ramiro Martínez, and Paz Morillo. Lattice-based proof of a shuffle. In Andrea Bracciali, Jeremy Clark, Federico Pintore, Peter B. Rønne, and Massimiliano Sala, editors, *FC 2019 Workshops*, volume 11599 of *LNCS*, pages 330–346. Springer, Heidelberg, February 2019.
- CS03. Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 126–144. Springer, Heidelberg, August 2003.
- CWF13. Henry Corrigan-Gibbs, David Isaac Wolinsky, and Bryan Ford. Proactively accountable anonymous messaging in verdict. In Samuel T. King, editor, *USENIX Security 2013*, pages 147–162. USENIX Association, August 2013.
- FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO’86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.
- FWK21. Valeh Farzaliyev, Jan Willemsen, and Jaan Kristjan Kaasik. Improved lattice-based mix-nets for electronic voting. *Cryptology ePrint Archive*, Report 2021/1499, 2021. <https://ia.cr/2021/1499>.
- GHM<sup>+</sup>22. Kristian Gjøsteen, Thomas Haines, Johannes Müller, Peter Rønne, and Tjerand Silde. Verifiable decryption in the head. *ACISP*, 2022. <https://eprint.iacr.org/2021/558.pdf>.
- HM20. Thomas Haines and Johannes Müller. SoK: Techniques for verifiable mix nets. In Limin Jia and Ralf Küsters, editors, *CSF 2020 Computer Security Foundations Symposium*, pages 49–64. IEEE Computer Society Press, 2020.
- LM06. Vadim Lyubashevsky and Daniele Micciancio. Generalized compact Knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 144–155. Springer, Heidelberg, July 2006.
- LNS21. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 215–241. Springer, Heidelberg, May 2021.
- LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*. Springer, Heidelberg, May / June 2010.
- LW18. Fucai Luo and Kumpeng Wang. Verifiable decryption for fully homomorphic encryption. In Liqun Chen, Mark Manulis, and Steve Schneider, editors, *ISC 2018*, volume 11060 of *LNCS*, pages 347–365. Springer, Heidelberg, September 2018.
- Sho21. Victor Shoup. Ntl: A library for doing number theory, 2021. <https://libntl.org/index.html>.